CITIZEN

Linux POS Print SDK

Programming Manual

For Ver. 1.10

CITIZEN SYSTEMS JAPAN CO., LTD.

Revision Record

Date	Version	Description
Sep 7, 2016	1.00	New issue.
Oct 19, 2016	1.01	Updated the version number only.
Jan 16, 2017	1.02	- Added CT-D150/CT-E351 to the support models.
Jun 8, 2017	1.03	- Added CT-D151/CT-E651 to the support models.
Dec 19, 2018	1.04	- Added CT-S751 to the support models. - Added "3.2 About printing UTF-8 encode characters".
Jan 25, 2019	1.05	- Added CT-S4500 to the support models.
Nov 11, 2019	1.06	- Added PMU3300 to the support models.
Mar 30, 2020	1.07	- Added Near Empty / Drawer / Paper is hold status to the status method. - Added the PrinterCheckEx method.
May 14, 2021	1.08	- Added CT-D101/CT-E301/CT-E601 to the support models. - Added the SetPrintCompletedTimeout method. - Modified the PageModeArea property of the PMU3300. - Added the predefined constants list.
Dec 23, 2021	1.09	- Added Rasppberry Pi OS (Bullseye) to the support OS. - Added CT-S4000 to the support models.
Jul 29, 2022	1.10	- Added Rasppberry Pi OS (Bullseye) 64bit to the support OS, and removed older than Raspberry Pi OS Legacy. (Page 6) - Added CT-S281II to the support models. (Page 7,12)
Nov 21, 2023		- Added CT-S801III/CT-S851III to the support models. (Page 7,13)

Caution

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 Library.

Trademark

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

1. Introduction

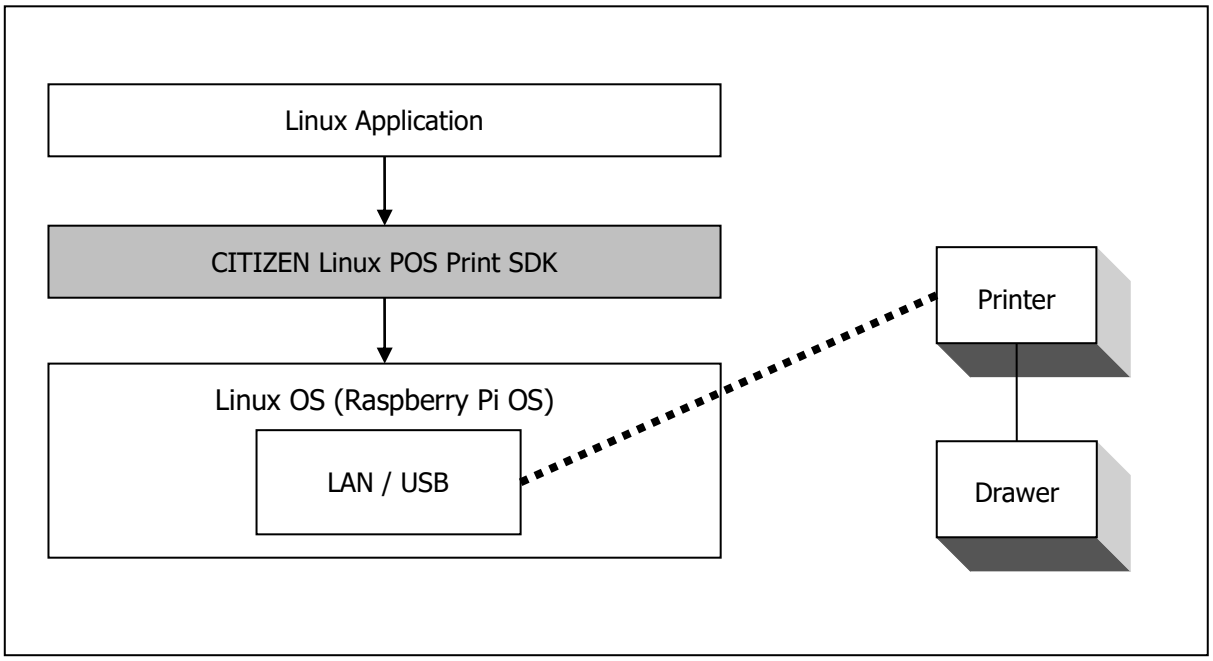
This document is a programming manual for the CITIZEN Linux POS Print SDK.

1.1. Document target range

This document is intended for developers who integrate their Linux application programs with CITIZEN POS printers.

1.2. System summary

This SDK is referred by Linux application program to control CITIZEN POS printers.



System diagram of the SDK

Library files

This library is structured by the following two files.

- libCSJPOSLib.so : Extended library
- CSJPOSLib.h : Header file for C++ language

Supported operating systems

This SDK supports the following Linux operating systems.

- Raspberry Pi OS (32bit, ARM Debian 11.0; Bullseye, Linux for Raspberry Pi)
- Raspberry Pi OS (64bit, ARM Debian 11.0; Bullseye, Linux for Raspberry Pi)
- Raspberry Pi OS (Legacy) (32bit, ARM Debian 10.0; Buster, Linux for Raspberry Pi)

Supported printer 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-D101	CT-D101	Wired LAN, USB	Standard
CT-E301	CT-E301	Wired LAN, USB	Standard
CT-E601	CT-E601	Wired/Wireless LAN, USB	Standard
CT-D150	CT-D150	Wired LAN, USB	Standard
CT-D151	CT-D151	Wired/Wireless LAN, USB	Standard
CT-E351	CT-E351	Wired LAN, USB	Standard
CT-E651	CT-E651	Wired/Wireless LAN, USB	Standard
CT-S251	CT-S251	Wired/Wireless LAN, USB	Standard
CT-S281	CT-S281/281BT/ 281BD	USB	Standard
	CT-S281-XL-M1		Blackmark paper is supported.
	CT-S281-XL		Label paper is supported.
CT-S281II	CT-S281II	USB	Standard
	CT-S281II-L		Label/Blackmark paper is supported.
CT-S310II	CT-S310II	Wired LAN, USB	Standard
CT-S601/651/801/ 851	CT-S601/651/801/8 51	Wired/Wireless LAN, USB	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	Wired/Wireless LAN, USB	Standard
	CT-S801II/851II-M		Blackmark paper is supported.
	CT-S801II-L		Label paper is supported.
CT-S801III/851III	CT-S801III/851III	Wired/Wireless LAN, USB	Standard
CT-S751	CT-S751	Wired/Wireless LAN, USB	Standard
CT-S4000	CT-S4000	Wired LAN, USB	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.
CT-S4500	CT-S4500	Wired/Wireless LAN, USB	Standard (Label/Blackmark paper is supported.)
PMU3300	PMU3300	USB	Standard

1.3. Printer setting

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

CT-D101 Series Memory Switch Setting

MSW No.	Function	Setting
1-1	Power ON Info	Valid
1-2	Buffer Size	4K bytes
1-3	Busy condition	Full
1-4	Receive Error	Print ?
1-5	CR Mode	Ignored
2-2	Auto cutter	Valid
2-4	Full Col Print	Wait Data
3-1	Resume Cttr Err	Valid
3-7	CBM1000 Mode	Valid
3-8	Resume Open Err	Close
4-8	Partial Only	Invalid
5-2	Line Pitch	1/360
5-3	USB Mode	Printer Class
6-1	Act. For Driver	Valid
7-6	DMA control	Valid
9-1	Code page	Katakana (*1)
9-2	Int'Char Set	Japan (*1)
9-4	Multi-byte Char (*2)	SJIS(CP932) GB18030 EUC Hangul BIG5

