CITIZEN

UWP POS Print SDK

Programming Manual

For Ver. 2.00

CITIZEN SYSTEMS JAPAN CO., LTD.

Revision Record

Date	Version	Description
May 22 2017	1.00	New issue.
May 8, 2019	1.01	- Added CT-S751/CT-S4500 to supported models. - Added “3.3 About printing UTF-8 encode characters”.
Jul. 28, 2021	2.00	- Added CT-D101/CT-E301/CT-E601 and CT-D151-L/CT-E651-L to supported models. - SDK supports USB host interface and line display or barcode scanner which is connected to it. - Added SetPrintCompletedTimeout method. - Unused constants were removed.
Nov. 21, 2023		Added CT-S801III and CT-S851III to the printer support models. (Page 8, 15, 17, 40, 60, 63, 72) Added DSP01-LT2 and DSP02-LS2 to the display support models. (Page 18)

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

Microsoft, Windows, Visual Studio, Visual Basic, Visual C#, and Visual C++ are registered trademarks of Microsoft Corporation in the United States and/or other countries. (Official name for Windows is Microsoft Windows Operating System.)

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

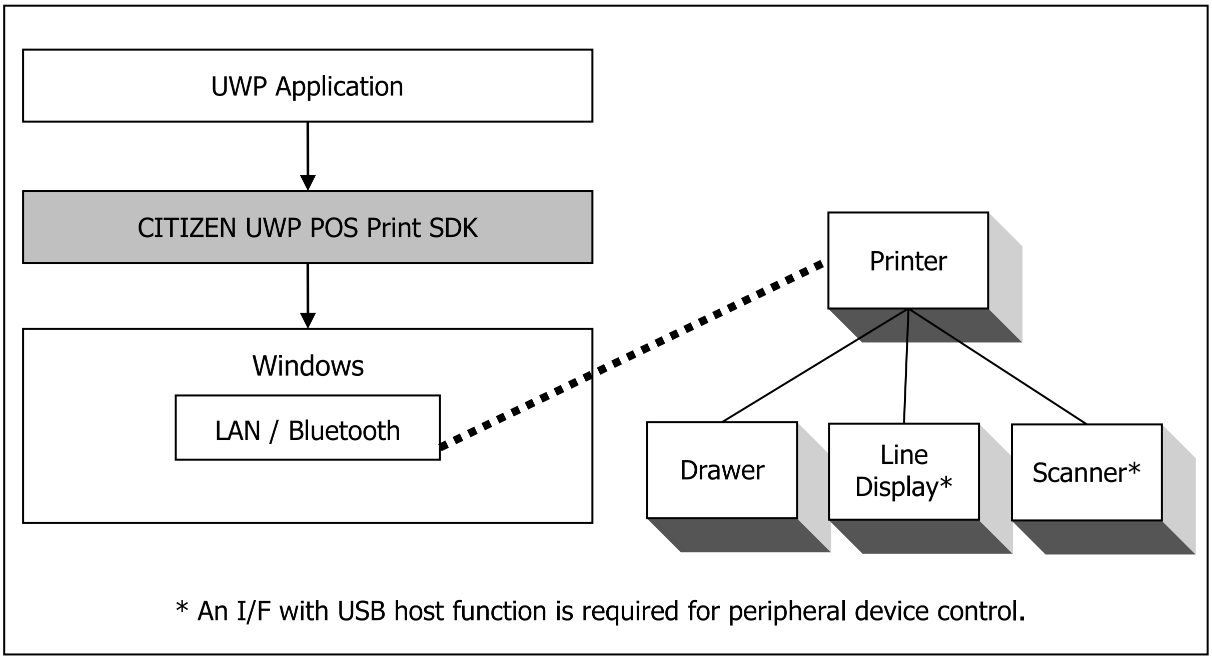
This document is a programming manual for the CITIZEN UWP POS Print SDK.

1.1. Document target range

This document is intended for developers who integrate their UWP (Universal Windows Platform) application programs with CITIZEN POS printers.

1.2. System summary

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


UWP Application
CITIZEN UWP POS Print SDK
Windows
LAN / Bluetooth
Printer
Drawer
Line Display*
Scanner*
* An I/F with USB host function is required for peripheral device control.

System diagram of the SDK

<u>SDK files</u>

This SDK is structured by some files included in the VSIX installer package. Visual Studio will locate these files automatically when an application is built.

<u>Supported operating systems</u>

This SDK supports the following Microsoft Windows operating systems.
- Windows 10 (x86 or x64. ARM is not supported).

<u>Supported Integrated Development Environment.</u>

This SDK supports the following Integrated Development Environment.
- Visual Studio 2015 Update 1 or later.

1.3. 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 Series	CT-D101	Wired LAN	Standard
CT-D150 Series	CT-D150	Wired LAN	Standard
CT-D151 Series	CT-D151	Wired/Wireless LAN, Bluetooth	Standard
	CT-D151-L		Blackmark paper is supported
CT-E301 Series	CT-E301	Wired LAN	Standard
CT-E351 Series	CT-E351	Wired LAN	Standard
CT-E601 Series	CT-E601	Wired/Wireless LAN, Bluetooth	Standard
CT-E651 Series	CT-E651	Wired/Wireless LAN, Bluetooth	Standard
	CT-E651-L		Blackmark paper is supported
CT-S251 Series	CT-S251	Wired/Wireless LAN, Bluetooth	Standard
CT-S281 Series	CT-S281BT/281BD	Bluetooth	Standard
CT-S310II Series	CT-S310II	Wired LAN	Standard
CT-S601/651/801/851 Series	CT-S601/651/801/851	Wired/Wireless LAN	Standard
	CT-S801/851-M		Blackmark paper is supported.
	CT-S801-L		Label paper is supported.
CT-S601II/651II/801II/851II Series	CT-S601II/651II/801II/851II	Wired/Wireless LAN, Bluetooth	Standard
	CT-S801II/851II-M		Blackmark paper is supported.
	CT-S801II-L		Label paper is supported.
CT-S801III/851III Series	CT-S801III/851III	Wired/Wireless LAN, Bluetooth	Standard
CT-S751 Series	CT-S751	Wired/Wireless LAN, Bluetooth	Standard
CT-S2000 Series	CT-S2000	Wired LAN	Standard
	CT-S2000-M		Blackmark paper is supported.
	CT-S2000-L		Label paper is supported.
CT-S4000 Series	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.
CT-S4500 Series	CT-S4500	Wired/Wireless LAN, Bluetooth	Standard (Label/Blackmark paper is supported.)

1.4. Printer setting

It is the prerequisite for the use of this SDK that the memory switch of the printer is 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
6-1	Act. For Driver	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
6-1	Act. For Driver	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
6-1	Act. For Driver	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
13-6	Auto Reconnect (When Bluetooth I/F is used)	Invalid

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
6-1	Act. For Driver	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	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 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
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	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
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
13-6	Auto Reconnect (When Bluetooth I/F is used)	Invalid

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	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 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
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
13-6	Auto Reconnect (When Bluetooth I/F is used)	Invalid

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
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)
13-6	Auto Reconnect (When Bluetooth I/F is used)	Invalid

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
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)
13-6	Auto Reconnect (When Bluetooth I/F is used)	Invalid

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
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-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
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
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
3-7	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
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 Timing	Before BUSY
13-6	Auto Reconnect (When Bluetooth I/F is used)	Invalid

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
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
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 Timing	Before BUSY
13-6	Auto Reconnect (When Bluetooth I/F is used)	Invalid

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
6-1	Act. For Driver	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
13-6	Auto Reconnect (When Bluetooth I/F is used)	Invalid

CT-S2000 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	Disabled
5-2	Line Pitch	1/360
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-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
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
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 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
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
13-6	Auto Reconnect (When Bluetooth I/F is used)	Invalid

*1 MSW No.9-1~4 in the previous tables are the settings to use Japanese. Please set up them for your environment.

*2 CT-D101/150/151, CT-E301/351/601/651, CT-S601II/651II/801II/851II/801III/851III/751/4500 series can be set up Multi-byte character as Shift_JIS, GB18030, EUC-KR or BIG5. Please select from them for your environment.

Firmware

The firmware version of the printer must be the followings to use this SDK with CT-S601/651/801/851 Series.

The firmware of a printer with older than followings, 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. Supported Peripheral Devices models

The models of peripheral devices applicable for control with this service are as follows.
For details on the functions of each model, refer to the instruction manual of each peripheral device.
Network I/F or Bluetooth I/F with USB host function is required for peripheral device control.

[Line Display]

Applicable Display	I/F	Product Specification Overview
DSP01-LT/DSP01-LT2	USB	TFT line display
DSP02-LS/DSP02-LS2	USB	STN line display

[Barcode Scanner]

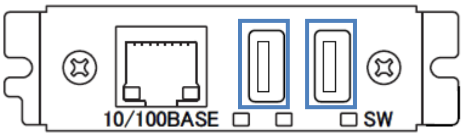
Applicable Display	I/F	Product Specification Overview
SCN01-Z1D	USB	1D barcode scanner
SCN02-Z2D	USB	2D barcode scanner
BC-NL3000U	USB	2D barcode scanner

Check that the scanner to be connected is set as follows. For the setting procedure, refer to the instruction manual of the peripheral device to be used.

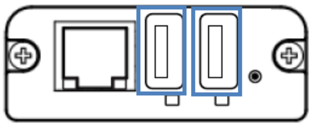
Item	Value	Description
Interface	USB HID Class	Communication protocol
Keyboard	US Keyboard	Keyboard language
Terminator	Enter	Data suffix

About connection to printer

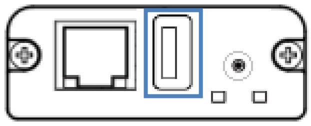
For the connection to the applicable peripheral device, first turn off the printer power and then connect to a USB port of the corresponding interface shown in the figure below. Next, turn on the printer power, wait about 30 seconds until the applicable peripheral device becomes ready to use so as to ensure stable operation, and then execute the control start process of the peripheral device.



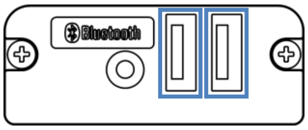
IF1-EFX2/IF1-WFx6



IF2-EFX2/IF2-WFx6



IF2-WFx5



IF2-BT03/IF2-BT04

The following lists prohibited actions that must not be performed with regard to a peripheral device connection.

Prohibited Actions

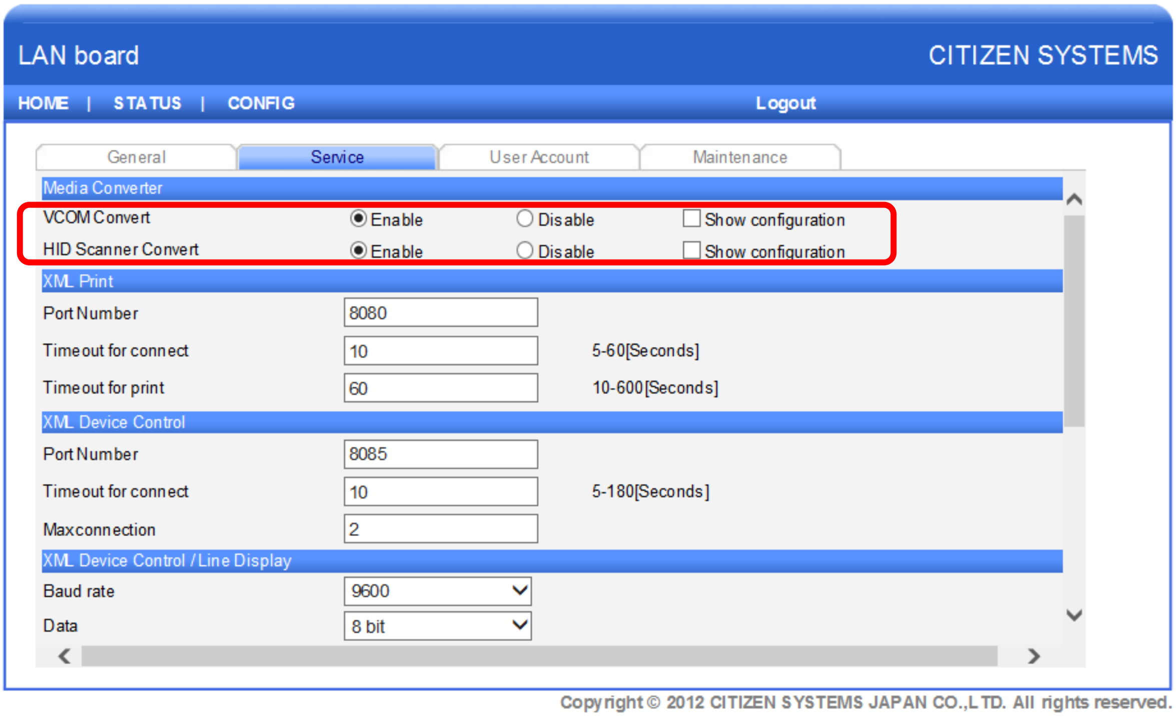
- Connecting other than a supported peripheral device (USB hub, smartphone, etc.) to a USB port of the interface.
- Inserting and removing the cable connector of the peripheral device into/from a USB port of the interface while the printer power is on.
- Connecting multiple peripheral devices of the same type to a USB port of the interface (e.g. connecting two displays).

