BIXOLON

API Reference Guide

Cordova Plugin

Ver. 1.17

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 **7**
1-1 Supported Cordova Version 7
1-2 Supported Devices 8

2. Cordova Plugin Information **12**
2-1 Supported Platforms 12
2-2 Dependency plugin 12
2-3 Multicast Networking Entitlement Request 12

3. MposControllerConfig Reference **13**
3-1 Overview 13
3-2 Methods 13
3-2-1 Select Interface 13
3-2-2 Select Command Mode 14
3-2-3 Open Service 14
3-2-4 Close Service 15
3-2-5 Search Devices 15
3-2-6 Get B-gate Serial Number 16
3-2-7 Get Serial Number 17

4. MPosControllerLookup Reference **18**
4-1 Overview 18
4-2 Methods 18
4-2-1 Refresh Deivces List 18
4-2-2 Get Device List 19

5. MPosControllerPrinter Reference **20**
5-1 Overview 20
5-2 Methods 20
5-2-1 Select Interface 20
5-2-2 Select Command Mode 21
5-2-3 Open Service 22
5-2-4 Close Service 23
5-2-5 Direct IO 23
5-2-6 Is Open 24
5-2-7 Get Device ID 24
5-2-8 Set Transaction 25
5-2-9 Set Text Encoding 26
5-2-10 Print Text 28
5-2-11 Set Character set 30
5-2-12 Set International Character set 32
5-2-13 Set Page Mode 34
5-2-14 Set Page Mode Print Area 35

5-2-15 Set Page Mode Direction....35
5-2-16 Set Page Mode Position....36
5-2-17 Print Bitmap File....37
5-2-18 Print Bitmap with Base64....38
5-2-19 Print 1D Barcode....39
5-2-20 Print QR Code....40
5-2-21 Print PDF417....41
5-2-22 Print Data Matrix....42
5-2-23 Check Batt Status....43
5-2-24 Check Printer Status....44
5-2-25 Asb Enable....45
5-2-26 Set Status Update Event....46
5-2-27 Open Drawer....46
5-2-28 Print PDF File....47

6 MPosControllerLabelPrinter Reference....48
6-1 Overview....48
6-2 Methods....48
6-2-1 Select Interface....48
6-2-2 Select Command Mode....49
6-2-3 Open Service....50
6-2-4 Close Service....51
6-2-5 Direct IO....52
6-2-6 Is Open....53
6-2-7 Get Device ID....53
6-2-8 Set Transaction....54
6-2-9 Set Text Encoding....55
6-2-10 Set Character set....57
6-2-11 Set Printing Type....59
6-2-12 Set Margin....60
6-2-13 Set Back Feed Option....61
6-2-14 Set Length....62
6-2-15 Set Width....63
6-2-16 Set Buffer Mode....64
6-2-17 Set Speed....65
6-2-18 Set Density....65
6-2-19 Set Orientation....66
6-2-20 Set Offset....66
6-2-21 Set Cutting Position....67
6-2-22 Set Auto Cutter....68
6-2-23 Print Buffer....69
6-2-24 Check Printer Status....70
6-2-25 Draw Text Device Font....71
6-2-26 Draw Text Vector Font....73
6-2-27 Draw Circle....75
6-2-28 Draw Block....76
6-2-29 Draw Image File....77
6-2-30 Draw Image with Base64....78

6-2-31 Draw Barcode 1D....79
6-2-32 Draw Barcode Maxi Code....81
6-2-33 Draw Barcode PDF417....82
6-2-34 Draw Barcode QR Code....84
6-2-35 Draw Barcode Data Matrix....85
6-2-36 Draw Barcode Aztec....86
6-2-37 Draw Barcode Code49....87
6-2-38 Draw Barcode Coda Block....89
6-2-39 Draw Barcode Micro PDF....90
6-2-40 Draw Barcode IMB....91
6-2-41 Draw Barcode MSI....92
6-2-42 Draw Barcode Plessey....93
6-2-43 Draw Barcode TLC39....94
6-2-44 Draw Barcode RSS....95
6-2-45 Setup RFID....97
6-2-46 Set RFID Position....98
6-2-47 Set RFID Password....99
6-2-48 Set EPC Data Structure....100
6-2-49 Write RFID....101
6-2-50 Lock RFID....102
6-5-51 Draw PDF File....103

7. MPosControllerBCD Reference....104
7-1 Overview....104
7-2 Methods....104
7-2-1 Display String....104
7-2-2 Clean Screen....107
7-2-3 Store Image File....108
7-2-4 Display Image....109
7-2-5 Clear Image....110
7-2-6 Show Cursor....111

8. MPosControllerDallasKey Reference....112
8-1 Overview....112

9. MPosControllerHID Reference....113
9-1 Overview....113

10. MPosControllerMSR Reference....114
10-1 Overview....114

11. MPosControllerNFC Reference....115
11-1 Overview....115

12. MPosControllerRFID Reference....116
12-1 Overview....116

13. MPosControllerScale Reference....117

13-1 Overview.....117

14. MPosControllerScanner Reference..... 118
14-1 Overview.....118

15. MPosControllerUSBSerial Reference..... 119
15-1 Overview.....119

1. Manual Guide

This manual contains a description of the libraries required to develop applications using the Cordova Custom Plugin to control printers and peripherals.

1-1 Supported Cordova Version

• Development environment: Cordova 6.5.0 or later

- The installed version of Cordova can be checked using the Command Line Interface below.
  Cordova –version
- For more information on the Command Line Interface, please visit:
  https://cordova.apache.org/docs/en/3.1.0/guide/cli/index.html

1-2 Supported Devices

Receipt Printer	Resolution	Supported Communications	Max Printable Width(dot)
SPP-R200III	203 dpi	Bluetooth, Wi-Fi, USB	384 dots
SPP-R210	203 dpi		384 dots
SPP-R310	203 dpi		576 dots
SPP-R410	203 dpi		832 dots
SPP-C200	203 dpi	Bluetooth, USB	384 dots
SPP-C300	203 dpi	Bluetooth, USB	576 dots
SRP-350III	180 dpi	Ethernet, USB	512 dots
SRP-352III	203 dpi		576 dots
SRP-330II	180 dpi		512 dots
SRP-332II	203 dpi		576 dots
SRP-350plusIII	180 dpi	Bluetooth, Wi-Fi, Ethernet, USB	512 dots
SRP-352plusIII	203 dpi		576 dots
SRP-380	180 dpi		512 dots
SRP-380plus	180 dpi		512 dots
SRP-382	203 dpi		576 dots
SPR-382plus	203 dpi		576 dots
SRP-383	300 dpi		864 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-S3000	203 dpi		576 dots
BGT-100P	180 dpi		512 dots
BGT-102P	203 dpi		576 dots
SRP-Q300H	180 dpi		512 dots
SRP-Q302H	203 dpi		576 dots
SRP-350plusV	180 dpi		512 dots
SRP-352plusV	203 dpi		576 dots
SRP-S200	203 dpi	Ethernet, USB	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-350V	180 dpi		512 dots
SRP-352V	203 dpi		576 dots
SRP-330III	180 dpi		512 dots
SRP-332III	203 dpi		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

Label Printer	Resolution	Supported Communications	Max Printable Width(dot)
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
SLP-DL410	203 dpi		864 dots
SLP-DL413	300 dpi		1248 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		384 dots
XM7-30	203 dpi		576 dots
XM7-40(RFID)	203 dpi		832 dots
SRP-770III	203 dpi	Bluetooth, Wi-Fi, Ethernet, USB	832 dots
SRP-E770III	203 dpi	Ethernet, USB	832 dots
XT5-40(RFID)	203 dpi	Bluetooth, Wi-Fi, Ethernet, USB	832 dots
XT5-43(RFID)	300 dpi		1248 dots
XT5-46(RFID)	600 dpi		2496 dots
XD5-40d	203 dpi		864 dots
XD5-43d	300 dpi		1248 dots
XD5-40t(RFID)	203 dpi		864 dots
XD5-43t(RFID)	300 dpi		1248 dots
XL5-40	203 dpi		864 dots
XL5-43	300 dpi		1248 dots
SRP-S3000_LABEL	203 dpi		576 dots
XT3-40	203 dpi		864 dots
XT3-43	300 dpi		1248 dots
XQ-840II	203 dpi	USB	864 dots
XQ-843II	300 dpi	USB	1248 dots
XD5-40IId	203 dpi	Bluetooth, Wi-Fi, Ethernet, USB	864 dots
XD5-43IId	300 dpi		1248 dots
XD5-40IIt	203 dpi		864 dots
XD5-43IIt	300 dpi		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

2. Cordova Plugin Information

2-1 Supported Platforms

Platform	Description
Android	Android 4.0.3(API Level 15: ice cream sandwich) or later
iOS	iOS 12.0 or later
UWP	Windows 10 version Build 14393 or later

2-2 Dependency plugin

Plugin name	Description
cordova-promise-polyfill	This dependency plugin is a polyfill for using ES6 Promise objects in Cordova. There is code written using this plug-in in the sample programs provided. Install this plug-in if your development environment cannot use ES6 Promise objects.

2-3 Multicast Networking Entitlement Request

In iOS, the Bixolon Cordova Plugin has APIs for discovering printers in local network by sending broadcast IP packet. Since iOS 14.5 or later, the APIs does not work without Apple permission because the policy for local networks has been changed. See the link below.

1) https://developer.apple.com/news/?id=0oi77447
2) https://developer.apple.com/contact/request/networking-multicast

- IP address used for discovering devices: 255.255.255.255
- Port numbers used for discovering devices: 48780, 48781, 9000, 3337

3. MposControllerConfig Reference

3-1 Overview

• The “MPosControllerConfig” object provides methods for registering or removing additional peripherals to the B-gate device or for setting up communication of registered peripherals.

3-2 Methods

3-2-1 Select Interface

Set the communication interface type and address of the device to be connected.

[Parameters]

Parameters	Data Type	Description
successCallback	Function	Callback function that is called when succeeded.
errorCallback	Function	Callback function that is called when failed.
interfaceType	Number	Communication Interface **1:** Wi-Fi **2:** Ethernet **4:** Bluetooth **16:** USB
address	String	IP Address: portNumber, MAC address, Bluetooth S/N Example) 192.168.1.2:9100 Example) 74:F0:7D:E3:DF:EE



USB communication is only available on Android.

3-2-2 Select Command Mode

Selects whether to communicate directly with the device or through the “B-gate”.

[Parameters]

Parameters	Data Type	Description
successCallback	Function	Callback function that is called when succeeded.
errorCallback	Function	Callback function that is called when failed.
commandMode	Number	0 (Fixed)

3-2-3 Open Service

Initiates a communications connection with the device.

[Parameters]

Parameters	Data Type	Description
successCallback	Function	Callback function that is called when succeeded.
errorCallback	Function	Callback function that is called when failed.
id	Number	0 (Fixed)
timeout	Number	Time out in second for the connection with host.

[Example]

```
return new Promise(function (returnResolve, returnReject) {
    new Promise(function (resolve, reject) { _Config.selectInterface(resolve, reject, _ConnectionInfo.interfaceType,0);})
     .then(function () { return new Promise(function (resolve, reject) { _Config.selectCommandMode(resolve, reject,
0);});});})
     .then(function () { return new Promise(function (resolve, reject) {
          _Config.openService(resolve, function(error){
                    returnReject();
               }, $('#deviceID').val(), _ConnectionInfo.timeout);
          });
     })
     .then(Function{
          returnResolve();
     });
});
```

3-2-4 Close Service

Terminates the communication connection with the device.

[Parameters]

Parameters	Data Type	Description
successCallback	Function	Callback function that is called when succeeded.
errorCallback	Function	Callback function that is called when failed.
timeout	Number	Time out in second for the disconnection with host.

