

CITIZEN

Xamarin POS Print SDK

プログラムマニュアル

Ver. 1.0.1.0 用

シチズン・システムズ株式会社

更新履歴

年月日	バージョン	履歴
2020/01/06	1.0.0.0	新規
2021/04/09	1.0.1.0	Lightning I/F に関する説明を追加 (5,16,32,91,118 頁) プリンターの対応機種に CT-E301 と CT-E601 を追加 (8,9 頁) プリンターの対応機種(iOS)に CMP-20/30/40 を追加 (8 頁) PrintTextLocalFont メソッドの iOS 対応 (42 頁) PrintBitmap メソッドのモード引き数に CMP_BM_MODE_CMD_GRAY16DOWNLOAD を追加 (43 ~44 頁) SetPrintCompletedTimeout メソッドを追加 (71 頁)

ご注意

1. 本書の内容の一部、または全部を無断で転載することは、固くお断りいたします。
2. 本書の内容については、事前の予告なしに変更することがあります。
3. 本書の内容については万全を期して作成いたしましたが、万一誤り・お気付きの点がございましたら、ご連絡くださいますようお願いいたします。
4. 運用した結果の影響につきましては、3項にかかわらず責任を負いかねますのでご了承ください。
5. 上記に同意いただけない場合は、本SDKをご使用いただけません。

商標

CITIZEN はシチズン時計株式会社の商標または登録商標です。

Android は米国およびその他の国における Google Inc.の商標または登録商標です。

Java は米国およびその他の国における Sun Microsystems, Inc.の商標または登録商標です。

Bluetooth®のワードマークおよびロゴは Bluetooth SIG, Inc.が所有する登録商標です。

その他、記載されている会社名、製品名は、各社の商標または登録商標です。

オープンソースソフトウェアについて

本製品で利用しているオープンソースソフトウェアとオープンソースソフトウェアのライセンスは以下の通りです。

・Cross-Platform .NET Standard Plugin Templates

本製品は、Cross-Platform .NET Standard Plugin Templates を利用しています。

The MIT License (MIT)

Copyright (c) 2016 James Montemagno / Refractored LLC

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

目次

1. はじめに	7
1.1. ドキュメント対象範囲	7
1.2. システム概要	7
1.3. 対応端末	7
1.4. 対応モデル（プリンター）	8
1.5. 対象モデル（周辺機器）	16
1.6. 定義方法	19
2. プリンター制御	26
2.1. プログラム構造	26
2.2. 機能一覧	27
2.3. ライブラリインターフェース	29
2.3.1. 戻り値	29
2.3.2. コンストラクタ	30
2.3.3. <i>SetContext</i> メソッド (Android のみ)	31
2.3.4. <i>Connect</i> メソッド	32
2.3.5. <i>Disconnect</i> メソッド	34
2.3.6. <i>SetEncoding</i> メソッド	35
2.3.7. <i>PrinterCheck</i> メソッド	36
2.3.8. <i>Status</i> メソッド	37
2.3.9. <i>PrintText</i> メソッド	39
2.3.10. <i>PrintPaddingText</i> メソッド	40
2.3.11. <i>PrintTextLocalFont</i> メソッド	42
2.3.12. <i>PrintBitmap</i> メソッド	43
2.3.13. <i>SetNVBitmap</i> メソッド	45
2.3.14. <i>PrintNVBitmap</i> メソッド	47
2.3.15. <i>PrintBarCode</i> メソッド	48
2.3.16. <i>PrintPDF417</i> メソッド	50
2.3.17. <i>PrintQRCode</i> メソッド	51
2.3.18. <i>PrintGS1DataBarStacked</i> メソッド	52
2.3.19. <i>CutPaper</i> メソッド	53
2.3.20. <i>UnitFeed</i> メソッド	54
2.3.21. <i>MarkFeed</i> メソッド	55
2.3.22. <i>OpenDrawer</i> メソッド	56
2.3.23. <i>TransactionPrint</i> メソッド	57
2.3.24. <i>RotatePrint</i> メソッド	58
2.3.25. <i>PageModePrint</i> メソッド	59
2.3.26. <i>ClearPrintArea</i> メソッド	61
2.3.27. <i>ClearOutput</i> メソッド	62
2.3.28. <i>PrintData</i> メソッド	63
2.3.29. <i>PrintNormal</i> メソッド	64
2.3.30. <i>WatermarkPrint</i> メソッド	65
2.3.31. <i>SearchCitizenPrinter</i> メソッド	66
2.3.32. <i>SearchESCPOSPrinter</i> メソッド	68
2.3.33. <i>SetLog</i> メソッド (Android のみ)	70
2.3.34. <i>SetPrintCompletedTimeout</i> メソッド	71
2.3.35. <i>GetVersionCode</i> メソッド	72
2.3.36. <i>GetVersionName</i> メソッド	73
2.3.37. <i>PageModeArea</i> プロパティ	74

2.3.38. PageModePrintArea プロパティ.....	75
2.3.39. PageModePrintDirection プロパティ.....	76
2.3.40. PageModeHorizontalPosition プロパティ.....	77
2.3.41. PageModeVerticalPosition プロパティ.....	78
2.3.42. RecLineSpacing プロパティ.....	79
2.3.43. MapMode プロパティ.....	80
2.4. 注意事項.....	81
2.4.1. 印刷完了確認機能について.....	81
2.4.2. UTF-8 エンコード文字列の印刷について.....	81
2.4.3. JIS 第3、第4水準漢字の印刷について.....	82
2.4.4. ログ機能について(Android のみ).....	83
2.4.5. 定数定義一覧.....	85
3. ラインディスプレイ制御.....	88
3.1. プログラム構造.....	88
3.2. 機能一覧.....	89
3.3. ライブラリインターフェース.....	90
3.3.1. 戻り値.....	90
3.3.2. コンストラクタ.....	91
3.3.3. Connect メソッド.....	92
3.3.4. Disconnect メソッド.....	94
3.3.5. DisplayText メソッド.....	95
3.3.6. ClearDisplay メソッド.....	96
3.3.7. BlinkDisplay メソッド.....	97
3.3.8. SetDisplayMode メソッド.....	98
3.3.9. SetDisplayConfig メソッド.....	99
3.3.10. SetCursorPosition メソッド.....	100
3.3.11. MoveCursor メソッド.....	101
3.3.12. SetCursorType メソッド.....	102
3.3.13. InitializeDisplay メソッド.....	103
3.3.14. DisplayData メソッド.....	104
3.3.15. SetEncoding メソッド.....	105
3.3.16. SetCodePage メソッド.....	106
3.3.17. SetInternationalCharacterSet メソッド.....	107
3.3.18. DisplayCheck メソッド.....	108
3.3.19. GetVersionCode メソッド.....	109
3.3.20. GetVersionName メソッド.....	110
3.3.21. SetLog メソッド (Android のみ).....	111
3.4. 注意事項.....	112
3.4.1. ログ機能について(Android のみ).....	112
3.4.2. 定数定義一覧.....	114
4. バーコードスキャナー制御.....	115
4.1. プログラム構造.....	115
4.2. 機能一覧.....	116
4.3. ライブラリインターフェース.....	117
4.3.1. 戻り値.....	117
4.3.2. コンストラクタ.....	118
4.3.3. Connect メソッド.....	119
4.3.4. Disconnect メソッド.....	121
4.3.5. SetDataEventCallback メソッド.....	122
4.3.6. SetStatusUpdateEventCallback メソッド.....	123

4.3.7. <i>GetVersionCode</i> メソッド	124
4.3.8. <i>GetVersionName</i> メソッド.....	125
4.3.9. <i>SetLog</i> メソッド (<i>Android</i> のみ).....	126
4.4. 注意事項	127
4.4.1. ログ機能について(<i>Android</i> のみ).....	127
4.4.2. 定数定義一覧.....	129

1. はじめに

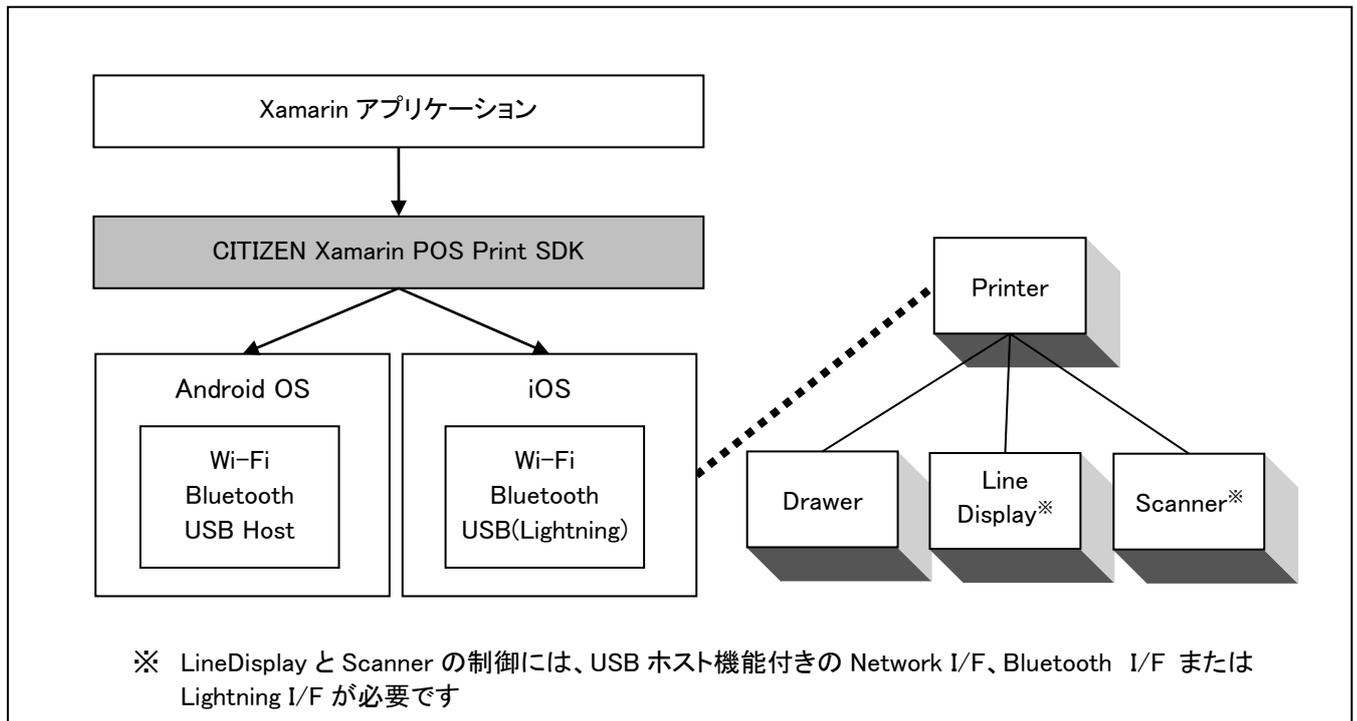
本ドキュメントは、CITIZEN Xamarin POS Print SDK のプログラムマニュアルです。

1.1. ドキュメント対象範囲

本ドキュメントは、CITIZEN POS プリンターおよびプリンターに接続された周辺機器を利用する Xamarin アプリケーションの開発者が参照することを目的としています。

1.2. システム概要

本 SDK は CITIZEN POS プリンターおよびプリンターに接続された周辺機器を利用する Xamarin アプリケーションから参照されることを想定しています。



SDK システム構成図

1.3. 対応端末

本 SDK が対応する端末の仕様は以下の通りです。

Android OS バージョン :	対応インターフェース :
Android 5.0 (API Level 21) 以上	Wi-Fi, Bluetooth, USB Host
iOS バージョン :	対応インターフェース :
iOS 8.0 以上	Wi-Fi, Bluetooth
iOS 10.0.2 以上	USB(Lightning)

1.4. 対応モデル(プリンター)

本 SDK の対象プリンターモデルおよびそのモデルに対応するインターフェースは以下の通りです。
各モデルの機能詳細についてはプリンターの取り扱い説明書をご参照ください。

モデル系列	対象モデル	インターフェース	機能
CT-E301 系	CT-E301	有線 LAN USB (Android のみ)	標準
CT-E601 系	CT-E601	Bluetooth 有線/無線 LAN USB	標準
CT-S251 系	CT-S251	Bluetooth 有線/無線 LAN USB	標準
CT-S253 系	CT-S253	有線 LAN USB (Android のみ)	標準
CT-S255 系	CT-S255	Bluetooth 有線/無線 LAN USB	標準
	CT-S255-L		ラベル/ブラックマーク紙対応
CT-S257 系	CT-S257	Bluetooth 有線/無線 LAN USB	標準
CT-S281 系	CT-S281	Bluetooth USB (Android のみ)	標準
	CT-S281-L		ラベル/ブラックマーク紙対応
CT-S401 系	CT-S401	有線 LAN USB (Android のみ)	標準
CT-S601/651/801/851 系	CT-S601/651/801/851	有線/無線 LAN USB (Android のみ)	標準
	CT-S801/851-M		ブラックマーク紙対応
	CT-S801-L		ラベル紙対応
CT-S601II/651II/801II/ 851II 系	CT-S601II/651II/801II/ 851II	Bluetooth 有線/無線 LAN USB (Android のみ)	標準
	CT-S801II/851II-M		ブラックマーク紙対応
	CT-S801II-L		ラベル紙対応
CT-S2000 系	CT-S2000	有線 LAN USB (Android のみ)	標準
	CT-S2000-M		ブラックマーク紙対応
	CT-S2000-L		ラベル紙対応
CT-S4000 系	CT-S4000	有線 LAN USB (Android のみ)	標準(表面ブラックマーク紙対応)
	CT-S4000-M		裏面ブラックマーク紙対応
	CT-S4000-L		ラベル紙対応
CT-S4500 系	CT-S4500	Bluetooth 有線/無線 LAN USB	標準(ラベル/ブラックマーク紙対応)
PMU3300 系 (Android のみ)	PMU3300	USB (Android のみ)	標準
CMP-20/30/40/20II/30II 系	CMP-20/30/40/20II/30II (ESC/POS)	Bluetooth 無線 LAN USB (Android のみ)	標準