CT-D150 Series Memory Switch Setting

MSW No.	Function	Setting
1-1	Power ON Info	Valid
1-2	Buffer Size	4K bytes
1-3	Busy condition	Full
1-4	Receive Error	Print ?
1-5	CR Mode	Ignored
2-2	Auto cutter	Valid
2-4	Full Col Print	Wait Data
3-1	Resume Cttr Err	Valid
3-7	CBM1000 Mode	Valid
3-8	Resume Open Err	Close
4-8	Partial Only	Invalid
5-2	Line Pitch	1/360
5-3	USB Mode	Printer Class
6-1	Act. For Driver	Valid
7-6	DMA control	Valid
9-1	Code page	Katakana (*1)
9-2	Int'Char Set	Japan (*1)
9-4	Multi-byte Char (*2)	SJIS(CP932) GB18030 EUC Hangul BIG5

CT-D151 Series Memory Switch Setting

MSW No.	Function	Setting
1-1	Power ON Info	Valid
1-2	Buffer Size	4K bytes
1-3	Busy condition	Full
1-4	Receive Error	Print ?
1-5	CR Mode	Ignored
2-2	Auto cutter	Valid
2-4	Full Col Print	Wait Data
3-1	Resume Cttr Err	Valid
3-7	CBM1000 Mode	Valid
3-8	Resume Open Err	Close
4-8	Partial Only	Invalid
5-2	Line Pitch	1/360
5-3	USB Mode	Printer Class
6-1	Act. For Driver	Valid
7-6	DMA control	Valid
9-1	Code page	Katakana (*1)
9-2	Int'Char Set	Japan (*1)
9-4	Multi-byte Char (*2)	SJIS(CP932) GB18030 EUC Hangul BIG5

CT-E301 Series Memory Switch Setting

MSW No.	Function	Setting
1-1	Power ON Info	Valid
1-2	Buffer Size	4K bytes
1-3	Busy condition	Full
1-4	Receive Error	Print ?
1-5	CR Mode	Ignored
2-2	Auto cutter	Valid
2-4	Full Col Print	Wait Data
3-1	Resume Cttr Err	Valid
3-7	CBM1000 Mode	Valid
3-8	Resume Open Err	Close
4-8	Partial Only	Invalid
5-2	Line Pitch	1/360
5-3	USB Mode	Printer Class
6-1	Act. For Driver	Valid
7-6	DMA control	Valid
9-1	Code page	Katakana (*1)
9-2	Int'Char Set	Japan (*1)
9-4	Multi-byte Char (*2)	SJIS(CP932) GB18030 EUC Hangul BIG5

CT-E351 Series Memory Switch Setting

MSW No.	Function	Setting
1-1	Power ON Info	Valid
1-2	Buffer Size	4K bytes
1-3	Busy condition	Full
1-4	Receive Error	Print ?

MSW No.	Function	Setting
1-5	CR Mode	Ignored
2-2	Auto cutter	Valid
2-4	Full Col Print	Wait Data
3-1	Resume Cttr Err	Valid
3-7	CBM1000 Mode	Valid
3-8	Resume Open Err	Close
4-8	Partial Only	Invalid
5-2	Line Pitch	1/360
5-3	USB Mode	Printer Class
6-1	Act. For Driver	Valid
7-6	DMA control	Valid
9-1	Code page	Katakana (*1)
9-2	Int'Char Set	Japan (*1)
9-4	Multi-byte Char (*2)	SJIS(CP932) GB18030 EUC Hangul BIG5

CT-E601 Series Memory Switch Setting

MSW No.	Function	Setting
1-1	Power ON Info	Valid
1-2	Buffer Size	4K bytes
1-3	Busy condition	Full
1-4	Receive Error	Print ?
1-5	CR Mode	Ignored
2-2	Auto cutter	Valid
2-4	Full Col Print	Wait Data
3-1	Resume Cttr Err	Valid
3-7	CBM1000 Mode	Valid
3-8	Resume Open Err	Close
4-8	Partial Only	Invalid
5-2	Line Pitch	1/360
5-3	USB Mode	Printer Class
6-1	Act. For Driver	Valid
7-6	DMA control	Valid
9-1	Code page	Katakana (*1)
9-2	Int'Char Set	Japan (*1)
9-4	Multi-byte Char (*2)	SJIS(CP932) GB18030 EUC Hangul BIG5

CT-E651 Series Memory Switch Setting

MSW No.	Function	Setting
1-1	Power ON Info	Valid
1-2	Buffer Size	4K bytes
1-3	Busy condition	Full
1-4	Receive Error	Print ?
1-5	CR Mode	Ignored
2-2	Auto cutter	Valid
2-4	Full Col Print	Wait Data
3-1	Resume Cttr Err	Valid
3-7	CBM1000 Mode	Valid

MSW No.	Function	Setting
3-8	Resume Open Err	Close
4-8	Partial Only	Invalid
5-2	Line Pitch	1/360
5-3	USB Mode	Printer Class
6-1	Act. For Driver	Valid
7-6	DMA control	Valid
9-1	Code page	Katakana (*1)
9-2	Int'Char Set	Japan (*1)
9-4	Multi-byte Char (*2)	SJIS(CP932) GB18030 EUC Hangul BIG5

CT-S251 Series Memory Switch Setting

MSW No.	Function	Setting
1-1	Power ON Info	Valid
1-2	Buffer Size	4K bytes
1-3	Busy condition	Full
1-4	Receiving Error	Print ?
1-5	CR Mode	Ignored
2-2	Auto cutter	Valid
2-4	Full Col Print	Wait Data
3-1	Resume Cttr Err	Valid
3-7	CBM1000-compatible mode	Valid
3-8	Resume Open Err	Close
4-8	Partial Only	Invalid
5-2	Line Pitch	1/360
6-1	Act. For Driver	Valid
7-6	DMA control	Valid
9-1	Code page	Katakana (*1)
9-2	Int'Char Set	Japan (*1)
9-3	Kanji	ON (*1)
9-4	JIS/Shift-JIS	Shift-JIS (*1)

CT-S281 Series Memory Switch Setting

MSW No.	Function	Setting
1-1	Power ON Info	Valid
1-2	Buffer Size	4K bytes
1-3	Busy condition	Full
1-4	Receiving Error	Print ?
1-5	CR Mode	Ignored
2-2	Auto cutter	Valid
2-4	Full Col Print	Wait Data
3-1	Resume Cttr Err	L/F enabled
3-7	CBM-270-compatible mode	Valid
3-8	Resume Open Err	Close
4-8	Partial Only	Invalid
5-3	USB Mode	Printer Class
9-1	Code page	Katakana (*1)
9-2	Int'Char Set	Japan (*1)
9-3	Kanji	ON (*1)

9-4	JIS/Shift-JIS	Shift-JIS (*1)

CT-S281II Series Memory Switch Setting

MSW No.	Function	Setting
1-1	Power ON Info	Valid
1-2	Buffer Size	4K bytes
1-3	Busy condition	Full
1-4	Receiving Error	Print ?
1-5	CR Mode	Ignored
2-2	Auto cutter	Valid
2-4	Full Col Print	Wait Data
3-1	Resume Cttr Err	L/F enabled
3-7	CBM-270-compatible mode	Valid
3-8	Resume Open Err	Close
4-8	Partial Only	Invalid
5-3	USB Mode	Printer Class
6-1	Act. For Driver	Valid
9-1	Code page	Katakana (*1)
9-2	Int'Char Set	Japan (*1)
9-3	Kanji	ON (*1)
9-4	JIS/Shift-JIS	Shift-JIS (*1)