3-2-5 Search Devices

Returns a list of device identifier values connected with the B-gate.

[Parameters]

Parameters	Data Type	Description
successCallback	Function	Callback function that is called when succeeded.
errorCallback	Function	Callback function that is called when failed.
interfaceType	Number	Communication interface of the target devices **1:** Wi-Fi **2:** Ethernet **4:** Bluetooth

[Example]

```
_Config.searchDevices(function(list){
      for(var i = 0; i < list.length; i++) {
           ...
      }
}, function(error){

});
```

3-2-6 Get B-gate Serial Number

Gets the B-gate product serial number.

[Parameters]

Parameters	Data Type	Description
successCallback	Function	Callback function to receive product serial number value.
errorCallback	Function	Callback function that is called when failed.

[Example]

```
_Config.getBgateSerialNumber(
  function(bGateSN){ console.log("getBgateSerialNumber success"); },
  function{ console.log("getBgateSerialNumber failed"); }
)
```

3-2-7 Get Serial Number

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

[Parameters]

Parameters	Data Type	Description
successCallback	Function	Callback function to receive product serial number value.
errorCallback	Function	Callback function that is called when failed.
deviceID	Number	ID of the device to be connected

[Example]

```
_Config.getSerialNumber(
  function{ console.log("getSerialNumber success"); },
  function{ console.log("getBgateSerialNumber failed");
  10/*POS Printer*/ }
)
```

4. MPosControllerLookup Reference

4-1 Overview

• “MPosControllerLookup” provides methods related to device list information that can be connected.

4-2 Methods

4-2-1 Refresh Deivces List

Update the list of available networks (Wi-Fi, Ethernet) and Bluetooth devices.

[Parameters]

Parameters	Data Type	Description
successCallback	Function	Callback function that is called when succeeded.
errorCallback	Function	Callback function that is called when failed.
interfaceType	Number	Communication type **1:** Wi-Fi **2:** Ethernet **4:** Bluetooth
timeout	Number	Timeout in second for updating the list of devices that can be connected.

[Example]

```
_Lookup.refreshDeivcesList(
  function{ console.log("refreshDeivcesList success"); },
  function{ console.log("refreshDeivcesList failed"); },
  4 /*Bluetooth*/, 2 /*Second*/
)
```

4-2-2 Get Device List

Gets a list of connectable networks (Wi-Fi, Ethernet) and Bluetooth devices in JSON (JavaScript Object Notation) format.

Before calling this method, the 'refreshDevicesList' method should be called first.

[Parameters]

Parameters	Data Type	Description
successCallback	Function	Callback function to receive the list of connectable devices.
errorCallback	Function	Callback function that is called when failed.
interfaceType	Number	Communication type **1:** Wi-Fi **2:** Ethernet **4:** Bluetooth

[Example]

```
_Lookup.getDeviceList(
  function(list){
    console.log("getDeviceList success");
    var jsonObjArray = JSON.parse(list);
    for(var i = 0; i < jsonObjArray.length; i++){
      // jsonObjArray[i].interfaceType
      // jsonObjArray[i].name
      // jsonObjArray[i].address
    }
  },
  function{ console.log("getDeviceList failed"); },
  4 /*Bluetooth*/,
)
```

5. MPosControllerPrinter Reference

5-1 Overview

• The “MPosControllerPrinter” object provides methods for controlling receipt printers.

5-2 Methods

5-2-1 Select Interface

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

[Parameters]

Parameters	Data Type	Description
successCallback	Function	Callback function that is called when succeeded.
errorCallback	Function	Callback function that is called when failed.
interfaceType	Number	Communication Interface **1:** Wi-Fi **2:** Ethernet **4:** Bluetooth **16:** USB
address	String	IP Address:portNumber, MAC address, Bluetooth S/N Example) 192.168.1.2:9100 Example) 74:F0:7D:E3:DF:EE



USB communication is only available on Android.

5-2-2 Select Command Mode

Selects whether to communicate directly with the device or through the “B-gate”.

[Parameters]

Parameters	Data Type	Description
successCallback	Function	Callback function that is called when succeeded.
errorCallback	Function	Callback function that is called when failed.
commandMode	Number	Communication method between HOST and a device. **0:** via B-gate **1:** direct communication

5-2-3 Open Service

Initiates a communications connection with the device.

[Parameters]

Parameters	Data Type	Description
successCallback	Function	Callback function that is called when succeeded.
errorCallback	Function	Callback function that is called when failed.
id	Number	Identifier of the device to connect.
timeout	Number	Time out in second for the connection with host.

- Using “searchDevices” described in 3-2-5 of this manual, you can get A list of connectable device identifiers (IDs).
- If not using B-gate, set the device identifier (id) to -1.
  * Receipt printer identifier range: 10 ~ 19

[Example]

```
return new Promise(function (returnResolve, returnReject) {
    new Promise(function (resolve, reject) { _Printer.selectInterface(resolve, reject, _ConnectionInfo.interfaceType,0);})
     .then(function () { return new Promise(function (resolve, reject) { _Printer.selectCommandMode(resolve, reject, 0);});})
     .then(function () { return new Promise(function (resolve, reject) {
          _Printer.openService(resolve, function(error){
                    returnReject();
               }, $('#deviceID').val(), _ConnectionInfo.timeout);
          });
     })
     .then(Function{
          returnResolve();
     });
});
```

5-2-4 Close Service

Terminates the communication connection with the device.



Calling this method during data transfer can cause the communication connection to close without completing the data transfer. Call this method by setting the appropriate timeout value according to the print speed and print length.

[Parameters]

Parameters	Data Type	Description
successCallback	Function	Callback function that is called when succeeded.
errorCallback	Function	Callback function that is called when failed.
timeout	Number	Time out in second for the disconnection with host.

5-2-5 Direct IO

Sends custom data to the device.

[Parameters]

Parameters	Data Type	Description
successCallback	Function	Callback function that is called when succeeded.
errorCallback	Function	Callback function that is called when failed.
data	Object	User defined byte array.

[Example]

```
_Printer.directIO(function(result){
   ...
},function(result){
   ...
}, [0x1D, 0x56, 0x01] /* Cut */);
```

5-2-6 Is Open

Returns whether the device is open for communication.

[Parameters]

Parameters	Data Type	Description
successCallback	Function	Callback function that receives the information of that the device is open or not.
errorCallback	Function	Callback function that is called when failed.

[Example]

```
_Printer.isOpen(
  function(open){
    console.log("isOpen success");
    if(open) { ... }
    else { ... }
  },
  function{ console.log("isOpen failed"); }
)
```

5-2-7 Get Device ID

Gets the device identifier value.

[Parameters]

Parameters	Data Type	Description
successCallback	Function	Callback function that receives the device identifier.
errorCallback	Function	Callback function that is called when failed.

[Example]

```
_Printer.getDeviceID(
  function(deviceId){ console.log("getDeviceID success"); },
  function { console.log("getDeviceID failed"); }
)
```

5-2-8 Set Transaction

Enter or exit transaction mode to transfer the data in the buffer to the device (printer).

[Parameters]

Parameters	Data Type	Description
successCallback	Function	Callback function that is called when succeeded.
errorCallback	Function	Callback function that is called when failed.
TransactionMode	Number	Transaction Mode **0:** Transaction mode out and send data to the printer **1:** Transaction mode in

[Example]

```
return new Promise(function (returnResolve, returnReject) {
    new Promise(function (resolve, reject) { _Printer.setTransaction(resolve, reject, 1);})
     /* add more operations as required */
     .then(Function { return new Promise(function (resolve, reject) { _Printer.setTransaction(resolve, reject, 0);});})
     .then(Function{
          returnResolve();
     });
});
```

5-2-9 Set Text Encoding

Sets the code page to use for character encoding.

The printer prints characters referring to the preset code page (character encoding table). If the set code page information is not changed, characters may not be output normally. Therefore, set the appropriate code page according to the characters.

[Parameters]

Parameters	Data Type	Description
successCallback	Function	Callback function that is called when succeeded.
errorCallback	Function	Callback function that is called when failed.
textEncoding	Number	Code page for encoding characters **437**: English (PC437) **850**: Western Europe (PC850) **860**: Portuguese (PC860) **863**: Canadian French (PC863) **865**: Norwegian (PC865) **1252**: Latin1 (WPC1252) **866**: Cyrillic (PC866) **852**: Latin2 (PC852) **858**: Eastern Europe (PC858) **862**: Hebrew (PC862) **34**: Thai (THAI 11) **31**: Thai (THAI 14) **39**: Thai (THAI 16) **35**: Thai (THAI 18) **23**: Thai (THAI 42) **1253**: Greek (WPC1253) **1254**: Turkish (WPC1254) **1257**: Baltic (WPC1257) **1251**: Cyrillic (WPC1251) **27**: Persian (FARSI) **28**: Persian (FARSI) **737**: Greek (PC737) **775**: Baltic (PC775) **1255**: Hebrew (WPC1255) **855**: Cyrillic (PC855) **857**: Turkish (PC857) **928**: Greek (PC928) **1250**: Czech (WPC1250) **28605**: Latin9 **1258**: Vietnam (WPC1258)

		49: Vietnam (TCVN3) **50**: Vietnam (TCVN3 Capital) **51**: Vietnam (VISCII) **864**: Arabic (PC864) **1256**: Arabic (WPC1256) **949**: Korean (KS5601) **932**: Japanese (ShiftJIS) **950**: Chinese (BIG5) **936**: Chinese (GB2312)

[Example]

```
_Printer.setTextEncoding(
  function{ console.log("setTextEncoding success"); },
  function{ console.log("setTextEncoding failed"); },
  437
)
```

5-2-10 Print Text

Prints characters by specifying various font attributes of the character.



Depending on the default code page value set for the printer, characters may be printed as "?" or not printed. If the characters you want to print do not print correctly, change the code page values using the "setTextEncoding" or "setCharacterset" methods in this manual.

[Parameters]

Parameters	Data Type	Description
successCallback	Function	Callback function that is called when succeeded.
errorCallback	Function	Callback function that is called when failed.
data	String	Characters
textPrintProperty	Object	Character properties

<TextPrintProperty Object>

```
var TextPrintProperty = function(codePage, fontType, width, height, underline,
alignment, bold, color, reverse){
      this.codePage = !!codePage ? codePage : 1252 /* WPC1252 */;
      this.fontType = !!fontType ? fontType : 0 /* Font A */;
      this.width = !!width ? width : 0;
      this.height = !!height ? height : 0;
      this.underline = !!underline ? underline : 0 /* No underline */;
      this.alignment = !!alignment ? alignment : 0 /* Left */;
      this.bold = !!bold ? bold : false;
      this.reverse = !!reverse ? reverse : false;
};
```

Character print properties

- **codePage**: Code page to use for character encoding.
- **fontType**: Font type (Font A, Font B, Font C)
- **width**: font width (0, 16, 32, 48, 64, 80, 96, 112)
- **height**: font height (0 ~7)
- **underline**: Underline (0: None, 1: Underline 1dot, 2: underline 2dot)
- **alignment**: Alignment (0: Left, 1: Center, 2: Right)
- **bold**: bold or not
- **reverse**: reverse or not

[Example]

```
_Printer.printText(
  function{ console.log("printText success"); },
  function{ console.log("printText failed"); },
  'Hello Cordova' + '\n', new TextPrintProperty(437, 1, 0, 0, 0,false, false, false)
)
```

5-2-11 Set Character set

Sets the code page to use for character encoding.

The printer prints characters by referring to a preset code page (character encoding table).
If the set code page is not changed, characters may not be printed properly, so be sure to set the appropriate code page according to the characters.

