

CITIZEN

Android POS Print SDK Programming Manual

For Ver. 2.14

CITIZEN SYSTEMS JAPAN CO., LTD.

Revision Record

Date	Version	Description
Oct 12, 2012	1.00	New issue.
Nov 16, 2012	1.01	Updated the version number only.
Feb 14, 2013	1.02	Added a bitmap data type and mode to the printBitmap method.
Mar 18, 2013	1.03	Updated the required OS version of supported terminals. Added the USB host to the interface of supported terminals. Added the USB to the interface of supported models. Added the CT-S310II to the supported models. Added the setContext method. Added the CMP_E_NOCONTEXT to the return values. Added the USB connection to the connect method.
Jun 6, 2013	1.04	Added the ReLineSpacing property. Added the function of the automatic detection of the Bluetooth device to the connect method. Added the arguments of timeout to the connect method.
Oct 7, 2013	1.05	Added CT-S2000 and CT-S4000 to the support models.
Jan 17, 2014	1.06	Updated the version number only.
May 23, 2014	1.07	Added the example of Chinese to the setEncoding method.
Jul 24, 2014	1.08	Added Near Empty and Drawer Status to the status method. Added Font C and Reverse to the printText method. Added the arguments of byte array to the printBitmap method. Added GS1 DataBar to the printBarCode method. Added the printGS1DataBarStacked method. Added the searchESCPOSPrinter method. Added the MapMode property.
Aug 29, 2014	1.09	Updated the version number only.
Oct 10, 2014	1.10	Changed the description of PageModeArea property.
May 5, 2015	1.11	Added CT-S251 and CT-S601II/651II/801II/851II to the support models. Changed the description about the automatic detection function using the Bluetooth device in the connect method. Added CMP_MODE_CMD_GRAY16 to mode argument of the printBitmap method. Added the setNVBitmap method. Added the watermarkPrint method.
Apr 22, 2015	1.12	Changed the description about the multi-byte character setting of CT-S601II/651II/801II/851II. Added the example of EUC-KR and Big5 to the setEncoding method.
Jan 29, 2016	1.13	Added CT-S310II Wired LAN to the supported models. Modified the description in the connect method, the disconnect method and the printerCheck method. Added the searchCitizenPrinter method. Modified the description about the search of Bluetooth device in the searchESCPOSPrinter method.
Oct 19, 2016	1.14	Added "1.6 Program structure".
Nov 22, 2016	1.15	- Added CT-D150/CT-E351 to the support models. - Added the explanation of the log function.
Mar 3, 2017	1.16	- Added CMP-20/30/40/20II/30II to the support models. - Added the printPaddingText method. - Added the printTextPCFont method. - Added "Predefined Constants List".
Jun 19, 2017	1.17	Added CT-D151/CT-E651 to the support models.
Jul 12, 2017	1.18	Updated the version number only.
May 15, 2018		Modified the mistake in the example of rotatePrint method.
Jul 2, 2018	2.00	Added "1.5 Supported models (Peripheral Devices)". Added "3. Linedisplay Control". Added "4. Barcode Scanner Control".

Oct 24, 2018	2.01	Added CT-S751 to the printer support models. Added BC-NL3000U to the barcode scanner support models. Added "2.5.2 About printing UTF-8 encode characters". Added explanation of Bluetooth connection to "3.4.3 connect method". Added explanation of Bluetooth connection to "4.4.3 connect method".
Feb 6, 2019	2.02	Added CT-S4500 to the printer support models.
Jul 24, 2019	2.03	Updated the version number only.
Nov 11, 2019	2.04	Added PMU3300 to the printer support models.
Dec 26, 2019	2.05	Updated the version number only.
Mar 30, 2020	2.06	Added the paper is hold status to the Status method.
Oct 7, 2020	2.07	Added CMP_MODE_CMD_GRAY16DOWNLOAD to mode argument of the printBitmap method. Added the setPrintCompletedTimeout method.
Mar 24, 2021	2.08	Added CT-D101/CT-E301/CT-E601 to the printer support models. Modified the PageModeArea property of the PMU3300.
Apr 14, 2022	2.09	Added the printerCheckEx and openDrawerEx method.
Aug 30, 2022	2.10	Added CT-S281II to the printer support models.
Sep 8, 2023	2.11	Added RecCharacterSpacing, RecKanjiLeftSpacing, RecKanjiRightSpacing properties to the printer control. Added ErrorCodeExtended property to the printer and the display and the scanner control.
Nov 21, 2023		Added CT-D801III/CT-S851III to the printer support models. Added DSP01-LT2/DSP02-LS2 to the display support models.
Jan 24, 2024	2.12	Updated the version number and correction of errors.
Apr 3, 2025	2.13	Modified the description of the log function. (Page 71, 87, 115, 117, 128, 132)
Aug 14, 2025	2.14	Updated the version number only.

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 SDK.

Trademark

Android is registered trademarks of Google Inc. in the United States and/or other countries.

Java is registered trademarks of Sun Microsystems, Inc. in the United States and/or other countries.

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

Table of Contents

1. Introduction	8
1.1. Document target range	8
1.2. System summary	8
1.3. Supported terminals	8
1.4. Supported models (Printers)	9
1.5. Supported models (Peripheral Devices)	20
2. Printer Control	23
2.1. Definition method	23
2.2. Program structure	23
2.3. Functions list	24
2.4. Library interfaces	26
2.4.1. <i>Return value</i>	26
2.4.2. <i>Constructor</i>	27
2.4.3. <i>setContext method</i>	28
2.4.4. <i>connect method</i>	29
2.4.5. <i>disconnect method</i>	31
2.4.6. <i>setEncoding method</i>	32
2.4.7. <i>printerCheck method</i>	33
2.4.8. <i>status method</i>	34
2.4.9. <i>printText method</i>	36
2.4.10. <i>printPaddingText method</i>	37
2.4.11. <i>printTextLocalFont method</i>	39
2.4.12. <i>printBitmap method</i>	40
2.4.13. <i>setNVBitmap method</i>	42
2.4.14. <i>printNVBitmap method</i>	44
2.4.15. <i>printBarCode method</i>	45
2.4.16. <i>printPDF417 method</i>	47
2.4.17. <i>printQRCode method</i>	48
2.4.18. <i>printGS1DataBarStacked method</i>	49
2.4.19. <i>cutPaper method</i>	50
2.4.20. <i>unitFeed method</i>	51
2.4.21. <i>markFeed method</i>	52
2.4.22. <i>openDrawer method</i>	53
2.4.23. <i>transactionPrint method</i>	54
2.4.24. <i>rotatePrint method</i>	55
2.4.25. <i>pageModePrint method</i>	56
2.4.26. <i>clearPrintArea method</i>	58
2.4.27. <i>clearOutput method</i>	59
2.4.28. <i>printData method</i>	60
2.4.29. <i>printNormal method</i>	61
2.4.30. <i>watermarkPrint method</i>	62
2.4.31. <i>searchCitizenPrinter method</i>	63
2.4.32. <i>searchESCPOSPrinter method</i>	65
2.4.33. <i>printerCheckEx method</i>	67
2.4.34. <i>openDrawerEx method</i>	69
2.4.35. <i>setPrintCompletedTimeout method</i>	70
2.4.36. <i>setLog method</i>	71

2.4.37. <i>getVersionCode</i> method.....	72
2.4.38. <i>getVersionName</i> method.....	73
2.4.39. <i>PageModeArea</i> property.....	74
2.4.40. <i>PageModePrintArea</i> property.....	75
2.4.41. <i>PageModePrintDirection</i> property.....	76
2.4.42. <i>PageModeHorizontalPosition</i> property.....	77
2.4.43. <i>PageModeVerticalPosition</i> property.....	78
2.4.44. <i>RecLineSpacing</i> property.....	79
2.4.45. <i>RecCharacterSpacing</i> property.....	80
2.4.46. <i>RecKanjiLeftSpacing</i> property.....	81
2.4.47. <i>RecKanjiRightSpacing</i> property.....	82
2.4.48. <i>MapMode</i> property.....	83
2.4.49. <i>ErrorCodeExtended</i> property.....	85
2.5. Notes.....	86
2.5.1. <i>Function to detect the completion of printing</i>	86
2.5.2. <i>About printing UTF-8 encode characters</i>	86
2.5.3. <i>Logging function</i>	87
2.5.4. <i>Predefined Constants List</i>	90
3. Line Display Control	93
3.1. Definition method.....	93
3.2. Program structure.....	93
3.3. Functions list.....	94
3.4. Library interfaces.....	95
3.4.1. <i>Return value</i>	95
3.4.2. <i>Constructor</i>	96
3.4.3. <i>connect</i> method.....	97
3.4.4. <i>disconnect</i> method.....	98
3.4.5. <i>displayText</i> method.....	99
3.4.6. <i>clearDisplay</i> method.....	100
3.4.7. <i>blinkDisplay</i> method.....	101
3.4.8. <i>setDisplayMode</i> method.....	102
3.4.9. <i>setDisplayConfig</i> method.....	103
3.4.10. <i>setCursorPosition</i> method.....	104
3.4.11. <i>moveCursor</i> method.....	105
3.4.12. <i>setCursorType</i> method.....	106
3.4.13. <i>initializeDisplay</i> method.....	107
3.4.14. <i>displayData</i> method.....	108
3.4.15. <i>setEncoding</i> method.....	109
3.4.16. <i>setCodePage</i> method.....	110
3.4.17. <i>setInternationalCharacteraset</i> method.....	111
3.4.18. <i>displayCheck</i> method.....	112
3.4.19. <i>getVersionCode</i> method.....	113
3.4.20. <i>getVersionName</i> method.....	114
3.4.21. <i>setLog</i> method.....	115
3.4.22. <i>ErrorCodeExtended</i> property.....	116
3.5. Notes.....	117
3.5.1. <i>Logging function</i>	117
3.5.2. <i>Predefined Constants List</i>	119
4. Barcode Scanner Control	120

4.1. Definition method	120
4.2. Program structure.....	120
4.3. Functions list	121
4.4. Library interfaces.....	122
4.4.1. <i>Return value</i>	122
4.4.2. <i>Constructor</i>	123
4.4.3. <i>connect method</i>	124
4.4.4. <i>disconnect method</i>	125
4.4.5. <i>getVersionCode method</i>	126
4.4.6. <i>getVersionName method</i>	127
4.4.7. <i>setLog method</i>	128
4.4.8. <i>ErrorCodeExtended property</i>	129
4.4.9. <i>DataEvent event</i>	130
4.4.10. <i>StatusUpdateEvent event</i>	131
4.5. Notes	132
4.5.1. <i>Logging function</i>	132
4.5.2. <i>Predefined Constants List</i>	134

1.Introduction

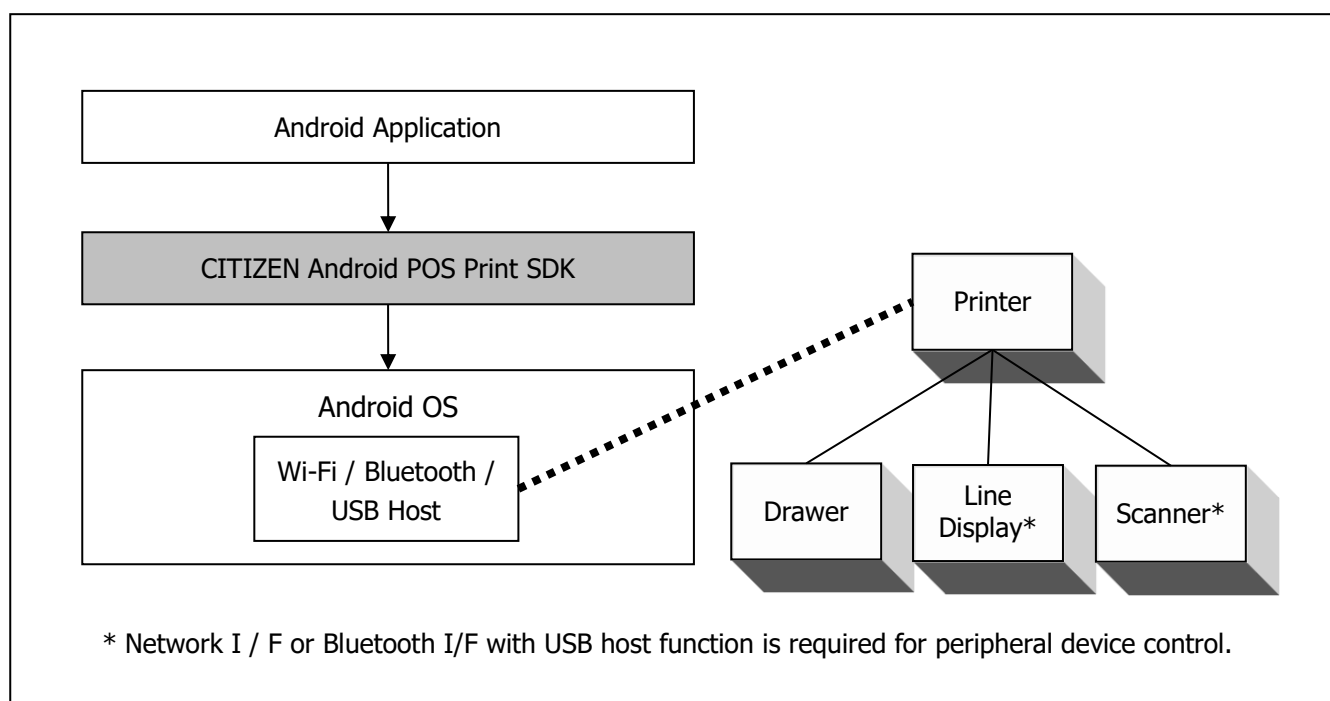
This document is a programming manual for the CITIZEN Android POS Print SDK.