If any of the above actions is performed, it may lead to the misoperation and, in the worst case, cause a failure of the printer or connected peripheral device.

About the Network I/F setting

When using the line display and the barcode scanner with the Network I/F, it is necessary to change the setting related to the service. For the basic operation, refer to the instruction manual of the interface board of the printer.

Please connect to each printer from web browser and display the following Service screen. You can set the services provided by the printer.



Media Converter

VCOM Convert ● Enable ○ Disable ☐ Show configuration

HID Scanner Convert ● Enable ○ Disable ☐ Show configuration

XML Print

Port Number 8080

Timeout for connect 10 5-60[Seconds]

Timeout for print 60 10-600[Seconds]

XML Device Control

Port Number 8085

Timeout for connect 10 5-180[Seconds]

Maxconnection 2

XML Device Control / Line Display

Baud rate 9600

Data 8 bit

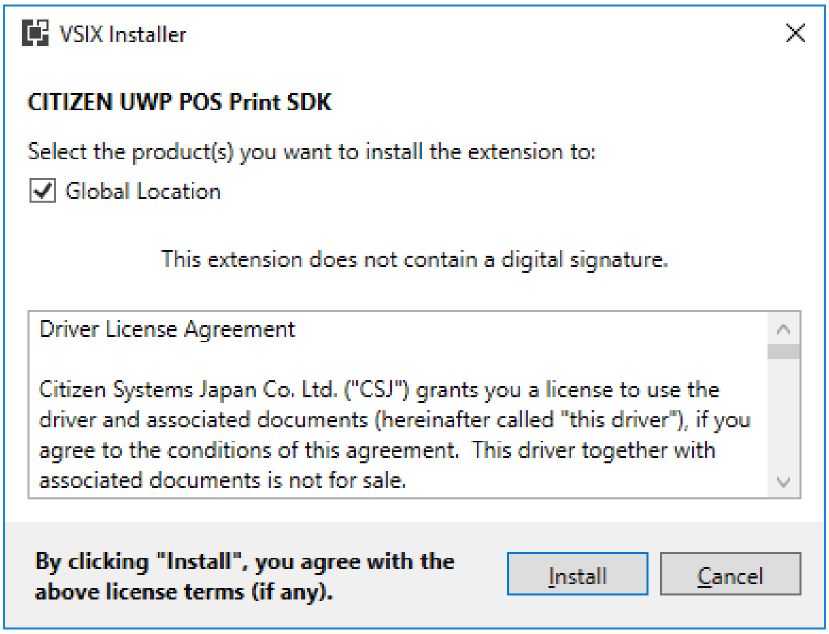
Select "Enable" of "VCOM Converter" and "HID Scanner Convert" with reference to the inside of the red frame. Then scroll to the bottom and press the "Submit" button.
Finally, press the "Save & Reboot" button on the "Maintenance" tab, select "Yes", and when the buzzer beeps from the printer, the setting is completed

When checking "Show configuration" in the above red frame, the setting screen of "Media Converter Configuration / VCOM Convert" is displayed, but since it already has an appropriate value for the corresponding display, it is not changed by normal use Please do.

Each setting value holds the value even when the power is turned off. When factory default setting (Factor Default) processing is done, set each setting value to the initial value.

1.6. Install instruction

1. Close Visual Studio if you are using it.
2. Open "CITIZEN_UWP_POS_SDK###.vsix" (the installer, where ### represents its version).
3. Click "Install" on the "VSIX Installer" window.



CITIZEN UWP POS Print SDK

Select the product(s) you want to install the extension to:

☑ Global Location

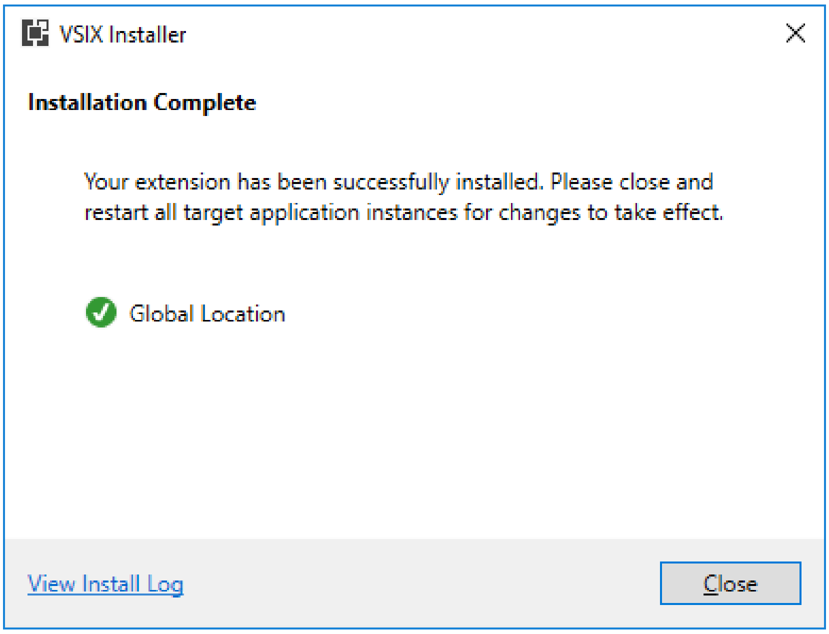
This extension does not contain a digital signature.

Driver License Agreement

Citizen Systems Japan Co. Ltd. ("CSJ") grants you a license to use the driver and associated documents (hereinafter called "this driver"), if you agree to the conditions of this agreement. This driver together with associated documents is not for sale.

By clicking "Install", you agree with the above license terms (if any).

4. Click "Close"

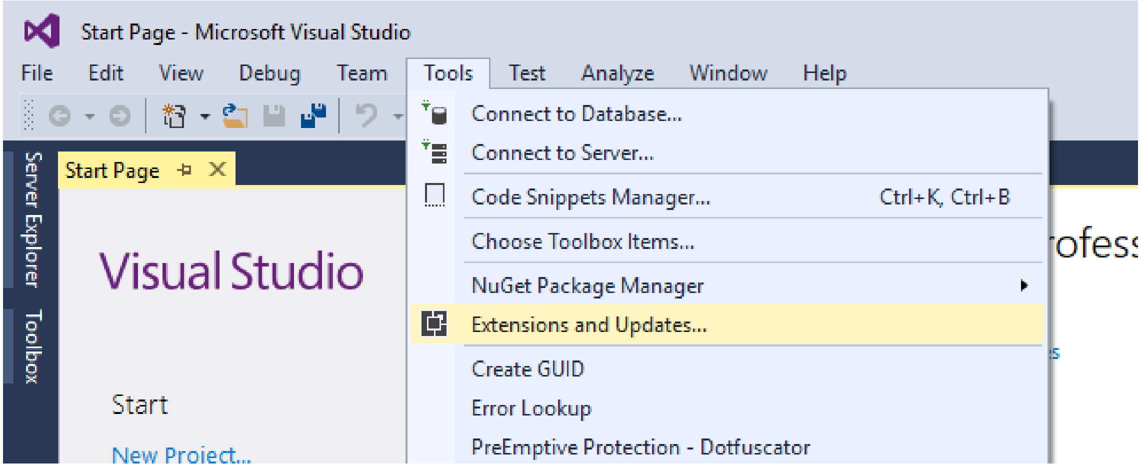


Installation Complete

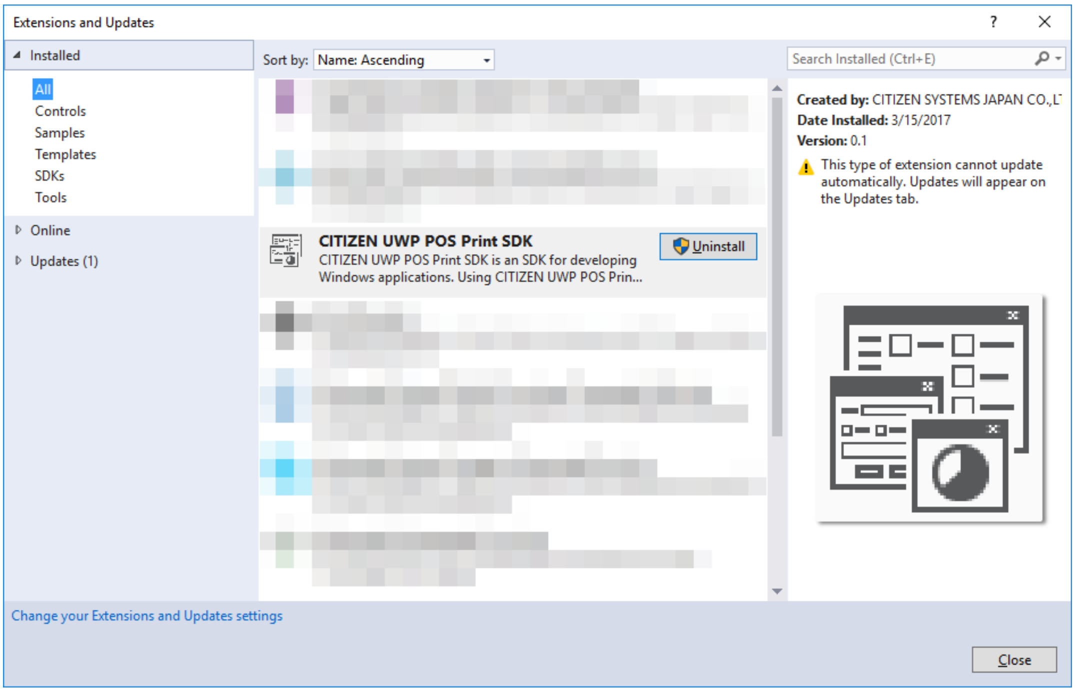
Your extension has been successfully installed. Please close and restart all target application instances for changes to take effect.

✓ Global Location

5. Start Visual Studio and trace menu "Tools", "Extensions and Updates..." (See below),



and the "Extensions and Updates..." dialog will be opened.



CITIZEN UWP POS Print SDK
CITIZEN UWP POS Print SDK is an SDK for developing Windows applications. Using CITIZEN UWP POS Prin...

Created by: CITIZEN SYSTEMS JAPAN CO.,L
Date Installed: 3/15/2017
Version: 0.1

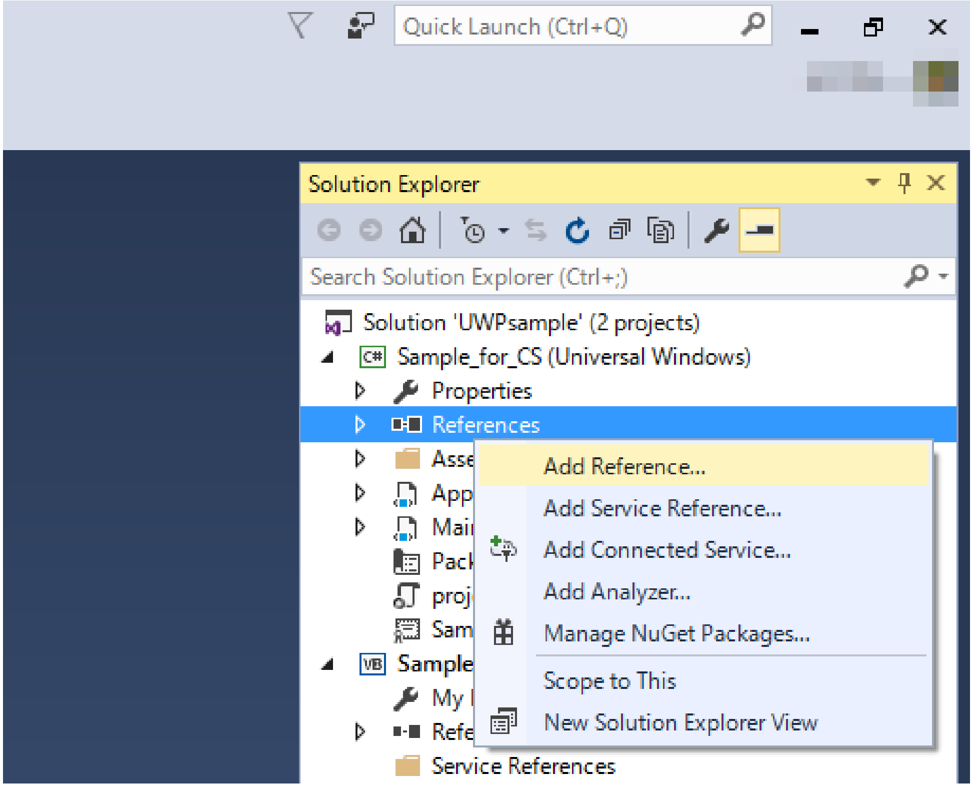
This type of extension cannot update automatically. Updates will appear on the Updates tab.

If "CITIZEN UWP POS Print SDK" is listed in this dialog, UWP POS Print SDK is installed successfully.