本 SDK をご使用になる際は、プリンターのメモリースイッチ設定が以下の通り設定されていることが条件となります。

CT-E301 系 メモリースイッチ設定

MSW No.	機能	設定
1-1	電源 ON 通知設定	有効
1-2	インプットバッファ	4K bytes
1-3	Busy 条件	バッファフル
1-4	受信エラー文字	?文字
1-5	CR モード	無効
2-2	オートカッター動作	有効
2-4	フル桁印字	データ待ち
3-1	オートカッター復帰	L/F 有効
3-7	CBM1000 互換モード	有効
3-8	印字中カバーオープン	自動復帰
4-8	強制パーシャル	無効
5-2	縦基本計算ピッチ	360
5-3	USB モード	プリンタクラス
6-1	ドライバ用動作	有効
7-6	DMA 制御	有効
9-1	コードページ	Katakana (*1)
9-2	国際文字	日本 (*1)
9-4	漢字コード (*2)	SJIS(CP932) GB18030 EUC Hangul BIG5

CT-E601 系 メモリースイッチ設定

MSW No.	機能	設定
1-1	電源 ON 通知設定	有効
1-2	インプットバッファ	4K bytes
1-3	Busy 条件	バッファフル
1-4	受信エラー文字	?文字
1-5	CR モード	無効
2-2	オートカッター動作	有効
2-4	フル桁印字	データ待ち
3-1	オートカッター復帰	L/F 有効
3-7	CBM1000 互換モード	有効
3-8	印字中カバーオープン	自動復帰
4-8	強制パーシャル	無効
5-2	縦基本計算ピッチ	360
6-1	ドライバ用動作	有効
7-6	DMA 制御	有効
9-1	コードページ	Katakana (*1)
9-2	国際文字	日本 (*1)
9-4	漢字コード (*2)	SJIS(CP932) GB18030 EUC Hangul BIG5
13-6	再接続要求 (Bluetooth I/F 使用時)	Android 使用時: 無効 iOS 使用時: 有効

CT-S251 系 メモリースイッチ設定

MSW No.	機能	設定
1-1	電源 ON 通知設定	有効
1-2	インプットバッファ	4K bytes

MSW No.	機能	設定
1-3	Busy 条件	バッファフル
1-4	受信エラー文字	?文字
1-5	CR モード	無効
2-2	オートカット動作	有効
2-4	フル桁印字	データ待ち
3-1	オートカット復帰	L/F 有効
3-7	CBM1000 互換モード	有効
3-8	印字中カバーオープン	自動復帰
4-8	強制パーシャル	無効
5-2	縦基本計算ピッチ	360
6-1	ドライバ用動作	有効
7-6	DMA 制御	有効
9-1	コードページ	Katakana (*1)
9-2	国際文字	日本 (*1)
9-3	漢字	有効 (*1)
9-4	JIS/シフト JIS	シフト JIS (*1)
13-6	再接続要求 (Bluetooth I/F 使用時)	Android 使用時: 無効 iOS 使用時: 有効

CT-S253 系 メモリースイッチ設定

MSW No.	機能	設定
1-1	電源 ON 通知設定	有効
1-2	インプットバッファ	4K bytes
1-3	Busy 条件	バッファフル
1-4	受信エラー文字	?文字
1-5	CR モード	無効
2-2	オートカッター動作	有効
2-4	フル桁印字	データ待ち
3-1	オートカッター復帰	L/F 有効
3-7	CBM1000 互換モード	有効
3-8	印字中カバーオープン	自動復帰
4-8	強制パーシャル	無効
5-2	縦基本計算ピッチ	360
5-3	USB モード	プリンタクラス
6-1	ドライバ用動作	有効
7-6	DMA 制御	有効
9-1	コードページ	Katakana (*1)
9-2	国際文字	日本 (*1)
9-4	漢字コード (*2)	SJIS(CP932) GB18030 EUC Hangul BIG5

CT-S255 系 メモリースイッチ設定

MSW No.	機能	設定
1-1	電源 ON 通知設定	有効
1-2	インプットバッファ	4K bytes
1-3	Busy 条件	バッファフル
1-4	受信エラー文字	?文字
1-5	CR モード	無効
2-2	オートカッター動作	有効

MSW No.	機能	設定
2-4	フル桁印字	データ待ち
3-1	オートカッター復帰	L/F 有効
3-7	CBM1000 互換モード	有効
3-8	印字中カバーオープン	自動復帰
4-8	強制パーシャル	無効
5-2	縦基本計算ピッチ	360
6-1	ドライバ用動作	有効
7-6	DMA 制御	有効
9-1	コードページ	Katakana (*1)
9-2	国際文字	日本 (*1)
9-4	漢字コード (*2)	SJIS(CP932) GB18030 EUC Hangul BIG5
13-6	再接続要求 (Bluetooth I/F 使用時)	Android 使用時: 無効 iOS 使用時: 有効

CT-S257 系 メモリースイッチ設定

MSW No.	機能	設定
1-1	電源 ON 通知設定	有効
1-2	インプットバッファ	4K bytes
1-3	Busy 条件	バッファフル
1-4	受信エラー文字	?文字
1-5	CR モード	無効
2-2	オートカッター動作	有効
2-4	フル桁印字	データ待ち
3-1	オートカッター復帰	L/F 有効
3-7	CBM1000 互換モード	有効
3-8	印字中カバーオープン	自動復帰
4-8	強制パーシャル	無効
5-2	縦基本計算ピッチ	360
6-1	ドライバ用動作	有効
7-6	DMA 制御	有効
9-1	コードページ	Katakana (*1)
9-2	国際文字	日本 (*1)
9-4	漢字コード (*2)	SJIS(CP932) GB18030 EUC Hangul BIG5-HKSCS
13-6	再接続要求 (Bluetooth I/F 使用時)	Android 使用時: 無効 iOS 使用時: 有効

CT-S281 系 メモリースイッチ設定

MSW No.	機能	設定
1-1	電源 ON 通知設定	有効
1-2	インプットバッファ	4K bytes
1-3	Busy 条件	バッファフル
1-4	受信エラー文字	?文字
1-5	CR モード	無効
2-2	オートカッター動作	有効
2-4	フル桁印字	データ待ち
3-1	オートカッター復帰	L/F 有効

3-7	CBM-270 互換モード	有効
3-8	印字中カバーオープン	自動復帰
4-8	強制パーシャル	無効
5-3	USB モード	プリンタクラス
9-1	コードページ	Katakana (*1)
9-2	国際文字	日本 (*1)
9-3	漢字	有効 (*1)
9-4	JIS/シフト JIS	シフト JIS (*1)
13-6	再接続要求 (Bluetooth I/F 使用時)	Android 使用時: 無効 iOS 使用時: 有効

CT-S401 系 メモリースイッチ設定

MSW No.	機能	設定
1-1	電源 ON 通知設定	有効
1-2	インプットバッファ	4K bytes
1-3	Busy 条件	バッファフル
1-4	受信エラー文字	?文字
1-5	CR モード	無効
2-2	オートカット動作	有効
2-4	フル桁印字	データ待ち
3-1	オートカット復帰	L/F 有効
3-7	CBM1000 互換モード	有効
3-8	印字中カバーオープン	自動復帰
4-8	強制パーシャル	無効
5-2	縦基本計算ピッチ	360
5-3	USB モード	プリンタクラス
6-1	ドライバ用動作	有効
7-6	DMA 制御	有効
9-1	コードページ	Katakana (*1)
9-2	国際文字	日本 (*1)
9-3	漢字	有効 (*1)
9-4	JIS/シフト JIS	シフト JIS (*1)

CT-S601/651/801/851 系 メモリースイッチ設定

MSW No.	機能	設定
1-1	電源 ON 通知設定	有効
1-2	インプットバッファ	4K bytes
1-3	Busy 条件	バッファフル
1-4	受信エラー文字	?文字
1-5	CR モード	無効
2-2	オートカット動作	有効
2-4	フル桁印字	データ待ち
3-1	オートカット復帰	L/F 有効
3-3	パラレル 31Pin	リセット
3-7	CBM1000 互換モード	有効
3-8	印字中カバーオープン	自動復帰
4-8	強制パーシャル	無効
5-2	縦基本計算ピッチ	360
5-3	USB モード	プリンタクラス
6-1	ドライバ用動作	有効
7-6	DMA 制御	有効

MSW No.	機能	設定
9-1	コードページ	Katakana (*1)
9-2	国際文字	日本 (*1)
9-3	漢字	有効 (*1)
9-4	JIS/シフト JIS	シフト JIS (*1)
10-3	ACK 出カタイミング	BUSY 前

CT-S601II/651II/801II/851II 系 メモリースイッチ設定

MSW No.	機能	設定
1-1	電源 ON 通知設定	有効
1-2	インプットバッファ	4K bytes
1-3	Busy 条件	バッファフル
1-4	受信エラー文字	?文字
1-5	CR モード	無効
2-2	オートカット動作	有効
2-4	フル桁印字	データ待ち
3-1	オートカット復帰	L/F 有効
3-3	パラレル 31Pin	リセット
3-7	CBM1000 互換モード	有効
3-8	印字中カバーオープン	自動復帰
4-8	強制パーシャル	無効
5-2	縦基本計算ピッチ	360
5-3	USB モード	プリンタクラス
6-1	ドライバ用動作	有効
7-6	DMA 制御	有効
9-1	コードページ	Katakana (*1)
9-2	国際文字	日本 (*1)
9-4	漢字コード (*2)	SJIS(CP932) GB18030 EUC Hangul BIG5
10-3	ACK 出カタイミング	BUSY 前
13-6	再接続要求 (Bluetooth I/F 使用時)	Android 使用時: 無効 iOS 使用時: 有効

CT-S2000 系 メモリースイッチ設定

MSW No.	機能	設定
1-1	電源 ON 通知設定	有効
1-2	インプットバッファ	4K bytes
1-3	Busy 条件	バッファフル
1-4	受信エラー文字	?文字
1-5	CR モード	無効
2-2	オートカット動作	有効
2-4	フル桁印字	データ待ち
3-1	オートカット復帰	L/F 有効
3-3	パラレル 31Pin	リセット
3-7	CBM1000 互換モード	有効
3-8	印字中カバーオープン	自動復帰
4-8	強制パーシャル	無効
5-2	縦基本計算ピッチ	360
7-6	DMA 制御	有効
9-1	コードページ	Katakana (*1)

MSW No.	機能	設定
9-2	国際文字	日本 (*1)
9-3	漢字	有効 (*1)
9-4	JIS/シフト JIS	シフト JIS (*1)
10-3	ACK 出カタイミング	BUSY 前

CT-S4000 系 メモリースイッチ設定

MSW No.	機能	設定
1-1	電源 ON 通知設定	有効
1-2	インプットバッファ	4K bytes
1-3	Busy 条件	バッファフル
1-4	受信エラー文字	?文字
1-5	CR モード	無効
2-2	オートカット動作	有効
2-4	フル桁印字	データ待ち
3-1	オートカット復帰	L/F 有効
3-3	パラレル 31Pin	リセット
3-7	CBM1000 互換モード	有効
3-8	印字中カバーオープン	自動復帰
4-8	強制パーシャル	無効
5-2	縦基本計算ピッチ	360
7-6	DMA 制御	有効
9-1	コードページ	Katakana (*1)
9-2	国際文字	日本 (*1)
9-3	漢字	有効 (*1)
9-4	JIS/シフト JIS	シフト JIS (*1)
10-3	ACK 出カタイミング	BUSY 前

CT-S4500 系 メモリースイッチ設定

MSW No.	機能	設定
1-1	電源 ON 通知設定	有効
1-2	インプットバッファ	4K bytes
1-3	Busy 条件	バッファフル
1-4	受信エラー文字	?文字
1-5	CR モード	無効
2-2	オートカッター動作	有効
2-4	フル桁印字	データ待ち
3-1	オートカッター復帰	L/F 有効
3-7	CBM1000 互換モード	有効
3-8	印字中カバーオープン	自動復帰
4-8	強制パーシャル	無効
5-2	縦基本計算ピッチ	360
6-1	ドライバ用動作	有効
7-6	DMA 制御	有効
9-1	コードページ	Katakana (*1)
9-2	国際文字	日本 (*1)
9-4	漢字コード (*2)	SJIS(CP932) GB18030 EUC Hangul BIG5-HKSCS
13-6	再接続要求 (Bluetooth I/F 使用時)	Android 使用時: 無効 iOS 使用時: 有効

*1 MSW No.9-1~4 は、日本語使用時の設定です。ご使用環境に合わせて変更してください。

*2 CT-E301/601、CT-S253/255/257/601II/651II/801II/851II/4500 系は、漢字コードを Shift_JIS、GB18030、EUC-KR、Big5 に切り替える事ができます。ご使用の環境に合わせて変更してください。

ファームウェア

CT-S601/651/801/851 系モデルにおいて本 SDK を正常に動作させるには、プリンターのファームウェアバージョンが下記の条件である必要があります。下記プリンターよりも古いプリンターをご使用の際は、ファームウェアをバージョンアップする必要があります。

機種	ファームウェアバージョン
CT-S601	DL00-2000 以降
CT-S651	DM00-2000 以降
CT-S801	DH00-2000 以降
CT-S851	DK00-2000 以降

1.5. 対象モデル(周辺機器)

本 SDK の周辺機器の対象モデルは以下の通りです。

各モデルの機能詳細については周辺機器の取扱説明書をご参照ください。

周辺機器の制御には、USB ホスト機能付きの Network I/F、Bluetooth I/F または Lightning I/F が必要です。

ラインディスプレイ

対象モデル	インターフェース	機能
DSP01-LT	USB	TFT 型ラインディスプレイ
DSP02-LS	USB	STN 型ラインディスプレイ

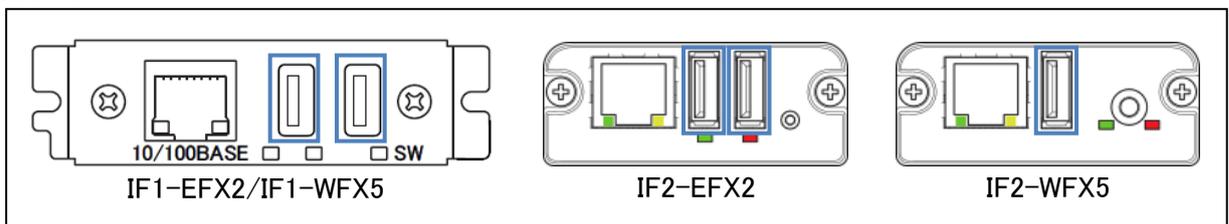
バーコードスキャナー

対象モデル	インターフェース	機能
SCN01-Z1D	USB	1 次元バーコードスキャナー
SCN02-Z2D	USB	2 次元バーコードスキャナー
BC-NL3000U	USB	2 次元バーコードスキャナー

