

CITIZEN

iOS POS Print SDK (Swift) Programing Manual

For Ver. 1.17

CITIZEN SYSTEMS JAPAN CO., LTD.

Revision Record

Date	Version	Description
2015/08/18	1.12	New issue
2015/12/10	1.13	Added the description of the development environment to the supported terminals. Added CT-S310II to the support models.
2016/01/29	1.14	Modified the description in the connect method, the disconnect method and the printerCheck method.
2016/05/23	1.15	Added the description to the section, "1.5 Definition method – Add the SDK to Xcode". Added the section "1.5 Definition method – Setting at the time of the archive file making".
2016/12/22	1.16	Added the section "2.38 MapMode property".
2017/04/14	1.17	Added CT-E351/CT-D150 to the support models. Added the setting contents of MSW13-1 and MSW13-5 to "CT-S601 / 651/801/851 / 601II / 651II / 801II / 851II Series Memory Switch Setting" in the section "1.4 Supported models".

Notes

1. Unauthorized use of all or any part of this document is prohibited.
2. The information in this document is subject to change without prior notice.
3. This document has been created with full attention. If, however, you find an error or question, please contact us.
4. We shall not be liable for any effect resulting from operation regardless of the above item 3.
5. If you do not agree with the above terms, you are not permitted to use this driver.

Trademark

iOS is registered trademarks of Cisco in the United States and/or other countries.

iPod touch, Swift, and Xcode is registered trademarks of Apple Inc. in the United States and/or other countries.

Company names and product names appearing on this document are trademarks and/or registered trademarks of respective companies.

CITIZEN is a registered trademark of Citizen Watch Co., Ltd.

Table of Contents

Revision Record	1
Notes.....	2
Trademark.....	2
Table of Contents.....	3
1. Introduction.....	5
1.1. Document target range	5
1.2. System summary.....	5
1.3. Supported terminals	5
1.4. Supported models.....	6
1.5. Definition method	11
1.6. Use of Bluetooth.....	15
1.7. Functions list.....	17
2. SDK interfaces.....	19
2.1. Return value.....	19
2.2. Instance	20
2.3. connect method.....	21
2.4. disconnect method.....	23
2.5. setEncoding method.....	24
2.6. printerCheck method	25
2.7. status method.....	26
2.8. printText method.....	27
2.9. printBitmap/printBitmapData method	28
2.10. setNVBitmap method	30
2.11. printNVBitmap method.....	32
2.12. printBarCode method	33
2.13. printPDF417 method	35
2.14. printQRCode method	36
2.15. printGS1DataBarStacked method.....	37
2.16. cutPaper method.....	38
2.17. unitFeed method	39
2.18. markFeed method	40
2.19. openDrawer method.....	41
2.20. transactionPrint method	42
2.21. rotatePrint method	43
2.22. pageModePrint method	44
2.23. clearPrintArea method	46
2.24. clearOutput method	47
2.25. printData method.....	48
2.26. printNormal method	49
2.27. getVersionCode method	50
2.28. getVersionName method.....	51
2.29. watermarkPrint method.....	52
2.30. searchCitizenPrinter method	53
2.31. searchESCPOSPrinter method	55
2.32. PageModeArea property	56
2.33. PageModePrintArea property	57
2.34. PageModePrintDirection property.....	58

2.35. PageModeHorizontalPosition property.....	59
2.36. PageModeVerticalPosition property.....	60
2.37. RecLineSpacing property.....	61
2.38. MapMode property.....	62
3. Notes.....	63
3.1. Function to detect the completion of printing.....	63

1. Introduction

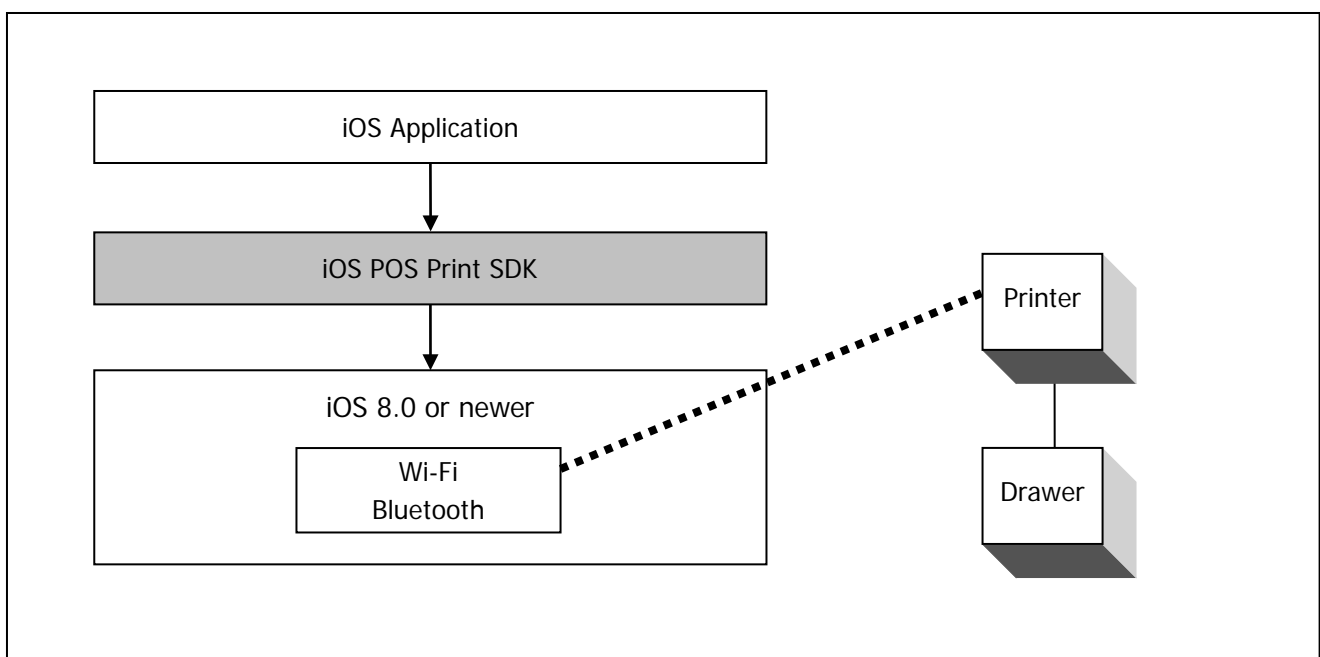
This document is a iOS POS Print SDK Programming Manual.

1.1. Document target range

This document is intended to be iOS application (Swift) developers to take advantage of CITIZEN POS printers.

1.2. System summary

This SDK assumes that it is referred from iOS application using CITIZEN POS printers.



System diagram of the SDK

1.3. Supported terminals

The specification of terminals supported by this SDK are as follows.

OS: iOS 8.0 or newer

Interface: Wi-Fi, Bluetooth

Development environment: Xcode7

1.4. Supported models

The models supported by this SDK and the corresponding interfaces are as listed below.
Refer to the user's manual of the printer for the detailed functions of each model.

Series of Model	Object Model	Interface	Printer Functions
CT-D150 Series	CT-D150	Wired LAN	Standard
CT-E351 Series	CT-D351	Wired LAN	Standard
CT-S251	CT-S251	Bluetooth Wired LAN Wireless LAN	Standard
CT-S281	CT-S281BD	Bluetooth	Standard
	CT-S281BD-M		Blackmark paper is supported.
	CT-S281BD-L		Label paper is supported.
CT-S310II	CT-S310II	Wired LAN	Standard
CT-S601/651/801/851	CT-S601/651/801/851	Wired LAN Wireless LAN	Standard
	CT-S801/851-M		Blackmark paper is supported.
	CT-S801-L		Label paper is supported.
CT-S601II/651II/801II/ 851II	CT-S601II/651II/801II /851II	Bluetooth Wired LAN Wireless LAN	Standard
	CT-S801II/851II-M		Blackmark paper is supported.
	CT-S801II-L		Label paper is supported.
CT-S2000	CT-S2000	Wired LAN	Standard
	CT-S2000-M		Blackmark paper is supported.
	CT-S2000-L		Label paper is supported.
CT-S4000	CT-S4000	Wired LAN	Standard (Paper with blackmark on front side is supported.)
	CT-S4000-M		Paper with blackmark on back side is supported.
	CT-S4000-L		Label paper is supported.

It is the prerequisite for the use of this SDK that the memory switch of the printer are set as listed below.

CT-D150 Series Memory Switch Setting

MSW No.	Function	Setting	Description
1-1	Power ON notify setting	OFF	Enabled
1-2	Input buffer	OFF	4KB
1-3	Busy condition	ON	Buffer full
1-4	Receiving error character	OFF	Character?
1-5	CR mode	OFF	Disabled
2-2	Auto cutter operation	ON	Enabled
2-4	Full Col Print	ON	Wait Data
3-1	Auto cutter restore	OFF	L/F enabled
3-7	CBM1000-compatible mode	ON	Compatible
3-8	Cover opened while printing	OFF	Auto restore
4-8	Forced partial	OFF	Disabled
5-2	Vertical basic calculation pitch	OFF	360 dots
5-3	USB Mode	ON	Printer Class
6-1	Act. For Driver	ON	Valid
7-6	DMA control	Enable	-
9-1	Code page	Katakana (*1)	-
9-2	Foreign language characters	Japan (*1)	-
9-3	Kanji	Enable (*1)	-
9-4	JIS/Shift-JIS	Shift-JIS (*1)	-

CT-E351 Series Memory Switch Setting

MSW No.	Function	Setting	Description
1-1	Power ON notify setting	OFF	Enabled
1-2	Input buffer	OFF	4KB
1-3	Busy condition	ON	Buffer full
1-4	Receiving error character	OFF	Character?
1-5	CR mode	OFF	Disabled
2-2	Auto cutter operation	ON	Enabled
2-4	Full Col Print	ON	Wait Data
3-1	Auto cutter restore	OFF	L/F enabled
3-7	CBM1000-compatible mode	ON	Compatible
3-8	Cover opened while printing	OFF	Auto restore
4-8	Forced partial	OFF	Disabled
5-2	Vertical basic calculation pitch	OFF	360 dots
5-3	USB Mode	ON	Printer Class
6-1	Act. For Driver	ON	Valid
7-6	DMA control	Enable	-
9-1	Code page	Katakana (*1)	-
9-2	Foreign language characters	Japan (*1)	-
9-3	Kanji	Enable (*1)	-
9-4	JIS/Shift-JIS	Shift-JIS (*1)	-

CT-S251 Series Memory Switch Setting

MSW No.	Function	Setting	Description
1-1	Power ON notify setting	OFF	Enabled
1-2	Input buffer	OFF	4KB
1-3	Busy condition	ON	Buffer full
1-4	Receiving error character	OFF	Character?
1-5	CR mode	OFF	Disabled
2-2	Auto cutter operation	ON	Enabled
2-4	Full Col Print	ON	Wait Data
3-1	Auto cutter restore	OFF	L/F enabled
3-7	CBM1000-compatible mode	ON	Compatible
3-8	Cover opened while printing	OFF	Auto restore
4-8	Forced partial	OFF	Disabled
5-2	Vertical basic calculation pitch	OFF	360 dots
6-1	Act. For Driver	ON	Valid
7-6	DMA control	Enable	-
9-1	Code page	Katakana (*1)	-
9-2	Foreign language characters	Japan (*1)	-
9-3	Kanji	Enable (*1)	-
9-4	JIS/Shift-JIS	Shift-JIS (*1)	-
13-1	Security / Connect Device	Low Follow / All	-
13-5	BT Device Scan	Discoverable	-
13-6	Auto Reconnect (When Bluetooth I/F is used)	Valid	-

CT-S281 Series Memory Switch Setting

MSW No.	Function	Setting	Description
1-1	Power ON notify setting	OFF	Enabled
1-2	Input buffer	OFF	4KB
1-3	Busy condition	ON	Buffer full
1-4	Receiving error character	OFF	Character?
1-5	CR mode	OFF	Disabled
2-2	Auto cutter operation	ON	Enabled
2-4	Full column print	ON	Wait data

MSW No.	Function	Setting	Description
3-1	Auto cutter restore	OFF	L/F enabled
3-7	CBM270-compatible mode	ON	compatible
3-8	Cover opened while printing	OFF	Auto restore
4-8	Full cut command	OFF	Full cut
9-1	Code page	Katakana (*1)	-
9-2	Foreign language characters	Japan (*1)	-
9-3	Kanji	Enabled (*1)	-
9-4	JIS/Shift-JIS	Shift-JIS (*1)	-
13-6	Auto Reconnect (When Bluetooth I/F is used)	Valid	-

CT-S310II Series Memory Switch Setting

MSW No.	Function	Setting	Description
1-1	Power ON notify setting	OFF	Enabled
1-2	Input buffer	OFF	4KB
1-3	Busy condition	ON	Buffer full
1-4	Receiving error character	OFF	Character?
1-5	CR mode	OFF	Disabled
2-2	Auto cutter operation	ON	Enabled
2-4	Full Col Print	ON	Wait Data
3-1	Auto cutter restore	OFF	L/F enabled
3-7	CBM1000-compatible mode	ON	Compatible
3-8	Cover opened while printing	OFF	Auto restore
4-8	Forced partial	OFF	Disabled
5-2	Vertical basic calculation pitch	OFF	360 dots
6-1	Act. For Driver	ON	Valid
7-6	DMA control	Enable	-
9-1	Code page	Katakana (*1)	-
9-2	Foreign language characters	Japan (*1)	-
9-3	Kanji	Enable (*1)	-
9-4	JIS/Shift-JIS	Shift-JIS (*1)	-