1.1. Document target range

This document is intended for developers who integrate their Android application programs with CITIZEN POS printers and peripheral devices connected to the printer.

1.2. System summary

This SDK is referred by Android application program to control CITIZEN POS printers and peripheral devices connected to the printer.



System diagram of the SDK

1.3. Supported terminals

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

Android OS version:	Supported interfaces:
Android 2.3.3 (API Level 10) or newer	Wi-Fi, Bluetooth
Android 3.1 (API Level 12) or newer	Wi-Fi, Bluetooth, USB Host

1.4. Supported models (Printers)

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 USB	Standard
CT-D150 Series	CT-D150	Wired LAN USB	Standard
CT-D151 Series	CT-D151	Bluetooth Wired LAN	Standard
	CT-D151-L	Wireless LAN USB	Label/Blackmark paper is supported.
CT-E301 Series	CT-E301	Wired LAN USB	Standard
CT-E351 Series	CT-E351	Wired LAN USB	Standard
CT-E601 Series	CT-E601	Bluetooth Wired LAN Wireless LAN USB	Standard
CT-E651 Series	CT-E651	Bluetooth Wired LAN	Standard
	CT-E651-L	Wireless LAN USB	Label/Blackmark paper is supported.
CT-S251 Series	CT-S251	Bluetooth Wired LAN Wireless LAN USB	Standard
CT-S281 Series	CT-S281	Bluetooth USB	Standard
	CT-S281-L		Label/Blackmark paper is supported.
CT-S281II Series	CT-S281II	USB	Standard
	CT-S281II-L		Label/Blackmark paper is supported.
CT-S310II Series	CT-S310II	Wired LAN USB	Standard
CT-S601/651/801/851 Series	CT-S601/651/801/851	Wired LAN Wireless LAN USB	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	Bluetooth Wired LAN Wireless LAN USB	Standard
	CT-S801II/851II-M		Blackmark paper is supported.
	CT-S801II-L		Label paper is supported.
CT-S801III/851III Series	CT-S801III/851III	Bluetooth Wired LAN Wireless LAN USB	Standard

CT-S751 Series	CT-S751	Bluetooth Wired LAN Wireless LAN USB	Standard
CT-S2000 Series	CT-S2000	Wired LAN USB	Standard
	CT-S2000-M		Blackmark paper is supported.
	CT-S2000-L		Label paper is supported.
CT-S4000 Series	CT-S4000	Wired LAN USB	Standard (Paper with blackmark on front side is supported.)
	CT-S4000-M		Paper with blackmark on back side is supported.
	CT-S4000-L		Label paper is supported.
CT-S4500 Series	CT-S4500	Bluetooth Wired LAN Wireless LAN USB	Standard (Label/Blackmark paper is supported.)
PMU2300III Series	PMU2300III	USB	Standard (With presenter)
PMU3300 Series	PMU3300	USB	Standard (With paper exit sensor)
CMP-20/30/40/20II/ 30II Series	CMP-20/30/40/20II/ 30II (ESC/POS)	Bluetooth Wireless LAN USB	Standard

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

CT-D101 Series Memory Switch Setting

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

CT-D150 Series Memory Switch Setting

MSW No.	Function	Setting
1-1	Power ON Info	Valid
1-2	Buffer Size	4K bytes

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

CT-D151 Series Memory Switch Setting

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

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

CT-E351 Series Memory Switch Setting

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

CT-E601 Series Memory Switch Setting

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

MSW No.	Function	Setting
3-8	Resume Open Err	Close
4-8	Partial Only	Invalid
5-2	Line Pitch	1/360
5-3	USB Mode	Printer Class
6-1	Act. For Driver	Valid
7-6	DMA control	Valid
9-1	Code Page	Katakana (*1)
9-2	Int'Char Set	Japan (*1)
9-4	Multi-byte Char (*2)	SJIS(CP932) GB18030 EUC Hangul BIG5
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 Ctrr Err	Valid
3-7	CBM1000 Mode	Valid
3-8	Resume Open Err	Close
4-8	Partial Only	Invalid
5-2	Line Pitch	1/360
5-3	USB Mode	Printer Class
6-1	Act. For Driver	Valid
7-6	DMA control	Valid
9-1	Code Page	Katakana (*1)
9-2	Int'Char Set	Japan (*1)
9-4	Multi-byte Char (*2)	SJIS(CP932) GB18030 EUC Hangul BIG5
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 Ctrr Err	Valid
3-7	CBM1000-compatible mode	Valid

MSW No.	Function	Setting
3-8	Resume Open Err	Close
4-8	Partial Only	Invalid
5-2	Line Pitch	1/360
6-1	Act. For Driver	Valid
7-6	DMA control	Valid
9-1	Code page	Katakana (*1)
9-2	Int'Char Set	Japan (*1)
9-3	Kanji	ON (*1)
9-4	JIS/Shift-JIS	Shift-JIS (*1)
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 Ctrr Err	L/F enabled
3-7	CBM-270-compatible mode	Valid
3-8	Resume Open Err	Close
4-8	Partial Only	Invalid
5-3	USB Mode	Printer Class
9-1	Code page	Katakana (*1)
9-2	Int'Char Set	Japan (*1)
9-3	Kanji	ON (*1)
9-4	JIS/Shift-JIS	Shift-JIS (*1)
13-6	Auto Reconnect (When Bluetooth I/F is used)	Invalid

CT-S281II Series Memory Switch Setting

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

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

CT-S310II Series Memory Switch Setting

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

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

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

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

MSW No.	Function	Setting
1-1	Power ON Info	Valid
1-2	Buffer Size	4K bytes

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

CT-S4000 Series Memory Switch Setting

MSW No.	Function	Setting
1-1	Power ON Info	Valid

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

CT-S4500 Series Memory Switch Setting

MSW No.	Function	Setting
1-1	Power ON Info	Valid
1-2	Buffer Size	4K bytes
1-3	Busy condition	Full
1-4	Receiving Error	Print ?
1-5	CR Mode	Ignored
2-2	Auto cutter	Valid
2-4	Full Col Print	Wait Data
3-1	Resume Ctrr Err	Valid
3-7	CBM1000 Mode	Valid
3-8	Resume Open Err	Close
4-8	Partial Only	Invalid
5-2	Line Pitch	1/360
5-3	USB Mode	Printer Class
6-1	Act. For Driver	Valid
7-6	DMA control	Valid
9-1	Code Page	Katakana (*1)
9-2	Int'Char Set	Japan (*1)
9-4	Multi-byte Char (*2)	SJIS(CP932) GB18030 EUC Hangul BIG5-HKSCS
13-6	Auto Reconnect (When Bluetooth I/F is used)	Invalid

PMU2300III Series Memory Switch Setting

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

1-4	Receiving Error	Print ?
1-5	CR Mode	Ignored
2-2	Auto cutter	Valid
2-4	Full Col Print	Wait Data
3-7	CBM1000 Mode	Valid
3-8	Resume Open Err	Close
4-8	Partial Only	Invalid
5-3	USB Mode	Printer Class
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)

PMU3300 Series Memory Switch Setting

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

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

*2 The CT-D101/150/151, CT-E301/351/601/651, CT-S601II/651II/801II/851II/801III/851III/751/4500 series can change the Multi-byte character to Shift_JIS, GB18030, EUC-KR and Big5. Please change it by use environment.

Firmware

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

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

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

1.5. Supported models (Peripheral Devices)

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

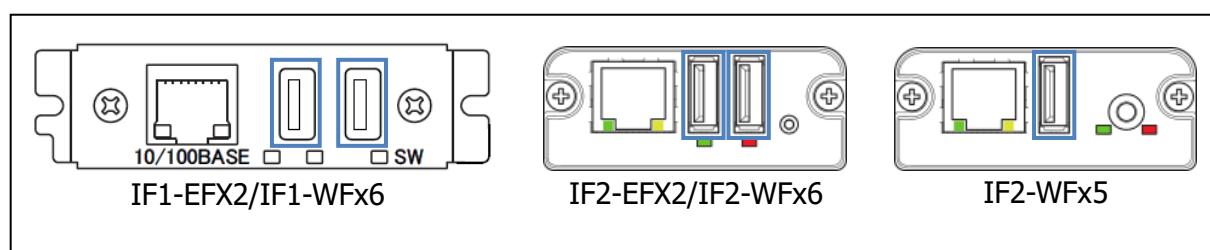
Check that the scanner to be connected is set as follows. For the setting procedure, refer to the instruction manual of the peripheral device to be used.

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

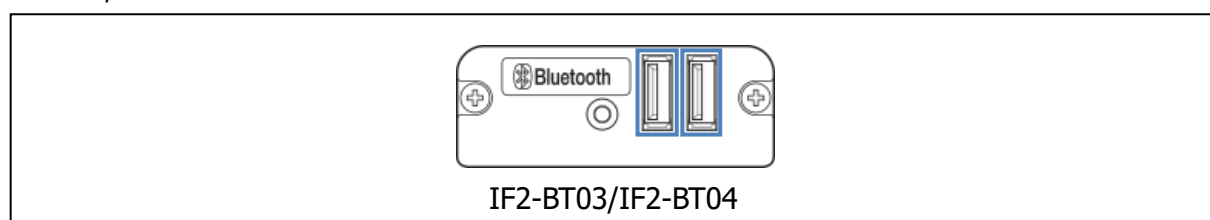
About connection to printer

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.

Network I/F



Bluetooth I/F



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.

The screenshot shows the 'LAN board' web interface for 'CITIZEN SYSTEMS'. The 'Service' tab is selected. Under the 'Media Converter' section, 'VCOM Convert' and 'HID Scanner Convert' are both set to 'Enable'. These two rows are highlighted with a red rectangle. Below this, the 'XML Print' section shows port and timeout settings. The 'XML Device Control' section shows port, timeout, and max connection settings. The 'XML Device Control / Line Display' section shows baud rate and data settings.

Section	Item	Value	Range / Unit
Media Converter	VCOM Convert	Enable	
	HID Scanner Convert	Enable	
XML Print	Port Number	8080	
	Time out for connect	10	5-60[Seconds]
	Time out for print	60	10-600[Seconds]
XML Device Control	Port Number	8085	
	Time out for connect	10	5-180[Seconds]
	Maxconnection	2	
	XML Device Control / Line Display	Baud rate	9600
	Data	8 bit	

Copyright © 2012 CITIZEN SYSTEMS JAPAN CO.,LTD. All rights reserved.

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 (Factory Default) processing is done, set each setting value to the initial value.

2.Printer Control

2.1. Definition method

Add the library file that is provided, if you use this SDK, please import the following two classes.

```
import com.citizen.sdk.ESCPOSConst;
import com.citizen.sdk.ESCPOSPrinter;
```

2.2. Program structure

Here is an example program which uses the SDK.

<pre>// Create an instance. ESCPOSPrinter printer = new ESCPOSPrinter(); // Connect printer int result = printer.connect(ESCPOSConst.CMP_PORT_WiFi, "192.168.0.10"); if (ESCPOSConst.CMP_SUCCESS == result) { // Set encoding printer.setEncoding("ISO-8859-1"); // Start Transaction (Batch) printer.transactionPrint(ESCPOSConst.CMP_TP_TRANSACTION); // Print Text printer.printText("Citizen POS sample1\n\n", ESCPOSConst.CMP_ALIGNMENT_CENTER, ESCPOSConst.CMP_FNT_DEFAULT, ESCPOSConst.CMP_TXT_1WIDTH ESCPOSConst.CMP_TXT_1HEIGHT); printer.printText("- Sample Print 1 -\n", ESCPOSConst.CMP_ALIGNMENT_CENTER, ESCPOSConst.CMP_FNT_DEFAULT, ESCPOSConst.CMP_TXT_1WIDTH ESCPOSConst.CMP_TXT_2HEIGHT); printer.printText("123456789012345678901234567890\n", ESCPOSConst.CMP_ALIGNMENT_RIGHT, ESCPOSConst.CMP_FNT_DEFAULT, ESCPOSConst.CMP_TXT_1WIDTH ESCPOSConst.CMP_TXT_1HEIGHT); // Print QRcode printer.printQRCode("http://www.citizen-systems.co.jp/", 6, ESCPOSConst.CMP_QRCODE_EC_LEVEL_L, ESCPOSConst.CMP_ALIGNMENT_RIGHT); // Partial Cut with Pre-Feed printer.cutPaper(ESCPOSConst.CMP_CUT_PARTIAL_PREFEED); // End Transaction (Batch) result = printer.transactionPrint(ESCPOSConst.CMP_TP_NORMAL); // Disconnect printer.disconnect(); if (ESCPOSConst.CMP_SUCCESS != result) { // Print process Error Log.e("Citizen_POS_sample1", "Transaction Error : " + Integer.toString(result)); } } else { // Connect Error Log.e("Citizen_POS_sample1", "Connect or Printer Error : " + Integer.toString(result)); } }</pre>	<div style="margin-bottom: 10px;">} Class definition</div> <div style="margin-bottom: 10px;">} Connect</div> <div style="margin-bottom: 10px;">} Print processes</div> <div style="margin-bottom: 10px;">} Disconnect</div>
------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

2.3. Functions list

This SDK provides the following functions.