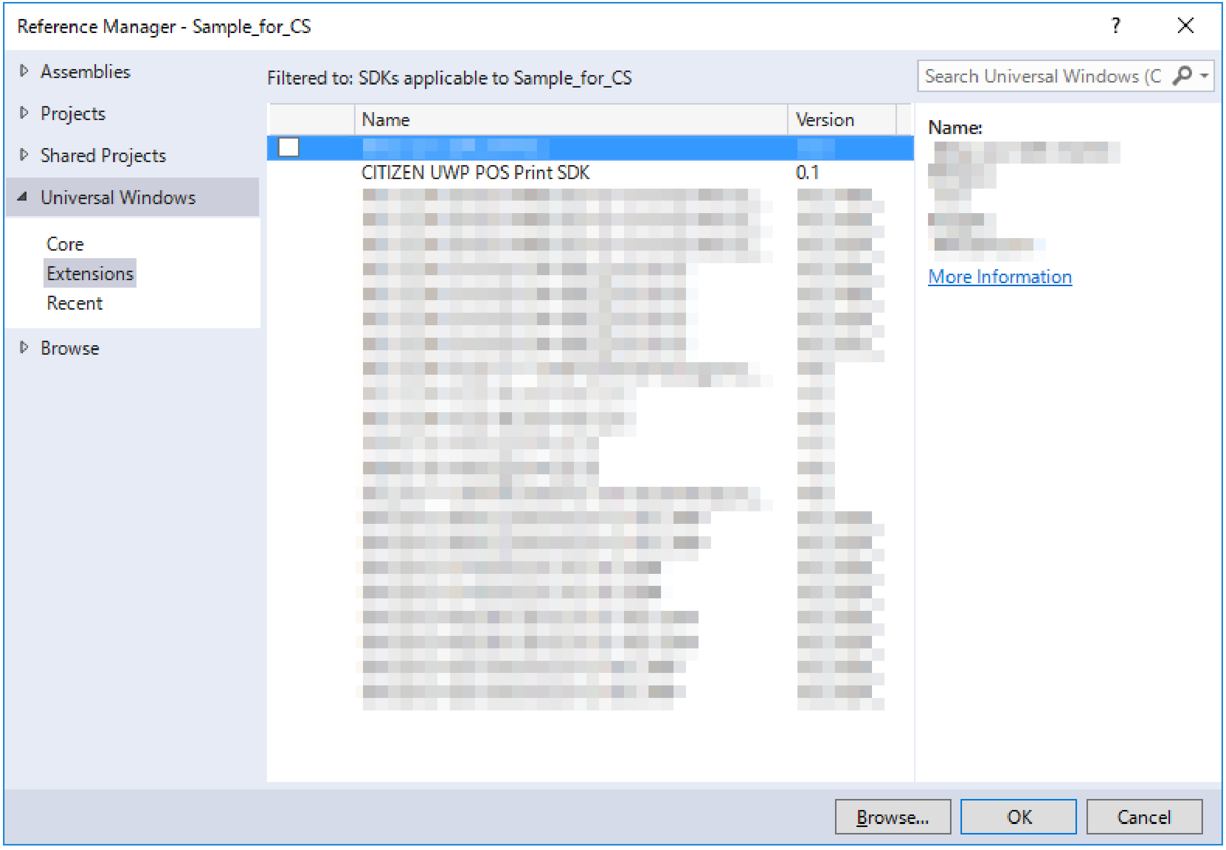
1.7. Definition method

Adding Library

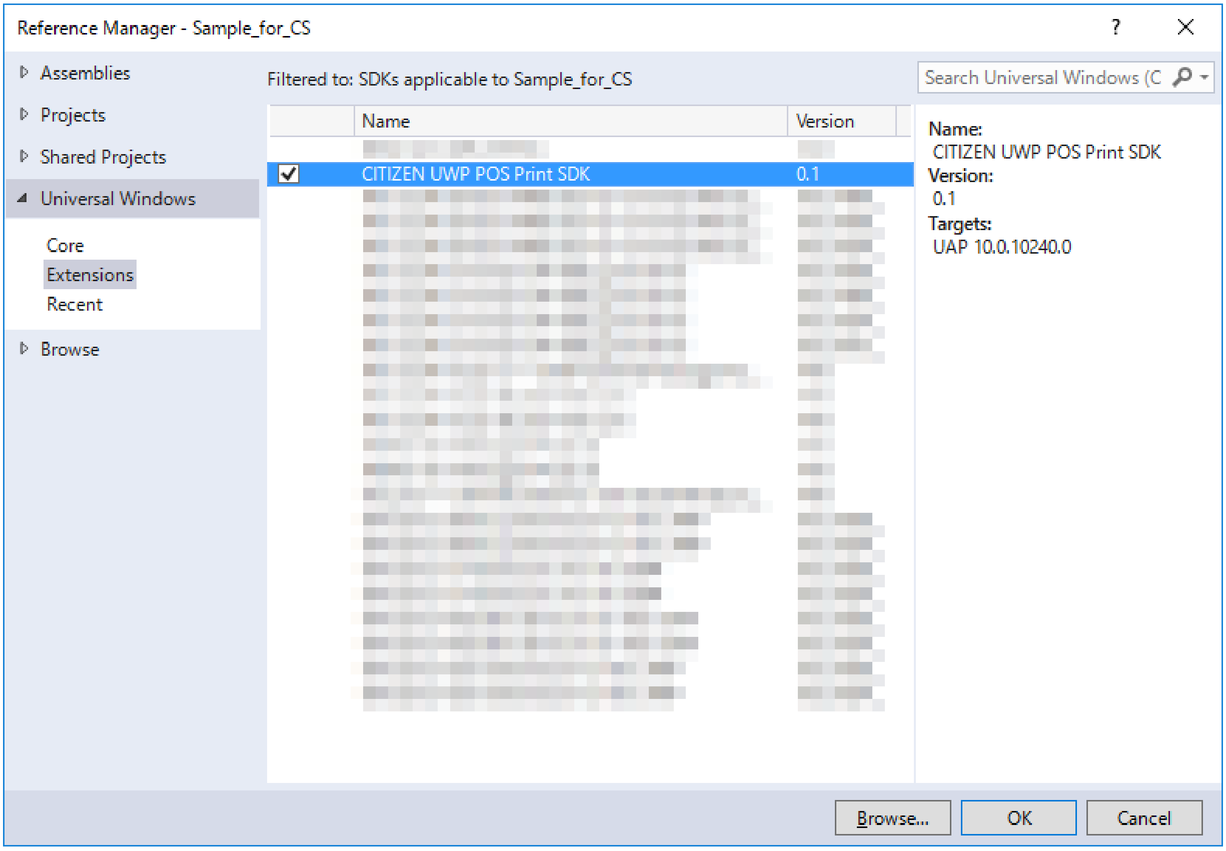
In your Universal Windows application project in the Solution Explorer window, right-click the "References", and click "Add Reference...".



Then the Reference Manager (see below) will open.


Reference Manager - Sample_for_CS
?
Assemblies
Projects
Shared Projects
Universal Windows
Core
Extensions
Recent
Browse
Filtered to: SDKs applicable to Sample_for_CS
Search Universal Windows (C
Name
Version
CITIZEN UWP POS Print SDK
0.1
Name:
More Information
Browse...
OK
Cancel

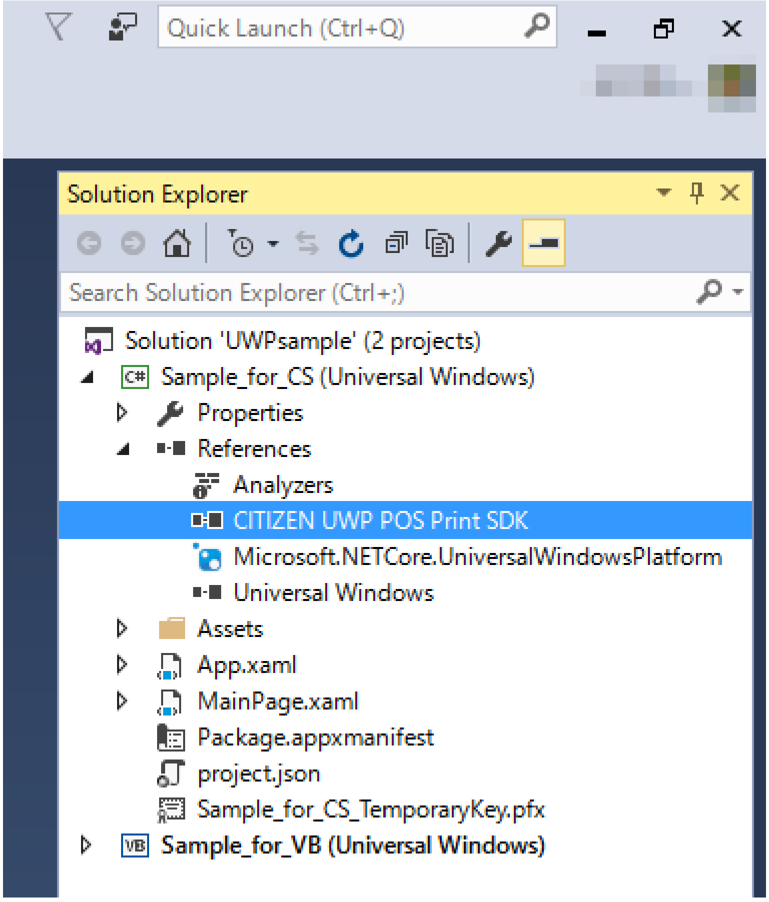
Click "Extensions" on the left side tree, and installed extensions are listed in the Reference Manager.


Reference Manager - Sample_for_CS
?
Assemblies
Projects
Shared Projects
Universal Windows
Core
Extensions
Recent
Browse
Filtered to: SDKs applicable to Sample_for_CS
Search Universal Windows (C
Name
Version
Browse...
OK
Cancel

	Name	Version
☑	CITIZEN UWP POS Print SDK	0.1

Name:
CITIZEN UWP POS Print SDK
Version:
0.1
Targets:
UAP 10.0.10240.0

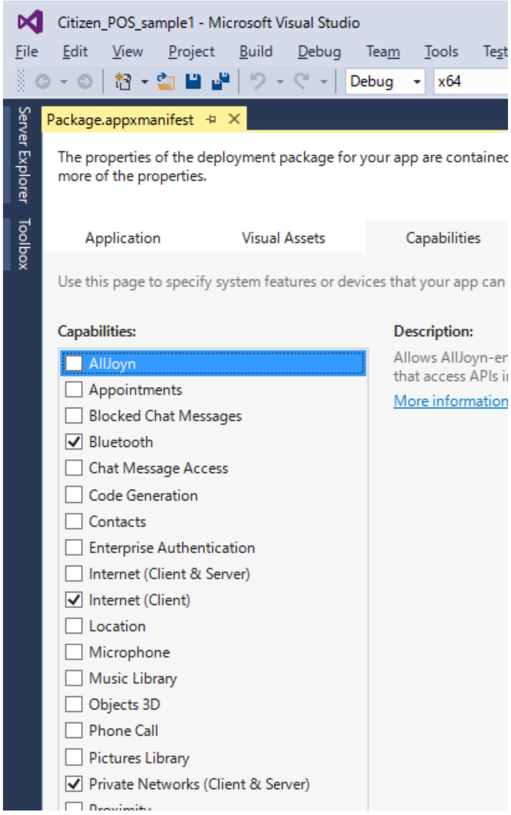
Check "CITIZEN UWP POS Print SDK" in this list and click "OK".


Quick Launch (Ctrl+Q)
Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'UWPsample' (2 projects)
Sample_for_CS (Universal Windows)
Properties
References
Analyzers
CITIZEN UWP POS Print SDK
Microsoft.NETCore.UniversalWindowsPlatform
Universal Windows
Assets
App.xaml
MainPage.xaml
Package.appxmanifest
project.json
Sample_for_CS_TemporaryKey.pfx
Sample_for_VB (Universal Windows)

Then CITIZEN UWP POS Print SDK will be registered in the project. You can find it in the Solution Explorer.

Configure Capabilities

Open Package.appxmanifest in the Solution Explorer and open "Capabilities" tab (see below), check "Bluetooth", "Internet (Clients)" and "Private Networks (Client & Server)".



The properties of the deployment package for your app are containe
more of the properties.

Use this page to specify system features or devices that your app can

Capabilities:

☐ AllJoyn
☐ Appointments
☐ Blocked Chat Messages
☑ Bluetooth
☐ Chat Message Access
☐ Code Generation
☐ Contacts
☐ Enterprise Authentication
☐ Internet (Client & Server)
☑ Internet (Client)
☐ Location
☐ Microphone
☐ Music Library
☐ Objects 3D
☐ Phone Call
☐ Pictures Library
☑ Private Networks (Client & Server)

Description:

Allows AllJoyn-er
that access APIs i

Adding Namespace

A reference to the name space "com.citizen.sdk" must be stated at the top of the program source code.