CT-S601/651/801/851 Series Memory Switch Setting

MSW No.	Function	Setting	Description
1-1	Power ON notify setting	OFF	Enabled
1-2	Input buffer	OFF	4KB
1-3	Busy condition	ON	Buffer full
1-4	Receiving error character	OFF	Character?
1-5	CR mode	OFF	Disabled
2-2	Auto cutter operation	ON	Enabled
2-4	Full Col Print	ON	Wait Data
3-1	Auto cutter restore	OFF	L/F enabled
3-3	Parallel 31Pin	OFF	Reset
3-7	CBM1000-compatible mode	ON	Compatible
3-8	Cover opened while printing	OFF	Auto restore
4-8	Forced partial	OFF	Disabled
5-2	Vertical basic calculation pitch	OFF	360 dots
6-1	Act. For Driver	ON	Valid
7-6	DMA control	Enable	-
9-1	Code page	Katakana (*1)	-
9-2	Foreign language characters	Japan (*1)	-
9-3	Kanji	Enable (*1)	-
9-4	JIS/Shift-JIS	Shift-JIS (*1)	-
10-3	ACK output timing	Before BUSY	-

CT-S601/651/801/851/601II/651II/801II/851II Series Memory Switch Setting

MSW No.	Function	Setting	Description
1-1	Power ON notify setting	OFF	Enabled
1-2	Input buffer	OFF	4KB
1-3	Busy condition	ON	Buffer full
1-4	Receiving error character	OFF	Character?
1-5	CR mode	OFF	Disabled
2-2	Auto cutter operation	ON	Enabled
2-4	Full Col Print	ON	Wait Data
3-1	Auto cutter restore	OFF	L/F enabled
3-3	Parallel 31Pin	OFF	Reset
3-7	CBM1000-compatible mode	ON	Compatible
3-8	Cover opened while printing	OFF	Auto restore
4-8	Forced partial	OFF	Disabled
5-2	Vertical basic calculation pitch	OFF	360 dots
6-1	Act. For Driver	ON	Valid
7-6	DMA control	Enable	-
9-1	Code page	Katakana (*1)	-
9-2	Foreign language characters	Japan (*1)	-
9-4	Multi-byte Char (*2)	Japanese: SJIS(CP932) Chinese: GB18030 Korean: EUC Hangul Taiwanese: BIG5	-
10-3	ACK output timing	Before BUSY	-
13-1	Security / Connect Device	Low Follow / All	-
13-5	BT Device Scan	Discoverable	-
13-6*	Auto Reconnect (When Bluetooth I/F is used)	Valid	-

CT-S2000 Series Memory Switch Setting

MSW No.	Function	Setting	Description
1-1	Power ON notify setting	OFF	Enabled
1-2	Input buffer	OFF	4KB
1-3	Busy condition	ON	Buffer full
1-4	Receiving error character	OFF	Character?
1-5	CR mode	OFF	Disabled
2-2	Auto cutter operation	ON	Enabled
2-4	Full Col Print	ON	Wait Data
3-1	Auto cutter restore	OFF	L/F enabled
3-3	Parallel 31Pin	OFF	Reset
3-7	CBM1000-compatible mode	ON	Compatible
3-8	Cover opened while printing	OFF	Auto restore
4-8	Forced partial	OFF	Disabled
5-2	Vertical basic calculation pitch	OFF	360 dots
7-6	DMA control	Enable	-
9-1	Code page	Katakana (*1)	-
9-2	Foreign language characters	Japan (*1)	-
9-3	Kanji	Enable (*1)	-
9-4	JIS/Shift-JIS	Shift-JIS (*1)	-
10-3	ACK output timing	Before BUSY	-

CT-S4000 Series Memory Switch Setting

MSW No.	Function	Setting	Description
1-1	Power ON notify setting	OFF	Enabled
1-2	Input buffer	OFF	4KB
1-3	Busy condition	ON	Buffer full
1-4	Receiving error character	OFF	Character?
1-5	CR mode	OFF	Disabled
2-2	Auto cutter operation	ON	Enabled
2-4	Full Col Print	ON	Wait Data
3-1	Auto cutter restore	OFF	L/F enabled
3-3	Parallel 31Pin	OFF	Reset
3-7	CBM1000-compatible mode	ON	Compatible

MSW No.	Function	Setting	Description
3-8	Cover opened while printing	OFF	Auto restore
4-8	Forced partial	OFF	Disabled
5-2	Vertical basic calculation pitch	OFF	360 dots
7-6	DMA control	Enable	-
9-1	Code page	Katakana (*1)	-
9-2	Foreign language characters	Japan (*1)	-
9-3	Kanji	Enable (*1)	-
9-4	JIS/Shift-JIS	Shift-JIS (*1)	-
10-3	ACK output timing	Before BUSY	-

*1 MSW No.9-1~4 is the setting when using Japanese. Please change it by use environment.

*2 The CT-S601II/651II/801II/851II series can change the Multi-byte character to Japanese, Chinese, Korean and Taiwanese. Please change it by use environment.

Firmware

The firmware version of the printer has to be the following condition to use of this SDK in CT-S601/651/801/851 Series.

With the older printer than following, it is necessary to update the firmware.

Model	Firmware Version
CT-S601	DL00-2000 or newer
CT-S651	DM00-2000 or newer
CT-S801	DH00-2000 or newer
CT-S851	DK00-2000 or newer

1.5. Definition method

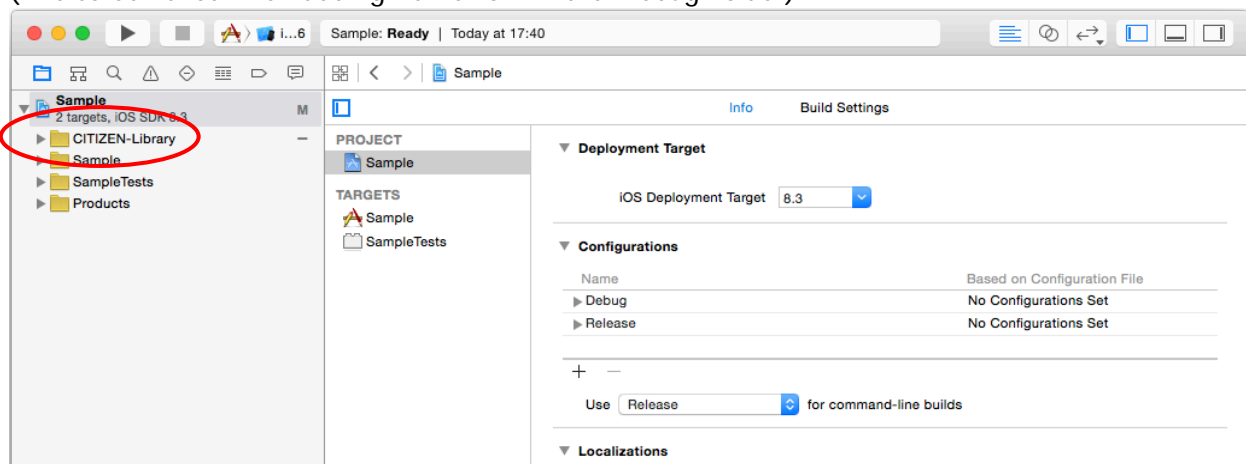
Add the SDK to Xcode

Add the framework file (CSJPOSLibSwift.framework) to the project to develop.

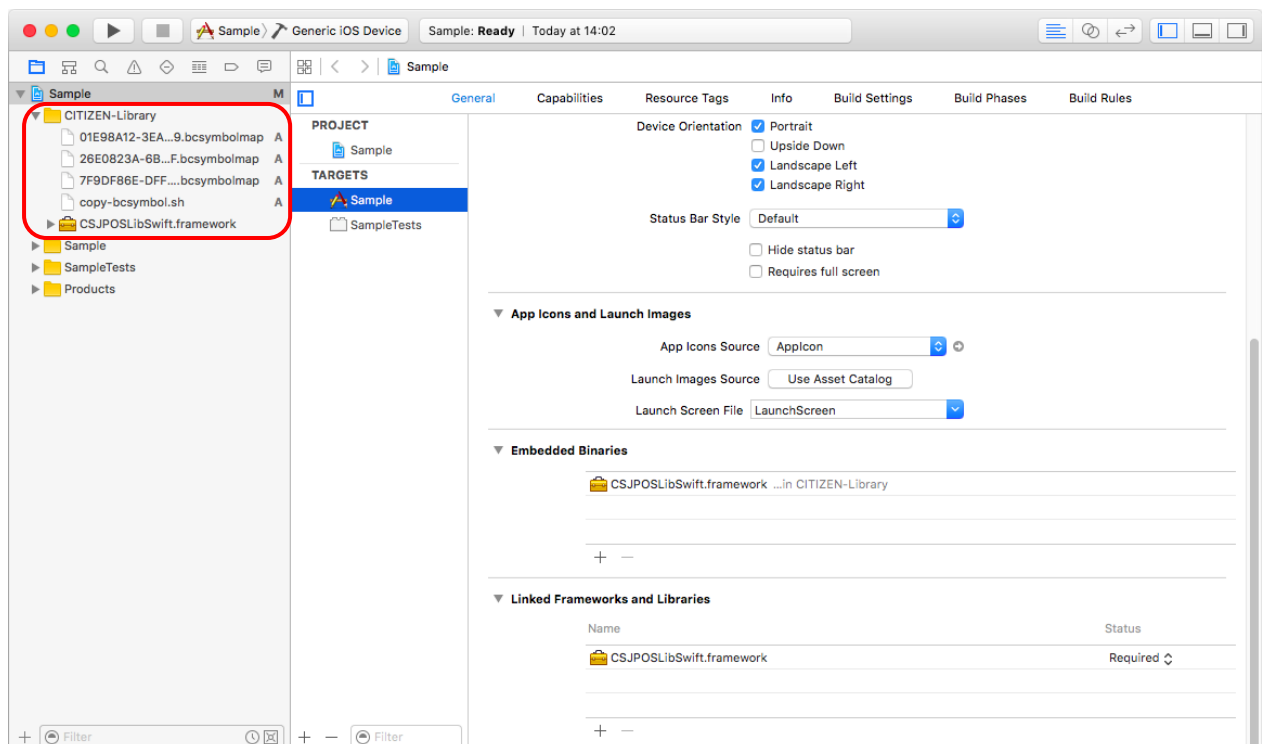
Framework files are stored is divided into 'Debug' folder and 'Release' folder. Since the framework file in the 'Debug' folder contains the binary for the iPhone-simulator, please use this at the time of application development. The framework file in the 'Release' folder does not contain the binary for the simulator. Please use this at the time of archive file making to submit to iTunes Connect.

Drag the framework file from the 'Finder' to any place in the "Project Navigator" of Xcode.(because it is contained in the folder named 'CITIZEN-Library', drag this folder every.) In the 'Release' folder, three bcsymbolmap files and one script files (copy-bcsymbol.sh) other than 'CSJPOSLibSwift.framework' are contained. Please copy these files to the same place as 'CSJPOSLibSwift.framework' from the 'Finder'.

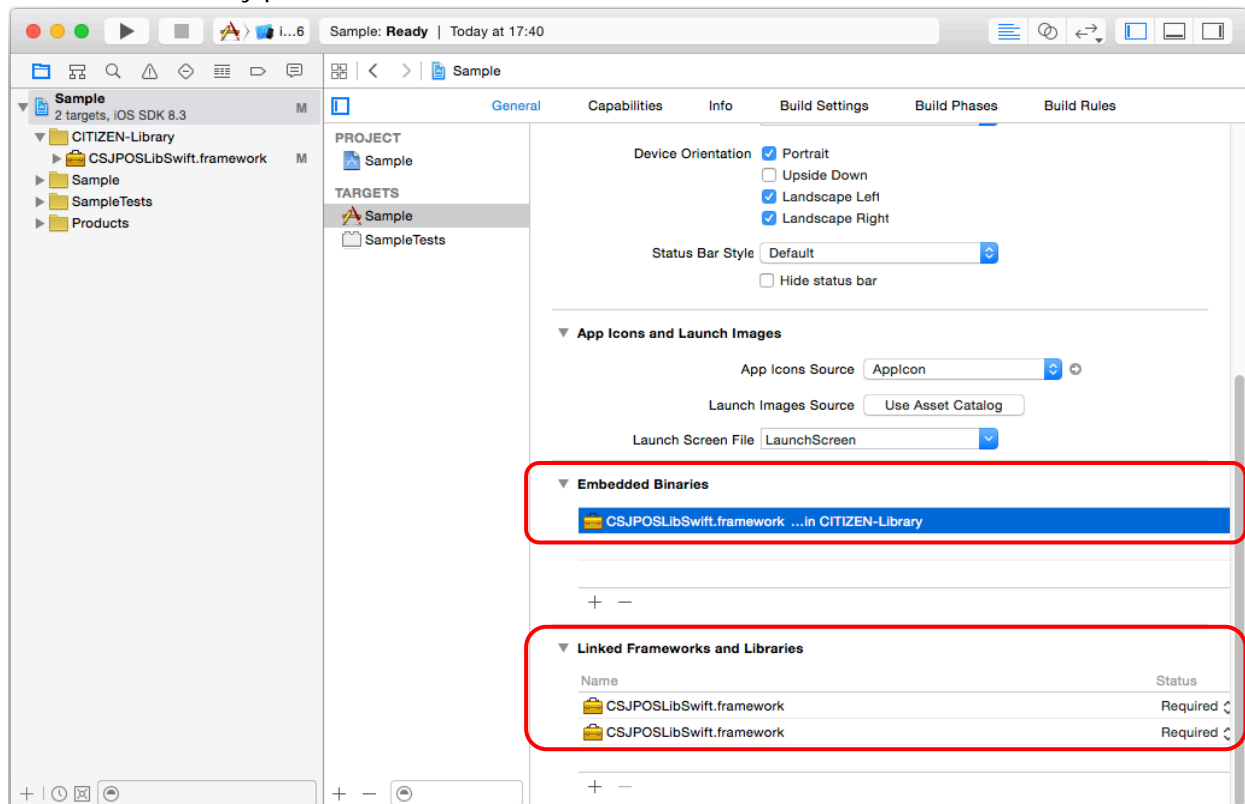
(The screen shot when adding framework in the 'Debug' folder)



