BIXOLON

API Reference Guide

Xamarin SDK

Ver. 1.25

www.bixolon.com

Copyright

© BIXOLON Co., Ltd. All rights reserved.

This user manual and all property of the product are protected under copyright law.
It is strictly prohibited to copy, store, and transmit the whole or any part of the manual and any property of the product without the prior written approval of BIXOLON Co., Ltd.
The information contained herein is designed only for use with this BIXOLON product.
BIXOLON is not responsible for any direct or indirect damages, arising from or related to use of this information.

• The BIXOLON logo is the registered trademark of BIXOLON Co., Ltd.
• All other brand or product names are trademarks of their respective companies or organizations.

BIXOLON Co., Ltd. maintains ongoing efforts to enhance and upgrade the functions and quality of all our products.
In the following, product specifications and/or user manual content may be changed without prior notice.

Caution

Some semiconductor devices are easily damaged by static electricity. You should turn the printer “OFF”, before you connect or remove the cables on the rear side, in order to guard the printer against the static electricity. If the printer is damaged by the static electricity, you should turn the printer “OFF”.

Table of Contents

Copyright **2**

1. Manual Guide **8**

2. Supported Devices **9**

3. Xamarin SDK Installation **13**
3-1 Supported Platforms 13
3-1-1 iOS 13
3-1-2 Android 13
3-1-3 Windows 13
3-2 Supported Communications 14
3-2-1 Bluetooth 14
3-2-2 WiFi / Ethernet 14
3-2-3 USB 14
3-2-4 BLE 14
3-3 Installation Instruction 17
3-3-1 Installation steps from NuGet server 17
3-3-2 Installation steps from a local nupkg file 18

4. IMPos DeviceFactory Reference **21**
4-1 Over view 21
4-2 Methods 21
4-2-1 Create Device 21

5. MPos Controller Devices Reference **22**
5-1 Over view 22
5-2 Properties 22
5-2-1 Data Event 22
5-2-2 Status Update Event 22
5-2-3 Device ID 23
5-2-4 Is Open 23
5-2-5 Command Mode 23
5-2-6 Interface Type 23
5-3 Methods 24
5-3-1 Select Interface 24
5-3-2 Select Command Mode 26
5-3-3 Open Service 27
5-3-4 Close Service 28
5-3-5 Direct IO 29
5-3-6 Set Transaction 30

6. MPos Controller Printer Reference 31
6-1 Over view 31
6-2 Methods 32
6-2-1 Print Text 32
6-2-2 Print Text 33
6-2-3 Set Character set 34
6-2-4 Set International Characters et 36
6-2-5 Set Page mode 37
6-2-6 Set Pagemode Print Area 38
6-2-7 Set Pagemode Direction 39
6-2-8 Set Pagemode Position 40
6-2-9 Get Firmware Version 41
6-2-10 Print Bitmap 42
6-2-11 Print Bitmap File 43
6-2-12 Print 1D Barcode 44
6-2-13 Print QR Code 46
6-2-14 Print PDF 417 47
6-2-15 Check Printer Status 48
6-2-16 Check Batt Status 49
6-2-17 Asb Enable 50
6-2-18 Print Line 51
6-2-19 Print Box 52
6-2-20 Print Pdf File 53
6-2-21 Print Pdf File 54
6-2-22 Cut Paper 55
6-2-23 Open Drawer 56

7. MPos Lookup Reference 57
7-1 Over view 57
7-2 Methods 57
7-2-1 Refresh Devices List 57
7-2-2 Get Devices List 58

8. MPos Controller Label Printer Reference 59
8-1 Over view 59
8-2 Methods 59
8-2-1 Set Text Encoding 59
8-2-2 Set Characterset 61
8-2-3 Check Printer Status 63
8-2-4 Print Buffer 64
8-2-5 Print Raw Data 65
8-2-6 Draw Text Device Font 66
8-2-7 Draw Text Vector Font 68

8-2-8 Draw Barcode 1D 70
8-2-9 Draw Barcode Maxi Code 72
8-2-10 Draw Barcode PDF 417 73
8-2-11 Draw Barcode QR Code 75
8-2-12 Draw Barcode Data Matrix 76
8-2-13 Draw Barcode Aztec 77
8-2-14 Draw Barcode Code 49 78
8-2-15 Draw Barcode Coda Block 80
8-2-16 Draw Barcode Micro PDF 81
8-2-17 Draw Barcode IMB 82
8-2-18 Draw Barcode MSI 83
8-2-19 Draw Barcode Plessey 84
8-2-20 Draw Barcode TLC39 85
8-2-21 Draw Barcode RSS 86
8-2-22 Draw Block 88
8-2-23 Draw Circle 89
8-2-24 Draw Image 90
8-2-25 Draw Image File 91
8-2-26 Set Printing Type 92
8-2-27 Set Margin 93
8-2-28 Set Length 94
8-2-29 Set Width 95
8-2-30 Set Speed 96
8-2-31 Set Density 97
8-2-32 Set Orientation 98
8-2-33 Set Offset 99
8-2-34 Set Cutting Position 100
8-2-35 Set Auto Cutter 101
8-2-36 Get Model Name 102
8-2-37 Get Firmware Version 103
8-2-38 Get Printer Information 104
8-2-39 Setup RFID 105
8-2-40 Set RFID Position 106
8-2-41 Set RFID Password 107
8-2-42 Set EPC Data Structure 108
8-2-43 Write RFID 109
8-2-44 Lock RFID 110
8-2-45 Draw PDF File 111

9. MPos Controller BCD Reference 112
9-1 Over view 112
9-2 Method 112
9-2-1 Display String 112
9-2-2 Clean Screen 115

9-2-3 Store Image 116
9-2-4 Store Image File 117
9-2-5 Display Image 118
9-2-6 Clear Image 119
9-2-7 Show Cursor 120

10. MPos Controller DallasKey Reference 121
10-1 Over view 121

11. MPos Controller HID Reference 122
11-1 Over view 122
12. MPos Controller MSR Reference 123
12-1 Over view 123

13. MPos Controller NFC Reference 124
13-1 Over view 124

14. MPos Controller RFID Reference 125
14-1 Over view 125

15. MPos Controller Scale Reference 126
15-1 Over view 126

16. MPos Controller Scanner Reference 127
16-1 Over view 127

17. MPos Controller USB Serial Reference 128
17-1 Over view 128

18. MPos Controller Config Reference 129
18-1 Over view 129
18-2 Methods 129
18-2-1 Search Devices 129
18-2-2 Get BGate Serial Number 130
18-2-3 Get USB Device 131
18-2-4 Get Custom Devices 132
18-2-5 Get Serial Configuration 133
18-2-6 Set Serial Configuration 134
18-2-7 Add Custom Device 136
18-2-8 Delete Custom Device 137
18-2-9 ReInit Custom DeviceType 138
18-2-10 Get Serial Number 139

19. Appendix 140
19-1 Error Code Table..... 140

1. Manual Guide

This SDK manual provides descriptions of contents necessary for developing Xamarin.Forms (Portable Class Library, NET Standard) applications. Xamarin library and Demo application were built using Visual Studio 2015 on Windows, and all the contents in this manual are explained based on Visual Studio 2015 on Windows.

2. Supported Devices

The below table summarizes the supported devices (printers) that are available in this Xamarin SDK.

Receipt Printers	Resolution	Supported Communication	Max Printable Width
SPP-R200II	203 dpi	Bluetooth, Wi-Fi, USB	384 dots
SPP-R200III	203 dpi		384 dots
SPP-R210	203 dpi		384 dots
SPP-R300	203 dpi		576 dots
SPP-R310	203 dpi		576 dots
SPP-R400	203 dpi		832 dots
SPP-R410	203 dpi		832 dots
SPP-C200	203 dpi	Bluetooth, USB	384 dots
SPP-C300	203 dpi		576 dots
SRP-350IIOBE	180 dpi	Bluetooth, Ethernet, USB	512 dots
SRP-350plusIII	180 dpi	Bluetooth, Ethernet, Wi-Fi, USB	512 dots
SRP-352plusIII	203 dpi		576 dots
SRP-350plusV	180 dpi		512 dots
SRP-352plusV	203 dpi		576 dots
SRP-380	180 dpi		512 dots
SRP-382	203 dpi		576 dots
SRP-383	300 dpi		864 dots
SRP-380plus	180 dpi		512 dots
SRP-382plus	203 dpi		576 dots
SRP-Q200	203 dpi		432 dots
SRP-Q300	180 dpi		512 dots
SRP-Q302	203 dpi		576 dots
SRP-Q300II	203 dpi		576 dots
SRP-F310II	180 dpi		512 dots
SRP-F312II	203 dpi		576 dots
SRP-F313II	203 dpi		640 dots
SRP-S300	203 dpi		576 dots
SRP-S300II	203 dpi		576 dots
SRP-S320	203 dpi		576 dots
SRP-350III	180 dpi	Ethernet, USB	512 dots
SRP-352III	203 dpi		576 dots
SRP-350V	180 dpi		512 dots
SRP-352V	203 dpi		576 dots
SRP-330II	180 dpi		512 dots
SRP-330III	180 dpi		512 dots

SRP-332II	203 dpi		576 dots
SRP-332III	203 dpi		576 dots
SRP-S200	203 dpi		432 dots
SRP-QE300	180 dpi		512 dots
SRP-QE302	203 dpi		576 dots
SRP-E300	180 dpi		512 dots
SRP-E302	203 dpi		576 dots
SRP-S3000	203 dpi	Ethernet, Wi-Fi, USB	576 dots
BGT-100P	180 dpi	Bluetooth, Wi-Fi, Ethernet, USB	512 dots
BGT-102P	203 dpi		576 dots
SRP-Q300H,	180 dpi		512 dots
SRP-Q302H	203 dpi		576 dots
SRP-F310IIHi	180 dpi		512 dots
SRP-F312IIHi	203 dpi		576 dots
SRP-S300Hi	180 dpi		512 dots
SRP-S320Hi	203 dpi		576 dots
SRP-G300	203 dpi	USB	576 dots

Label Printers	Resolution	Supported Communication	Max Printable Width
SLP-DX220	203 dpi	Bluetooth, Wi-Fi, Ethernet, USB	432 dots
SLP-DX223	300 dpi		672 dots
SLP-TX220	203 dpi		432 dots
SLP-TX223	300 dpi		672 dots
SLP-DX420	203 dpi		864 dots
SLP-DX423	300 dpi		1248 dots
SLP-TX420	203 dpi		864 dots
SLP-TX423	300 dpi		1248 dots
SLP-TX400(RFID)	203 dpi		864 dots
SLP-TX403(RFID)	300 dpi		1248 dots
SRP-770III	203 dpi		832 dots
SLP-DL410	203 dpi		864 dots
SLP-DL413	300 dpi		1248 dots
XT5-40(RFID)	203 dpi		832 dots
XT5-43(RFID)	300 dpi		1248 dots
XT5-46(RFID)	600 dpi		2496 dots
XD5-40d	203 dpi		864 dots
XD5-40IId	203 dpi		864 dots
XD5-43d	300 dpi		1248 dots
XD5-43IId	300 dpi		1248 dots
XD5-40t(RFID)	203 dpi		864 dots
XD5-40IIt(RFID)	203 dpi		864 dots
XD5-43t(RFID)	300 dpi		1248 dots
XD5-43IIt(RFID)	300 dpi		1248 dots
BT3-40	203 dpi		864 dots
BT3-43	300 dpi		1248 dots
BT5-40(RFID)	203 dpi		864 dots
BT5-43(RFID)	300 dpi		1248 dots
BT5-46(RFID)	600 dpi		2496 dots
BD5-40d	203 dpi		864 dots
BD5-43d	300 dpi		1248 dots
BD5-40t(RFID)	203 dpi		864 dots
BD5-43t(RFID)	300 dpi		1248 dots
XL5-40	203 dpi		864 dots
XL5-43	300 dpi		1248 dots
XT3-40	203 dpi		864 dots
XT3-43	300 dpi		1248 dots
SRP-E770III	203 dpi	Ethernet, USB	832 dots
SPP-L3000	203 dpi	Bluetooth, Wi-Fi, USB	576 dots
SPP-L310	203 dpi		576 dots

SPP-L410	203 dpi		832 dots
XM7-20	203 dpi	Bluetooth, Wi-Fi, * Ethernet (Cradle Required), USB	384 dots
XM7-40(RFID)	203 dpi		832 dots
XM7-30	203 dpi	Bluetooth, Wi-Fi, USB	576 dots
SRP-S3000_LABEL	203 dpi	Wi-Fi, Ethernet, USB	576 dots
XQ-840II	203 dpi	USB	864 dots
XQ-843II	300 dpi	USB	1248 dots
XT6-60	203 dpi	Bluetooth, LAN, WLAN, USB	1344 dots
XT6-63	300 dpi	Bluetooth, LAN, WLAN, USB	1984 dots
XM5-30	203 dpi	Bluetooth, WLAN, USB	576 dots
XD7-20d	203 dpi	WLAN, USB	432 dots
XD7-23d	300 dpi	USB	672 dots

3. Xamarin SDK Installation

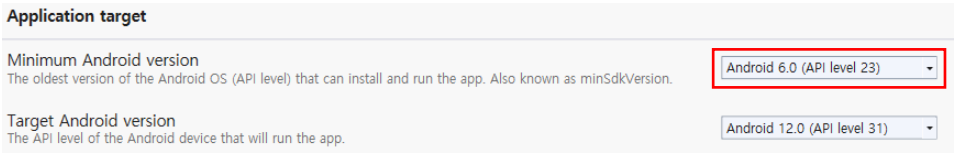
3-1 Supported Platforms

3-1-1 iOS

iOS 12.0 or later
In your Xamarin project for iOS, set Deployment Target version to 12.0.

3-1-2 Android

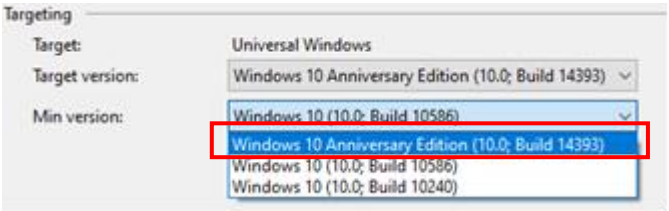
Android 6.0(API Level 23: Marshmallow) or later
In your Xamarin project for android, set minimum android version to API Level 23.



You should leave the target Android version set to Use Compile using SDK version (as shown) so that its value automatically matches the target framework setting. In general, the Target Android Version should be bounded by the Minimum Android Version and the Target Framework.
That is: **Minimum Android Version < Target Android Version < Target Framework.**

3-1-3 Windows

Windows 10 version Build 14393 or later.
In your Xamarin project for Universal Windows, set minimum version to Winddows 10 (10.0 Build 14393) as below.



3-2 Supported Communications

3-2-1 Bluetooth

HOST must be paired with Bluetooth device for Bluetooth communication in advance. Make sure to include necessary permissions (or capabilities) to each platform before using APIs in this manual. Please refer to the table following for permission settings.

3-2-2 WiFi / Ethernet

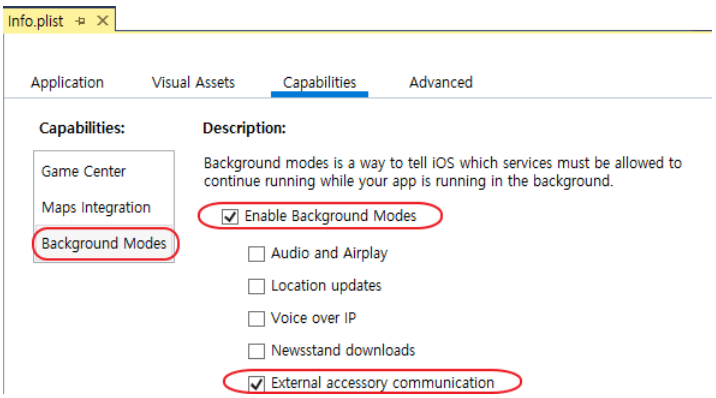
HOST and device (printer) can only communicate over TCP / IP on the same wired / wireless network. Make sure to include necessary permissions (or capabilities) to each platform before using APIs in this manual. Please refer to “3-3” in this manual for permission settings.

3-2-3 USB

USB communication is possible while the HOST and device (printer) are connected to each other with a USB cable

3-2-4 BLE

For BLE communication, it is possible to communicate with a Bluetooth device (printer) without pairing with the host, and the Bluetooth device (printer) must support BLE mode. Before using APIs in this manual. Please refer to the table following for permission settings.