Methods list

No	Function	Detail
1	Create class (Constructor)	This is constructor method.
2	Set Context (setContext method)	Set the context.
3	Connect printer (connect method)	Connect to the printer.
4	Disconnect printer (disconnect method)	Disconnect the printer connection.
5	Set encoding (setEncoding method)	Set the encoding of character.
6	Check printer status (printerCheck method)	Sends command for status check of the printer.
7	Get printer status (status method)	Get the status of the printer.
8	Print text (printText method)	Prints text data.
9	Print space padding text (printPaddingText method)	Prints text data with space padding.
10	Print local font text (printTextLocalFont method)	Prints text data using a font installed in the terminal.
11	Print bitmap (printBitmap method)	Prints a bitmap file. (BMP/JPG/PNG/GIF format)
12	Store NV bitmap (setNVBitmap method)	Stores a bitmap image in the flash memory.
13	Print NV bitmap (printNVBitmap method)	Prints a bitmap image that is stored in the flash memory.
14	Print BarCode (printBarCode method)	Prints a one-dimensional barcode.
15	Print PDF-417 (printPDF417 method)	Prints a PDF417 barcode.
16	Print QRcode (printQRCode method)	Prints a QRCode barcode.
17	Print 2D GS1DataBar (printGS1DataBarStacked method)	Prints a 2-dimensional GS1DataBar barcode.
18	Cut paper (cutPaper method)	Cuts the paper.
19	Feed dot units (unitFeed method)	Feeds the paper forward by dot units.
20	Feed mark (markFeed method)	Support for label / black mark paper.
21	Open drawer (openDrawer method)	Opens the drawer.
22	Transaction print (transactionPrint method)	Enters or exits transaction mode.
23	Rotate print (rotatePrint method)	Enters or exits rotated print mode. (180°)
24	PageMode print (pageModePrint method)	Enters or exits page mode.
25	PageMode clear print area (clearPrintArea method)	Clear the area of the page mode print area.
26	Clear output data (clearOutput method)	Clears all buffered output data. (data and printer buffer)

27	Output data (printData method)	Sends to the printer without changing the data.
28	Print OPOS format (printNormal method)	Prints text using OPOS escape sequences.
29	Watermark print (watermarkPrint method)	Enters or exits watermark print mode.
30	Search printer (searchCitizenPrinter method)	Search the printer and get the list of printer information.
31	Search printer (searchESCPOSPrinter method)	Search the printer and get the list of addresses.
32	Connect and Check printer status (PrinterCheckEx method)	Connect and get the status of the printer.
33	Connect and Open Drawer (OpenDrawerEx method)	Connect and open the drawer.
34	Set print completed timeout (setPrintCompletedTimeout method)	Set the timeout to check the print completion notification.
35	Set log (setLog method)	Sets the log function.
36	Get version code (getVersionCode method)	Get a numerical value for the version number of this SDK.
37	Get version name (getVersionName method)	Get a string for the version number of this SDK.

Properties List

No	Function	Attribute	Detail
1	PageMode area (PageModeArea property)	R	Shows the page area of page mode.
2	PageMode print area (PageModePrintArea property)	R/W	Shows the print area of page mode.
3	PageMode print direction (PageModePrintDirection property)	R/W	Shows the print direction of page mode.
4	PageMode horizontal position (PageModeHorizontalPosition property)	R/W	Shows the horizontal start position offset within the print area of page mode.
5	PageMode vertical position (PageModeVerticalPosition property)	R/W	Shows the vertical start position offset within the print area of page mode.
6	Line spacing (RecLineSpacing property)	R/W	Show the spacing of each single-high print line.
7	Character spacing (RecCharacterSpacing property)	R/W	Show the right spacing of Character.
8	Kanji left spacing (RecKanjiLeftSpacing property)	R/W	Show the left spacing of Kanji.
9	Kanji right spacing (RecKanjiRightSpacing property)	R/W	Show the right spacing of Kanji.
10	Mapping mode (MapMode property)	R/W	Show the mapping mode (the unit of measure) of the printer.
11	Extended error code (ErrorCodeExtended property)	R	Show the extended error code when connecting.

2.4. Library interfaces

The following are the interfaces of this SDK.

2.4.1. 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_NOCONTEXT (1006)	The context is not specified.
CMP_E_BT_DISABLE (1007)	The setting of the Bluetooth device is invalid.
CMP_E_BT_NODEVICE (1008)	The Bluetooth device is not found.
CMP_E_ILLEGAL (1101)	Unsupported operation with the Device, or an invalid parameter value was used.
CMP_E_OFFLINE (1102)	The printer is off-line.
CMP_E_NOEXIST (1103)	The file name does not exist.
CMP_E_FAILURE (1104)	The Service cannot perform the requested procedure.
CMP_E_TIMEOUT (1105)	The Service timed out waiting for a response from the printer.
CMP_E_NO_LIST (1106)	The printer cannot be found in the printer search.
CMP_EPTR_COVER_OPEN (1201)	The cover of the printer opens.
CMP_EPTR_REC_EMPTY (1202)	The printer is out of paper.
CMP_EPTR_BADFORMAT (1203)	The specified file is in an unsupported format.
CMP_EPTR_TOOBIG (1204)	The specified bitmap is either too big.

2.4.2. Constructor

Syntax

ESCPOSPrinter ()

Parameter

Not exist.

Description

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

Return value

Not exist.

Example

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

2.4.3. setContext method

Syntax

```
int setContext (Context context)
```

Parameter

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

Value	[IN/OUT]	Meaning	Setting range
Context	[IN]	Context (android.content.Context)	Context:

Description

This method is used to set the context.

Must be executed this method before the [connect method](#) on the USB connection and the [searchESCPOSPrinter method](#).

Return value

Return CMP_SUCCESS (0) in success. Please refer to "[2.4.1 Return value](#)" for the error code except it.

Example

```
printer.setContext (this);
printer.setContext (getApplicationContext());
```

2.4.4. connect method

Syntax

- 1) int connect (int connectType, String addr)
- 2) int connect (int connectType, String addr, int port)
- 3) int connect (int connectType, UsbDevice device)

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 CMP_PORT_Bluetooth_Insecure CMP_PORT_USB
addr	[IN]	IP address to connect or Bluetooth device address or Bluetooth device name	WiFi: 0.0.0.0~255.255.255.255 Bluetooth: 00:00:00:00:00:00~FF:FF:FF:FF:FF:FF Device name (Automatic detection)
device	[IN]	UsbDevice class (android.hardware.usb.UsbDevice)	null: Automatic detection UsbDevice: Any USB device
port	[IN]	Connection port number	
timeout	[IN]	Timeout (msec)	

Description

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

Bluetooth device address letters, please specify in uppcase.

If the Bluetooth Device name is specified, the device that was paired is detected automatically. If you omit the Device name, the supported model device that was paired is detected automatically.

If you want to use the Bluetooth device insecure communications provided by the Android 2.3.3 or later, please specify the connection type CMP_PORT_Bluetooth_Insecure.

If you want to use the USB device, must execute the [setContext method](#) before the execution of this method.

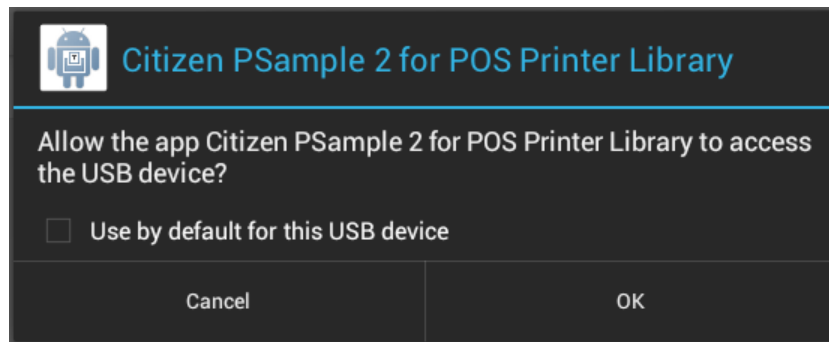
Connection port number is valid only if you specify the connection type CMP_PORT_WiFi. If it is omitted, you connected with number 9100.

Timeout is gives the maximum number of milliseconds to connect printer. Timeout is invalid if you specify the connection type CMP_PORT_USB. If it is omitted, you connected with 4000 milliseconds when using WiFi and connected with 8000 milliseconds when using Bluetooth.

When connecting to the printer, this SDK also checks the status of the printer and the supporting models. When communication with the printer is not necessary, must execute the [disconnect method](#) to disconnect the printer connection. When not disconnect, the next connection will be an error.

Note

When you first connect with USB, a dialog asking permission to access the USB device on the Android terminal will be displayed, please tap the OK button.



Access to the USB device dialog (example)

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 "[2.4.1 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.
CMP_E_NOCONTEXT (1006)	The context is not specified.
CMP_E_BT_DISABLE (1007)	The setting of the Bluetooth device is invalid.
CMP_E_BT_NODEVICE (1008)	The Bluetooth device is not found.

Example

```
printer.connect( ESCPOSConst.CMP_PORT_WiFi, "192.168.182.100" );

printer.connect( ESCPOSConst.CMP_PORT_Bluetooth, "00:01:90:F0:81:AB" );

UsbDevice usbDevice = null;
printer.connect( ESCPOSConst.CMP_PORT_USB, usbDevice );
```

2.4.5. disconnect method

Syntax

```
int disconnect ()
```

Parameter

Not exist.

Description

This method is used to disconnect the printer connection.

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

Return value

Return CMP_SUCCESS(0) in success. Please refer to "[2.4.1 Return value](#)" for the error code except it.

Example

```
printer.disconnect();
```

2.4.6. 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	The character set name that is supported depends on the operating environment.

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 "[1.4 Supported models \(Printers\)](#)")

This SDK supports printing UTF-8 encoded characters. Please refer to "[2.5.2 About printing UTF-8 encode characters](#)" for the detail.

Return value

Return CMP_SUCCESS (0) in success. Please refer to "[2.4.1 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" );
```


2.4.7. printerCheck method

Syntax

int printerCheck ()

Parameter

Not exist.

Description

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

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

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

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

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

Return value

Return CMP_SUCCESS (0) in success. Please refer to "[2.4.1 Return value](#)" for the error code except it.

Example

```
if ( ESCPOSConst.CMP_SUCCESS == posPtr.printerCheck() ) {  
    // Success  
} else {  
    // Fail  
}
```

2.4.8. status method

Syntax

- 1) int status ()
- 2) int status (int type)

Parameter

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

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

Description

This method is used to get the status of the printer obtained by the [printerCheck method](#).

Before the execution of this method, you must run the [printerCheck method](#).

When there is not a parameter, return the logical sum of the status (CMP_STS_COVER_OPEN, CMP_STS_PAPER_EMPTY, CMP_STS_PRINTEROFF) indicating the error of the printer.

When the status type is specified, return the status that matches. Status type can be specified in combination. If you want to combine, please specify the logical sum.

Return value

Return the following status codes.

status codes	Description
CMP_STS_NORMAL (0)	The printer is normal.
CMP_STS_PRINTEROFF (128)	The printer is off-line.
CMP_STS_MSR_READ (64)	Currently MSR in read mode. (CMP-20/30/40/20II/30II only)
CMP_STS_PAPER_EMPTY (32)	The printer is out of paper.
CMP_STS_COVER_OPEN (16)	The cover of the printer opens.
CMP_STS_BATTERY_LOW (8)	Printer battery capacity is low. (CMP-20/30/40/20II/30II only)
CMP_STS_PAPER_NEAREMPTY (4)	Paper near empty. (when the type parameter is set)
CMP_STS_DRAWER_LEVEL_H (2)	Status of pin 3 of drawer kick-out connector = H (when the type parameter is set)
CMP_STS_ONPRESENTER (1)	Status the paper is hold on the presenter or the paper exit sensor. (PMU2300III/PMU3300 only, when the type parameter is set)

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

2.4.9. printText method

Syntax

int printText (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 ["2.4.1 Return value"](#) for the error code except it.

Example

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

2.4.10. printPaddingText method

Syntax

int printPaddingText (String data, int attribute, int textSize, int length, int side)

Parameter

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

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

Description

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

Cannot use the combining characters in the text data.

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

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

Return value

Return CMP_SUCCESS (0) in success. Please refer to "[2.4.1 Return value](#)" for the error code except it.

Example

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

// Line 1
printer.printPaddingText( "Sandwich",
    ESCPOSConst.CMP_FNT_DEFAULT, ESCPOSConst.CMP_TXT_1WIDTH,
    nameSize, ESCPOSConst.CMP_SIDE_RIGHT );
printer.pPrintPaddingText( "5.00",
    ESCPOSConst.CMP_FNT_DEFAULT, ESCPOSConst.CMP_TXT_1WIDTH,
    priceSize, ESCPOSConst.CMP_SIDE_LEFT );
printer.printNormal("\n");

// Line 2
printer.printPaddingText( "Hamburg steak",
    ESCPOSConst.CMP_FNT_DEFAULT, ESCPOSConst.CMP_TXT_1WIDTH,
    nameSize, ESCPOSConst.CMP_SIDE_RIGHT );
printer.printPaddingText( "12.00",
    ESCPOSConst.CMP_FNT_DEFAULT, ESCPOSConst.CMP_TXT_1WIDTH,
    priceSize, ESCPOSConst.CMP_SIDE_LEFT );
printer.printNormal("\n");

// Line 3
printer.printPaddingText( "Coffee",
    ESCPOSConst.CMP_FNT_DEFAULT, ESCPOSConst.CMP_TXT_1WIDTH,
    nameSize, ESCPOSConst.CMP_SIDE_RIGHT );
printer.printPaddingText( "2.00",
    ESCPOSConst.CMP_FNT_DEFAULT, ESCPOSConst.CMP_TXT_1WIDTH,
    priceSize, ESCPOSConst.CMP_SIDE_LEFT );
printer.printNormal("\n");
```

2.4.11. printTextLocalFont method

Syntax

```
int printTextLocalFont (String data, int alignment, Typeface fontType, int point, int style,
                        int hRatio, int vRatio)
```

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
fontType	[IN]	Font type [Typeface]	The supported fonts depend on the OS implementation.
point	[IN]	Font size	Points
style	[IN]	Font style	CMP_FNT_DEFAULT: Default font CMP_FNT_BOLD: Bold CMP_FNT_REVERSE: Reverse CMP_FNT_UNDERLINE: Underline CMP_FNT_ITALIC: Italic CMP_FNT_STRIKEOUT: Strikethrough
hRatio	[IN]	Horizontal ratio [%]	1 - 1000
vRatio	[IN]	Vertical ratio [%]	1 - 1000

Description

This method is used to print text by using a font installed in the computer, which specifies alignment, font, size, style, and ratio.