CT-S310II Series Memory Switch Setting

MSW No.	Function	Setting
1-1	Power ON Info	Valid
1-2	Buffer Size	4K bytes
1-3	Busy condition	Full
1-4	Receiving Error	Print ?
1-5	CR Mode	Ignored
2-2	Auto cutter	Valid
2-4	Full Col Print	Wait Data
3-1	Resume Cttr Err	Valid
3-7	CBM1000-compatible mode	Valid
3-8	Resume Open Err	Close
4-8	Partial Only	Invalid
5-2	Line Pitch	1/360
5-3	USB Mode	Printer Class
6-1	Act. For Driver	Valid
7-6	DMA control	Valid
9-1	Code page	Katakana (*1)
9-2	Int'Char Set	Japan (*1)
9-3	Kanji	ON (*1)
9-4	JIS/Shift-JIS	Shift-JIS (*1)

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

MSW No.	Function	Setting
1-1	Power ON Info	Valid
1-2	Buffer Size	4K bytes
1-3	Busy condition	Full
1-4	Receiving Error	Print ?
1-5	CR Mode	Ignored

MSW No.	Function	Setting
2-2	Auto cutter	Valid
2-4	Full Col Print	Wait Data
3-1	Resume Cttr Err	Valid
3-3	Parallel 31Pin	Reset
3-7	CBM1000-compatible mode	Valid
3-8	Resume Open Err	Close
4-8	Partial Only	Invalid
5-2	Line Pitch	1/360
5-3	USB Mode	Printer Class
6-1	Act. For Driver	Valid
7-6	DMA control	Valid
9-1	Code page	Katakana (*1)
9-2	Int'Char Set	Japan (*1)
9-3	Kanji	ON (*1)
9-4	JIS/Shift-JIS	Shift-JIS (*1)
10-3	ACK output timing	Before BUSY

CT-S601II/651II/801II/851II Series Memory Switch Setting

MSW No.	Function	Setting
1-1	Power ON Info	Valid
1-2	Buffer Size	4K bytes
1-3	Busy condition	Full
1-4	Receiving Error	Print ?
1-5	CR Mode	Ignored
2-2	Auto cutter	Valid
2-4	Full Col Print	Wait Data
3-1	Resume Cttr Err	Valid
3-3	Parallel 31Pin	Reset
3-7	CBM1000-compatible mode	Valid
3-8	Resume Open Err	Close
4-8	Partial Only	Invalid
5-2	Line Pitch	1/360
5-3	USB Mode	Printer Class
6-1	Act. For Driver	Valid
7-6	DMA control	Valid
9-1	Code page	Katakana (*1)
9-2	Int'Char Set	Japan (*1)
9-4	Multi-byte Char (*2)	SJIS(CP932) GB18030 EUC Hangul BIG5
10-3	ACK output timing	Before BUSY

CT-S801III/851III Series Memory Switch Setting

MSW No.	Function	Setting
1-1	Power ON Info	Valid
1-2	Buffer Size	4K bytes
1-3	Busy condition	Full
1-4	Receiving Error	Print ?
1-5	CR Mode	Ignored
2-2	Auto cutter	Valid

MSW No.	Function	Setting
2-4	Full Col Print	Wait Data
3-1	Resume Cttr Err	Valid
3-3	Parallel 31Pin	Reset
3-7	CBM1000-compatible mode	Valid
3-8	Resume Open Err	Close
4-8	Partial Only	Invalid
5-2	Line Pitch	1/360
5-3	USB Mode	Printer Class
6-1	Act. For Driver	Valid
7-6	DMA control	Valid
9-1	Code page	Katakana (*1)
9-2	Int'Char Set	Japan (*1)
9-4	Multi-byte Char (*2)	SJIS(CP932) GB18030 EUC Hangul BIG5
10-3	ACK output timing	Before BUSY

CT-S751 Series Memory Switch Setting

MSW No.	Function	Setting
1-1	Power ON Info	Valid
1-2	Buffer Size	4K bytes
1-3	Busy condition	Full
1-4	Receiving Error	Print ?
1-5	CR Mode	Ignored
2-2	Auto cutter	Valid
2-4	Full Col Print	Wait Data
3-1	Resume Cttr Err	Valid
3-7	CBM1000-compatible mode	Valid
3-8	Resume Open Err	Close
4-8	Partial Only	Invalid
5-2	Line Pitch	1/360
5-3	USB Mode	Printer Class
6-1	Act. For Driver	Valid
7-6	DMA control	Valid
9-1	Code page	Katakana (*1)
9-2	Int'Char Set	Japan (*1)
9-4	Multi-byte Char (*2)	SJIS(CP932) GB18030 EUC Hangul BIG5-HKSCS

CT-S4000 Series Memory Switch Setting

MSW No.	Function	Setting
1-1	Power ON Info	Valid
1-2	Buffer Size	4K bytes
1-3	Busy condition	Full
1-4	Receiving Error	Print ?
1-5	CR Mode	Ignored
1-7	DSR signal (when serial interface is used)	Invalid
1-8	Init signal (when serial interface is used)	Invalid

MSW No.	Function	Setting
2-2	Auto cutter	Valid
2-4	Full Col Print	Wait Data
3-1	Resume Cttr Err	Valid
3-3	Parallel 31Pin	Reset
3-7	CBM1000-compatible mode	Valid
3-8	Resume Open Err	Close
4-8	Partial Only	Invalid
5-2	Line Pitch	1/360
5-3	USB Mode	Printer Class
7-6	DMA control	Valid
9-1	Code page	Katakana (*1)
9-2	Int'Char Set	Japan (*1)
9-3	Kanji	ON (*1)
9-4	JIS/Shift-JIS	Shift-JIS (*1)
10-3	ACK Timing	Before BUSY

CT-S4500 Series Memory Switch Setting

MSW No.	Function	Setting
1-1	Power ON Info	Valid
1-2	Buffer Size	4K bytes
1-3	Busy condition	Full
1-4	Receiving Error	Print ?
1-5	CR Mode	Ignored
2-2	Auto cutter	Valid
2-4	Full Col Print	Wait Data
3-1	Resume Cttr Err	Valid
3-7	CBM1000-compatible mode	Valid
3-8	Resume Open Err	Close
4-8	Partial Only	Invalid
5-2	Line Pitch	1/360
5-3	USB Mode	Printer Class
6-1	Act. For Driver	Valid
7-6	DMA control	Valid
9-1	Code page	Katakana (*1)
9-2	Int'Char Set	Japan (*1)
9-4	Multi-byte Char (*2)	SJIS(CP932) GB18030 EUC Hangul BIG5-HKSCS

PMU3300 Series Memory Switch Setting

MSW No.	Function	Setting
1-1	Power ON Info	Valid
1-2	Buffer Size	4K bytes
1-3	Busy condition	Full
1-4	Receiving Error	Print ?
1-5	CR Mode	Ignored
2-2	Auto cutter	Valid
2-4	Full Col Print	Wait Data
3-7	CBM1000 Mode	Valid
3-8	Resume Open Err	Close

4-8	Partial Only	Invalid
5-3	USB Mode	Printer Class
6-1	Act. For Driver	Valid
9-1	Code page	Katakana (*1)
9-2	Int'Char Set	Japan (*1)
9-3	Kanji	ON (*1)
9-4	JIS/Shift-JIS	Shift-JIS (*1)

*1 MSW No.9-1～4 is the setting when using Japanese. Please change it by use environment.
*2 The CT-D101/150/151, CT-E301/351/601/651, CT-S601II/651II/801II/851II/801III/851III/751/4500 series can change the Multi-byte character to Shift_JIS, GB18030, EUC-KR and Big5. 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

1.4. Definition method

Adding Library

Add the extended library file to the system library (/usr/lib or /usr/local/lib).