Platform	Required permission / capabilities
iOS	1. Update the **Info.plist file** by opening it in XML editor or text editor. Add the following to the file. This is to allow your app to access the Bluetooth, BLE and Network ports. <key>**UISupportedExternalAccessoryProtocols**</key> <array><string>**com.bixolon.protocol**</string></array> <key>**NSBluetoothAlwaysUsageDescription**</key> <string>Communication with the printer for printing.</string> <key>**NSBluetoothPeripheralUsageDescription**</key> <string>Communication with the printer for printing.</string> 2. Put a checkmark on "**External Accessory communication**" checkbox as a below.  **Multicast Networking Entitlement Request** If the iOS project deployment target version is 12 or higher or the iPhone/iPad OS version is 14 or higher, you must need to request multicast permission from Apple to discover printer devices. See the link below. 1) https://developer.apple.com/news/?id=0oi77447 2) https://developer.apple.com/contact/request/networking-multicast **[Note]** - IP address used for discovering devices: 255.255.255.255 - Port numbers used for discovering devices: 48780, 48781, 9000, 3337a

<table>
<tr><td>Android</td><td>Make sure to include following capabilities in the AndroidManifest.xml file.
- BLUETOOTH, BLUETOOTH_ADMIN, BLUETOOTH_PRIVILEGED
- ACCESS_WIFI_STATE, CHANGE_WIFI_STATE
- ACCESS_FINE_LOCATION or ACCESS_COARSE_LOCATION

<manifest xmlns:android="http://schemas.android.com/apk/res/android" android:installLocation="auto">
<uses-sdk android:minSdkVersion="15" />
<uses-permission android:name="android.permission.BLUETOOTH" />
<uses-permission android:name="android.permission.BLUETOOTH_ADMIN" />
<uses-permission android:name="android.permission.BLUETOOTH_PRIVILEGED" />
<uses-permission android:name="android.permission. ACCESS_WIFI_STATE " />
<uses-permission android:name="android.permission. CHANGE_WIFI_STATE " />
<uses-permission android:name="android.permission. ACCESS_FINE_LOCATION" />
</manifest></td></tr>
<tr><td>Windows</td><td>Make sure to include following capabilities in the “Package.appxmanifest” file.
- Bluetooth, Internet (Client), Private Networks (Client & Server)</td></tr>
</table>

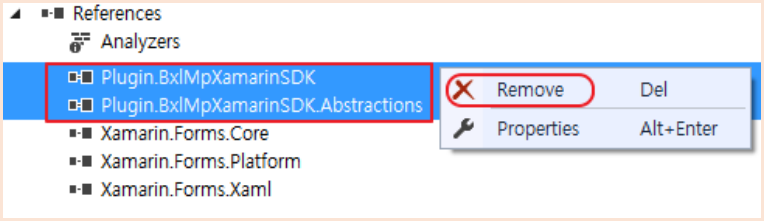


When using Bluetooth, BLE and WiFi, there are a few things to modify your project’s settings and configuration information related to permissions for each platform. Demo application provided helps you modifying the necessary settings of your own Xamarin application.

3-3 Installation Instruction

• Xamarin SDK can be installed as a **NuGet** package into MS Visual Studio.
When you use NuGet to install a package, it copies the library files to your solution and automatically updates your project (add references, change config files, etc). If you remove a package, **NuGet** reverses whatever changes it maded. If you need more information about NuGet, visit at https://www.nuget.org/

Before installing NuGet Package for Bixolon Xamarin SDK, library references below must be removed if you added them manually on each platform.

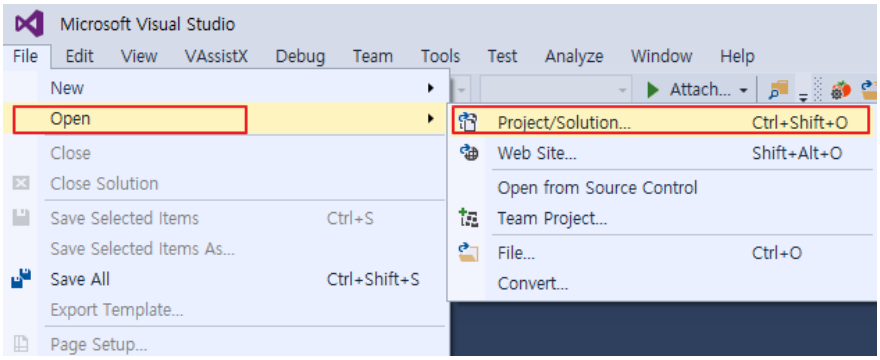


3-3-1 Installation steps from NuGet server

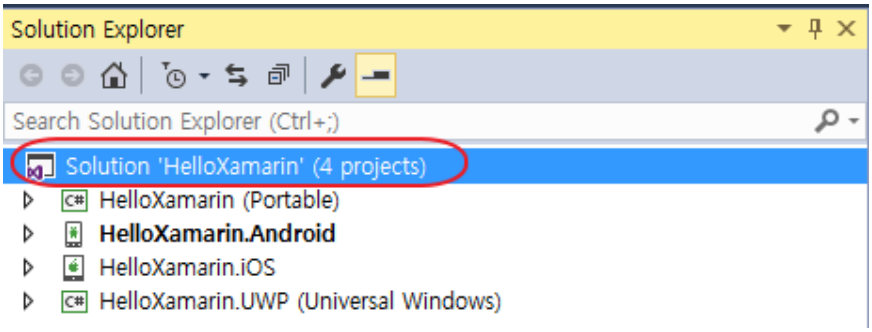
This way isn't supported yet. Refer to "3-3-2" below in this munual to install Xamarin SDK.

3-3-2 Installation steps from a local nupkg file

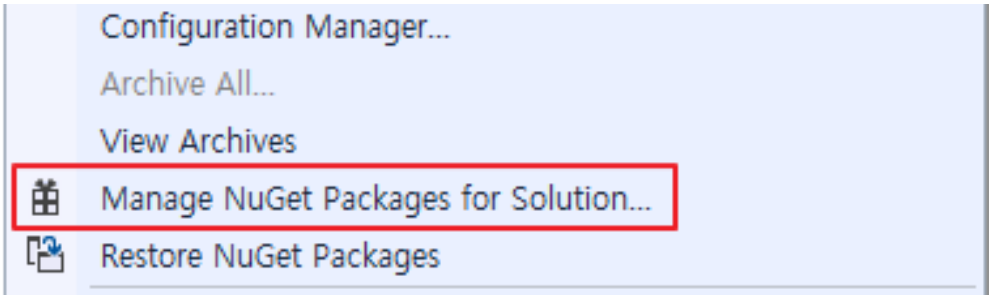
1) Open Visual Studio, and open your own Xamarin.Forms solution.



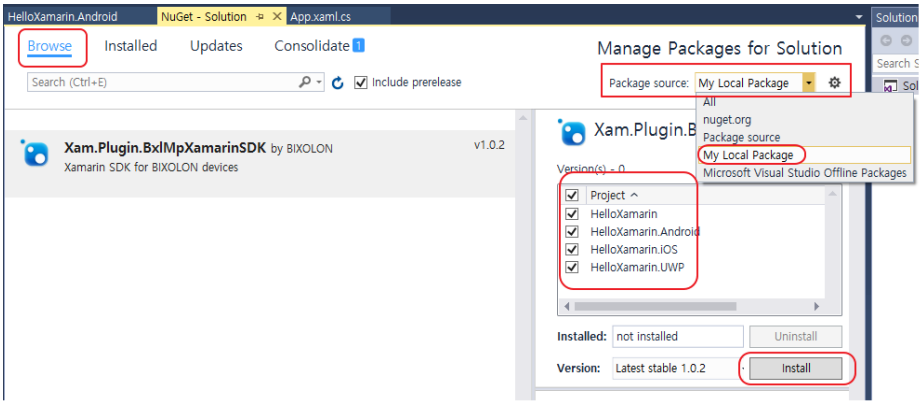
2) From the Solution Explorer, right click the top level solution.



3) Click "**Manage NuGet Packages for Solution...**"



4) On the top of the right side, click "**Setting**" button in red.



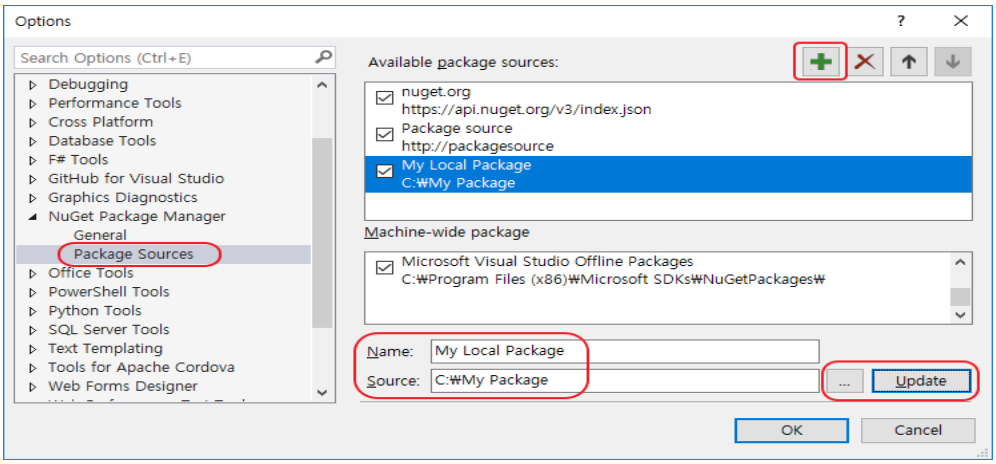
5) Click “**Package Sources**”, and then click “**+**” button.

6) Browse to the directory where “**Xam.Plugin.BxlMpXamarinSDK.x.x.x.nupkg**” file exists, and then enter any name you want.



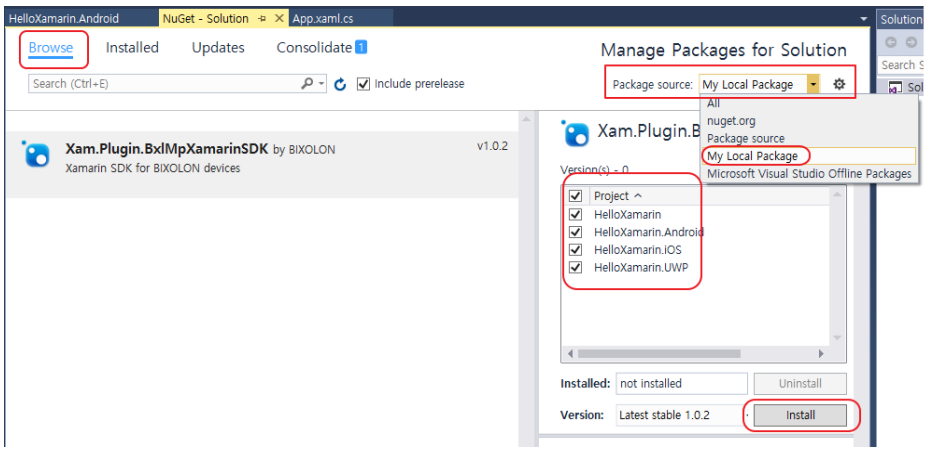
“x.x.x” is the version number for the SDK.

7) Click “**Update**”

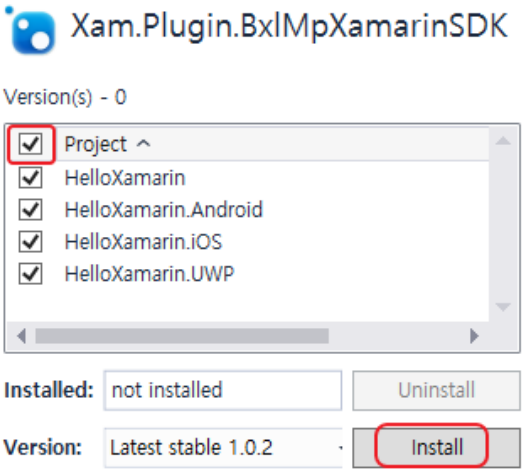


8) Click “**OK**”.

9) “**Xam.Plugin.BxlMpXamarinSDK**” package can be found when you select the name you entered in step 6 in the “**Package source**” list.



10) Click "**Project**" checkbox to install the entire package.



11) Click "**Install**" to start installing all the appropriate references to the libraries for each platform project.

4. IMPos DeviceFactory Reference

4-1 Over view

• "**IMPosDeviceFactory**" interface can be used to create an instance implementing "**MPosControllerDevices**" interface for communication with a specific device.
All instances for device (printer) control are created using the "**createDevice**" method.

4-2 Methods



Use the "**current**" property in "**MPosDeviceFactory**" class to call the methods defined in this interface.
eg) MPosDeviceFactory.Current.createDevice(10)

4-2-1 Create Device

Gets an instance for communication with a specific device.

[Syntax]
MPosControllerDevices createDevice (MPosDeviceType deviceType);

[Parameters]
MPosDevice deviceType: A value of the device type to create.

MPosDeviceType Enumeration	Value	Description
MPOS_DEVICE_LABEL_PRINTER	1	Label Prniter (SLCS)
MPOS_DEVICE_PRINTER	10	POS Printer (ESC/POS)

[Returns]
A reference to the instance implementing "**MPosControllerDevices**" interface specified or null.

5. MPos Controller Devices Reference

5-1 Over view

• "**MPosControllerDevices**" interface provides common member methods with regards to the device, and the properties and methods can be used in all the classes implementing this interface.

5-2 Properties

5-2-1 Data Event

If data entry occurs in a device, the data in byte array will be sent to the application through this event.

5-2-2 Status Update Event

If the status of a printer device changes, the new status will be sent to the application through the this event. The sent value is an integer type constant value that has been defined in advance.

MPosPrinterStatus Enumeration	Value	Description
MPOS_STATUS_NORMAL	0	Printing is possible
MPOS_STATUS_PRINTER_COVEROPEN	1	Cover open
MPOS_STATUS_PRINTER_PAPEREMPTY	2	No paper
MPOS_STATUS_PRINTER_PAPER_NEAREND	4	Insufficient paper
MPOS_STATUS_PRINTER_ERROR	8	Offline (The device is disconnected with HOST for some reasons)
MPOS_STATUS_BATT_HIGH	16	Battery level is in high
MPOS_STATUS_BATT_MID	32	Battery level is in middle
MPOS_STATUS_BATT_LOW	64	Battery level is in low
MPOS_STATUS_PRINTER_OFFLINE	128	Offline by the printer's power off



The values related to battery level can be detected on SRP-Q300 and SRP-Q302.

5-2-3 Device ID

The device Id of the device.

5-2-4 Is Open

The device in the usable status or not.

5-2-5 Command Mode

Communication way to the device via “B-gate” or directly connection.

5-2-6 Interface Type

Communication Interface such as Bluetooth, Wireless LAN, Wired Lan, USB or BLE.

5-3 Methods

5-3-1 Select Interface

Configures the interface type and address of the device(printer) to connect.



This method must be used before calling the “**openService**” method.

[Syntax]
int selectInterface (int interfaceType, string address);

[Parameters]
• int interface type: communication type to connect the device.

MPosInterfaceType Enumeration	Value	Description
MPOS_INTERFACE_WIFI	1	WiFi.
MPOS_INTERFACE_ETHERNET	2	Ethernet.
MPOS_INTERFACE_BLUETOOTH	4	Bluetooth.
MPOS_INTERFACE_BLE	8	BLE.
MPOS_INTERFACE_USB	16	USB

• string address:
In case of TCP (Wi-Fi or Ethernet) communication, it should be used in the form of "IP address [: port number]". Port number is 9100 by default and can be omitted.
In case of Bluetooth communication, the values described in the table below must be entered depending on the platform.

Platform	Address value
iOS	Bluetooth MAC Address
Android	Bluetooth MAC Address
Windows	Bluetooth MAC Address

In case of BLE communication, the values described in the table below must be entered depending on the platform.

Platform	Address value
iOS	Bluetooth MAC Address
Android	Bluetooth MAC Address
Windows	Not supported

In case of USB communication, the values described in the table below must be entered depending on the platform.

Platform	Address value
iOS	Not supported
Android	Value of USB Name
Windows	Not supported

[Returns]

If the method succeeds, the return value is zero (0). Other cases, returns non-zero.

5-3-2 Select Command Mode

Select whether to communicate directly with the device or via "B-gate".

[Syntax]
int selectCommandMode (int commandMode);

[Parameters]
• int commandMode: communication way with the device

MPosCommandMode Enumeration	Value	Description
MPOS_MODE_BGATE	0	Communicate through B-gate **(default)**
MPOS_MODE_DIRECT	1	Direct communication

[Returns]
If the method succeeds, the return value is zero (0). Other cases, returns non-zero.

5-3-3 Open Service

Establishes a communication connection with the device.

[Syntax]
Task<int> **openService** (uint timeout = 3);

[Parameters]
• uint timeout: Timeout value for connecting a device.

[Returns]
If the method succeeds, the return value is zero (0). Other cases, returns non-zero.

5-3-4 Close Service

Disconnects the communication with the device.

[Syntax]
Task<int> closeService (int closeTimeout = 0);

[Parameters]
• int closeTimeout: Timeout value for disconnecting a device.

[Returns]
If the method succeeds, the return value is zero (0). Other cases, returns non-zero.

5-3-5 Direct IO

Sends user-defined data to the device.

[Syntax]
Task<int> directIO(byte[] data);

[Parameters]
• byte[] data: Data to be sent to device.

[Returns]
If the method succeeds, the return value is zero (0). Other cases, returns non-zero.

5-3-6 Set Transaction

Enters or Leaves 'Transaction' mode, and then sends the data in the buffer.