(The screen shot when adding framework in the 'Release' folder)



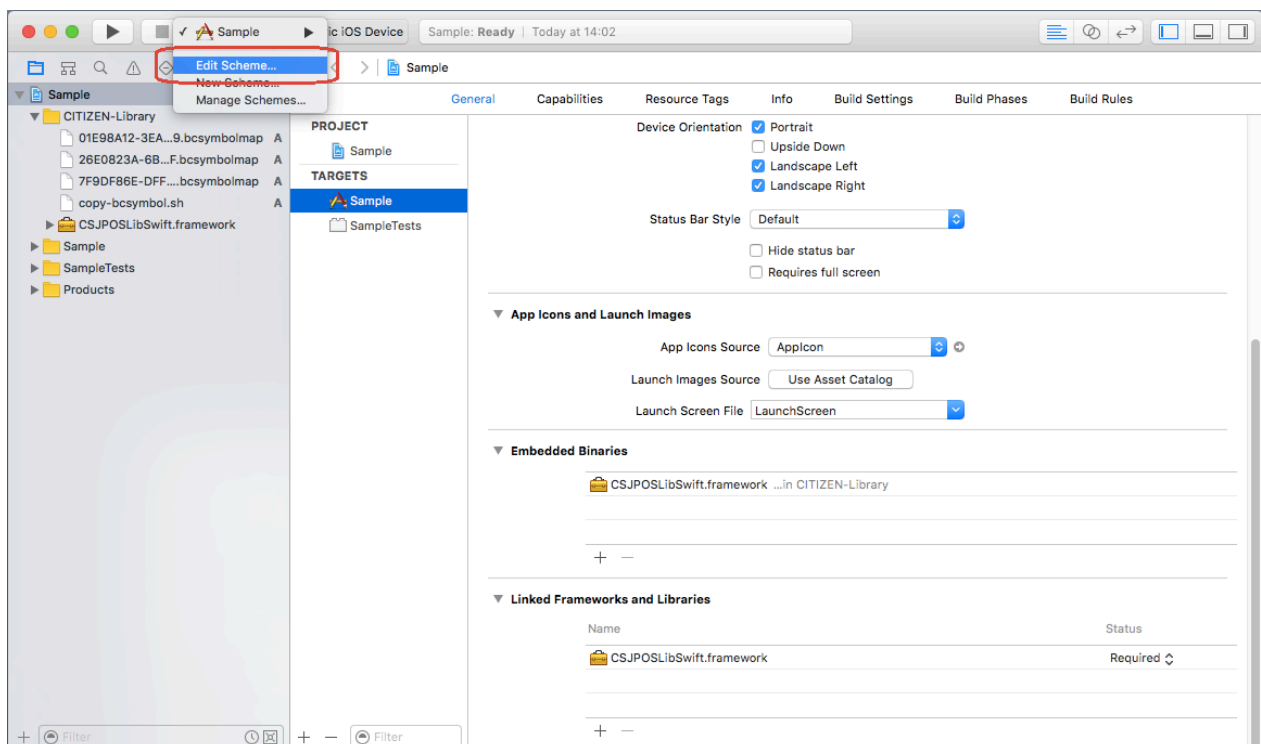
The Project file is selected from "Project Navigator", and it clicks 'Targets' -> 'General'. And expand the items in the "Embedded Binaries", and add the 'CSJPOSLibSwift.framework' by pressing the '+' button. If two 'libCSJTSOLib_iOS.framework' was added to "Linked Frameworks and Libraries" after addition, select one and delete it by push the '-' button.



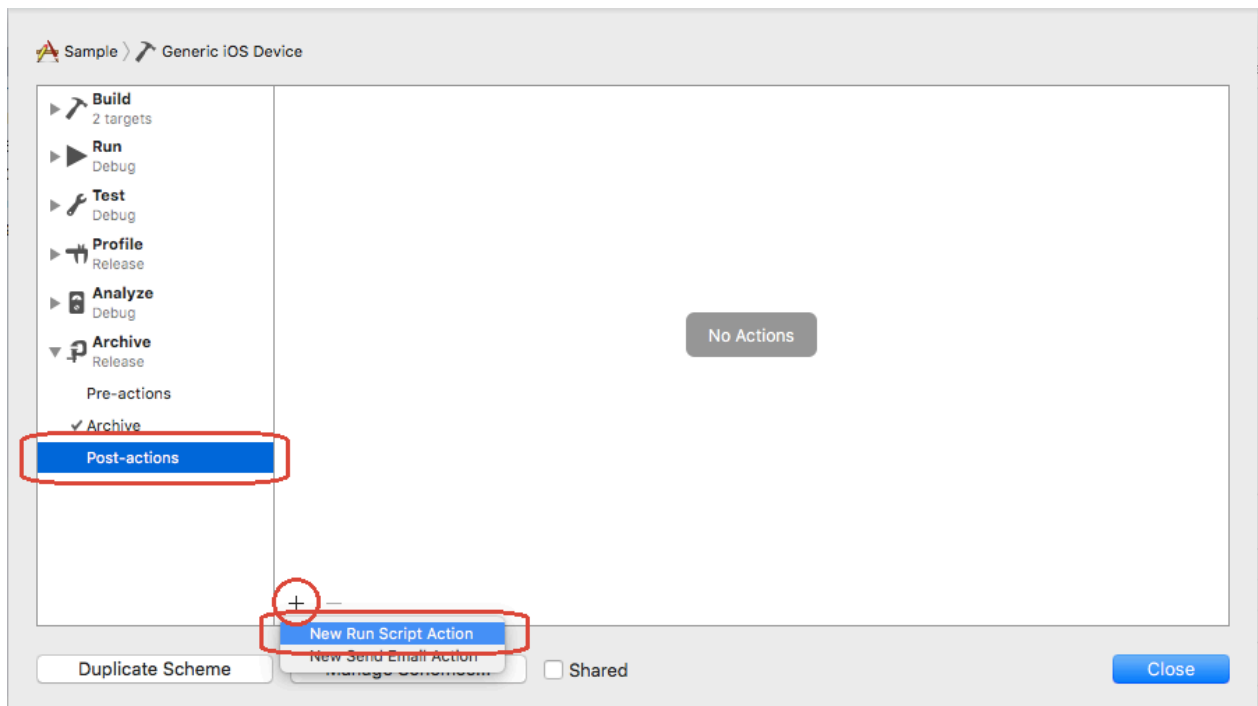
Setting at the time of the archive file making

Add a script to run when the archive.

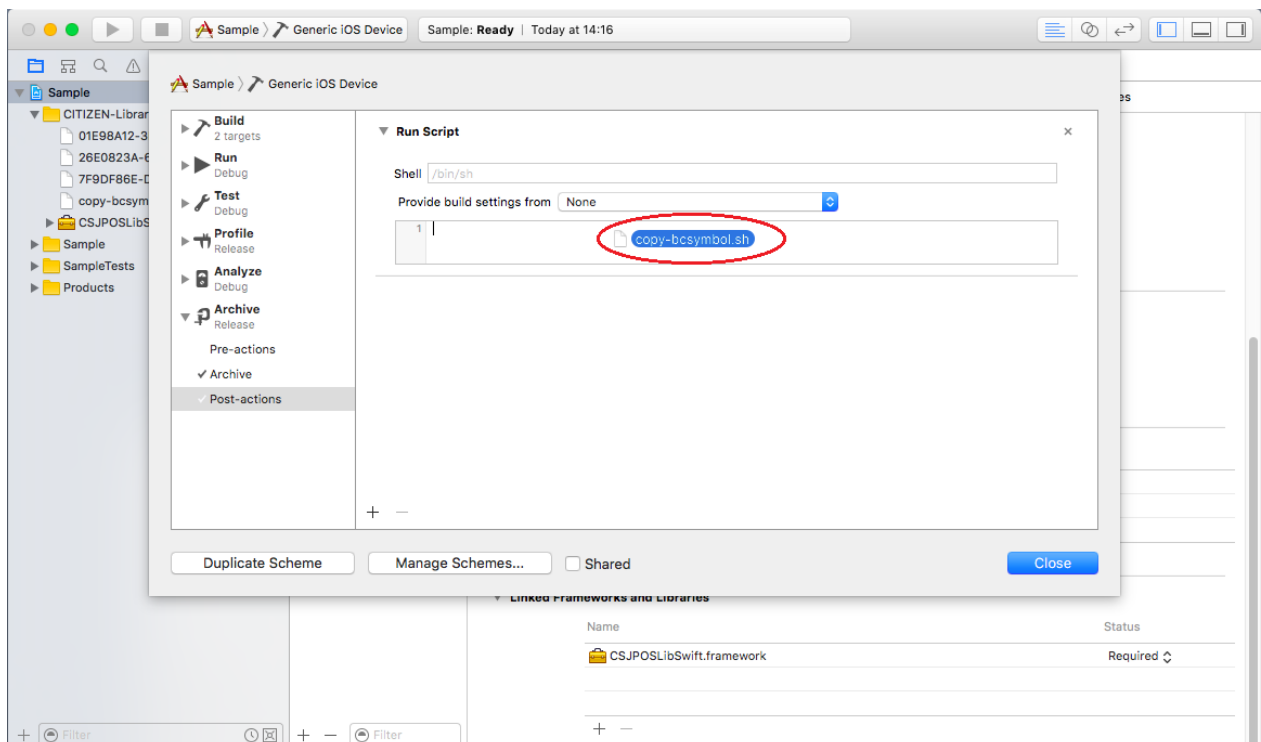
Select the "Edit Scheme".



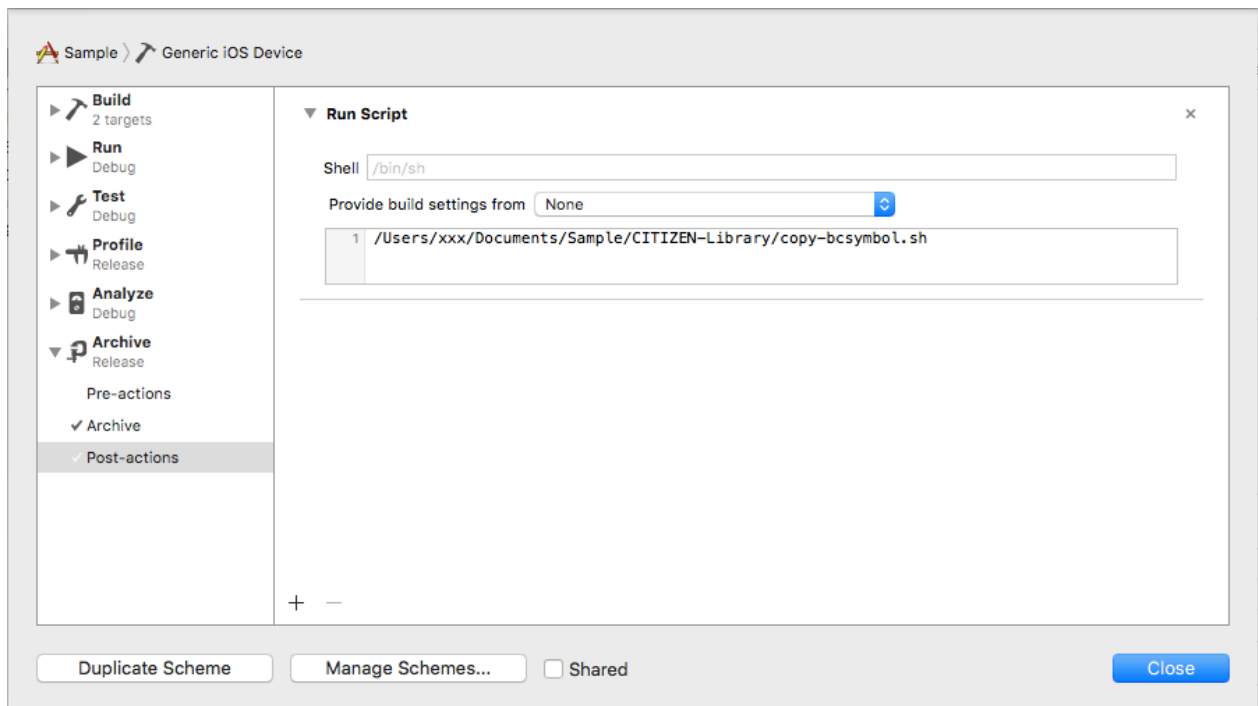
Select 'Archive' -> 'Post-actions': Press the bottom left of the '+' and then select the "New Run Script Action".