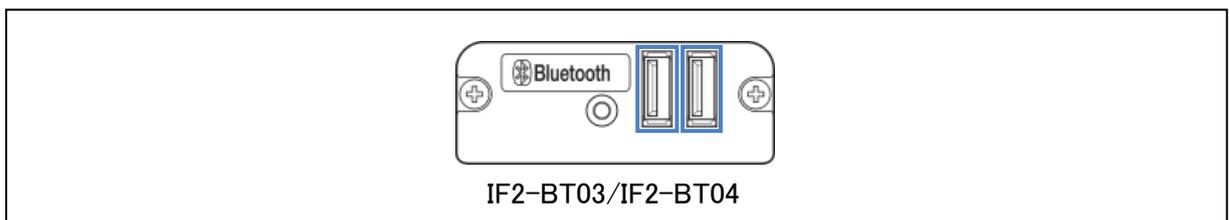
プリンターへの接続について

対象周辺機器の接続は、プリンター電源を一旦 OFF にしてから、下図に示す LAN インターフェース、または Bluetooth インターフェースの USB 端子に接続してご利用ください。その後、プリンター電源を ON にしてから、対象周辺機器が利用可能になるまで、安定動作のため、周辺機器の制御開始処理を 30 秒ほど待機してから実行してください。

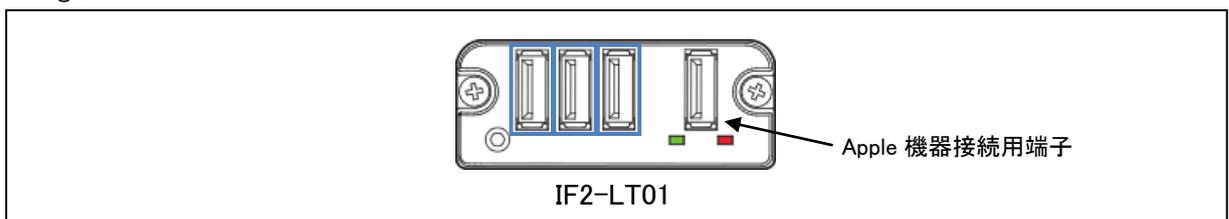
Network I/F



Bluetooth I/F



Lightning I/F



以下は、周辺機器接続に関しての、してはいけない「禁止」内容になります。

禁止事項

- 対応周辺機器以外(USB ハブやスマートフォン等)をインターフェースの USB 端子に接続
- プリンター電源が ON のまま、インターフェースの USB 端子から周辺機器のケーブルを挿抜
- 同種の周辺機器をインターフェースの USB 端子に複数接続 (例:ディスプレイを 2 台接続)
- Lightning I/F の周辺機器用 USB 端子に iPad/iPod/iPhone 接続

もし、上記事項を実施された場合、プリンターや接続周辺機器について、誤動作を招く原因や、最悪、故障の原因となりますので、おやめください。

LAN インターフェース設定について

LAN インターフェースにラインディスプレイおよびバーコードスキャナーを使用する場合、サービスに関する設定を変更する必要があります。基本的な操作につきましては、プリンターのインターフェースボード取り扱い説明書をご参照ください。

Web ブラウザーから各プリンターに接続して、以下の Service 画面を表示して下さい。プリンターが提供するサービスの設定を行います。

The screenshot shows the 'LAN board' configuration interface for 'CITIZEN SYSTEMS'. The 'Service' tab is selected, and the 'Media Converter' section is expanded. Two options are highlighted with a red box: 'VCOM Convert' and 'HID Scanner Convert', both with the 'Enable' radio button selected. Below this, there are sections for 'XML Print' and 'XML Device Control' with various numerical settings and dropdown menus.

Section	Option	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	
	Baud rate	9600	
XML Device Control /Line Display	Data	8 bit	

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

上記赤枠内を参照に、「VCOM Converter」と「HID Scanner Convert」の Enable を選択します。

その後、最下部までスクロールし「Submit」ボタンを押します。

最後に「Maintenance」タブの「Save&Reboot」ボタンを押し、「Yes」を選択し、プリンターから音が鳴ると設定完了です。

上記赤枠内の「Show configuration」にチェックを入れると「Media Converter Configuration / VCOM Convert」の設定画面が表示されますが、既に対応ディスプレイに対して適切値になっているので、通常利用では変更しないでください。

各設定値は、電源を切断しても値を保持します。工場初期設定(Factory Default)の処理が行われた時には、各設定値を初期値に設定します。

バーコードスキャナーの設定について

バーコードスキャナーを使用する場合は、以下の通り設定されている必要があります。

項目	値		説明
	Network または Bluetooth I/F	Lightning I/F	
Interface	USB HID Class	USB virtual COM	通信プロトコル
Keyboard	US Keyboard	US Keyboard	キーボード言語
Terminator	Enter	<CR><LF>	データのサフィックス

本 SDK でバーコードスキャナーをご使用になる際は、プリンターのインターフェースによりバーコードスキャナーの通信プロトコル設定を変更する必要があります。設定方法は以下の通りです。

・SCN01-Z1D

下記のバーコードを上から順番に全てスキャンし、設定を変更してください。

Network および Bluetooth I/F		Lightning I/F	
	設定開始		設定開始
	USB HID 設定		USB virtual COM 設定
	設定終了		設定終了

・SCN02-Z2D

下記のバーコードをスキャンし、設定を変更してください。

Network および Bluetooth I/F		Lightning I/F	
	USB HID 設定		USB virtual COM 設定

・BC-NL3000U

下記のバーコードを上から順番に全てスキャンし、設定を変更してください。

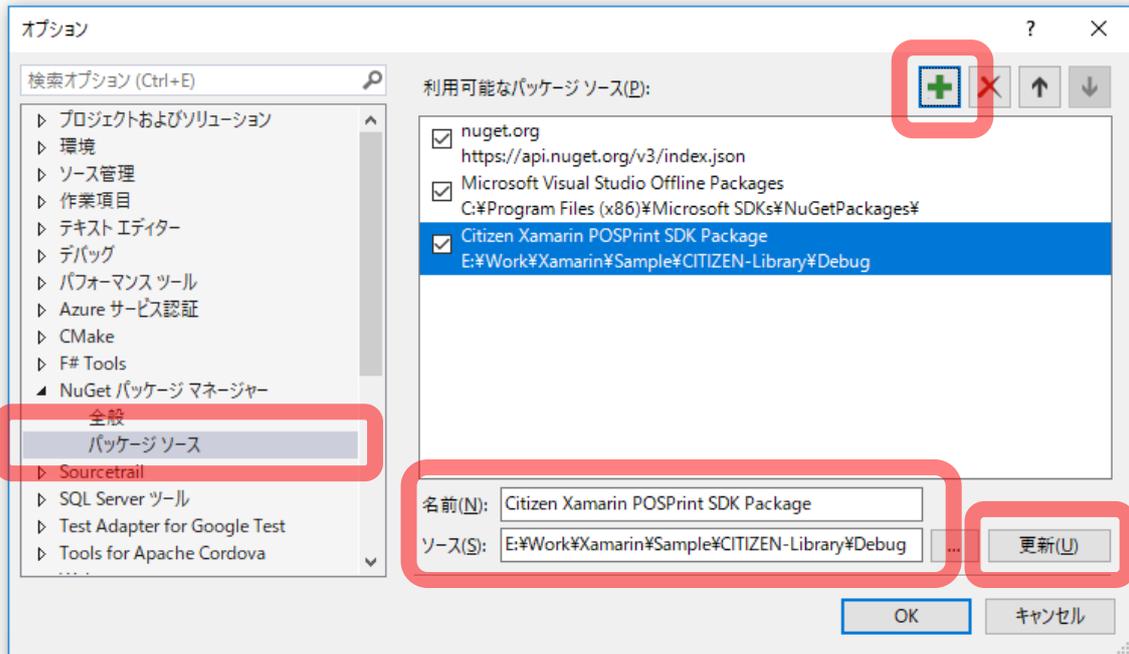
Network および Bluetooth I/F		Lightning I/F	
 0006010	設定開始	 0006010	設定開始
 1100080	USB HID 設定	 1100060	USB virtual COM 設定
 0006000	設定終了	 0006000	設定終了

1.6. 定義方法

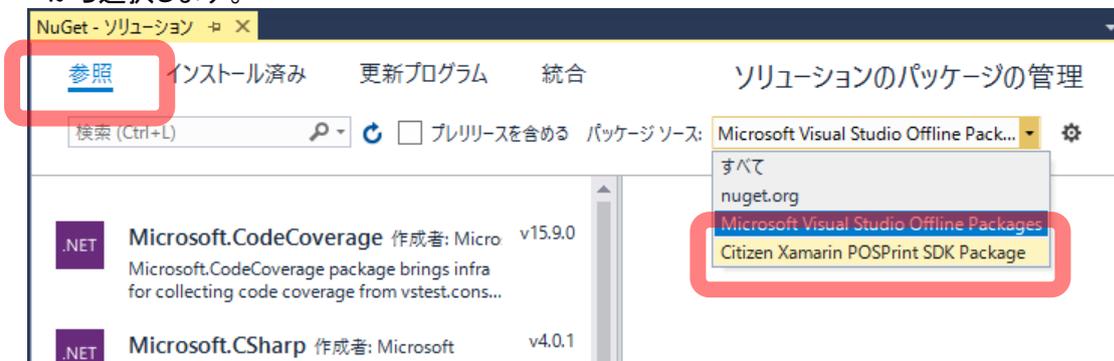
NuGet パッケージのインストール

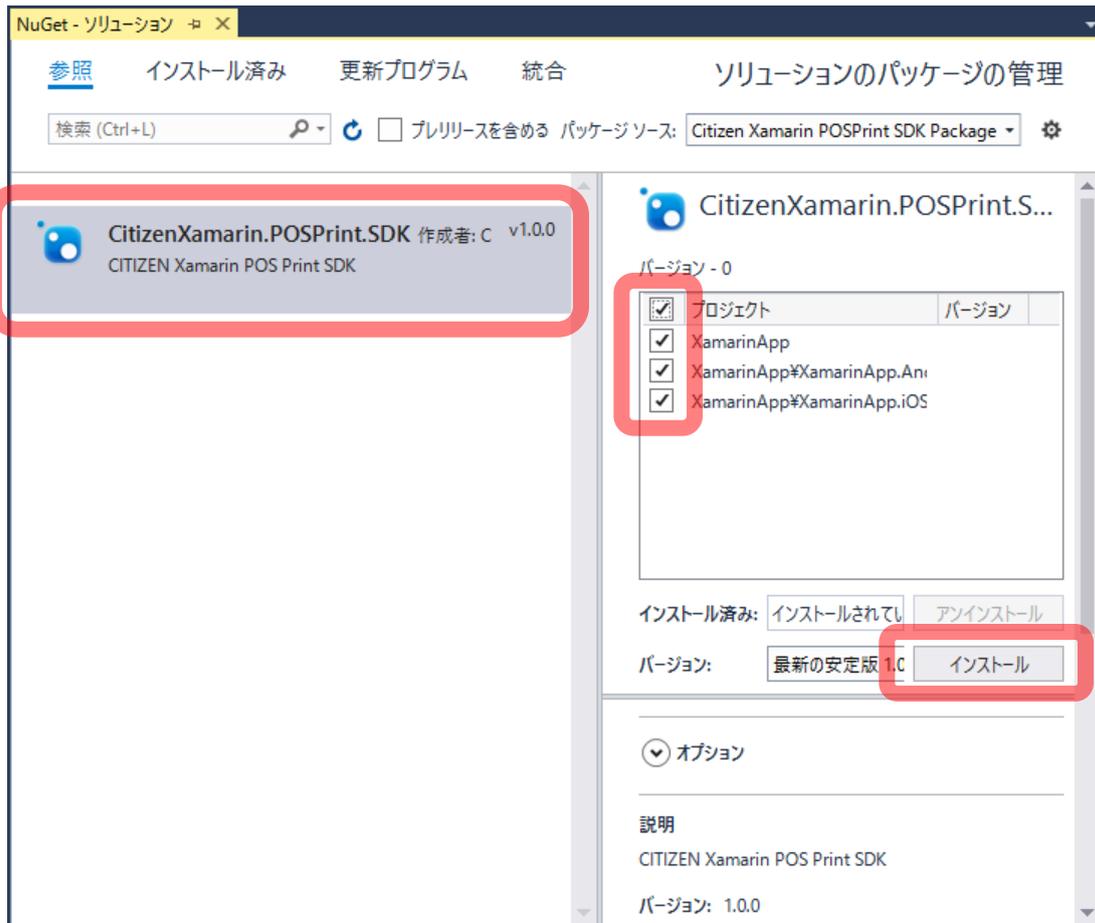
開発環境に NuGet パッケージをインストールします。

1. Visual Studio のメニューの「ツール」>「NuGet パッケージ マネージャー」>「パッケージ マネージャーの管理」を選択し、「オプション」ダイアログボックスを表示します。
2. 「パッケージ ソース」ノードを選択します。
3. ソースを追加するために “+” を選択します。

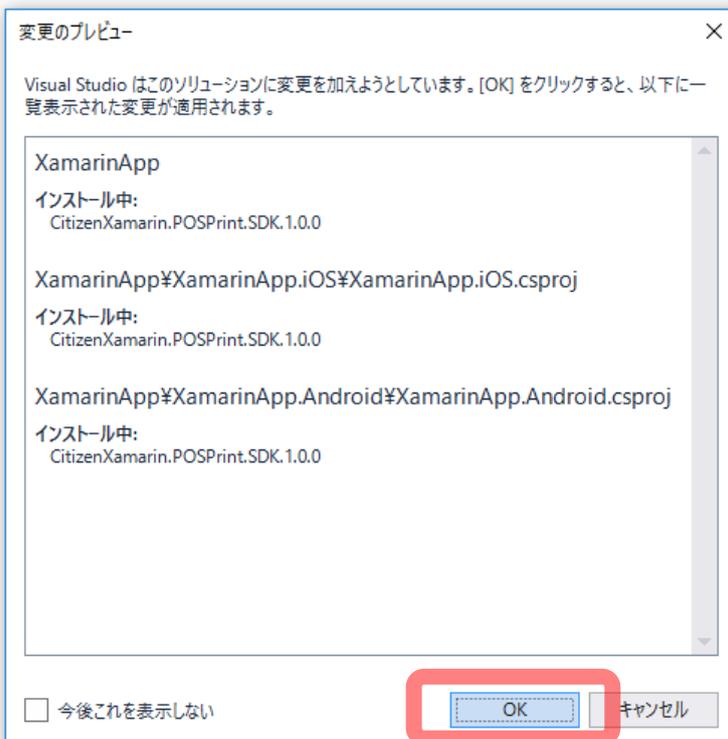


4. 「名前」を編集し、「ソース」に NuGet パッケージのパスを入力して、「更新」を選択します。「利用可能なパッケージソース」に追加されたことを確認してください。
5. Visual Studio のメニューの「ツール」>「NuGet パッケージ マネージャー」>「ソリューションの NuGet パッケージの管理」を選択し、「ソリューションのパッケージの管理」画面を表示します。
6. 「参照」タブを選択し、「利用可能なパッケージソース」で追加したパッケージ ソースを「パッケージ ソース」から選択します。