It is recommended to use the Transaction mode when using output devices such as printers, Customer displays, label printers, etc.

[Syntax]
Task<int> setTransaction (int mode);

[Parameters]
• int mode: Configure the transaction mode

MPosTransactionMode Enumeration	Value	Description
MPOS_PRINTER_TRANSACTION_OUT	0	Cancel transaction mode after data transmission
MPOS_PRINTER_TRANSACTION_IN	1	Enter transaction mode

[Returns]
If the method succeeds, the return value is zero (0). Other cases, returns non-zero.

6. MPos Controller Printer Reference

6-1 Over view

• The MPosControllerPrinter interface provides methods for controlling a thermal printer.

6-2 Methods

6-2-1 Print Text

Prints characters by specifying various font attributes of the character.

[Syntax]
Task<int> printText (string data, MPosFontAttribute attribute);

[Parameters]
• string data: Character string to be printed.
• MPosFontAttribute: Class Oject for the font attributes.

```
class MPosFontAttribute
{
    public MPosAlignment Alignment { get; set; }
    public bool Bold { get; set; }
    public int CodePage { get; set; }
    public MPosFontType FontType { get; set; }
    public MPosFontSizeHeight Height { get; set; }
    public bool Reverse { get; set; }
    public MPosFontUnderline Underline { get; set; }
    public MPosFontSizeWidth Width { get; set; }
}
```

Properties
• Alignment: specifies characters alignments (Left, Center, Right)
• Bold: specifies whether the font is bold.
• CodePage: code page used for endcoding a character string.
Note: 'CodePage' value should be one of 'MPosCodePage' or 'MPosEastAsiaCodePage' enumeration.
• FontType: specifies one of the font types (Font A, Font B, F*o*nt C)
• Height: specifies the font height.
• Width: specifies the font width.
• Reverse: specifies whether the font is reversed.
• Underline: specifies whether the font is underlined.

[Returns]
If the method succeeds, the return value is zero (0). Other cases, returns non-zero.

6-2-2 Print Text

Prints characters by specifying various font attributes of the character.

[Syntax]
Task<int> printText(byte[] data, MPosFontAttribute attribute);

[Parameters]
• byte[] data: An array of bytes in a character string encoded to be printed.
• MPosFontAttribute: Class Oject for the font attributes.

```
class MPosFontAttribute
{
    public MPosAlignment Alignment { get; set; }
    public bool Bold { get; set; }
    public int CodePage { get; set; }
    public MPosFontType FontType { get; set; }
    public MPosFontSizeHeight Height { get; set; }
    public bool Reverse { get; set; }
    public MPosFontUnderline Underline { get; set; }
    public MPosFontSizeWidth Width { get; set; }
}
```

Properties
• Alignment: specifies characters alignments (Left, Center, Right)
• Bold: specifies whether the font is bold.
• CodePage: no used (not vaild in this method).
• FontType: specifies one of the font types (Font A, Font B, Font C)
• Height: specifies the font height.
• Width: specifies the font width.
• Reverse: specifies whether the font is reversed.
• Underline: specifies whether the font is underlined.

[Returns]
If the method succeeds, the return value is zero (0). Other cases, returns non-zero.

6-2-3 Set Character set

Sets the code page used for encoding characters.

[Syntax]
Task<int> setCharacterset (int characterset);

[Parameters]
• int characterset: Code page to use when encoding a character string.

MPosCodePage Enumeration	Value	Description
MPOS_CODEPAGE_CP437	437	USA, Standard Europe (PC437)
MPOS_CODEPAGE_CP850	850	Western European (PC850)
MPOS_CODEPAGE_CP860	860	Portuguese (PC860)
MPOS_CODEPAGE_CP863	863	Canadian-French (PC863)
MPOS_CODEPAGE_CP865	865	Nordic (PC865)
MPOS_CODEPAGE_WPC1252	1252	Latin 1 (WPC1252)
MPOS_CODEPAGE_CP866	866	Cyrillic #2 (PC866)
MPOS_CODEPAGE_CP852	852	Latin 2 (PC852)
MPOS_CODEPAGE_CP858	858	Euro (PC858)
MPOS_CODEPAGE_CP862	862	Hebrew DOS code (PC862)
MPOS_CODEPAGE_THAI11	34	Thai character code 11 (THAI 11)
MPOS_CODEPAGE_THAI14	31	Thai character code 14 (THAI 14)
MPOS_CODEPAGE_THAI16	39	Thai character code 16 (THAI 16)
MPOS_CODEPAGE_THAI18	35	Thai character code 18 (THAI 18)
MPOS_CODEPAGE_THAI42	23	Thai character code 42 (THAI 42)
MPOS_CODEPAGE_WPC1253	1253	Greek (WPC1253)
MPOS_CODEPAGE_WPC1254	1254	Turkce (WPC1254)
MPOS_CODEPAGE_WPC1257	1257	Baltic (WPC1257)
MPOS_CODEPAGE_WPC1251	1251	Cyrillic (WPC1251)
MPOS_CODEPAGE_FARSI	27	Persian
MPOS_CODEPAGE_FARSI_II	28	Persian
MPOS_CODEPAGE_CP737	737	Greek (PC 737)
MPOS_CODEPAGE_CP775	775	Baltic (PC775)
MPOS_CODEPAGE_WPC1255	1255	Hebrew New code (WPC1255)
MPOS_CODEPAGE_CP855	855	Cyrillic (PC855)
MPOS_CODEPAGE_CP857	857	Turkce (PC857)
MPOS_CODEPAGE_CP928	928	Greek (PC928)
MPOS_CODEPAGE_WPC1258	1258	Vietnamese (PC1258)
MPOS_CODEPAGE_WPC1250	1250	Central European (PC1250)
MPOS_CODEPAGE_LATIN9	28605	Latin 9 (Latin 9)
MPOS_CODEPAGE_TCVN	49	Vietnamese (TCVN3)
MPOS_CODEPAGE_TCVN_CAPTIAL	50	Vietnamese (TCVN3 Capital)

[Returns]
If the method succeeds, the return value is zero (0). Other cases, returns non-zero.

6-2-4 Set International Characters et

Sets the character set used for encoding international characters.

[Syntax]
Task<int> setInternationalCharacterset (int internationalCharset);

[Parameters]
• int internationalCharset: International character set

MPosICS Enumeration	Value	Description
MPOS_ICS_USA	0	USA Code
MPOS_ICS_FRANCE	1	FRANCE Code
MPOS_ICS_GERMANY	2	GERMANY Code
MPOS_ICS_UK	3	UK Code
MPOS_ICS_DENMARK_I	4	DENMARK1 Code
MPOS_ICS_SWEDEN	5	SWEDEN Code
MPOS_ICS_ITALY	6	ITALY Code
MPOS_ICS_SPAIN_I	7	SPAIN1 Code
MPOS_ICS_JAPAN	8	JAPAN Code
MPOS_ICS_NORWAY	9	NORWAY Code
MPOS_ICS_DENMARK_II	10	DENMARK 2 Code
MPOS_ICS_SPAIN_II	11	SPAIN 2 Code
MPOS_ICS_LATIN_AMERICA	12	LATIN AMERICA Code
MPOS_ICS_KOREA	13	KOREA Code
MPOS_ICS_SLOVENIA_CROATIA	14	SLOVENIA Code
MPOS_ICS_CHINA	15	CHINA Code

[Returns]
If the method succeeds, the return value is zero (0). Other cases, returns non-zero.

6-2-5 Set Page mode

Enters or Leaves 'Page' mode, and then sends the data in the buffer.



Pagemode cannot be used simultaneously with transaction mode.

[Syntax]
Task<int> setPagemode (int mode);

[Parameters]
• int mode: Enter or leave page mode

MPosPageMode Enumeration	Value	Description
MPOS_PRINTER_PAGEMODE_IN	1	Enter Pagemode
MPOS_PRINTER_PAGEMODE_OUT	0	Leave Pagemode and then start printing

[Returns]
If the method succeeds, the return value is zero (0). Other cases, returns non-zero.

6-2-6 Set Pagemode Print Area

Sets the printing area of pagemode.

[Syntax]
int setPagemodePrintArea (int x, int y, int width, int height);

[Parameters]
• int x: The starting X axis coordinate
• int y: The starting Y axis coordinate
• int width: Printing Width
• int height: Printing Height

[Returns]
If the method succeeds, the return value is zero (0). Other cases, returns non-zero.

6-2-7 Set Pagemode Direction

Sets the printing direction in pagemode.

[Syntax]
int setPagemodeDirection (int direction);

[Parameters]
• int direction: Printing direction.

MPosPageMode Enumeration	Value	Description
MPOS_PRINTER_PD_0	0	0 degree turn
MPOS_PRINTER_PD_LEFT90	1	90 degree turn counter-clockwise
MPOS_PRINTER_PD_180	2	180 degree turn
MPOS_PRINTER_PD_RIGHT90	3	90 degree turn clockwise

[Returns]
If the method succeeds, the return value is zero (0). Other cases, returns non-zero.

6-2-8 Set Pagemode Position

Sets the coordinates of the printing target in pagemode.

[Syntax]
int setPagemodePosition (int x, int y);

[Parameters]
• int x: x-coordinate, in dot unit, of the object (text, barcode, image) to be printed.
• int y: y-coordinate, in dot unit, of the object (text, barcode, image) to be printed.

[Returns]
If the method succeeds, the return value is zero (0). Other cases, returns non-zero.

6-2-9 Get Firmware Version

Gets the printer’s firmware version.

[Syntax]
Task<int> getFirmwareVersion (Action<string> versionDelegate)

[Parameters]
• Action<string> versionDelegate: A delegate with a parameter used for getting a firmware version.

[Returns]
If the method succeeds, the return value is zero (0). Other cases, returns non-zero.

6-2-10 Print Bitmap

Prints an image using a platform dependent image object.

[Syntax]
Task<int> printBitmap (object platformImage, int width, int alignment, int brightness, bool isDithering, bool isCompress);

[Parameters]
• object platformImage: A platform dependent image object.

iOS	UIKit.UIImage
Android	Android.Graphics.Bitmap
Windows	Windows.UI.Xaml.Media.Imaging.WriteableBitmap Windows.UI.Xaml.Media.Imaging.BitmapImage (deprecated)

• int width: Image width to be printed or original width ot the image.

MPosImageWidth Enumeration	Value	Description
MPOS_IMAGE_WIDTH_ASIS	-2	Print as original image size

• int alignment: Alignment.

MPosAlignment Enumeration	Value	Description
MPOS_ALIGNMENT_LEFT	0	Align to the left
MPOS_ALIGNMENT_CENTER	1	Align at the center
MPOS_ALIGNMENT_RIGHT	2	Align to the right

• int brightness: The brightness level

Platform	Brightness Range
Android, UWP	0 ≤ brightness ≤ 100
iOS	-100 ≤ brightness ≤ 100

• bool isDithering: Whether or not to apply dithering
• bool isCompress: Whether or not to compress data

[Returns]
If the method succeeds, the return value is zero (0). Other cases, returns non-zero.

6-2-11 Print Bitmap File

Prints Bitmap file specified by a local path.

[Syntax]
Task<int> printBitmapFile (string path, int width, int alignment, int brightness, bool isDithering, bool isCompress);

[Parameters]
• string path: The path of the image file.



In Android, the image file should be in the SD card.

• int width: Image width to be printed or original width ot the image.

MPosImageWidth Enumeration	Value	Description
MPOS_IMAGE_WIDTH_ASIS	-2	Print as original image size

• int alignment: Alignment.

MPosAlignment Enumeration	Value	Description
MPOS_ALIGNMENT_LEFT	0	Align to the left
MPOS_ALIGNMENT_CENTER	1	Align at the center
MPOS_ALIGNMENT_RIGHT	2	Align to the right

• int brightness: The brightness level

Platform	Brightness Range
Android, UWP	0 ≤ brightness ≤ 100
iOS	-100 ≤ brightness ≤ 100

• bool isDithering: Whether or not to apply dithering
• bool isCompress: Whether or not to compress data

[Returns]
If the method succeeds, the return value is zero (0). Other cases, returns non-zero.

6-2-12 Print 1D Barcode

Prints a 1-D barcode.

[Syntax]
Task<int> print1DBarcode (string data, int symbology, int height, int barWidth, int alignment, int textPosition);

[Parameters]
• string data: data of the barcode to be printed
• int symbology: The type of 1-D barcode.

MPos1dCodeType Enumeration	Value	Description
MPOS_BARCODE_UPCA	101	UPC A barcode
MPOS_BARCODE_UPCE	102	UPC E barcode
MPOS_BARCODE_EAN8 MPOS_BARCODE_JAN8	103	EAN 8/JAN 8 barcode
MPOS_BARCODE_EAN13 MPOS_BARCODE_JAN13	104	EAN 13/JAN 13 barcode
MPOS_BARCODE_ITF	106	ITF barcode
MPOS_BARCODE_CODABAR	107	CODABAR barcode
MPOS_BARCODE_CODE39	108	CODE39 barcode
MPOS_BARCODE_CODE93	109	CODE93 barcode
MPOS_BARCODE_CODE128	110	CODE128 barcode

Code128 subset A, B, or C can be printed by entering characters such as "{A", "{B" or "C" at the start of the data. Data in Code128 subset C is only allowed in numbers and must consist of even data.

• int height: The barcode height. (Range: 1 ≤ height ≤ 255)
• int barWidth: The barcode width. (Range: 2 ≤ barWidth ≤ 6)
• int alignment: Alignment value.

MPosAlignment Enumeration	Value	Description
MPOS_ALIGNMENT_LEFT	0	Align left
MPOS_ALIGNMENT_CENTER	1	Align center
MPOS_ALIGNMENT_RIGHT	2	Align right

• int textPosition: The printing position of the HRI (Human Readable Interface).

MPosLabelHRI Enumeration	Value	Description
MPOS_BARCODE_TEXTNONE	0	Does not print
MPOS_BARCODE_TEXTABOVE	1	Print above barcode
MPOS_BARCODE_TEXTBELOW	2	Print below barcode

[Returns]

If the method succeeds, the return value is zero (0). Other cases, returns non-zero.

6-2-13 Print QR Code

Prints the QR Code.

[Syntax]
int printQRCode (string data, int model, int alignment, int moduleSize, int eccLevel);

[Parameters]
• string data: The QR Code data
• int model: The model type

MPos2dCodeType Enumeration	Value	Description
MPOS_BARCODE_QRCODE_MODEL1	204	MODEL 1
MPOS_BARCODE_QRCODE_MODEL2	205	MODEL 2

• int alignment: Alignment value.

MPosAlignment Enumeration	Value	Description
MPOS_ALIGNMENT_LEFT	0	Align left
MPOS_ALIGNMENT_CENTER	1	Align center
MPOS_ALIGNMENT_RIGHT	2	Align right

• int moduleSize: Module size (Range: 1 ≤ moduleSize ≤ 8)
• int eccLevel: Error correction level

MPosQRCodeECCLevel Enumeration	Value	Description
MPOS_QRCODE_ECC_LEVEL_L	48	LEVEL L
MPOS_QRCODE_ECC_LEVEL_M	49	LEVEL M
MPOS_QRCODE_ECC_LEVEL_Q	50	LEVEL Q
MPOS_QRCODE_ECC_LEVEL_H	51	LEVEL H

[Returns]
If the method succeeds, the return value is zero (0). Other cases, returns non-zero.

6-2-14 Print PDF 417

Prints the PDF417 code.

[Syntax]

Task<int> **printPDF417**(string data, int symbol, int alignment, int columnNumber, int rowNumber, int moduleWidth, int moduleHeight, int eccLevel);

[Parameters]

• string data: The PDF417 code data

• int symbol: The type of the PDF417 code

MPos2dCodeType Enumeration	Value	Description
MPOS_BARCODE_PDF417	201	PDF417 Standard
MPOS_BARCODE_PDF417_SIMPLIFIED	202	PDF417 Simplified

• int alignment: Alignment value.

MPosAlignment Enumeration	Value	Description
MPOS_ALIGNMENT_LEFT	0	Align left
MPOS_ALIGNMENT_CENTER	1	Align center
MPOS_ALIGNMENT_RIGHT	2	Align right

• int columnNumber: Number of columns (Range: 0 ≤ columnNumber ≤ 30)

• int rowNumber: Number of rows (Range: 3 ≤ rowNumber ≤ 90)

• int moduleWidth: Module width (Range: 1 ≤ moduleWidth ≤ 4)

• int moduleHeight: Module height (Range: 2 ≤moduleHeight ≤ 8)

• int eccLevel: The error correction level

MPosPDF417ECCLevel Enumeration	Value	Description
MPOS_PDF417_ECC_LEVEL_0	48	Error Correction Level 0
MPOS_PDF417_ECC_LEVEL_1	49	Error Correction Level 1
MPOS_PDF417_ECC_LEVEL_2	50	Error Correction Level 2
MPOS_PDF417_ECC_LEVEL_3	51	Error Correction Level 3
MPOS_PDF417_ECC_LEVEL_4	52	Error Correction Level 4
MPOS_PDF417_ECC_LEVEL_5	53	Error Correction Level 5
MPOS_PDF417_ECC_LEVEL_6	54	Error Correction Level 6
MPOS_PDF417_ECC_LEVEL_7	55	Error Correction Level 7
MPOS_PDF417_ECC_LEVEL_8	56	Error Correction Level 8

[Returns]

If the method succeeds, the return value is zero (0). Other cases, returns non-zero.