In the case of C#:

```
using com.citizen.sdk;
```

In the case of Basic:

```
Imports com.citizen.sdk
```

2. Printer Control

2.1. Program structure

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

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

// Connect printer
int result = await printer.ConnectAsync (
                       ESCPOSConst.CMP_PORT_WiFi, "192.168.123.45");

if (ESCPOSConst.CMP_SUCCESS == result)
{
    // Set encoding
    printer.SetEncoding(437);

    // Start Transaction ( Batch )
    await printer.TransactionPrintAsync(ESCPOSConst.CMP_TP_TRANSACTION);

    // Print Text
    await printer.PrintTextAsync("Citizen_POS_sample1_CS\n\n",
        ESCPOSConst.CMP_ALIGNMENT_CENTER,ESCPOSConst.CMP_FNT_DEFAULT,
        ESCPOSConst.CMP_TXT_1WIDTH | ESCPOSConst.CMP_TXT_1HEIGHT);
    await printer.PrintTextAsync("- Sample Print 1 -\n",
        ESCPOSConst.CMP_ALIGNMENT_CENTER, ESCPOSConst.CMP_FNT_DEFAULT,
        ESCPOSConst.CMP_TXT_1WIDTH | ESCPOSConst.CMP_TXT_2HEIGHT);
    await printer.PrintTextAsync("123456789012345678901234567890\n",
        ESCPOSConst.CMP_ALIGNMENT_RIGHT, ESCPOSConst.CMP_FNT_DEFAULT,
        ESCPOSConst.CMP_TXT_1WIDTH | ESCPOSConst.CMP_TXT_1HEIGHT);

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

    // Partial Cut with Pre-Feed
    await printer. CutPaperAsync(ESCPOSConst.CMP_CUT_PARTIAL_PREFEED);

    // End Transaction ( Batch )
    result = await printer.TransactionPrintAsync(
                          ESCPOSConst.CMP_TP_NORMAL);

    // Disconnect
    await printer.DisconnectAsync ();

    if (ESCPOSConst.CMP_SUCCESS != result)
    {
        // Print process Error
        MessageDialog msgbox = new MessageDialog(
            "Transaction Error : " + result.ToString(),
            "Citizen_POS_sample1" );
        await msgbox.ShowAsync();
    }
}
else
{
    // Connect Error
    MessageDialog msgbox = new MessageDialog(
        "Connect Error : " + result.ToString(),
        "Citizen_POS_sample1" );
    await msgbox.ShowAsync();
}
```

Class definition

Connect

Print process

Disconnect

2.2. Functions list

This SDK provides the following functions.

Methods list

No	Function	Detail
1	Create class (Constructor)	This is constructor method.
2	Connect printer (ConnectAsync method)	Connect to the printer.
3	Disconnect printer (DisconnectAsync method)	Disconnect the printer connection.
4	Set encoding (SetEncoding method)	Set the encoding of character.
5	Check printer status (PrinterCheckAsync method)	Sends command for status check of the printer.
6	Get printer status (Status method)	Get the status of the printer.
7	Print text (PrintTextAsync method)	Prints a text data.
8	Print bitmap (PrintBitmapAsync method)	Prints a bitmap file. (BMP/JPG/PNG/GIF format)
9	Print NV bitmap (PrintNVBitmapAsync method)	Prints a bitmap image that is stored in the flash memory.
10	Print BarCode (PrintBarCodeAsync method)	Prints a one-dimensional barcode.
11	Print PDF-417 (PrintPDF417Async method)	Prints a PDF417 barcode.
12	Print QRcode (PrintQRCodeAsync method)	Prints a QRCode barcode.
13	Print 2D GS1DataBar (PrintGS1DataBarStackedAsync method)	Prints a 2-dimensional GS1DataBar barcode.
14	Cut paper (CutPaperAsync method)	Cuts the paper.
15	Feed dot units (UnitFeedAsync method)	Feeds the paper forward by dot units.
16	Feed mark (MarkFeedAsync method)	Support for label / black mark paper.
17	Open drawer (OpenDrawerAsync method)	Opens the drawer.
18	Transaction print (TransactionPrintAsync method)	Enters or exits transaction mode.
19	Rotate print (RotatePrintAsync method)	Enters or exits rotated print mode. (180°)
20	PageMode print (PageModePrintAsync method)	Enters or exits page mode.
21	PageMode clear print area (ClearPrintArea method)	Clear the area of the page mode print area.
22	Clear output data (ClearOutputAsync method)	Clears all buffered output data. (data and printer buffer)
23	Output data (PrintDataAsync method)	Sends to the printer without changing the data.
24	Print OPOS format (PrintNormalAsync method)	Prints text using OPOS escape sequences.
25	Get version code (GetVersionCode method)	Get a numerical value for the version number of this SDK.
26	Get version name (GetVersionName method)	Get a string for the version number of this SDK.
27	Watermark print (WatermarkPrintAsync method)	Enters or exits watermark print mode.

28	Set print completed timeout (SetPrintCompletedTimeout method)	Set the timeout to check the print completion notification.
29	Log settings (SetLog method)	Set the log function.

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.
7	Mapping mode (MapMode property)	R/W	Show the mapping mode (the unit of measure) of the printer.

2.3. Library interfaces

The following are the interfaces of this SDK.

2.3.1. 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.

2.3.2. Constructor

Syntax

ESCPOSPrinter ()

Parameter

Not exist.

Description

It is the constructor for this library. It creates an instance.

Return value

Not exist.

Example

```
ESCPOSPrinter printer = new ESCPOSPrinter();
```

2.3.3. ConnectAsync method

Syntax

1) Task<int> ConnectAsync (int connectType, string addr)
2) Task<int> ConnectAsync (int connectType, string addr, int port)
3) Task<int> ConnectAsync (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
connectType	[IN]	Connect type	CMP_PORT_WiFi CMP_PORT_Bluetooth
addr	[IN]	IP address or BD address to connect.	WiFi: 0.0.0.0～255.255.255.255 Bluetooth: 00:00:00:00:00:00~ FF:FF:FF:FF:FF:FF
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 giving the maximum number of milliseconds to connect printer. If it is omitted, you connected with 8000 milliseconds when using Bluetooth and connected with 4000 milliseconds in other cases.

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 DisconnectAsync method to disconnect the printer connection. When not disconnect, the next connection will be an error.

An application with this SDK connects to a Bluetooth printer for the first time, Windows shows the consent dialog to accept to use the Bluetooth device (printer). The "timeout" parameter is invalid while this dialog is shown. To show this dialog, write ConnectAsync method in the UI thread, or it will not show and fails to connect.

There are declarations to use "Bluetooth" or "Internet (Client)", "Private Networks (Client & Server)" in the "Capabilities" tab of the Package.appxmanifest which is created by the Visual Studio in the new project files. Set them correctly, or Windows prevent to use these interfaces.

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 "2.3.1. 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

```
await printer.ConnectAsync ( ESCPOSConst.CMP_PORT_WiFi, "192.168.182.100" );

await printer.ConnectAsync ( ESCPOSConst.CMP_PORT_Bluetooth,
                                                   "00:01:90:01:23:45" );
```

2.3.4. DisconnectAsync method

Syntax

Task<int> DisconnectAsync ()

Parameter

Not exist.

Description

This method is used to disconnect the printer connection.

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

Return value

Return CMP_SUCCESS(0) in success. Please refer to "2.3.1. Return value" for the error code except it.

Example

```
await printer.DisconnectAsync();
```

2.3.5. SetEncoding method

Syntax

int SetEncoding (string charset)

Parameter

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

Value	[IN/OUT]	Meaning	Setting range
charset	[IN]	Character set name	Encoding that is supported depends on the implementation of Windows.

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 "1.3 Supported models")
This SDK supports printing UTF-8 encoded characters. Please refer to "2.4.3 About printing UTF-8 encode characters" for the detail.
When used in Japanese, it is necessary to specify the "Shift-JIS".

Return value

Return CMP_SUCCESS (0) in success. Please refer to "2.3.1. Return value" for the error code except it.

Example

```
printer.SetEncoding( "Shift_JIS" );

printer.SetEncoding( "GB18030" );

printer.SetEncoding( "EUC-KR" );

printer.SetEncoding( "BIG5" );

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

2.3.6. PrinterCheckAsync method

Syntax

Task<int> PrinterCheckAsync ()

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 DisconnectAsync method and the ConnectAsync 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 "2.3.1. Return value" for the error code except it.

Example

```
if ( ESCPOSConst.CMP_SUCCESS ==await printer.PrinterCheckAsync() ) {
    // Success
} else {
    // Fail
}
```

2.3.7. 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_DRAWER_LEVEL_H CMP_STS_PAPER_NEAREMPTY CMP_STS_COVER_OPEN CMP_STS_PAPER_EMPTY CMP_STS_PRINTEROFF

Description

This method is used to get the status of the printer obtained by the PrinterCheckAsync method. Before the execution of this method, you must run the PrinterCheckAsync 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_DRAWER_LEVEL_H (2)	Status of pin 3 of drawer kick-out connector = H (when the type parameter is set)
CMP_STS_PAPER_NEAREMPTY (4)	Paper near empty. (when the type parameter is set)
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

```
int status = printer.Status();
if ( ESCPOSConst.CMP_STS_NORMAL == status ) {
    // No Error
    int status2 = printer.Status(ESCPOSConst.CMP_STS_PAPER_NEAREMPTY);
    if ( (ESCPOSConst.CMP_STS_PAPER_NEAREMPTY & status2) > 0 ) {
        // Paper Near Empty
    }
} else {
    if ( (ESCPOSConst.CMP_STS_COVER_OPEN & status) > 0 ) {
        // Cover Open
    }
    if ( (ESCPOSConst.CMP_STS_PAPER_EMPTY & status) > 0 ) {
        // Paper Empty
    }
    if ( (ESCPOSConst.CMP_STS_PRINTEROFF & status) > 0 ) {
        // Printer Offline
    }
}

int status3 = printer.Status(ESCPOSConst.CMP_STS_DRAWER_LEVEL_H);
if ( (ESCPOSConst.CMP_STS_DRAWER_LEVEL_H & status3) > 0 ) {
    // Status of pin 3 of drawer kick-out connector = H
}
```

2.3.8. PrintTextAsync method

Syntax

Task<int> PrintTextAsync (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.
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 "2.3.1. Return value" for the error code except it.

Example

```
await printer.PrintTextAsync( "Print text data.\n",
          ESCPOSConst.CMP_ALIGNMENT_CENTER,
          ESCPOSConst.CMP_FNT_BOLD | ESCPOSConst.CMP_FNT_UNDERLINE,
          ESCPOSConst.CMP_TXT_2WIDTH | ESCPOSConst.CMP_TXT_2HEIGHT );
```

2.3.9. PrintBitmapAsync method

Syntax

1) Task<int> PrintBitmapAsync (string fileName, int alignment)
2) Task<int> PrintBitmapAsync (string fileName, int width, int alignment)
3) Task<int> PrintBitmapAsync (string fileName, int width, int alignment, int mode)
4) Task<int> PrintBitmapAsync (BitmapDecoder bitmap, int alignment)
5) Task<int> PrintBitmapAsync (BitmapDecoder bitmap, int width, int alignment)
6) Task<int> PrintBitmapAsync (BitmapDecoder bitmap, int width, int alignment, int mode)
7) Task<int> PrintBitmapAsync (byte[] bytes, int alignment)
8) Task<int> PrintBitmapAsync (byte[] bytes, int width, int alignment)
9) Task<int> PrintBitmapAsync (byte[] bytes, int width, int alignment, int mode)

Parameter

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

Value	[IN/OUT]	Meaning	Setting range
fileName	[IN]	Bitmap file name	
bitmap	[IN]	BitmapDecoder type data	
bytes	[IN]	Byte array representation of the bitmap	
width	[IN]	Bitmap width	CMP_BM_ASIS: Print the bitmap with one bitmap pixel per printer dot. Other Values: Bitmap width expressed. Expressed in the unit of measure given by MapMode (default 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. Expressed in the unit of measure given by MapMode (default dots).
mode	[IN]	Bitmap mode	CMP_BM_MODE_HT_THRESHOLD: Halftone threshold CMP_BM_MODE_HT_DITHER: Halftone dither CMP_BM_MODE_CMD_RASTER: Raster command output CMP_BM_MODE_CMD_BITIMAGE: Bitimage command output CMP_BM_MODE_CMD_GRAY16: Grayscale (4bpp) output

Description

This method is used to print bitmap which specifies file name or bitmap and width and alignment and mode. File must be located in the local folder (Windows.Storage.ApplicationData.Current.LocalFolder).
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	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	Bit image 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-D101/151,CT-E601/651,CT-S251/601II/651II/801II/851II/801II/851II/751. Graphic can be printed more beautifully.

Return value

Return CMP_SUCCESS (0) in success. Please refer to "2.3.1. Return value" for the error code except it.

Example

```
await printer.PrintBitmapAsync ( "samplebitmap.bmp",
    ESCPOSConst.CMP_BM_ASIS,
    ESCPOSConst.CMP_ALIGNMENT_CENTER
    ESCPOSConst.CMP_BM_MODE_HT_DITHER|ESCPOSConst.CMP_BM_MODE_CMD_RASTER );
```

2.3.10. PrintNVBitmapAsync method

Syntax

Task<int> PrintNVBitmapAsync (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 by 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]

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-S251/310II/601/651/801/851/601II/651II/801II/851II/801III/851III/751/2000/4000/4500 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 "2.3.1. Return value" for the error code except it.

Example

```
await printer.PrintNVBitmapAsync( 1 );
```

2.3.11. PrintBarCodeAsync method

Syntax

Task<int> PrintBarCodeAsync (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 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	1 - 255 (dots) Expressed in the unit of measure given by MapMode (default dots).
width	[IN]	Barcode horizontal size (magnification)	2 - 6 (dots) Expressed in the unit of measure given by MapMode (default 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. Expressed in the unit of measure given by MapMode (default 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.

GS1 DataBar (CMP_BCS_GS1DATABAR, CMP_BCS_GS1DATABAR_E, CMP_BCS_GS1DATABAR_T, CMP_BCS_GS1DATABAR_L) can use only the printers of CT-D101/150/151,CT-E301/351/601/651, CT-S251/310II/601/651/801/851/601II/651II/801II/851II/801II/851II/751/4500 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 "2.3.1. Return value" for the error code except it.