[Parameters]

Parameters	Data Type	Description
successCallback	Function	Callback function that is called when succeeded.
errorCallback	Function	Callback function that is called when failed.
codePage	Number	Code page for encoding characters **437**: English (PC437) **850**: Western Europe (PC850) **860**: Portuguese (PC860) **863**: Canadian French (PC863) **865**: Norwegian (PC865) **1252**: Latin1 (WPC1252) **866**: Cyrillic (PC866) **852**: Latin2 (PC852) **858**: Eastern Europe (PC858) **862**: Hebrew (PC862) **34**: Thai (THAI 11) **31**: Thai (THAI 14) **39**: Thai (THAI 16) **35**: Thai (THAI 18) **23**: Thai (THAI 42) **1253**: Greek (WPC1253) **1254**: Turkish (WPC1254) **1257**: Baltic (WPC1257) **1251**: Cyrillic (WPC1251) **27**: Persian (FARSI) **28**: Persian (FARSI) **737**: Greek (PC737) **775**: Baltic (PC775) **1255**: Hebrew (WPC1255) **855**: Cyrillic (PC855) **857**: Turkish (PC857) **928**: Greek (PC928) **1250**: Czech (WPC1250) **28605**: Latin9

		1258: Vietnam (WPC1258) **49**: Vietnam (TCVN3) **50**: Vietnam (TCVN3 Capital) **51**: Vietnam (VISCII) **864**: Arabic (PC864) **1256**: Arabic (WPC1256) **949**: Korean (KS5601) **932**: Japanese (ShiftJIS) **950**: Chinese (BIG5) **936**: Chinese (GB2312)

[Example]

```
_Printer.setCharacterset(
  function{ console.log("setCharacterset success"); },
  function{ console.log("setCharacterset failed"); },
  437 /*PC437*/
)
```

5-2-12 Set International Character set

Sets the code table to use for international character encoding.

The 14 characters represented by the hexadecimal values below are treated separately from the code page set by the '**setCharacterset**' or '**setTextEncoding**' method.

No	Country	ASCII code (Hexadecimal number)													
		23	24	25	26	40	5B	5C	5D	5E	60	7B	7C	7D	7E
0	U.S.A	#	$	%	&	@	[	\	]	^	`	{	|	}	~
1	France	#	$	%	&	à	°	ç	§	^	`	é	ù	è	¨
2	Germany	#	$	%	&	§	Ä	Ö	Ü	^	`	ä	ö	ü	β
3	U.K.	£	$	%	&	@	[	\	]	^	`	{	|	}	~
4	Denmark I	#	$	%	&	@	Æ	Ø	Å	^	`	æ	ø	å	~
5	Sweden	#	¤	%	&	É	Ä	Ö	Å	Ü	é	ä	ö	å	ü
6	Italy	#	$	%	&	@	°	\	é	^	ù	à	ò	è	ì
7	Spain I	Pt	$	%	&	@	¡	Ñ	¿	^	`	¨	ñ	}	~
8	Japan	#	$	%	&	@	[	¥	]	^	`	{	|	}	~
9	Norway	#	¤	%	&	É	Æ	Ø	Å	Ü	é	æ	ø	å	ü
10	Denmark II	#	$	%	&	É	Æ	Ø	Å	Ü	é	æ	ø	å	ü
11	Spain II	#	$	%	&	á	¡	Ñ	¿	é	`	í	ñ	ó	ú
12	Latin America	#	$	%	&	á	¡	Ñ	¿	é	ü	í	ñ	ó	ú
13	Korea	#	$	%	&	@	[	\	]	^		{	|	}	~

[Parameters]

Parameters	Data Type	Description
successCallback	Function	Callback function that is called when succeeded.
errorCallback	Function	Callback function that is called when failed.
characterset	Number	International character set **0:** USA **1:** FRANCE **2:** GERMANY **3:** UK **4:** DENMARK1 **5:** SWEDEN **6:** ITALY **7:** SPAIN **8:** JAPAN **9:** NORWAY **10:** DENMARK 2 **11:** SPAIN 2 **12:** LATIN AMERICA **13:** KOREA **14:** SLOVENIA

		15: CHINA

[Example]

```
_Printer.setInternationalCharacterset(
  function{ console.log("setInternationalCharacterset success"); },
  function{ console.log("setInternationalCharacterset failed"); },
  0 /* USA */
)
```

5-2-13 Set Page Mode

Enter or exit page mode to transfer data in the buffer to the device.

Page mode cannot be used together with transaction mode.

[Parameters]

Parameters	Data Type	Description
successCallback	Function	Callback function that is called when succeeded.
errorCallback	Function	Callback function that is called when failed.
pageMode	Number	Enter or leave page mode **1:** page mode in **0:** page mode out and send the data

[Example]

```
return new Promise(function (returnResolve, returnReject) {
    new Promise(function (resolve, reject) { _Printer. setPagemode(resolve, reject, 1);})
     .then(Function { return new Promise(function (resolve, reject) {
         _Printer.setPagemodePrintArea(resolve,reject,10, 10, 384, 2000);
     });})
     .then(Function { return new Promise(function (resolve, reject) {
         _Printer. setPagemodeDirection(resolve,reject, 0);
     });})
     .then(Function { return new Promise(function (resolve, reject) {
         _Printer. setPagemodePosition (resolve,reject, 100,100);
     });})
     .then(Function { return new Promise(function (resolve, reject) { _Printer. setPagemode(resolve, reject, 0);});});})
     .then(Function{
          returnResolve();
     });
});
```

5-2-14 Set Page Mode Print Area

Sets the printing area in page mode.

[Parameters]

Parameters	Data Type	Description
successCallback	Function	Callback function that is called when succeeded.
errorCallback	Function	Callback function that is called when failed.
x	Number	The starting X axis coordinate (dot).
y	Number	The starting Y axis coordinate (dot).
width	Number	Print width (dot)
height	Number	Print Height (dot)

5-2-15 Set Page Mode Direction

Sets the printing direction in pagemode.

[Parameters]

Parameters	Data Type	Description
successCallback	Function	Callback function that is called when succeeded.
errorCallback	Function	Callback function that is called when failed.
direction	Number	Print direction **0:** left to right (starting position: upper left) **1:** bottom to top (starting position: lower left) **2:** right to left (starting position: lower right) **3:** top to bottom (starting position: upper right)

5-2-16 Set Page Mode Position

Sets the coordinates of the printing target in page mode.

[Parameters]

Parameters	Data Type	Description
successCallback	Function	Callback function that is called when succeeded.
errorCallback	Function	Callback function that is called when failed.
x	Number	X coordinate, in dot unit, of the object (text, barcode, image) to be printed.
y	Number	Y coordinate, in dot unit, of the object (text, barcode, image) to be printed.

5-2-17 Print Bitmap File

Prints a bitmap file existing on the HOST.

- For Android, only files on the SD card can be printed.
- Depending on the version of the platform (Android, iOS, UWP), you may not be able to print if you do not have file access.

[Parameters]

Parameters	Data Type	Description
successCallback	Function	Callback function that is called when succeeded.
errorCallback	Function	Callback function that is called when failed.
path	String	The location of the image file.
width	Number	Print width **-2:** Print with the original size of the image
alignment	Number	Alignment **0:** Left **1:** Center **2:** Right
brightness	Number	Brightness Range: 0 ≤ brightness ≤ 100 (Android, UWP) Range: -100 ≤ brightness ≤ 100 (iOS)
isDithering	Boolean	Whether or not to apply dithering
isCompress	Boolean	Whether or not to compress data

[Example]

```
_Printer.printBitmapFile(
  function{ console.log("printBitmapFile success"); },
  function{ console.log("printBitmapFile failed"); },
  path, -2, 0, 50, true, true
)
```

5-2-18 Print Bitmap with Base64

Print a graphic image encoded in base64.

[Parameters]

Parameters	Data Type	Description
successCallback	Function	Callback function that is called when succeeded.
errorCallback	Function	Callback function that is called when failed.
imageSource	String	Image data encoded in Base64.
width	Number	Print width **-2:** Print with the original size of the image
alignment	Number	Alignment **0:** Left **1:** Center **2:** Right
brightness	Number	Brightness Range: 0 ≤ brightness ≤ 100 (Android, UWP) Range: -100 ≤ brightness ≤ 100 (iOS)
isDithering	Boolean	Whether or not to apply dithering
isCompress	Boolean	Whether or not to compress data

[Example]

```
_Printer.printBitmapWithBase64(
  function{ console.log("printBitmapWithBase64 success"); },
  function{ console.log("printBitmapWithBase64 failed"); },
  new ImgInBase64Format().getImageData(0),-2, 0, 50, true, true
)
```

5-2-19 Print 1D Barcode

Prints a 1-D barcode.

[Parameters]

Parameters	Data Type	Description
successCallback	Function	Callback function that is called when succeeded.
errorCallback	Function	Callback function that is called when failed.
data	String	Barcode Data
barcodeType	Number	Barcode Type **101:** UPC-A **102:** UPC-E **103:** JAN-8 (EAN) **104:** JAN-13 (EAN) **106:** ITF **107:** CODABAR **108:** CODE39 **109:** CODE93 **110:** CODE128
barWidth	Number	Barcode width (Range: 2 ≤ barWidth ≤ 6)
height	Number	Barcode height (Range: 1 ≤ height ≤ 255)
alignment	Number	Alignment **0:** Left **1:** Center **2:** Right
textPosition	Number	HRI (Human Readable Interface) print position **0:** None **1:** Above **2:** Below

[Example]

```
_Printer.print1DBarcode(
  function{ console.log("print1DBarcode success"); },
  function{ console.log("print1DBarcode failed"); },
  '1234567890', 110 /*Code128*/, 3, 50, 0, 1
)
```

5-2-20 Print QR Code

Prints the QR Code.

[Parameters]

Parameters	Data Type	Description
successCallback	Function	Callback function that is called when succeeded.
errorCallback	Function	Callback function that is called when failed.
data	String	QR code data
model	Number	QR code model **204:** model 1 **205:** model 2
alignment	Number	Alignment **0:** Left **1:** Center **2:** Right
moduleSize	Number	QR code size (Range: 1 ≤ moduleSize ≤ 8)
eccLevel	Number	Error correction level **48:** Level L **49:** Level M **50:** Level Q **51:** Level H

[Example]

```
_Printer.printQRCode(
  function{ console.log("printQRCode success"); },
  function{ console.log("printQRCode failed"); },
  'QRCode', 204, 0, 5, 49
)
```

5-2-21 Print PDF417

Prints the PDF417 code.

[Parameters]

Parameters	Data Type	Description
successCallback	Function	Callback function that is called when succeeded.
errorCallback	Function	Callback function that is called when failed.
data	String	PDF417 data
model	Number	PDF417 type **201:** standard PDF417 **202:** simplified PDF417
alignment	Number	Alignment **0:** Left **1:** Center **2:** Right
columnNumber	Number	The number of columns (Range: 0 ≤ columnNumber ≤ 30)
rowNumber	Number	The number of rows (Range: 3 ≤ rowNumber ≤ 90)
moduleWidth	Number	Module width (Range: 1 ≤ moduleWidth ≤ 4)
moduleHeight	Number	Module height (Range: 2 ≤ moduleHeight ≤ 8)
eccLevel	Number	Error correction level **48:** Error correction level 0 **49:** Error correction level 1 **50:** Error correction level 2 **51:** Error correction level 3 **52:** Error correction level 4 **53:** Error correction level 5 **54:** Error correction level 6 **55:** Error correction level 7

[Example]

```
_Printer.printPDF417(
  function{ console.log("printPDF417 success"); },
  function{ console.log("printPDF417 failed"); },
  'Hello Cordova', 201 /* standard PDF417 */, 0, 0, 0, 3, 4, 48
)
```

5-2-22 Print Data Matrix

Prints the data matrix code.