Drag the script file (copy-bcsymbol.sh) previously described from the 'Finder' like screen shots.

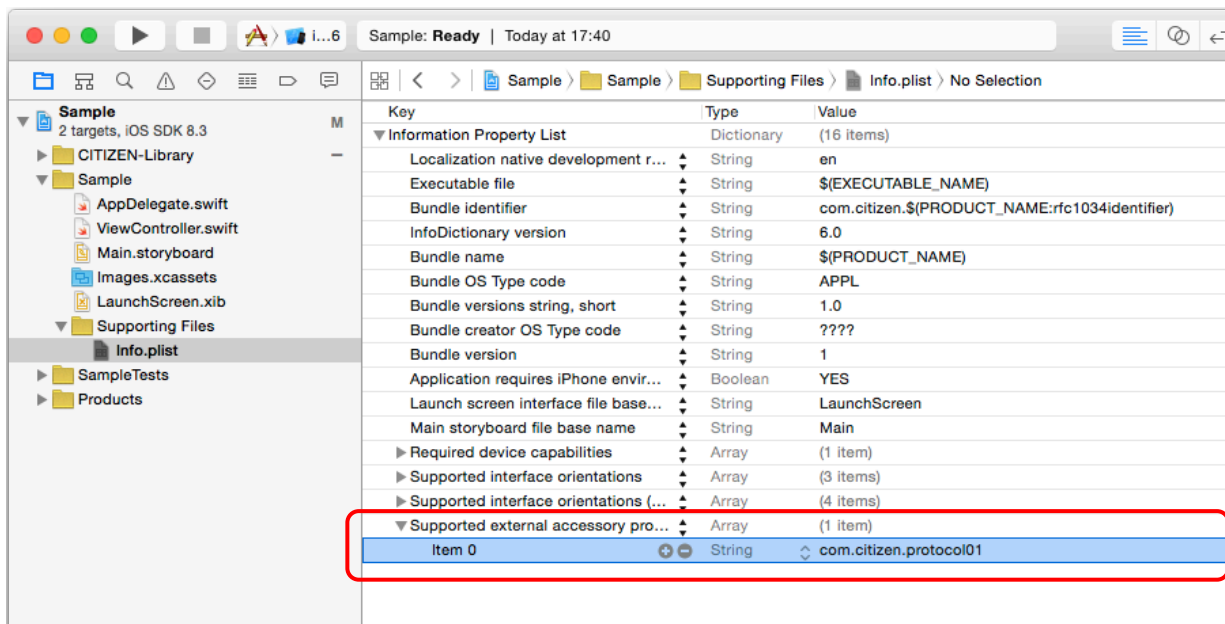


Press the 'Close' button to complete the setting.



Set the development of Bluetooth

Protocol name is added. "Info.plist" file is opened from "Project Navigator". "Add Row" is chosen from the pop-up menu (Open when Control + click), "Supported external accessory protocols" is selected and added. After that, "Supported external accessory protocols" is opened, and "com.citizen.protocol01" is input to the column of "Item 0".



It is completion in the above.

1.6. Use of Bluetooth

Pairing and connection of Bluetooth

To communicate the iOS device with the printer in Bluetooth, it is necessary to discover the printer from the iOS device and to do a pairing. (The following screenshot are examples in iPod Touch/iOS7.)

The power supply of the printer is turned on. 'Bluetooth' is chosen from the 'Settings' of the iOS device, and it looks for a peripheral Bluetooth devices. It begins to look for when turning it on when Bluetooth is off.



Peripheral Bluetooth devices are displayed after a while. The device displayed as "CITIZEN SYSTEMS" is a printer. The pairing is necessary when using it for the first time. The connection establishes the printer that the pairing has already been done at once.*

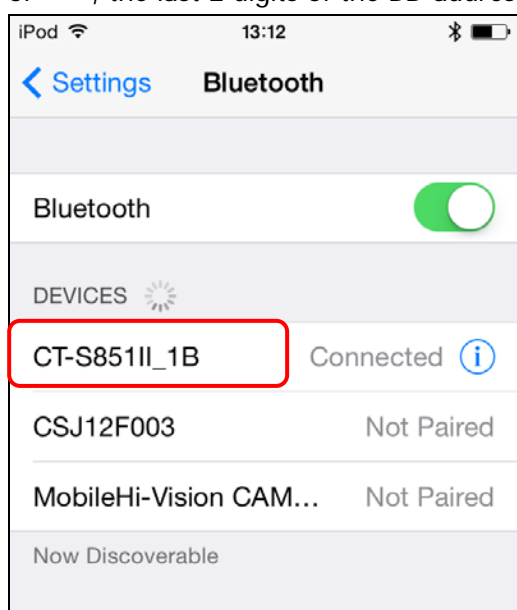


To do the pairing, the tap does the part of "CITIZEN SYSTEMS-Not Paired". The connection establishes it after a while.



* In the case of using CT-S251/601II/651II/801II/851II

Bluetooth device name is displayed in "Model Name_**" without in "CITIZEN SYSTEMS". As for the part of "**", the last 2 digits of the BD address is displayed.



*About Auto Reconnect

The connection from the printer side to the iOS device is recovered for memory switch 13-6 "Auto Reconnect" of the printer. It is necessary to connect it from the "Settings – Bluetooth" of iOS device every time the Bluetooth is disconnected for the specification of iOS. To request the connection automatically to the iOS device connected at the end from the printer side when this function is valid, the connection on the iOS device becomes unnecessary.

Please be invalid "Auto Reconnect" because this function decreases convenience when two or more iOS devices and printers are alternately used.

Please be invalid "Auto Reconnect" when it is connected with the device of other OS.

1.7. Functions list

This SDK provides the following functions.

Methods list

No	Function	Detail
1	Create class (instance)	This is instance method.
2	Connect printer (connect method)	Connect to the printer.
3	Disconnect printer (disconnect method)	Disconnect the printer connection.
4	Set encoding (setEncoding method)	Set the encoding of character.
5	Check printer status (printerCheck method)	Sends command for status check of the printer.
6	Get printer status (status method)	Get the status of the printer.
7	Print text (printText method)	Prints text data.
8	Print bitmap (printBitmap/printBitmapData method)	Prints a bitmap file. (BMP/JPG/PNG/GIF format)
9	Store NV bitmap (setNVBitmap method)	Stores a bitmap image in the flash memory.
10	Print NV bitmap (printNVBitmap method)	Prints a bitmap image that is stored in the flash memory.
11	Print BarCode (printBarCode method)	Prints a one-dimensional barcode.
12	Print PDF-417 (printPDF417 method)	Prints a PDF417 barcode.
13	Print QRcode (printQRCode method)	Prints a QRCode barcode.
14	Print 2D GS1DataBar (printGS1DataBarStacked method)	Prints a 2-dimensional GS1DataBar barcode.
15	Cut paper (cutPaper method)	Cuts the paper.
16	Feed dot units (unitFeed method)	Feeds the paper forward by dot units.
17	Feed mark (markFeed method)	Support for label / black mark paper.
18	Open drawer (openDrawer method)	Opens the drawer.
19	Transaction print (transactionPrint method)	Enters or exits transaction mode.
20	Rotate print (rotatePrint method)	Enters or exits rotated print mode. (180°)
21	PageMode print (pageModePrint method)	Enters or exits page mode.
22	PageMode clear print area (clearPrintArea method)	Clear the area of the page mode print area.
23	Clear output data (clearOutput method)	Clears all buffered output data. (data and printer buffer)
24	Output data (printData method)	Sends to the printer without changing the data.
25	Print OPOS format (printNormal method)	Prints text using OPOS escape sequences.

26	Get version code (getVersionCode method)	Get a numerical value for the version number of this SDK.
27	Get version name (getVersionName method)	Get a string for the version number of this SDK.
28	Watermark print (watermarkPrint method)	Enters or exits watermark print mode.
29	Search printer (searchCitizenPrinter method)	Search the printer and get the list of the printer information.
30	Search printer (searchESCPOSPrinter method)	Search the printer and get the list of addresses.

Properties List

No	Function	Attribute	Detail
1	PageMode area (PageModeArea property)	R	Shows the page area of page mode.
2	PageMode print area (PageModePrintArea property)	R/W	Shows the print area of page mode.
3	PageMode print direction (PageModePrintDirection property)	R/W	Shows the print direction of page mode.
4	PageMode horizontal position (PageModeHorizontalPosition property)	R/W	Shows the horizontal start position offset within the print area of page mode.
5	PageMode vertical position (PageModeVerticalPosition property)	R/W	Shows the vertical start position offset within the print area of page mode.
6	Line spacing (RecLineSpacing property)	R/W	Shows the spacing of each single-high print line.
7	Mapping mode (MapMode property)	R/W	Show the mapping mode (the unit of measure) of the printer.

2. SDK interfaces

The following are the interfaces of this SDK.

2.1. Return value

Methods to be described later return the value in the list below.

Return value	Description
CMP_SUCCESS (0)	The operation is success.
CMP_E_CONNECTED (1001)	The printer is already connected.
CMP_E_DISCONNECT (1002)	The printer is not connected.
CMP_E_NOTCONNECT (1003)	Failed connection to the printer.
CMP_E_CONNECT_NOTFOUND (1004)	Failed to check the support model after connecting to the device.
CMP_E_CONNECT_OFFLINE (1005)	Failed to check the printer status after connecting to the device.
CMP_E_ILLEGAL (1101)	Unsupported operation with the Device, or an invalid parameter value was used.
CMP_E_OFFLINE (1102)	The printer is off-line.
CMP_E_NOEXIST (1103)	The file name does not exist.
CMP_E_FAILURE (1104)	The Service cannot perform the requested procedure.
CMP_E_TIMEOUT (1105)	The Service timed out waiting for a response from the printer.
CMP_E_NO_LIST (1106)	The printer cannot be found in the printer search.
CMP_EPTR_COVER_OPEN (1201)	The cover of the printer opens.
CMP_EPTR_REC_EMPTY (1202)	The printer is out of paper.
CMP_EPTR_BADFORMAT (1203)	The specified file is in an unsupported format.
CMP_EPTR_TOOBIG (1204)	The specified bitmap is either too big.

2.2. Instance

Syntax

ESCPOSPrinter

Description

To generate the class and to initialize it.

Example

```
var escp: ESCPOSPrinter? = CSJPOSLibSwift.ESCPOSPrinter()
```

2.3. connect method

Syntax

- 1) func connect(connectType: Int32, withAddress addr: String?) -> Int32
- 2) func connect(connectType: Int32, withAddress addr: String?, withPort port: Int32) -> Int32
- 3) func connect(connectType: Int32, withAddress addr: String?, withPort port: Int32, withTimeout timeout: Int32) -> Int32

Parameter

The meaning and the setting range of the parameters are as follows.

Value	[IN/OUT]	Meaning	Setting range
connectType	[IN]	Connect type	CMP_PORT_WIFI: Wi-Fi connection CMP_PORT_BLUETOOTH: Bluetooth connection
addr	[IN]	IP address to connect or Bluetooth device address or Bluetooth device name	WiFi: 0.0.0.0 - 255.255.255.255 Bluetooth: 00:00:00:00:00:00 – FF:FF:FF:FF:FF:FF Device name
port	[IN]	Connection port number	
timeout	[IN]	Timeout (msec)	

Description

This method is used to connect the printer. Please specify the type and address of the printer connection.

When Bluetooth is used, it is possible to connect with only the printer with which pairing and connecting by the BLUETOOTH Setting of the iOS device (Please refer to "1.6 Use of Bluetooth"). And, it should be iOS6 or more to be connected by using the Bluetooth device address. Please connect by using the Bluetooth device name besides.

Connection port number is valid only if Wi-Fi is specified for the connection type. If it is omitted, it connects with number 9100.

Timeout is gives the maximum number of milliseconds to connect printer. If it is omitted, it connects with 4000 milliseconds in the case of Wi-Fi and 8000 milliseconds in the case of Bluetooth.

The recommended timeout is more than 8000 milliseconds in the case of Bluetooth.

When connecting to the printer, this SDK also checks the status of the printer and the supporting models.

When communication with the printer is not necessary, must execute the [disconnect method](#) to disconnect the printer connection. When not disconnect, the next connection will be an error.

Return value

Return CMP_SUCCESS (0) in success. Please check the description of the error codes below in the case of failure. Please refer to "2.1 Return value" for the error code except it.

Error codes	Description
CMP_E_NOTCONNECT (1003)	Failed connection to the printer. (1) The printer is under none-connection status. (2) The printer is not turned ON. (3) Cannot obtain handle of interface board.
CMP_E_CONNECT_NOTFOUND (1004)	Failed to check the support model after connecting to the printer. (1) The model is not supported.
CMP_E_CONNECT_OFFLINE (1005)	Failed to check the printer status after connecting to the printer. The printer is connected but the following errors occurred.