Example

```
await printer.PrintBarCodeAsync ( "123456789012",
          ESCPOSConst.CMP_BCS_UPCA,
          50,
          2,
          ESCPOSConst.CMP_ALIGNMENT_LEFT,
          ESCPOSConst.CMP_HRI_TEXT_ABOVE );
```

2.3.12. PrintPDF417Async method

Syntax

Task<int> PrintPDF417Async (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) Expressed in the unit of measure given by MapMode (default 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. Expressed in the unit of measure given by MapMode (default 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 "2.3.1. Return value" for the error code except it.

Example

```
await printer.PrintPDF417Async(
          "http://www.citizen-systems.co.jp/printer/index.html",
          0, 0, 3, 3,
          ESCPOSConst.CMP_PDF417_EC_LEVEL_0,
          ESCPOSConst.CMP_ALIGNMENT_LEFT );
```

2.3.13. PrintQRCodeAsync method

Syntax

int PrintQRCodeAsync (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) Expressed in the unit of measure given by MapMode (default 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. Expressed in the unit of measure given by MapMode (default 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 "2.3.1. Return value" for the error code except it.

Example

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

2.3.14. PrintGS1DataBarStackedAsync method

Syntax

Task<int> PrintGS1DataBarStackedAsync (string data, int symbology, int moduleSize, int maxSize, 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	
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	2 - 8 (dots) Expressed in the unit of measure given by MapMode (default dots).
maxSize	[IN]	Max width	106 - 39528 (dots) Max width of GS1 DataBar Expanded Stacked. Expressed in the unit of measure given by MapMode (default 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. Expressed in the unit of measure given by MapMode (default dots).

Description

This method is used to print 2-dimensional GS1 DataBar barcode.
This method can use only the printers of CT-D101/150/151, CT-E301/351/601/651, CT-S251/310II/601/651/801/851/601II/651II/801II/851II/801II/851II/4500 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 "2.3.1. Return value" for the error code except it.

Example

```
await printer.PrintGS1DataBarStackedAsync(
        "0123456789012",
        ESCPOSConst.CMP_BCS_GS1DATABAR_S,
        4,
        300,
        ESCPOSConst.CMP_ALIGNMENT_LEFT );
```

2.3.15. CutPaperAsync method

Syntax

Task<int> CutPaperAsync (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 "2.3.1. Return value" for the error code except it.

Example

```
await printer.CutPaperAsync( ESCPOSConst.CMP_CUT_PARTIAL_PREFEED );
```

2.3.16. UnitFeedAsync method

Syntax

Task<int> UnitFeedAsync (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	Expressed in the unit of measure given by MapMode (default dots).

Description

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

Return value

Return CMP_SUCCESS (0) in success. Please refer to "2.3.1. Return value" for the error code except it.

Example

```
await printer.UnitFeed( 200 );
```

2.3.17. MarkFeedAsync method

Syntax

Task<int> MarkFeedAsync (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 "2.3.1. Return value" for the error code except it.

Example

```
await printer.MarkFeedAsync( ESCPOSConst.CMP_MF_TO_CUTTER );
```

2.3.18. OpenDrawerAsync method

Syntax

Task<int> OpenDrawerAsync (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 "2.3.1. Return value" for the error code except it.

Example

```
await printer.OpenDrawerAsync( ESCPOSConst.CMP_DRAWER_1, 1 );
```

2.3.19. TransactionPrintAsync method

Syntax

Task<int> TransactionPrintAsync (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.

PrintTextAsync, PrintBitmapAsync, PrintNVBitmapAsync, PrintBarCodeAsync, PrintPDF417Async, PrintQRCodeAsync, PrintGS1DataBarStackedAsync, CutPaperAsync, UnitFeedAsync, OpenDrawerAsync, RotatePrintAsync, PageModePrintAsync, ClearePrintArea, PrintDataAsync, PrintNormalAsync

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 ClearOutputAsync method cancels transaction mode. Any buffered print lines are also cleared.

Return value

Return CMP_SUCCESS (0) in success. Please refer to "2.3.1. Return value" for the error code except it.

Example

```
await printer.TransactionPrintAsync( ESCPOSConst.CMP_TP_TRANSACTION );
await printer.PrintNVBitmapAsync( 1 );
await printer.PrintBarCodeAsync ( "123456789012", ESCPOSConst.CMP_BCS_UPCA,
          50, 2, ESCPOSConst.CMP_ALIGNMENT_LEFT,
          ESCPOSConst.CMP_HRI_TEXT_ABOVE );
await printer.PrintTextAsync( "Line 1\n", ESCPOSConst.CMP_ALIGNMENT_LEFT,
          ESCPOSConst.CMP_FNT_DEFAULT, ESCPOSConst.CMP_TXT_1WIDTH );
await printer.PrintTextAsync( "Line 2\n", ESCPOSConst.CMP_ALIGNMENT_LEFT,
          ESCPOSConst.CMP_FNT_DEFAULT, ESCPOSConst.CMP_TXT_1WIDTH );
await printer.PrintTextAsync( "Line 3\n", ESCPOSConst.CMP_ALIGNMENT_LEFT,
          ESCPOSConst.CMP_FNT_DEFAULT, ESCPOSConst.CMP_TXT_1WIDTH );
await printer.PrintBarCodeAsync ( "123456789012", ESCPOSConst.CMP_BCS_UPCA,
          50, 2, ESCPOSConst.CMP_ALIGNMENT_LEFT,
          ESCPOSConst.CMP_HRI_TEXT_ABOVE );
await printer.PrintNVBitmapAsync( 1 );
await printer.CutPaperAsync( ESCPOSConst.CMP_CUT_PARTIAL_PREFEED );
await printer.TransactionPrintAsync( ESCPOSConst.CMP_TP_NORMAL );
```

2.3.20. RotatePrintAsync method

Syntax

Task<int> RotatePrintAsync (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.

PrintTextAsync, PrintNormalAsync

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

PrintBarcodAsync, PrintPDF417Async, PrintQRCodeAsync, PrintGS1DataBarStackedAsync and/or PrintBitmapAsync

If rotation is CMP_RP_NORMAL, then rotation mode is exited.

Return value

Return CMP_SUCCESS (0) in success. Please refer to "2.3.1. Return value" for the error code except it.

Example

```
await printer.RotatePrintAsync( ESCPOSConst.CMP_RP_ROTATE180 |
            ESCPOSConst.CMP_RP_BARCODE | ESCPOSConst.CMP_RP_BITMAP );
await printer.PrintBitmapAsync ( "samplebitmap.bmp", ESCPOSConst.CMP_BM_ASIS,
            ESCPOSConst.CMP_ALIGNMENT_CENTER );
await printer.PrintBarCodeAsync ( "123456789012", ESCPOSConst.CMP_BCS_UPCA,
            50, 2, ESCPOSConst.CMP_ALIGNMENT_LEFT,
            ESCPOSConst.CMP_HRI_TEXT_ABOVE );
await printer.PrintTextAsync( "Line 3\n", ESCPOSConst.CMP_ALIGNMENT_LEFT,
            ESCPOSConst.CMP_FNT_DEFAULT, ESCPOSConst.CMP_TXT_1WIDTH );
await printer.PrintTextAsync( "Line 2\n", ESCPOSConst.CMP_ALIGNMENT_LEFT,
            ESCPOSConst.CMP_FNT_DEFAULT, ESCPOSConst.CMP_TXT_1WIDTH );
await printer.PrintTextAsync( "Line 1\n", ESCPOSConst.CMP_ALIGNMENT_LEFT,
            ESCPOSConst.CMP_FNT_DEFAULT, ESCPOSConst.CMP_TXT_1WIDTH );
await printer.RotatePrintAsync( ESCPOSConst.CMP_RP_NORMAL );
```

2.3.21. PageModePrintAsync method

Syntax

Task<int> PageModePrintAsync (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.

PrintTextAsync, PrintBitmapAsync, PrintBarCodeAsync, PrintPDF417Async, PrintQRCodeAsync, PrintGS1DataBarStackedAsync, PrintNormalAsync

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 PageModePrintAsync 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 ClearOutputAsync method cancels Page Mode. Any buffered print lines are also cleared.

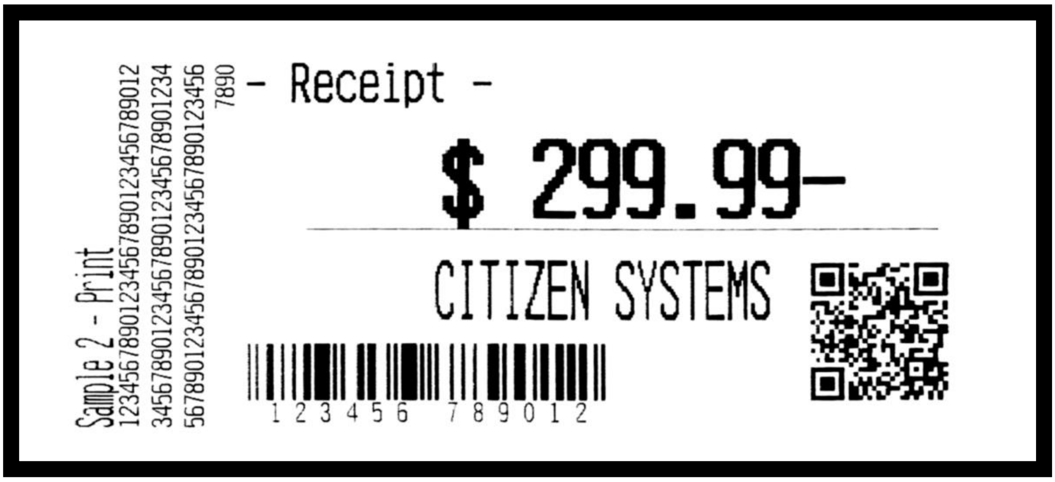
Return value

Return CMP_SUCCESS (0) in success. Please refer to "2.3.1. Return value" for the error code except it.

Example

```
// Standard print
await printer.PrintNormalAsync( "\u001b|2vCSample 2 - Print\n");
await printer.PrintTextAsync(
          "123456789012345678901234567890123456789012345678901234567890123
          4567890123456789012345678901234567890\n",
          ESCPOSConst.CMP_ALIGNMENT_RIGHT, ESCPOSConst.CMP_FNT_DEFAULT,
          ESCPOSConst.CMP_TXT_1WIDTH | ESCPOSConst.CMP_TXT_1HEIGHT);
// Start of Page Mode
await printer. PageModePrintAsync( ESCPOSConst.CMP_PM_PAGE_MODE );
// Set offset of Page Mode
printer.SetPageModeVerticalPosition( 0 );
printer.SetPageModeHorizontalPosition( 0 );
// Set direction of Page Mode
printer.SetPageModePrintDirection( ESCPOSConst.CMP_PD_TOP_TO_BOTTOM );
// Set print area of Page Mode
printer.SetPageModePrintArea( "308,0,76,800" );
await printer.PrintNormalAsync( "\u001b|4C- Receipt -\n" );
// Set print area of Page Mode
printer.SetPageModePrintArea( "184,0,120,800" );
await printer.PrintTextAsync( "   $ 299.99-  \n",
          ESCPOSConst.CMP_ALIGNMENT_CENTER, ESCPOSConst.CMP_FNT_UNDERLINE |
          ESCPOSConst.CMP_FNT_BOLD, ESCPOSConst.CMP_TXT_4WIDTH |
          ESCPOSConst.CMP_TXT_4HEIGHT );
// Set print area of Page Mode
printer.SetPageModePrintArea( "88,0,88,560" );
await printer.PrintTextAsync( "CITIZEN SYSTEMS\n",
          ESCPOSConst.CMP_ALIGNMENT_RIGHT, ESCPOSConst.CMP_FNT_DEFAULT,
          ESCPOSConst.CMP_TXT_2WIDTH | ESCPOSConst.CMP_TXT_3HEIGHT );
// Set print area of Page Mode
printer.SetPageModePrintArea( "0,0,88,480" );
await printer.PrintBarCodeAsync ( "123456789012",
          ESCPOSConst.CMP_BCS_UPCA, 64, 4, ESCPOSConst.CMP_ALIGNMENT_LEFT,
          ESCPOSConst.CMP_HRI_TEXT_BELOW );
// Set print area of Page Mode
printer.SetPageModePrintArea( "0,600,192,192" );
await printer.PrintQRCodeAsync(
          "http://www.citizen-systems.co.jp/", 5,
          ESCPOSConst.CMP_QRCODE_EC_LEVEL_L,
          ESCPOSConst.CMP_ALIGNMENT_LEFT );
// End of Page Mode
await printer.PageModePrintAsync( ESCPOSConst.CMP_PM_NORMAL );
```

Print image



Sample 2 - Print
12345678901234567890123456789012
34567890123456789012345678901234
56789012345678901234567890123456
7890

- Receipt -

$ 299.99-

CITIZEN SYSTEMS

1 2 3 4 5 6 7 8 9 0 1 2

2.3.22. 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 "2.3.1. Return value" for the error code except it.

Example

```
printer.ClearPrintArea();
```

2.3.23. ClearOutputAsync method

Syntax

Task<int> ClearOutputAsync ()

Parameter

Not exist.

Description

This method is used to clear all buffered output data by TransactionPrintAsync method and PageModePrintAsync 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 "2.3.1. Return value" for the error code except it.