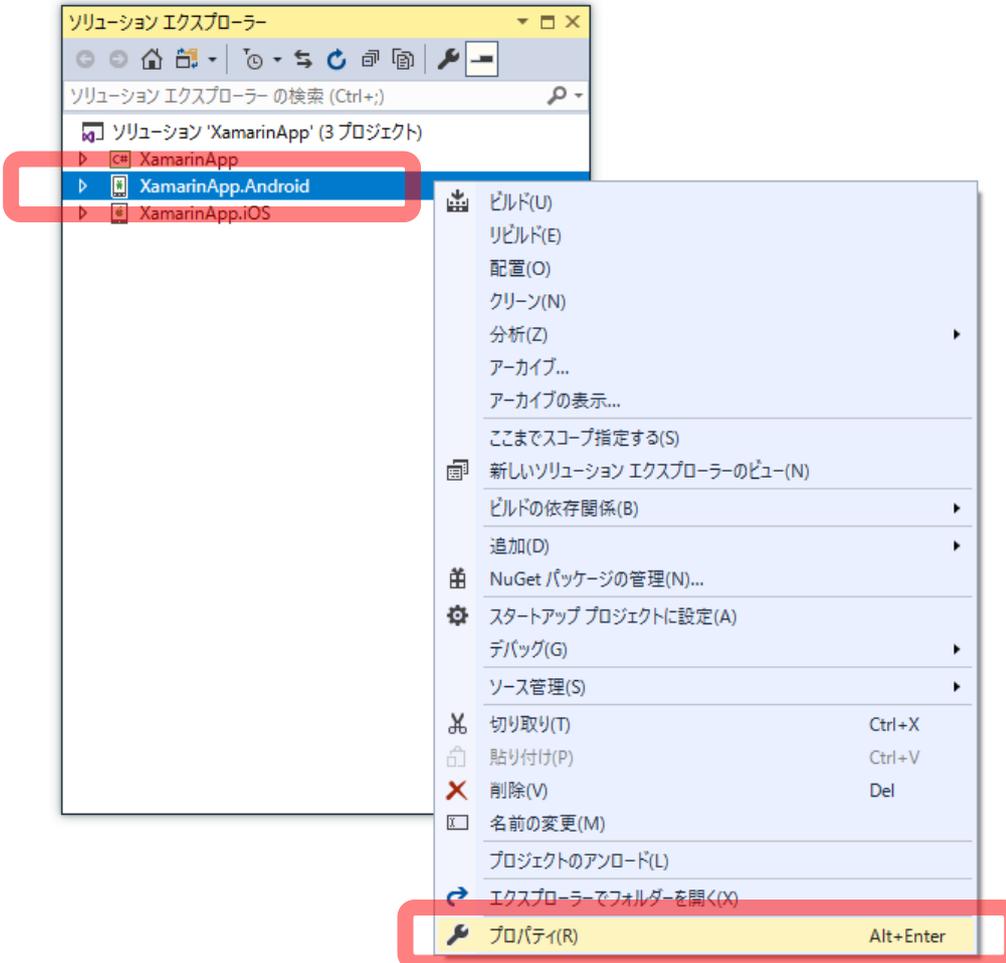
7. "CitizenXamarin.POSPrint.SDK"パッケージを選択します。
8. 「プロジェクト」をチェックし、「インストール」ボタンを有効にします。
9. 「インストール」ボタンを選択します。
10. 「OK」ボタンを選択します。



通信の許可設定

Android プロジェクトにアクセス許可を設定します。

1. Visual Studio のソリューション エクスプローラーで Android プロジェクトを右クリックし、「プロパティ」を選択します。



2. 「Android マニフェスト」を選択します。

XamarinApp.Android* ×

アプリケーション
Android マニフェスト*

Android オプション
Android パッケージへの署名
ビルド
ビルドイベント
参照パス

構成(C): なし プラットフォーム(M): なし

アプリケーション名:
XamarinApp.Android

パッケージ名:
com.companyname.XamarinApp

アプリケーションアイコン:
[選択]

アプリケーションテーマ:
[選択]

バージョン番号:
1

バージョン名:
1.0

インストール場所:
内部優先

最小 Android バージョン:
Android 5.0 (API レベル 21 - Lollipop)

ターゲット Android バージョン:
Android 9.0 (API レベル 28 - Pie)

必要なアクセス許可:

- ACCESS_CHECKIN_PROPERTIES
- ACCESS_COARSE_LOCATION
- ACCESS_FINE_LOCATION
- ACCESS_LOCATION_EXTRA_COMMANDS
- ACCESS_MOCK_LOCATION
- ACCESS_NETWORK_STATE
- ACCESS_NOTIFICATION_POLICY
- ACCESS_SURFACE_FLINGER
- ACCESS_WIFI_STATE

[詳細情報](#)

3. 「必要なアクセス許可」で以下の項目を選択してください。

Bluetooth

- BLUETOOTH
- BLUETOOTH_ADMIN
- ACCESS_COARSE_LOCATION

“ACCESS_COARSE_LOCATION”は、Android 6.0 以降で必要です。

Wi-Fi

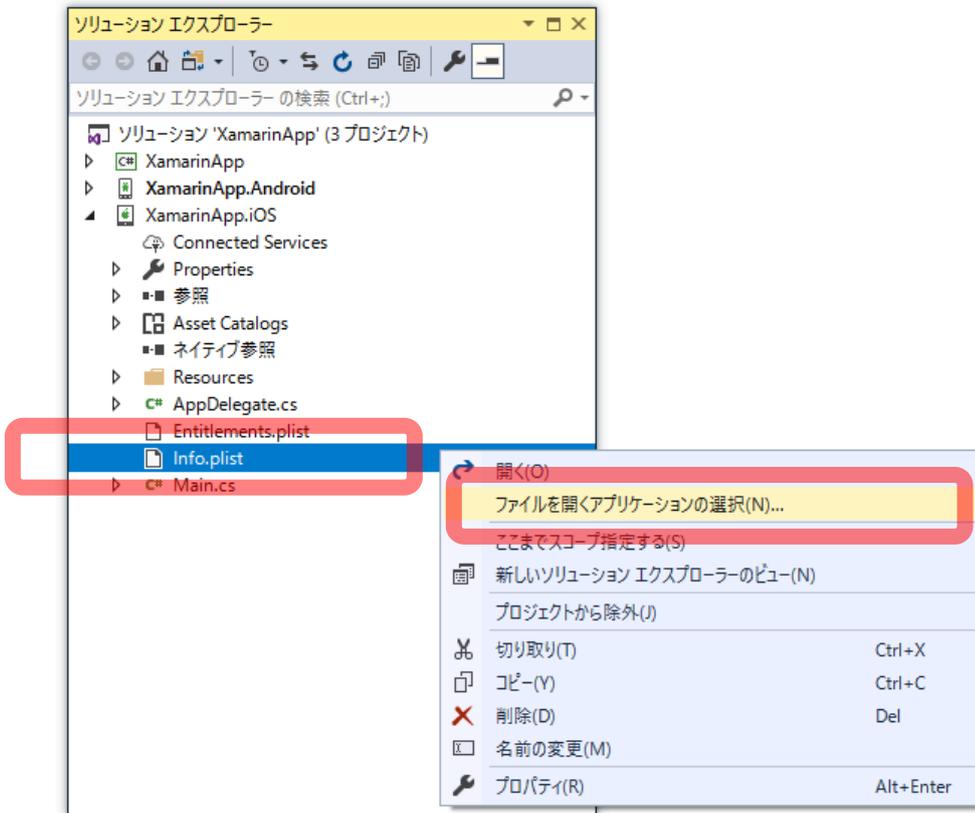
- INTERNET
- ACCESS_WIFI_STATE
- CHANGE_WIFI_MULTICAST_STATE

“ACCESS_WIFI_STATE”, “CHANGE_WIFI_MULTICAST_STATE”は、プリンタ検索に必要です。

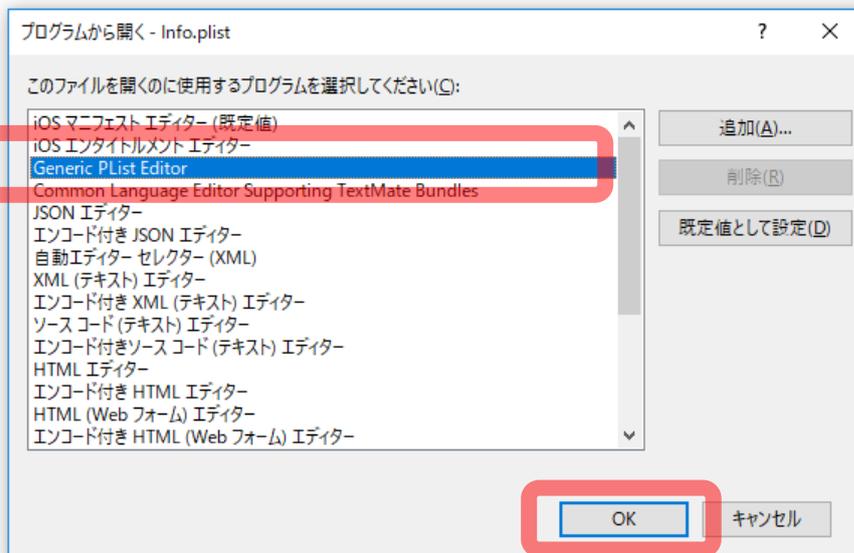
“CHANGE_WIFI_MULTICAST_STATE”は、端末によっては必要ない場合があります。

iOS プロジェクトに Protocol name を追加します。

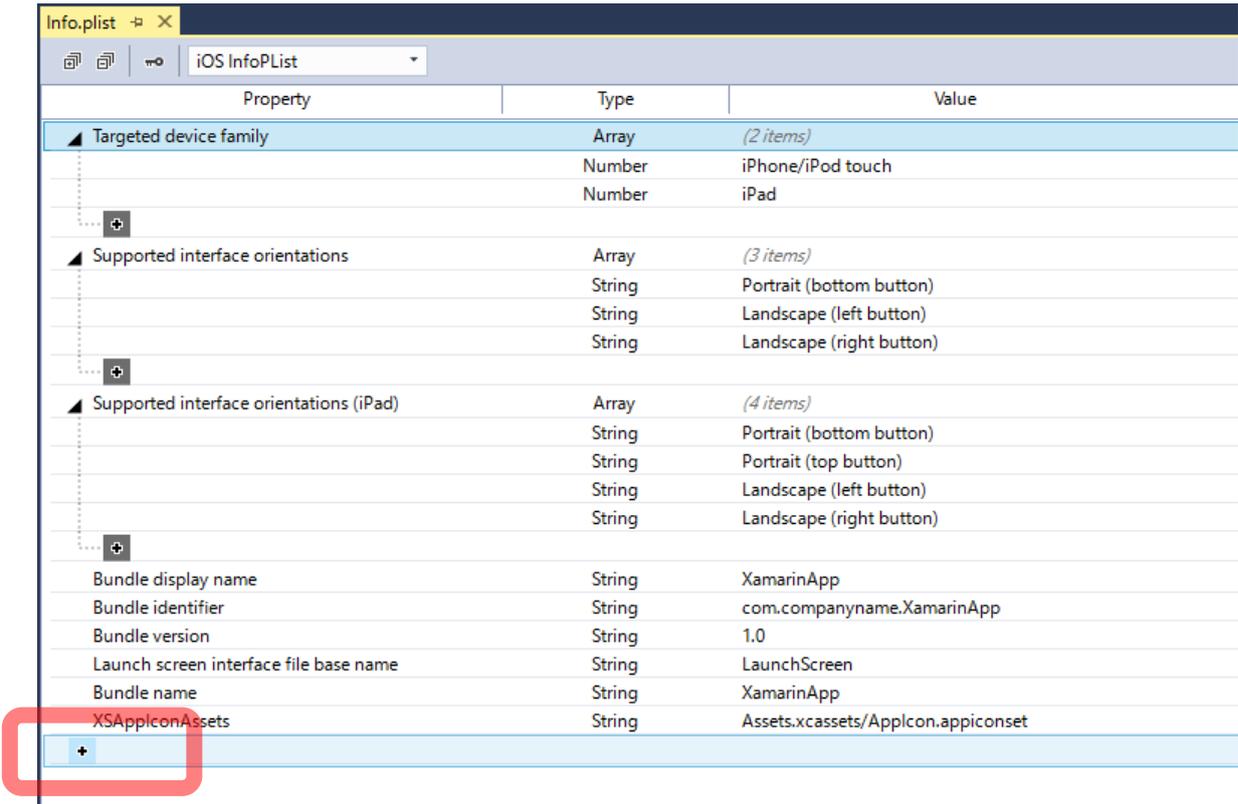
1. Visual Studio のソリューション エクスプローラーで iOS プロジェクトの「Info.plist」を右クリックし、「ファイルを開くアプリケーションの選択」を選択します。