What this method does internally is to generate a graphic image based on the given parameters, to print the graphic image.

Font style can be specified in combination bold, reverse, underline, italic and strikeout. If you want to combine, please specify the logical sum.

Return value

Return CMP_SUCCESS (0) in success. Please refer to "[2.4.1 Return value](#)" for the error code except it.

Example

```
printer.printTextLocalFont( "Print local font text data.\n",
    ESCPOSConst.CMP_ALIGNMENT_CENTER,
    Typeface.SERIF, 12,
    ESCPOSConst.CMP_FNT_BOLD | ESCPOSConst.CMP_FNT_UNDERLINE,
    100, 100 );
```

2.4.12. printBitmap method

Syntax

- 1) int printBitmap (String fileName, int alignment)
- 2) int printBitmap (String fileName, int width, int alignment)
- 3) int printBitmap (String fileName, int width, int alignment, int mode)
- 4) int printBitmap (Bitmap bitmap, int alignment)
- 5) int printBitmap (Bitmap bitmap, int width, int alignment)
- 6) int printBitmap (Bitmap bitmap, int width, int alignment, int mode)
- 7) int printBitmap (byte[] bytes, int alignment)
- 8) int printBitmap (byte[] bytes, int width, int alignment)
- 9) int printBitmap (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]	Bitmap 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) raster command output CMP_BM_MODE_CMD_GRAY16DOWNLOAD: Grayscale (4bpp) download graphics command output

Description

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

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

If the bitmap width is omitted, printing in CMP_BM_ASIS.

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	Bitimage command output Suitable for large data printing. In order to output the split data, there is no height limit.
CMP_BM_MODE_CMD_GRAY16	Grayscale(4bpp) raster command output Available in CT-D151, CT-E601/651, CT-S251/601II/651II/801II/851II/801III/851III/751. Graphic can be printed more beautifully. The data is split output.
CMP_BM_MODE_CMD_GRAY16DOWNLOAD	Grayscale(4bpp) download graphics command output Available in CT-D151, CT-E601/651, CT-S251/601II/651II/801II/851II/801III/851III/751. Graphic can be printed more beautifully. In order to output the data collectively, there is a 384KB limit on the size of 4bpp.

Return value

Return CMP_SUCCESS (0) in success. Please refer to "[2.4.1 Return value](#)" for the error code except it.

Example

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

2.4.13. setNVBitmap method

Syntax

- 1) int setNVBitmap (int number, String fileName, int width)
- 2) int setNVBitmap (int number, String fileName, int width, int mode)

Parameter

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

Value	[IN/OUT]	Meaning	Setting range
number	[IN]	Number of bitmap to store in the flash memory of the printer	1 - 20
fileName	[IN]	File name of bitmap to store	
width	[IN]	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).
mode	[IN]	Bitmap mode	CMP_BM_MODE_HT_THRESHOLD: Halftone threshold CMP_BM_MODE_HT_DITHER: Halftone dither CMP_BM_MODE_CMD_MONO Monochrome storing CMP_BM_MODE_CMD_GRAY16 Grayscale (4bpp) storing

Description

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

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

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

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

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

For more information on the mode parameter is as follows.

Halftone Specify the halftone treatment method.

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

Storing Specify the storing method.

Value	Description
-------	-------------

CMP_BM_MODE_CMD_MONO	Monochrome storing
CMP_BM_MODE_CMD_GRAY16	Grayscale storing Available in CT-D151, CT-E601/651, CT-S251/601II/ 651II/801II/851II/751. Graphic can be stored more beautifully.

[CT-S281, PMU3300, CMP-20/30 Series]

It is necessary that the bitmap numbers are contiguous from number 1. If you register a new bitmap after the connection, the bitmap that was previously registered will be erased.

The CMP-20/30 series, please register with a USB connection. The CMP-20 series is automatically disconnected because the printer is reset when the registration is completed.

[CT-D101/150/151, CT-E301/351/601/651, CT-S251/281II/310II/601/651/801/851/601II/651II/801II/851II/801III/851III/2000/4000 Series]

It is not necessary that the bitmap numbers are contiguous. And it is possible to remove a registered image by assigning the fileName parameter as an empty string.

Return value

Return CMP_SUCCESS (0) in success. Please refer to "[2.4.1 Return value](#)" for the error code except it.

Example

```
printer.setNVBitmap( 1, "samplebitmap.bmp",
    ESCPOSConst.CMP_BM_ASIS,
    ESCPOSConst.CMP_BM_MODE_HT_DITHER|ESCPOSConst.CMP_BM_MODE_CMD_MONO )
;
```

2.4.14. printNVBitmap method

Syntax

```
int printNVBitmap (int nvImageNumber)
```

Parameter

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

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

Description

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

[CT-S281, PMU3300, CMP-20/30 Series]

Please register the logo with "Unused key code mode".

To the image number to use, it is necessary to register the logo sequentially.

[CT-D101/150/151, CT-E301/351/601/651, CT-S251/281II/310II/601/651/801/851/601II/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 ["2.4.1 Return value"](#) for the error code except it.

Example

```
printer.printNVBitmap( 1 );
```

2.4.15. printBarcode method

Syntax

int printBarcode (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/801III/851III/751/4500 series.

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

Note: The data has restriction such as characters type, the number of digits and the addition of Code Set characters. For details, please refer to the printer command reference.

Return value

Return CMP_SUCCESS (0) in success. Please refer to "[2.4.1 Return value](#)" for the error code except it.

Example

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

2.4.16. printPDF417 method

Syntax

int printPDF417 (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 ["2.4.1 Return value"](#) for the error code except it.

Example

```
printer.printPDF417(
    "http://www.citizen-systems.co.jp/printer/tps/index.html",
    0, 0, 3, 3,
    ESCPOSConst.CMP_PDF417_EC_LEVEL_0,
    ESCPOSConst.CMP_ALIGNMENT_LEFT );
```

2.4.17. printQRCode method

Syntax

int printQRCode (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 "[2.4.1 Return value](#)" for the error code except it.

Example

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


2.4.18. printGS1DataBarStacked method

Syntax

int printGS1DataBarStacked (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/801III/851III/751/4500 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 "[2.4.1 Return value](#)" for the error code except it.

Example

```
printer.printGS1DataBarStacked(
    "0123456789012",
    ESCPOSConst.CMP_BCS_GS1DATABAR_S,
    4,
    300,
    ESCPOSConst.CMP_ALIGNMENT_LEFT );
```

2.4.19. cutPaper method

Syntax

int cutPaper (int type)

Parameter

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

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

Description

This method is used to cut the paper.

Return value

Return CMP_SUCCESS (0) in success. Please refer to "[2.4.1 Return value](#)" for the error code except it.

Example

```
printer.cutPaper( ESCPOSConst.CMP_CUT_PARTIAL_PREFEED );
```

2.4.20. unitFeed method

Syntax

```
int unitFeed (int ufCount)
```

Parameter

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

Value	[IN/OUT]	Meaning	Setting range
ufCount	[IN]	Number of paper feed	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 "[2.4.1 Return value](#)" for the error code except it.

Example

```
printer.unitFeed( 200 );
```

2.4.21. markFeed method

Syntax

int markFeed (int type)

Parameter

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

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

Description

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

Return value

Return CMP_SUCCESS (0) in success. Please refer to "[2.4.1 Return value](#)" for the error code except it.

Example

```
printer.markFeed( ESCPOSConst.CMP_MF_TO_CUTTER );
```

2.4.22. openDrawer method

Syntax

int openDrawer (int drawer, int pulseLen)

Parameter

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

Value	[IN/OUT]	Meaning	Setting range
drawer	[IN]	Cash drawer number	CMP_DRAWER_1: Drawer 1 CMP_DRAWER_2: Drawer 2
pulseLen	[IN]	Signal length	1 - 8 (x 100) msec

Description

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

Return value

Return CMP_SUCCESS (0) in success. Please refer to "[2.4.1 Return value](#)" for the error code except it.

Example

```
printer.openDrawer( ESCPOSConst.CMP_DRAWER_1, 1 );
```

2.4.23. transactionPrint method

Syntax

int transactionPrint (int control)

Parameter

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

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

Description

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

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

printText, printBitmap, printNVBitmap, printBarCode, printPDF417, printQRCode,
printGS1DataBarStacked, cutPaper, unitFeed, markFeed, openDrawer, rotatePrint, pageModePrint,
clearPrintArea, printData, printNormal

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

Calling the [clearOutput method](#) cancels transaction mode. Any buffered print lines are also cleared.

Return value

Return CMP_SUCCESS (0) in success. Please refer to ["2.4.1 Return value"](#) for the error code except it.

Example

```
printer.transactionPrint( ESCPOSConst.CMP_TP_TRANSACTION );
printer.printNVBitmap( 1 );
printer.printBarCode( "123456789012", ESCPOSConst.CMP_BCS_UPCA, 50, 2,
    ESCPOSConst.CMP_ALIGNMENT_LEFT, ESCPOSConst.CMP_HRI_TEXT_ABOVE );
printer.printText( "Line 1\n", ESCPOSConst.CMP_ALIGNMENT_LEFT,
    ESCPOSConst.CMP_FNT_DEFAULT, ESCPOSConst.CMP_TXT_1WIDTH );
printer.printText( "Line 2\n", ESCPOSConst.CMP_ALIGNMENT_LEFT,
    ESCPOSConst.CMP_FNT_DEFAULT, ESCPOSConst.CMP_TXT_1WIDTH );
printer.printText( "Line 3\n", ESCPOSConst.CMP_ALIGNMENT_LEFT,
    ESCPOSConst.CMP_FNT_DEFAULT, ESCPOSConst.CMP_TXT_1WIDTH );
printer.printBarCode( "123456789012", ESCPOSConst.CMP_BCS_UPCA, 50, 2,
    ESCPOSConst.CMP_ALIGNMENT_LEFT, ESCPOSConst.CMP_HRI_TEXT_ABOVE );
printer.printNVBitmap( 1 );
printer.cutPaper( ESCPOSConst.CMP_CUT_PARTIAL_PREFEED );
printer.transactionPrint( ESCPOSConst.CMP_TP_NORMAL );
```

2.4.24. rotatePrint method

Syntax

```
int rotatePrint (int rotation)
```

Parameter

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

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

Description

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

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

printText, printNormal

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

printBarcode, printPDF417, printQRCode, printGS1DataBarStacked and/or printBitmap

If rotation is CMP_RP_NORMAL, then rotation mode is exited.

Return value

Return CMP_SUCCESS (0) in success. Please refer to ["2.4.1 Return value"](#) for the error code except it.

Example

```
printer.rotatePrint( ESCPOSConst.CMP_RP_ROTATE180 |
    ESCPOSConst.CMP_RP_BARCODE | ESCPOSConst.CMP_RP_BITMAP );
printer.printBitmap( "samplebitmap.bmp", ESCPOSConst.CMP_BM_ASIS,
    ESCPOSConst.CMP_ALIGNMENT_CENTER );
printer.printBarcode( "123456789012", ESCPOSConst.CMP_BCS_UPCA, 50, 2,
    ESCPOSConst.CMP_ALIGNMENT_LEFT, ESCPOSConst.CMP_HRI_TEXT_ABOVE );
printer.printText( "Line 3\n", ESCPOSConst.CMP_ALIGNMENT_LEFT,
    ESCPOSConst.CMP_FNT_DEFAULT, ESCPOSConst.CMP_TXT_1WIDTH );
printer.printText( "Line 2\n", ESCPOSConst.CMP_ALIGNMENT_LEFT,
    ESCPOSConst.CMP_FNT_DEFAULT, ESCPOSConst.CMP_TXT_1WIDTH );
printer.printText( "Line 1\n", ESCPOSConst.CMP_ALIGNMENT_LEFT,
    ESCPOSConst.CMP_FNT_DEFAULT, ESCPOSConst.CMP_TXT_1WIDTH );
printer.cutPaper( ESCPOSConst.CMP_CUT_PARTIAL_PREFEED );
printer.rotatePrint( ESCPOSConst.CMP_RP_NORMAL );
```

2.4.25. pageModePrint method

Syntax

```
int pageModePrint (int control)
```

Parameter

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

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

Description

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

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

PrintText, PrintBitmap, PrintBarCode, printPDF417, printQRCode, printGS1DataBarStacked, PrintNormal

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

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

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

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

The entire Page Mode transaction is treated as one message.