Installation of the necessary packages

To use this SDK, install the next packages.
- g++
- libgdk-pixbuf2.0
- libsnmp40 or libsnmp30

Execute the following command to get a list of packages from the server.

```
sudo apt-get update
```

Execute the following commands to install. (Different by the use environment.)

Case of Raspberry Pi OS:
```
apt-get install g++ libgdk-pixbuf2.0 libsnmp40
```

Case of Raspbian or Raspberry Pi OS (Legacy):
```
apt-get install g++ libgdk-pixbuf2.0 libsnmp30
```

Adding access permission of USB

To print using the USB ports, needs the addition of access permission to the user to run the application.
Modify "/etc/group" as follows. (for example, add a pi)

```
lp:x:7:pi
```

Definition header file

A reference to the header file must be stated at the top of the program source code.

```
#include "CSJPOSLib.h"
```

How to Build

To use this SDK, build using the g++.
Add the header file and the library of this SDK to build settings of the application.
The example to build in the directory which located the source files and the header files is as follows.

```
g++ [source file] -o [execute file] -l CSJPOSLib -I.
```

The build of the sample program, execute the following command.

```
g++ Citizen_POS_Sample1.cpp -o Citizen_POS_Sample1 -l CSJPOSLib -I.
```

'Citizen_POS_Sample1' of the execute file is generated.

1.5. Program structure

Here is an example program in C++ which uses the SDK.

```
// Create an instance.
CSJPOSLib::ESCPOSPrinter *printer = new CSJPOSLib::ESCPOSPrinter();

// Connect printer
int result = printer->Connect(CMP_PORT_USB, "/dev/usb/lp0");
if (CMP_SUCCESS == result)
{
    // Set encoding
    printer->SetEncoding(437);

    // Start Transaction ( Batch )
    printer->TransactionPrint(CMP_TP_TRANSACTION);

    // Print Text
    printer->PrintText("Citizen_POS_sample1_CS\n\n",
        CMP_ALIGNMENT_CENTER, CMP_FNT_DEFAULT,
        CMP_TXT_1WIDTH | CMP_TXT_1HEIGHT);
    printer->PrintText("- Sample Print 1 -\n",
        CMP_ALIGNMENT_CENTER, CMP_FNT_DEFAULT,
        CMP_TXT_1WIDTH | CMP_TXT_2HEIGHT);
    printer->PrintText("12345678901234567890123456789O\n",
        CMP_ALIGNMENT_RIGHT, CMP_FNT_DEFAULT,
        CMP_TXT_1WIDTH | CMP_TXT_1HEIGHT);

    // Print QRcode
    printer->PrintQRCode("http://www.citizen-systems.co.jp/", 6,
        CMP_QRCODE_EC_LEVEL_L, CMP_ALIGNMENT_RIGHT);

    // Partial Cut with Pre-Feed
    printer->CutPaper(CMP_CUT_PARTIAL_PREFEED);

    // End Transaction ( Batch )
    result = printer->TransactionPrint(CMP_TP_NORMAL);

    // Disconnect
    printer->Disconnect();

    if (CMP_SUCCESS != result)
    {
        // Print process Error
        printf("Transaction Error : %d¥n", result);
    }
}
else
{
    // Connect Error
    printf("Connect Error : %d¥n", result);
}
```

Class definition

Connect

Print process

Disconnect

1.6. Functions list

This SDK provides the following functions.

Methods list

No	Function	Detail
1	Create class (Constructor)	This is constructor 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 a text data.
8	Print space padding text (PrintPaddingText method)	Prints a text data with space padding.
9	Print bitmap (PrintBitmap method)	Prints a bitmap file. (BMP/JPG/PNG/GIF format)
10	Print Memory bitmap (PrintMemoryBitmap method)	Prints a bitmap data. (BMP/JPG/PNG/GIF format)
11	Print NV bitmap (PrintNVBitmap method)	Prints a bitmap image that is stored in the flash memory.
12	Print BarCode (PrintBarCode method)	Prints a one-dimensional barcode.
13	Print PDF-417 (PrintPDF417 method)	Prints a PDF417 barcode.
14	Print QRcode (PrintQRCode method)	Prints a QRCode 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	Set print completed timeout (SetPrintCompletedTimeout method)	Set the timeout to check the print completion notification.
27	Get version code (GetVersionCode method)	Get a numerical value for the version number of this SDK.

28	Get version name (GetVersionName method)	Get a string for the version number of this SDK.
29	Connect and Check printer status (PrinterCheckEx method)	Get the status of the printer from the unconnected state.

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 positon (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	Show the spacing of each single-high print line.

2. Library 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_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. Constructor

Syntax

ESCPOSPrinter ()

Parameter

Not exist.

Description

It is the constructor for the library. Create an instance.

Return value

Not exist.

Example

```
CSJPOSLib::ESCPOSPrinter *printer = new CSJPOSLib::ESCPOSPrinter();
```

2.3. Connect method

Syntax

1) int Connect (int connectType, std::string addr)
2) int Connect (int connectType, std::string addr, int port)
3) int Connect (int connectType, std::string addr, int port, int timeout)

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 CMP_PORT_USB
addr	[IN]	IP address to connect or USB device file.	WiFi: 0.0.0.0 - 255.255.255.255 USB: /dev/usb/lp0 -
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.

Connection port number is valid only if you specify the connection type CMP_PORT_WiFi. If it is omitted, you connected with number 9100.

Timeout is gives the maximum number of milliseconds to connect printer. If it is omitted, you connected with 4000 milliseconds.

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. (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

```
printer->Connect( ESCPOSConst.CMP_PORT_WiFi, "192.168.182.100" );

printer->Coonnect( ESCPOSConst.CMP_PORT_USB, "/dev/usb/lp0" );
```

2.4. Disconnect method

Syntax

int Disconnect ()

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

```
printer->Disconnect();
```

2.5. SetEncoding method

Syntax

int SetEncoding (int codepage)

Parameter

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

Value	[IN/OUT]	Meaning	Setting range
codepage	[IN]	Codepage	437:Code437 850:Code850 Multilingual 852:Code852 EasternEurope 857:Code857 Turkey 858:Code858 860:Code860 Portugal 863:Code863 Canada-French 864:Code864 Arabic 865:Code865 Norway 866:Code866 Russia 874:Code874 Tai Code 18 932:KANA 949:EUCKR 950:BIG5 1252:Windows Code 54936:GB18030 65001:UTF-8

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")
This SDK supports printing UTF-8 encoded characters. Please refer to "3.2 About printing UTF-8 encode characters" for the detail.