2. 「Generic PList Editor」を選択します。

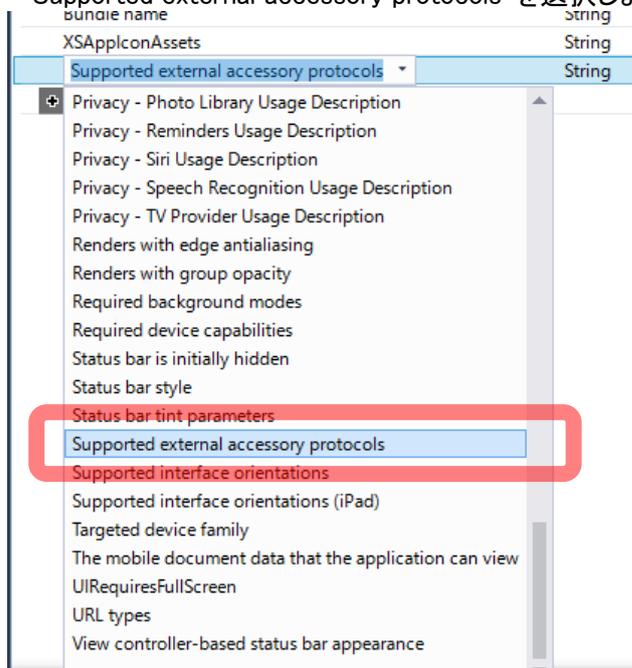


3. “+”を選択します。



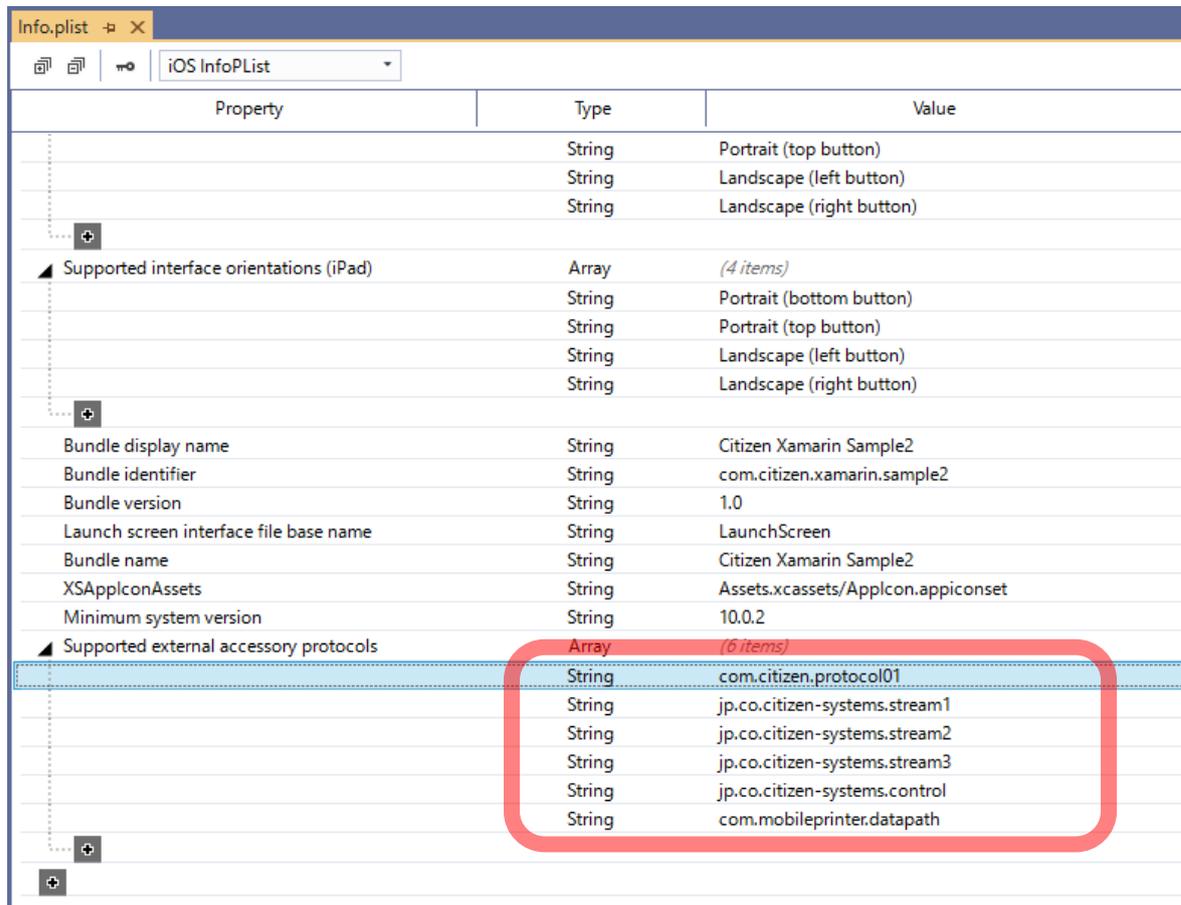
Property	Type	Value
Targeted device family	Array (2 items)	
	Number	iPhone/iPod touch
	Number	iPad
Supported interface orientations	Array (3 items)	
	String	Portrait (bottom button)
	String	Landscape (left button)
	String	Landscape (right button)
Supported interface orientations (iPad)	Array (4 items)	
	String	Portrait (bottom button)
	String	Portrait (top button)
	String	Landscape (left button)
	String	Landscape (right button)
Bundle display name	String	XamarinApp
Bundle identifier	String	com.companyname.XamarinApp
Bundle version	String	1.0
Launch screen interface file base name	String	LaunchScreen
Bundle name	String	XamarinApp
XSAppIconAssets	String	Assets.xcassets/AppIcon.appiconset

4. “Supported external accessory protocols”を選択します。



Property	Type
bundle name	String
XSAppIconAssets	String
Supported external accessory protocols	String
Privacy - Photo Library Usage Description	
Privacy - Reminders Usage Description	
Privacy - Siri Usage Description	
Privacy - Speech Recognition Usage Description	
Privacy - TV Provider Usage Description	
Renders with edge antialiasing	
Renders with group opacity	
Required background modes	
Required device capabilities	
Status bar is initially hidden	
Status bar style	
Status bar tint parameters	
Supported external accessory protocols	
Supported interface orientations	
Supported interface orientations (iPad)	
Targeted device family	
The mobile document data that the application can view	
UIRequiresFullScreen	
URL types	
View controller-based status bar appearance	

5. “com.citizen.protocol01”, “com.mobileprinter.datapath”を入力します。
 更に Lightning I/F で周辺機器を使用する場合は、“jp.co.citizen-systems.stream1”, “jp.co.citizen-systems.stream2”, “jp.co.citizen-systems.stream3”, “jp.co.citizen-systems.control”を入力します。



Property	Type	Value
	String	Portrait (top button)
	String	Landscape (left button)
	String	Landscape (right button)
+		
Supported interface orientations (iPad)	Array	(4 items)
	String	Portrait (bottom button)
	String	Portrait (top button)
	String	Landscape (left button)
	String	Landscape (right button)
+		
Bundle display name	String	Citizen Xamarin Sample2
Bundle identifier	String	com.citizen.xamarin.sample2
Bundle version	String	1.0
Launch screen interface file base name	String	LaunchScreen
Bundle name	String	Citizen Xamarin Sample2
XSApplconAssets	String	Assets.xcassets/AppIcon.appiconset
Minimum system version	String	10.0.2
Supported external accessory protocols	Array	(6 items)
	String	com.citizen.protocol01
	String	jp.co.citizen-systems.stream1
	String	jp.co.citizen-systems.stream2
	String	jp.co.citizen-systems.stream3
	String	jp.co.citizen-systems.control
	String	com.mobileprinter.datapath
+		
+		

名前空間の定義

本 SDK を使用する場合は、プログラムの先頭で名前空間「Plugin.CSJPOSlib」を定義してください。

```
using Plugin.CSJPOSlib;
```

2. プリンター制御

2.1. プログラム構造

本 SDK を使用する場合のプログラム構造は、以下の通りです。

```

// インスタンス生成
IESCPOSPrinter posPtr = CrossCSJPOSLib.CreateESCPOSPrinter();

// プリンターへ接続
int result = printer.Connect(ESCPOSConst.CMP_PORT_WiFi, "192.168.0.10");

if (ESCPOSConst.CMP_SUCCESS == result)
{
    // エンコード設定
    printer.SetEncoding("Shift_JIS");

    // 一括処理開始設定
    printer.TransactionPrint(ESCPOSConst.CMP_TP_TRANSACTION);

    // テキスト印刷
    printer.PrintText("Citizen_POS_sample1_CS¥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);

    // QRCode 印刷
    printer.PrintQRCode("http://www.citizen-systems.co.jp/", 6,
        ESCPOSConst.CMP_QRCODE_EC_LEVEL_L,
        ESCPOSConst.CMP_ALIGNMENT_RIGHT);

    // カット位置紙送り後、パーシャルカット
    printer.CutPaper(ESCPOSConst.CMP_CUT_PARTIAL_PREFEED);

    // 一括処理印刷指定
    result = printer.TransactionPrint(ESCPOSConst.CMP_TP_NORMAL);

    // 切断処理
    printer.Disconnect();

    if (ESCPOSConst.CMP_SUCCESS != result)
    {
        // 印刷処理エラー
        DisplayAlert("Citizen_POS_sample1", "Transaction Error : " +
            result.ToString(), "OK");
    }
}
else
{
    // 接続エラー
    DisplayAlert("Citizen_POS_sample1", "Connect Error : " +
        result.ToString(), "OK");
}

```

クラス定義

接続処理

印刷処理

切断処理

2.2. 機能一覧

本 SDK は以下の機能を提供します。

メソッド一覧

No	機能	詳細
1	クラス生成 (コンストラクタ)	コンストラクタです。
2	コンテキスト設定処理 (SetContext メソッド)	コンテキストを設定します。(Android のみ)
3	プリンター接続処理 (Connect メソッド)	プリンターと接続します。
4	プリンター切断処理 (Disconnect メソッド)	プリンターとの接続を切断します。
5	エンコード設定処理 (SetEncoding メソッド)	エンコードを設定します。
6	プリンター状態確認処理 (PrinterCheck メソッド)	ステータスチェックコマンドを送信します。
7	プリンター状態取得処理 (Status メソッド)	プリンターのステータスを取得します。
8	印字処理 (PrintText メソッド)	テキストデータを印刷します。
9	空白埋め込み印字処理 (PrintPaddingText メソッド)	空白埋め込みテキストデータを印刷します。
10	Local フォント印字処理 (PrintTextLocalFont メソッド)	端末のフォントを使用してテキストデータを印刷します。
11	ビットマップ印刷処理 (PrintBitmap メソッド)	画像ファイル[BMP/JPG/PNG/GIF ファイル]を印刷します。
12	NV ビットマップ登録処理 (SetNVBitmap メソッド)	ビットマップ画像をフラッシュメモリへ登録します。
13	NV 登録ビットマップ印刷処理 (PrintNVBitmap メソッド)	フラッシュメモリに保存されたビットマップ画像を印刷します。
14	バーコード印刷処理 (PrintBarCode メソッド)	一次元バーコードを印刷します。
15	PDF-417 印刷処理 (PrintPDF417 メソッド)	PDF417 バーコードを印刷します。
16	QR コード印刷処理 (PrintQRCode メソッド)	QRCode バーコードを印刷します。
17	2次元 GS1DataBar 印刷処理 (PrintGS1DataBarStacked メソッド)	2次元 GS1DataBar バーコードを印刷します。
18	用紙カット処理 (CutPaper メソッド)	用紙をカットします。
19	ドット単位紙送り処理 (UnitFeed メソッド)	ドット単位で紙送りします。
20	マーク紙送り処理 (MarkFeed メソッド)	ラベル/ブラックマーク紙用をサポートします。
21	ドロワー開処理 (OpenDrawer メソッド)	キャッシュドロワーを開けるコマンドを送信します。
22	一括処理開始/終了処理 (TransactionPrint メソッド)	一括処理モードを開始/終了します。
23	回転印刷処理 (RotatePrint メソッド)	回転方向モード(180度)を開始/終了します。
24	ページモード開始/終了処理 (PageModePrint メソッド)	ページモードを開始/終了します。
25	ページモード印刷領域消去処理 (ClearPrintArea メソッド)	ページモード印刷領域上の印刷データを消去します。
26	出力データクリア処理 (ClearOutput メソッド)	処理中のデータおよびプリンターのバッファをクリアします。

27	データ出力処理 (PrintData メソッド)	データをそのままプリンターに送信します。
28	OPOS 形式印刷処理 (PrintNormal メソッド)	OPOS エスケープシーケンスを使用してテキストを印刷します。
29	透かし印刷開始／終了処理 (WatermarkPrint メソッド)	透かし印刷を開始／終了します。
30	プリンター検索 (SearchCitizenPrinter メソッド)	プリンターを検索しプリンター情報のリストを取得します。
31	プリンター検索 (SearchESCPOSPrinter メソッド)	プリンターを検索しアドレスのリストを取得します。
32	ログ設定 (SetLog メソッド)	ログ機能を設定します。(Android のみ)
33	印刷完了通知タイムアウト設定 (SetPrintCompletedTimeout メソッド)	印刷完了通知を確認するタイムアウトを設定します。
34	バージョンコード取得 (GetVersionCode メソッド)	SDK のバージョン番号を数値で取得します。
35	バージョン文字列取得 (GetVersionName メソッド)	SDK のバージョン番号を文字列で取得します。

プロパティ一覧

No	機能	属性	詳細
1	ページモード領域取得 (PageModeArea プロパティ)	R	使用可能なページモード領域を示します。
2	ページモード印刷領域設定/取得 (PageModePrintArea プロパティ)	R/W	ページモード印刷領域を示します。
3	ページモード印刷方向設定/取得 (PageModePrintDirection プロパティ)	R/W	ページモード印刷領域内の印刷方向を示します。
4	印刷開始水平方向オフセット値設定/取得 (PageModeHorizontalPosition プロパティ)	R/W	ページモード印刷領域内の印刷開始位置の水平方向オフセット値を示します。
5	印刷開始垂直方向オフセット値設定/取得 (PageModeVerticalPosition プロパティ)	R/W	ページモード印刷領域内の印刷開始位置の垂直方向オフセット値を示します。
6	行間設定/取得 (RecLineSpacing プロパティ)	R/W	通常文字の印刷行の高さを示します。
7	マッピングモード設定/取得 (MapMode プロパティ)	R/W	プリンターのマッピングモード(長さの単位)を示します。

2.3. ライブラリインターフェース

本 SDK のインターフェースを以下に示します。

2.3.1. 戻り値

以降に示すメソッドは、下記の値を返します。

戻り値	説明
CMP_SUCCESS (0)	正常終了
CMP_E_CONNECTED (1001)	プリンターへ既に接続済みです。
CMP_E_DISCONNECT (1002)	プリンターへ接続していません。
CMP_E_NOTCONNECT (1003)	プリンターへ接続できませんでした。
CMP_E_CONNECT_NOTFOUND (1004)	プリンター接続後の対応機種確認に失敗しました。
CMP_E_CONNECT_OFFLINE (1005)	プリンター接続後のプリンター状態確認に失敗しました。
CMP_E_NOCONTEXT (1006)	コンテキストを指定していません。
CMP_E_BT_DISABLE (1007)	Bluetooth デバイス設定が無効です。
CMP_E_BT_NODEVICE (1008)	Bluetooth デバイスが見つかりませんでした。
CMP_E_ILLEGAL (1101)	サポートされていない処理または無効なパラメータ値です。
CMP_E_OFFLINE (1102)	プリンターがオフラインです。
CMP_E_NOEXIST (1103)	指定のファイルが存在しません。
CMP_E_FAILURE (1104)	要求された処理が実行できません。
CMP_E_TIMEOUT (1105)	所定の時間が経過してもプリンターからの応答がありません。
CMP_E_NO_LIST (1106)	プリンター検索にてプリンターが見つかりません。
CMP_EPTR_COVER_OPEN (1201)	プリンターのカバーが開いています。
CMP_EPTR_REC_EMPTY (1202)	用紙切れです。
CMP_EPTR_BADFORMAT (1203)	指定されたファイルの書式がサポートされていません。
CMP_EPTR_TOOBIG (1204)	指定されたビットマップのサイズが大きすぎます。