Calling the [clearOutput method](#) cancels Page Mode. Any buffered print lines are also cleared.

Return value

Return CMP_SUCCESS (0) in success. Please refer to ["2.4.1 Return value"](#) for the error code except it.

Example

```
// Standard print
posPtr.printNormal( "\u001b|2vCSample 2 - Print\n");
posPtr.printText("12345678901234567890123456789012345678901234567890123456789012345678901234567890\n",
    ESCPOSConst.CMP_ALIGNMENT_RIGHT, ESCPOSConst.CMP_FNT_DEFAULT,
    ESCPOSConst.CMP_TXT_1WIDTH | ESCPOSConst.CMP_TXT_1HEIGHT);

// Start of Page Mode
printer.pageModePrint( 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" );
printer.printNormal( "\u001b|4C- Receipt -\n" );
// Set print area of Page Mode
printer.setPageModePrintArea( "184,0,120,800" );
printer.printText( "    $ 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" );
printer.printText( "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" );
printer.printBarcode( "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" );
printer.printQRCode( "http://www.citizen-systems.co.jp/", 5,
    ESCPOSConst.CMP_QRCODE_EC_LEVEL_L,
    ESCPOSConst.CMP_ALIGNMENT_LEFT );
// End of Page Mode
printer.pageModePrint( ESCPOSConst.CMP_PM_NORMAL );
```

Print**image**

2.4.26. 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 "[2.4.1 Return value](#)" for the error code except it.

Example

```
printer.clearPrintArea();
```

2.4.27. clearOutput method

Syntax

int clearOutput ()

Parameter

Not exist.

Description

This method is used to clear all buffered output data by [transactionPrint method](#) and [pageModePrint method](#).

Also, when possible, halts outputs that are in progress. At the same time, the command to clear print data on the printer is sent.

Return value

Return CMP_SUCCESS (0) in success. Please refer to "[2.4.1 Return value](#)" for the error code except it.

Example

```
printer.clearOutput();
```

2.4.28. printData method

Syntax

```
int printData (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 "[2.4.1 Return value](#)" for the error code except it.

Example

```
// Sound the buzzer (The printer must support buzzer.)  
byte[] data = { 0x1b, 0x1e };  
printer.printData( data );
```

2.4.29. printNormal method

Syntax

int printNormal (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 #IF	
Unit feed	ESC #uF	
Barcode print	ESC #R	
Font type specification	ESC #fT	
Bold	ESC bC	
Underline	ESC #uC	
Custom color	ESC #rC	Effective only when dedicated 2-color paper is used.
Red	ESC rC	Effective only when dedicated 2-color paper is used.
Reverse character	ESC rvC	
Standard	ESC 1C	
Double width	ESC 2C	
Double height	ESC 3C	
Quadruple	ESC 4C	
Horizontal magnification	ESC #hC	1-8
Vertical magnification	ESC #vC	1-8
Centering	ESC cA	
Right adjustment	ESC rA	
Normal	ESC N	

Return value

Return CMP_SUCCESS (0) in success. Please refer to ["2.4.1 Return value"](#) for the error code except it.

Example

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

2.4.30. watermarkPrint method

Syntax

```
int watermarkPrint (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 using [setNVBitmap method](#) or use the "POS Printer utility" of utility software for the printer.

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

Return value

Return CMP_SUCCESS (0) in success. Please refer to ["2.4.1 Return value"](#) for the error code except it.

Example

```
printer.watermarkPrint( ESCPOSConst.CMP_WM_START, 1, 0, 0, 0 );
```

2.4.31. searchCitizenPrinter method

Syntax

CitizenPrinterInfo[] seachCitizenPrinter (int connectType, int serchTime, int[] result)

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 CMP_PORT_Bluetooth_Insecure
serchTime	[IN]	Search time (sec)	0: Get paired address in the case of CMP_PORT_Bluetooth or CMP_PORT_Bluetooth_Insecure. 1 - 30: Search for a specified time.
result	[OUT]	Error code	

Return CMP_SUCCESS (0) to result in success. Please check the description of the error codes below in the case of failure. Please refer to "[2.4.1 Return value](#)" for the error code except it.

Error codes	Description
CMP_E_NOCONTEXT (1006)	The context is not specified.
CMP_E_ILLEGAL (1101)	Invalid parameter. (1) The connect type is unsupported. (2) The search time is out of range.
CMP_E_FAILURE (1104)	Error occurs during the search, the search failed.
CMP_E_NO_LIST (1106)	As a result of search, the printer cannot be found.

Description

This method is used to search the printer. Please specify the type of the printer connection and the search time. Before the execution of this method, must execute the [setContext method](#). This method cannot be used on the simulator.

After search time passed, set a result to the result parameter and return the information of the found printers as array type.

In the case of CMP_PORT_WiFi for the connection type, you can search only the printers of CT-D101/150/151, CT-E301/351/601/651, CT-S251/310II/601/651/801/851/601II/651II/801II/851II/801III/851III/751/4500 series. Recommended value of search time is more than 3 seconds. When the search time is shorter than the second, a search may fail by the network situation.

In the case of CMP_PORT_Bluetooth or CMP_PORT_Bluetooth_Insecure for the connection type, you can get the paired address when specifying 0 for the search time. When specifying 1 - 30 for the search time you can get the connectable address. Recommended value of search time is more than 10 seconds. When the search time is shorter than the second, a search may fail by the Bluetooth situation.

Return value

Return the list of addresses of the printers when a search succeeded. When a search fails, return the empty list.

The list of information of the printer is stored as a CitizenPrinterInfo-type, and available information varies according to connectType parameter.

connectType	CitizenPrinterInfo	Information to be obtained
CMP_PORT_WiFi	ipAddress	IP Address
	macAddress	MAC Address
	deviceName	(Empty character)
	bdAddress	(Empty character)
CMP_PORT_BLUETOOTH	ipAddress	(Empty character)
	macAddress	(Empty character)
	deviceName	Bluetooth device name
	bdAddress	Bluetooth device address

Example

```

int[] errCode = new int[1];
CitizenPrinterInfo[] list = printer.searchCitizenPrinter(
    ESCPOSConst.CMP_PORT_WiFi, 3, errCode );
for (int i = 0; i < list.length; i++)
{
    Log.d("Search", "IP Address: " + list[i].ipAddress);
    Log.d("Search", "MAC Address: " + list[i].macAddress);
}

```


2.4.32. searchESCPOSPrinter method

Syntax

String[] seachESCPOSPrinter (int connectType, int serchTime, int[] result)

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 CMP_PORT_Bluetooth_Insecure
serchTime	[IN]	Search time (sec)	0: Get paired address in the case of CMP_PORT_Bluetooth or CMP_PORT_Bluetooth_Insecure. 1 - 30: Search for a specified time.
result	[OUT]	Error code	

Return CMP_SUCCESS (0) to result in success. Please check the description of the error codes below in the case of failure. Please refer to "[2.4.1 Return value](#)" for the error code except it.

Error codes	Description
CMP_E_NOCONTEXT (1006)	The context is not specified.
CMP_E_ILLEGAL (1101)	Invalid parameter. (1) The connect type is unsupported. (2) The search time is out of range.
CMP_E_FAILURE (1104)	Error occurs during the search, the search failed.
CMP_E_NO_LIST (1106)	As a result of search, the printer cannot be found.

Description

This method is used to search the printer. Please specify the type of the printer connection and the search time. Before the execution of this method, must execute the [setContext method](#). This method cannot be used on the simulator.

After search time passed, set a result to the result parameter and return the information of the found printers as String array type.

In the case of CMP_PORT_WiFi for the connection type, you can search only the printers of CT-D101/150/151, CT-E301/351/601/651, CT-S251/310II/601/651/801/851/601II/651II/801II/851II/801III/851III/751/4500 series. Recommended value of search time is more than 3 seconds. When the search time is shorter than the second, a search may fail by the network situation.

In the case of CMP_PORT_Bluetooth or CMP_PORT_Bluetooth_Insecure for the connection type, you can get the paired address when specifying 0 for the search time. When specifying 1 – 30 for the search time you can get the connectable address. Recommended value of search time is more than 10 seconds. When the search time is shorter than the second, a search may fail by the Bluetooth situation.

Return value

Return the list of addresses of the printers when a search succeeded. When a search fails, return the empty list.

Example

```
int[] errCode = new int[1];
String[] list = printer.searchESCPOSPrinter(
    ESCPOSConst.CMP_PORT_WiFi, 3, errCode );
```

2.4.33. printerCheckEx method

Syntax

- 1) int printerCheckEx (int[] status, int connectType, string addr)
- 2) int printerCheckEx (int[] status, int connectType, string addr, int port)
- 3) int printerCheckEx (int[] status, int connectType, string addr, int port, int timeout)
- 4) int printerCheckEx (int[] status, int connectType, UsbDevice device)

Parameter

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

Value	[IN/OUT]	Meaning	Setting range
status	[OUT]	Status codes	
connectType	[IN]	Connect type	CMP_PORT_WiFi CMP_PORT_Bluetooth CMP_PORT_Bluetooth_Insecure CMP_PORT_USB CMP_PORT_SNMP
addr	[IN]	IP address to connect or Bluetooth device address or Bluetooth device name	WiFi: 0.0.0.0~255.255.255.255 Bluetooth: 00:00:00:00:00:00~FF:FF:FF:FF:FF:FF Device name (Automatic detection)
device	[IN]	UsbDevice class (android.hardware.usb.UsbDevice)	null: Automatic detection UsbDevice: Any USB device
port	[IN]	Connection port number	
timeout	[IN]	Timeout (msec)	

* The acquisition of CMP_STS_BATTERY_LOW and CMP_STS_MSR_READ for CMP-20/30 does not support.

Description

This method is used to connect printer and get the status of the printer. After the process is complete, disconnect the connection. (except connection type CMP_PORT_SNMP)

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

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

Status codes	Description
CMP_STS_NORMAL (0)	The printer is normal.
CMP_STS_PRINTEROFF (128)	The printer is off-line.
CMP_STS_PAPER_EMPTY (32)	The printer is out of paper.
CMP_STS_COVER_OPEN (16)	The cover of the printer opens.
CMP_STS_PAPER_NEAREMPTY (4)	Paper near empty.
CMP_STS_DRAWER_LEVEL_H (2)	Status of pin 3 of drawer kick-out connector = H.
CMP_STS_PAPER_ONPRESENTER (1)	Status the paper is hold on the presenter or the paper exit sensor. (PMU2300III/PMU3300 only, Not supported by SNMP connect)

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 "[2.4.1 Return value](#)" for the error code except it.

Error codes	Description
CMP_E_NOTCONNECT (1003)	Failed connection to the printer. (1) The printer is under none-connection status. (2) The printer is not turned ON. (3) Cannot obtain handle of interface board. (4) The printer is in use by another connection. (Except SNMP)
CMP_E_CONNECT_NOTFOUND (1004)	Failed to check the support model after connecting to the printer. (1) The model is not supported.

Example

```

int status = new int[1];
int result = printer.printerCheckEx(status, ESCPOSConst.CMP_PORT_WiFi,
                                     "192.168.182.100");
if ( ESCPOSConst.CMP_SUCCESS == result ) {
    if ( ESCPOSConst.CMP_STS_NORMAL == status[0] ) {
        // Status Normal
    } else {
        if ( (ESCPOSConst.CMP_STS_COVER_OPEN & status[0]) > 0 ) {
            // Cover Open
        }
        if ( (ESCPOSConst.CMP_STS_PAPER_EMPTY & status[0]) > 0 ) {
            // Paper Empty
        }
        if ( (ESCPOSConst.CMP_STS_PRINTEROFF & status[0]) > 0 ) {
            // Printer Offline
        }
        if ( (ESCPOSConst.CMP_STS_PAPER_NEAREMPTY & status[0]) > 0 ) {
            // Paper Near Empty
        }
        if ( (ESCPOSConst.CMP_STS_DRAWER_LEVEL_H & status[0]) > 0 ) {
            // Pin 3 of drawer kick-out connector = H
        }
        if ( (CMP_STS_PAPER_ONPRESENTER & status[0]) > 0 ) {
            // Paper is hold on presenter or paper exit sensor
        }
    }
} else {
    // PrinterCheckEx Error
}

```

2.4.34. openDrawerEx method

Syntax

- 1) int openDrawerEx (int drawer, int pulseLen, int connectType, string addr)
- 2) int openDrawerEx (int drawer, int pulseLen, int connectType, string addr, int port)
- 3) int openDrawerEx (int drawer, int pulseLen, int connectType, string addr, int port, int timeout)
- 4) int openDrawerEx (int drawer, int pulseLen, int connectType, UsbDevice device)

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
connectType	[IN]	Connect type	CMP_PORT_WiFi CMP_PORT_Bluetooth CMP_PORT_Bluetooth_Insecure CMP_PORT_USB
addr	[IN]	IP address to connect or Bluetooth device address or Bluetooth device name	WiFi: 0.0.0.0~255.255.255.255 Bluetooth: 00:00:00:00:00:00~FF:FF:FF:FF:FF:FF Device name (Automatic detection)
device	[IN]	UsbDevice class (android.hardware.usb.UsbDevice)	null: Automatic detection UsbDevice: Any USB device
port	[IN]	Connection port number	
timeout	[IN]	Timeout (msec)	

Description

This method is used to connect printer and open the cash drawer is connected to the printer. After the process is complete, disconnect the connection.