[Parameters]

Parameters	Data Type	Description
successCallback	Function	Callback function that is called when succeeded.
errorCallback	Function	Callback function that is called when failed.
data	String	Data Matrix data
alignment	Number	Alignment **0:** Left **1:** Center **2:** Right
moduleSize	Number	Module size (Range: 2 ≤ moduleSize ≤ 3)

[Example]

```
_Printer.printDataMatrix(
  function{ console.log("printDataMatrix success"); },
  function{ console.log("printDataMatrix failed"); },
  'DataMatrix', 0, 3
)
```

5-2-23 Check Batt Status

Gets the printer's current battery level.



Only works on models with batteries.

[Parameters]

Parameters	Data Type	Description
successCallback	Function	Callback function that receives a status of the battery. **0:** Battery level is full **16:** Battery level in in high **32:** Battery level in in middle **64:** Battery level in in low
errorCallback	Function	Callback function that is called when failed.

[Example]

```
_Printer.checkBattStatus(
  function(level){
     ...
  },
  Function{
     console.log("checkBattStatus failed");
  }
)
```

5-2-24 Check Printer Status

Gets the printer's current status.

[Parameters]

Parameters	Data Type	Description
successCallback	Function	Callback function that receives a status of the printer. **0:** Normal (No error) **1:** Cover open **2:** Paper empty **4:** Near-end paper **8:** Error (offline) **128:** Cash drawer signal is in LOW **256:** Cash drawer signal is in HIGH
errorCallback	Function	Callback function that is called when failed.

[Example]

```
_Printer.checkPrinterStatus(
  function(status){ console.log("checkPrinterStatus success");
        // if (status & POS_CONST.PrinterStatus.COVEROPEN)
        // if (status & POS_CONST.PrinterStatus.PAPEREMPTY)
        // if (status & POS_CONST.PrinterStatus.PAPER_NEAREND)
        // if (status & POS_CONST.PrinterStatus.ERROR)
        // if (status & POS_CONST.PrinterStatus.DRAWER_LOW)
        // if (status & POS_CONST.PrinterStatus.DRAWER_HIGH)
        // if (status & POS_CONST.PrinterStatus.BATT_HIGH)
        // if (status & POS_CONST.PrinterStatus.BATT_MID)
        // if (status & POS_CONST.PrinterStatus.BATT_LOW)
  },
  function{ console.log("checkPrinterStatus failed"); }
)
```

5-2-25 Asb Enable

Enable or disable the ASB (Automatic Status Back) function. When enabled, the printer status value is passed to the specified callback function whenever the printer status changes.

Use the '**setStatusUpdateEvent**' method to set the callback function to receive printer status information.

[Parameters]

Parameters	Data Type	Description
successCallback	Function	Callback function that is called when succeeded.
errorCallback	Function	Callback function that is called when failed.
enable	Boolean	Enable or disable printer status notification **true:** ASB enable **false:** ASB disable

[Example]

```
var PrinterStatusEventHandler = function(status) {
      // if (status & POS_CONST.PrinterStatus.COVEROPEN)
      // if (status & POS_CONST.PrinterStatus.PAPEREMPTY)
      // if (status & POS_CONST.PrinterStatus.PAPER_NEAREND)
      // if (status & POS_CONST.PrinterStatus.ERROR)
      // if (status & POS_CONST.PrinterStatus.DRAWER_LOW)
      // if (status & POS_CONST.PrinterStatus.DRAWER_HIGH)
      // if (status & POS_CONST.PrinterStatus.BATT_HIGH)
      // if (status & POS_CONST.PrinterStatus.BATT_MID)
      // if (status & POS_CONST.PrinterStatus.BATT_LOW)
};

_Printer.setStatusUpdateEvent(Function{
      _Printer.asbEnable(function () {
         console.log("asbEnable success");
      },function(error){
         console.log("asbEnable failed");
      }, true /*Enable*/);
},function(error){
   console.log("setStatusUpdateEvent failed");
}, PrinterStatusEventHandler);
```

5-2-26 Set Status Update Event

Sets a callback function to receive printer status information.



If the target platform is UWP, you must pass a callback function as the value to the third parameter "handler" to receive printer status information.

[Parameters]

Parameters	Data Type	Description
successCallback	Function	Callback function that receives a status of the printer.
errorCallback	Function	Callback function that is called when failed.
handler	Function	Callback function that receives a status of the printer. *** For Android and iOS, ignore this parameter.**

5-2-27 Open Drawer

Sends the cash drawer open signal to the printer.



Please note that the cash drawer has two PIN numbers (2 and 5), and the cash drawer may not open depending on the value of the third parameter.

[Parameters]

Parameters	Data Type	Description
successCallback	Function	Callback function that is called when succeeded.
errorCallback	Function	Callback function that is called when failed.
pinNubmer	Number	Cash drawer's PIN number **0:** PIN Number #2 **1:** PIN Number #5

5-2-28 Print PDF File

Prints a PDF file.



UWP does not support this method.

[Parameters]

Parameters	Data Type	Description
successCallback	Function	Callback function that is called when succeeded.
errorCallback	Function	Callback function that is called when failed.
filePath	String	The location of a PDF file
width	Number	Print width **-2:** Print with the original size of the image
alignment	Number	Alignment **0:** Left **1:** Center **2:** Right
startPage	Number	Start page number to print * The first page starts from '0'.
endPage	Number	End page number to print
threshold	Number	Threshold Range: 0 ≤ threshold ≤ 255
ditheringType	Number	Dithering type **0:** No dithering **1:** dithering
compressType	Number	Compress type **0:** No compress **1:** RLE **2:** LZMA

6 MPosControllerLabelPrinter Reference

6-1 Overview

• The “MPosControllerLabelPrinter” object provides methods for controlling label printers.

6-2 Methods

6-2-1 Select Interface

Set the communication interface type and address of the device (label printer) to connect.

[Parameters]

Parameters	Data Type	Description
successCallback	Function	Callback function that is called when succeeded.
errorCallback	Function	Callback function that is called when failed.
interfaceType	Number	Communication Interface **1:** Wi-Fi **2:** Ethernet **4:** Bluetooth **16:** USB
address	String	IP Address:portNumber, MAC address, Bluetooth S/N Example) 192.168.1.2:9100 Example) 74:F0:7D:E3:DF:EE

USB communication is only available on Android

6-2-2 Select Command Mode

Selects whether to communicate directly with the device or through the “B-gate”.

[Parameters]

Parameters	Data Type	Description
successCallback	Function	Callback function that is called when succeeded.
errorCallback	Function	Callback function that is called when failed.
commandMode	Number	Communication method between HOST and a device. **0:** via B-gate **1:** direct communication

6-2-3 Open Service

Initiates a communications connection with the device.

[Parameters]

Parameters	Data Type	Description
successCallback	Function	Callback function that is called when succeeded.
errorCallback	Function	Callback function that is called when failed.
id	Number	Identifier of the device to connect.
timeout	Number	Time out in second for the connection with host.

- Using “searchDevices” described in 3-2-5 of this manual, you can get A list of connectable device identifiers (IDs).
- If not using B-gate, set the device identifier (id) to -1.
  * Label printer identifier range: 1 ~ 9

[Example]

```
return new Promise(function (returnResolve, returnReject) {
    new Promise(function (resolve, reject) { _LabelPrinter.selectInterface(resolve, reject, _ConnectionInfo.interfaceType,0);})
     .then(function () { return new Promise(function (resolve, reject) { _LabelPrinter.selectCommandMode(resolve, reject, 0);});})
     .then(function () { return new Promise(function (resolve, reject) {
          _LabelPrinter.openService(resolve, function(error){
                returnReject();
            }, $('#deviceID').val(), _ConnectionInfo.timeout);
          });
     })
     .then(Function{
          returnResolve();
     });
});
```

6-2-4 Close Service

Terminates the communication connection with the device.



Calling this method during data transfer can cause the communication connection to close without completing the data transfer. Call this method by setting the appropriate timeout value according to the print speed and print length.

[Parameters]

Parameters	Data Type	Description
successCallback	Function	Callback function that is called when succeeded.
errorCallback	Function	Callback function that is called when failed.
timeout	Number	Time out in second for the disconnection with host.

[Example]

```
_LabelPrinter.closeService(
  function{ console.log("closeService success"); },
  function{ console.log("closeService failed"); },
  3
)
```

6-2-5 Direct IO

Sends custom data to the device.

[Parameters]

Parameters	Data Type	Description
successCallback	Function	Callback function that is called when succeeded.
errorCallback	Function	Callback function that is called when failed.
data	Object	User defined byte array.

[Example]

```
_LabelPrinter.directIO(
  function{ console.log("directIO success"); },
  function{ console.log("directIO failed"); },
  [0x0D, 0x0A]
)
```

6-2-6 Is Open

Returns whether the device is open for communication.

[Parameters]

Parameters	Data Type	Description
successCallback	Function	Callback function that receives the information of that the device is open or not.
errorCallback	Function	Callback function that is called when failed.

[Example]

```
_LabelPrinter.isOpen(
  function(open){
    if(open) { ... }
    else { ... }
  },
  function{ console.log("isOpen failed"); }
)
```

6-2-7 Get Device ID

Gets the device identifier value.

[Parameters]

Parameters	Data Type	Description
successCallback	Function	Callback function that receives the device identifier.
errorCallback	Function	Callback function that is called when failed.

[Example]

```
_LabelPrinter.getDeviceID(
  function(deviceId){ console.log("getDeviceID success"); },
  function{ console.log("getDeviceID failed"); }
)
```

6-2-8 Set Transaction

Enter or exit transaction mode to transfer the data in the buffer to the device (label printer).

[Parameters]

Parameters	Data Type	Description
successCallback	Function	Callback function that is called when succeeded.
errorCallback	Function	Callback function that is called when failed.
transactionMode	Number	Transaction Mode **0:** Transaction mode out and send data to the printer **1:** Transaction mode in

[Example]

```
return new Promise(function (returnResolve, returnReject) {
    new Promise(function (resolve, reject) { _LabelPrinter.setTransaction(resolve, reject, 1);})
     /* add more operations as required */
     .then(Function { return new Promise(function (resolve, reject) { _LabelPrinter.setTransaction(resolve, reject,
0);});});})
     .then(Function{
          returnResolve();
     });
});
```

6-2-9 Set Text Encoding

Sets the code page to use for character encoding.

[Parameters]

Parameters	Data Type	Description
successCallback	Function	Callback function that is called when succeeded.
errorCallback	Function	Callback function that is called when failed.
textEncoding	Number	Code page for encoding characters **437**: English (PC437) **850**: Western Europe (PC850) **852**: Latin2 (PC852) **860**: Portuguese (PC860) **863**: Canadian French (PC863) **865**: Norwegian (PC865) **1252**: Latin1 (WPC1252) **857**: Turkish (PC857) **737**: Greek (PC737) **1250**: Czech (WPC1250) **1253**: Greek (WPC1253) **1254**: Turkish (WPC1254) **855**: Cyrillic (PC855) **862**: Hebrew (PC862) **866**: Cyrillic (PC866) **1251**: Cyrillic (WPC1251) **1255**: Hebrew (WPC1255) **928**: Greek (PC928) **864**: Arabic (PC864) **775**: Baltic (PC775) **1257**: Baltic (WPC1257) **858**: Western Europe (PC858) **949**: Korean (KS5601) **932**: Japanese (ShiftJIS) **950**: Traditional Chinese (BIG5) **936**: Simplified Chinese (GB2312)

[Example]

```
_LabelPrinter.setTextEncoding(
  function{ console.log("setTextEncoding success"); },
  function{ console.log("setTextEncoding failed"); },
  437
)
```

6-2-10 Set Character set

Sets the code page to use for character encoding.