Return value

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

Example

```
// Japanese
printer->SetEncoding( 932 );

// Chinese
printer->SetEncoding( 54936 );

// Korean
printer->SetEncoding( 949 );

// Taiwanese
printer->SetEncoding( 950 );

// UTF-8
printer->SetEncoding( 65001 );
```

2.6. PrinterCheck method

Syntax

int PrinterCheck ()

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 == printer->PrinterCheck() ) {
    // Success
} else {
    // Fail
}
```

2.7. Status method

Syntax

1) int Status ()
2) int Status (int type)

Parameter

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

Value	[IN/OUT]	Meaning	Setting range
Type	[IN]	Status type	CMP_STS_PRINTEROFF CMP_STS_PAPER_EMPTY CMP_STS_COVER_OPEN CMP_STS_PAPER_NEAREMPTY CMP_STS_DRAWER_LEVEL_H CMP_STS_ONPRESENTER

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.
When there is not a parameter, return the logical sum of the status (CMP_STS_COVER_OPEN, CMP_STS_PAPER_EMPTY, CMP_STS_PRINTEROFF) indicating the error of the printer.
When the status type is specified, return the status that matches. Status type can be specified in combination. If you want to combine, please specify the logical sum.

Return value

Return the following status codes.

Status codes	Description
CMP_STS_NORMAL (0)	The printer is normal.
CMP_STS_PRINTEROFF (128)	The printer is off-line.
CMP_STS_PAPER_EMPTY (32)	The printer is out of paper.
CMP_STS_COVER_OPEN (16)	The cover of the printer opens.
CMP_STS_PAPER_NEAREMPTY (4)	Paper near empty. (when the type parameter is set)
CMP_STS_DRAWER_LEVEL_H (2)	Status of pin 3 of drawer kick-out connector = H (when the type parameter is set)
CMP_STS_ONPRESENTER (1)	Status the paper is hold on the presenter or the paper exit sensor. (PMU3300 only, when the type parameter is set)

Example

```
int status = printer->Status();
if ( CMP_STS_NORMAL == status ) {
    // No Error
    int status2 = printer->Status(CMP_STS_PAPER_NEAREMPTY);
    if ( (CMP_STS_PAPER_NEAREMPTY & status2) > 0 ) {
        // Paper Near Empty
    }
} 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
    }
}

int status3 = printer->Status(CMP_STS_DRAWER_LEVEL_H | CMP_STS_ONPRESENTER);
if ( (CMP_STS_DRAWER_LEVEL_H & status3) > 0 ) {
    // Status of pin 3 of drawer kick-out connector = H
}
if ( (CMP_STS_ONPRESENTER & status3) > 0 ) {
    // Paper is hold on presenter or paper exit sensor
}
```

2.8. PrintText method

Syntax

int PrintText (std::string data, int alignment, int attribute, int textSize)

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.

Use UTF-8 for the character code of the text data.

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

```
printer->PrintText( "Print text data.\n",
          CMP_ALIGNMENT_CENTER,
          CMP_FNT_BOLD | CMP_FNT_UNDERLINE,
          CMP_TXT_2WIDTH | CMP_TXT_2HEIGHT );
```

2.9. PrintPaddingText method

Syntax

int PrintPaddingText (std::string data, int attribute, int textSize, int length, int side)

Parameter

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

Value	[IN/OUT]	Meaning	Setting range
data	[IN]	Text data	
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
length	[IN]		1 -
side	[IN]		CMP_SIDE_RIGHT: Right side of text data CMP_SIDE_LEFT: Left side of text data

Description

This method is used to print text with space padding which specifies attribute and size and length of the single-byte character equivalent and side where space is added.

Use UTF-8 for the character code of the text data. Cannot use the combining characters in the text data.

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

```
int nameSize = 24;              // Order name size
int priceSize = 7;              // Price size

// Line 1
printer->PrintPaddingText( "Sandwich",
        CMP_FNT_DEFAULT, CMP_TXT_1WIDTH, nameSize, CMP_SIDE_RIGHT );
printer->PrintPaddingText( "5.00",
        CMP_FNT_DEFAULT, CMP_TXT_1WIDTH, priceSize, CMP_SIDE_LEFT );
printer->PrintNormal("\n");

// Line 2
printer->PrintPaddingText( "Hamburg steak",
        CMP_FNT_DEFAULT, CMP_TXT_1WIDTH, nameSize, CMP_SIDE_RIGHT );
printer->PrintPaddingText( "12.00",
        CMP_FNT_DEFAULT, CMP_TXT_1WIDTH, priceSize, CMP_SIDE_LEFT );
printer->PrintNormal("\n");

// Line 3
printer->PrintPaddingText( "Coffee",
        CMP_FNT_DEFAULT, CMP_TXT_1WIDTH, nameSize, CMP_SIDE_RIGHT );
printer->PrintPaddingText( "2.00",
        CMP_FNT_DEFAULT, CMP_TXT_1WIDTH, priceSize, CMP_SIDE_LEFT );
printer->PrintNormal("\n");
```

2.10. PrintBitmap method

Syntax

1) int PrintBitmap (std::string fileName, int alignment)
2) int PrintBitmap (std::string fileName, int width, int alignment)

Parameter

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

Value	[IN/OUT]	Meaning	Setting range
fileName	[IN]	Bitmap file name	
width	[IN]	Bitmap width	CMP_BM_ASIS: Print the bitmap with one bitmap pixel per printer dot. Other Values: Bitmap width expressed (dots).
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 (dots).

Description

This method is used to print bitmap which specifies file name or bitmap and width and alignment and mode.
Printable bitmap formats are BMP / JPG / PNG / GIF.
If the bitmap width is omitted, printing in CMP_BM_ASIS.

Return value

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

Example

```
printer->PrintBitmap( "samplebitmap.bmp",
     CMP_BM_ASIS, CMP_ALIGNMENT_CENTER);
```

2.11. PrintMemoryBitmap method

Syntax

int PrintMemoryBitmap (char* data, int dataLength, int width, int alignment)

Parameter

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

Value	[IN/OUT]	Meaning	Setting range
data	[IN]	Bitmap data	
dataLength	[IN]	Bitmap data length	
width	[IN]	Bitmap width	CMP_BM_ASIS: Print the bitmap with one bitmap pixel per printer dot. Other Values: Bitmap width expressed (dots).
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 (dots).

Description

This method is used to print bitmap data and width and alignment and mode.
Printable bitmap formats are BMP / JPG / PNG / GIF.

Return value

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

Example

```
char* data;
int dataLength;

// Omitted create bitmap data.

printer->PrintMemoryBitmap(data, dataLength,
     CMP_BM_ASIS, CMP_ALIGNMENT_CENTER);
```

2.12. PrintNVBitmap method

Syntax

int PrintNVBitmap (int nvImageNumber)

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 the "POS Printer utility" of utility software for the printer.
Registration mode varies among the model of the printer. Please register as follows.

[CT-S281, PMU3300 Series]
Please register the logo with "Unused key code mode".
To the image number to use, it is necessary to register the logo sequentially.

[CT-D101/150/151, CT-E301/351/601/651,
CT-S251/281II/310II/601/651/751/801/851/601II/651II/801II/851II/4000/4500 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

```
printer->PrintNVBitmap( 1 );
```

2.13. PrintBarCode method

Syntax

int PrintBarCode (std::string data, int symbology, int height, int width, int alignment, int textPosition)

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
height	[IN]	Barcode height	1 - 255 (dots)
width	[IN]	Barcode horizontal size (magnification)	2 - 6 (dots)
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 (dots).
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.

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

```
printer->PrintBarCode( "123456789012",
          CMP_BCS_UPCA,
          50,
          2,
          CMP_ALIGNMENT_LEFT,
          CMP_HRI_TEXT_ABOVE );
```

2.14. PrintPDF417 method

Syntax

int PrintPDF417 (std::string data, int digits, int steps, int moduleWidth, int stepHeight, int ECLevel, int alignment)

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	2 - 8 (dots)
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 (dots).

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

```
printer->PrintPDF417(
          "http://www.citizen-systems.co.jp/printer/index.html",
          0, 0, 3, 3,
          CMP_PDF417_EC_LEVEL_0,
          CMP_ALIGNMENT_LEFT );
```

2.15. PrintQRCode method

Syntax

int PrintQRCode (std::string data, int moduleSize, int ECLevel, int alignment)

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	1 - 16 (dots)
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 (dots).

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