6-2-15 Check Printer Status

Gets the printer’s current status.

[Syntax]
Task<uint> checkPrinterStatus(void);

[Returns]
If the method succeeds, the return value is zero (0). Other cases, returns non-zero.

MPosPrinterStatus Enumeration	Value	Description
MPOS_STATUS_NORMAL	0	Printing is possible
MPOS_STATUS_PRINTER_COVEROPEN	1	Cover open
MPOS_STATUS_PRINTER_PAPEREMPTY	2	No paper
MPOS_STATUS_PRINTER_PAPER_NEAREND	4	Insufficient paper
MPOS_STATUS_PRINTER_ERROR	8	Error (Offline or unknown error)
MPOS_STATUS_BATT_HIGH	16	Battery level is high
MPOS_STATUS_BATT_MID	32	Battery level is middle
MPOS_STATUS_BATT_LOW	64	Battery level is low
MPOS_STATUS_PRINTER_OFFLINE	128	Offline by the printer's power off

6-2-16 Check Batt Status

Gets the printer's current battery level.



This method is available only for "SPP-Q300", "SPP-Q302" or mobile printers.

[Syntax]
Task<uint> checkBattStatus(void);

[Returns]
If the method succeeds, the return value is zero (0). Other cases, returns non-zero.

MPosPrinterStatus Enumeration	Value	Description
MPOS_STATUS_NORMAL	0	Battery Power Level: Full
MPOS_STATUS_BATT_HIGH	16	Battery Power Level: High
MPOS_STATUS_BATT_MID	32	Battery Power Level: Middle
MPOS_STATUS_BATT_LOW	64	Battery Power Level: Low

6-2-17 Asb Enable

Enable or disable ASB (Automatic Status Back).



If enable, whenever printer status changes, the method registered by ‘statusUpdateEvent’ delegate is called. For more information about the printer status values, refer to 5-2-2.

[Syntax]
Task<uint> asbEnable (bool enable);

[Parameters]
• bool enable: start or stop notifying of a printer status by ‘statusUpdateEvent’ delegate.

[Returns]
If the method succeeds, the return value is zero (0). Other cases, returns non-zero.

6-2-18 Print Line

Prints the straight line.

- This method is available in page mode, and print direction must be left to right (no rotation), and only straight line can be printed.
- This method is available only for mobile printers.

[Syntax]
Task<int> printLine (int x1, int y1, int x2, int y2, int thickness);

[Parameters]
• int x1: x-coordinate, in dot unit, of the start point of the straight line.
• int y1: y-coordinate, in dot unit, of the start point of the straight line.
• int x2: x-coordinate, in dot unit, of the end point of the straight line.
• int y2: y-coordinate, in dot unit, of the end point of the straight line.
• int thickness: Thickness of the straight line (Range: 0 ≤ thickness ≤ 16)

[Returns]
If the method succeeds, the return value is zero (0). Other cases, returns non-zero.

6-2-19 Print Box

Prints the rectangle.

- This method is available in page mode, and print direction must be left to right (no rotation).
- This method is available only for mobile printers.

[Syntax]
Task<int> printBox (int left, int top, int right, int bottom, int thickness);

[Parameters]
• int left: x-coordinate, in dot unit, of the upper-left corner of the rectangle.
• int top: y-coordinate, in dot unit, of the upper-left corner of the rectangle.
• int right: x-coordinate, in dot unit, of the lower-right corner of the rectangle.
• int bottom: y-coordinate, in dot unit, of the lower-right corner of the rectangle.
• int thickness: Thickness of the rectangle (Range: 0 ≤ thickness ≤ 16)

[Returns]
If the method succeeds, the return value is zero (0). Other cases, returns non-zero.

6-2-20 Print Pdf File

Prints PDF file specified by a local path.

[Syntax]
Task<int> printPdfFile (object context, string filePath, int startPage, int endPage, int width, int alignment, int brightness, bool isDithering, bool isCompress);

[Parameters]
• object context: not used



Set this as null.

• string filePath: The path of the PDF file.
• int startPage: The number of start page to print.



The first page start with zero (0)

• int endPage: The number of last page to print.



To print all pages, use a value of -1.

• int width: Image width to be printed or original width ot the image.

MPosImageWidth Enumeration	Value	Description
MPOS_IMAGE_WIDTH_ASIS	-2	Print as original image size

• int alignment: Alignment.

MPosAlignment Enumeration	Value	Description
MPOS_ALIGNMENT_LEFT	0	Align to the left
MPOS_ALIGNMENT_CENTER	1	Align at the center
MPOS_ALIGNMENT_RIGHT	2	Align to the right

• int brightness: The brightness level

Platform	Brightness Range
Android, UWP	0 ≤ brightness ≤ 100
iOS	-100 ≤ brightness ≤ 100

• bool isDithering: Whether or not to apply dithering
• bool isCompress: Whether or not to compress data

[Returns]
If the method succeeds, the return value is zero (0). Other cases, returns non-zero.

6-2-21 Print Pdf File

Prints PDF file specified by a local path.

[Syntax]
Task<int> printPdfFile (object context, string filePath, int startPage, int endPage, int width, int alignment, int threshold, int didtheringType, int compressType);

[Parameters]
• object context: not used



Set this as null.

• string filePath: The path of the PDF file.
• int startPage: The number of start page to print.



The first page start with zero (0)

• int endPage: The number of last page to print.



To print all pages, use a value of -1.

• int width: Image width to be printed or original width ot the image.

MPosImageWidth Enumeration	Value	Description
MPOS_IMAGE_WIDTH_ASIS	-2	Print as original image size

• int alignment: Alignment.

MPosAlignment Enumeration	Value	Description
MPOS_ALIGNMENT_LEFT	0	Align to the left
MPOS_ALIGNMENT_CENTER	1	Align at the center
MPOS_ALIGNMENT_RIGHT	2	Align to the right

• int threshold: Threshold (0 ≤ threshold ≤ 255)
• int ditheringType: dithering type (0: No dithering, 1: dithering)
• int compressType: Compress type

MPosCompressType Enumeration	Value	Description
MPOS_IMAGE_COMPRESS_NONE	0	No compress
MPOS_IMAGE_COMPRESS_RLE	1	RLE
MPOS_IMAGE_COMPRESS_LZMA	2	LZMA

[Returns]
If the method succeeds, the return value is zero (0). Other cases, returns non-zero.

6-2-22 Cut Paper

Cuts the paper.



Call this method only on printers equipped with auto-cutters.

[Syntax]
Task<int> cutPaper (bool feed, bool fullCut);

[Parameters]
• bool feed: No paper feed or paper feed to the cutting position.
• bool fullCut: Full cut or partial cut.

[Returns]
If the method succeeds, the return value is zero (0). Other cases, returns non-zero.

6-2-23 Open Drawer

Opens the cash drawer.



Call this method only on printers equipped with cash drawer.

[Syntax]
Task<int> openDrawer (int pinNumber, int onTimePulse, int offTimePulse);

[Parameters]
• int pinNumber: Cash drawer PIN number

Value	Description
0	Number 2
1	Number 5

• onTimePulse: Ontime 0 ≤ onTimePulse ≤ 255
• offTimePulse: OffTime 0 ≤ onTimePulse ≤ 255

[Returns]
If the method succeeds, the return value is zero (0). Other cases, returns non-zero.

7. MPos Lookup Reference

7-1 Over view

• The "**MPosLookup**" interface provides useful methods for searching and retrieving the information needed to connect devices (printers).

7-2 Methods



Use the "**current**" property in "**MPosLookupUtil**" class to call the methods defined in this interface.
eg) MPosLookupUtil.Current.refreshDevicesList()
eg) MPosLookupUtil.Current.getDeviceList()

7-2-1 Refresh Devices List

Start searching for devices available to connect with HOST.

[Syntax]
Task<int> refreshDevicesList (int interfaceType)

[Parameters]
• int interface type: communication type searching for devices.

MPosInterfaceType Enumeration	Value	Description
MPOS_INTERFACE_WIFI	1	WiFi (wireless LAN)
MPOS_INTERFACE_ETHERNET	2	Ethernet (wired LAN)
MPOS_INTERFACE_BLUETOOTH	4	Bluetooth
MPOS_INTERFACE_BLE	8	BLE
MPOS_INTERFACE_USB	16	USB

[Returns]
If the method succeeds, the return value is zero(0). Other cases, returns non-zero.

7-2-2 Get Devices List

Gets the data collctions containing the connection information searched by "**refreshDeviceList**" method.

[Parameters]
• int interface type: communication type for getting device list

MPosInterfaceType Enumeration	Value	Description
MPOS_INTERFACE_WIFI	1	WiFi (wireless LAN)
MPOS_INTERFACE_ETHERNET	2	Ethernet (wired LAN)
MPOS_INTERFACE_BLUETOOTH	4	Bluetooth
MPOS_INTERFACE_BLE	8	BLE
MPOS_INTERFACE_USB	16	USB

[Syntax]
ObservableCollection<MposConnectionInformation> getDevicesList (int interfaceType)

[Returns]
ObservableCollection<MposConnectionInformation>: data collctions containing the connection information.

```
class MposConnectionInformation
{
    public MPosInterfaceType IntefaceType { get; set; }
    public string Name { get; set; }
    public string Address
    public string MacAddress { get; set; }
    public string IpAddress { get; set; }
    public string PortNumber { get; set; }
    public string BluetoothDeviceId { get; set; }
    public string USBName { get; set; }
    public string ProductName { get; set; }
}
```

Properties
• InterfaceType: The communication type supported.
• Name: The display name for the device.
• Address: The address for the device (IP Address, S/N, or MAC address)
• MACAddress: MAC address or S/N for Bluetooth device.
• IPAddress: IP address for TCP communication.
• PortNumber: Port number for TCP communication.
• BluetoothDeviceId: Bluetooth deivce ID
• USBName: Device name for USB communication
• ProductName: USB device name

8. MPos Controller Label Printer Reference

8-1 Over view

• The MPosControllerLabelPrinter interface provides methods for controlling a Label printer.

8-2 Methods

8-2-1 Set Text Encoding

Sets the codepage used for encoding characters.

[Syntax]
int setTextEncoding (int textEncoding);

[Parameters]
• int textEncoding: Code page to use when encoding a character string.

MPosLabelCodePage Enumeration	Value	Description
MPOS_CODEPAGE_CP437	437	USA, Standard Europe (PC437)
MPOS_CODEPAGE_CP850	850	Western European (PC850)
MPOS_CODEPAGE_CP852	852	Latin 2 (PC852)
MPOS_CODEPAGE_CP860	860	Portuguese (PC860)
MPOS_CODEPAGE_CP863	863	Canadian-French (PC863)
MPOS_CODEPAGE_CP865	865	Nordic (PC865)
MPOS_CODEPAGE_WPC1252	1252	Latin 1 (WPC1252)
MPOS_CODEPAGE_CP857	857	Turkce (PC857)
MPOS_CODEPAGE_CP737	737	Greek (PC 737)
MPOS_CODEPAGE_WPC1250	1250	Central European (PC1250)
MPOS_CODEPAGE_WPC1253	1253	Greek (WPC1253)
MPOS_CODEPAGE_WPC1254	1254	Turkce (WPC1254)
MPOS_CODEPAGE_CP855	855	Cyrillic (PC855)
MPOS_CODEPAGE_CP862	862	Hebrew DOS code (PC862)
MPOS_CODEPAGE_CP866	866	Cyrillic #2 (PC866)
MPOS_CODEPAGE_WPC1251	1251	Cyrillic (WPC1251)
MPOS_CODEPAGE_WPC1255	1255	Hebrew New code (WPC1255)
MPOS_CODEPAGE_CP928	928	Greek (PC928)
MPOS_CODEPAGE_CP775	775	Baltic (PC775)
MPOS_CODEPAGE_WPC1257	1257	Baltic (WPC1257)
MPOS_CODEPAGE_CP858	858	Euro (PC858)
MPOS_CODEPAGE_CHINA_GB2312	936	Simplified Chinese (GB2312)
MPOS_CODEPAGE_CHINA_BIG5	950	Traditional Chinese (BIG5)
MPOS_CODEPAGE_KOREA_KS5601	949	Korean (KS5601)

MPOS_CODEPAGE_JAPAN_SHIFTJIS	932	Japanese (SHIFT JIS)

[Returns]
If the method succeeds, the return value is zero (0). Other cases, returns non-zero.

8-2-2 Set Characterset

Sets the codepage and international characterset used for encoding characters.

[Syntax]
Task<int> setCharacterset (int characterset, int internationalCharset);

[Parameters]
• int characterset: Code page to use when encoding a character string

MPosLabelCodePage Enumeration	Value	Description
MPOS_CODEPAGE_CP437	437	USA, Standard Europe (PC437)
MPOS_CODEPAGE_CP850	850	Western European (PC850)
MPOS_CODEPAGE_CP852	852	Latin 2 (PC852)
MPOS_CODEPAGE_CP860	860	Portuguese (PC860)
MPOS_CODEPAGE_CP863	863	Canadian-French (PC863)
MPOS_CODEPAGE_CP865	865	Nordic (PC865)
MPOS_CODEPAGE_WPC1252	1252	Latin 1 (WPC1252)
MPOS_CODEPAGE_CP857	857	Turkce (PC857)
MPOS_CODEPAGE_CP737	737	Greek (PC 737)
MPOS_CODEPAGE_WPC1250	1250	Central European (PC1250)
MPOS_CODEPAGE_WPC1253	1253	Greek (WPC1253)
MPOS_CODEPAGE_WPC1254	1254	Turkce (WPC1254)
MPOS_CODEPAGE_CP855	855	Cyrillic (PC855)
MPOS_CODEPAGE_CP862	862	Hebrew DOS code (PC862)
MPOS_CODEPAGE_CP866	866	Cyrillic #2 (PC866)
MPOS_CODEPAGE_WPC1251	1251	Cyrillic (WPC1251)
MPOS_CODEPAGE_WPC1255	1255	Hebrew New code (WPC1255)
MPOS_CODEPAGE_CP928	928	Greek (PC928)
MPOS_CODEPAGE_CP775	775	Baltic (PC775)
MPOS_CODEPAGE_WPC1257	1257	Baltic (WPC1257)
MPOS_CODEPAGE_CP858	858	Euro (PC858)
MPOS_CODEPAGE_CHINA_GB2312	936	Simplified Chinese (GB2312)
MPOS_CODEPAGE_CHINA_BIG5	950	Traditional Chinese (BIG5)
MPOS_CODEPAGE_KOREA_KS5601	949	Korean (KS5601)
MPOS_CODEPAGE_JAPAN_SHIFTJIS	932	Japanese (SHIFT JIS)

• int internationalCharset: International character set

MPosICS Enumeration	Value	Description
MPOS_ICS_USA	0	USA Code
MPOS_ICS_FRANCE	1	FRANCE Code
MPOS_ICS_GERMANY	2	GERMANY Code
MPOS_ICS_UK	3	UK Code
MPOS_ICS_DENMARK_I	4	DENMARK1 Code
MPOS_ICS_SWEDEN	5	SWEDEN Code
MPOS_ICS_ITALY	6	ITALY Code
MPOS_ICS_SPAIN_I	7	SPAIN1 Code
MPOS_ICS_JAPAN	8	JAPAN Code
MPOS_ICS_NORWAY	9	NORWAY Code
MPOS_ICS_DENMARK_II	10	DENMARK 2 Code
MPOS_ICS_SPAIN_II	11	SPAIN 2 Code
MPOS_ICS_LATIN_AMERICA	12	LATIN AMERICA Code
MPOS_ICS_KOREA	13	KOREA Code
MPOS_ICS_SLOVENIA_CROATIA	14	SLOVENIA Code
MPOS_ICS_CHINA	15	CHINA Code

[Returns]

If the method succeeds, the return value is zero (0). Other cases, returns non-zero.

8-2-3 Check Printer Status

Gets the printer's current status.

[Syntax]
Task<uint> checkPrinterStatus (void);;

[Returns]

MPosLabelPrinterStatus Enumeration	Value	Description
MPOS_LABEL_STATUS_IDLE	0	Printing is possible
MPOS_LABEL_STATUS_PRINTER_PAPEREMPTY	1	No paper
MPOS_LABEL_STATUS_PRINTER_COVEROPEN	2	Cover is open
MPOS_LABEL_STATUS_PRINTER_CUTTER_JAMMED	4	Cutter error
MPOS_LABEL_STATUS_PRINTER_TPH_OVERHEAT	8	TPH overheating
MPOS_LABEL_STATUS_PRINTER_GAP_ERROR	16	GAP sensor error
MPOS_LABEL_STATUS_PRINTER_RIBBON_END	32	No ribbon
MPOS_LABEL_STATUS_PRINTER_ON_BUILDING _LABEL_TO_BE_PRINTED	64	Building label buffer
MPOS_LABEL_STATUS_PRINTER_ON_PRINTING _LABEL	128	Printing
MPOS_LABEL_STATUS_PRINTER_ISSUED_LABEL _ISPAUSED	256	Paper stuck to filler
MPOS_LABEL_STATUS_PRINTER_WAIT_PAPER_TAKEN	2048	Printing pauses
MPOS_LABEL_STATUS_PRINTER_RFID_WRITING_ERROR	4096	RFID writing error

8-2-4 Print Buffer

Starts printing the contents saved in the printer

[Syntax]
Task<int> printBuffer (int numberOfCopies);

[Parameters]
• int numberOfCopies: The number of copies (Range: 1 to 65535)

[Returns]
If the method succeeds, the return value is zero (0). Other cases, returns non-zero.