[Parameters]

Parameters	Data Type	Description
successCallback	Function	Callback function that is called when succeeded.
errorCallback	Function	Callback function that is called when failed.
characterset	Number	Code page for encoding characters **437**: English (PC437) **850**: Western Europe (PC850) **852**: Latin2 (PC852) **860**: Portuguese (PC860) **863**: Canadian French (PC863) **865**: Norwegian (PC865) **1252**: Latin1 (WPC1252) **857**: Turkish (PC857) **737**: Greek (PC737) **1250**: Czech (WPC1250) **1253**: Greek (WPC1253) **1254**: Turkish (WPC1254) **855**: Cyrillic (PC855) **862**: Hebrew (PC862) **866**: Cyrillic (PC866) **1251**: Cyrillic (WPC1251) **1255**: Hebrew (WPC1255) **928**: Greek (PC928) **864**: Arabic (PC864) **775**: Baltic (PC775) **1257**: Baltic (WPC1257) **858**: Western Europe (PC858) **949**: Korean (KS5601) **932**: Japanese (ShiftJIS) **950**: Traditional Chinese (BIG5) **936**: Simplified Chinese (GB2312)
internationalCharset	Number	International character set **0:** USA **1:** FRANCE **2:** GERMANY **3:** UK **4:** DENMARK1 **5:** SWEDEN **6:** ITALY **7:** SPAIN

		8: JAPAN **9:** NORWAY **10:** DENMARK 2 **11:** SPAIN 2 **12:** LATIN AMERICA **13:** KOREA **14:** SLOVENIA **15:** CHINA

[Example]

```
_LabelPrinter.setCharacterset(
  function{ console.log("setCharacterset success"); },
  function{ console.log("setCharacterset failed"); },
  437, 1
)
```

6-2-11 Set Printing Type

Sets the printing type of the printer.



Label printers have two types of thermal transfer: "direct thermal transfer" and "indirect thermal transfer" that transfers heat to the ribbon. Make sure that the model is loaded with ribbon, and then select the appropriate method.

[Parameters]

Parameters	Data Type	Description
successCallback	Function	Callback function that is called when succeeded.
errorCallback	Function	Callback function that is called when failed.
setPrintingType	String	Thermal transfer method **"d":** Direct thermal **"t":** Thermal transfer (Ribbon)

[Example]

```
_LabelPrinter.setPrintingType(
  function{ console.log("setPrintingType success"); },
  function{ console.log("setPrintingType failed"); },
  'd' /* Direct Thermal */
)
```

6-2-12 Set Margin

Sets the margins in the printing area.

[Parameters]

Parameters	Data Type	Description
successCallback	Function	Callback function that is called when succeeded.
errorCallback	Function	Callback function that is called when failed.
horizontalMargin	Number	Horizontal margin, in dot unit.
verticalMargin	Number	Vertical margin, in dot unit.

[Example]

```
_LabelPrinter.setMargin(
  function{ console.log("setMargin success"); },
  function{ console.log("setMargin failed"); },
  0, 0
)
```

6-2-13 Set Back Feed Option

Sets the back feed operations.

[Parameters]

Parameters	Data Type	Description
successCallback	Function	Callback function that is called when succeeded.
errorCallback	Function	Callback function that is called when failed.
isBackFeed	Boolean	Enable or disable back feed operations
stepQuantity	Number	Back feed step quantity **0**: default feeding quantity

[Example]

```
_LabelPrinter.setBackFeedOption(
  function{ console.log("setBackFeedOption success"); },
  function{ console.log("setBackFeedOption failed"); },
  true, 1
)
```

6-2-14 Set Length

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

[Parameters]

Parameters	Data Type	Description
successCallback	Function	Callback function that is called when succeeded.
errorCallback	Function	Callback function that is called when failed.
labelLength	Number	Label length, in dot unit.
gapLength	Number	Gap length between labels, in dot unit.
mediaType	String	Media type **"G":** Gap **"C":** Continuous **"B":** Black mark
offsetLength	Number	Offset length, in dot unit.

[Example]

```
_LabelPrinter.setLength(
  function{ console.log("setLength success"); },
  function{ console.log("setLength failed"); },
  1218, 24, 'G', 0
)
```

6-2-15 Set Width

Sets the print width.



For the maximum print width, see 1-2 “Supported Devices” in this manual.

[Parameters]

Parameters	Data Type	Description
successCallback	Function	Callback function that is called when succeeded.
errorCallback	Function	Callback function that is called when failed.
labelWidth	Number	Print width, in dot unit.

[Example]

```
_LabelPrinter.setWidth(
  function{ console.log("setWidth success"); },
  function{ console.log("setWidth failed"); },
  832
)
```

6-2-16 Set Buffer Mode

Enable or disable double buffer mode.

[Parameters]

Parameters	Data Type	Description
successCallback	Function	Callback function that is called when succeeded.
errorCallback	Function	Callback function that is called when failed.
doubleBuffer	Boolean	Double buffer mode **true:** Enable **false:** Disable

[Example]

```
_LabelPrinter.setBufferMode(
  function{ console.log("setBufferMode success"); },
  function{ console.log("setBufferMode failed"); },
  false
)
```

6-2-17 Set Speed

Sets print speed.

[Parameters]

Parameters	Data Type	Description
successCallback	Function	Callback function that is called when succeeded.
errorCallback	Function	Callback function that is called when failed.
speed	Number	Print speed (inch per second)

[Example]

```
_LabelPrinter.setSpeed(
  function{ console.log("setSpeed success"); },
  function{ console.log("setSpeed failed"); },
  4
)
```

6-2-18 Set Density

Sets the print density.

[Parameters]

Parameters	Data Type	Description
successCallback	Function	Callback function that is called when succeeded.
errorCallback	Function	Callback function that is called when failed.
density	Number	Print density (Range: 1 ~ 20)

[Example]

```
_LabelPrinter.setDensity(
  function{ console.log("setDensity success"); },
  function{ console.log("setDensity failed"); },
  10
)
```

6-2-19 Set Orientation

Sets the print orientation of the label.

[Parameters]

Parameters	Data Type	Description
successCallback	Function	Callback function that is called when succeeded.
errorCallback	Function	Callback function that is called when failed.
orientation	String	Print orientation **"T":** Top to bottom **"B":** Bottom to top

[Example]

```
_LabelPrinter.setOrientation(
  function{ console.log("setOrientation success"); },
  function{ console.log("setOrientation failed"); },
  'T'
)
```

6-2-20 Set Offset

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

[Parameters]

Parameters	Data Type	Description
successCallback	Function	Callback function that is called when succeeded.
errorCallback	Function	Callback function that is called when failed.
orientation	Number	offset length, in dot unit (Range: -100 dot ~ 100 dot)

[Example]

```
_LabelPrinter.setOffset(
  function{ console.log("setOffset success"); },
  function{ console.log("setOffset failed"); },
  0
)
```

6-2-21 Set Cutting Position

Sets the cutting position or tear-off position.

[Parameters]

Parameters	Data Type	Description
successCallback	Function	Callback function that is called when succeeded.
errorCallback	Function	Callback function that is called when failed.
cuttingPosition	Number	Offset length (Range: -100 dot ~ 100 dot)

[Example]

```
_LabelPrinter.setCuttingPosition(
  function{ console.log("setCuttingPosition success"); },
  function{ console.log("setCuttingPosition failed"); },
  0
)
```

6-2-22 Set Auto Cutter

Sets the cutting operations for label paper.

[Parameters]

Parameters	Data Type	Description
successCallback	Function	Callback function that is called when succeeded.
errorCallback	Function	Callback function that is called when failed.
isAutoCutter	Boolean	Cut or no cut. **true:** cut **false:** no cut
period	Number	Cutting cycle For example.) If set to 2, the paper will be cut every 2 sheets.

[Example]

```
_LabelPrinter.setAutoCutter(
  function{ console.log("setAutoCutter success"); },
  function{ console.log("setAutoCutter failed"); },
  1
)
```

6-2-23 Print Buffer

Starts printing the contents saved in the printer

[Parameters]

Parameters	Data Type	Description
successCallback	Function	Callback function that is called when succeeded.
errorCallback	Function	Callback function that is called when failed.
numberOfCopies	Number	The number of copies (Range: 1 ~ 65535)

[Example]

```
_LabelPrinter.printBuffer(
  function{ console.log("printBuffer success"); },
  function{ console.log("printBuffer failed"); },
  1
)
```

6-2-24 Check Printer Status

Gets the printer's current status.

[Parameters]

Parameters	Data Type	Description
successCallback	Function	Callback function that receives the status of the printer. **0:** Normal (no error) **1:** No paper **2:** Cover open **3:** Cutter error **4:** TPH overheat **5:** GAP sensor error **6:** No ribbon **7:** Building the buffer of labels **8:** Printing **9:** Paper stuck to filler **10**: Wait paper to be taken **11**: RFID write error
errorCallback	Function	Callback function that is called when failed.

[Example]

```
_LabelPrinter.checkPrinterStatus(
  function(status){ console.log("checkPrinterStatus success");
        // if (status & LBL_CONST.PrinterStatus.COVEROPEN)
        // if (status & LBL_CONST.PrinterStatus.PAPEREMPTY)
        // if (status & LBL_CONST.PrinterStatus.CUTTER_JAMMED)
        // if (status & LBL_CONST.PrinterStatus.TPH_OVERHEAT)
        // if (status & LBL_CONST.PrinterStatus.GAP_ERROR)
        // if (status & LBL_CONST.PrinterStatus.RIBBON_END)
        // if (status & LBL_CONST.PrinterStatus.BUILDING_LABEL_TO_BE_PRINTED)
        // if (status & LBL_CONST.PrinterStatus.PRINTING_LABEL)
        // if (status & LBL_CONST.PrinterStatus.LABEL_ISPAUSED)
  },
  function{ console.log("checkPrinterStatus failed"); }
)
```

6-2-25 Draw Text Device Font

Saves data for printing of the characters in the printer buffer using a bitmap font.

[Parameters]

Parameters	Data Type	Description
successCallback	Function	Callback function that is called when succeeded.
errorCallback	Function	Callback function that is called when failed.
data	String	Text data
xPos	Number	X coordinate, in dot unit.
yPos	Number	Y coordinate, in dot unit.
fontSelection	String	Font Selection **"0":** ASCII, 9 X 15 dots **"1":** ASCII, 12 X 20 dots **"2":** ASCII, 16 X 25 dots **"3":** ASCII, 19 X 30 dots **"4":** ASCII, 24 X 38 dots **"5":** ASCII, 32 X 40 dots **"6":** ASCII, 48 X 76 dots **"7":** ASCII, 22 X 34 dots **"8":** ASCII 28 X 44 dots **"9":** ASCII, 37 X 58 dots **"a":** Korean 16 X 16 dots alphabetic character or number: 9 X 15 dots **"b":** Korean 24 X 24 dots alphabetic character or number: 12 X 24 dots **"c":** Korean 20 X 20 dots alphabetic character or number: 12 X 20 dots **"d":** Korean 26 X 26 dots alphabetic character or number: 16 X 30 dots **"e":** Korean 20 X 26 dots alphabetic character or number: 12 X 20 dots **"f":** Korean 38 X 38 dots alphabetic character or number: 22 X 34 dots **"m":** Simplified Chinese(GB2312) 24 X 24 dots alphabetic character or number: 12 X 24 dots **"m":** Traditional Chinese(BIG5) 24 X 24 dots alphabetic character or number: 12 X 24 dots **"j":** Japanese 24 X 24 dots alphabetic character or number: 12 X 24 dots
fontWidth	Number	Magnification of the font width (Range: 1 ~ 4)
fontHeight	Number	Magnification of the font height (Range: 1 ~ 4)
rightSpace	Number	Right spacing (Default: 0 dot)
rotation	Number	Rotation

		0: None **1:** Rotate 90 degrees **2:** Rotate 180 degrees **3:** Rotate 270 degrees
reverse	Boolean	Reverse or not
bold	Boolean	Bold or not
rightToLeft	Boolean	Text print direction (left to right, right to left)
alignment	Number	Text alignment **0:** Left **1:** right

[Example]

```
_LabelPrinter.drawTextDeviceFont(
  function{ console.log("drawTextDeviceFont success"); },
  function{ console.log("drawTextDeviceFont failed"); },
  "Bitmap Font", 50, 0, '1', 1, 1, 0, 0, false, false, false, 0
)
```

6-2-26 Draw Text Vector Font

Saves data for printing of the characters in the printer buffer using a vector font.