2.3.2. コンストラクタ

形式

IESCPOSPrinter

パラメータ

ありません。

説明

ライブラリのコンストラクタです。インスタンスを生成します。

戻り値

ありません。

使用例

```
IESCPOSPrinter printer = CrossCSJPOSLib.CreateESCPOSPrinter();
```

2.3.3. SetContext メソッド (Android のみ)

形式

```
int SetContext(object context )
```

パラメータ

パラメータの意味と設定可能な値は以下の通りです。

値	[IN/OUT]	意味	設定可能範囲
context	[IN]	コンテキスト (android.content.Context)	Context :

説明

このメソッドは、Android で USB 接続およびプリンター検索のために使用します。コンテキストを指定してください。USB 接続での [Connect メソッド](#) および [SearchCitizenPrinter メソッド](#)、[SearchESCPOSPrinter メソッド](#) 実行前に本メソッドを実行してください。

戻り値

成功時は CMP_SUCCESS(0) を返します。それ以外のエラーコードは「[2.3.1 戻り値](#)」を参照してください。

使用例

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

2.3.4. Connect メソッド

形式

- 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)

パラメータ

パラメータの意味と設定可能な値は以下の通りです。

値	[IN/OUT]	意味	設定可能範囲
connectType	[IN]	接続タイプ	CMP_PORT_WiFi CMP_PORT_Bluetooth CMP_PORT_Bluetooth_Insecure (Android のみ) CMP_PORT_USB
addr	[IN]	接続先の IP アドレス、 または Bluetooth デバイスアドレス または Bluetooth デバイス名 またはシリアル番号	WiFi : 0.0.0.0~255.255.255.255 Bluetooth : 00:00:00:00:00:00~FF:FF:FF:FF:FF:FF デバイス名 (自動検出) USB : 空白またはシリアル番号(iOS のみ)
port	[IN]	接続先ポート番号	
timeout	[IN]	タイムアウト (msec)	

説明

このメソッドは、プリンターと接続するために使用します。プリンターの接続タイプとアドレスを指定してください。Bluetooth デバイスアドレスの英字は、大文字で指定してください。

Bluetooth デバイス名が指定された場合は、一致するペアリング済みのデバイスを自動検出します。

アドレスが省略された場合は、ペアリング済みの対応機種デバイスを自動検出します。

Android 2.3.3 以降に用意される Bluetooth デバイスの Insecure 通信を使用する場合は、接続タイプに CMP_PORT_Bluetooth_Insecure を指定してください。

Android で USB デバイスを使用する場合は、メソッド実行前に [SetContext メソッド](#) を実行する必要があります。

USB 接続の場合、アドレスに空白を指定した場合は自動的に接続されます。iOS の場合、プリンターのシリアル番号を指定して特定のプリンターへ接続する事も可能です。

接続先ポート番号は、接続タイプに CMP_PORT_WiFi を指定した場合のみ有効です。省略された場合は、9100 番で接続します。

タイムアウトは、プリンターへの接続の最大時間(ミリ秒単位)を指定します。接続タイプに CMP_PORT_USB を指定した場合は無効です。省略された場合は、WiFi 使用時に 4000 ミリ秒、Bluetooth 使用時に 8000 ミリ秒で接続します。

プリンターと接続した際に、プリンターのステータスと対応機種を同時に確認します。

プリンターとの通信が不要になった場合は、必ず [Disconnect メソッド](#) を実行し、プリンターとの接続を切断してください。切断しなかった場合は、次の接続がエラーとなります。

【注意】

初回、USB で接続する際、Android 端末画面に USB デバイスへのアクセス許可を求めるダイアログが表示されるので、OK をタップしてください。



USB デバイスへのアクセス許可ダイアログ（一例）

戻り値

成功時は CMP_SUCCESS(0) を返します。失敗時は下記のエラーコードの説明を確認してください。それ以外のエラーコードは「[2.3.1 戻り値](#)」を参照してください。

エラーコード	説明
CMP_E_NOTCONNECT (1003)	プリンターへ接続できませんでした。 ①プリンターが未接続 ②プリンターの電源が入っていない ③インターフェースポートのハンドルを取得できない
CMP_E_CONNECT_NOTFOUND (1004)	プリンター接続後の対応機種確認に失敗しました。 ①対応機種でない
CMP_E_CONNECT_OFFLINE (1005)	プリンター接続後のプリンター状態確認に失敗しました。プリンターに次のエラーが発生している可能性があります。 ①カバーが開いている ②用紙が無い ③紙ジャム等によりオートカッターエラーが発生 ④回路故障等による復帰不可能エラーが発生
CMP_E_NOCONTEXT (1006)	コンテキストが指定されていません。(Android のみ)
CMP_E_BT_DISABLE (1007)	Bluetooth デバイス設定が無効です。
CMP_E_BT_NODEVICE (1008)	Bluetooth デバイスが見つかりませんでした。

使用例

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

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

printer.Connect( ESCPOSConst.CMP_PORT_USB, null );
```

2.3.5. Disconnect メソッド

形式

```
int Disconnect ()
```

パラメータ

ありません。

説明

このメソッドは、プリンターとの接続を切断するために使用します。

印刷の終了、あるいは、何らかのエラーが発生した場合は、本メソッドを実行して接続を切断してください。

戻り値

成功時は CMP_SUCCESS(0) を返します。それ以外のエラーコードは「[2.3.1 戻り値](#)」を参照してください。

使用例

```
printer.Disconnect ();
```

2.3.6. SetEncoding メソッド

形式

int SetEncoding (string charset)

パラメータ

パラメータの意味と設定可能な値は以下の通りです。

値	[IN/OUT]	意味	設定可能範囲
charset	[IN]	文字セット名	サポートされる文字セット名は、動作環境により異なります。

説明

このメソッドは、プリンター送信データのエンコードを設定するために使用します。

インスタンスを生成時に OS のデフォルト文字セットに初期化します。

エンコードはプリンターのメモリスイッチの設定に合わせて設定してください(「[1.5 対応機種](#)」参照)。

本 SDK は、UTF-8 でエンコードされた文字列の印刷をサポートします。詳しくは、「[2.4.2 UTF-8 エンコード文字列の印刷について](#)」を参照してください。

戻り値

成功時は CMP_SUCCESS(0) を返します。それ以外のエラーコードは「[2.3.1 戻り値](#)」を参照してください。

使用例

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

```
printer.SetEncoding( "GB18030" );
```

```
printer.SetEncoding( "EUC-KR" );
```

```
printer.SetEncoding( "Big5" );
```

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

2.3.7. PrinterCheck メソッド

形式

```
int PrinterCheck ()
```

パラメータ

ありません。

説明

このメソッドは、プリンターのステータス取得コマンドを送信するために使用します。
本メソッドの実行結果が成功の場合は、[Status メソッド](#)でプリンターのステータスを取得できます。
本メソッドの実行結果が失敗の場合は、通信異常やデバイスの異常が発生した可能性があります。この場合、[Disconnect メソッド](#)および [Connect メソッド](#)を使用して再接続してください。

接続後に時間を空けて印刷する場合は、必ず事前に本メソッドと [Status メソッド](#)を実行してプリンターのステータスを確認してください。

ネットワーク接続の場合、長時間放置すると自動的に切断されます。接続を保持する場合は、定期的に本メソッドを実行してください。

戻り値

成功時は CMP_SUCCESS(0) を返します。それ以外のエラーコードは「[2.3.1 戻り値](#)」を参照してください。

使用例

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

2.3.8. Status メソッド

形式

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

パラメータ

パラメータの意味と設定可能な値は以下の通りです。

値	[IN/OUT]	意味	設定可能範囲
Typ	[IN]	ステータス種類	CMP_STS_DRAWER_LEVEL_H CMP_STS_PAPER_NEAREMPTY CMP_STS_BATTERY_LOW CMP_STS_COVER_OPEN CMP_STS_PAPER_EMPTY CMP_STS_MSR_READ CMP_STS_PRINTEROFF

説明

このメソッドは、[PrinterCheck メソッド](#)によって取得されたプリンターのステータスを確認するために使用します。メソッドの実行前に、[PrinterCheck メソッド](#)を実行する必要があります。引き数が指定されなかった場合は、プリンターのエラーを示すステータス(CMP_STS_COVER_OPEN、CMP_STS_PAPER_EMPTY、CMP_STS_PRINTEROFF)の論理和を返します。ステータス種類が指定された場合は、該当するステータスを返します。ステータス種類は組み合わせて指定できます。組み合わせる場合は論理和を指定してください。

戻り値

下記のステータスコードを返します。

ステータスコード	説明
CMP_STS_NORMAL (0)	プリンターは正常です。
CMP_STS_DRAWER_LEVEL_H (2)	ドロワーキックコネクタ3番ピンの状態が H です。(type 指定時のみ) [iOS は未サポート]
CMP_STS_PAPER_NEAREMPTY (4)	ニアエンプティです。(type 指定時のみ) [iOS は未サポート]
CMP_STS_BATTERY_LOW (8)	プリンターバッテリー容量が低下しました。(CMP-20/30/20II/30II のみ) [iOS は未サポート]
CMP_STS_COVER_OPEN (16)	プリンタカバーが開いています。
CMP_STS_PAPER_EMPTY (32)	用紙がありません。
CMP_STS_MSR_READ (64)	MSR 読み取りモード状態です。(CMP-20/30/20II/30II のみ) [iOS は未サポート]
CMP_STS_PRINTEROFF (128)	プリンターはオフラインです。

使用例

```
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 (Android Only)
    }
} else {
    if ( (ESCPOSConst.CMP_STS_COVER_OPEN & status) > 0 ) {
        // Cover Open
    }
}
```

```
    }
    if ( (ESCPOSConst.CMP_STS_PAPER_EMPTY & status) > 0 ) {
        // Paper Empty
    }
    if ( (ESCPOSConst.CMP_STS_PRINTEROFF & status) > 0 ) {
        // Printer Offline
    }
}

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

2.3.9. PrintText メソッド

形式

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

パラメータ

パラメータの意味と設定可能な値は以下の通りです。

値	[IN/OUT]	意味	設定可能範囲
data	[IN]	テキストデータ	
alignment	[IN]	テキスト配置	CMP_ALIGNMENT_LEFT: 左揃え CMP_ALIGNMENT_CENTER: 中央揃え CMP_ALIGNMENT_RIGHT: 右揃え
attribute	[IN]	テキスト属性	CMP_FNT_DEFAULT: 標準フォント CMP_FNT_FONTB: フォント B CMP_FNT_FONTC: フォント C CMP_FNT_BOLD: 太字 CMP_FNT_REVERSE: 反転 CMP_FNT_UNDERLINE: 下線
textSize	[IN]	テキストサイズ	CMP_TXT_1WIDTH: 幅 1 倍 CMP_TXT_2WIDTH: 幅 2 倍 CMP_TXT_3WIDTH: 幅 3 倍 CMP_TXT_4WIDTH: 幅 4 倍 CMP_TXT_5WIDTH: 幅 5 倍 CMP_TXT_6WIDTH: 幅 6 倍 CMP_TXT_7WIDTH: 幅 7 倍 CMP_TXT_8WIDTH: 幅 8 倍 CMP_TXT_1HEIGHT: 高さ 1 倍 CMP_TXT_2HEIGHT: 高さ 2 倍 CMP_TXT_3HEIGHT: 高さ 3 倍 CMP_TXT_4HEIGHT: 高さ 4 倍 CMP_TXT_5HEIGHT: 高さ 5 倍 CMP_TXT_6HEIGHT: 高さ 6 倍 CMP_TXT_7HEIGHT: 高さ 7 倍 CMP_TXT_8HEIGHT: 高さ 8 倍

説明

このメソッドは、テキストの配置、属性、サイズを指定して、テキストを印刷するために使用します。

テキスト属性は、フォント B、フォント C、太字、反転、下線を組み合わせて指定可能です。組み合わせる場合は論理和を指定してください。

テキストサイズは、幅と高さを組み合わせて指定可能です。組み合わせる場合は論理和を指定してください。

戻り値

成功時は CMP_SUCCESS(0) を返します。それ以外のエラーコードは「[2.3.1 戻り値](#)」を参照してください。

使用例

```
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 );
```

2.3.10. PrintPaddingText メソッド

形式

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

パラメータ

パラメータの意味と設定可能な値は以下の通りです。

値	[IN/OUT]	意味	設定可能範囲
data	[IN]	テキストデータ	
attribute	[IN]	テキスト属性	CMP_FNT_DEFAULT: 標準フォント CMP_FNT_FONTB: フォント B CMP_FNT_FONTC: フォント C CMP_FNT_BOLD: 太字 CMP_FNT_REVERSE: 反転 CMP_FNT_UNDERLINE: 下線
textSize	[IN]	テキストサイズ	CMP_TXT_1WIDTH: 幅 1 倍 CMP_TXT_2WIDTH: 幅 2 倍 CMP_TXT_3WIDTH: 幅 3 倍 CMP_TXT_4WIDTH: 幅 4 倍 CMP_TXT_5WIDTH: 幅 5 倍 CMP_TXT_6WIDTH: 幅 6 倍 CMP_TXT_7WIDTH: 幅 7 倍 CMP_TXT_8WIDTH: 幅 8 倍 CMP_TXT_1HEIGHT: 高さ 1 倍 CMP_TXT_2HEIGHT: 高さ 2 倍 CMP_TXT_3HEIGHT: 高さ 3 倍 CMP_TXT_4HEIGHT: 高さ 4 倍 CMP_TXT_5HEIGHT: 高さ 5 倍 CMP_TXT_6HEIGHT: 高さ 6 倍 CMP_TXT_7HEIGHT: 高さ 7 倍 CMP_TXT_8HEIGHT: 高さ 8 倍
length	[IN]	半角文字相当の長さ	1～
side	[IN]	空白埋め込みサイド	CMP_SIDE_RIGHT: テキストデータの右側 CMP_SIDE_LEFT: テキストデータの左側