8-2-5 Print Raw Data

Sends user-defined data to the printer.

[Syntax]
Task<int> printRawData(byte[] data);

[Parameters]
• byte[] data: Data to be sent

[Returns]
If the method succeeds, the return value is zero (0). Other cases, returns non-zero.

8-2-6 Draw Text Device Font

Saves data of the character strings to the printer buffer using the bitmap font.

[Syntax]
Task<int> drawTextDeviceFont (string data, int xPos, int yPos, char fontSelection, int fontWidth, int fontHeight, int rightSpace, int rotation, bool reverse, bool bold, bool rightToLeft, int alignment);

[Parameters]
• string data: data for characters to be printed
• int xPos: x axis coordinates of the character string
• int yPos: y axis axis coordinates of the character string
• char fontSelection: font selection

MPosLabelBitmapFont Enumeration	Value	Description
MPOS_LABEL_DEVICE_FONT_6PT	'0'	9 X 15 (dots)
MPOS_LABEL_DEVICE_FONT_8PT	'1'	12 X 20 (dots)
MPOS_LABEL_DEVICE_FONT_10PT	'2'	16 X 25 (dots)
MPOS_LABEL_DEVICE_FONT_12PT	'3'	19 X 30 (dots)
MPOS_LABEL_DEVICE_FONT_15PT	'4'	24 X 38 (dots)
MPOS_LABEL_DEVICE_FONT_20PT	'5'	32 X 40 (dots)
MPOS_LABEL_DEVICE_FONT_30PT	'6'	48 X 76 (dots)
MPOS_LABEL_DEVICE_FONT_14PT	'7'	22 X 34 (dots)
MPOS_LABEL_DEVICE_FONT_18PT	'8'	28 X 44 (dots)
MPOS_LABEL_DEVICE_FONT_24PT	'9'	37 X 58 (dots)
MPOS_LABEL_DEVICE_FONT_KOREAN1	'a'	16 X 16 (dots) (English. Numbers 9 X 15)
MPOS_LABEL_DEVICE_FONT_KOREAN2	'b'	24 X 24 (dots) (English. Numbers 12 X 24)
MPOS_LABEL_DEVICE_FONT_KOREAN3	'c'	20 X 20 (dots) (English. Numbers 12 X 20)
MPOS_LABEL_DEVICE_FONT_KOREAN4	'd'	26 X 26 (dots) (English. Numbers 16 X 30)
MPOS_LABEL_DEVICE_FONT_KOREAN5	'e'	20 X 26 (dots) (English. Numbers 16 X 30)
MPOS_LABEL_DEVICE_FONT_KOREAN6	'f'	38 X 38 (dots) (English. Numbers 22 X 34)
MPOS_LABEL_DEVICE_FONT_GB2312	'm'	24 X 24 (dots) (English. Numbers 12 X 24)
MPOS_LABEL_DEVICE_FONT_BIG5	'n'	24 X 24 (dots) (English. Numbers 12 X 24)
MPOS_LABEL_DEVICE_FONT_SHIFT_JIS	'j'	24 X 24 (dots) (English. Numbers 12 X 24)

• int fontWidth: The width expansion ratio of a character. (Range: 1 to 4)
• int fontHeight: The height expansion ratio of a character. (Range: 1 to 4)
• int rightSpace: The space on the right of a character (default value: 0)
• int rotation: The printing direction of characters

MPosLabelRotation Enumeration	Value	Description
MPOS_LABEL_ROTATION_DEGREES_0	0	No rotated
MPOS_LABEL_ROTATION_DEGREES_90	1	Rotated 90 degree (clockwise)
MPOS_LABEL_ROTATION_DEGREES_180	2	Rotated 180 degree
MPOS_LABEL_ROTATION_DEGREES_270	3	Rotated 270 degre (clockwise)

• bool reverse: prints characters with reverse property or not
• bool bold: prints characters with bold property or not
• bool rightToLeft: direction of printing characters (left to right or right to left)
• int alignment: alignment

MPosLabelAlignment Enumeration	Value	Description
MPOS_LABEL_ALIGNMENT_LEFT	0	Align to the left
MPOS_LABEL_ALIGNMENT_RIGHT	1	Align to the right

[Returns]
If the method succeeds, the return value is zero (0). Other cases, returns non-zero.

8-2-7 Draw Text Vector Font

Saves data of the character strings to the printer buffer using the vector font.

[Syntax]
Task<int> drawTextVectorFont (string data, int xPos, int yPos, char fontSelection, int fontWidth, int fontHeight, int rightSpace, int rotation, bool reverse, bool bold, bool italic, bool rightToLeft, int alignment)

[Parameters]
• string data: data for characters to be printed
• int xPos: x axis coordinates of the character string
• int yPos: y axis axis coordinates of the character string
• char fontSelection: font selection

MPosLabelVectorFont Enumeration	Value	Description
MPOS_LABEL_VECTOR_FONT_ASCII	'U'	ASCII (1Byte code)
MPOS_LABEL_VECTOR_FONT_KS5601	'K'	KS5601 (2Byte code)
MPOS_LABEL_VECTOR_FONT_BIG5	'B'	BG5 (2Byte code)
MPOS_LABEL_VECTOR_FONT_GB2312	'G'	GB2312 (2Byte code)
MPOS_LABEL_VECTOR_FONT_SHIFT_JIS	'J'	Shift-JIS (2Byte code)
MPOS_LABEL_VECTOR_FONT_OCR_A	'a'	OCR-A (1Byte code)
MPOS_LABEL_VECTOR_FONT_OCR_B	'b'	OCR-B (1Byte code)

• int fontWidth: character width
• int fontHeight: character height
• int rightSpace: right space of character
• int rotation: printing direction

MPosLabelRotation Enumeration	Value	Description
MPOS_LABEL_ROTATION_DEGREES_0	0	No rotated
MPOS_LABEL_ROTATION_DEGREES_90	1	Rotated 90 degree (clockwise)
MPOS_LABEL_ROTATION_DEGREES_180	2	Rotated 180 degree
MPOS_LABEL_ROTATION_DEGREES_270	3	Rotated 270 degre (clockwise)

• bool reverse: prints characters with reverse property or not
• bool bold: prints characters with bold property or not
• bool italic: prints characters with italic property or not
• bool rightToLeft: direction of printing characters (left to right or right to left)

• int alignment: alignment

MPosLabelAlignment Enumeration	Value	Description
MPOS_LABEL_ALIGNMENT_LEFT	0	Align to the left
MPOS_LABEL_ALIGNMENT_RIGHT	1	Align to the right
MPOS_LABEL_ALIGNMENT_CENTER	2	Align at the center

[Returns]

If the method succeeds, the return value is zero (0). Other cases, returns non-zero.

8-2-8 Draw Barcode 1D

Saves data of the 1d barcode to the printer buffer.

[Syntax]
Task<int> drawBarcode1D(string data, int xPos, int yPos, int barcodeType, int widthNarrow, int widthWide, int height, int hri, int quietZoneWidth, int rotation)

[Parameters]
• string data: data of the barcode
• int xPos: x axis coordinates of barcode
• int yPos: y axis coordinates of barcode
• int barcodeType: Type of barcode
• int widthNarrow: Narrow bar width.
• int widthWide: Wide bar width
• int height: Barcode height
• int hri: HRI (Human Readable Interface) printing position

Code	Value	Description
MPOS_BARCODE_TEXTNONE	0	No HRI print
MPOS_BARCODE_TEXTBELOW	1	above barcode
MPOS_BARCODE_TEXTABOVE	2	below barcode
MPOS_BARCODE_TEXTBELOW_FONTSIZE2	3	above barcode (font size 2)
MPOS_BARCODE_TEXTABOVE_FONTSIZE2	4	below barcode (font size 2)
MPOS_BARCODE_TEXTBELOW_FONTSIZE3	5	above barcode (font size 3)
MPOS_BARCODE_TEXTABOVE_FONTSIZE3	6	below barcode (font size 3)
MPOS_BARCODE_TEXTBELOW_FONTSIZE4	7	above barcode (font size 4)
MPOS_BARCODE_TEXTABOVE_FONTSIZE4	8	below barcode (font size 4)

• int quietZoneWidth: Quiet zone width (Range of 0 to20)

• int rotation: printing direction of barcode

MPosLabelRotation Enumeration	Value	Description
MPOS_LABEL_ROTATION_DEGREES_0	0	No rotated
MPOS_LABEL_ROTATION_DEGREES_90	1	Rotated 90 degree (clockwise)
MPOS_LABEL_ROTATION_DEGREES_180	2	Rotated 180 degree
MPOS_LABEL_ROTATION_DEGREES_270	3	Rotated 270 degree (clockwise)

[Returns]

If the method succeeds, the return value is zero (0). Other cases, returns non-zero.

8-2-9 Draw Barcode Maxi Code

Saves data of the Maxicode barcode to the printer buffer.

[Syntax]
Task<int> drawBarcodeMaxiCode (string data, int xPos, int yPos, int mode)

[Parameters]
• string data: data of the barcode
• int xPos: x axis coordinates of barcode
• int yPos: y axis coordinates of barcode
• int mode: Maxicode mode

MPosLabelMaxicodeMode Enumeration	Value	Description
MPOS_LABEL_MAXICODE_MODE_0	0	MaxiCode Mode 0
MPOS_LABEL_MAXICODE_MODE_2	2	MaxiCode Mode 2
MPOS_LABEL_MAXICODE_MODE_3	3	MaxiCode Mode 3
MPOS_LABEL_MAXICODE_MODE_4	4	MaxiCode Mode 4

[Returns]
If the method succeeds, the return value is zero (0). Other cases, returns non-zero.

8-2-10 Draw Barcode PDF 417

Saves data of the PDF417 barcode to the printer buffer.

[Syntax]
Task<int> drawBarcodePDF417(string data, int xPos, int yPos, int maximumRowCount, int maximumColumnCount, int errorCorrectionLevel, int dataCompressionMethod, bool hri, int barcodeOriginPoint, int moduleWidth, int barHeight, int rotation)

[Parameters]
• string data: data of the barcode
• int xPos: x axis coordinates of barcode
• int yPos: y axis coordinates of barcode
• int maximumRowCount: Number of barcode Rows. (Range: 3 to 90)
• int maximumColumnCount: Number of barcode Columns. (Range: 1 to 30)
• int errorCorrectionLevel: Error correction level (Recognition rate when a part of a barcode is damaged)

MPosLabelPDF417ECL Enumeration	Value	Description
MPOS_LABEL_ERROR_CORRECTION_LEVEL0	0	Error correction level 0
MPOS_LABEL_ERROR_CORRECTION_LEVEL1	1	Error correction level 1
MPOS_LABEL_ERROR_CORRECTION_LEVEL2	2	Error correction level 2
MPOS_LABEL_ERROR_CORRECTION_LEVEL3	3	Error correction level 3
MPOS_LABEL_ERROR_CORRECTION_LEVEL4	4	Error correction level 4
MPOS_LABEL_ERROR_CORRECTION_LEVEL5	5	Error correction level 5
MPOS_LABEL_ERROR_CORRECTION_LEVEL6	6	Error correction level 6
MPOS_LABEL_ERROR_CORRECTION_LEVEL7	7	Error correction level 7
MPOS_LABEL_ERROR_CORRECTION_LEVEL8	8	Error correction level 8

• int dataCompressionMethod: data compression method

PosLabelPDF41Compress Enumeration	Value	Description
MPOS_LABEL_PDF417_DATA _COMPRESSION_TEXT	0	2 characters per codeword
MPOS_LABEL_PDF417_DATA _COMPRESSION_NUMERIC	1	2.93 characters per codeword
MPOS_LABEL_PDF417_DATA _COMPRESSION_BINARY	2	1.2 characters per codeword

• int hri: HRI (Human Readable Interface) printing position

MPosLabelHRI Enumeration	Value	Description
MPOS_BARCODE_TEXTNONE	0	Does not print
MPOS_BARCODE_TEXTBELOW	2	Print below barcode

• int startPosition: starting point of a barcode (default value: 1)

MPosLabelPDF41StartPosition Enumeration	Value	Description
MPOS_LABEL_BARCODE_ORIGIN_POINT_CENTER	0	Coordinate value based on middle of barcode
MPOS_LABEL_BARCODE_ORIGIN_POINT_UPPER	1	Coordinate value based on top left of barcode

• int moduleWidth: width of the barcode module. (Range: 2 to 9)
• int barHeight: height of the bar. (Range: 4 to 99)
• int rotation: printing direction of barcode

MPosLabelRotation Enumeration	Value	Description
MPOS_LABEL_ROTATION_DEGREES_0	0	No rotated
MPOS_LABEL_ROTATION_DEGREES_90	1	Rotated 90 degree (clockwise)
MPOS_LABEL_ROTATION_DEGREES_180	2	Rotated 180 degree
MPOS_LABEL_ROTATION_DEGREES_270	3	Rotated 270 degree (clockwise)

[Returns]
If the method succeeds, the return value is zero (0). Other cases, returns non-zero.

8-2-11 Draw Barcode QR Code

Saves data of the QRCode barcode to the printer buffer.

[Syntax]
Task<int> drawBarcodeQRCode (string data, int xPos, int yPos, int size, int model, int errorCorrectionLevel, int rotation)

[Parameters]
• string data: data of the barcode
• int xPos: x axis coordinates of barcode
• int yPos: y axis coordinates of barcode
• int size: QRCode size (Range of 1 to 9)
• int model: QRCode Model

MPosLabelQRCodeModel Enumeration	Value	Description
MPOS_LABEL_QRCODE_MODEL_1	1	Model 1
MPOS_LABEL_QRCODE_MODEL_2	2	Model 2

• int errorCorrectionLevel: Error correction level

MPosLabelQRCodeECL 열거형	Value	Description
MPOS_LABEL_QRCODE_ECCLEVEL_L	'L'	Error correction rate is configured to 7%
MPOS_LABEL_QRCODE_ECCLEVEL_M	'M'	Error correction rate is configured to 15%
MPOS_LABEL_QRCODE_ECCLEVEL_Q	'Q'	Error correction rate is configured to 25%
MPOS_LABEL_QRCODE_ECCLEVEL_H	'H'	Error correction rate is configured to 35%

• int rotation: printing direction of barcode

MPosLabelRotation Enumeration	Value	Description
MPOS_LABEL_ROTATION_DEGREES_0	0	No rotated
MPOS_LABEL_ROTATION_DEGREES_90	1	Rotated 90 degree (clockwise)
MPOS_LABEL_ROTATION_DEGREES_180	2	Rotated 180 degree
MPOS_LABEL_ROTATION_DEGREES_270	3	Rotated 270 degree (clockwise)

[Returns]
If the method succeeds, the return value is zero (0). Other cases, returns non-zero.

8-2-12 Draw Barcode Data Matrix

Saves data of the DataMatrix barcode to the printer buffer.

[Syntax]
Task<int> drawBarcodeDataMatrix (string data, int xPos, int yPos, int size, bool reverse, int rotation);

[Parameters]
• string data: data of the barcode
• int xPos: x axis coordinates of barcode
• int yPos: y axis coordinates of barcode
• int size: DataMatrix size (Range of 1 to 4)
• int reverse: barcode reverse or normal
• int rotation: printing direction of barcode

MPosLabelRotation Enumeration	Value	Description
MPOS_LABEL_ROTATION_DEGREES_0	0	No rotated
MPOS_LABEL_ROTATION_DEGREES_90	1	Rotated 90 degree (clockwise)
MPOS_LABEL_ROTATION_DEGREES_180	2	Rotated 180 degree
MPOS_LABEL_ROTATION_DEGREES_270	3	Rotated 270 degree (clockwise)

[Returns]
If the method succeeds, the return value is zero (0). Other cases, returns non-zero.

8-2-13 Draw Barcode Aztec

Saves data of the Aztec barcode to the printer buffer.