	<ul style="list-style-type: none">(1) The cover of the printer opens.(2) The printer is out of paper.(3) Auto Cutter Error occurred due to paper jam, etc.(4) Unrecoverable error occurred due to circuit failure, etc.
--	--

Example

```
escp!.connect(CMP_PORT_WiFi, withAddress: "192.168.182.100", withPort: 9100)
escp!.connect(CMP_PORT_BLUETOOTH, withAddress: "00:01:90:F0:81:AB",
              withPort: 9100, withTimeout:8000)
escp!.connect(CMP_PORT_BLUETOOTH, withAddress: "CITIZEN SYSTEMS",
              withPort: 9100, withTimeout:8000)
```

2.4. disconnect method

Syntax

```
func disconnect() -> Int32
```

Parameter

Not exist.

Description

This method is used to disconnect the printer connection.

When the end of the print or some kind of errors occurs, please disconnect the connection by the execution of this method.

Return value

Return CMP_SUCCESS(0) in success. Please refer to "[2.1 Return value](#)" for the error code except it.

Example

```
escp!.disconnect()
```


2.5. setEncoding method

Syntax

```
func setEncoding(charset: NSStringEncoding) -> Int32
```

Parameter

The meaning and the setting range of the parameters are as follows.

Value	[IN/OUT]	Meaning	Setting range
Charset	[IN]	Character set name	

Description

This method is used to set the encoding of the send data to the printer.

When you create an instance, it is initialized to the default character set of the OS.

Please set the encoding by the setting of the memory switch of the printer. (Please refer to "[1.5 Supported models](#)")

When used in Japanese, it is necessary to specify the NSShiftJISStringEncoding.

Please refer to the following URL for the details of the setting range.

http://developer.apple.com/library/mac/#documentation/Cocoa/Reference/Foundation/Classes/NSStringClass/Reference/NSString.html#//apple_ref/doc/uid/20000154-BAJJAICE

Return value

Return CMP_SUCCESS (0) in success. Please refer to "[2.1 Return value](#)" for the error code except it.

Example

```
// Japanese
escp!.setEncoding(NSShiftJISStringEncoding)

// Chinese(Simplified Chinese)
let encode = CFStringConvertEncodingToNSStringEncoding( 0x0632 )
escp!.setEncoding(encode)

// Chinese(Traditional Chinese)
let encode = CFStringConvertEncodingToNSStringEncoding( 0x0A03 )
escp!.setEncoding(encode)

// Korean
let encode = CFStringConvertEncodingToNSStringEncoding( 0x0940 )
escp!.setEncoding(encode)
```

2.6. printerCheck method

Syntax

```
func printerCheck() -> Int32
```

Parameter

Not exist.

Description

This method is used to send the command to get the status of the printer.

If the result of this method is successful, you can get the status of the printer by [status method](#).

If the result of this method is failure, there is a possibility that the connection or the printer abnormality has occurred. In this case, please reconnect using the [disconnect method](#) and the [connect method](#).

If you want to print after the connected and some time passed, please check the status of the printer by the execution of this method and the [status method](#) beforehand.

In the case of network connection, it is automatically disconnected when passed a long time. If you want to keep a connection, please execute this method regularly.

Return value

Return CMP_SUCCESS (0) in success. Please refer to "[2.1 Return value](#)" for the error code except it.

Example

```
if CMP_SUCCESS == escp!.printerCheck() {  
    // Success  
} else {  
    // Fail  
}
```

2.7. status method

Syntax

func status() -> Int32

Parameter

Not exist.

Description

This method is used to get the status of the printer obtained by the printerCheck method.
Before the execution of this method, you must run the printerCheck method.

Return value

Return the following status codes.

status codes	Description
CMP_STS_NORMAL (0)	The printer is normal.
CMP_STS_COVER_OPEN (16)	The cover of the printer opens.
CMP_STS_PAPER_EMPTY (32)	The printer is out of paper.
CMP_STS_PRINTEROFF (128)	The printer is off-line.

Example

```
var status = escp!.status()
if CMP_STS_NORMAL == status {
    // No Error
} else {
    if (CMP_STS_COVER_OPEN & status) > 0 {
        // Cover Open
    }
    if (CMP_STS_PAPER_EMPTY & status) > 0 {
        // Paper Empty
    }
    if (CMP_STS_PRINTEROFF & status) > 0 {
        // Printer Offline
    }
}
```

2.8. printText method

Syntax

```
func printText(data: String?, withAlignment alignment: Int32, withAttribute attribute: Int32,
               withTextSize textSize: Int32) -> Int32
```

Parameter

The meaning and the setting range of the parameters are as follows.

Value	[IN/OUT]	Meaning	Setting range
data	[IN]	Text data	
alignment	[IN]	Text alignment	CMP_ALIGNMENT_LEFT: Left alignment CMP_ALIGNMENT_CENTER: Center alignment CMP_ALIGNMENT_RIGHT: Right alignment
attribute	[IN]	Text attribute	CMP_FNT_DEFAULT: Default font CMP_FNT_FONTB: Font B CMP_FNT_FONTC: Font C CMP_FNT_BOLD: Bold CMP_FNT_REVERSE: Reverse CMP_FNT_UNDERLINE: Underline
textSize	[IN]	Text size	CMP_TXT_1WIDTH: 1 times width CMP_TXT_2WIDTH: 2 times width CMP_TXT_3WIDTH: 3 times width CMP_TXT_4WIDTH: 4 times width CMP_TXT_5WIDTH: 5 times width CMP_TXT_6WIDTH: 6 times width CMP_TXT_7WIDTH: 7 times width CMP_TXT_8WIDTH: 8 times width CMP_TXT_1HEIGHT: 1 times height CMP_TXT_2HEIGHT: 2 times height CMP_TXT_3HEIGHT: 3 times height CMP_TXT_4HEIGHT: 4 times height CMP_TXT_5HEIGHT: 5 times height CMP_TXT_6HEIGHT: 6 times height CMP_TXT_7HEIGHT: 7 times height CMP_TXT_8HEIGHT: 8 times height

Description

This method is used to print text which specifies alignment and attribute and size.

Text attribute can be specified in combination font B, font C, bold, reverse, and underline. If you want to combine, please specify the logical sum.

Text size can be specified in combination with the width and height. If you want to combine, please specify the logical sum.

Return value

Return CMP_SUCCESS (0) in success. Please refer to "[2.1 Return value](#)" for the error code except it.

Example

```
escp!.printText("Print text data.\n",
                withAlignment: CMP_ALIGNMENT_CENTER,
                withAttribute: CMP_FNT_BOLD,
                withTextSize: CMP_TXT_2WIDTH | CMP_TXT_2HEIGHT)
```

2.9. printBitmap/printBitmapData method

Syntax

- 1) func printBitmap(fileName: String?, withAlignment alignment: Int32) -> Int32
- 2) func printBitmap(fileName: String?, withWidth width: Int32, withAlignment alignment: Int32) -> Int32
- 3) func printBitmap(fileName: String?, withWidth width: Int32, withAlignment alignment: Int32, withMode mode: Int32) -> Int32
- 4) func printBitmapData(imageData: UIImage?, withAlignment alignment: Int32) -> Int32
- 5) func printBitmapData(imageData: UIImage?, withWidth width: Int32, withAlignment alignment: Int32) -> Int32
- 6) func printBitmapData(imageData: UIImage?, withWidth width: Int32, withAlignment alignment: Int32, withMode mode: Int32) -> Int32

Parameter

The meaning and the setting range of the parameters are as follows.

Value	[IN/OUT]	Meaning	Setting range
fileName	[IN]	Bitmap file name	
imageData	[IN]	Bitmap data	
width	[IN]	Bitmap width	CMP_BM_ASIS: Print the bitmap with one bitmap pixel per printer dot. Other Values: Bitmap width expressed in the unit of dot.
alignment	[IN]	Bitmap alignment	CMP_ALIGNMENT_LEFT: Left alignment CMP_ALIGNMENT_CENTER: Center alignment CMP_ALIGNMENT_RIGHT: Right alignment Other Values: Distance from the left-most print column to the start of the bitmap.
mode	[IN]	Bitmap mode	CMP_BM_MODE_HT_THRESHOLD: Halftone threshold CMP_BM_MODE_HT_DITHER: Halftone dither CMP_BM_MODE_CMD_RASTER: Monochrome raster command output CMP_BM_MODE_CMD_BITIMAGE: Monochrome bitimage command output CMP_BM_MODE_CMD_GRAY16 Grayscale (4bpp) output

Description

This method is used to print bitmap which specifies file name and width and alignment.

Printable bitmap formats are BMP / JPG / PNG / GIF.

If the bitmap width is omitted, printing in CMP_BM_ASIS.

Mode can be specified in combination with the halftone and output method. If you want to combine, please specify the logical sum. If mode is omitted, printed at CMP_BM_MODE_HT_THRESHOLD | CMP_BM_MODE_CMD_RASTER.

For more information on mode is as follows.

Halftone Specify the halftone treatment method.

Value	Description
CMP_BM_MODE_HT_THRESHOLD	Threshold Suitable for characters printing.
CMP_BM_MODE_HT_DITHER	Dither Suitable for graphics printing.

Output Specify the output method.

Value	Description
CMP_BM_MODE_CMD_RASTER	Monochrome raster command output Suitable for small data printing. In order to output the data collectively, there is a height limit (2,304 dots 28cm approximately).
CMP_BM_MODE_CMD_BITIMAGE	Monochrome bitimage command output Suitable for large data printing. In order to output the split data, there is no height limit.
CMP_BM_MODE_CMD_GRAY16	Grayscale(4bpp) output Available in CT-S251/601II/651II/801II/851II. Graphic can be printed more beautifully.

Return value

Return CMP_SUCCESS (0) in success. Please refer to ["2.1 Return value"](#) for the error code except it.

Example

- In the case of file designation in the resource


```
let FileName = NSBundle.mainBundle().pathForResource("sample_1.jpg", ofType:
    nil)
var errCode = escp!.printBitmap(FileName, withWidth: CMP_BM_ASIS,
    withAlignment: CMP_ALIGNMENT_LEFT)
```
- In the case of file designation in the images folder


```
// Get the documents folder path
let docDir = NSSearchPathForDirectoriesInDomains(.DocumentDirectory,
    .UserDomainMask, true)[0] as! String
// Get the images folder path
let image_path = docDir + "/Images"
let FileName = image_path + "/sample_1.jpg"
var errCode = escp!.printBitmap(FileName, withWidth: CMP_BM_ASIS,
    withAlignment: CMP_ALIGNMENT_LEFT)
```

2.10. setNVBitmap method

Syntax

- 1) func setNVBitmap(number: Int32, withFileName fileName: String?, withWidth width: Int32) -> Int32
- 2) func setNVBitmap(number: Int32, withFileName fileName: String?, withWidth width: Int32, withMode mode: Int32) -> Int32

Parameter

The meaning and the setting range of the parameters are as follows.

Value	[IN/OUT]	Meaning	Setting range
number	[IN]	Number of bitmap to store in the flash memory of the printer	1 - 20
fileName	[IN]	File name of bitmap to store	
width	[IN]	Width of bitmap to store	CMP_BM_ASIS: Store the bitmap with one bitmap pixel per printer dot. Other Values: Bitmap width expressed in the unit of dot.
mode	[IN]	Bitmap mode	CMP_BM_MODE_HT_THRESHOLD: Halftone threshold CMP_BM_MODE_HT_DITHER: Halftone dither CMP_BM_MODE_CMD_MONO Monochrome storing CMP_BM_MODE_CMD_GRAY16 Grayscale (4bpp) storing

Description

This method is used to store bitmap which specifies number, file name, width, mode as parameters. The stored bitmap can print using [printNVBitmap method](#) or [watermarkPrint method](#).

The fileName parameter sets the full path of the bitmap file to store.

The bitmap formats that can be stored are BMP / JPG / PNG / GIF.

If the width parameter is omitted, it is in CMP_BM_ASIS to store.

The mode parameter can be specified in combination with the halftone and store method. To use of the combination, please specify the logical sum. If the mode parameter is omitted, it is in CMP_BM_MODE_HT_THRESHOLD | CMP_BM_MODE_CMD_MONO to store.

For more information on the mode parameter is as follows.

Halftone

Specify the halftone treatment method.

Value	Description
CMP_BM_MODE_HT_THRESHOLD	Threshold Suitable for characters printing.
CMP_BM_MODE_HT_DITHER	Dither Suitable for graphics printing.

Storing

Specify the storing method.

Value	Description
CMP_BM_MODE_CMD_GRAY16	Monochrome storing
CMP_BM_MODE_CMD_GRAY16	Grayscale storing Available in CT-S251/60111/65111/80111/85111.

	Graphic can be stored more beautifully.
--	---

Return value

Return CMP_SUCCESS (0) in success. Please refer to "[2.1 Return value](#)" for the error code except it.

Example

```
let FileName = NSBundle.mainBundle().pathForResource("sample_1.jpg", ofType:
    nil)
var errCode = escp!.setNVBitmap(1, withFileName: FileName, withWidth:
    CMP_BM_ASIS, withMode:
    CMP_BM_MODE_HT_DITHER | CMP_BM_MODE_CMD_GRAY16)
```


2.11. printNVBitmap method

Syntax

```
func printNVBitmap(nvImageNumber: Int32) -> Int32
```

Parameter

The meaning and the setting range of the parameters are as follows.

Value	[IN/OUT]	Meaning	Setting range
nvImageNumber	[IN]	Bitmap image number that is stored in the flash memory of the printer	1 - 20

Description

This method is used to print bitmap image (Logo) that is stored in the flash memory of the printer. To use this method, you need to register of the logo in advance. Logo registration, please store it using [setNVBitmap method](#) or use the "POS Printer utility" of utility software for the printer. Registration mode varies among the model of the printer. Please register as follows.