説明

このメソッドは、テキストの属性、サイズ、半角文字相当の長さ、空白埋め込みサイドを指定して、空白が埋め込まれたテキストを印刷するために使用します。

テキストデータに、結合文字を使用する事はできません。

テキスト属性は、フォント B、フォント C、太字、反転、下線を組み合わせて指定可能です。組み合わせる場合は論理和を指定してください。

テキストサイズは、幅と高さを組み合わせて指定可能です。組み合わせる場合は論理和を指定してください。

戻り値

成功時は CMP_SUCCESS(0) を返します。それ以外のエラーコードは「[2.3.1 戻り値](#)」を参照してください。

使用例

```
int nameSize = 24;      // 商品名サイズ
int priceSize = 7;     // 価格サイズ

// 1行目
printer.PrintPaddingText( "Sandwich",
    ESCPOSConst.CMP_FNT_DEFAULT, ESCPOSConst.CMP_TXT_1WIDTH,
    nameSize, ESCPOSConst.CMP_SIDE_RIGHT);
printer.PrintPaddingText( "5.00",
    ESCPOSConst.CMP_FNT_DEFAULT, ESCPOSConst.CMP_TXT_1WIDTH,
    priceSize, ESCPOSConst.CMP_SIDE_LEFT);
printer.PrintNormal("\n");

// 2行目
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");

// 3行目
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");
```

2.3.11. PrintTextLocalFont メソッド

形式

```
int PrintTextLocalFont (string data, int alignment, object fontType, int point, int style,
                       int hRatio, int vRatio)
```

パラメータ

パラメータの意味と設定可能な値は以下の通りです。

値	[IN/OUT]	意味	設定可能範囲
data	[IN]	テキストデータ	
alignment	[IN]	テキスト配置	CMP_ALIGNMENT_LEFT: 左揃え CMP_ALIGNMENT_CENTER: 中央揃え CMP_ALIGNMENT_RIGHT: 右揃え
fontType	[IN]	フォントタイプ	Android: Typeface iOS: NSString
fontName	[IN]	フォント名	(iOS 用)
point	[IN]	フォントサイズ	ポイント
style	[IN]	フォントスタイル	CMP_FNT_DEFAULT: デフォルト CMP_FNT_BOLD: 太字 CMP_FNT_REVERSE: 反転 CMP_FNT_UNDERLINE: 下線 CMP_FNT_ITALIC: イタリック CMP_FNT_STRIKEOUT: 取り消し線
hRatio	[IN]	水平拡大率 [%]	1~1000
vRatio	[IN]	垂直拡大率 [%]	1~1000

説明

このメソッドは、テキストの配置、フォント、サイズ、スタイル、拡大率を指定して、端末にインストールされたフォントを使用してテキストを印刷するために使用します。

このメソッドを使用すると、入力された内容を画像化し、その画像データを印刷します。

フォントタイプは、環境に合わせて指定してください。null の場合は、デフォルトのフォントで印刷します。

フォントスタイルは、太字、反転、下線、イタリック、取り消し線を組み合わせて指定可能です。組み合わせる場合は論理和を指定してください。

戻り値

成功時は CMP_SUCCESS(0) を返します。それ以外のエラーコードは「[2.3.1 戻り値](#)」を参照してください。

使用例

```
printer.PrintTextLocalFont( "Print local font text data.\r\n",
                           ESCPOSConst.CMP_ALIGNMENT_CENTER,
                           null, 12,
                           ESCPOSConst.CMP_FNT_BOLD | ESCPOSConst.CMP_FNT_UNDERLINE,
                           100, 100 );
```

2.3.12. PrintBitmap メソッド

形式

- 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 (object bitmap, int alignment)
- 5) int PrintBitmap (object bitmap, int width, int alignment)
- 6) int PrintBitmap (object 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)

パラメータ

パラメータの意味と設定可能な値は以下の通りです。

値	[IN/OUT]	意味	設定可能範囲
fileName	[IN]	ビットマップファイル名	
bitmap	[IN]	ビットマップ型データ	Android は Bitmap データ型を、iOS は UIImage 型を指定します。
bytes	[IN]	ビットマップを保持するバイト配列	
width	[IN]	ビットマップ印刷幅	CMP_BM_ASIS: プリンターのドット当たり 1 ビットマップピクセルでビットマップを印刷します。 上記定数以外の 1 以上の値: ビットマップ幅を MapMode プロパティ で定義された単位 (デフォルトはドット単位) で指定します。
alignment	[IN]	ビットマップ配置位置	CMP_ALIGNMENT_LEFT: 左揃え CMP_ALIGNMENT_CENTER: 中央揃え CMP_ALIGNMENT_RIGHT: 右揃え 上記定数以外の 3 以上の値: ビットマップ印刷を開始する左からの距離を MapMode プロパティ で定義された単位 (デフォルトはドット単位) 指定します。
mode	[IN]	ビットマップモード	CMP_BM_MODE_HT_THRESHOLD: ハーフトーン しきい値 CMP_BM_MODE_HT_DITHER: ハーフトーン デイザー CMP_BM_MODE_CMD_RASTER: モノクロ ラスター コマンド出力 CMP_BM_MODE_CMD_BITIMAGE: モノクロ ビットイメージ コマンド出力 CMP_BM_MODE_CMD_GRAY16: グレースケール (16 階調) 出力 CMP_BM_MODE_CMD_GRAY16DOWNLOAD: グレースケール (16 階調) ダウンロードグラフィックスコマンド出力

説明

このメソッドは、ファイル名あるいはデータ、印刷幅、配置位置、モードを指定して、ビットマップを印刷するた

めに使用します。

印刷可能なビットマップ形式は、BMP/JPG/PNG/GIF です。

印刷幅が省略された場合は、CMP_BM_ASIS で印刷します。

モードは、ハーフトーンと出力方法を組み合わせて指定可能です。組み合わせる場合は、論理和を指定してください。モードが省略された場合は、CMP_BM_MODE_HT_THRESHOLD | CMP_BM_MODE_CMD_RASTER で印刷します。モードに関する詳細は、次の通りです。

ハーフトーン ハーフトーン処理方法を指定します。

設定値	説明
CMP_BM_MODE_HT_THRESHOLD	しきい値 文字の印刷に適します。
CMP_BM_MODE_HT_DITHER	ディザー グラフィックの印刷に適します。

出力 出力方法を指定します。

設定値	説明
CMP_BM_MODE_CMD_RASTER	モノクロ ラスター コマンド出力 小さなデータの印刷に適します。データを一括して出力するため、高さ制限(2,304ドット 約 28cm)があります。
CMP_BM_MODE_CMD_BITIMAGE	モノクロ ビットイメージ コマンド出力 大きなデータの印刷に適します。データを分割して出力するため、高さ制限はありません。
CMP_BM_MODE_CMD_GRAY16	グレースケール出力 CT-E601、CT-S251/255/257/601II/651II/801II/851II系で利用可能です。より綺麗なグラフィックの印刷が行えます。
CMP_BM_MODE_CMD_GRAY16DOWNLOAD	グレースケール ダウンロードグラフィックス コマンド出力 CT-E601、CT-S251/255/257/601II/651II/801II/851II系で利用可能です。より綺麗なグラフィックの印刷を行えます。データを一括して出力するため、16 階調のサイズで 384KB の制限があります。

戻り値

成功時は CMP_SUCCESS(0) を返します。それ以外のエラーコードは「[2.3.1 戻り値](#)」を参照してください。

使用例

```
string path = string.Empty;
if (Device.RuntimePlatform == Device.Android)
{
    path = "//sdcard//";
}
else if (Device.RuntimePlatform == Device.iOS)
{
    path = platform.iosBundlePath + "/";
}

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

2.3.13. SetNVBitmap メソッド

形式

- 1) int SetNVBitmap (int number, string fileName, int width)
- 2) int SetNVBitmap (int number, string fileName, int width, int mode)

パラメータ

パラメータの意味と設定可能な値は以下の通りです。

値	[IN/OUT]	意味	設定可能範囲
number	[IN]	プリンターのフラッシュメモリ内に格納する画像番号	1～20
fileName	[IN]	ビットマップファイル名	
width	[IN]	ビットマップ印刷幅	CMP_BM_ASIS: プリンターのドット当たり1ビットマップピクセルでビットマップを印刷します。 上記定数以外の1以上の値: ビットマップ幅を MapMode プロパティ で定義された単位(デフォルトはドット単位)で指定します。
mode	[IN]	ビットマップモード	CMP_BM_MODE_HT_THRESHOLD: ハーフトーン しきい値 CMP_BM_MODE_HT_DITHER: ハーフトーン デイザー CMP_BM_MODE_CMD_MONO: モノクロ登録 CMP_BM_MODE_CMD_GRAY16: グレースケール(16階調)登録

説明

このメソッドは、画像番号、ビットマップのファイル名、印刷幅、モードを指定して、プリンターのフラッシュメモリにビットマップ画像(ロゴ)を保存するために使用します。保存したロゴは、[PrintNVBitmap メソッド](#)や[WatermarkPrint メソッド](#)を使用して印刷できます。

印刷可能なビットマップ形式は、BMP/JPG/PNG/GIF です。

モードは、ハーフトーンと登録方法を組み合わせて指定可能です。組み合わせる場合は、論理和を指定してください。モードが省略された場合は、CMP_BM_MODE_HT_THRESHOLD | CMP_BM_MODE_CMD_MONO で登録します。モードに関する詳細は、次の通りです。

ハーフトーン ハーフトーン処理方法を指定します。

設定値	説明
CMP_BM_MODE_HT_THRESHOLD	しきい値 文字の印刷に適します。
CMP_BM_MODE_HT_DITHER	デイザー グラフィックの印刷に適します。

登録

登録方法を指定します。

設定値	説明
CMP_BM_MODE_CMD_MONO	モノクロ登録

CMP_BM_MODE_CMD_GRAY16	グレースケール登録 CT-E601、CT-S251/255/257/601II/651II/801II/851II 系で 利用可能です。より綺麗なグラフィックの登録が行えます。
------------------------	--

戻り値

成功時は CMP_SUCCESS(0) を返します。それ以外のエラーコードは「[2.3.1 戻り値](#)」を参照してください。

使用例

```
string path = string.Empty;
if (Device.RuntimePlatform == Device.Android)
{
    path = "//sdcard//";
}
else if (Device.RuntimePlatform == Device.iOS)
{
    path = platform.iosBundlePath + "/";
}

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

2.3.14. PrintNVBitmap メソッド

形式

```
int PrintNVBitmap (int nvImageNumber)
```

パラメータ

パラメータの意味と設定可能な値は以下の通りです。

値	[IN/OUT]	意味	設定可能範囲
nvImageNumber	[IN]	プリンターのフラッシュメモリ内に格納されている画像番号	1-20

説明

このメソッドは、プリンターのフラッシュメモリに保存されているビットマップ画像(ロゴ)を印刷するために使用します。

このメソッドを使用するためには、事前にロゴの登録が必要です。ロゴ登録は、[SetNVBitmap メソッド](#)で登録するか、プリンター用ユーティリティソフトウェアの「POS プリンターユーティリティ」を使用してください。

「POS プリンターユーティリティ」でロゴを登録する場合は、プリンターのモデルによって、登録モードが異なります。次の通り登録してください。

[CT-S281, PMU2300III/3300, CMP-20/30/40 系]

キーコード未使用モードでロゴ登録してください。

使用する画像番号に合わせて、順にロゴ登録する必要があります。

[CT-E301/601, CT-S251/253/255/257/401/601/651/801/851/601II/651II/801II/851II/2000/4000/4500 系]

キーコードモードでロゴ登録してください。

使用する画像番号に合わせて、キーコードを指定して登録する必要があります。

画像番号に対応するキーコードは次の通りです。

画像番号	キーコード (文字列)
1	"01"
2	"02"
3	"03"
⋮	⋮
19	"19"
20	"20"

戻り値

成功時は CMP_SUCCESS(0) を返します。それ以外のエラーコードは「[2.3.1 戻り値](#)」を参照してください。

使用例

```
printer.PrintNVBitmap( 1 );
```

2.3.15. PrintBarCode メソッド

形式

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

パラメータ

パラメータの意味と設定可能な値は以下の通りです。

値	[IN/OUT]	意味	設定可能範囲
data	[IN]	印刷するバーコードデータ	
symbology	[IN]	バーコードタイプ	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: コード 39 CMP_BCS_Code93: コード 93 CMP_BCS_Code128: コード 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]	バーコード高さ	1~255(ドット単位時) バーコードの高さを MapMode プロパティ で定義された単位(デフォルトはドット単位)で指定します。
width	[IN]	バーコード横サイズ(倍率)	2~6(ドット単位時) バーコードのモジュール幅を MapMode プロパティ で定義された単位(デフォルトはドット単位)で指定します。
alignment	[IN]	バーコード配置位置	CMP_ALIGNMENT_LEFT: 左揃え CMP_ALIGNMENT_CENTER: 中央揃え CMP_ALIGNMENT_RIGHT: 右揃え 上記定数以外の 3 以上の値: バーコード印刷を開始する左からの距離を MapMode プロパティ で定義された単位(デフォルトはドット単位)指定します。
textPosition	[IN]	可視コード印字位置	CMP_HRI_TEXT_NONE: 印刷しない CMP_HRI_TEXT_ABOVE: バーコードの上 CMP_HRI_TEXT_BELOW: バーコードの下

説明

このメソッドは、一次元バーコードを印刷するために使用します。

GS1 DataBar(CMP_BCS_GS1DATABAR、CMP_BCS_GS1DATABAR_E、CMP_BCS_GS1DATABAR_T、CMP_BCS_GS1DATABAR_L)は、CT-E301/601、CT-S251/253/255/257/401/601/651/801/851/601II/651II/801II/851II/4500 系のプリンターのみ使用できます。

ページモード中のバーコード配置位置の CMP_ALIGNMENT_CENTER と CMP_ALIGNMENT_RIGHT の指定は無視されます。

註: データには、使える文字種、桁数の制約があったり、Code Set 文字の追加が必要なものがあります。詳しくは、プリンターのコマンドリファレンスを参照ください。

戻り値

成功時は CMP_SUCCESS(0) を返します。それ以外のエラーコードは「[2.3.1 戻り値](#)」を参照してください。

使用例

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

2.3.16. PrintPDF417 メソッド

形式

int PrintPDF417 (string data, int digits, int steps, int moduleWidth, int stepHeight, int ECLLevel, int alignment)