[Syntax]
Task<int> drawBarcodeAztec (string data, int xPos, int yPos, int size, bool extended Channel, int errorCorrectionLevel, bool menuSymbol, int numberOfSymbols, string optionalID, int rotation)

[Parameters]
• string data: data of the barcode
• int xPos: x axis coordinates of barcode
• int yPos: y axis coordinates of barcode
• int size: Aztec size (Range of 1 to 10)
• bool extendedChannel: Extended channel interpretation code
• int errorCorrectionLevel: Error correction level

Code	Description
0	Automatically configures the error correction level.
1~99	Directly enter the error correction level.
101~104	1~4 layer compact symbol
201~232	1~32 layer compact symbol
300	Simple Aztec ”Rune”

• bool menuSymbol: Configures the menu symbol
• int numberOfSymbols: Number of symbols for structured append (Range of 1 to 26)
• string optionalID: Optional ID field for structured append: ID field string (Maximum 24 letters)
• int rotation: printing direction of barcode

MPosLabelRotation Enumeration	Value	Description
MPOS_LABEL_ROTATION_DEGREES_0	0	No rotated
MPOS_LABEL_ROTATION_DEGREES_90	1	Rotated 90 degree (clockwise)
MPOS_LABEL_ROTATION_DEGREES_180	2	Rotated 180 degree
MPOS_LABEL_ROTATION_DEGREES_270	3	Rotated 270 degree (clockwise)

[Returns]
If the method succeeds, the return value is zero (0). Other cases, returns non-zero.

8-2-14 Draw Barcode Code 49

Saves data of the Code49 barcode to the printer buffer.

[Syntax]
Task<int> drawBarcodeCode49(string data, int xPos, int yPos, int widthNarrow, int widthWide, int height, int hri, int startingMode, int rotation)

[Parameters]
• string data: data of the barcode
• int xPos: x axis coordinates of barcode
• int yPos: y axis coordinates of barcode
• int widthNarrow: Width of Narrow Bar
• int widthWide: Width of Wide Bar
• int height: Barcode height
• int hri: HRI (Human Readable Interface) printing position.

MPosLabelHRI Enumeration	Value	Description
MPOS_BARCODE_TEXTNONE	0	Does not print
MPOS_BARCODE_TEXTABOVE	1	Print above barcode
MPOS_BARCODE_TEXTBELOW	2	Print below barcode

• int startingMode: Starting Mode of a barcode

MPosLabelCode49Mode Enumeration	Value	Description
MPOS_LABEL_STARTINGMODE_REGULAR_ALPHANUMERIC	0	Regular Alphanumeric Mode
MPOS_LABEL_STARTINGMODE_MULTIPLE_READ_ALPHANUMERIC	1	Multiple Read Alphanumeric
MPOS_LABEL_STARTINGMODE_REGULAR_NUMERIC	2	Regular Numeric Mode
MPOS_LABEL_STARTINGMODE_GROUP_ALPHANUMERIC	3	Group Alphanumeric Mode
MPOS_LABEL_STARTINGMODE_REGULAR_ALPHANUMRIC_SHIFT1	4	Regular Alphanumeric Shift 1
MPOS_LABEL_STARTINGMODE_REGULAR_ALPHANUMRIC_SHIFT2	5	Regular Alphanumeric Shift 2
MPOS_LABEL_STARTINGMODE_AUTOMATIC	7	Automatic Mode

• int rotation: printing direction of barcode

MPosLabelRotation Enumeration	Value	Description
MPOS_LABEL_ROTATION_DEGREES_0	0	No rotated
MPOS_LABEL_ROTATION_DEGREES_90	1	Rotated 90 degree (clockwise)
MPOS_LABEL_ROTATION_DEGREES_180	2	Rotated 180 degree
MPOS_LABEL_ROTATION_DEGREES_270	3	Rotated 270 degree (clockwise)

[Returns]

If the method succeeds, the return value is zero (0). Other cases, returns non-zero.

8-2-15 Draw Barcode Coda Block

Saves data of the CodaBlock barcode to the printer buffer.

[Syntax]
Task<int> drawBarcodeCodaBlock (string data, int xPos, int yPos, int widthNarrow, int widthWide, int height, bool securityLevel, int numberOfCharactersPerRow, char mode, int numberOfRowToEncode)

[Parameters]
• string data: data of the barcode
• int xPos: x axis coordinates of barcode
• int yPos: y axis coordinates of barcode
• int widthNarrow: Width of Narrow Bar
• int widthWide: Width of Wide Bar
• int height: Barcode height
• bool securityLevel: Barcode security level
• int numberOfCharactersPerRow: Number of characters per row(Range of 2 to 62)
• char mode: Codablock mode

MPosLabelCodaBlockMode Enumeration	Value	Description
MPOS_LABEL_CODABLOCK_MODE_A	'A'	Creates a code that uses the Code 39 character set
MPOS_LABEL_CODABLOCK_MODE_E	'E'	Creates a code that uses the Code 128 character set
MPOS_LABEL_CODABLOCK_MODE_F	'F'	Creates a code that uses the Code 128 character set and that automatically has func1 added to

• int numberOfRowToEncode: The following values can be used for each mode.

MPosLabelCodaBlockMode Enumeration	Value	Description
MPOS_LABEL_CODABLOCK_MODE_A	'A'	Can use a number between 1 and 18
MPOS_LABEL_CODABLOCK_MODE_E'	'E'	Can use a number between 2 and 4
MPOS_LABEL_CODABLOCK_MODE_F	'F'	Can use a number between 2 and 4

[Returns]
If the method succeeds, the return value is zero (0). Other cases, returns non-zero.

8-2-16 Draw Barcode Micro PDF

Saves data of the MicroPDF barcode to the printer buffer.

[Syntax]
Task<int> drawBarcodeMicroPDF (string data, int xPos, int yPos, int moduleWidth, int moduleHeight, int mode, int rotation)

[Parameters]
• string data: data of the barcode
• int xPos: x axis coordinates of barcode
• int yPos: y axis coordinates of barcode
• int moduleWidth: module width (Range of 2 to 8)
• int moduleHeight: barcode height (Range of 1 to 99)
• int mode: barcode mode (Range of 0 to 33)
• int rotation: printing direction of barcode

MPosLabelRotation Enumeration	Value	Description
MPOS_LABEL_ROTATION_DEGREES_0	0	No rotated
MPOS_LABEL_ROTATION_DEGREES_90	1	Rotated 90 degree (clockwise)
MPOS_LABEL_ROTATION_DEGREES_180	2	Rotated 180 degree
MPOS_LABEL_ROTATION_DEGREES_270	3	Rotated 270 degree (clockwise)

[Returns]
If the method succeeds, the return value is zero (0). Other cases, returns non-zero.

8-2-17 Draw Barcode IMB

Saves data of the IMB barcode to the printer buffer.

[Syntax]
Task<int> drawBarcodeIMB (string data, int xPos, int yPos, bool hri, int rotation)

[Parameters]
• string data: data of the barcode
• int xPos: x axis coordinates of barcode
• int yPos: y axis coordinates of barcode
• bool hri: Determines whether or not to print HRI (Human Readable Interface)
• int rotation: printing direction of barcode

MPosLabelRotation Enumeration	Value	Description
MPOS_LABEL_ROTATION_DEGREES_0	0	No rotated
MPOS_LABEL_ROTATION_DEGREES_90	1	Rotated 90 degree (clockwise)
MPOS_LABEL_ROTATION_DEGREES_180	2	Rotated 180 degree
MPOS_LABEL_ROTATION_DEGREES_270	3	Rotated 270 degree (clockwise)

[Returns]
If the method succeeds, the return value is zero (0). Other cases, returns non-zero.

8-2-18 Draw Barcode MSI

Saves data of the MSI barcode to the printer buffer.

[Syntax]
Task<int> drawBarcodeMSI (string data, int xPos, int yPos, int narrowWidth, int widthWide, int height, int checkDigitSelection, bool printCheckDigit, int hri, int rotation);

[Parameters]
• string data: data of the barcode
• int xPos: x axis coordinates of barcode
• int yPos: y axis coordinates of barcode
• int widthNarrow: Width of Narrow Bar
• int widthWide: Width of Wide Bar
• int height: Barcode height
• int checkDigitSelection: Check digit mode

MPosLabelMSICheckDigit Enumeration	Value	Description
MPOS_LABEL_CHECKDIGIT_NONE	0	Check digit not configured
MPOS_LABEL_CHECKDIGIT_1MOD10	1	1 Mod 10
MPOS_LABEL_CHECKDIGIT_2MOD10	2	2 Mode 10
MPOS_LABEL_CHECKDIGIT_1MOD11_AND_1MOD_10	3	1 Mod 11 and Mod 10

• bool printCheckDigit: Determines whether or not to print check digit
• int hri: HRI (Human Readable Interface) printing position.

MPosLabelHRI Enumeration	Value	Description
MPOS_BARCODE_TEXTNONE	0	Does not print
MPOS_BARCODE_TEXTABOVE	1	Print above barcode
MPOS_BARCODE_TEXTBELOW	2	Print below barcode

• int rotation: printing direction of barcode

MPosLabelRotation Enumeration	Value	Description
MPOS_LABEL_ROTATION_DEGREES_0	0	No rotated
MPOS_LABEL_ROTATION_DEGREES_90	1	Rotated 90 degree (clockwise)
MPOS_LABEL_ROTATION_DEGREES_180	2	Rotated 180 degree
MPOS_LABEL_ROTATION_DEGREES_270	3	Rotated 270 degree (clockwise)

[Returns]
If the method succeeds, the return value is zero (0). Other cases, returns non-zero.

8-2-19 Draw Barcode Plessey

Saves data of the Plessey barcode to the printer buffer.

[Syntax]
Task<int> drawBarcodePlessey (string data, int xPos, int yPos, int narrowWidth, int widthWide, int height, bool printCheckDigit, int hri, int rotation)

[Parameters]
• string data: data of the barcode
• int xPos: x axis coordinates of barcode
• int yPos: y axis coordinates of barcode
• int widthNarrow: Width of Narrow Bar
• int widthWide: Width of Wide Bar
• int height: Barcode height
• bool printCheckDigit: Determines whether or not to print check digit
• int hri: HRI (Human Readable Interface) printing position.

Code	Value	Description
MPOS_BARCODE_TEXTNONE	0	Does not print
MPOS_BARCODE_TEXTABOVE	1	Print above barcode
MPOS_BARCODE_TEXTBELOW	2	Print below barcode

• int rotation: printing direction of barcode

MPosLabelRotation Enumeration	Value	Description
MPOS_LABEL_ROTATION_DEGREES_0	0	No rotated
MPOS_LABEL_ROTATION_DEGREES_90	1	Rotated 90 degree (clockwise)
MPOS_LABEL_ROTATION_DEGREES_180	2	Rotated 180 degree
MPOS_LABEL_ROTATION_DEGREES_270	3	Rotated 270 degree (clockwise)

[Returns]
If the method succeeds, the return value is zero (0). Other cases, returns non-zero.

8-2-20 Draw Barcode TLC39

Saves data of the TLC39 barcode to the printer buffer.

[Syntax]
Task<int> drawBarcodeTLC39(string data, int xPos, int yPos, int narrowWidth,
int widthWide, int height, int rowHeightOfMicroPDF417, int narrowWidthOfMicroPDF417,
int rotation)

[Parameters]
• string data: data of the barcode
• int xPos: x axis coordinates of barcode
• int yPos: y axis coordinates of barcode
• int narrowWidth: Width of Narrow Bar
• int widthWide: Width of Wide Bar
• int height: Barcode height
• int rowHeightOfMicroPDF417: Height of Micro PDF417's row (Range of 1 to 255)
• int narrowWidthOfMicroPDF417: Width of Micro PDF417's Narrow bar
(Range of 1 to 10)
• int rotation: Printing direction of barcode to be printed

MPosLabelRotation Enumeration	Value	Description
MPOS_LABEL_ROTATION_DEGREES_0	0	No rotated
MPOS_LABEL_ROTATION_DEGREES_90	1	Rotated 90 degree (clockwise)
MPOS_LABEL_ROTATION_DEGREES_180	2	Rotated 180 degree
MPOS_LABEL_ROTATION_DEGREES_270	3	Rotated 270 degree (clockwise)

[Returns]
If the method succeeds, the return value is zero (0). Other cases, returns non-zero.

8-2-21 Draw Barcode RSS

Saves data of the RSS barcode to the printer buffer.

[Syntax]
Task<int> drawBarcodeRSS (string data, int xPos, int yPos, int barcodeType, int magnification, .int separatorHeight, int barcodeHeight, int segmentWidth, int rotation)

[Parameters]
• string data: data of the barcode
• int xPos: x axis coordinates of barcode
• int yPos: y axis coordinates of barcode
• int barcodeType: Configures the barcode type.

MPosLabelRSSType Enumeration	Value	Description
MPOS_LABEL_RSS_BARCODE_TYPE_RSS14	0	RSS14
MPOS_LABEL_RSS_BARCODE_TYPE_RSS14_TRUNCATED	1	RSS14 truncated
MPOS_LABEL_RSS_BARCODE_TYPE_RSS14_STACKED	2	RSS14 stacked
MPOS_LABEL_RSS_BARCODE_TYPE_RSS14_STACKED_OMNIDIRECTIONAL	3	RSS14 Stacked omnidirectional
MPOS_LABEL_RSS_BARCODE_TYPE_RSS_LIMITED	4	RSS limited
MPOS_LABEL_RSS_BARCODE_TYPE_RSS_EXPANDED	5	RSS Expanded
MPOS_LABEL_RSS_BARCODE_TYPE_UPC_A	6	RSS UPC A
MPOS_LABEL_RSS_BARCODE_TYPE_UPC_E	7	RSS UPC E
MPOS_LABEL_RSS_BARCODE_TYPE_EAN13	8	EAN13
MPOS_LABEL_RSS_BARCODE_TYPE_EAN8	9	EAN 8
MPOS_LABEL_RSS_BARCODE_TYPE_UCC_EAN128_CC_A_B	10	EAN128 CC-A/B
MPOS_LABEL_RSS_BARCODE_TYPE_UCC_EAN128_CC_C	11	EAN128 CC-C

• int magnification: Can use a number between 1 and 10
• int separatorHeight: Configures the height of an RSS barcode identifier (Range of 0 to 22)
• int barcodeHeight: Configures the barcode height
• int segmentWidth: Configures segment width (even number between 0 and 22)

• int rotation: printing direction of barcode

MPosLabelRotation Enumeration	Value	Description
MPOS_LABEL_ROTATION_DEGREES_0	0	No rotated
MPOS_LABEL_ROTATION_DEGREES_90	1	Rotated 90 degree (clockwise)
MPOS_LABEL_ROTATION_DEGREES_180	2	Rotated 180 degree
MPOS_LABEL_ROTATION_DEGREES_270	3	Rotated 270 degree (clockwise)

[Returns]

If the method succeeds, the return value is zero (0). Other cases, returns non-zero.

8-2-22 Draw Block

Saves data of a quadrangle or line to the printer buffer.

[Syntax]
Task<int> drawBlock (int startPosX, int startPosY, int endPosX, int endPosY, string option, int thickness)

[Parameters]
• int startPosX: x axis coordinates of top left
• int startPosY: y axis coordinates of top left
• int endPosX: x axis coordinates of bottom right
• int endPosY: y axis coordinates of bottom right
• char option: drawing option

MPosLabelBlock Enumeration	Value	Description
MPOS_LABEL_DRAW_BLOCK_OPTION_LINE_OVERWRITING	'O'	Repeat drawing On places where lines overlap
MPOS_LABEL_DRAW_BLOCK_OPTION_LINE_EXCLUSIVE_OR	'E'	Do not draw on places where lines overlap
MPOS_LABEL_DRAW_BLOCK_OPTION_LINE_DELETE	'D'	Delete line
MPOS_LABEL_DRAW_BLOCK_OPTION_SLOPE	'S'	Diagonal line
MPOS_LABEL_DRAW_BLOCK_OPTION_BOX	'B'	Quadrangle edge

• int thickness: Selects line thickness. This only applies if the 'option' value is 'S' or 'B'.

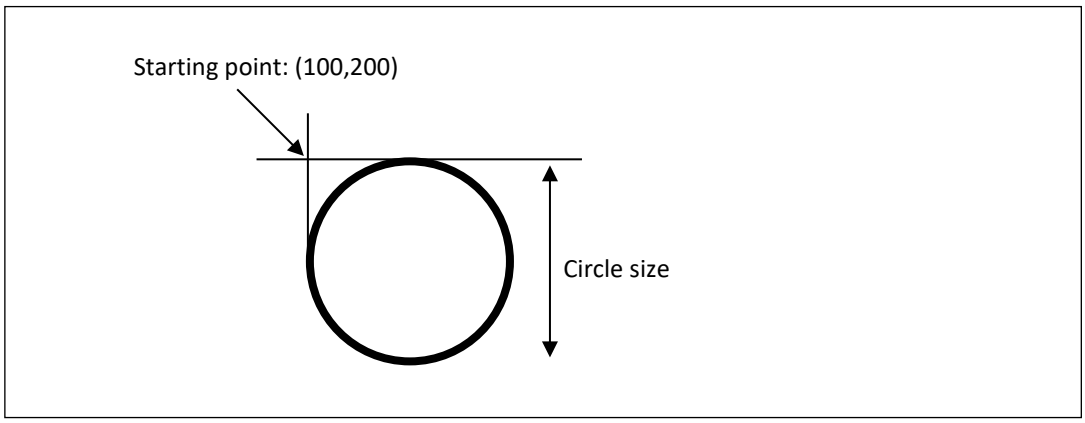
[Returns]
If the method succeeds, the return value is zero (0). Other cases, returns non-zero.