```
printer->PrintQRCode(
          "http://www.citizen-systems.co.jp/printer/index.html",
          4,
          CMP_QRCODE_EC_LEVEL_L,
          CMP_ALIGNMENT_LEFT );
```

2.16. CutPaper method

Syntax

int CutPaper (int type)

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

```
printer->CutPaper( CMP_CUT_PARTIAL_PREFEED );
```

2.17. UnitFeed method

Syntax

int UnitFeed (int ufCount)

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 (dots)	

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

```
printer->UnitFeed( 200 );
```

2.18. MarkFeed method

Syntax

int MarkFeed (int type)

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

```
printer->MarkFeed( CMP_MF_TO_CUTTER );
```

2.19. OpenDrawer method

Syntax

int OpenDrawer (int drawer, int pulseLen)

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 (x 100) msec

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

```
printer->OpenDrawer( CMP_DRAWER_1, 1 );
```

2.20. TransactionPrint method

Syntax

int TransactionPrint (int control)

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, PrintPaddingText, PrintBitmap, PrintMemoryBitmap, PrintNVBitmap, PrintBarCode, PrintPDF417, PrintQRCode, CutPaper, UnitFeed, MarkFeed, OpenDrawer, RotatePrint, PageModePrint, ClearePrintArea, 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

```
printer->TransactionPrint( CMP_TP_TRANSACTION );
printer->PrintNVBitmap( 1 );
printer->PrintBarCode( "123456789012", CMP_BCS_UPCA, 50, 2,
          CMP_ALIGNMENT_LEFT, CMP_HRI_TEXT_ABOVE );
printer->PrintText( "Line 1\n", CMP_ALIGNMENT_LEFT, CMP_FNT_DEFAULT,
          CMP_TXT_1WIDTH );
printer->PrintText( "Line 2\n", CMP_ALIGNMENT_LEFT, CMP_FNT_DEFAULT,
          CMP_TXT_1WIDTH );
printer->PrintText( "Line 3\n", CMP_ALIGNMENT_LEFT, CMP_FNT_DEFAULT,
          CMP_TXT_1WIDTH );
printer->PrintBarCode( "123456789012", CMP_BCS_UPCA, 50, 2, CMP_ALIGNMENT_LEFT,
          CMP_HRI_TEXT_ABOVE );
printer->PrintNVBitmap( 1 );
printer->CutPaper( CMP_CUT_PARTIAL_PREFEED );
printer->TransactionPrint( CMP_TP_NORMAL );
```

2.21. RotatePrint method

Syntax

int RotatePrint (int rotation)

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, PrintPaddingText, PrintNormal

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

PrintBarcod, PrintPDF417, PrintQRCode and/or PrintBitmap, PrintMemoryBitmap

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

```
printer->RotatePrint( CMP_RP_ROTATE180 | CMP_RP_BARCODE | CMP_RP_BITMAP );
printer->PrintBitmap( "samplebitmap.bmp", CMP_BM_ASIS,
          CMP_ALIGNMENT_CENTER );
printer->PrintBarCode( "123456789012", CMP_BCS_UPCA, 50, 2, CMP_ALIGNMENT_LEFT,
          CMP_HRI_TEXT_ABOVE );
printer->PrintText( "Line 3\n", CMP_ALIGNMENT_LEFT, CMP_FNT_DEFAULT,
          CMP_TXT_1WIDTH );
printer->PrintText( "Line 2\n", CMP_ALIGNMENT_LEFT, CMP_FNT_DEFAULT,
          CMP_TXT_1WIDTH );
printer->PrintText( "Line 1\n", CMP_ALIGNMENT_LEFT, CMP_FNT_DEFAULT,
          CMP_TXT_1WIDTH );
printer->CutPaper( CMP_CUT_PARTIAL_PREFEED );
printer->RotatePrint( CMP_RP_NORMAL );
```

2.22. PageModePrint method

Syntax

int PageModePrint (int control)

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, PrintPaddingText, PrintBitmap, PrintMemoryBitmap, 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
printer->PrintNormal( "\u001b|2vCSample 2 - Print\n");
printer.PrintText("123456789012345678901234567890123456789012345678901234567890123456
          7890123456789012345678901234567890123456789012345678901234567890\n",
          CMP_ALIGNMENT_RIGHT, CMP_FNT_DEFAULT,
          CMP_TXT_1WIDTH | CMP_TXT_1HEIGHT);
// Start of Page Mode
printer->PageModePrint( CMP_PM_PAGE_MODE );
// Set offset of Page Mode
printer->SetPageModeVerticalPosition( 0 );
printer->SetPageModeHorizontalPosition( 0 );
// Set direction of Page Mode
printer->SetPageModePrintDirection( CMP_PD_TOP_TO_BOTTOM );
// Set print area of Page Mode
printer->SetPageModePrintArea( "308,0,76,800" );
printer->PrintNormal( "\u001b|4C- Receipt -\n" );
// Set print area of Page Mode
printer->SetPageModePrintArea( "184,0,120,800" );
printer->PrintText( "   $ 299.99-  \n", CMP_ALIGNMENT_CENTER,
          CMP_FNT_UNDERLINE | CMP_FNT_BOLD,
          CMP_TXT_4WIDTH | CMP_TXT_4HEIGHT );
// Set print area of Page Mode
printer->SetPageModePrintArea( "88,0,88,560" );
printer->PrintText( "CITIZEN SYSTEMS\n", CMP_ALIGNMENT_RIGHT, CMP_FNT_DEFAULT,
          CMP_TXT_2WIDTH | CMP_TXT_3HEIGHT );
// Set print area of Page Mode
printer->SetPageModePrintArea( "0,0,88,480" );
printer->PrintBarCode( "123456789012", CMP_BCS_UPCA, 64, 4, CMP_ALIGNMENT_LEFT,
          CMP_HRI_TEXT_BELOW );
// Set print area of Page Mode
printer->SetPageModePrintArea( "0,600,192,192" );
printer->PrintQRCode( "http://www.citizen-systems.co.jp/", 5,
          CMP_QRCODE_EC_LEVEL_L, CMP_ALIGNMENT_LEFT );
// End of Page Mode
printer->PageModePrint( CMP_PM_NORMAL );
```

Print image

2.23. ClearPrintArea method

Syntax

int ClearPrintArea ()

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

```
printer->ClearPrintArea();
```

2.24. ClearOutput method

Syntax

int ClearOutput ()

Parameter

Not exist.