[Parameters]

Parameters	Data Type	Description
successCallback	Function	Callback function that is called when succeeded.
errorCallback	Function	Callback function that is called when failed.
data	String	Text data
xPos	Number	X coordinate, in dot unit.
yPos	Number	Y coordinate, in dot unit.
fontSelection	String	Font selection **"U":** ASCII **"K":** Korean (KS5601) **"B":** Chinse (BIG5) **"G":** Chinese (GB2312) **"J":** Japanese (Shift-JIS) **"a":** OCR-A **"b":** OCR-B
fontWidth	Number	Font width, in dot unit.
fontHeight	Number	Font height, in dot unit.
rightSpace	Number	Right spacing (default: 0 dot)
rotation	Number	Rotation **0:** None **1:** Rotate 90 degrees **2:** Rotate 180 degrees **3:** Rotate 270 degrees
reverse	Boolean	Reverse or not
bold	Boolean	Bold or not
italic	Boolean	Italic or not
rightToLeft	Boolean	Text print direction (left to right, right to left)
alignment	Number	Text alignment **0:** Left **1:** Right **2:** Center

[Example]

```
_LabelPrinter.drawTextVectorFont(
  function{ console.log("drawTextVectorFont success"); },
  function{ console.log("drawTextVectorFont failed"); },
  "Vector Font", 50, 50, ‘U’, 24, 24, 3, 0, false, false, false, false, 0
)
```

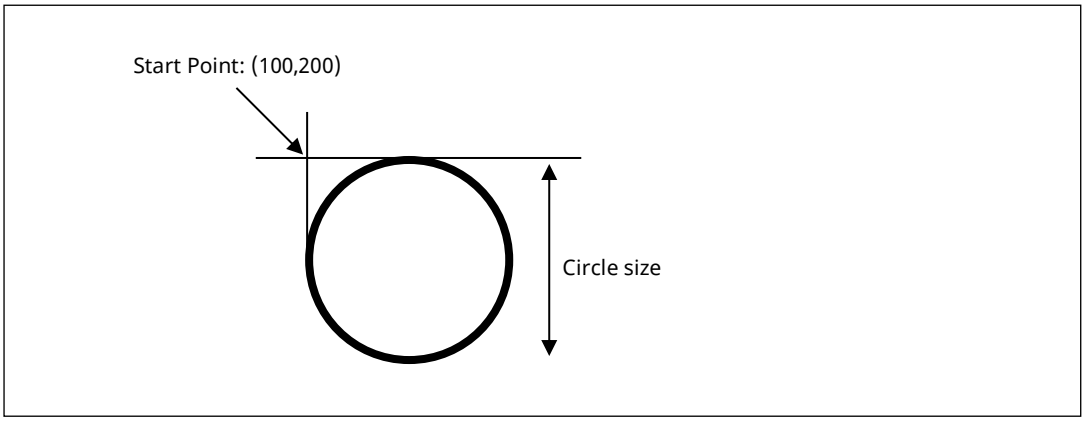
6-2-27 Draw Circle

Saves circle print data in the printer buffer.

[Parameters]

Parameters	Data Type	Description
successCallback	Function	Callback function that is called when succeeded.
errorCallback	Function	Callback function that is called when failed.
startPosX	Number	X coordinate, in dot unit.
startPosY	Number	Y coordinate, in dot unit.
sizeSelection	Number	Circle size **1:** 40 × 40 dots (W × H) **2:** 56 × 56 dots (W × H) **3:** 72 × 72 dots (W × H) **4:** 88 × 88 dots (W × H) **5:** 104 × 104 dots (W × H) **6:** 168 × 168 dots (W × H)
multiplier	Number	Magnification (Range: 1 ~ 4)

* Start: intersection of tangents on the left and top



[Example]

```
_LabelPrinter.drawCircle(
  function{ console.log("drawCircle success"); },
  function{ console.log("drawCircle failed"); },
  50, 10, 3 /* 72x72 */, 1
)
```

6-2-28 Draw Block

Saves square print data in the printer buffer.

[Parameters]

Parameters	Data Type	Description
successCallback	Function	Callback function that is called when succeeded.
errorCallback	Function	Callback function that is called when failed.
startPosX	Number	Top-left X coordinate, in dot unit.
startPosY	Number	Top-left Y coordinate, in dot unit.
endPosX	Number	Bottom-right X coordinate, in dot unit.
endPosY	Number	Bottom-right Y coordinate, in dot unit.
option	String	Drawing options **'O':** Repeat drawing on places where lines overlap **'E':** Do not draw on places where lines overlap **'D':** Delete line **'S':** Diagonal line **'B':** Quadrangle edge
thickness	Number	Thickness * Only applies if the 'option' value is 'S' or 'B'

[Example]

```
_LabelPrinter.drawBlock(
  function{ console.log("drawBlock success"); },
  function{ console.log("drawBlock failed"); },
  50, 100, 100, 200, 'B', 4
)
```

6-2-29 Draw Image File

Saves print data for a bitmap file residing on the HOST in the printer buffer.

- For Android, only files on the SD card can be printed.
- Depending on the version of the platform (Android, iOS, UWP), you may not be able to print if you do not have file access.

[Parameters]

Parameters	Data Type	Description
successCallback	Function	Callback function that is called when succeeded.
errorCallback	Function	Callback function that is called when failed.
path	String	The location of the image file.
xPos	Number	X coordinate, in dot unit.
yPos	Number	Y coordinate, in dot unit.
width	Number	Print width
brightness	Number	Brightness Range: 0 ≤ brightness ≤ 100 (Android, UWP) Range: -100 ≤ brightness ≤ 100 (iOS)
isDithering	Boolean	Whether or not to apply dithering
isCompress	Boolean	Whether or not to compress data

[Example]

```
_LabelPrinter.drawImageFile(
  function{ console.log("drawImageFile success"); },
  function{ console.log("drawImageFile failed"); },
  path, 50, 50, 300, 50, true, true
)
```

6-2-30 Draw Image with Base64

Saves print data of the graphic image encoded in Base64 in the printer buffer.

[Parameters]

Parameters	Data Type	Description
successCallback	Function	Callback function that is called when succeeded.
errorCallback	Function	Callback function that is called when failed.
imageSource	String	Image data encoded in Base64.
xPos	Number	X coordinate, in dot unit.
yPos	Number	Y coordinate, in dot unit.
width	Number	Print width
brightness	Number	Brightness Range: 0 ≤ brightness ≤ 100 (Android, UWP) Range: -100 ≤ brightness ≤ 100 (iOS)
isDithering	Boolean	Whether or not to apply dithering
isCompress	Boolean	Whether or not to compress data

[Example]

```
_LabelPrinter.drawImageWithBase64(
  function{ console.log("drawImageWithBase64 success"); },
  function{ console.log("drawImageWithBase64 failed"); },
  imageSource, 50, 50, 300, 50, true, true
)
```

6-2-31 Draw Barcode 1D

Saves one-dimensional barcode print data in the printer buffer.

[Parameters]

Parameters	Data Type	Description
successCallback	Function	Callback function that is called when succeeded.
errorCallback	Function	Callback function that is called when failed.
data	String	Barcode data.
xPos	Number	X coordinate, in dot unit.
yPos	Number	Y coordinate, in dot unit.
barcodeType	Number	Barcode type **0:** Code39 **1:** Code128 **2:** Interleaved 2of 5 **3:** Codabar **4:** Code93 **5:** UPC-A **6:** UPC-E **7:** EAN13 **8:** EAN8 **9:** UCC/EAN128 **10:** Code11 **11:** Planet **12:** Industrial 2 of 5 **13:** Standard 2 of 5 **14:** Logmars **15:** UPC/EAN Extensions **16:** Postnet
widthNarrow	Number	Narrow bar's width
widthWide	Number	Wide bar's width
height	Number	Barcode height
HRI	Number	HRI (Human Readable Interface) print position **0:** None **1:** Barcode below (Font size 1) **2:** Barcode above (Font size 1) **3:** Barcode below (Font size 2) **4:** Barcode above (Font size 2) **5:** Barcode below (Font size 3) **6:** Barcode above (Font size 3) **7:** Barcode below (Font size 4) **8:** Barcode above (Font size 4)
quietZoneWidth	Number	Quiet zone width (Range: 0 ~ 20)
rotation	Number	Rotation

		0: None **1:** Rotate 90 degrees **2:** Rotate 180 degrees **3:** Rotate 270 degrees

[Example]

```
_LabelPrinter.drawBarcode1D(
  function{ console.log("drawBarcode1D success"); },
  function{ console.log("drawBarcode1D failed"); },
  "1234567890", 50, 50, 1/*Code128*/, 2, 2, 50, 5, 0, 0
)
```

6-2-32 Draw Barcode Maxi Code

Saves Maxicode print data to the printer buffer.

[Parameters]

Parameters	Data Type	Description
successCallback	Function	Callback function that is called when succeeded.
errorCallback	Function	Callback function that is called when failed.
data	String	Barcode data
xPos	Number	X coordinate, in dot unit.
yPos	Number	Y coordinate, in dot unit.
mode	Number	Maxicode mode **0:** mode 0 **2:** mode 2 **3:** mode 3 **4:** mode 4

[Example]

```
_LabelPrinter.drawBarcodeMaxiCode(
  function{ console.log("drawBarcodeMaxiCode success"); },
  function{ console.log("drawBarcodeMaxiCode failed"); },
  "1234567890", 50, 50, 0
)
```

6-2-33 Draw Barcode PDF417

Saves PDF417 print data to the printer buffer.

[Parameters]

Parameters	Data Type	Description
successCallback	Function	Callback function that is called when succeeded.
errorCallback	Function	Callback function that is called when failed.
Data	String	Barcode data.
xPos	Number	X coordinate, in dot unit.
yPos	Number	Y coordinate, in dot unit.
maximumRowCount	Number	The number of rows (Range: 3 ~ 90)
maximumColumnCount	Number	The number of columns (Range: 1 ~ 30)
errorCorrectionLevel	Number	Error correction level (Range: 0 ~ 8)
dataCompressionMethod	Number	Data compression method **0:** Text, 2 characters per codeword **1:** Numeric, 2.93 characters per codeword **2:** Binary, 1.2 bytes per codeword
hri	Number	HRI (Human Readable Interface) print position **0:** None **2:** Below
barcodeOriginPoint	Number	The origin point of the barcode (default: 1) **0:** Center **1:** Left-top
moduleWidth	Number	Module width (Range: 2 ~ 9)
barHeight	Number	Bar's height (Range: 4 ~ 99)
rotation	Number	Rotation **0:** None **1:** Rotate 90 degrees **2:** Rotate 180 degrees **3:** Rotate 270 degrees

[Example]

```
_LabelPrinter.drawBarcodePDF417(
  function{ console.log("drawBarcodePDF417 success"); },
  function{ console.log("drawBarcodePDF417 failed"); },
  "PDF417CODE", 50, 150, 30, 5, 0, 0, true, 1, 3, 10, 0
)
```

6-2-34 Draw Barcode QR Code

Saves QR code print data to the printer buffer.