8-2-23 Draw Circle

Saves data of a circle to the printer buffer.

[Syntax]
Task<int> drawCircle (int startPosX, int startPosY, int sizeSelection, int multiplier)

[Parameters]
• int startPosX: x axis coordinates of starting point* of circle region
Add starting point (intersection point of tangent of left and top) description
• int startPosY: y axis coordinates of starting point* of circle region
* Starting point: Intersection point of tangent of left and top



• int sizeSelection: size of circle

MPosLabelCircle Enumeraton	Value	W x H Size (dots)
MPOS_LABEL_DRAW_CIRCLE_SIZE_40X40	1	40 × 40
MPOS_LABEL_DRAW_CIRCLE_SIZE_56X56	2	56 × 56
MPOS_LABEL_DRAW_CIRCLE_SIZE_72X72	3	72 × 72
MPOS_LABEL_DRAW_CIRCLE_SIZE_88X88	4	88 × 88
MPOS_LABEL_DRAW_CIRCLE_SIZE_104X104	5	104 × 104
MPOS_LABEL_DRAW_CIRCLE_SIZE_168X168	6	168 × 168

• int multiplier: Expands a circle by the scaling unit (Range of 1 to 4)

[Returns]
If the method succeeds, the return value is zero (0). Other cases, returns non-zero.

8-2-24 Draw Image

Saves data of an image to the printer buffer.

[Syntax]
Task<int> drawImage (object bitmapSrc, int startPosX, int startPosY, int width, int brightness, bool isDithering, bool isCompress)

[Parameters]
• object bitmapSrc: A platform dependent image object.

iOS	UIKit.UIImage
Android	Android.Graphics.Bitmap
Windows	Windows.UI.Xaml.Media.Imaging.WriteableBitmap Windows.UI.Xaml.Media.Imaging.BitmapImage (deprecated)

• int startPosX: x axis coordinates of top left of image to be printed
• int startPosY: y axis coordinates of top left of image to be printed
• int width: Width of printed image
• int brightness: The brightness level (Range: 0 ≤ brightness ≤ 100)

Platform	Brightness Range
Android, UWP	0 ≤ brightness ≤ 100
iOS	-100 ≤ brightness ≤ 100

• bool isDithering: Whether or not to apply dithering
• bool isCompress: Whether or not to compress data

[Returns]
If the method succeeds, the return value is zero (0). Other cases, returns non-zero.

8-2-25 Draw Image File

Saves data of the image file specified by the path to the printer buffer.

[Syntax]
Task<int> drawImageFile (string filePath, int startPosX, int startPosY, int width, int brightness, bool isDithering, bool isCompress)

[Parameters]
• string filePath: The path of the image file



In Android, the image file should be in the SD card.

• int startPosX: x axis coordinates of top left of image to be printed
• int startPosY: y axis coordinates of top left of image to be printed
• int width: Width of printed image
• int brightness: The brightness level

Platform	Brightness Range
Android, UWP	0 ≤ brightness ≤ 100
iOS	-100 ≤ brightness ≤ 100

• bool isDithering: Whether or not to apply dithering
• bool isCompress: Whether or not to compress data

[Returns]
If the method succeeds, the return value is zero (0). Other cases, returns non-zero.

8-2-26 Set Printing Type

Sets the printing type of the printer.



For Bixolon label printers, printers using the direct Thermal method cannot be configured to the Transfer Thermal method.

[Syntax]
Task<int> setPrintingType (char printingType)

[Parameters]
• char printingType: Configures the printing typ

MPosLabelPrintingType Enumeration	Value	Description
MPOS_LABEL_PRINTING_TYPE_DIRECT_THERMAL	'd'	Direct Thermal
MPOS_LABEL_PRINTING_TYPE_THERMAL_TRANSFER	't'	Transfer Thermal

[Returns]
If the method succeeds, the return value is zero (0). Other cases, returns non-zero.

8-2-27 Set Margin

Sets the margins in the printing area.

[Syntax]
Task<int> setMargin (int horizontalMargin, int verticalMargin)

[Parameters]
• int horizontalMargin: Left margin
• int verticalMargin: Top margin

[Returns]
If the method succeeds, the return value is zero (0). Other cases, returns non-zero.

8-2-28 Set Length

Sets the label length, gap, media type, and offset.

[Syntax]
Task<int> setLength (int labelLength, int gapLength, char mediaType, int offsetLength)

[Parameters]
• int labelLength: Media length
• int gapLength: Gap length of the media
• char mediaType: Media type

MPosLabelMediaType Enumeration	Value	Description
MPOS_LABEL_MEDIA_TYPE_GAP	'G'	Gap paper
MPOS_LABEL_MEDIA_TYPE_CONTINUOUS	'C'	Continuous paper
MPOS_LABEL_MEDIA_TYPE_BLACK_MARK	'B'	Black mark paper

• int offsetLength: Offset length

[Returns]
If the method succeeds, the return value is zero (0). Other cases, returns non-zero.

8-2-29 Set Width

Sets the print width.

[Syntax]
Task<int> setWidth (int labelWidth)

[Parameters]
• int labelWidth: print width

[Returns]
If the method succeeds, the return value is zero (0). Other cases, returns non-zero.

8-2-30 Set Speed

Sets the printing speed.

[Syntax]
Task<int> setSpeed (int speed)

[Parameters]
• int speed: printing speed

[Returns]
If the method succeeds, the return value is zero (0). Other cases, returns non-zero.

8-2-31 Set Density

Sets the print density.

[Syntax]
Task<int> setDensity (int density)

[Parameters]
• int density: print density (Range of 1 to 20)

[Returns]
If the method succeeds, the return value is zero (0). Other cases, returns non-zero.

8-2-32 Set Orientation

Sets the printing orientation of the label.

[Syntax]
Task<int> setOrientation (char orientation)

[Parameters]
• char orientation: Printing orientation

MPosLabelOrientation Enumeration	Value	Description
MPOS_LABEL_PRINTING_DIRECTION _TOP_TO_BOTTOM	'T'	Print from top to bottom
MPOS_LABEL_PRINTING_DIRECTION _BOTTOM_TO_TOP	'B'	Print from bottom to top

[Returns]
If the method succeeds, the return value is zero (0). Other cases, returns non-zero.

8-2-33 Set Offset

Sets offset length between black mark (or gap) and dotted lines.

[Syntax]
Task<int> setOffset (int offset)

[Parameters]
• int offset: offset value (Range of -100 to 100)

[Returns]
If the method succeeds, the return value is zero (0). Other cases, returns non-zero.

8-2-34 Set Cutting Position

Sets the cutting position or tear-off position after printing labels.

[Syntax]
Task<int> setCuttingPosition (int cuttingPosition)

[Parameters]
• int position: Cutting position value (Range of -100 to 100).

[Returns]
If the method succeeds, the return value is zero (0). Other cases, returns non-zero.

8-2-35 Set Auto Cutter

Sets whether to use auto cutter.



This method only works if an auto cutter is equipped in the printer.

[Syntax]
long setAutoCutter (bool enable, int period)

[Parameters]
• bool enable: Determines whether or not to use auto cutter
• int period: Configures the cutting cycle. When configured to 2, it will cut every second paper.

[Returns]
If the method succeeds, the return value is zero (0). Other cases, returns non-zero.

8-2-36 Get Model Name

Gets the printer's model name.

[Syntax]
Task<int> getModelName (Action<string> modelDelegate)

[Parameters]
• Action<string> modelDelegate: A delegate with a parameter used for getting a model name.

[Returns]
If the method succeeds, the return value is zero (0). Other cases, returns non-zero.

8-2-37 Get Firmware Version

Gets the printer’s firmware version.

[Syntax]
Task<int> getFirmwareVersion (Action<string> versionDelegate)

[Parameters]
• Action<string> versionDelegate: A delegate with a parameter used for getting a firmware version.

[Returns]
If the method succeeds, the return value is zero (0). Other cases, returns non-zero.

8-2-38 Get Printer Information

Gets the printer's mechanical information

[Syntax]
Task<int> getPrinterInformation (int item, Action<string> infoDelegate)

[Parameters]
• int item: Mechanical information to request to the printer

MPosLabelMechanicalInformation Enumeration	Value	Description	Unit
MPOS_LABEL_TPH_TEMPERATURE	0	TPH temperature	°C
MPOS_LABEL_PRINTING_DENSITY	1	Print density	-
MPOS_LABEL_TEAR_OFF_CUTTER_POSITION	2	Tear off/Cutter position	dot
MPOS_LABEL_MOTOR_TEMPERATURE	3	Motor temperature	°C
MPOS_LABEL_BOARD_TEMPERATURE	4	Board temperature	°C
MPOS_LABEL_BATTERY_VOLTAG	5	Battery voltage(%)	V(%)

• Action<string> infoDelegate: A delegate with a parameter used for getting a mechanical information

[Returns]
If the method succeeds, the return value is zero (0). Other cases, returns non-zero.

8-2-39 Setup RFID

Set the RFID transponder type, the number of retry codings, the number of retry labels, and the signal strength.

[Syntax]
Task<int> setupRFID (int rfidType, int numberOfRetries, int numberofLabel, int radioPower)

[Parameters]
• int rfidType: RFID transponder Type

MPosLabelRfidTransponderType Enumeration	Value	Description
MPOS_LABEL_RFID_TRANSPONDER_TYPE_NONE	0	none
MPOS_LABEL_RFID_TRANSPONDER_TYPE_18000_6_TYPE_A	1	ISO 18000-6 Type A
MPOS_LABEL_RFID_TRANSPONDER_TYPE_18000_6_TYPE_B	2	ISO 18000-6 Type B
MPOS_LABEL_RFID_TRANSPONDER_TYPE_CLASS_0	3	EPC Class 0
MPOS_LABEL_RFID_TRANSPONDER_TYPE_CLASS_1	4	EPC Class 1
MPOS_LABEL_RFID_TRANSPONDER_TYPE_CLASS_1_GEN2	5	EPC Class 1 Generation 2

• int numberOfRetries: Number of retries when coding fails (Range: 1 ~ 10)
• int numberOfLabel: Number of retry labels when coding fails (Range: 1 ~ 5)
• int radioPower: Signal strength (Range: 0 ~ 30)

[Returns]
If the method succeeds, the return value is zero (0). Other cases, returns non-zero.

8-2-40 Set RFID Position

Set the coding position directly on the RFID label.

[Syntax]
Task<int> setRFIDPosition (int transPosition)

[Parameters]
• int transPosition: Coding position of RFID label (y-axis value)

[Returns]
If the method succeeds, the return value is zero (0). Other cases, returns non-zero.

8-2-41 Set RFID Password

Set RFID Access Password and Kill Password.

[Syntax]
Task<int> setRFIDPassword (string oldAccessPwd, string oldKillPwd, string newAccessPwd, string newKillPwd)

[Parameters]
• string oldAccessPwd: Access Password currently in use
• string oldKillPwd: Kill Password currently in use
• string newAccessPwd: Access Password to be modified
• string newKillPwd: Kill Password to modify

[Returns]
If the method succeeds, the return value is zero (0). Other cases, returns non-zero.

8-2-42 Set EPC Data Structure

Define the EPC data structure for writing EPC data.

[Syntax]
Task<int> setEPCDataStructure (int totalSize, string fieldSizes)

[Parameters]
• int totalSize: Total bits in the field
• string fieldSizes: Bits of each field (Comma (',') is used for the separator)

[Returns]
If the method succeeds, the return value is zero (0). Other cases, returns non-zero.

8-2-43 Write RFID

Write RFID label.

[Syntax]
Task<int> writeRFID(int dataType, int startingBlockNumber, int dataLength, string data)

[Parameters]
• int numberOfRetries: data Type

MPosLabelRfidDataType Enumeration	Value	Description
MPOS_LABEL_RFID_DATA_TYPE_ASCII	'A'	ASCII
MPOS_LABEL_RFID_DATA_TYPE_HEX	'H'	HEX
MPOS_LABEL_RFID_DATA_TYPE_EPC	'E'	EPC
MPOS_LABEL_RFID_DATA_TYPE_USER_HEX	'U'	User area hex



must be used after setEPCDataStructure.

• int startingBlockNumber: Block to start writing (Range: 4 ~ 10)
• int dataLength: Number of bytes to write
• int data: Data (Input according to dataType)

[Returns]
If the method succeeds, the return value is zero (0). Other cases, returns non-zero.

8-2-44 Lock RFID

Lock Kill / Access / EPC Data using Access Password.



Be sure to use it after running setRFIDPassword.

[Syntax]
Task<int> lockRFID()

[Returns]
If the method succeeds, the return value is zero (0). Other cases, returns non-zero.

8-2-45 Draw PDF File

Saves data of the PDF file specified by the path to the printer buffer.

[Syntax]
Task<int> drawPDFFile (string filename, int startPosX, int startPosY, int width, int pageNumber, int threshold, int ditheringType, int compressType)

[Parameters]
• string filePath: The path of the PDF file.
• int startPosX: x axis coordinates of top left of image to be printed
• int startPosY: y axis coordinates of top left of image to be printed
• int width: Image width to be printed or original width ot the image.

MPosImageWidth Enumeration	Value	Description
MPOS_IMAGE_WIDTH_ASIS	-2	Print as original image size

• int pageNumber: Page number to print.



The first page start with zero (0)

• int threshold: Threshold (0 ≤ threshold ≤ 255)
• int ditheringType: dithering type (0: No dithering, 1: dithering)
• int compressType: Compress type

MPosCompressType Enumeration	Value	Description
MPOS_IMAGE_COMPRESS_NONE	0	No compress
MPOS_IMAGE_COMPRESS_RLE	1	Compress: RLE
MPOS_IMAGE_COMPRESS_LZMA	2	Compress: LZMA

[Returns]
If the method succeeds, the return value is zero (0). Other cases, returns non-zero.

9. MPos Controller BCD Reference

9-1 Over view

• The MPosControllerBCD interface provides methods for controlling display devices connected to the printer.



Only works on printers that support B-gate or USB peripheral control commands.

9-2 Method

9-2-1 Display String

Displays the string on the display device.

[Syntax]

int displayString(string data, int codePage, int internationalCharacterSet, int line);

[Parameters]

• string data: Character string to be printed

• int codePage: Code page to use when encoding a character string.

MPosLabelCodePage Enumeration	Value	Description
MPOS_CODEPAGE_CP437	437	USA, Standard Europe (PC437)
MPOS_CODEPAGE_CP850	850	Western European (PC850)
MPOS_CODEPAGE_CP852	852	Latin 2 (PC852)
MPOS_CODEPAGE_CP860	860	Portuguese (PC860)
MPOS_CODEPAGE_CP863	863	Canadian-French (PC863)
MPOS_CODEPAGE_CP865	865	Nordic (PC865)
MPOS_CODEPAGE_WPC1252	1252	Latin 1 (WPC1252)
MPOS_CODEPAGE_CP857	857	Turkce (PC857)
MPOS_CODEPAGE_CP737	737	Greek (PC 737)
MPOS_CODEPAGE_WPC1250	1250	Central European (PC1250)
MPOS_CODEPAGE_WPC1253	1253	Greek (WPC1253)
MPOS_CODEPAGE_WPC1254	1254	Turkce (WPC1254)
MPOS_CODEPAGE_CP855	855	Cyrillic (PC855)
MPOS_CODEPAGE_CP862	862	Hebrew DOS code (PC862)
MPOS_CODEPAGE_CP866	866	Cyrillic #2 (PC866)
MPOS_CODEPAGE_WPC1251	1251	Cyrillic (WPC1251)
MPOS_CODEPAGE_WPC1255	1255	Hebrew New code (WPC1255)
MPOS_CODEPAGE_CP928	928	Greek (PC928)
MPOS_CODEPAGE_CP775	775	Baltic (PC775)

MPOS_CODEPAGE_WPC1257	1257	Baltic (WPC1257)
MPOS_CODEPAGE_CP858	858	Euro (PC858)
MPOS_CODEPAGE_CHINA_GB2312	936	Simplified Chinese (GB2312)
MPOS_CODEPAGE_CHINA_BIG5	950	Traditional Chinese (BIG5)
MPOS_CODEPAGE_KOREA_KS5601	949	Korean (KS5601)
MPOS_CODEPAGE_JAPAN_SHIFTJIS	932	Japanese (SHIFT JIS)

• int internationalCharset: International character set

MPosICS Enumeration	Value	Description
MPOS_ICS_USA	0	USA Code
MPOS_ICS_FRANCE	1	FRANCE Code
MPOS_ICS_GERMANY	2	GERMANY Code
MPOS_ICS_UK	3	UK Code
MPOS_ICS_DENMARK_I	4	DENMARK1 Code
MPOS_ICS_SWEDEN	5	SWEDEN Code
MPOS_ICS_ITALY	6	ITALY Code
MPOS_ICS_SPAIN_I	7	SPAIN1 Code
MPOS_ICS_JAPAN	8	JAPAN Code
MPOS_ICS_NORWAY	9	NORWAY Code
MPOS_ICS_DENMARK_II	10	DENMARK 2 Code
MPOS_ICS_SPAIN_II	11	SPAIN 2 Code
MPOS_ICS_LATIN_AMERICA	12	LATIN AMERICA Code
MPOS_ICS_KOREA	13	KOREA Code
MPOS_ICS_SLOVENIA_CROATIA	14	SLOVENIA Code
MPOS_ICS_CHINA	15	CHINA Code

• int line: line number to display

1 ≤ line ≤ 4 (BCD-2000)

1 ≤ line ≤ 2 (BCD-3000)

[Returns]

If the method succeeds, the return value is zero (0). Other cases, returns non-zero.