Description

This method is used to clear all buffered output data by TransactionPrint method 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

```
printer->ClearOutput();
```

2.25. PrintData method

Syntax

int PrintData (char* data, int dataLength)

Parameter

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

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

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.)
char* data = { 0x1b, 0x1e };
printer->PrintData( data, sizeof(data) );
```

2.26. PrintNormal method

Syntax

int PrintNormal (std::string data)

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. Use UTF-8 for the character code of the print data.
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|#lF	
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

```
printer->PrintNormal( "\u001b|4C- Receipt -\n" );
```

2.27. SetPrintCompletedTimeout method

Syntax

int SetPrintCompletedTimeout(int timeout)

Parameter

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

Value	[IN/OUT]	Meaning	Setting range
timeout	[IN]	Timeout of print completion notification (msec)	0: Automatically adjusts the timeout. Other Values: Specify the timeout. Expressed in milliseconds.

Description

This method is used to set the timeout to check the print completion notification.
When you create an instance, the timeout is initialized to 0.
Please refer to "3.1. Function to detect the completion of printing" for details of the function to detect the completion of printing.

Return value

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

Example

```
// Automatically adjusts
printer.SetPrintCompletedTimeout( 0 );

// Fixed 90sec
printer.SetPrintCompletedTimeout( 90000 );
```

2.28. GetVersionCode method

Syntax

int GetVersionCode ()

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

```
printer->GetVersionCode();
```

2.29. GetVersionName method

Syntax

std::string GetVersionName ()

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

```
printer->GetVersionName();
```

2.30. PrinterCheckEx method

Syntax

1) int PrinterCheckEx (int *status, int connectType, string addr)
2) int PrinterCheckEx (int *status, int connectType, string addr, int port)
3) int PrinterCheckEx (int *status, int connectType, string addr, int port, int timeout)

Parameter

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

Value	[IN/OUT]	Meaning	Setting range
status	[OUT]	Status codes	
connectType	[IN]	Connect type	CMP_PORT_SNMP
addr	[IN]	IP address to connect	SNMP: 0.0.0.0 - 255.255.255.255
port	[IN]	Connection port number	
timeout	[IN]	Timeout (msec)	

Description

This method is used to get the status of the printer from the unconnected state. Please specify the type and address of the printer connection. In this version, only supported the connection type CMP_PORT_SNMP.

Connection port number is valid only if you specify the connection type CMP_PORT_WiFi. It is ignored in this version, so please set any value if necessary.

Timeout is gives the maximum number of milliseconds to connect printer. If it is omitted, you connected with 4000 milliseconds.

The CMP_PORT_SNMP in the connect type can be used with printers connected to the network. By using this connection type, you can get the status regardless of other connections. In order to use this connection type, the printer supported with this function.

If the acquisition of the status is successful, set the following status code to the status parameters in the logical sum.

Status codes	Description
CMP_STS_NORMAL (0)	The printer is normal.
CMP_STS_PRINTEROFF (128)	The printer is off-line.
CMP_STS_PAPER_EMPTY (32)	The printer is out of paper.
CMP_STS_COVER_OPEN (16)	The cover of the printer opens.
CMP_STS_PAPER_NEAREMPTY (4)	Paper near empty.
CMP_STS_DRAWER_LEVEL_H (2)	Status of pin 3 of drawer kick-out connector = H

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. (4) The printer is in use by another connection. (Except SNMP)
CMP_E_CONNECT_NOTFOUND (1004)	Failed to check the support model after connecting to the printer. (1) The model is not supported.

Example

```
int status;
int result = printer->PrinterCheckEx(&status, CMP_PORT_SNMP,
                                     "192.168.182.100");
if ( CMP_SUCCESS == result ) {
    if ( CMP_STS_NORMAL == status ) {
        // Status Normal
    } 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
        }
        if ( (CMP_STS_PAPER_NEAREMPTY & status) > 0 ) {
            // Paper Near Empty
        }
        if ( (CMP_STS_DRAWER_LEVEL_H & status) > 0 ) {
            // Status of pin 3 of drawer kick-out connector = H
        }
    }
} else {
    // PrinterCheckEx Error
}
```

2.31. PageModeArea property

Syntax

std::string PageModeArea

Attribute

Read only

Description

This property holds the page area. Expressed in the unit of measure given by MapMode (default dots). 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-S281/281II Series] : "384,938"
[CT-D101/150/151 Series] : "576,1662"
[CT-E301/351/601/651 Series] : "576,1662"
[CT-S310II/601/651/801/851/601II/651II/801II/851II/801III/851III/751 Series] : "576,1662"
[CT-S4000/4500 Series] : "832,1662"
[PMU3300 Series] : "576,1662"

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

std::string GetPageModeArea()

Returns the page area as the return value.

2.32. PageModePrintArea property

Syntax

std::string PageModePrintArea

Attribute

Read/Write

Description

This property holds the print area of Page Mode. Expressed in the unit of measure given by MapMode (default dots). 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

int SetPageModePrintArea (std::string area)

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

std::string GetPageModePrintArea ()

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

2.33. PageModePrintDirection property

Syntax

int PageModePrintDirection

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

int SetPageModePrintDirection (int direction)

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

int GetPageModePrintDirection ()

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

2.34. PageModeHorizontalPosition property

Syntax

int PageModeHorizontalPosition

Attribute

Read/Write

Description

This property holds the horizontal start position offset within the Page Mode print area. Expressed in the unit of measure given by MapMode (default dots).
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

int SetPageModeHorizontalPosition (int position)

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

int GetPageModeHorizontalPosition ()

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

2.35. PageModeVerticalPosition property

Syntax

int PageModeVerticalPosition

Attribute

Read/Write

Description

This property holds the vertical start position offset within the Page Mode print area. Expressed in the unit of measure given by MapMode (default dots).
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

int SetPageModeVerticalPosition (int position)

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

int GetPageModeVerticalPosition ()

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

2.36. RecLineSpacing property

Syntax

int RecLineSpacing

Attribute

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 measure given by MapMode (default dots).
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

int SetRecLineSpacing (int spacing)

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

int GetRecLineSpacing ()

Returns the spacing of each single-high print line 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)

The timeout for checking the print completion notification is automatically adjusted according to the print data. Depending on the print data, the timeout error may occur each time. In that case, set the timeout with the SetPrintCompletedTimeout method according to the actual print time.

3.2. About printing UTF-8 encode characters

This SDK supports printing UTF-8 encoded characters.
This feature is focusing on providing a way to interoperate East Asian legacy double-byte character sets for Japanese, Korean, Simplified and Traditional Chinese.

Example