[CT-D150,CT-E351,CT-S251/310II/601/651/801/851/601II/651II/801II/851II/2000/4000 Series]

Please register the logo with "Key code mode".

To the image number to use, it is necessary to register the logo that specifies the key code.

The key code corresponding to the image number is as follows.

Image number	Key code (Characters)
1	"01"
2	"02"
3	"03"
.	.
.	.
.	.
19	"19"
20	"20"

Return value

Return CMP_SUCCESS (0) in success. Please refer to ["2.1 Return value"](#) for the error code except it.

Example

```
escp!.printNVBitmap(1)
```

2.12. printBarcode method

Syntax

```
func printBarcode(data: String?, withSymbology symbology: Int32, withHeight height: Int32, withWidth width: Int32, withAlignment alignment: Int32, withTextPosition textPosition: Int32) -> Int32
```

Parameter

The meaning and the setting range of the parameters are as follows.

Value	[IN/OUT]	Meaning	Setting range
data	[IN]	Barcode data	
symbology	[IN]	Barcode symbol type	CMP_BCS_UPCA: UPC-A CMP_BCS_UPCE: UPC-E CMP_BCS_EAN8: EAN8 (=JAN8) CMP_BCS_JAN8: JAN8 (=EAN8) CMP_BCS_EAN13: EAN13 (=JAN13) CMP_BCS_JAN13: JAN13 (=EAN13) CMP_BCS_ITF: Interleaved 2 of 5 CMP_BCS_Codabar: Codabar CMP_BCS_Code39: Code 39 CMP_BCS_Code93: Code 93 CMP_BCS_Code128: Code 128 CMP_BCS_GS1DATABAR: GS1 DataBar Omnidirectional CMP_BCS_GS1DATABAR_E: GS1 DataBar Expanded CMP_BCS_GS1DATABAR_T: GS1 DataBar Truncated CMP_BCS_GS1DATABAR_L: GS1 DataBar Limited
height	[IN]	Barcode height (dot)	
width	[IN]	Barcode horizontal size (magnification)	2 - 6
alignment	[IN]	Barcode alignment	CMP_ALIGNMENT_LEFT: Left alignment CMP_ALIGNMENT_CENTER: Center alignment CMP_ALIGNMENT_RIGHT: Right alignment Other Values: Distance from the left-most print column to the start of the barcode.
textPosition	[IN]	HRI characters position	CMP_HRI_TEXT_NONE: No printing CMP_HRI_TEXT_ABOVE: Above the barcode CMP_HRI_TEXT_BELOW: Below the barcode

Description

This method is used to print one-dimensional barcode.

GS1 DataBar (CMP_BCS_GS1DATABAR, CMP_BCS_GS1DATABAR_E, CMP_BCS_GS1DATABAR_T, CMP_BCS_GS1DATABAR_L) can use only the printers of CT-D150/CT-E351/CT-S251/310II/601/651/801/851/601II/651II/801II/851II series.

The designation of CMP_ALIGNMENT_CENTER and CMP_ALIGNMENT_RIGHT of the Barcode alignment on the page mode is ignored.

Return value

Return CMP_SUCCESS (0) in success. Please refer to "[2.1 Return value](#)" for the error code except it.

Example

```
escp!.printBarcode("123456789012", withSymbology: CMP_BCS_UPCA, withHeight:  
    50, withWidth: 2, withAlignment: CMP_ALIGNMENT_LEFT,  
    withTextPosition: CMP_HRI_TEXT_ABOVE)
```

2.13. printPDF417 method

Syntax

```
func printPDF417(data: String?, withDigits digits: Int32, withSteps steps: Int32, withModuleWidth moduleWidth: Int32, withStepHeight stepHeight: Int32, withECLevel ECLevel: Int32, withAlignment alignment: Int32) -> Int32
```

Parameter

The meaning and the setting range of the parameters are as follows.

Value	[IN/OUT]	Meaning	Setting range
data	[IN]	Barcode data	
digits	[IN]	Digits number	0: automatic 1 - 30
steps	[IN]	Steps number	0: automatic 3 - 90
moduleWidth	[IN]	Module width (dot)	2 - 8
stepHeight	[IN]	Height of step	2 - 8
ECLevel	[IN]	Error correction level	CMP_PDF417_EC_LEVEL_0: Level 0 CMP_PDF417_EC_LEVEL_1: Level 2 CMP_PDF417_EC_LEVEL_2: Level 2 CMP_PDF417_EC_LEVEL_3: Level 3 CMP_PDF417_EC_LEVEL_4: Level 4 CMP_PDF417_EC_LEVEL_5: Level 5 CMP_PDF417_EC_LEVEL_6: Level 6 CMP_PDF417_EC_LEVEL_7: Level 7 CMP_PDF417_EC_LEVEL_8: Level 8
alignment	[IN]	Barcode alignment	CMP_ALIGNMENT_LEFT: Left alignment CMP_ALIGNMENT_CENTER: Center alignment CMP_ALIGNMENT_RIGHT: Right alignment Other Values: Distance from the left-most print column to the start of the barcode.

Description

This method is used to print PDF-417 barcode.

Please refer to the Command Reference of the printer for details on each parameter.

The designation of CMP_ALIGNMENT_CENTER and CMP_ALIGNMENT_RIGHT of the Barcode alignment on the page mode is ignored.

Return value

Return CMP_SUCCESS (0) in success. Please refer to ["2.1 Return value"](#) for the error code except it.

Example

```
escp!.printPDF417(
    "http://www.citizen-systems.co.jp/printer/tps/index.html",
    withDigits: 0,
    withSteps: 0,
    withModuleWidth: 3,
    withStepHeight: 3,
    withECLevel: CMP_PDF417_EC_LEVEL_0,
    withAlignment: CMP_ALIGNMENT_LEFT)
```

2.14. printQRCode method

Syntax

```
func printQRCode(data: String?, withModuleSize moduleSize: Int32, withECLevel ECLevel: Int32,
    withAlignment alignment: Int32) -> Int32
```

Parameter

The meaning and the setting range of the parameters are as follows.

Value	[IN/OUT]	Meaning	Setting range
data	[IN]	Barcode data	
moduleSize	[IN]	Module width (dot)	1 - 16
ECLevel	[IN]	Error correction level	CMP_QRCODE_EC_LEVEL_L: Level L (7%) CMP_QRCODE_EC_LEVEL_M: Level M (15%) CMP_QRCODE_EC_LEVEL_Q: Level Q (25%) CMP_QRCODE_EC_LEVEL_H: Level H (30%)
alignment	[IN]	Barcode alignment	CMP_ALIGNMENT_LEFT: Left alignment CMP_ALIGNMENT_CENTER: Center alignment CMP_ALIGNMENT_RIGHT: Right alignment Other Values: Distance from the left-most print column to the start of the barcode.

Description

This method is used to print QRCode barcode.

Please refer to the Command Reference of the printer for details on each parameter.

The designation of CMP_ALIGNMENT_CENTER and CMP_ALIGNMENT_RIGHT of the Barcode alignment on the page mode is ignored.

Return value

Return CMP_SUCCESS (0) in success. Please refer to "[2.1 Return value](#)" for the error code except it.

Example

```
escp!.printQRCode(
    "http://www.citizen-systems.co.jp/printer/tps/index.html",
    withModuleSize: 4,
    withECLevel: CMP_QRCODE_EC_LEVEL_L,
    withAlignment: CMP_ALIGNMENT_LEFT)
```

2.15. printGS1DataBarStacked method

Syntax

```
func printGS1DataBarStacked(data: String?, withSymbology symbology: Int32, withModuleSize
    moduleSize: Int32, withMaxWidth maxWidth: Int32, withAlignment alignment: Int32) ->
    Int32
```

Parameter

The meaning and the setting range of the parameters are as follows.

Value	[IN/OUT]	Meaning	Setting range
data	[IN]	Barcode data	
symbology	[IN]	Barcode symbol type	CMP_BCS_GS1DATABAR_S : GS1 DataBar Stacked CMP_BCS_GS1DATABAR_E_S : GS1 DataBar Expanded Stacked CMP_BCS_GS1DATABAR_S_O : GS1 DataBar Stacked Omnidirectional
moduleSize	[IN]	Module width (dot)	1 - 16
maxWidth	[IN]	Max width (dot)	106 - 39528 Max width of GS1 DataBar Expanded Stacked.
alignment	[IN]	Barcode alignment	CMP_ALIGNMENT_LEFT: Left alignment CMP_ALIGNMENT_CENTER: Center alignment CMP_ALIGNMENT_RIGHT: Right alignment Other Values: Distance from the left-most print column to the start of the barcode.

Description

This method is used to print 2-dimensional GS1 DataBar barcode.

This method can use only the printers of

CT-D150/CT-E351/CT-S251/310II/601/651/801/851/601II/651II/801II/851II series.

Please refer to the Command Reference of the printer for details on each parameter.

The designation of CMP_ALIGNMENT_CENTER and CMP_ALIGNMENT_RIGHT of the Barcode alignment on the page mode is ignored.

Return value

Return CMP_SUCCESS (0) in success. Please refer to "[2.1 Return value](#)" for the error code except it.

Example

```
escp!.printGS1DataBarStacked(
    "0123456789012",
    withSymbology: CMP_BCS_GS1DATABAR_S,
    withModuleSize: 4,
    withMaxWidth: 400,
    withAlignment: CMP_ALIGNMENT_LEFT)
```

2.16. cutPaper method

Syntax

```
func cutPaper(percentage: Int32) -> Int32
```

Parameter

The meaning and the setting range of the parameters are as follows.

Value	[IN/OUT]	Meaning	Setting range
type	[IN]	Cut type	CMP_CUT_FULL: Full cut CMP_CUT_PARTIAL: Partial cut CMP_CUT_FULL_PREFEED : After feed the paper to the cutting position, full cut. CMP_CUT_PARTIAL_PREFEED : After feed the paper to the cutting position, partial cut.

Description

This method is used to cut the paper.

Return value

Return CMP_SUCCESS (0) in success. Please refer to "[2.1 Return value](#)" for the error code except it.

Example

```
escp!.cutPaper(CMP_CUT_PARTIAL_PREFEED)
```

2.17. unitFeed method

Syntax

```
func unitFeed(IfConunt: Int32) -> Int32
```

Parameter

The meaning and the setting range of the parameters are as follows.

Value	[IN/OUT]	Meaning	Setting range
ufCount	[IN]	Number of paper feed (Dot units)	

Description

This method is used to feed the paper in dot units.

Return value

Return CMP_SUCCESS (0) in success. Please refer to "[2.1 Return value](#)" for the error code except it.

Example

```
escp!.unitFeed(200)
```


2.18. markFeed method

Syntax

```
func markFeed(type: Int32) -> Int32
```

Parameter

The meaning and the setting range of the parameters are as follows.

Value	[IN/OUT]	Meaning	Setting range
type	[IN]	Handling type of label paper or black mark paper	CMP_MF_TO_CUTTER : After feed the paper to the auto cutter cutting position, cut further. CMP_MF_TO_NEXT_TOF : Feed the paper to the next paper's top of form.

Description

This method is used to utilize label paper and black mark paper.

Return value

Return CMP_SUCCESS (0) in success. Please refer to "[2.1 Return value](#)" for the error code except it.

Example

```
escp!.markFeed(CMP_MF_TO_CUTTER)
```

2.19. openDrawer method

Syntax

```
func openDrawer(drawer: Int32, withPulseLength pulsLen: Int32) -> Int32
```

Parameter

The meaning and the setting range of the parameters are as follows.

Value	[IN/OUT]	Meaning	Setting range
drawer	[IN]	Cash drawer number	CMP_DRAWER_1: Drawer 1 CMP_DRAWER_2: Drawer 2
pulseLen	[IN]	Signal length	1 - 8 Set theONtime/OFF time to the value x100ms, respectively.

Description

This method is used to open the cash drawer is connected to the printer.

Return value

Return CMP_SUCCESS (0) in success. Please refer to "[2.1 Return value](#)" for the error code except it.

Example

```
escp?.openDrawer(CMP_DRAWER_1, withPulseLength: 1)
```

2.20. transactionPrint method

Syntax

```
func transactionPrint(control: Int32) -> Int32
```

Parameter

The meaning and the setting range of the parameters are as follows.

Value	[IN/OUT]	Meaning	Setting range
control	[IN]	Transaction control	CMP_TP_TRANSACTION : Begin a transaction. CMP_TP_NORMAL : End a transaction by printing the buffered data.

Description

This method is used to start or end a transaction mode.

If control is CMP_TP_TRANSACTION, then transaction mode is entered. Subsequent methods calls will buffer the print data. The methods applied to a transaction mode are as follows.

printText, printBitmap, printNVBitmap, printBarCode, printPDF417, printQRCode, cutPaper, unitFeed, markFeed, openDrawer, rotatePrint, pageModePrint, clearPrintArea, printData, printNormal

If control is CMP_TP_NORMAL, then transaction mode is exited. If some data was buffered, then the buffered data is printed. The entire transaction is treated as one message.

Calling the clearOutput method cancels transaction mode. Any buffered print lines are also cleared.

Return value

Return CMP_SUCCESS (0) in success. Please refer to ["2.1 Return value"](#) for the error code except it.