Example

```
await printer.ClearOutputAsync();
```

2.3.24. PrintDataAsync method

Syntax

Task<int> PrintDataAsync (byte[] data)

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 "2.3.1. Return value" for the error code except it.

Example

```
// Sound the buzzer (The printer must support buzzer.)
await printer.PrintDataAsync(new byte[]{0x1b, 0x1e});
```

2.3.25. PrintNormalAsync method

Syntax

Task<int> PrintNormalAsync (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.

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 "2.3.1. Return value" for the error code except it.

Example

```
await printer.PrintNormalAsync( "\u001b|4C- Receipt -\n" );
```

2.3.26. 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. (200 means Ver2.00)

Example

```
printer.GetVersionCode();
```

2.3.27. GetVersionName method

Syntax

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. ("2.00" means Ver2.00)

Example

```
printer.GetVersionName();
```

2.3.28. WatermarkPrintAsync method

Syntax

Task<int> WatermarkPrintAsync (int start, int nvImageNumber, int pass, int feed, int repeat)

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 (dots) Expressed in the unit of measure given by MapMode (default dots).
feed	[IN]	The blank length each watermark	0 - 65,535 (dots) Expressed in the unit of measure given by MapMode (default dots).
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-D151,CT-E601/651,CT-S251/601II/651II/801II/851II /801III/851III/751 series.

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

To use this method, you need to register of the logo in advance. Logo registration, please store it by using 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 "2.3.1. Return value" for the error code except it.

Example

```
await printer.WatermarkPrintAsync( ESCPOSConst.CMP_WM_START, 1, 0, 0, 0 );
```

2.3.29. 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 until the end of printing	0: Timeout is calcurated from print data, automatically. Other value: Timeout is specified to the value in millisecond.

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 "2.4.1. 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 "2.3.1. Return value" for the error code except it.

Example

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

printer.SetPrintCompletedTimeout( 90000 );     // Fixes to 90 seconds
```

2.3.30. SetLog method

Syntax

void SetLog (int mode, string path, int maxSize)

Parameter

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

Value	[IN/OUT]	Meaning	Setting range
mode	[IN]	Log mode	0: No Record 1: Recording of access history 2: Error only record
path	[IN]	Store folder	Relative path from LocalFolder
maxSize	[IN]	Log size	0: No size limit 1 - : Maximum size (MB)

Description

This method is used to set the log function. Please refer to "3.2 Log function" for details of the log function.

Return value

Not exist

Example

```
printer.SetLog( 1, ”Log”, 10 );
```

2.3.31. PageModeArea property

Syntax

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 Series] : "384,938"
[CT-D101/150/151, CT-E301/351/601/651, CT-S310II/601/651/801/851/601II/651II/801III/851III/751/2000 Series] : "576,1662"
[CT-S4000/4500 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 ConnectAsync method must be complete before accessing this property. This property is set in ConnectAsync method.

Set property

Not exist.

Get property

String GetPageModeArea()

Returns the page area as the return value.

2.3.32. PageModePrintArea property

Syntax

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

Set property

int SetPageModePrintArea (String area)

Please specify the property value that you want to set in the parameter.
Return CMP_SUCCESS (0) in success. Please refer to "2.3.1. Return value" for the error code except it.

Get property

String GetPageModePrintArea ()

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

2.3.33. 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 ConnectAsync method must be complete before accessing this property. This property is initialized to CMP_PD_LEFT_TO_RIGHT at ConnectAsync 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 "2.3.1. 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.

2.3.34. 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 ConnectAsync method must be complete before accessing this property. This property is initialized to zero (0) at ConnectAsync 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 "2.3.1. 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.

2.3.35. 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 ConnectAsync method must be complete before accessing this property. This property is initialized to zero (0) at ConnectAsync 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 "2.3.1. 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.

2.3.36. RecLineSpacing property

Syntax

int RecLineSpacing

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 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 ConnectAsync method must be complete before accessing this property. This property is initialized to 34 at ConnectAsync 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 "2.3.1. 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.

2.3.37. MapMode property

Syntax

int MapMode

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.

PrintBitmapAsync method : width, alignment
SetNVBitmapAsync method : width
PrintBarcodeAsync method : height, width, alignment
PrintPDF417Async method : moduleWidth, alignment
PrintQRCode method : moduleSize, alignment)
PrintGS1DataBarStackedAsync method : moduleSize, maxSize, alignment
UnitFeedAsync method : ufCount
WatermarkPrintAsync method : pass, feed
PageModeArea property
PageModePrintArea property
PageModeHorizontalPosition property
PageModeVerticalPosition property
RecLineSpacing property

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

Set property

int SetMapMode (int mode)

Please specify the property value that you want to set in the parameter.
Return CMP_SUCCESS (0) in success. Please refer to "2.3.1. Return value" for the error code except it.

Get property

int GetMapMode ()

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

Example

```
printer.SetMapMode( ESCPOSConst.CMP_MM_DOTS );
await printer.UnitFeedAsync( 200 );            // 200 dots feed
printer.SetMapMode( ESCPOSConst.CMP_MM_METRIC );
await printer.UnitFeedAsync( 2500 );           // 25 millimeter feed
```

2.4. Notes

Notes of this SDK are as follows.

2.4.1. Function to detect the completion of printing

In this SDK, 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 (TransactionPrintAsync method)
(2) At the time of completion of page mode (PageModePrintAsync 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 from the printer. If you want to print multiple methods continuously, transaction processing (TransactionPrintAsync method) can makes printing smooth.

Timeout to the end of printing is decided by its contents automatically.
Some print data makes timeout error, every time. In such case, use SetPrintCompletionTimeout method to modify timeout by its printing time.

2.4.2. Log function

This SDK supports the log function which records the methods and properties. When setting the log function, please place configuration file "CSJPOSLib.cfg" of the next format in the LocalFolder.

<Example of CSJPOSLib.cfg>

```
[LogSetting]          ... Section name (Fixed)
LogMode=1             ... Specifies the log mode.
LogPath=Log           ... Specifies the relative path from the LocalFolder to store the log files.
LogMaxSize=10         ... Specifies the maximum size of log file in MB.
```

Setting items

- Log mode
  Specifies the mode for recording the log.
    0: No Record
    1: Recording of access history
    2: Error only record

- Store folder
  Specifies a folder which log files will be stored. It must be described as relative path from the local folder. When this setting is not specified, log files will be stored into the local folder.

- Log size
  Specifies maximum size of a log file in MB. If 0 is specified, log data will be written without limit.

Log file name

The extension of log files is ".log". In the file name, a numeric character which means the day of week is followed to ″CSJPOSLib″. The numeric character is from 0 to 6. "0" means Sunday, "1" means Monday.
Example: CSJPOSLib_1.log

If a log file is already existing and it is older than today, it will be deleted, and the log data will be recorded into a new file. (It will be held for one week.)

Log format

A log file keeps information of the executed methods, accessed properties, timestamps and results.

--- Example 1 of method (Connect) ---

```
2019/12/24 13:31:44.138 9636 011 METHOD call   ConnectAsync(0, "192.168.10.100")
2019/12/24 13:31:45.684 9636 011 METHOD result ConnectAsync() -> Success(0)
```

--- Example 2 of method (PrintText) ---

```
2019/12/24 13:31:50.141 9636 011 METHOD call   PrintTextAsync([See below], 1, 1, 0)
-----------------Parameter Detail-----------------------
Print text 1
Print text 2
--------------------------------------------------------
2019/12/24 13:31:50.634 9636 011 METHOD result PrintTextAsync() -> Success(0)
```

--- Example to set to properties ---

```
2019/12/24 13:35:23.021 4488 008 PROPERTY set  RecLineSpacing <- 24 : Success(0)
```

--- Example to get from properties ---

```
2019/12/24 13:39:29.037 4488 008 PROPERTY get  RecLineSpacing -> 24
```

* When this SDK works with logging, it performs uncomfortably because a log file will be updated at every method and accessing properties.

* Because of the following reasons or else, log data will not be stored without any notification.
- A folder where is not under the local folder is specified.
- A folder or file without permission is specified.
- A write-protected log file is already existing.
- Another program (such as a text editor) is using (locking) the log file.
- There is not enough space to store log data in the device.

2.4.3. 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 CT-E301/351/601/651 CT-S801III/851III/S751/4500	All versions	*3

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)

2.4.4. List of Constants Predefined

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_E_NO_LIST	int	1106	Printer cannot be found
		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	Interface to connect	CMP_PORT_WiFi	int	0	Network
		CMP_PORT_Bluetooth	int	1	Bluetooth
4	Status	CMP_STS_NORMAL	int	0	Normal
		CMP_STS_DRAWER_LEVEL_H	int	2	Status of 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
5	Alignment	CMP_ALIGNMENT_LEFT	int	0	Left alignment
		CMP_ALIGNMENT_CENTER	Int	1	Center alignment
		CMP_ALIGNMENT_RIGHT	int	2	Right alignment
6	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	16	Reverse
		CMP_FNT_UNDERLINE	int	128	Underline
7	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
8	Side	CMP_SIDE_RIGHT	int	0	Right side
		CMP_SIDE_LEFT	int	1	Left side
9	Bitmap width	CMP_BM_ASIS	int	-11	One bitmap pixel per printer dot
10	Bitmap mode	CMP_BM_MODE_CMD_RASTER	int	1	Monochrome print (Raster command)
		CMP_BM_MODE_CMD_BITIMAGE	int	2	Monochrome print (Bit image command)
		CMP_BM_MODE_CMD_MONO	int	8	Monochrome register
		CMP_BM_MODE_CMD_GRAY16	int	8	Grayscale print/ register
		CMP_BM_MODE_HT_THRESHOLD	int	16	Halftone (Threshold)
		CMP_BM_MODE_HT_DITHER	int	32	Halftone (Dither)
11	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
		CMP_BCS_GS1DATABAR_T	int	135	GS1 DataBar Truncated
		CMP_BCS_GS1DATABAR_L	int	136	GS1 DataBar Limited
		CMP_BCS_GS1DATABAR_S_O	int	137	GS1 DataBar Stacked Omnidirectional
12	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
13	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
14	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%)
15	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
17	Drawer number	CMP_DRAWER_1	int	1	Drawer 1
		CMP_DRAWER_2	int	2	Drawer 2

18	Transaction control	CMP_TP_TRANSACTION	int	11	Begin transaction
		CMP_TP_NORMAL	int	12	End transaction
19	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
20	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
21	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
22	Watermark control	CMP_WM_STOP	int	0	End watermark
		CMP_WM_START	int	1	Begin watermark
23	Map mode type	CMP_MM_DOTS	int	1	The printer’s dot width
		CMP_MM_TWIPS	int	2	1/1440 of an inch
		CMP_MM_ENGLISH	int	3	0.001 inch
		CMP_MM_METRIC	int	4	0.01 millimeter

3. Linedisplay Control

3.1. Program structure

Here is an example program in C# which uses the SDK

```
// Create an instance.
LineDisplay display = new LineDisplay();

// Connect Linedisplay
int result = await display.ConnectAsync(LineDisplayConst.CDP_PORT_WiFi,
                                        "192.168.10.1");
if (LineDisplayConst.CDP_SUCCESS == result)
{
    // Set encoding
    display.SetEncoding("Shift_JIS");

    // Clear text
    await display.ClearDisplayAsync();

    // Display text
    await display.DisplayTextAsync("123456");

    // Set cursor position
    await display.SetCursorPosotionAsync(1,2);

    // Display text (Reverse)
    await display.DisplayTextAsync("123456",true);

    // Disconnect
    await display.DisconnectAsync();

}
else
{
    // Connect Error
    MessageDialog msgbox = new MessageDialog(
           "Connect or LineDisplay Error : " + result.ToString(),
           "Citizen_POS_sample1");
    await msgbox.ShowAsync();
}
```

Class definition

Connect

Linedisplay processes

Disconnect

3.2. Functions list

This SDK provides the following functions.

Methods list

No	Function	Detail
1	Connect display (ConnectAsync method)	This method connects to the line display
2	Disconnect display (DisconnectAsync method)	This method disconnects the line display connection.
3	Display the text (DisplayTextAsync method)	This method is used to display text.
4	Clear the displayed text (ClearDisplayAsync method)	This method clears the displayed text.
5	Blink the display (BlinkDisplayAsync method)	This method causes the entire display screen to blink.
6	Set display mode (SetDisplayModeAsync method)	This method sets the following display modes.
7	Set display config (SetDisplayConfigAsync method)	This method changes the brightness of the display screen.
8	Set cursor Position (SetCursorPositionAsync method)	This method is used to set the cursor position.
9	Move cursor (MoveCursorAsync method)	This method is used to move the cursor.
10	Show cursor position (SetCursorTypeAsync method)	This displays the current cursor position on the display.
11	Initialize (InitializeDisplayAsync method)	This method initializes the device.
12	Send command (DisplayDataAsync method)	This method sends the command.
13	Set encoding (SetEncoding method)	This method sets the encoding of character.
14	Set code page (SetCodePageAsync method)	This method sets the code page of character.
15	Set international characterset (SetInternationalCharactersetAsync method)	This sets the following international character sets.
16	Check display status (CheckDisplayAsync method)	This method is used to check the display connection status.
17	Get version code (GetVersionCode method)	This method gets a numerical value for the version number of this SDK.
18	Get version name (GetVersionName method)	This method gets a string for the version number of this SDK.
19	Log settings (SetLog method)	Set the log function.

3.3. Library interfaces

The following are the interfaces of this SDK.

3.3.1. Return value

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

Return value	Description
CDP_SUCCESS (0)	The operation is success.
CDP_E_CONNECTED (1001)	The device is already connected.
CDP_E_DISCONNECT (1002)	The device is not connected.
CDP_E_NOTCONNECT (1003)	Failed connection to the device.
CDP_E_CONNECT_NOTFOUND (1004)	Failed to check the support model after connecting to the device.
CDP_E_CONNECT_OFFLINE (1005)	Failed to check the printer status after connecting to the device.
CDP_E_ILLEGAL (1101)	Unsupported operation with the Device, or an invalid parameter value was used.
CDP_E_OFFLINE (1102)	The printer is off-line.
CDP_E_FAILURE (1104)	The Service cannot perform the requested procedure.

3.3.2. Constructor

Syntax

LineDisplay ()

Parameter

Not exist.

Description

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

Return value

Not exist.

Example