This method can execute even if the printer error (cover open or paper empty).

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 "[2.4.1 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.

Example

```
printer.openDrawerEx(ESCPOSConst.CMP_DRAWER_1, 1,
    ESCPOSConst.CMP_PORT_WiFi, "192.168.182.100");
```

2.4.35. setPrintCompletedTimeout method

Syntax

```
int setPrintCompletedTimeout(int timeout)
```

Parameter

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

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

Description

This method is used to set the timeout to check the print completion notification.

When you create an instance, the timeout is initialized to 0.

Please refer to "[2.5.1. 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 "[2.4.1 Return value](#)" for the error code except it.

Example

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

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

2.4.36. 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]	Log Output Folder	Folder path
maxSize	[IN]	Maximum Log Size	0: Unlimited 1 - : Maximum size (MB)

Description

Sets the logging function. See ["3.2 Logging function"](#) for more details.

For the output folder, specify the path to a folder that the application that outputs the log file can write to. If a full path other than the one beginning with '/storage/' is specified, it will be specified as the path from the top of the external storage path.

Return value

none

Example

```
// Specifying the folder for the application's specific area.
printer.setLog(1, this.getExternalFilesDir(null).toString() + "/Log", 10);

// Specifying the public folder (Download).
printer.setLog(1, Environment.getExternalStoragePublicDirectory(
    Environment.DIRECTORY_DOWNLOADS) + "/Log", 10);
```

2.4.37. 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 = printer.getVersionCode();
```


2.4.38. 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 = printer.getVersionName();
```

2.4.39. 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-S281/281II Series] : "384,938"

[CT-D101/150/151, CT-E301/351/601/651, CT-S310II/601/651/801/851/601II/651II/801II/851II/801III/851III/751/2000, PMU3300 Series] :

"576,1662"

[CT-S4000/4500 Series] : "832,1662"

[CMP-20 Series] : "384,938"

[CMP-30 Series] : "576,938"

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

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

Set property

Not exist.

Get property

String getPageModeArea()

Returns the page area as the return value.

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

Set property

int setPageModePrintArea (String area)

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

Return CMP_SUCCESS (0) in success. Please refer to "[2.4.1 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.

2.4.41. PageModePrintDirection property

Syntax

int PageModePrintDirection

Attribute

Read/Write

Description

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

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

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

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

Set property

int setPageModePrintDirection (int direction)

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

Return CMP_SUCCESS (0) in success. Please refer to "[2.4.1 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.

2.4.42. PageModeHorizontalPosition property

Syntax

int PageModeHorizontalPosition

Attribute

Read/Write

Description

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

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

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

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

Set property

int setPageModeHorizontalPosition (int position)

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

Return CMP_SUCCESS (0) in success. Please refer to "[2.4.1 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.

2.4.43. PageModeVerticalPosition property

Syntax

int PageModeVerticalPosition

Attribute

Read/Write

Description

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

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

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

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

Set property

int setPageModeVerticalPosition (int position)

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

Return CMP_SUCCESS (0) in success. Please refer to "[2.4.1 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.

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

Set property

int setRecLineSpacing (int spacing)

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

Return CMP_SUCCESS (0) in success. Please refer to "[2.4.1 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.

2.4.45. RecCharacterSpacing property

Syntax

int RecCharacterSpacing

Attribute

Read/Write

Description

This property holds the right spacing of Character. Expressed in the unit of measure given by [MapMode](#) (default dots). Possible values are 0 to 255 in dot units. Does not affect Kanji.

If the horizontal magnification of Character is 2 or more, the spacing increases with the magnification.

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

Set property

int setRecCharacterSpacing (int spacing)

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

Return CMP_SUCCESS (0) in success. Please refer to "[2.4.1 Return value](#)" for the error code except it.

Get property

int getRecCharacterSpacing ()

Returns the right spacing of Character that is set as the return value.

2.4.46. RecKanjiLeftSpacing property

Syntax

int RecKanjiLeftSpacing

Attribute

Read/Write

Description

This property holds the left spacing of Kanji. Expressed in the unit of measure given by [MapMode](#) (default dots). Possible values are 0 to 255 in dot units. Does not affect Character.

If the horizontal magnification of character is 2 or more, the spacing increases with the magnification.

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

Set property

int setRecKanjiLeftSpacing (int spacing)

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

Return CMP_SUCCESS (0) in success. Please refer to "[2.4.1 Return value](#)" for the error code except it.

Get property

int getRecKanjiLeftSpacing ()

Returns the left spacing of Kanji that is set as the return value.

2.4.47. RecKanjiRightSpacing property

Syntax

int RecKanjiRightSpacing

Attribute

Read/Write

Description

This property holds the right spacing of Kanji. Expressed in the unit of measure given by [MapMode](#) (default dots). Possible values are 0 to 255 in dot units. Does not affect Character.

If the horizontal magnification of character is 2 or more, the spacing increases with the magnification.

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

Set property

int setRecKanjiRightSpacing (int spacing)

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

Return CMP_SUCCESS (0) in success. Please refer to "[2.4.1 Return value](#)" for the error code except it.

Get property

int getRecKanjiRightSpacing ()

Returns the right spacing of Kanji that is set as the return value.

2.4.48. 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.

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

The connect method must be complete before accessing this property. This property is initialized to CMP_MM_DOTS at connect 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 "[2.4.1 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 );  
printer.unitFeed( 200 ); // 200 dots feed  
  
printer.setMapMode( ESCPOSConst.CMP_MM_METRIC );  
printer.unitFeed( 2500 ); // 25 millimeter feed
```

2.4.49. ErrorCodeExtended property

Syntax

int ErrorCodeExtended

Attribute

Read

Description

This property holds the extended error code when there is an error connecting to the printer. The values are as follows.

Value	Meaning
CMP_EX_DEV_NO_PRINTER (62000)	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_EX_DEV_OPEN_ERROR (62100)	Unable to connect socket or port. (1) Communication error. (2) Wrong address specification when connecting to WiFi. (3) USB access not allowed.
CMP_EX_DEV_SEND_ERROR (62101)	Failed to send command to printer. (1) Communication error.
CMP_EX_DEV_NO_RESPONSE (62102)	No command response from the printer. (1) The printer is in use by another application.

This property is initialized to 0 at the beginning of the connection process. Refer to this after executing the [connect method](#), [printerCheckEx method](#), and [openDrawerEx method](#).

Set property

Not exist.

Get property

int getErrorCodeExtended ()

Returns the extended error code as the return value.

2.5. Notes

Notes of this SDK are as follows.

2.5.1. Function to detect the completion of printing

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

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

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

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

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

2.5.2. About printing UTF-8 encode characters

This SDK supports printing UTF-8 encoded characters.

This feature is focusing on providing a way to interoperate East Asian legacy double-byte character sets for Japanese, Korean, Simplified and Traditional Chinese.

Example

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

Supported models

Model	Firmware Version	Conditions
CT-S251	EM01-0304 or newer	*1
CT-S310II	DT00-1000 or newer DT10-1100 or newer	
CT-S601II	EE00-0200 or newer	*2
CT-S651II	EA00-0200 or newer	
CT-S801II	ED00-0200 or newer	
CT-S851II	DY00-0200 or newer	
CT-D101/150/151	All versions	*3
CT-E301/351/601/651		
CT-S751/801III/851III/4500		

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, CT-S801III/851III 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)

2.5.3. Logging function

This SDK has a logging function which keeps history of executing methods or reading/writing properties. This can be enabled by using the [setLog method](#), or placing a file "CSJPOSLib.cfg" in the /temp/citizen at the top of the external storage path.

On Android 11 or later, to access the above path, you need to grant the application "Access to all files (MANAGE_EXTERNAL_STORAGE) permission" and allow the permission in the device's app settings.

Please refer to the attached sample "AndroidManifest.xml" for how to define the application.

```
<uses-permission android:name="android.permission.MANAGE_EXTERNAL_STORAGE" />
```

< Example of CSJPOSLib.cfg >

[LogSetting]	...Section name (Fixed)
LogMode=1	...Specifies the log mode.
LogPath=/storage/emulated/0/Download/Log	...Specifies the folder to output the log files.
LogMaxSize=10	...Specifies the maximum size of log file in MB.

The external storage path depends on the device. For many terminals, the path to place the file will be one of the following.

```
/storage/sdcard0/temp/citizen/Log/
or
/storage/emulated/0/temp/citizen/Log/
```

Setting items

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

2: Error log

- LogPath

Specifies a folder to store the log files.

If a full path starting with /storage/ is specified for the storage folder, the files will be stored in the specified path. Otherwise, it is specified as the path from the top of the external storage path. "/temp/citizen" of the external storage path by default.

- LogMaxSize

Specifies maximum size to log file in MB. "0" sets the size to "unlimited."

Log file name

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

Example: CSJPOSLib_1.log

When the same file name exists, the file will be overwritten if the file is one week older, new logs will be added to the existing file if the file is created on the same day.

Log format

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

```

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

2016/03/08 12:48:45.122 4268 001 METHOD call   connect(0, "192.168.10.100")
2016/03/08 12:48:46.151 4268 001 METHOD result connect() -> Success(0)

--- Example 2, method (PrintText) ---

2016/03/08 12:51:35.803 4268 001 METHOD call   printText([See below], 1, 1, 0)
-----Parameter Detail-----
Print text 1
Print text 2
-----
2016/03/08 12:51:36.099 4268 001 METHOD result printText() -> Success(0)

--- Example 3, set to a property ---

2016/03/08 13:00:23.021 4268 001 PROPERTY set   RecLineSpacing <- 24 : Success(0)

--- Example 4, get from a property ---

2016/03/08 13:00:23.037 4268 001 PROPERTY get   RecLineSpacing -> 24

```

* The logging function could be a "bottleneck" of processing since it tries to keep the information of every single execution of a method or access to a property.

* Logging could fail without a notification for the reason below.

- A write-protected device is selected as a storage location.
- No enough space in the selected storage device.

- The storage location contains a file with write-protection.
- No access privilege to a file or folder.
- Another program is using the log file.

2.5.4. Predefined Constants List

No	Type	Name	Data type	Value	Description
1	Result/Error	CMP_SUCCESS	int	0	Successfully completed
		CMP_E_CONNECTED	int	1001	Already connected
		CMP_E_DISCONNECT	int	1002	Not connected
		CMP_E_NOTCONNECT	int	1003	Failed to connect
		CMP_E_CONNECT_NOTFOUND	int	1004	Non supported model
		CMP_E_CONNECT_OFFLINE	int	1005	Failed printer status
		CMP_E_NOCONTEXT	int	1006	No context information specified
		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	Connection interface	CMP_PORT_WiFi	int	0	Network
		CMP_PORT_Bluetooth	int	1	Bluetooth
		CMP_PORT_Bluetooth_Insecure	int	2	Bluetooth Insecure
		CMP_PORT_USB	int	3	USB
3	Status	CMP_STS_NORMAL	int	0	Normal
		CMP_STS_PAPER_ONPRESENTER	int	1	Status the paper is hold on the presenter or the paper exit sensor.
		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_BATTERY_LOW	int	8	battery capacity low
		CMP_STS_COVER_OPEN	int	16	Cover opens
		CMP_STS_PAPER_EMPTY	int	32	Paper empty
		CMP_STS_MSR_READ	int	64	Currently MSR read mode
		CMP_STS_PRINTEROFF	int	128	Off-line
4	Alignment	CMP_ALIGNMENT_LEFT	int	0	Left alignment
		CMP_ALIGNMENT_CENTER	Int	1	Center alignment
		CMP_ALIGNMENT_RIGHT	int	2	Right alignment
5	Text attribute	CMP_FNT_DEFAULT	int	0	Default font
		CMP_FNT_FONTB	int	1	Font B
		CMP_FNT_FONTC	int	2	Font C
		CMP_FNT_BOLD	int	8	Bold
		CMP_FNT_REVERSE	int	16	Reverse
		CMP_FNT_UNDERLINE	int	128	Underline
		CMP_FNT_ITALIC	int	256	Italic
		CMP_FNT_STRIKEOUT	int	512	Strikeout
6	Text size	CMP_TXT_1WIDTH	int	0	1 times width
		CMP_TXT_2WIDTH	int	16	2 times width
		CMP_TXT_3WIDTH	int	32	3 times width
		CMP_TXT_4WIDTH	int	48	4 times width
		CMP_TXT_5WIDTH	int	64	5 times width

		CMP_TXT_6WIDTH	int	80	6 times width
		CMP_TXT_7WIDTH	int	96	7 times width
		CMP_TXT_8WIDTH	int	112	8 times width
		CMP_TXT_1HEIGHT	int	0	1 times height
		CMP_TXT_2HEIGHT	int	1	2 times height
		CMP_TXT_3HEIGHT	int	2	3 times height
		CMP_TXT_4HEIGHT	int	3	4 times height
		CMP_TXT_5HEIGHT	int	4	5 times height
		CMP_TXT_6HEIGHT	int	5	6 times height
		CMP_TXT_7HEIGHT	int	6	7 times height
		CMP_TXT_8HEIGHT	int	7	8 times height
7	Side	CMP_SIDE_RIGHT	int	0	Right side
		CMP_SIDE_LEFT	int	1	Left side
8	Bitmap width	CMP_BM_ASIS	int	-11	One bitmap pixel per printer dot
9	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)
10	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
11	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
12	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
13	Error correction	CMP_PDF417_EC_LEVEL_8	int	56	Level 8
		CMP_QRCODE_EC_LEVEL_L	int	48	Level L (7%)

	level (QR Code)	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%)
14	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
15	Mark feed type	CMP_MF_TO_CUTTER	int	2	Feed and cut
		CMP_MF_TO_NEXT_TOF	int	8	Feed to the next top
16	Drawer number	CMP_DRAWER_1	int	1	Drawer 1
		CMP_DRAWER_2	int	2	Drawer 2
17	Transaction control	CMP_TP_TRANSACTION	int	11	Begin transaction
		CMP_TP_NORMAL	int	12	End transaction
18	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
19	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
20	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
21	Watermark control	CMP_WM_STOP	int	0	End watermark
		CMP_WM_START	int	1	Begin watermark
22	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
23	Extended error code	CMP_EX_DEV_NO_PRINTER	int	62000	Printer not connected
		CMP_EX_DEV_OPEN_ERROR	int	62100	Failed to connect socket or port
		CMP_EX_DEV_SEND_ERROR	int	62100	Failed to send command
		CMP_EX_DEV_NO_RESPONSE	int	62102	Command no response

3. Line Display Control

3.1. Definition method

Add the library file that is provided, if you use this SDK, please import the following two classes.