Example

```
escp!.transactionPrint(CMP_TP_TRANSACTION)
escp!.printNVBitmap(1)
escp!.printBarCode("123456789012", withSymbology: CMP_BCS_UPCA,withHeight: 50,
    withWidth: 2, withAlignment: CMP_ALIGNMENT_LEFT, withTextPosition:
    CMP_HRI_TEXT_ABOVE)
escp!.printText("Line 1\n", withAlignment: CMP_ALIGNMENT_LEFT,
    withAttribute: CMP_FNT_DEFAULT, withTextSize: CMP_TXT_1WIDTH)
escp!.printText("Line 2\n", withAlignment: CMP_ALIGNMENT_LEFT,
    withAttribute: CMP_FNT_DEFAULT, withTextSize: CMP_TXT_1WIDTH)
escp!.printText("Line 3\n", withAlignment: CMP_ALIGNMENT_LEFT,
    withAttribute: CMP_FNT_DEFAULT, withTextSize: CMP_TXT_1WIDTH )
escp!.printBarCode("123456789012", withSymbology: CMP_BCS_UPCA,
    withHeight: 50, withWidth: 2, withAlignment: CMP_ALIGNMENT_LEFT,
    withTextPosition: CMP_HRI_TEXT_ABOVE)
escp!.printNVBitmap(1)
escp!.cutPaper(CMP_CUT_PARTIAL_PREFEED)
escp!.transactionPrint(CMP_TP_NORMAL)
```

2.21. rotatePrint method

Syntax

```
func rotatePrint(rotation: Int32) -> Int32
```

Parameter

The meaning and the setting range of the parameters are as follows.

Value	[IN/OUT]	Meaning	Setting range
rotation	[IN]	Direction of rotation	CMP_RP_ROTATE180: Start rotated printing 180°, that is, print upside-down CMP_RP_BARCODE : Start rotated bar code printing. This value is ORed with the above start rotated print values. CMP_RP_BITMAP : Start rotated bitmap printing. This value is ORed with the above start rotated print values. CMP_RP_NORMAL : End rotated printing

Description

This method is used to start or end a rotation print mode.

If rotation includes PTR_RP_ROTATE180, then upside-down print mode is entered. The methods applied to a rotation print mode are as follows.

printText, printNormal

If rotation includes PTR_RP_BARCODE and/or PTR_RP_BITMAP, the following methods are printed also rotated.

printBarcode, printPDF417, printQRCode and/or printBitmap

If rotation is CMP_RP_NORMAL, then rotation mode is exited.

Return value

Return CMP_SUCCESS (0) in success. Please refer to ["2.1 Return value"](#) for the error code except it.

Example

```
escp!.rotatePrint(CMP_RP_ROTATE180 | CMP_RP_BARCODE | CMP_RP_BITMAP)
escp!.printBitmap("samplebitmap.bmp", withWidth: CMP_BM_ASIS, withAlignment:
    CMP_ALIGNMENT_CENTER)
escp!.printBarCode("123456789012", withSymbology: CMP_BCS_UPCA, withHeight:
    50, withWidth: 2, withAlignment: CMP_ALIGNMENT_LEFT,
    withTextPosition: CMP_HRI_TEXT_ABOVE)
escp!.printText("Line 1\n", withAlignment: CMP_ALIGNMENT_LEFT, withAttribute:
    CMP_FNT_DEFAULT, withTextSize: CMP_TXT_1WIDTH)
escp!.printText("Line 2\n", withAlignment: CMP_ALIGNMENT_LEFT, withAttribute:
    CMP_FNT_DEFAULT, withTextSize: CMP_TXT_1WIDTH)
escp!.printText("Line 3\n", withAlignment: CMP_ALIGNMENT_LEFT, withAttribute:
    CMP_FNT_DEFAULT, withTextSize: CMP_TXT_1WIDTH)
escp!.cutPaper(CMP_CUT_PARTIAL_PREFEED)
escp!.rotatePrint(CMP_RP_NORMAL)
```

2.22. pageModePrint method

Syntax

```
func pageModePrint(control: Int32) -> Int32
```

Parameter

The meaning and the setting range of the parameters are as follows.

Value	[IN/OUT]	Meaning	Setting range
control	[IN]	Page Mode control	CMP_PM_PAGE_MODE: Enter Page Mode CMP_PM_PRINT_SAVE: Print PageModePrintArea and save the canvas CMP_PM_NORMAL: Print the print area and destroy the canvas and exit Page Mode. CMP_PM_CANCEL: Clear the page and exit the Page Mode without any printing of any print area

Description

This method is used to start or end a Page Mode.

If control is PTR_PM_PAGE_MODE, then Page Mode is entered. Subsequent methods calls will buffer the print data. The methods applied to a Page Mode are as follows.

PrintText, PrintBitmap, PrintBarCode, printPDF417, printQRCode, PrintNormal

If control is PTR_PM_PRINT_SAVE, then Page Mode is not exited. If some data is buffered, then the buffered data is saved and printed. This control is used to print the same page layout with additional print items inside of the page.

If control is PTR_PM_NORMAL, then Page Mode is exited. If some data is buffered, then the buffered data is printed. The buffered data will not be saved.

If control is PTR_PM_CANCEL, then Page Mode is exited. If some data is buffered, then the buffered data is not printed and is not saved.

Note that when the pageModePrint method is called, all of the data that is to be printed in the PageModePrintArea will be printed and the paper is fed to the end of the PageModePrintArea. If more than one PageModePrintArea is defined, then after the pageModePrint method is called, all of the data that is to be printed in the respective PageModePrintArea(s) will be printed and the paper will be fed to the end of the PageModePrintArea located the farthest "down" the sheet of paper.

The entire Page Mode transaction is treated as one message.

Calling the clearOutput method cancels Page Mode. Any buffered print lines are also cleared.

Return value

Return CMP_SUCCESS (0) in success. Please refer to ["2.1 Return value"](#) for the error code except it.

Example

```
// Standard print
var sample2: String = ""
var receipt: String = ""
sample2 = String(format: "%c|2vCSample 2 - Print\n", 27)
escp!.printNormal(sample2)
escp!.printText("12345678901234567890123456789012345678901234567890123456789012345678901234567890\n", withAlignment:
    CMP_ALIGNMENT_RIGHT, withAttribute: CMP_FNT_DEFAULT, withTextSize:
    (CMP_TXT_1WIDTH|CMP_TXT_1HEIGHT))
// Start of Page Mode
escp!.pageModePrint(CMP_PM_PAGE_MODE)
// Set offset of Page Mode
escp!.setPageModeVerticalPosition(0)
escp!.setPageModeHorizontalPosition(0)
// Set direction of Page Mode
escp!.setPageModePrintDirection(CMP_PD_TOP_TO_BOTTOM)
// Set print area of Page Mode
escp!.setPageModePrintArea("308,0,76,800")
receipt = String(format: "%c|4C- Receipt -\n", 27)
escp!.printNormal(receipt)
// Set print area of Page Mode
escp!.setPageModePrintArea("184,0,120,800")
escp!.printText(" $ 299.99- \n", withAlignment: CMP_ALIGNMENT_CENTER,
    withAttribute: (CMP_FNT_UNDERLINE|CMP_FNT_BOLD),
    withTextSize: (CMP_TXT_4WIDTH | CMP_TXT_4HEIGHT))
// Set print area of Page Mode
escp!.setPageModePrintArea("88,0,88,560")
escp!.printText("CITIZEN SYSTEMS\n", withAlignment: CMP_ALIGNMENT_RIGHT,
    withAttribute: CMP_FNT_DEFAULT, withTextSize: CMP_TXT_2WIDTH |
    CMP_TXT_3HEIGHT)
// Set print area of Page Mode
escp!.setPageModePrintArea("0,0,98,480")
escp!.printBarCode("123456789012", withSymbology: CMP_BCS_UPCA,
    withHeight: 64, withWidth: 4, withAlignment: CMP_ALIGNMENT_LEFT,
    withTextPosition: CMP_HRI_TEXT_BELOW)
// Set print area of Page Mode
escp!.setPageModePrintArea("0,600,192,192")
escp!.printQRCode("http://www.citizen-systems.co.jp/", withModuleSize: 5,
    withECLevel: CMP_QRCODE_EC_LEVEL_L,
    withAlignment: CMP_ALIGNMENT_LEFT)
// End of Page Mode
escp!.pageModePrint(CMP_PM_NORMAL)
```

Print image



2.23. clearPrintArea method

Syntax

```
func clearPrintArea() -> Int32
```

Parameter

Not exist.

Description

This method is used to clear the area defined by the PageModePrintArea property.

Return value

Return CMP_SUCCESS (0) in success. Please refer to "[2.1 Return value](#)" for the error code except it.

Example

```
escp!.clearPrintArea()
```

2.24. clearOutput method

Syntax

```
func clearOutput() -> Int32
```

Parameter

Not exist.

Description

This method is used to clear all buffered output data by tranzactionPrint and pageModePrint method. Also, when possible, halts outputs that are in progress. At the same time, the command to clear print data on the printer is sent.

Return value

Return CMP_SUCCESS (0) in success. Please refer to "[2.1 Return value](#)" for the error code except it.

Example

```
escp!.clearOutput()
```


2.25. printData method

Syntax

```
func printData(data: UnsafeMutablePointer<Int8>) -> Int32
```

Parameter

The meaning and the setting range of the parameters are as follows.

Value	[IN/OUT]	Meaning	Setting range
data	[IN]	Send data	

Description

This method is used to send data bytes to the printer directly.

It is usually not necessary, please use if you want to send ESC commands directly to the printer.

If you want to use, please be careful so as not to affect the other methods.

Return value

Return CMP_SUCCESS (0) in success. Please refer to "[2.1 Return value](#)" for the error code except it.

Example

```
// Sound the buzzer (The printer must support buzzer.)
var data: [Int8] = [0x1b, 0x1e]
res = escp!.printData(&data)
```

2.26. printNormal method

Syntax

```
func printNormal(data: String?) -> Int32
```

Parameter

The meaning and the setting range of the parameters are as follows.

Value	[IN/OUT]	Meaning	Setting range
Data	[IN]	Print data (Support OPOS escape sequence)	

Description

This method is used to print using the escape sequences that are defined in the OPOS. Please use this if you are familiar with the OPOS.

The supporting escape sequences in this SDK are as follows.

Please refer to specifications of OPOS for the details.

Escape Sequence		Notes
Paper cut	ESC #P	Partial cut (1-99), Full cut (0,100)
Feed and paper cut	ESC #fP	Partial cut (1-99), Full cut (0,100)
Bitmap print	ESC #B	1-20 (Bitmap image number that is stored in the flash memory of the printer) After Bitmap printing, print position returns to the initial state (left-justified).
Multi-line feed	ESC #IF	
Unit feed	ESC #uF	
Barcode print	ESC #R	
Font type specification	ESC #FT	
Bold	ESC bC	
Underline	ESC #uC	
Custom color	ESC #rC	Effective only when dedicated 2-color paper is used.
Red	ESC rC	Effective only when dedicated 2-color paper is used.
Reverse character	ESC rvC	
Standard	ESC 1C	
Double width	ESC 2C	
Double height	ESC 3C	
Quadruple	ESC 4C	
Horizontal magnification	ESC #hC	1-8
Vertical magnification	ESC #vC	1-8
Centering	ESC cA	
Right adjustment	ESC rA	
Normal	ESC N	

Return value

Return CMP_SUCCESS (0) in success. Please refer to "[2.1 Return value](#)" for the error code except it.

Example

```
var receipt = String(format: "%c|2vC Rceipt -\n", 27)
escp!.printNormal(receipt)
```

2.27. getVersionCode method

Syntax

```
func getVersionCode() -> Int32
```

Parameter

Not exist.

Description

This method is used to get a numerical value for the version number of this SDK.

Return value

Return a numerical value for the version number of this SDK. (Ver1.00 is 100)

Example

```
escp!.getVersionCode()
```

2.28. getVersionName method

Syntax

```
func getVersionName() -> String?
```

Parameter

Not exist.

Description

This method is used to get a string for the version number of this SDK.

Return value

Return a string for the version number of this SDK. (Ver1.00 is "1.00")

Example

```
escp!.getVersionName()
```

2.29. watermarkPrint method

Syntax

```
func watermarkPrint(start: Int32, withNVImageNumber nvImageNumber: Int32, withPass pass: Int32,
    withFeed feed: Int32, withRepeat repeat: Int32) -> Int32
```

Parameter

The meaning and the setting range of the parameters are as follows.

Value	[IN/OUT]	Meaning	Setting range
start	[IN]	The start / Stop of the watermark print	CMP_WM_START: The start of the watermark print CMP_WM_STOP: The stop of the watermark print
nvImageNumber	[IN]	The NV image number that is stored in the flash memory of the printer	1 - 20
pass	[IN]	The first start position (vertical direction) of the watermark	0 - 65,535 [dot]
feed	[IN]	The blank length each watermark	0 - 65,535 [dot]
repeat	[IN]	The print number of times of the watermark	0: Infinite repetition 1 - 65,535: The repetition number of times

Description

This method is used to print watermark.

This is available with a printer of the CT-S251/601II/651II/801II/851II series.

The bitmap image stored in the flash memory of the printer is printed out as watermark.

It is necessary to store the image beforehand to use this method.

About logo registration method, please store it using [setNVBitmap_method](#) or use the "POS Printer utility" of utility software for the printer.