```
LineDisplay display = new LineDisplay();
```

3.3.3. ConnectAsync method

Syntax

1) Task<int> ConnectAsync (int connectType, String addr)
2) Task<int> ConnectAsync (int connectType, String addr, int port)
3) Task<int> ConnectAsync (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
connectType	[IN]	Connection type of the printer	CDP_PORT_WiFi CDP_PORT_Bluetooth
addr	[IN]	IP address to connect	WiFi: 0.0.0.0～255.255.255.255 Bluetooth: 00:00:00:00:00:00~ FF:FF:FF:FF:FF:FF
port	[IN]	Connection port number	
timeout	[IN]	Timeout (msec)	

Description

This method is used to connect the line display. Please specify the type and address of the printer to which the line display is connected.

Connection port number is valid only if you specify the connection type CDP_PORT_WiFi. If it is omitted, it connects with number 9200.

Timeout is giving the maximum number of milliseconds to connect display. If it is omitted, it connects with 4000 milliseconds in the case of Wi-Fi.

When communication with the line display is not necessary, must execute the DisconnectAsync method to disconnect the line display connection. When not disconnect, the next connection will be an error.

Return value

Return CDP_SUCCESS (0) in success. Please check the description of the error codes below in the case of failure. Please refer to "3.3.1 Return value" for the error code except it.

Error codes	Description
CDP_E_NOTCONNECT (1003)	Failed connection to the line display. (1) The line display is under none-connection status. (2) The printer is not turned ON. (3) Cannot obtain handle of interface board.

Example

```
await display.ConnectAsync(LineDisplayConst.CDP_PORT_ WiFi, "192.168.0.10");
```

3.3.4. DisconnectAsync method

Syntax

Task<int> DisconnectAsync ()

Parameter

Not exist.

Description

This method is used to disconnect the line display connection.

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

Return value

Return CDP_SUCCESS(0) in success. Please refer to "3.3.1 Return value"" for the error code except it.

Example

```
await display.DisconnectAsync();
```

3.3.5. DisplayTextAsync method

Syntax

Task<int> DisplayTextAsync (String data, boolean reverseFlag)

Parameter

The meanings and settable values of the parameters are as follows.

Value	Meaning	Settable range
Data	Text data	String
ReverseFlag	Reverse specification flag	false: Standard true: Reverse When the argument is omitted, it is treated as false.

Description

This method is used to display text from the current cursor position.
Reverse can be specified for the text attribute.

Return value

Return CDP_SUCCESS (0) in success. Please refer to "3.3.1 Return value"" for the error code except it.

Example

```
await display.DisplayTextAsync("Hello, World!");
```

3.3.6. ClearDisplayAsync method

Syntax

await Task<int> ClearDisplayAsync (int displayArea)

Parameter

The meanings and settable values of the parameters are as follows.

Value	Meaning	Settable range
displayArea	Clear area	CDP_AREA_ALL(0): Entire area CDP_AREA_CURSORLINE(1):Cursor line When the argument is omitted, it is treated as CDP_AREA_ALL.

Description

This method clears the displayed text.

Return value

Return CDP_SUCCESS (0) in success. Please refer to "3.3.1 Return value"" for the error code except it.

Example

```
await display.ClearDisplayAsync(LineDisplayConst.CDP_AREA_ALL);
```

3.3.7. BlinkDisplayAsync method

Syntax

Task<int> BlinkDisplayAsync (int intervalBlink)

Parameter

The meanings and settable values of the parameters are as follows.

Value	Meaning	Settable range
IntervalBlink	Blink interval (msec)	From 0

Description

This method causes the entire display screen to blink.

The blink interval (msec) specifies the interval for on and off. If 0 is specified for the blink interval, blinking is disabled.

Return value

Return CDP_SUCCESS (0) in success. Please refer to "3.3.1 Return value"" for the error code except it.

Example

```
await display.BlinkDisplayAsync(1000);
```

3.3.8. SetDisplayModeAsync method

Syntax

Task<int> SetDisplayModeAsync (int displayMode)

Parameter

The meanings and settable values of the parameters are as follows.

Value	Meaning	Settable range
DisplayMode	Display mode	CDP_OVERWRITE(1): Overwrite mode CDP_VERTICALSCROLL(2): Vertical scroll mode CDP_HORIZONTALSCROLL(3): Horizontal scroll mode

Description

This method sets the following display modes.

DisplayMode	Overview
Overwrite	Overwrites the text at the cursor position and moves the cursor to the right. (The cursor moves to the bottom left edge for input when it is at the top right edge, and the cursor moves to the top left edge for input when it is at the bottom right edge.)
VerticalScroll	Scrolls the display line of the top edge to the bottom edge by cursor up movement when the cursor is at the top edge (or by left movement when it is at the left edge). Scrolls the display line of the bottom edge to the top edge by cursor down movement when the cursor is at the bottom edge (or by right movement when it is at the right edge).
HorizontalScroll	Scrolls the text leftward in respect to the current cursor line by cursor right movement (or by text input) when the cursor is at the right edge. Scrolls the text rightward in respect to the current cursor line by cursor left movement when the cursor is at the left edge.

Return value

Return CDP_SUCCESS (0) in success. Please refer to "3.3.1 Return value"" for the error code except it.

Example

```
await display.SetDisplayModeAsync(LineDisplayConst.CDP_VERTICALSCROLL);
```

3.3.9. SetDisplayConfigAsync method

Syntax

Task<int> SetDisplayConfigAsync (int brightness)

Parameter

The meanings and settable values of the parameters are as follows.

Value	Meaning	Settable range
Brightness	Brightness (%)	0 to 100

Description

This method changes the brightness of the display screen.

The higher the numerical value, the brighter the brightness becomes. If 0 is specified, the screen turns off (the display content is retained).

After this is set, blinking of the entire display screen is disabled.

Return value

Return CDP_SUCCESS (0) in success. Please refer to "3.3.1 Return value"" for the error code except it.

Example

```
await display.SetDisplayConfigAsync(40);
```

3.3.10. SetCursorPositionAsync method

Syntax

Task<int> SetCursorPositionAsync (int x, int y)

Parameter

The meanings and settable values of the parameters are as follows.

Value	Meaning	Settable range
x	Digit position	From 1
y	Line position	From 1

Description

This method is used to set the cursor position.

The cursor position is the movement coordinates of the cursor, and specifies the digit position and line position.

Return value

Return CDP_SUCCESS (0) in success. Please refer to "3.3.1 Return value"" for the error code except it.

Example

```
await display.SetCursorPositionAsync(1, 2);
```

3.3.11. MoveCursorAsync method

Syntax

Task<int> MoveCursorAsync (int dx, int dy)

Parameter

The meanings and settable values of the parameters are as follows.

Value	Meaning	Settable range
dx	Rightward/leftward movement amount	-128 to 127
dy	Upward/downward movement amount	-128 to 127

Description

This method is used to move the cursor.

Movement is from the current cursor position. Specify the leftward/rightward movement amount (-: leftward, +: rightward) and upward/downward movement amount (-: upward, +: downward) for the cursor movement amount.

Return value

Return CDP_SUCCESS (0) in success. Please refer to "3.3.1 Return value"" for the error code except it.

Example

```
await display.MoveCursorAsync(2, 0);
```

3.3.12. SetCursorTypeAsync method

Syntax

Task<int> SetCursorTypeAsync (int cursorType)

Parameter

The meanings and settable values of the parameters are as follows.

Element	Meaning	Settable range
CursorType	Cursor type specification	CDP_TYPE_NONE: Hide cursor CDP_TYPE_UNDERLINE: Display cursor (Omittable element, TYPE_UNDERLINE when omit)

Description

This displays the current cursor position on the display.

Return value

Return CDP_SUCCESS (0) in success. Please refer to "3.3.1 Return value"" for the error code except it.

Example

```
await display.SetCursorTypeAsync(LineDisplayConst.CDP_TYPE_UNDERLINE);
```

3.3.13. InitializeDisplayAsync method

Syntax

Task<int> InitializeDisplayAsync ()

Parameter

None

Description

Initializes the device.

Return value

Return CDP_SUCCESS (0) in success. Please refer to "3.3.1 Return value"" for the error code except it.

Example

```
await display.initializeDisplayAsync();
```

3.3.14. DisplayDataAsync method

Syntax

Task<int> DisplayDataAsync (byte[] data)

Parameter

The meanings and settable values of the parameters are as follows.

Element	Meaning	Settable range
data	Send data	

Description

This method is used to transmit byte data as it is to the device.
Be careful not to affect other methods when using it.

Return value

Return CDP_SUCCESS (0) in success. Please refer to "3.3.1 Return value"" for the error code except it.

Example

```
// Execute self test
res = await display.DisplayData(new byte[]{0x1f, 0x40});
```

3.3.15. SetEncoding method

Syntax

int SetEncoding (String charset)

Parameter

The meanings and settable values of the parameters are as follows.

Element	Meaning	Settable range
data	Send data	

Description

This method is used to set the encoding of the send data to the display.
When you create an instance, it is initialized to the default character set of the OS.
When used in Japanese, it is necessary to specify the "Shift-JIS".

Return value

Return CMP_SUCCESS (0) in success. Please refer to "3.3.1 Return value"" for the error code except it.

Example

```
// Japanese
display.SetEncoding( "Shift_JIS" );

// Chinese
display.SetEncoding( "GB18030" );

// Korean
display.SetEncoding( "EUC-KR" );

// Taiwanese
display.SetEncoding( "Big5" );
```

3.3.16. SetCodePageAsync method

Syntax

Task<int> SetCodePageAsync (int codePage)

Parameter

The meanings and settable values of the parameters are as follows.

Element	Meaning	Settable range
codePage	Code page specification	0 - 255

Description

Please refer to the command reference "ESC t" command of the utilization device for the set point

Return value

Return CDP_SUCCESS (0) in success. Please refer to "3.3.1 Return value"" for the error code except it.

Example

```
await display.SetCodePageAsync(1);
```

3.3.17. SetInternationalCharactersetAsync method

Syntax

Task<int> SetInternationalCharactersetAsync (int characterset)

Parameter

The meanings and settable values of the parameters are as follows.

Element	Meaning	Settable range
codePage	International character specification	0 - 16

Description

Set the following international character set.

characterset	InternationalCharacterset	characterset	InternationalCharacterset
0	America	9	Norway
1	France	10	Denmark II
2	Germany	11	Spain II
3	England	12	Latin America
4	Denmark I	13	Korea
5	Sweden	14	Croatia
6	Italy	15	China
7	Spain I	16	Vietnam
8	Japan		

Return value

Return CDP_SUCCESS (0) in success. Please refer to "3.3.1 Return value"" for the error code except it.

Example

```
await display.SetInternationalCharactersetAsync(8);
await display.DisplayTextAsync("Total:\\1,010");
```

3.3.18. DisplayCheckAsync method

Syntax

Task<int> DisplayCheckAsync ()

Parameter

Not exist.

Description

This method is used to check the display connection status.
When the execution result of this method is successful, you can confirm that the display is connected.
When the execution result of this method fails, communication error or device error may have occurred.
In this case, reconnect using DisconnectAsync method and ConnectAsync method.

In the case of network connection, it will be disconnected automatically when left for a long time. To keep the connection, please execute this method periodically.

Return value

Return CDP_SUCCESS (0) in success. Please refer to "3.3.1 Return value"" for the error code except it.

Example

```
await display.DisplayCheckAsync();
```

3.3.19. 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

```
int vno = display.GetVersionCode();
```

3.3.20. GetVersionName method

Syntax

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

```
String vname = display.GetVersionName();
```

3.3.21. SetLog method

Syntax

void SetLog (int mode, String path, int maxSize)

Parameters

Parameter	[IN/OUT]	Description	Setting range
mode	[IN]	Logging mode	0: None 1: Access logs 2: Error logs
path	[IN]	File path to store	
maxSize	[IN]	Maximum Log Size	0: Unlimited 1 - : Maximum size (MB)

Description

Sets the logging function. See "3.4.1 Logging function" for more details.

Return value

none

Example

```
display.SetLog(1, "Log", 10);
```

3.4. Notes

Notes of this SDK are as follows.

3.4.1. Logging function

This SDK supports the log function which records the methods and properties. When setting the log function, use the SetLog method, or placing a file "CSJPOSLibD.cfg" of the next format in the LocalFolder.

< Example of CSJPOSLibD.cfg >
[LogSetting] ... Section name (Fixed)
LogMode=1 ... Specifies the log mode.
LogPath=Log ... Specifies the relative path from the LocalFolder to store the log files.
LogMaxSize=10 ... Specifies the maximum size of log file in MB.

Setting items

- LogMode
  Specifies a log mode:
  - 0: None
  - 1: Access log
  - 2: Error log

- LogPath
  Specifies a folder which log files will be stored. It must be described as relative path from the local folder. When this setting is not specified, log files will be stored into the local folder.

- LogMaxSize
  Specifies maximum size of a log file in MB. If 0 is specified, log data will be written without limit.

Log file name

Log files will be stored with a file name "CSJPOSLibD_" and a number which indicates the day of week(0 to 6. 0: Sunday, 1: Monday...), and a file extension ".log."

Example: CSJPOSLibD_1.log

If a log file is already existing and it is older than today, it will be deleted, and the log data will recorded into a new file. (It will be held for one week.)

Log format

A log file keeps the information of executed methods, accessed properties, timestamps and results.