```
import com.citizen.sdk.LineDisplayConst;
import com.citizen.sdk.LineDisplay;
```

3.2. Program structure

Here is an example program which uses the SDK

<pre>// Create an instance. LineDisplay display = new LineDisplay(); // Connect Linedisplay int result = display.connect(LineDisplayConst.CDP_PORT_Bluetooth, ""); if (LineDisplayConst.CDP_SUCCESS == result) { // Set encoding display.setEncoding("Shift_JIS"); // Clear text display.clearDisplay(); // Display text display.displayText("123456"); // Set cursor position display.setCursorPosition(1,2); // Display text (Reverse) display.displayText("123456",true); // Disconnect display.disconnect(); } else { // Connect Error Log.e("Citizen_POS_sample1", "Connect or LineDisplay Error : " + Integer.toString(result)); }</pre>	<div style="display: flex; align-items: center;"> <div style="font-size: 3em; margin-right: 10px;">}</div> <div>Class definition</div> </div> <div style="display: flex; align-items: center; margin-top: 20px;"> <div style="font-size: 3em; margin-right: 10px;">}</div> <div>Connect</div> </div> <div style="display: flex; align-items: center; margin-top: 20px;"> <div style="font-size: 3em; margin-right: 10px;">}</div> <div>Linedisplay processes</div> </div> <div style="display: flex; align-items: center; margin-top: 20px;"> <div style="font-size: 3em; margin-right: 10px;">}</div> <div>Disconnect</div> </div>
----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

3.3. Functions list

This SDK provides the following functions.

Methods list

No	Function	Detail
1	Create class (Constructor)	This is constructor method.
2	Connect display (connect method)	This method connects to the line display
3	Disconnect display (disconnect method)	This method disconnects the line display connection.
4	Display the text (displayText method)	This method is used to display text.
5	Clear the displayed text (clearDisplay method)	This method clears the displayed text.
6	Blink the display (blinkDisplay method)	This method causes the entire display screen to blink.
7	Set display mode (setDisplayMode method)	This method sets the following display modes.
8	Set display config (setDisplayConfig method)	This method changes the brightness of the display screen.
9	Set cursor Position (setCursorPosition method)	This method is used to set the cursor position.
10	Move cursor (moveCursor method)	This method is used to move the cursor.
11	Show cursor position (setCursorPosition method)	This displays the current cursor position on the display.
12	Initialize (initializeDisplay method)	This method initializes the device.
13	Send command (displayData method)	This method sends the command.
14	Set encoding (setEncoding method)	This method sets the encoding of character.
15	Set code page (setCodePage method)	This method sets the code page of character.
16	Set international character set (setInternationalCharacterSet method)	This sets the following international character sets.
17	Check display status (checkDisplay method)	This method is used to check the display connection status.
18	Get version code (getVersionCode method)	This method gets a numerical value for the version number of this SDK.
19	Get version name (getVersionName method)	This method gets a string for the version number of this SDK.
20	Set log (setLog method)	Sets the log function.

Properties List

No	Function	Attribute	Detail
1	Extended error code (ErrorCodeExtended property)	R	Show the extended error code when connecting.

3.4. Library interfaces

The following are the interfaces of this SDK.

3.4.1. 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_NOCONTEXT (1006)	The context is not specified.
CDP_E_BT_DISABLE (1007)	The setting of the Bluetooth device is invalid.
CDP_E_BT_NODEVICE (1008)	The Bluetooth device is not found.
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.

3.4.2. 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();
```


3.4.3. connect method

Syntax

- 1) int connect (int connectType, String addr)
- 2) int connect (int connectType, String addr, int port)
- 3) int connect (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 CDP_PORT_Bluetooth_Insecure
addr	[IN]	IP address to connect or Bluetooth device address or Bluetooth device name	WiFi: 0.0.0.0~255.255.255.255 Bluetooth: 00:00:00:00:00:00~FF:FF:FF:FF:FF:FF Device name (Automatic detection)
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.

Bluetooth device address letters, please specify in uppercase.

If the Bluetooth Device name is specified, the device that was paired is detected automatically. If you omit the Device name, the supported model device that was paired is detected automatically.

If you want to use the Bluetooth device insecure communications provided by the Android 2.3.3 or later, please specify the connection type CDP_PORT_Bluetooth_Insecure.

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

When communication with the line display is not necessary, must execute the [disconnect 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 "[3.4.1 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.
CDP_E_BT_DISABLE (1007)	The setting of the Bluetooth device is invalid.
CDP_E_BT_NODEVICE (1008)	The Bluetooth device is not found.

Example

```
display.connect(LineDisplayConst.CDP_PORT_WiFi, "192.168.0.10");
```

3.4.4. disconnect method

Syntax

```
int disconnect ()
```

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 error occurs, please disconnect the connection by the execution of this method.

Return value

Return CDP_SUCCESS(0) in success. Please refer to "[3.4.1 Return value](#)" for the error code except it.

Example

```
display.disconnect();
```

3.4.5. displayText method

Syntax

- 1) int displayText (String data)
- 2) int displayText (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 "[3.4.1 Return value](#)" for the error code except it.

Example

```
display.displayText("Hello, World!");
```

3.4.6. clearDisplay method

Syntax

int clearDisplay (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 "[3.4.1 Return value](#)" for the error code except it.

Example

```
display.clearDisplay(LineDisplayConst.CDP_AREA_ALL);
```

3.4.7. blinkDisplay method

Syntax

int blinkDisplay (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 "[3.4.1 Return value](#)" for the error code except it.

Example

```
display.blinkDisplay(1000);
```

3.4.8. setDisplayMode method

Syntax

```
int setDisplayMode (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 "[3.4.1 Return value](#)" for the error code except it.

Example

```
display.setDisplayMode (LineDisplayConst.CDP_VERTICALSCROLL) ;
```

3.4.9. setDisplayConfig method

Syntax

```
int setDisplayConfig (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 "[3.4.1 Return value](#)" for the error code except it.

Example

```
display.setDisplayConfig(40);
```

3.4.10. setCursorPosition method

Syntax

```
int setCursorPosition(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 "[3.4.1 Return value](#)" for the error code except it.

Example

```
display.setCursorPosition(1, 2);
```


3.4.11. moveCursor method

Syntax

```
int moveCursor (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 "[3.4.1 Return value](#)" for the error code except it.

Example

```
display.moveCursor(2, 0);
```

3.4.12. setCursorType method

Syntax

```
int setCursorType (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 "[3.4.1 Return value](#)" for the error code except it.

Example

```
display.setCursorType (LineDisplayConst.CDP_TYPE_UNDERLINE);
```

3.4.13. initializeDisplay method

Syntax

```
int initializeDisplay ()
```

Parameter

None

Description

Initializes the device.

Return value

Return CDP_SUCCESS (0) in success. Please refer to "[3.4.1 Return value](#)" for the error code except it.

Example

```
display.initializeDisplay();
```

3.4.14. displayData method

Syntax

```
int displayData (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 "[3.4.1 Return value](#)" for the error code except it.

Example

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

3.4.15. 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 CDP_SUCCESS (0) in success. Please refer to "[3.4.1 Return value](#)" for the error code except it.

Example

```
display.setEncoding( "Shift_JIS" );
```

```
display.setEncoding( "GB18030" );
```

```
display.setEncoding( "EUC-KR" );
```

```
display.setEncoding( "Big5" );
```

3.4.16. setCodePage method

Syntax

```
int setCodePage (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 "[3.4.1 Return value](#)" for the error code except it.

Example

```
display.setCodePage(1);
```

3.4.17. setInternationalCharacterSet method

Syntax

```
int setInternationalCharacterSet (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 "[3.4.1 Return value](#)" for the error code except it.

Example

```
display.setInternationalCharacterSet(8);
display.displayText("Total:¥¥1,010");
```

3.4.18. displayCheck method

Syntax

```
int displayCheck ()
```

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 [disconnect method](#) and [connect 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 "[3.4.1 Return value](#)" for the error code except it.

Example

```
display.displayCheck() ;
```


3.4.19. 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

```
display.getVersionCode ();
```

3.4.20. 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

```
display.getVersionName();
```

3.4.21. 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]	Log Output Folder	Folder path
maxSize	[IN]	Maximum Log Size	0: Unlimited 1 - : Maximum size (MB)

Description

Sets the logging function. See ["3.5.1 Logging function"](#) for more details.

For the output folder, specify the path to a folder that the application that outputs the log file can write to. If a full path other than the one beginning with '/storage/' is specified, it will be specified as the path from the top of the external storage path.

Return value

none

Example

```
// Specifying the folder for the application's specific area.
display.setLog(1, this.getExternalFilesDir(null).toString() + "/Log", 10);

// Specifying the public folder (Download).
display.setLog(1, Environment.getExternalStoragePublicDirectory(
    Environment.DIRECTORY_DOWNLOADS) + "/Log", 10);
```

3.4.22. ErrorCodeExtended property

Syntax

```
int ErrorCodeExtended
```

Attribute

Read

Description

This property holds the extended error code when there is an error connecting to the display. The values are as follows.

Value	Meaning
CDP_EX_DEV_NO_PRINTER (62000)	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.
CDP_EX_DEV_NO_DEVICE (62001)	Display is not connected. (1) The display is not connected. (2) The setting of the interface port is incorrect.
CDP_EX_DEV_USING (62002)	Display is in use. (1) The display is in use by another application.
CDP_EX_DEV_CONFIRM (62003)	Failed to check the status after connecting to the display. (1) Wrong specified port number.
CDP_EX_DEV_OPEN_ERROR (62100)	Unable to connect socket. (1) Communication error.
CDP_EX_DEV_SEND_ERROR (62101)	Failed to send command to display. (1) Communication error.
CDP_EX_DEV_NO_RESPONSE (62102)	No command response from the display. (1) An unsupported display is connected. (2) The setting of the interface board is incorrect.

This property is initialized to 0 at the beginning of the connection process. Refer to this after executing the [connect method](#).

Set property

Not exist.

Get property

```
int getErrorCodeExtended ()
```

Returns the extended error code as the return value.

3.5. Notes

Notes of this SDK are as follows.

3.5.1. Logging function

This SDK has a logging function which keeps history of executing methods or reading/writing properties. This can be enabled by using the [setLog method](#), or placing a file "CSJPOSLibD.cfg" in the /temp/citizen at the top of the external storage path.

On Android 11 or later, to access the above path, you need to grant the application "Access to all files (MANAGE_EXTERNAL_STORAGE) permission" and allow the permission in the device's app settings. Please refer to the attached sample "AndroidManifest.xml" for how to define the application.

```
<uses-permission android:name="android.permission.MANAGE_EXTERNAL_STORAGE" />
```

< Example of CSJPOSLibD.cfg >

[LogSetting]	...Section name (Fixed)
LogMode=1	...Specifies the log mode.
LogPath=/storage/emulated/0/Download/Log	...Specifies the folder to output the log files.
LogMaxSize=10	...Specifies the maximum size of log file in MB.

The external storage path depends on the device. For many terminals, the path to place the file will be one of the following.

```
/storage/sdcard0/temp/citizen/Log/
or
/storage/emulated/0/temp/citizen/Log/
```

Setting items

- LogMode

Specifies a log mode:

- 0: None
- 1: Access log
- 2: Error log

- LogPath

Specifies a folder to store the log files.

If a full path starting with /storage/ is specified for the storage folder, the files will be stored in the specified path. Otherwise, it is specified as the path from the top of the external storage path. "/temp/citizen" of the external storage path by default.

- LogMaxSize

Specifies maximum size to log file in MB. "0" sets the size to "unlimited."

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

When the same file name exists, the file will be overwritten if the file is one week older, new logs will be added to the existing file if the file is created on the same day.

Log format

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

```
--- Example 1, method (Connect) ---  
  
2018/06/29 11:06:21.018 001 METHOD call    connect(0, "192.168.10.100")  
2018/06/29 11:06:21.196 001 METHOD result connect() -> Success(0)  
  
--- Example 2, method (DisplayText) ---  
  
2018/06/29 11:06:24.578 001 METHOD call    displayText([See below])  
-----Parameter Detail-----  
2018/06/29 11:06:23  
-----  
2018/06/29 11:06:24.588 001 METHOD result displayText()-> Success(0)
```

- * The logging function could be a "bottleneck" of processing since it tries to keep the information of every single execution of a method or access to a property.
- * Logging could fail without a notification for the reason below.
 - A write-protected device is selected as a storage location.
 - No enough space in the selected storage device.
 - The storage location contains a file with write-protection.
 - No access privilege to a file or folder.
 - Another program is using the log file.