```
printer.setEncoding( "UTF-8" );
```

Supported models

Model	Firmware Version	Conditions
CT-S251	EM01-0304 or newer	*1
CT-S310II	DT00-1000 or newer DT10-1100 or newer	
CT-S601II	EE00-0200 or newer	*2
CT-S651II	EA00-0200 or newer	
CT-S801II	ED00-0200 or newer	
CT-S851II	DY00-0200 or newer	
CT-D101/150/151	All versions	*3
CT-E301/351/601/651		
CT-S751		
CT-S4500		

Note

*1 These models don't support interoperating East Asian legacy double-byte character sets for

Japanese, Korean, Simplified and Traditional Chinese. The available language for printing is depending on the region where the printer unit was purchased.

*2 These models don't support interoperating East Asian legacy double-byte character sets for Japanese, Korean, Simplified and Traditional Chinese. The available language for printing is depending on the encoding selected for the MSW9-4.

*3 These models support interoperating East Asian legacy double-byte character sets for Japanese, Korean, Simplified and Traditional Chinese. The printer picks up available characters one by one based on the language assigned for the MSW9-4 selection. Please note that this may result in an inconsistency of the font typeface.

Language and typeface (CT-D150/151, CT-E351/651 Series)

Language	Typeface
Japanese Korean	"Gothic" (Sans-serif)
Simplified Chinese Traditional Chinese	"Mincho" (Serif)

Language and typeface (CT-D101, CT-E301/601, CT-S751/4500 Series)

Language	Typeface
Japanese Korean Simplified Chinese Traditional Chinese	"Gothic" (Sans-serif)

3.3. Predefined Constants List

No	Type	Name	Data type	Value	Description
1	Result/Error	CMP_SUCCESS	int	0	Successfully completed
		CMP_E_CONNECTED	int	1001	Already connected
		CMP_E_DISCONNECT	int	1002	Not connected
		CMP_E_NOTCONNECT	int	1003	Failed to connect
		CMP_E_CONNECT_NOTFOUND	int	1004	Non supported model
		CMP_E_CONNECT_OFFLINE	int	1005	Failed printer status
		CMP_E_ILLEGAL	int	1101	Unsupported or invalid parameter
		CMP_E_OFFLINE	int	1102	Off-line
		CMP_E_NOEXIST	int	1103	File does not exist
		CMP_E_FAILURE	int	1104	Process failure
		CMP_E_TIMEOUT	int	1105	Timeout
		CMP_EPTR_COVER_OPEN	int	1201	Cover opens
		CMP_EPTR_REC_EMPTY	int	1202	Out of paper
		CMP_EPTR_BADFORMAT	int	1203	Unsupported file format
		CMP_EPTR_CMP_EPTR_TOOBIG	int	1204	Bitmap size too big
2	Connection interface	CMP_PORT_WiFi	int	0	Network
		CMP_PORT_USB	int	3	USB
		CMP_PORT_SNMP	int	6	SNMP
3	Status	CMP_STS_NORMAL	int	0	Normal
		CMP_STS_DRAWER_LEVEL_H	int	2	Pin 3 of drawer kick-out connector = H
		CMP_STS_PAPER_NEAREMPTY	int	4	Paper near empty
		CMP_STS_COVER_OPEN	int	16	Cover opens
		CMP_STS_PAPER_EMPTY	int	32	Paper empty
		CMP_STS_PRINTEROFF	int	128	Off-line
4	Alignment	CMP_ALIGNMENT_LEFT	int	0	Left alignment
		CMP_ALIGNMENT_CENTER	Int	1	Center alignment
		CMP_ALIGNMENT_RIGHT	int	2	Right alignment
5	Text attribute	CMP_FNT_DEFAULT	int	0	Default font
		CMP_FNT_FONTB	int	1	Font B
		CMP_FNT_FONTC	int	2	Font C
		CMP_FNT_BOLD	int	8	Bold
		CMP_FNT_REVERSE	int	64	Reverse
		CMP_FNT_UNDERLINE	int	128	Underline
6	Text size	CMP_TXT_1WIDTH	int	0	1 times width
		CMP_TXT_2WIDTH	int	16	2 times width
		CMP_TXT_3WIDTH	int	32	3 times width
		CMP_TXT_4WIDTH	int	48	4 times width
		CMP_TXT_5WIDTH	int	64	5 times width
		CMP_TXT_6WIDTH	int	80	6 times width
		CMP_TXT_7WIDTH	int	96	7 times width
		CMP_TXT_8WIDTH	int	112	8 times width
		CMP_TXT_1HEIGHT	int	0	1 times height
		CMP_TXT_2HEIGHT	int	1	2 times height
		CMP_TXT_3HEIGHT	int	2	3 times height
		CMP_TXT_4HEIGHT	int	3	4 times height
		CMP_TXT_5HEIGHT	int	4	5 times height
		CMP_TXT_6HEIGHT	int	5	6 times height
		CMP_TXT_7HEIGHT	int	6	7 times height

		CMP_TXT_8HEIGHT	int	7	8 times height
7	Side	CMP_SIDE_RIGHT	int	0	Right side
		CMP_SIDE_LEFT	int	1	Left side
8	Bitmap width	CMP_BM_ASIS	int	-11	One bitmap pixel per printer dot
9	Barcode symbology	CMP_BCS_UPCA	int	101	UPC-A
		CMP_BCS_UPCE	int	102	UPC-E
		CMP_BCS_EAN8	int	103	EAN8
		CMP_BCS_EAN13	int	104	EAN13
		CMP_BCS_JAN8	int	105	JAN8
		CMP_BCS_JAN13	int	106	JAN13
		CMP_BCS_ITF	int	107	Interleaved 2 of 5
		CMP_BCS_Codabar	int	108	Codabar
		CMP_BCS_Code39	int	109	Code39
		CMP_BCS_Code93	int	110	Code93
		CMP_BCS_Code128	int	111	Code128
		CMP_BCS_GS1DATABAR	int	131	GS1 DataBar Omnidirectional
		CMP_BCS_GS1DATABAR_E	int	132	GS1 DataBar Expanded
		CMP_BCS_GS1DATABAR_S	int	133	GS1 DataBar Stacked
		CMP_BCS_GS1DATABAR_E_S	int	134	GS1 DataBar Expanded Stacked
10	HRI characters	CMP_HRI_TEXT_NONE	int	0	None
		CMP_HRI_TEXT_ABOVE	int	1	Above the barcode
		CMP_HRI_TEXT_BELOW	int	2	Below the barcode
11	Error correction level (PDF417)	CMP_PDF417_EC_LEVEL_0	int	48	Level 0
		CMP_PDF417_EC_LEVEL_1	int	49	Level 1
		CMP_PDF417_EC_LEVEL_2	int	50	Level 2
		CMP_PDF417_EC_LEVEL_3	int	51	Level 3
		CMP_PDF417_EC_LEVEL_4	int	52	Level 4
		CMP_PDF417_EC_LEVEL_5	int	53	Level 5
		CMP_PDF417_EC_LEVEL_6	int	54	Level 6
		CMP_PDF417_EC_LEVEL_7	int	55	Level 7
		CMP_PDF417_EC_LEVEL_8	int	56	Level 8
12	Error correction level (QR Code)	CMP_QRCODE_EC_LEVEL_L	int	48	Level L (7%)
		CMP_QRCODE_EC_LEVEL_M	int	49	Level M (15%)
		CMP_QRCODE_EC_LEVEL_Q	int	50	Level Q (25%)
		CMP_QRCODE_EC_LEVEL_H	int	51	Level H (30%)
13	Cut type	CMP_CUT_FULL	int	-1	Full cut
		CMP_CUT_PARTIAL	int	-2	Partial cut
		CMP_CUT_FULL_PREFEED	int	-3	Feed and full cut
		CMP_CUT_PARTIAL_PREFEED	int	-4	Feed and partial cut
14	Mark feed type	CMP_MF_TO_CUTTER	int	2	Feed and cut
		CMP_MF_TO_NEXT_TOF	int	8	Feed to the next top
15	Drawer number	CMP_DRAWER_1	int	1	Drawer 1
		CMP_DRAWER_2	int	2	Drawer 2
16	Transaction control	CMP_TP_TRANSACTION	int	11	Begin transaction
		CMP_TP_NORMAL	int	12	End transaction
17	Rotation control	CMP_RT_NORMAL	int	0x0001	End rotation
		CMP_RT_ROTATE180	int	0x0103	Begin upside-down rotation
		CMP_RP_BARCODE	int	0x1000	Begin barcode rotation
		CMP_RP_BITMAP	int	0x2000	Begin bitmap rotation
18	Page mode control	CMP_PM_PAGE_MODE	int	1	Begin page mode
		CMP_PM_PRINT_SAVE	int	2	Print and save canvas
		CMP_PM_NORMAL	int	3	Print and exit page mode

		CMP_PM_CANCEL	int	4	Cancel page mode
19	Page mode direction	CMP_PD_LEFT_TO_RIGHT	int	1	Normal printing
		CMP_PD_BOTTOM_TO_TOP	int	2	Rotated left 90° printing
		CMP_PD_RIGHT_TO_LEFT	int	3	Upside down printing
		CMP_PD_TOP_TO_BOTTOM	int	4	Rotated right 90° printing

CITIZEN Linux POS Print SDK - Programming Manual

November 21, 2023 For Ver. 1.10

CITIZEN SYSTEMS JAPAN CO., LTD.

https://csj.citizen.co.jp/