```
--- Example 1, method (Connect) ---

2019/12/24 13:26:31.857 8028 008 METHOD call   ConnectAsync(0, "192.168.10.100")
2019/12/24 13:26:33.313 8028 008 METHOD result ConnectAsync() -> Success(0)

--- Example 2, method (DisplayText) ---

2019/12/24 13:26:33.635 8028 008 METHOD call   DisplayTextAsync([See below])
-----------------Parameter Detail----------------------
```

```
2019/12/24 13:26:33
-------------------------------------------------------
2019/12/24 13:26:33.706 8028 008 METHOD result DisplayTextAsync() -> Success(0)
```

* When this SDK works with logging, it performs uncomfortably because a log file will be updated at every method and accessing properties.

* Because of the following reasons or else, log data will not be stored without any notification.
  - A folder where is not under the local folder is specified.
  - A folder or file without permission is specified.
  - Write-protected log file is already existing.
  - Another program (such as a text editor) is using (locking) the log file.
  - There is not enough space to store log data in the device.

3.4.2. List of Constants Predefined

No	Type	Name	Data type	Value	Description
1	Result/Error	CDP_SUCCESS	int	0	Successfully completed
		CDP_E_CONNECTED	int	1001	Already connected
		CDP_E_DISCONNECT	int	1002	Not connected
		CDP_E_NOTCONNECT	int	1003	Failed to connect
		CDP_E_CONNECT_OFFLINE	int	1005	Failed printer status
		CDP_E_ILLEGAL	int	1101	Unsupported or invalid parameter
		CDP_E_OFFLINE	int	1102	Off-line
		CDP_E_FAILURE	int	1104	Process failure
2	Interface to connect	CDP_PORT_WiFi	int	0	Network
		CMP_PORT_Bluetooth	int	1	Bluetooth
3	Area to clear	CDP_AREA_ALL	int	0	Entire area
		CDP_AREA_CURSORLINE	int	1	Cursor line
4	Display mode	CDP_OVERWRITE	int	1	Overwrite mode
		CDP_VERTICALSCROLL	int	2	Vertical scroll mode
		CDP_HORIZONTALSCROLL	int	3	Horizontal scroll mode
5	Cursor type	CDP_TYPE_NONE	int	0	Hide cursor
		CDP_TYPE_UNDERLINE	int	1	Display cursor

4. Barcode Scanner Control

4.1. Program structure

Here is an example program in C# which uses the SDK

```
// Create an instance.
Scanner scanner = new Scanner();

// Data event definition.
void OnDataEvent(byte[] data)
{
    Debug.WriteLine("Data call back: " + Encoding.UTF8.GetString(data));
}

// Status event definition.
void OnStatusUpdateEvent(int status)
{
    Debug.WriteLine("Status update call back: " + status);
}

// Start scan.
void StartScan()
{
    // Add event handler.
    scanner.DataEvent += new DataEventHandler(OnDataEvent);
    scanner.StatusUpdateEvent +=
                 new StatusUpdateEventHandler(OnStatusUpdateEvent);

    // Connect scanner.
    int result = await scanner.ConnectAsync(ScannerConst.CSC_PORT_WiFi,
                 "192.168.0.10");
}


// Stop scan.
void StopScan()
{
    // disonnect scanner.
    await scanner.DisconnectAsync();

    // Delete event handler.
    scanner.DataEvent -= new DataEventHandler(OnDataEvent);
    scanner.StatusUpdateEvent -=
                 new StatusUpdateEventHandler(OnStatusUpdateEvent);
}
```

Class definition

Callback processes

Connect processes

Disconnect processes

4.2. Functions list

This SDK provides the following functions.

Methods list

No	Function	Detail
1	Connect scanner (connectAsync method)	This method connects to the scanner.
2	Disconnect scanner (DisconnectAsync method)	This method disconnects the scanner connection.
3	Get version code (getVersionCode method)	This method gets a numerical value for the version number of this SDK.
4	Get version name (getVersionName method)	This method gets a string for the version number of this SDK.
5	Log settings (SetLog method)	Set the log function.

Events list

No	Function	Detail
1	Input data (DataEvent event)	This event notifies data entry from the barcode scanner.
2	Update status (StatusUpdateEvent event)	This event notifies update status of the device.

4.3. Library interfaces

The followings are the interfaces of this SDK.

4.3.1. Return value

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

Return value	Description
CSC_SUCCESS (0)	The operation is success.
CSC_E_CONNECTED (1001)	The device is already connected.
CSC_E_DISCONNECT (1002)	The device is not connected.
CSC_E_NOTCONNECT (1003)	Failed connection to the device.
CSC_E_CONNECT_NOTFOUND (1004)	Failed to check the support model after connecting to the device.
CSC_E_CONNECT_OFFLINE (1005)	Failed to check the printer status after connecting to the device.
CSC_E_ILLEGAL (1101)	Unsupported operation with the Device, or an invalid parameter value was used.
CSC_E_OFFLINE (1102)	The printer is off-line.
CSC_E_NOEXIST (1103)	The file name does not exist.
CSC_E_FAILURE (1104)	The Service cannot perform the requested procedure.

4.3.2. Constructor

Syntax

Scanner ()

Parameter

Not exist.

Description

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

Return value

Not exist.

Example

```
Scanner scanner = new Scanner();
```

4.3.3. ConnectAsync method

Syntax

1) Task<int> ConnectAsync (int connectType, String addr)
2) Task<int> ConnectAsync (int connectType, String addr, int port)
3) Task<int> ConnectAsync (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
connectType	[IN]	Connection type of the printer	CSC_PORT_WiFi CSC_PORT_Bluetooth
addr	[IN]	IP address or BD address to connect	WiFi: 0.0.0.0～255.255.255.255 Bluetooth: 00:00:00:00:00:00~ FF:FF:FF:FF:FF:FF
port	[IN]	Connection port number	
timeout	[IN]	Timeout (msec)	

Description

This method is used to connect the barcode scanner. Please specify the type and address of the printer to which the barcode scanner is connected.

Connection port number is valid only if you specify the connection type CSC_PORT_WiFi. If it is omitted, you connected with number 9210.

Timeout is gives the maximum number of milliseconds to connect scanner. If it is omitted, you connected with 4000 milliseconds when using WiFi.

When communication with the scanner is not necessary, must execute the DisconnectAsync method to disconnect the scanner connection. When not disconnect, the next connection will be an error.

Return value

Return CSC_SUCCESS (0) in success. Please check the description of the error codes below in the case of failure. Please refer to "4.3.1 Return value" for the error code except it.

Error codes	Description
CSC_E_NOTCONNECT (1003)	Failed connection to the scanner. (1) The scanner is under none-connection status. (2) The printer is not turned ON. (3) Cannot obtain handle of interface board.

Example

```
await scanner.ConnectAsync(ScannerConst.CSC_PORT_WiFi, "192.168.0.10");
```

4.3.4. DisconnectAsync method

Syntax

Task<int> DisconnectAsync ()

Parameter

Not exist.

Description

This method is used to disconnect the barcode scanner connection.

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

Return value

Return CSC_SUCCESS(0) in success. Please refer to "4.3.1 Return value" for the error code except it.

Example

```
await scanner.Disconnect();
```

4.3.5. 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

```
int vno = scanner.GetVersionCode();
```

4.3.6. GetVersionName method

Syntax

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

```
String vname = scanner.GetVersionName();
```

4.3.7. SetLog method

Syntax

void SetLog (int mode, String path, int maxSize)

Parameters

Parameter	[IN/OUT]	Description	Setting range
mode	[IN]	Logging mode	0: None 1: Access logs 2: Error logs
path	[IN]	File path to store	
maxSize	[IN]	Maximum Log Size	0: Unlimited 1 - : Maximum size (MB)

Description

Sets the logging function. See "4.4.1 Logging function" for more details.

Return value

none

Example

```
scanner.SetLog(1, "Log", 10);
```

4.3.8. DataEvent event

Syntax

void DataEventHandler(byte[] status)

Parameter

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

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

Description

This event notifies data entry from the barcode scanner.

The event handler receives byte array type arguments as information read by the barcode scanner from the barcode.

Example

```
// Data event definition.
void OnDataEvent(byte[] data)
{
    Debug.WriteLine("Data call back: " + Encoding.UTF8.GetString(data));
}

// Add event handler.
scanner.DataEvent += new DataEventHandler(OnDataEvent);
```

4.3.9. StatusUpdate event

Syntax

void StatusUpdateEventHandler(int status)

Parameter

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

Value	[IN/OUT]	Meaning	Setting range

Status	[IN]	Status information	CSC_SUE_POWER_ONLINE(2001):
			Device is ready
			CSC_SUE_POWER_OFF(2002):
			Connection fault or not connected to the printer

Description

This event notifies update status of the device.

The event handler receives an int type argument indicating the status as information on the status change of the device.

Example

```
// Status event definition.
void OnStatusUpdateEvent(int status)
{
    Debug.WriteLine("Status update call back: " + status);
}

// Add event handler.
scanner.StatusUpdateEvent +=
        new StatusUpdateEventHandler(OnStatusUpdateEvent);
```

4.4. Notes

Notes of this SDK are as follows.

4.4.1. Logging function

This SDK supports the log function which records the methods and the events. When setting the log function, use the SetLog method, or placing a file "CSJPOSLibS.cfg" of the next format in the LocalFolder.

< Example of CSJPOSLibS.cfg >
[LogSetting] ... Section name (Fixed)
LogMode=1 ... Specifies the log mode.
LogPath=Log ... Specifies the relative path from the LocalFolder to store the log files.
LogMaxSize=10 ... Specifies the maximum size of log file in MB.

Setting items

- LogMode
  Specifies a log mode:
  0: None
  1: Access log
  2: Error log

- LogPath
  Specifies a folder which log files will be stored. It must be described as relative path from the local folder. When this setting is not specified, log files will be stored into the local folder.

- LogMaxSize
  Specifies maximum size of a log file in MB. If 0 is specified, log data will be written without limit.

Log file name

Log files will be stored with a file name "CSJPOSLibS_" and a number which indicates the day of week (0 to 6. 0: Sunday, 1: Monday...), and a file extension ".log."

Example: CSJPOSLibS_1.log

If a log file is already existing and it is older than today, it will be deleted, and the log data will be recorded into a new file. (It will be held for one week.)

Log format

A log file keeps information of the executed methods, accessed properties, timestamps and results.

```
--- Example 1, method (Connect) ---

2019/12/24 13:26:41.951 8028 008 METHOD call   ConnectAsync(0, "192.168.10.100")
2019/12/24 13:26:43.114 8028 008 METHOD result ConnectAsync() -> Success(0)

--- Example 2, event (DataEvent) ---
2019/12/24 13:26:49.568 8028 008 EVENT        DataEvent : 31 32 33 34 35 36 37 38 39 30
```

* When this SDK works with logging, it performs uncomfortably because a log file will be updated at every method and accessing properties.

* Because of the following reasons or else, log data will not be stored without any notification.

- A folder where is not under the local folder is specified.
- A folder or file without permission is specified.
- A write-protected log file is already existing.
- Another program (such as a text editor) is using (locking) the log file.
- There is not enough space to store log data in the device.

4.4.2. List of Constants Predefined

No	Type	Name	Data type	Value	Description
1	Result/Error	CSC_SUCCESS	int	0	Successfully completed
		CSC_E_CONNECTED	int	1001	Already connected
		CSC_E_DISCONNECT	int	1002	Not connected
		CSC_E_NOTCONNECT	int	1003	Failed to connect
		CSC_E_CONNECT_OFFLINE	int	1005	Failed printer status
		CSC_E_ILLEGAL	int	1101	Unsupported or invalid parameter
		CSC_E_OFFLINE	int	1102	Off-line
		CSC_E_FAILURE	int	1104	Process failure
2	Interface to connect	CSC_PORT_WiFi	int	0	Network
		CMP_PORT_Bluetooth	int	1	Bluetooth
3	Status	CSC_SUE_POWER_ONLINE	int	2001	Device is ready
		CSC_SUE_POWER_OFF	int	2002	Connection fault or not connected to the printer

CITIZEN UWP POS Print SDK - Programming Manual
November 21, 2023 For Ver. 2.00
CITIZEN SYSTEMS JAPAN CO., LTD.
https://csj.citizen.co.jp/