When the printing of watermark was stopped in CMP_WM_STOP, all other arguments are ignored

Return value

Return CMP_SUCCESS (0) in success. Please refer to ["2.1 Return value"](#) for the error code except it.

Example

```
escp!.watermarkPrint(CMP_WM_START, withNVImageNumber: 1, withPass: 0,
    withFeed: 0, withRepeat: 0)
```

2.30. searchCitizenPrinter method

Syntax

```
class func searchCitizenPrinter(ifType: Int32, withSearchTime searchTime: Int32, result:
    UnsafeMutablePointer<Int32>) -> [AnyObject]?
```

Parameter

The meaning and the setting range of the parameters are as follows.

Value	[IN/OUT]	Meaning	Setting range
ifType	[IN]	Connect type	CMP_PORT_WiFi: Wi-Fi connection CMP_PORT_BLUETOOTH: Bluetooth connection
searchTime	[IN]	Search time (sec)	1 - 30
result	[OUT]	Error code	---

Return CMP_SUCCESS (0) to result in success. Please check the description of the error codes below in the case of failure. Please refer to "[2.1 Return value](#)" for the error code except it.

Error codes	Description
CMP_E_ILLEGAL (1101)	Invalid parameter. (1) The connect type is unsupported. (2) The search time is out of range.
CMP_E_FAILURE (1104)	Error occurs during the search, the search failed.
CMP_E_NO_LIST (1106)	As a result of search, the printer cannot be found.

Description

This method is used to search the printer and to obtain the list of the printer information. Please specify the type of the printer connection and the search time. Only WiFi connection is available at the time of the simulator use. After search time passed, set a result to the result parameter and return the information of the found printers as NSArray type.

In the case of CMP_PORT_WiFi for the connection type, you can search only the printers of CT-D150,CT-E351,CT-S251/310II/601/651/801/851/601II/651II/801II/851II series. Recommended value of search time is more than 3 seconds. When the search time is shorter than the second, a search may fail by the network situation.

In the case of CMP_PORT_BLUETOOTH for the connection type, CT-S251/281/601II/651II/801II/851II can be searched. It does not depend on the setting of searchTime, and the search is finished immediately.

Return value

A list of information of the searched printer is given back on success. An empty list is returned on failure. The list of information of the printer is stored as a CitizenPrinterInfo-type, and available information varies according to ifType parameter.

ifType	CitizenPrinterInfo	Information to be obtained
CMP_PORT_WiFi	ipAddress	IP Address
	macAddress	MAC Address
	deviceName	(Empty character)
	bdAddress	(Empty character)
CMP_PORT_BLUETOOTH	ipAddress	(Empty character)
	macAddress	(Empty character)
	deviceName	(Empty character) / Bluetooth device name *depend

		on the printer model
	bdAddress	(Empty character) / BD Address * in the case of iOS6 or more use

Example

```

var result = Int32(0)
let array = ESCPOSPrinter.searchCitizenPrinter(CMP_PORT_WiFi, withSearchTime: 4, result:
    &result)!
for var i = 0; i < array.count; i++ {
    let prninfo = array[i] as? CitizenPrinterInfo
    println("IP Address : \(prninfo.ipAddress)")
    println("MAC Address : \(prninfo.macAddress)")
}

```

2.31. searchESCPOSPrinter method

Syntax

```
class func searchESCPOSPrinter(ifType: Int32, withSearchTime searchTime: Int32, result:
    UnsafeMutablePointer<Int32>) -> [AnyObject]?
```

Parameter

The meaning and the setting range of the parameters are as follows.

Value	[IN/OUT]	Meaning	Setting range
ifType	[IN]	Connect type	CMP_PORT_WiFi: Wi-Fi connection CMP_PORT_BLUETOOTH: Bluetooth connection
searchTime	[IN]	Search time (sec)	1 - 30
result	[OUT]	Error code	---

Return CMP_SUCCESS (0) to result in success. Please check the description of the error codes below in the case of failure. Please refer to "[2.1 Return value](#)" for the error code except it.

Error codes	Description
CMP_E_ILLEGAL (1101)	Invalid parameter. (1) The connect type is unsupported. (2) The search time is out of range.
CMP_E_FAILURE (1104)	Error occurs during the search, the search failed.
CMP_E_NO_LIST (1106)	As a result of search, the printer cannot be found.

Description

This method is used to search the printer and to obtain the list of the addresses. Please specify the type of the printer connection and the search time. Only WiFi connection is available at the time of the simulator use. After search time passed, set a result to the result parameter and return the information of the found printers as NSArray type.

In the case of CMP_PORT_WiFi for the connection type, you can search only the printers of CT-D150,CT-E351,CT-S251/310II/601/651/801/851/601II/651II/801II/851II series. Recommended value of search time is more than 3 seconds. When the search time is shorter than the second, a search may fail by the network situation.

In the case of CMP_PORT_BLUETOOTH for the connection type, CT-S251/281/601II/651II/801II/851II can be searched. It does not depend on the setting of searchTime, and the search is finished immediately.

Return value

In the case of CMP_PORT_WiFi for the connection type, return the list of IP addresses of the printers when a search succeeded. When a search fails, return the empty list.

In the case of CMP_PORT_BLUETOOTH for the connection type and iOS6 or more, return the list of Bluetooth device addresses of the printers when a search succeeded. When a search fails, return the empty list. Because it does not correspond to information on the Bluetooth device address for less than iOS6, the empty list is returned.

Example

```
var result = Int32(0)
let array = ESCPOSPrinter.searchESCPOSPrinter(CMP_PORT_WiFi, withSearchTime:
    4, result: &result)!
```


2.32. PageModeArea property

Type

String

Attribute

Read only

Description

This property holds the page area expressed in the unit of Dot. The string consists of two ASCII numbers separated by a comma, in the following order: horizontal size, vertical size.

This page area is determined by the hardware capability of the printer.

[CT-S251 Series] : "432,1662"

[CT-S281 Series] : "384,938"

[CT-D150,CT-E351,CT-S310II/601/651/801/851/601II/651II/801II/851II/2000 Series] : "576,1662"

[CT-S4000 Series] : "832,1662"

For example, if the string is "384,938", then the page size is 384 horizontal units by 938 vertical units, and the station print area is a rectangle beginning at the top left point (0,0), and continuing up to the bottom right point (383,937).

The connect method must be complete before accessing this property. This property is set in connect method.

Set property

Not exist.

Get property

```
func getPageModeArea() -> String?
```

Returns the page area as the return value.

2.33. PageModePrintArea property

Type

String

Attribute

Read/Write

Description

This property holds the print area of Page Mode, expressed in the unit of dot. The maximum print area is the page area.

The string consists of four ASCII numbers separated by commas, in the following order: horizontal start, vertical start, horizontal size, vertical size.

Text written to the right edge of the print area will wrap to the next line. Any text or image written beyond the bottom of the print area will be truncated.

For example, if the string is "50,100,200,400", then the station print area is a rectangle beginning at the point (50,100), and continuing up to the point (249,499).

The connect method must be complete before accessing this property. This property is initialized to "0,0,0,0" at connect method.

Set property

```
func setPageModePrintArea(area: String?) -> Int32
```

Please specify the property value that you want to set in the parameter.

Return CMP_SUCCESS (0) in success. Please refer to ["2.1 Return value"](#) for the error code except it.

Get property

```
func getPageModePrintArea() -> String?
```

Returns the Page Mode print area that is set as the return value.

2.34. PageModePrintDirection property

Type

Int32

Attribute

Read/Write

Description

This property holds the print direction of the Page Mode print area. The print direction values are as follows.

Value	Meaning
CMP_PD_LEFT_TO_RIGHT	Print left to right, starting at top left position of the print area, i.e., normal printing.
CMP_PD_BOTTOM_TO_TOP	Print bottom to top, starting at the bottom left position of the print area, i.e., rotated left 90° printing.
CMP_PD_RIGHT_TO_LEFT	Print right to left, starting at the bottom right position of the print area, i.e., upside down printing.
CMP_PD_TOP_TO_BOTTOM	Print top to bottom, starting at the top right position of the print area, i.e., rotated right 90° printing.

Setting this property may also change PageModeHorizontalPosition and PageModeVerticalPosition. Setting this property will have an effect on the current print area. By changing the print area, it is possible to generate a receipt or slip with text printed in multiple rotations.

The connect method must be complete before accessing this property. This property is initialized to CMP_PD_LEFT_TO_RIGHT at connect method.

Set property

```
func setPageModePrintDirection(direction: Int32) -> Int32
```

Please specify the property value that you want to set in the parameter.

Return CMP_SUCCESS (0) in success. Please refer to ["2.1 Return value"](#) for the error code except it.

Get property

```
func getPageModePrintDirection() -> Int32
```

Returns the print direction of Page Mode print area that is set as the return value.

2.35. PageModeHorizontalPosition property

Type

Int32

Attribute

Read/Write

Description

This property holds the horizontal start position offset within the Page Mode print area, expressed in the unit of dot.

The horizontal direction is the same as the actual PageModePrintDirection property.

A read/get on this property will return the horizontal position offset set by the last write/set and not the current position.

The connect method must be complete before accessing this property. This property is initialized to zero (0) at connect method.

Set property

```
func setPageModeHorizontalPosition(position: Int32) -> Int32
```

Please specify the property value that you want to set in the parameter.

Return CMP_SUCCESS (0) in success. Please refer to "[2.1 Return value](#)" for the error code except it.

Get property

```
func getPageModeHorizontalPosition() -> Int32
```

Returns the horizontal position of Page Mode print area that is set as the return value.

2.36. PageModeVerticalPosition property

Type

Int32

Attribute

Read/Write

Description

This property holds the vertical start position offset within the Page Mode print area, expressed in the unit of dot.

The vertical direction is perpendicular to the direction specified in the actual PageModePrintDirection property.

A read/get on this property will return the vertical position offset set by the last write/set and not the current position.

The connect method must be complete before accessing this property. This property is initialized to zero (0) at connect method.

Set property

```
func setPageModeVerticalPosition(position: Int32) -> Int32
```

Please specify the property value that you want to set in the parameter.

Return CMP_SUCCESS (0) in success. Please refer to "[2.1 Return value](#)" for the error code except it.

Get property

```
func getPageModeVerticalPosition() -> Int32
```

Returns the vertical position of Page Mode print area that is set as the return value.

2.37. RecLineSpacing property

Type

Int32

Attribute

Read/Write

Description

This property holds the spacing of each single-high print line, including both the printed line height plus the whitespace between each pair of lines, expressed in the unit of dot.

Depending upon the current line spacing, a multi-high print line might exceed this value. In this case the whitespace is zero.

The connect method must be complete before accessing this property. This property is initialized to 34 at connect method.

Set property

```
func setRecLineSpacing(spacing: Int32) -> Int32
```

Please specify the property value that you want to set in the parameter.

Return CMP_SUCCESS (0) in success. Please refer to ["2.1 Return value"](#) for the error code except it.

Get property

```
func getRecLineSpacing() -> Int32
```

Returns the spacing of each single-high print line that is set as the return value.

2.38. MapMode property

Syntax

Int32

Attribute

Read/Write

Description

This property holds the mapping mode of the printer. The mapping mode defines the unit of measure used for other properties, such as line heights and line spacing. The map mode values are as follows.

Value	Meaning
CMP_MM_DOTS	The printer's dot width.
CMP_MM_TWIPS	1/1440 of an inch.
CMP_MM_ENGLISH	0.001 inch.
CMP_MM_METRIC	0.01 millimeter.

The method and the properties to be affected by the MapMode property are as follows.

[printBitmap method](#) : width, alignment
[printBitmapData method](#) : width, alignment
[setNVBitmap method](#) : width
[printBarcode method](#) : height, width, alignment
[printPDF417 method](#) : moduleWidth, alignment
[printQRCode method](#) : moduleSize, alignment)
[printGS1DataBarStacked method](#) : moduleSize, maxSize, alignment
[unitFeed method](#) : ufCount
[watermarkPrint method](#) : pass, feed
[PageModeArea property](#)
[PageModePrintArea property](#)
[PageModeHorizontalPosition property](#)
[PageModeVerticalPosition property](#)
[RecLineSpacing property](#)

The connect method must be complete before accessing this property. This property is initialized to CMP_MM_DOTS at connect method.

Set property

```
func setMapMode (number: Int32)->Int32
```

Please specify the property value that you want to set in the parameter.

Return CMP_SUCCESS (0) in success. Please refer to "[2.1 Return value](#)" for the error code except it.

Get property

```
func getMapMode ()->Int32
```

Returns the mapping mode that is set as the return value.

3. Notes

Notes of this SDK are as follows.

3.1. Function to detect the completion of printing

In this library, after the printing output, the SDK waits for the printing completion reply from a printer and judge the success / failure of the method.

The function to detect the completion of printing is processed in the following cases.

- (1) At the time of completion of transaction processing (transactionPrint method)
- (2) At the time of completion of page mode (pagePrint method)
- (3) At the time of data output of the methods except during the buffering process in transaction or page mode

The function to detect the completion of printing need a few time to wait for the response of the printer. If you want to print continuously multiple methods, smooth printing is possible by using transaction processing. (transactionPrint method)

iOS POS Print SDK (Swift) Programing Manual

2017/04/14 For Ver. 1.17

CITIZEN SYSTEMS JAPAN CO., LTD.

<http://www.citizen-systems.co.jp/>