[Parameters]

Parameters	Data Type	Description
successCallback	Function	Callback function that is called when succeeded.
errorCallback	Function	Callback function that is called when failed.
data	String	Barcode data
xPos	Number	X coordinate, in dot unit.
yPos	Number	Y coordinate, in dot unit.
size	Number	Size (Range: 1 ~ 9)
model	Number	Model **1:** Model 1 **2:** Model 2
errorCorrectionLevel	Number	Error correction level **76:** Error correction level 7% **77:** Error correction level 15% **81:** Error correction level 25% **72:** Error correction level 35%
rotation	Number	Rotation **0:** None **1:** Rotate 90 degrees **2:** Rotate 180 degrees **3:** Rotate 270 degrees

[Example]

```
_LabelPrinter.drawBarcodeQRCode(
  function{ console.log("drawBarcodeQRCode success"); },
  function{ console.log("drawBarcodeQRCode failed"); },
  "QRCODE", 50, 250, 4, 2 /* Model 2*/, 76, 0
)
```

6-2-35 Draw Barcode Data Matrix

Saves Data matrix print data to the printer buffer.

[Parameters]

Parameters	Data Type	Description
successCallback	Function	Callback function that is called when succeeded.
errorCallback	Function	Callback function that is called when failed.
data	String	Barcode data.
xPos	Number	X coordinate, in dot unit.
yPos	Number	Y coordinate, in dot unit.
size	Number	Size (Range: 1 ~ 4)
reverse	Boolean	Normal or reverse
rotation	Number	Rotation **0:** None **1:** Rotate 90 degrees **2:** Rotate 180 degrees **3:** Rotate 270 degrees

[Example]

```
_LabelPrinter.drawBarcodeDataMatrix(
  function{ console.log("drawBarcodeDataMatrix success"); },
  function{ console.log("drawBarcodeDataMatrix failed"); },
  "DataMatrix", 50, 350, 4, false, 0
)
```

6-2-36 Draw Barcode Aztec

Saves Aztec print data to the printer buffer.

[Parameters]

Parameters	Data Type	Description
successCallback	Function	Callback function that is called when succeeded.
errorCallback	Function	Callback function that is called when failed.
data	String	Barcode data.
xPos	Number	X coordinate, in dot unit.
yPos	Number	Y coordinate, in dot unit.
size	Number	Size (Range: 1 ~ 10)
extendedChannel	Boolean	Extended channel interpretation code
errorCorrectionLevel	Number	Error correction level **0:** default Error correction level **1~99:** Error correction level by percent (1~99%) **101~104:** 1~4 layer compact symbol **201~232:** 1~32 layer full range symbol **300:** Simple Aztec "Rune"
menuSymbol	Boolean	Menu symbol **true:** Enable menu symbol **false:** Disable menu symbol
numberOfSymbols	Number	Number of symbols for structured append (Range: 1 ~ 26)
optionalID	String	Optional ID field for structured append: ID field string (Maximum 24)
rotation	Number	Rotation **0:** None **1:** Rotate 90 degrees **2:** Rotate 180 degrees **3:** Rotate 270 degrees

[Example]

```
_LabelPrinter.drawBarcodeAztec(
  function{ console.log("drawBarcodeAztec success"); },
  function{ console.log("drawBarcodeAztec failed"); },
  "Aztec", 50, 350, 4, 0, true, 1, “Aztec1”, 0
)
```

6-2-37 Draw Barcode Code49

Saves Code 49 print data to the printer buffer.

[Parameters]

Parameters	Data Type	Description
successCallback	Function	Callback function that is called when succeeded.
errorCallback	Function	Callback function that is called when failed.
data	String	Barcode data.
xPos	Number	X coordinate, in dot unit.
yPos	Number	Y coordinate, in dot unit.
widthNarrow	Number	Narrow bar's width.
widthWide	Number	Wide bar's width.
height	Number	Barcode height.
hri	Number	HRI (Human Readable Interface) print position. **0:** None **1:** Above **2:** Below
startingMode	Number	Starting Mode **0:** Regular Alphanumeric Mode **1:** Multiple Read Alphanumeric **2:** Regular numeric Mode **3:** Group Alphanumeric Mode **4:** Regular Alphanumeric Shift 1 **5:** Regular Alphanumeric Shift 2 **7:** Automatic Mode
rotation	Number	Rotation **0:** None **1:** Rotate 90 degrees **2:** Rotate 180 degrees **3:** Rotate 270 degrees

[Example]

```
_LabelPrinter.drawBarcodeCode49(
  function{ console.log("drawBarcodeCode49 success"); },
  function{ console.log("drawBarcodeCode49 failed"); },
  "Code49", 50, 350, 1, 1, 300, 2, 0, 0
)
```

6-2-38 Draw Barcode Coda Block

Saves Coda block print data to the printer buffer.

[Parameters]

Parameters	Data Type	Description
successCallback	Function	Callback function that is called when succeeded.
errorCallback	Function	Callback function that is called when failed.
data	String	Barcode data.
xPos	Number	X coordinate, in dot unit.
yPos	Number	Y coordinate, in dot unit.
widthNarrow	Number	Narrow bar's width
widthWide	Number	Wide bar's width
height	Number	Barcode height
securityLevel	Boolean	Enable or disable the security level
numberOfCharactersPerRow	Number	The number of characters per line (Range: 2 ~ 62)
mode	String	Coda block mode **"A":** Code 39 character set **"E":** Code 128 character set **"F":** Code 128 character set + FNC1
rotation	Number	Rotation **0:** None **1:** Rotate 90 degrees **2:** Rotate 180 degrees **3:** Rotate 270 degrees

[Example]

```
_LabelPrinter.drawBarcodeCodaBlock(
  function{ console.log("drawBarcodeCodaBlock success"); },
  function{ console.log("drawBarcodeCodaBlock failed"); },
  "CodeBlock", 50, 350, 1, 1, 300, true, 2, 'A', 0
)
```

6-2-39 Draw Barcode Micro PDF

Saves Micro PDF print data to the printer buffer.

[Parameters]

Parameters	Data Type	Description
successCallback	Function	Callback function that is called when succeeded.
errorCallback	Function	Callback function that is called when failed.
data	String	Barcode data.
xPos	Number	X coordinate, in dot unit.
yPos	Number	Y coordinate, in dot unit.
moduleWidth	Number	Module width (Range: 2 ~ 8)
moduleHeight	Number	Module height (Range: 1 ~ 99)
mode	Number	Mode (Range: 0 ~ 33)
rotation	Number	Rotation **0:** None **1:** Rotate 90 degrees **2:** Rotate 180 degrees **3:** Rotate 270 degrees

[Example]

```
_LabelPrinter.drawBarcodeMicroPDF(
  function{ console.log("drawBarcodeMicroPDF success"); },
  function{ console.log("drawBarcodeMicroPDF failed"); },
  "MicroPDF", 50, 350, 2, 50, 0, 0
)
```

6-2-40 Draw Barcode IMB

Saves IMB print data to the printer buffer.

[Parameters]

Parameters	Data Type	Description
successCallback	Function	Callback function that is called when succeeded.
errorCallback	Function	Callback function that is called when failed.
data	String	Barcode data
xPos	Number	X coordinate, in dot unit.
yPos	Number	Y coordinate, in dot unit.
hri	Number	HRI (Human Readable Interface) print position **0:** None **1:** Below
rotation	Number	Rotation **0:** None **1:** Rotate 90 degrees **2:** Rotate 180 degrees **3:** Rotate 270 degrees

[Example]

```
_LabelPrinter.drawBarcodeIMB(
  function{ console.log("drawBarcodeIMB success"); },
  function{ console.log("drawBarcodeIMB failed"); },
  "BarcodeIMB", 50, 350, true, 0
)
```

6-2-41 Draw Barcode MSI

Saves MSI print data to the printer buffer.

[Parameters]

Parameters	Data Type	Description
successCallback	Function	Callback function that is called when succeeded.
errorCallback	Function	Callback function that is called when failed.
data	String	Barcode data.
xPos	Number	X coordinate, in dot unit.
yPos	Number	Y coordinate, in dot unit.
narrowWidth	Number	Narrow bar's width
widthWide	Number	Wide bar's width
height	Number	Barcode height.
checkDigitSelection	Number	Check digit **0:** None **1:** 1 mod 10 **2:** 2 mod 10 **3:** 1 mod 11, 1 mod 10
printCheckDigit	Boolean	Print the check-digit or not.
hri	Number	HRI (Human Readable Interface) print position **0:** None **1:** Above **2:** Below
rotation	Number	Rotation **0:** None **1:** Rotate 90 degrees **2:** Rotate 180 degrees **3:** Rotate 270 degrees

[Example]

```
_LabelPrinter.drawBarcodeMSI(
  function{ console.log("drawBarcodeMSI success"); },
  function{ console.log("drawBarcodeMSI failed"); },
  "BarcodeMSI", 50, 350, 2, 2, 50, 0, false, 2 , 0
)
```

6-2-42 Draw Barcode Plessey

Saves Plessey print data to the printer buffer.

[Parameters]

Parameters	Data Type	Description
successCallback	Function	Callback function that is called when succeeded.
errorCallback	Function	Callback function that is called when failed.
data	String	Barcode data.
xPos	Number	X coordinate, in dot unit.
yPos	Number	Y coordinate, in dot unit.
narrowWidth	Number	Narrow bar's width
widthWide	Number	Wide bar's width
height	Number	Barcode height.
printCheckDigit	Boolean	Print the check-digit on HRI (Human Readable Interface) or not.
hri	Number	HRI (Human Readable Interface) print position **0:** None **1:** Above **2:** Below
rotation	Number	Rotation **0:** None **1:** Rotate 90 degrees **2:** Rotate 180 degrees **3:** Rotate 270 degrees

[Example]

```
_LabelPrinter.drawBarcodePlessey(
  function{ console.log("drawBarcodePlessey success"); },
  function{ console.log("drawBarcodePlessey failed"); },
  "BarcodeMSI", 50, 350, 2, 2, 50, false, 2 , 0
)
```

6-2-43 Draw Barcode TLC39

Saves TLC39 print data to the printer buffer.

[Parameters]

Parameters	Data Type	Description
successCallback	Function	Callback function that is called when succeeded.
errorCallback	Function	Callback function that is called when failed.
data	String	Barcode data
xPos	Number	X coordinate, in dot unit.
yPos	Number	Y coordinate, in dot unit.
narrowWidth	Number	Narrow bar's width
widthWide	Number	Wide bar's width
height	Number	Barcode height
rowHeightOfMicroPDF417	Number	Row height of the Micro PDF417 (Range: 1 ~ 255)
narrowWidthOfMicroPDF417	Number	Narrow bar width of the Micro PDF417 (Range: 1 ~ 10)
rotation	Number	Rotation **0:** None **1:** Rotate 90 degrees **2:** Rotate 180 degrees **3:** Rotate 270 degrees

[Example]

```
_LabelPrinter.drawBarcodeTLC39(
  function{ console.log("drawBarcodeTLC39 success"); },
  function{ console.log("drawBarcodeTLC39 failed"); },
  " BarcodeTLC39", 50, 350, 2, 2, 50, 50, 2 , 0
)
```

6-2-44 Draw Barcode RSS

Saves RSS print data to the printer buffer.