3.5.2. Predefined Constants List

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_NOCONTEXT	int	1006	No context information specified
		CDP_E_ILLEGAL	int	1101	Unsupported or invalid parameter
		CDP_E_OFFLINE	int	1102	Off-line
		CDP_E_FAILURE	int	1104	Process failure
2	Connection interface	CDP_PORT_WiFi	int	0	Network
		CDP_PORT_Bluetooth	int	1	Bluetooth
		CDP_PORT_Bluetooth_Insecure	int	2	Bluetooth Insecure
3	Clear area	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 specification	CDP_TYPE_NONE	int	0	Hide cursor
		CDP_TYPE_UNDERLINE	int	1	Display cursor
6	Extended error code	CDP_EX_DEV_NO_PRINTER	int	62000	Printer not connected
		CDP_EX_DEV_NO_DEVICE	int	62001	Peripheral device not connected
		CDP_EX_DEV_USING	int	62002	Peripheral device in use
		CDP_EX_DEV_CONFIRM	int	62003	Confirmation failure after peripheral device connection
		CDP_EX_DEV_OPEN_ERROR	int	62100	Failed to connect socket
		CDP_EX_DEV_SEND_ERROR	int	62101	Failed to send command
		CDP_EX_DEV_NO_RESPONSE	int	62102	Command no response

4. Barcode Scanner Control

4.1. Definition method

Add the library file that is provided, if you use this SDK, please import the following two classes.

```
import com.citizen.sdk.ScannerConst;
import com.citizen.sdk.Scanner;
import com.citizen.sdk.DataListener;
import com.citizen.sdk.StatusUpdateListener;
```

4.2. Program structure

Here is an example program which uses the SDK

<pre>// Create an instance. Scanner scanner = new Scanner(); // Data event callback DataListener datalistener = new DataListener(){ @Override public void onDataEvent(final byte[] scanData) { Log.d("Scanner", "Data call back: " + new String(scanData)); } }; scanner.setDataEventCallback(datalistener); // Status update event callback StatusUpdateListener statuslistener = new StatusUpdateListener(){ @Override public void onStatusUpdateEvent(final int status) { Log.d("Scanner", "Status Call back: " + status); } }; scanner.setStatusUpdateEventCallback(statuslistener); // Connect Scanner void connectScanner() { int result = scanner.connect(ScannerConst.CSC_PORT_WiFi, "192.168.0.10"); if (ScannerConst.CSC_SUCCESS != result){ Log.e("Scanner", "Connect Error : " + result); } } // Disconnect Scanner void disconnectScanner() { scanner.disconnect(); }</pre>	<div style="margin-bottom: 20px;"> } Class definition </div> <div style="margin-bottom: 20px;"> } Callback processes </div> <div style="margin-bottom: 20px;"> } Connect </div> <div> } Disconnect </div>
----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

4.3. Functions list

This SDK provides the following functions.

Methods list

No	Function	Detail
1	Create class (Constructor)	This is constructor method.
2	Connect scanner (connect method)	This method connects to the scanner.
3	Disconnect scanner (disconnect method)	This method disconnects the scanner connection.
4	Get version code (getVersionCode method)	This method gets a numerical value for the version number of this SDK.
5	Get version name (getVersionName method)	This method gets a string for the version number of this SDK.
6	Set log (setLog method)	Sets the log function.

Properties List

No	Function	Attribute	Detail
1	Extended error code (ErrorCodeExtended property)	R	Show the extended error code when connecting.

Events list

No	Function	Detail
1	Scan data notification (DataEvent event)	Notifies barcode reading data.
2	Status update notification (StatusUpdateEvent event)	Notify device status update.

4.4. Library interfaces

The following are the interfaces of this SDK.

4.4.1. 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_NOCONTEXT (1006)	The context is not specified.
CSC_E_BT_DISABLE (1007)	The setting of the Bluetooth device is invalid.
CSC_E_BT_NODEVICE (1008)	The Bluetooth device is not found.
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_FAILURE (1104)	The Service cannot perform the requested procedure.

4.4.2. 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();
```

4.4.3. connect method

Syntax

- 1) int connect (int connectType, String addr)
- 2) int connect (int connectType, String addr, int port)
- 3) int connect (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 CSC_PORT_Bluetooth_Insecure
addr	[IN]	IP address to connect or Bluetooth device address or Bluetooth device name	WiFi: 0.0.0.0~255.255.255.255 Bluetooth: 00:00:00:00:00:00~FF:FF:FF:FF:FF:FF Device name (Automatic detection)
port	[IN]	Connection port number	
timeout	[IN]	Timeout (msec)	

Description

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

Bluetooth device address letters, please specify in uppercase.

If the Bluetooth Device name is specified, the device that was paired is detected automatically. If you omit the Device name, the supported model device that was paired is detected automatically.

If you want to use the Bluetooth device insecure communications provided by the Android 2.3.3 or later, please specify the connection type CSC_PORT_Bluetooth_Insecure.

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 and connected with 8000 milliseconds when using Bluetooth.

When communication with the scanner is not necessary, must execute the [disconnect 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 "[4.4.1 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.
CSC_E_BT_DISABLE (1007)	The setting of the Bluetooth device is invalid.
CSC_E_BT_NODEVICE (1008)	The Bluetooth device is not found.

Example

```
scanner.connect(ScannerConst.CSC_PORT_WiFi, "192.168.0.10");
```

4.4.4. disconnect method

Syntax

```
int disconnect ()
```

Parameter

Not exist.

Description

This method is used to disconnect the scanner connection.

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

Return value

Return CSC_SUCCESS(0) in success. Please refer to "[4.4.1 Return value](#)" for the error code except it.

Example

```
scanner.disconnect();
```

4.4.5. 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

```
scanner.getVersionCode ();
```

4.4.6. 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

```
scanner.getVersionName();
```

4.4.7. 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]	Log Output Folder	Folder path
maxSize	[IN]	Maximum Log Size	0: Unlimited 1 - : Maximum size (MB)

Description

Sets the logging function. See ["4.5.1 Logging function"](#) for more details.

For the output folder, specify the path to a folder that the application that outputs the log file can write to. If a full path other than the one beginning with '/storage/' is specified, it will be specified as the path from the top of the external storage path.

Return value

none

Example

```
// Specifying the folder for the application's specific area.
scanner.setLog(1, this.getExternalFilesDir(null).toString() + "/Log", 10);

// Specifying the public folder (Download).
scanner.setLog(1, Environment.getExternalStoragePublicDirectory(
    Environment.DIRECTORY_DOWNLOADS) + "/Log", 10);
```


3.4.8. ErrorCodeExtended property

Syntax

int ErrorCodeExtended

Attribute

Read

Description

This property holds the extended error code when there is an error connecting to the scanner. The values are as follows.

Value	Meaning
CSC_EX_DEV_NO_PRINTER (62000)	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.
CSC_EX_DEV_NO_DEVICE (62001)	Scanner is not connected. (1) The scanner is not connected. (2) The setting of the interface port is incorrect.
CSC_EX_DEV_USING (62002)	Scanner is in use. (1) The scanner is in use by another application.
CSC_EX_DEV_CONFIRM (62003)	Failed to check the status after connecting to the scanner. (1) Wrong specified port number.
CSC_EX_DEV_OPEN_ERROR (62100)	Unable to connect socket. (1) Communication error.

This property is initialized to 0 at the beginning of the connection process. Refer to this after executing the [connect method](#).

Set property

Not exist.

Get property

int getErrorCodeExtended ()

Returns the extended error code as the return value.

4.4.9. DataEvent event

Syntax

```
void onDataEvent(byte[] data)
```

Parameter

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

Value	[IN/OUT]	Meaning
data	[IN]	Barcode read data

Description

The barcode scanner notifies of the data read from the barcode.

The callback method receives byte array type arguments as information read by the barcode scanner from the barcode.

Set callback

```
int setDataEventCallback(DataListener listener)
```

Registers a callback method to receive the information that the barcode scanner read from the barcode. When setDataEventCallback method is executed out several times, it is overwritten an appointed callback method last. When null is set for object designation, a callback is canceled.

[Callback interface]

```
public interface DataListener {
    // callback method
    public abstract void onDataEvent(byte[] data);
}
```

Example

```
// Scan data callback definition
DataListener datalistener = new DataListener() {
    @Override
    public void onDataEvent(final byte[] scanData) {
        Log.d("Scanner", "Data call back: " + new String(scanData));
    }
};

scanner.setDataEventCallback(datalistener);
```

4.4.10. StatusUpdateEvent event

Syntax

```
void onStatusUpdateEvent(int status)
int setStatusUpdateEventCallback(StatusUpdateListener listener)
```

Parameter

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

Value	[IN/OUT]	Meaning
status	[IN]	Status update of device

Description

Notifies device status update.

The callback method receives an int type argument indicating the status as information on the status update of the device.

Status code	Description
CSC_SUE_POWER_ONLINE (2001)	Device is ready
CSC_SUE_POWER_OFF (2002)	Connection fault or not connected to the printer

Set callback

```
int setStatusUpdateEventCallback(StatusUpdateListener listener)
```

Registers a callback method that status receives update information.

When setStatusUpdateEventCallback method is executed out several times, it is overwritten an appointed callback method last. When null is set for object designation, a callback is canceled.

[Callback interface]

```
public interface StatusUpdateListener {
    // callback method
    public abstract void onStatusUpdateEvent(int status);
}
```

Example

```
// Status callback definition
StatusUpdateListener statuslistener = new StatusUpdateListener() {
    @Override
    public void onStatusUpdateEvent(final int status) {
        Log.d("Scanner", "Status update call back: " + status);
    }
};

scanner.setStatusUpdateEventCallback(statuslistener);
```

4.5. Notes

Notes of this SDK are as follows.

4.5.1. Logging function

This SDK has a logging function which keeps history of executing methods or reading/writing properties. This can be enabled by using the [setLog method](#), or placing a file "CSJPOSLibS.cfg" in the /temp/citizen at the top of the external storage path.

On Android 11 or later, to access the above path, you need to grant the application "Access to all files (MANAGE_EXTERNAL_STORAGE) permission" and allow the permission in the device's app settings. Please refer to the attached sample "AndroidManifest.xml" for how to define the application.

```
<uses-permission android:name="android.permission.MANAGE_EXTERNAL_STORAGE" />
```

< Example of CSJPOSLibS.cfg >

[LogSetting]	...Section name (Fixed)
LogMode=1	...Specifies the log mode.
LogPath=/storage/emulated/0/Download/Log	...Specifies the folder to output the log files.
LogMaxSize=10	...Specifies the maximum size of log file in MB.

The external storage path depends on the device. For many terminals, the path to place the file will be one of the following.

```
/storage/sdcard0/temp/citizen/Log/
or
/storage/emulated/0/temp/citizen/Log/
```

Setting items

- LogMode

Specifies a log mode:

- 0: None
- 1: Access log
- 2: Error log

- LogPath

Specifies a folder to store the log files.

If a full path starting with /storage/ is specified for the storage folder, the files will be stored in the specified path. Otherwise, it is specified as the path from the top of the external storage path. "/temp/citizen" of the external storage path by default.

- LogMaxSize

Specifies maximum size to log file in MB. "0" sets the size to "unlimited."

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

When the same file name exists, the file will be overwritten if the file is one week older, new logs will be added to the existing file if the file is created on the same day.

Log format

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

```
--- Example 1, method (Connect) ---  
2018/06/29 12:06:21.018 001 METHOD call    connect(0, "192.168.10.100")  
2018/06/29 11:06:21.196 001 METHOD result connect() -> Success(0)  
  
--- Example 2, event (DataEvent) ---  
2018/06/29 12:58:57.956 4481 EVENT      DataEvent : 31 32 33 34 35 36 37 38 39 30 31 32
```

- * The logging function could be a "bottleneck" of processing since it tries to keep the information of every single execution of a method or access to a property.
- * Logging could fail without a notification for the reason below.
 - A write-protected device is selected as a storage location.
 - No enough space in the selected storage device.
 - The storage location contains a file with write-protection.
 - No access privilege to a file or folder.
 - Another program is using the log file.

4.5.2. Predefined Constants List

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_NOCONTEXT	int	1006	No context information specified
		CSC_E_ILLEGAL	int	1101	Unsupported or invalid parameter
		CSC_E_OFFLINE	int	1102	Off-line
		CSC_E_FAILURE	int	1104	Process failure
2	Connection interface	CSC_PORT_WiFi	int	0	Network
		CSC_PORT_Bluetooth	int	1	Bluetooth
		CSC_PORT_Bluetooth_Insecure	int	2	Bluetooth Insecure
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
4	Extended error code	CSC_EX_DEV_NO_PRINTER	int	62000	Printer not connected
		CSC_EX_DEV_NO_DEVICE	int	62001	Peripheral device not connected
		CSC_EX_DEV_USING	int	62002	Peripheral device in use
		CSC_EX_DEV_CONFIRM	int	62003	Confirmation failure after peripheral device connection
		CSC_EX_DEV_OPEN_ERROR	int	62100	Failed to connect socket or stream

CITIZEN Android POS Print SDK - Programming Manual

August 14, 2025 For Ver. 2.14

CITIZEN SYSTEMS JAPAN CO., LTD.

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