9-2-2 Clean Screen

Deletes everything on the display device.

[Syntax]
int cleanScreen();

[Returns]
If the method succeeds, the return value is zero (0). Other cases, returns non-zero.

9-2-3 Store Image

Downloads images to the display device.

[Syntax]
int storeImage(ImageData data, int width, int imageNumbe);

[Parameters]
• ImageData data: A platform dependent image object.

iOS	UIKit.UIImage
Android	Android.Graphics.Bitmap
Windows	Windows.UI.Xaml.Media.Imaging.WriteableBitmap Windows.UI.Xaml.Media.Imaging.BitmapImage (deprecated)

• int width: image width
-1 or 1 ≤ width ≤ 240 (BCD-2000)
-1 or 1 ≤ width ≤ 160 (BCD-3000)
• int imageNumber: index for saving images
1 ≤ imageNumber ≤ 5

[Returns]
If the method succeeds, the return value is zero (0). Other cases, returns non-zero.

9-2-4 Store Image File

Downloads images to the display device.

[Syntax]
int storeImageFile(string fileName, int width, int imageNumbe);

[Parameters]
• string path: The path of the image file.



In Android, the image file should be in the SD card.

• int width: image width
-1 or 1 ≤ width ≤ 240 (BCD-2000)
-1 or 1 ≤ width ≤ 160 (BCD-3000)
• int imageNumber: index for saving images
1 ≤ imageNumber ≤ 5

[Returns]
If the method succeeds, the return value is zero (0). Other cases, returns non-zero.

9-2-5 Display Image

Displays images on the display device.

[Syntax]
int displayImage (int imageNumber, int xPos, int yPos);

[Parameters]
• int imageNumber: index of images to be displayed
1 ≤ imageNumber ≤ 5
• int xPos: display X-coordinates
0 ≤ xPos < 240 (BCD-2000)
0 ≤ xPos < 160 (BCD-3000)
• int yPos: display Y-coordinates
0 ≤ yPos < 64 (BCD-2000)
0 ≤ yPos < 32 (BCD-3000)

[Returns]
If the method succeeds, the return value is zero (0). Other cases, returns non-zero.

9-2-6 Clear Image

Deletes the images downloaded to the display device.

[Syntax]
int clearImage (int imageNumbe);

[Parameters]
• int imageNumber: index of images to be displayed
1 ≤ imageNumber ≤ 5

[Returns]
If the method succeeds, the return value is zero (0). Other cases, returns non-zero.

9-2-7 Show Cursor

Shows or hides the cursor on the display device.

[Syntax]
int showCursor (int cursor);

[Parameters]
• int cursor: show/hide cursor

MPosCursorType Enumeration	Value	Description
MPOS_CURSOR_DISABLE	0	Hides the cursor
MPOS_CURSOR_ENABLE	1	Shows the cursor

[Returns]
If the method succeeds, the return value is zero (0). Other cases, returns non-zero.

10. MPos Controller DallasKey Reference

10-1 Over view

• MPosControllerDallasKey interface is a class for controlling the Dallas Key connected to the printer. MPosControllerDevices DataEvent can be inherited and used.



Only works on printers that support B-gate or USB peripheral control commands.

11. MPos Controller HID Reference

11-1 Over view

• MPosControllerHID interface is a class for controlling the HID connected to the printer. MPosControllerDevices DataEvent can be inherited and used.



Only works on printers that support B-gate or USB peripheral control commands.

12. MPos Controller MSR Reference

12-1 Over view

• MPosControllerMSR interface is a class for controlling the MSR device connected to the printer. MPosControllerDevices DataEvent can be inherited and used.



Only works on printers that support B-gate or USB peripheral control commands.

13. MPos Controller NFC Reference

13-1 Over view

• MPosControllerNFC interface is a class for controlling the NFC connected to the printer. MPosControllerDevices DataEvent can be inherited and used.



Only works on printers that support B-gate or USB peripheral control commands.

14. MPos Controller RFID Reference

14-1 Over view

• MPosControllerRFID interface is a class for controlling the RFID connected to the printer. MPosControllerDevices DataEvent can be inherited and used.



Only works on printers that support B-gate or USB peripheral control commands.

15. MPos Controller Scale Reference

15-1 Over view

• MPosControllerScale interface is a class for controlling the scale connected to the printer. MPosControllerDevices DataEvent can be inherited and used.



Only works on printers that support B-gate or USB peripheral control commands.

16. MPos Controller Scanner Reference

16-1 Over view

• MPosControllerScanner interface is a class for controlling the barcode scanner connected to the printer. MPosControllerDevices DataEvent can be inherited and used.



Only works on printers that support B-gate or USB peripheral control commands.

17. MPos Controller USB Serial Reference

17-1 Over view

• MPosControllerUSBSerial interface is a class for controlling the USB to Serial device connected to the printer. MPosControllerDevices DataEvent can be inherited and used.



Only works on printers that support B-gate or USB peripheral control commands.

18. MPos Controller Config Reference

18-1 Over view

• MPosControllerConfig interface provides a method of controlling peripherals connected to the printer.



Only works on printers that support B-gate or USB peripheral control commands.

18-2 Methods

18-2-1 Search Devices

Searches for USB-type devices connected to the printer and returns the ID of the device found.

[Syntax]
int[] searchDevices();

[Returns]
Returns the ID of the device found.

18-2-2 Get BGate Serial Number

Returns the serial number of the B-gate printer.

[Syntax]
string getBGateSerialNumber();

[Returns]
Returns the B-gate's serial number.

18-2-3 Get USB Device

Returns the PID/VID of USB devices connected to the printer.

[Syntax]
string getUSBDevice(int deviceId);

[Returns]
• int deviceId: USB device ID

[Returns]
Returns the PID and VID of the connected USB device

18-2-4 Get Custom Devices

Returns information about user-defined USB devices connected to the printer.

[Syntax]
string getCustomDevices(int deviceType);

[Parameters]
• int deviceType: USB Device Type

MPosDeviceType Enumeration	Value	Description
MPOS_DEVICE_LABEL_PRINTER	1	POS Printer
MPOS_DEVICE_PRINTER	10	Label Pritner
MPOS_DEVICE_MSR	30	MSR
MPOS_DEVICE_SCANNER	40	Barcord Scanner
MPOS_DEVICE_RFID	60	RFID
MPOS_DEVICE_DALLASKEY	70	Dallas Key Reader
MPOS_DEVICE_NFC	80	NFC
MPOS_DEVICE_DISPLAY	110	Customer Display
MPOS_DEVICE_USB_TO_SERIAL	120	USB to Serial
MPOS_DEVICE_SCALE	130	Scale

[Returns]
Returns the VID/PID of all supported USB devices.

18-2-5 Get Serial Configuration

Returns the device's serial number.

[Syntax]
Int[] getSerialConfiguration(int deviceId);

[Parameters]
• int deviceId: USB device ID

device ID	Description
1 ≤ ID ≤ 9	POS Printer
11 ≤ ID ≤ 19	Label Pritner
20 ≤ ID ≤ 29	Unregistered
30 ≤ ID ≤ 39	MSR
40 ≤ ID ≤ 49	Barcord Scanner
60 ≤ ID ≤ 69	RFID
70 ≤ ID ≤ 79	Dallas Key Reader
80 ≤ ID ≤ 89	NFC
100 ≤ ID ≤ 109	Unregistered USB UART Device
110 ≤ ID ≤ 119	Customer Display
120 ≤ ID ≤ 129	USB to Serial
130 ≤ ID ≤ 139	Scale

[Returns]
Returns the device's baud rate, data bits, stop bits, parity bits, and flow control.

18-2-6 Set Serial Configuration

Change the serial information for that device.

[Syntax]
int setSerialConfiguration(int deviceId, int baudRate, int dataBit, int stopBit, int parityBit, int flowControl);

[Parameters]
• int deviceId: USB Device ID
• int baudRate: Baud Rate

MPosBaudRateType Enumeration	Value	Description
MPOS_BAUDRATE_2400	0	2400 bps
MPOS_BAUDRATE_4800	1	4800 bps
MPOS_BAUDRATE_9600	2	9600 bps
MPOS_BAUDRATE_19200	3	19200 bps
MPOS_BAUDRATE_38400	4	38400 bps
MPOS_BAUDRATE_57600	5	57600 bps
MPOS_BAUDRATE_115200	6	115200 bps
MPOS_BAUDRATE_230400	7	230400 bps

• int dataBit: Data bit

MPosDataBitType Enumeration	Value	Description
MPOS_DATABIT_7	0	7 bits
MPOS_DATABIT_8	1	8 bits

• int stopBit: Stop bit

MPosStopBitType Enumeration	Value	Description
MPOS_STOPBIT_1	0	1 bits
MPOS_STOPBIT_2	1	2 bits

• int parityBit: Parity bit

MPosParityBitType Enumeration	Value	Description
MPOS_PARITYBIT_NONE	0	None
MPOS_PARITYBIT_EVEN	1	Even
MPOS_PARITYBIT_ODD	2	Odd

• int flowControl: Flow control

MPosFlowControlType Enumeration	Value	Description
MPOS_FLOWCONTROL_NONE	0	None
MPOS_FLOWCONTROL_HW	1	H/W

[Returns]
If the method succeeds, the return value is zero (0). Other cases, returns non-zero.

18-2-7 Add Custom Device

Adds USB devices.
If a USB device is not recognized as a predefined type, make sure to define it as such before using it.

[Syntax]
int addCustomDevice (int deviceType, string vid, string pid);

[Parameters]
• int deviceType: Device Type

MPosDeviceType Enumeration	Value	Description
MPOS_DEVICE_LABEL_PRINTER	1	POS Printer
MPOS_DEVICE_PRINTER	10	Label Pritner
MPOS_DEVICE_MSR	30	MSR
MPOS_DEVICE_SCANNER	40	Barcord Scanner
MPOS_DEVICE_RFID	60	RFID
MPOS_DEVICE_DALLASKEY	70	Dallas Key Reader
MPOS_DEVICE_NFC	80	NFC
MPOS_DEVICE_DISPLAY	110	Customer Display
MPOS_DEVICE_USB_TO_SERIAL	120	USB to Serial
MPOS_DEVICE_SCALE	130	Scales

• string vid: String to USB Vender ID
• string pid: String to USB Product ID

[Returns]
If the method succeeds, the return value is zero (0). Other cases, returns non-zero.

18-2-8 Delete Custom Device

Deletes custom devices on the printer that can communicate with USB peripherals.

[Syntax]
int deleteCustomDevice(int deviceType, string vid, string pid);

[Parameters]
• int deviceType: Device Type

MPosDeviceType Enumeration	Value	Description
MPOS_DEVICE_LABEL_PRINTER	1	POS Printer
MPOS_DEVICE_PRINTER	10	Label Pritner
MPOS_DEVICE_MSR	30	MSR
MPOS_DEVICE_SCANNER	40	Barcord Scanner
MPOS_DEVICE_RFID	60	RFID
MPOS_DEVICE_DALLASKEY	70	Dallas Key Reader
MPOS_DEVICE_NFC	80	NFC
MPOS_DEVICE_DISPLAY	110	Customer Display
MPOS_DEVICE_USB_TO_SERIAL	120	USB to Serial
MPOS_DEVICE_SCALE	130	Scales

• string vid: String to USB Vender ID
• string pid: String to USB Product ID

[Returns]
If the method succeeds, the return value is zero (0). Other cases, returns non-zero.

18-2-9 ReInit Custom DeviceType

Initializes custom devices saved on the printer that can communicate with USB peripherals.

[Syntax]
int reInitCustomDeviceType (int deviceType);

[Parameters]
• int deviceType: Device Type

MPosDeviceType Enumeration	Value	Description
MPOS_DEVICE_LABEL_PRINTER	1	POS Printer
MPOS_DEVICE_PRINTER	10	Label Pritner
MPOS_DEVICE_MSR	30	MSR
MPOS_DEVICE_SCANNER	40	Barcord Scanner
MPOS_DEVICE_RFID	60	RFID
MPOS_DEVICE_DALLASKEY	70	Dallas Key Reader
MPOS_DEVICE_NFC	80	NFC
MPOS_DEVICE_DISPLAY	110	Customer Display
MPOS_DEVICE_USB_TO_SERIAL	120	USB to Serial
MPOS_DEVICE_SCALE	130	Scales

[Returns]
If the method succeeds, the return value is zero (0). Other cases, returns non-zero.

18-2-10 Get Serial Number

Gets the serial number of USB devices connected to the printer.

[Syntax]
String getSerialNumber (int deviceID);

[Parameters]
• int deviceID: Device ID

[Returns]
The serial number of the device

19. Appendix

19-1 Error Code Table

• The error code table below summarizes the values returned when calling the API provided in this manual.

MPosResult Enumeration	Value	Description
MPOS_SUCCESS	0	Suceess of the method Operation
MPOS_ALREADY_OPEN	1	Already Open
MPOS_FAIL	1000	Failure of the method Operation
MPOS_FAIL_INVALID_INTERFACE	1001	Not Supported Communication Interface Type
MPOS_FAIL_NO_OPEN	1002	Tried to access the device which is not opened
MPOS_FAIL_NOT_SUPPORT	1003	Not supported API by the device
MPOS_FAIL_INVALID_PARAMETER	1004	Invalid parameter for the method
MPOS_FAIL_NO_RESPONSE	1005	No response from the device
MPOS_FAIL_NOT_CONNECT	1006	Failed to connect the device
MPOS_FAIL_NO_FILE	1008	No file in the the path specified
MPOS_FAIL_NOT_SUPPORT_CODEPAGE	1012	Not supported code page
MPOS_FAIL_NOT_SUPPORT_ICS	1013	Not supported code international code page
MPOS_PAGEMODE_ALREADY_IN	1014	Already entered in page mode.
MPOS_TRANSACTION_ALREADY_IN	1015	Already entered in transactoin mode.
MPOS_FAIL_NOT_SUPPORT_ESCSEQ	1017	Not supported escape sequence
MPOS_FAIL_NO_DEVICE_FOUND	5000	No device found
MPOS_FAIL_NOT_PERMISSION_ALLOWD	5001	Failed to get permission required
MPOS_FAIL_NOT_SUPPORTED_IMAGE_TYPE	5002	Not supported file extention
MPOS_FAIL_IMAGE_DECODE	5003	Failed to decode image in BASE64

Revision history

Ver.	Date	Description
1.00	06-09-2018	New
1.01	30-11-2018	'printPdfFile' and 'cutPaper' added
	04-03-2019	APIs for label printers are added.
1.02	18-06-2019	- Information of the supported iOS version chaged. - Information of the supported models updated - Type of option in 'drawBlock" API modified.
	01-08-2019	Brightness range for iOS changed.
	26-09-2019	Descriptions for 'printPdfFile' changed.
1.03	12-03-2020	- Information of the supported Windows version chaged. - Information of the supported models updated
1.04	17-09-2020	- Information of the supported models updated - APIs for label printers rfid are added
1.05	28-10-2020	Information of the supported models updated
1.06	02-03-2021	Information of the supported models updated
1.07	20-08-2021	- Information of the supported models updated - Descriptions for the network entitlement added
1.08	06-10-2021	Information of the supported modes updated
1.09	23-03-2022	- Information of the supported devices and platform version update. - Methods releated to printing PDF files added and updated.
1.10	25-04-2022	Information of the supported models updated.
1.11	22-06-2022	Information of the supported models updated.
1.12	05-07-2022	Remove setPDFLicenseKey method.
1.13	16-08-2022	Information of the supported models updated.
1.14	09-01-2023	'getPrinterInformation' and 'openDrawer' added
1.15	28-03-2023	Add Android support USB API
	18-04-2023	Add Android, iOS support BLE API
1.16	30-08-2023	Fix Android Minimum SDK Version Add BK3, BK5 series models
1.17	05-01-2024	Added USB Peripheral Class Added API in MPosControllerConfig
1.18	31-01-2024	Added XQ-840II, XQ-843II
1.19	08-08-2024	Added SRP-S300II
1.20	21-10-2024	Added XD5-40IId, XD5-43IId, XD5-40IIt, XD5-43IIt
1.21	26-12-2024	SRP-380II, SRP-382II model name updated to SRP-380plus, SRP-382plus respectively
1.22	17-03-2025	Added XT6-60, XT6-63, XM5-30
1.23	28-03-2025	Added SRP-G300
1.24	28-04-2025	- Information of supported printers updated

		: BT3-40/ BT-43 : BT5-40(RFID) / BT5-43(RFID) / BT5-46(RFID) : BD5-40d/ BD5-43d : BD5-40t(RFID)/ BD5-43t(RFID)
1.25	11-08-2025	Added XD7-20d, XD7-23d