[Parameters]

Parameters	Data Type	Description
successCallback	Function	Callback function that is called when succeeded.
errorCallback	Function	Callback function that is called when failed.
data	String	Barcode data.
xPos	Number	X coordinate, in dot unit.
yPos	Number	Y coordinate, in dot unit.
barcodeType	Number	RSS Barcode type **0:** RSS14 **1:** RSS14 Truncated **2:** RSS14 Stacked **3:** RSS14 Stacked Omnidirectional **4:** RSS Limited **5:** RSS Expanded **6:** UPC-A **7:** UPC-E **8:** EAN-13 **9:** EAN-8 **10:** UCC/EAN-128, CC-A/B **11:** UCC/EAN-128, CC-C
magnification	Number	Magnification (Range: 1 ~ 10)
separatorHeight	Number	Separator height (Range: 0 ~ 22)
barcodeHeight	Number	Barcode height
segmentWidth	Number	Segment width (Range: 0 ~ 22, Even numbers)
rotation	Number	Rotation **0:** None **1:** Rotate 90 degrees **2:** Rotate 180 degrees **3:** Rotate 270 degrees

[Example]

```
_LabelPrinter.drawBarcodeTLC39(
  function{ console.log("drawBarcodeTLC39 success"); },
  function{ console.log("drawBarcodeTLC39 failed"); },
  "BarcodeRSS", 50, 350, 0, 1, 5, 50, 2, 0
)
```

6-2-45 Setup RFID

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

[Parameters]

Parameters	Data Type	Description
successCallback	Function	Callback function that is called when succeeded.
errorCallback	Function	Callback function that is called when failed.
rfidType	Number	RFID transponder Type **0:** none **1:** ISO 18000-6 Type A **2**: ISO 18000-6 Type B **3**: EPC Class 0 **4**: EPC Class 1 **5**: EPC Class 1 Generation 2
numberOfRetries	Number	Number of retries when coding fails (Range: 1 ~ 10)
numberOfLabel	Number	Number of retry labels when coding fails (Range: 1 ~ 5)
radioPower	Number	Signal strength (Range: 0 ~ 30)

[Example]

```
_LabelPrinter.setupRFID(
  function{ console.log("setupRFID success"); },
  function{ console.log("setupRFID failed"); },
  5, 3, 2, 15
)
```

6-2-46 Set RFID Position

Set the coding position directly on the RFID label.

[Parameters]

Parameters	Data Type	Description
successCallback	Function	Callback function that is called when succeeded.
errorCallback	Function	Callback function that is called when failed.
transPosition	Number	Coding position of RFID label (y-axis value)

[Example]

```
_LabelPrinter.setRFIDPosition(
  function{ console.log("setRFIDPosition success"); },
  function{ console.log("setRFIDPosition failed"); },
  320
)
```

6-2-47 Set RFID Password

Set RFID Access Password and Kill Password.

[Parameters]

Parameters	Data Type	Description
successCallback	Function	Callback function that is called when succeeded.
errorCallback	Function	Callback function that is called when failed.
oldAccessPwd	String	Access Password currently in use
oldKillPwd	String	Kill Password currently in use
newAccessPwd	String	Access Password to be modified
newKillPwd	String	Kill Password to modify

[Example]

```
_LabelPrinter.setRFIDPassword(
  function{ console.log("setRFIDPassword success"); },
  function{ console.log("setRFIDPassword failed"); },
  “00000000”, “00000000”, “33333333”, “33333333”
)
```

6-2-48 Set EPC Data Structure

Define the EPC data structure for writing EPC data.

[Parameters]

Parameters	Data Type	Description
successCallback	Function	Callback function that is called when succeeded.
errorCallback	Function	Callback function that is called when failed.
totalSize	Number	Total bits in the field
fieldSizes	String	Bits of each field (Comma (',') is used for the separator)

[Example]

```
_LabelPrinter.setEPCDataStructure(
  function{ console.log("setEPCDataStructure success"); },
  function{ console.log("setEPCDataStructure failed"); },
  64, “2,3,14,20,25”
)
```

6-2-49 Write RFID

Write RFID label.

[Parameters]

Parameters	Data Type	Description
successCallback	Function	Callback function that is called when succeeded.
errorCallback	Function	Callback function that is called when failed.
dataType	Number	data type **65:** ASCII **72:** HEX **69**: EPC (must be used after setEPCDataStructure) **85**: User area hex
startingBlockNumber	Number	Block to start writing
dataLength	Number	Number of bytes to write
data	String	Input according to dataType **65:** ASCII format **72:** hexa format **69**: decimal number (bits per set field) **85**: hexa format

[Example]

```
_LabelPrinter.writeRFID(
  function{ console.log("writeRFID success"); },
  function{ console.log("writeRFID failed"); },
  65, 4, 12, '123456789012'
)
```

6-2-50 Lock RFID

Lock Kill/Access/EPC Data using Access Password.



Be sure to use it after running setRFIDPassword.

[Parameters]

Parameters	Data Type	Description
successCallback	Function	Callback function that is called when succeeded.
errorCallback	Function	Callback function that is called when failed.

[Example]

```
_LabelPrinter.writeRFID(
  function{ console.log("lockRFID success"); },
  function{ console.log("lockRFID failed"); }
)
```

6-5-51 Draw PDF File

Saves print data for a PDF file in the printer buffer.



UWP does not support this method.

[Parameters]

Parameters	Data Type	Description
successCallback	Function	Callback function that is called when succeeded.
errorCallback	Function	Callback function that is called when failed.
filePath	String	The location of the image file.
xPos	Number	X coordinate, in dot unit.
yPos	Number	Y coordinate, in dot unit.
width	Number	Print width **-2:** Print with the original size of the image
pageNumber	Number	Page number to print * The first page starts from '0'.
threshold	Number	Threshold Range: 0 ≤ threshold ≤ 255
ditheringType	Number	Dithering type **0:** No dithering **1:** dithering
compressType	Number	Compress type **0:** No compress **1:** RLE **2:** LZMA

7. MPosControllerBCD Reference

7-1 Overview

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

7-2 Methods

7-2-1 Display String

Displays the string on the display device.

[Parameters]

- string data: string to display
- int codePage: code page for encoding characters

DeviceType Enum	Value	Description
MPOS_CODEPAGE_PC437	0	English (PC437)
MPOS_CODEPAGE_KATAKANA	1	Japanese (KATAKANA)
MPOS_CODEPAGE_PC850	2	Western Europe (PC850)
MPOS_CODEPAGE_PC860	3	Portuguese (PC860)
MPOS_CODEPAGE_PC863	4	Canadian French (PC863)
MPOS_CODEPAGE_PC865	5	Norwegian (PC865)
MPOS_CODEPAGE_WPC1252	16	Latin1 (WPC1252)
MPOS_CODEPAGE_PC866	17	Cyrillic (PC866)
MPOS_CODEPAGE_PC852	18	Latin 2 (PC852)
MPOS_CODEPAGE_PC858	19	Western Europe (PC858)
MPOS_CODEPAGE_PC862	21	Hebrew (PC862)
MPOS_CODEPAGE_PC864	22	Arabic (PC864)
MPOS_CODEPAGE_WPC1253	24	Greek (WPC1253)
MPOS_CODEPAGE_WPC1254	25	Turkish (WPC1254)
MPOS_CODEPAGE_WPC1257	26	Baltic (WPC1257)
MPOS_CODEPAGE_FARSI	27	Persian
MPOS_CODEPAGE_WPC1251	28	Cyrillic (WPC1251)
MPOS_CODEPAGE_PC737	29	Greek (PC737)
MPOS_CODEPAGE_PC775	30	Baltic (PC775)
MPOS_CODEPAGE_HEBREW_OLD	32	Hebrew (Hebrew Old)
MPOS_CODEPAGE_WPC1255	33	Hebrew (WPC1255)
MPOS_CODEPAGE_PC855	36	Cyrillic (PC855)
MPOS_CODEPAGE_PC857	37	Turkish (PC857)
MPOS_CODEPAGE_PC928	38	Greek (PC928)
MPOS_CODEPAGE_WPC1256	40	Arabic (WPC1256)

MPOS_CODEPAGE_WPC1258	41	Vietnamese (WPC1258)
MPOS_CODEPAGE_WPC1250	47	Czech (WPC1250)
MPOS_CODEPAGE_LATIN9	48	Latin 9 (Latin 9)
MPOS_CODEPAGE_TCVN3	49	Vietnamese (TCVN3)
MPOS_CODEPAGE_TCVN3_CAPTIAL	50	Vietnamese (TCVN3 Capital)
MPOS_CODEPAGE_VISCII	51	Vietnamese (VISCII)

• int internationalCharacterSet: characters to display

DeviceType Enum	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	SPAIN 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)

[Example]

```
_Bcd.displayString(
  function{ console.log("displayString success"); },
  function{ console.log("displayString failed"); },
  'Hello Cordova' + '\n', 0, 0, 1
)
```

7-2-2 Clean Screen

Deletes everything on the display device.

[Example]

```
_Bcd.cleanScreen (
  function{ console.log("cleanScreen success"); },
  function{ console.log("cleanScreen failed");}
)
```

7-2-3 Store Image File

Downloads images to the display device.

[Parameters]
• string fileName: image path
• 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

[Example]

```
_Bcd. storeImageFile (
  function{ console.log("storeImageFile success"); },
  function{ console.log("storeImageFile failed"); } ,
  “fileName”, 160, 1
)
```

7-2-4 Display Image

Displays images on the display device.

[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)

[Example]

```
_Bcd. displayImage (
  function{ console.log("displayImage success"); },
  function{ console.log("displayImage failed"); } ,
  1, 0, 0
)
```

7-2-5 Clear Image

Deletes the images downloaded to the display device.

[Parameters]

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

[Example]

```
_Bcd. clearImage (
  function{ console.log("clearImage success"); },
  function{ console.log("clearImage failed"); } ,
  1
)
```

7-2-6 Show Cursor

Shows or hides the cursor on the display device.

[Parameters]

• int cursor: show/hide cursor

DeviceType Enum	Value	Description
MPOS_CURSOR_DISABLE	0	Hides the cursor
MPOS_CURSOR_ENABLE	1	Shows the cursor

[Example]

```
_Bcd.showCursor (
  function{ console.log("showCursor success"); },
  function{ console.log("showCursor failed"); } ,
  1
)
```

8. MPosControllerDallasKey Reference

8-1 Overview

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

9. MPosControllerHID Reference

9-1 Overview

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

10. MPosControllerMSR Reference

10-1 Overview

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

11. MPosControllerNFC Reference

11-1 Overview

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

12. MPosControllerRFID Reference

12-1 Overview

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

13. MPosControllerScale Reference

13-1 Overview

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

14. MPosControllerScanner Reference

14-1 Overview

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

15. MPosControllerUSBSerial Reference

15-1 Overview

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

Revision history

Ver.	Date	Description
1.00	20-09-2019	- New
1.01	11-03-2020	- Add supported device. - Fixed incorrect DPI.
1.02	17-09-2020	- Add support device. - Add printer status value - Add RFID API
1.03	30-10-2020	- Add support device. - Information of the supported Windows version changed. - Fix PrintText parameter contents
1.04	09-02-2021	- Add support device.
1.05	19-08-2021	- Add support device. - 'Multicast Networking Entitlement Request' added
1.06	06-10-2021	- Add support device.
1.07	30-03-2022	- Add support device. - Supported methods for printing PDF files are added.
1.08	26-04-2022	- Add support device.
1.09	23-06-2022	- Add support device.
1.10	04-07-2022	- Modify supported methods for printing PDF files (removed setPdfLicenseKey method)
1.11	18-07-2022	- Add support device.
1.12	27-12-2023	- Add getSerialNumber API - Add USB Pripheral Reference
1.13	26-01-2024	- Added XQ-840II, XQ-843II
1.14	14-08-2024	- Added SRP-S300II
1.15	07-10-2024	- Added XD5-40IId/43IId/40IIt/43IIt
1.16	26-12-2024	- SRP-380II -> SRP-380plus change model name - SRP-382II -> SRP-382plus change model name
1.17	07-03-2025	- Added XT6-60,63,XM5-30