パラメータ

パラメータの意味と設定可能な値は以下の通りです。

値	[IN/OUT]	意味	設定可能範囲
data	[IN]	印刷データ	
digits	[IN]	桁数	0: 自動 1～30
steps	[IN]	段数	0: 自動 3～90
moduleWidth	[IN]	モジュール幅	2～8(ドット単位時) バーコードのモジュール幅を MapMode プロパティ で定義された単位(デフォルトはドット単位)で指定します。
stepHeight	[IN]	段の高さ	2～8
ECLLevel	[IN]	エラー訂正レベル	CMP_PDF417_EC_LEVEL_0: レベル 0 CMP_PDF417_EC_LEVEL_1: レベル 1 CMP_PDF417_EC_LEVEL_2: レベル 2 CMP_PDF417_EC_LEVEL_3: レベル 3 CMP_PDF417_EC_LEVEL_4: レベル 4 CMP_PDF417_EC_LEVEL_5: レベル 5 CMP_PDF417_EC_LEVEL_6: レベル 6 CMP_PDF417_EC_LEVEL_7: レベル 7 CMP_PDF417_EC_LEVEL_8: レベル 8
alignment	[IN]	バーコード配置位置	CMP_ALIGNMENT_LEFT: 左揃え CMP_ALIGNMENT_CENTER: 中央揃え CMP_ALIGNMENT_RIGHT: 右揃え 上記定数以外の 3 以上の値: バーコード印刷を開始する左からの距離を MapMode プロパティ で定義された単位(デフォルトはドット単位)指定します。

説明

このメソッドは、PDF417 バーコードを印刷するために使用します。

各パラメータの詳細は、各プリンターのコマンドリファレンスを参照してください。

ページモード中のバーコード配置位置の CMP_ALIGNMENT_CENTER と CMP_ALIGNMENT_RIGHT の指定は無視されます。

戻り値

成功時は CMP_SUCCESS(0) を返します。それ以外のエラーコードは「[2.3.1 戻り値](#)」を参照してください。

使用例

```
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 );
```

2.3.17. PrintQRCode メソッド

形式

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

パラメータ

パラメータの意味と設定可能な値は以下の通りです。

値	[IN/OUT]	意味	設定可能範囲
data	[IN]	印刷データ	
moduleSize	[IN]	モジュールサイズ	1~16(ドット単位時) バーコードのモジュールサイズを MapMode プロパティ で定義された単位(デフォルトはドット単位)で指定します。
ECLevel	[IN]	エラー修正レベル	CMP_QRCODE_EC_LEVEL_L: レベル L(7%) CMP_QRCODE_EC_LEVEL_M: レベル M(15%) CMP_QRCODE_EC_LEVEL_Q: レベル Q(25%) CMP_QRCODE_EC_LEVEL_H: レベル H(30%)
alignment	[IN]	バーコード配置位置	CMP_ALIGNMENT_LEFT: 左揃え CMP_ALIGNMENT_CENTER: 中央揃え CMP_ALIGNMENT_RIGHT: 右揃え 上記定数以外の 3 以上の値: バーコード印刷を開始する左からの距離を MapMode プロパティ で定義された単位(デフォルトはドット単位)指定します。

説明

このメソッドは、QR コードを印刷するために使用します。

各パラメータの詳細は、各プリンターのコマンドリファレンスを参照してください。

ページモード中のバーコード配置位置の CMP_ALIGNMENT_CENTER と CMP_ALIGNMENT_RIGHT の指定は無視されます。

戻り値

成功時は CMP_SUCCESS(0) を返します。それ以外のエラーコードは「[2.3.1 戻り値](#)」を参照してください。

使用例

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

2.3.18. PrintGS1DataBarStacked メソッド

形式

int PrintGS1DataBarStacked (string data, int symbology, int moduleSize, int maxSize, int alignment)

パラメータ

パラメータの意味と設定可能な値は以下の通りです。

値	[IN/OUT]	意味	設定可能範囲
data	[IN]	印刷データ	
symbology	[IN]	バーコードタイプ	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]	モジュール幅	2~8(ドット単位時) バーコードのモジュール幅を MapMode プロパティ で定義された単位(デフォルトはドット単位)で指定します。
maxSize	[IN]	最大幅	106~39528(ドット単位時) GS1 DataBar Expanded Stacked の最大幅を MapMode プロパティ で定義された単位(デフォルトはドット単位)で指定します。
alignment	[IN]	バーコード配置位置	CMP_ALIGNMENT_LEFT: 左揃え CMP_ALIGNMENT_CENTER: 中央揃え CMP_ALIGNMENT_RIGHT: 右揃え 上記定数以外の 3 以上の値: バーコード印刷を開始する左からの距離を MapMode プロパティ で定義された単位(デフォルトはドット単位)指定します。

説明

このメソッドは、2次元の GS1 DataBar を印刷するために使用します。

このメソッドは、CT-E301/601、CT-S251/253/255/257/401/601/651/801/851/601II/651II/801II/851II/4500 系のプリンターのみ使用できます。

各パラメータの詳細は、各プリンターのコマンドリファレンスを参照してください。

ページモード中のバーコード配置位置の CMP_ALIGNMENT_CENTER と CMP_ALIGNMENT_RIGHT の指定は無視されます。

戻り値

成功時は CMP_SUCCESS(0) を返します。それ以外のエラーコードは「[2.3.1 戻り値](#)」を参照してください。

使用例

```
printer.PrintGS1DataBarStacked(
    "0123456789012",
    ESCPOSConst.CMP_BCS_GS1DATABAR_S,
    4,
    300,
    ESCPOSConst.CMP_ALIGNMENT_LEFT );
```

2.3.19. CutPaper メソッド

形式

int CutPaper (int type)

パラメータ

パラメータの意味と設定可能な値は以下の通りです。

値	[IN/OUT]	意味	設定可能範囲
type	[IN]	カット種類	CMP_CUT_FULL: フルカット CMP_CUT_PARTIAL: パーシャルカット CMP_CUT_FULL_PREFEED: カット位置紙送り後、フルカット CMP_CUT_PARTIAL_PREFEED: カット位置紙送り後、パーシャルカット

説明

このメソッドは、用紙をカットするために使用します。

戻り値

成功時は CMP_SUCCESS(0) を返します。それ以外のエラーコードは「[2.3.1 戻り値](#)」を参照してください。

使用例

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

2.3.20. UnitFeed メソッド

形式

```
int UnitFeed (int ufCount)
```

パラメータ

パラメータの意味と設定可能な値は以下の通りです。

値	[IN/OUT]	意味	設定可能範囲
ufCount	[IN]	紙送り数	MapMode プロパティ で定義された単位 (デフォルトはドット単位)で指定します。

説明

このメソッドは、ドット単位で紙送りするために使用します。

戻り値

成功時は CMP_SUCCESS(0) を返します。それ以外のエラーコードは「[2.3.1 戻り値](#)」を参照してください。

使用例

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

2.3.21. MarkFeed メソッド

形式

int MarkFeed (int type)

パラメータ

パラメータの意味と設定可能な値は以下の通りです。

値	[IN/OUT]	意味	設定可能範囲
type	[IN]	ラベル用紙／ブラックマーク用紙のハンドリングの種類	CMP_MF_TO_CUTTER: 自動カッターのカット位置の上までフィードし、さらにカット CMP_MF_TO_NEXT_TOF: 次の用紙先頭印字位置までフィード

説明

このメソッドは、ラベル用紙／ブラックマーク用紙を利用するために使用します。

戻り値

成功時は CMP_SUCCESS(0) を返します。それ以外のエラーコードは「[2.3.1 戻り値](#)」を参照してください。

使用例

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

2.3.22. OpenDrawer メソッド

形式

int OpenDrawer (int drawer, int pulseLen)

パラメータ

パラメータの意味と設定可能な値は以下の通りです。

値	[IN/OUT]	意味	設定可能範囲
drawer	[IN]	キャッシュドロワー番号	CMP_DRAWER_1: ドロワー1 CMP_DRAWER_2: ドロワー2
pulseLen	[IN]	シグナルの長さ	1~8 (×100) msec

説明

このメソッドは、プリンターに接続されたキャッシュドロワーをオープンするために使用します。

戻り値

成功時は CMP_SUCCESS(0) を返します。それ以外のエラーコードは「[2.3.1 戻り値](#)」を参照してください。

使用例

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

2.3.23. TransactionPrint メソッド

形式

int TransactionPrint (int control)

パラメータ

パラメータの意味と設定可能な値は以下の通りです。

値	[IN/OUT]	意味	設定可能範囲
control	[IN]	一括処理を開始／終了を指定	CMP_TP_TRANSACTION: 一括処理の開始 CMP_TP_NORMAL: バッファデータを印刷して一括処理を終了

説明

このメソッドは、一括処理モードを開始または終了するために使用します。

control が CMP_TP_TRANSACTION ならば、一括処理モードに入ります。これ以降のメソッド呼び出しは、印刷データをバッファリングします。一括処理モードに該当するメソッドは以下の通りです。

PrintText, PrintPaddingText, PrintTextLocalFont, PrintBitmap, PrintNVBitmap, PrintBarCode, PrintPDF417, PrintQRCode, PrintGS1DataBarStacked, CutPaper, UnitFeed, MarkFeed, OpenDrawer, RotatePrint, PageModePrint, ClearPrintArea, PrintData, PrintNormal

control が CMP_TP_NORMAL ならば、一括処理モードを抜けます。データがバッファリングされていたならば、そのデータは印刷されます。一括処理全体は、一つのメッセージとして処理されます。

[ClearOutput メソッド](#)を呼び出すことによって、一括処理モードは取り消されます。バッファされた印刷行も削除されます。

戻り値

成功時は CMP_SUCCESS(0) を返します。それ以外のエラーコードは「[2.3.1 戻り値](#)」を参照してください。

使用例

```
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 );
```

2.3.24. RotatePrint メソッド

形式

int RotatePrint (int rotation)

パラメータ

パラメータの意味と設定可能な値は以下の通りです。

値	[IN/OUT]	意味	設定可能範囲
rotation	[IN]	回転方向を指定	CMP_RP_ROTATE180: 180度回転印刷、つまり倒立印字の開始 CMP_RP_BARCODE: 回転バーコード印刷の開始、この値は上記の回転印刷開始の値との論理和 CMP_RP_BITMAP: 回転ビットマップ印刷の開始、この値は上記の回転印刷開始の値との論理和 CMP_RP_NORMAL: 回転印刷の終了

説明

このメソッドは、回転印刷モードを開始または終了するために使用します。

rotation に CMP_RP_ROTATE180 が含まれている場合は、倒立印刷モードに入ります。回転印刷モードに該当するメソッドは以下の通りです。

PrintText、PrintNormal

更に CMP_RP_BARCODE あるいは CMP_RP_BITMAP が含まれている場合は、次のメソッドも回転して印刷されます。

PrintBarcode, PrintPDF417, PrintQRCode, PrintGS1DataBarStacked あるいは PrintBitmap

rotation が CMP_RP_NORMAL ならば、回転印刷モードを抜けます。

戻り値

成功時は CMP_SUCCESS(0) を返します。それ以外のエラーコードは「[2.3.1 戻り値](#)」を参照してください。

使用例

```
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.TransactionPrint( ESCPOSConst.CMP_TP_NORMAL );
```

2.3.25. PageModePrint メソッド

形式

int PageModePrint (int control)

パラメータ

パラメータの意味と設定可能な値は以下の通りです。

値	[IN/OUT]	意味	設定可能範囲
control	[IN]	ページモードコントロール	CMP_PM_PAGE_MODE: ページモードの開始 CMP_PM_PRINT_SAVE: ページモード印刷領域の印刷データを印刷し、その印刷データを保存 CMP_PM_NORMAL: ページモード印刷領域の印刷データを印刷し、その印刷データを消去し、ページモードを終了 CMP_PM_CANCEL: ページモード印刷領域の印刷データを消去し、何も印刷せずにページモードを終了

説明

このメソッドは、ページモードを開始または終了するために使用します。

control に CMP_PM_PAGE_MODE を指定した場合、ページモードを開始します。これ以降のメソッド呼び出しは、印刷データをバッファリングします。ページモードで利用可能なメソッドは以下の通りです。

PrintText, PrintBitmap, PrintBarCode, PrintPDF417, PrintQRCode, PrintGS1DataBarStacked, PrintNormal

control に CMP_PM_PRINT_SAVE を指定した場合、ページモードが継続します。バッファリングした印刷データがあれば、それを保存して、印刷します。ページモード領域内に印刷データを追加して、同じレイアウトで印刷したい場合に用います。

control に CMP_PM_NORMAL を指定した場合、ページモードを終了し、通常の状態へ移行します。バッファリングした印刷データがあれば、それを印刷します。バッファリングした印刷データは保存されません。

control に CMP_PM_CANCEL を指定した場合、ページモードを終了し、通常の状態へ移行します。バッファリングした印刷データがあっても、それらは印刷されず、保存もされません。

PageModePrint メソッドが呼び出された時、[PageModePrintArea プロパティ](#)で定義されたページモード印刷領域上のすべての印刷データが印刷され、ページモード印刷領域の下端まで紙送りされます。

複数のページモード印刷領域が定義されていた場合は、それぞれのページモード印刷領域上のすべての印刷データが印刷され、最も下方に配置されているページモード印刷領域の下端まで紙送りされます。

ページモード領域全体を1つのデータとして送信します。

[ClearOutput メソッド](#)を呼び出すことによって、ページモードは取り消されます。バッファされた印刷行も削除されます。

戻り値

成功時は CMP_SUCCESS(0) を返します。それ以外のエラーコードは「[2.3.1 戻り値](#)」を参照してください。

2.3.26. ClearPrintArea メソッド

形式

```
int ClearPrintArea ()
```

パラメータ

ありません。

説明

このメソッドは、[PageModePrintArea プロパティ](#)で定義したページモード印刷領域上の印刷データを消去するために使用します。

戻り値

成功時は CMP_SUCCESS(0) を返します。それ以外のエラーコードは「[2.3.1 戻り値](#)」を参照してください。

使用例

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

2.3.27. ClearOutput メソッド

形式

```
int ClearOutput ()
```

パラメータ

ありません。

説明

このメソッドは、[TransactionPrint メソッド](#)や [PageModePrint メソッド](#)でバッファリングされている全ての送信データをクリアするために使用します。同時にプリンター上の印刷中データをクリアするコマンドを送信します。

戻り値

成功時は CMP_SUCCESS(0) を返します。それ以外のエラーコードは「[2.3.1 戻り値](#)」を参照してください。

使用例

```
printer.ClearOutput ();
```

2.3.28. PrintData メソッド

形式

```
int PrintData (byte[] data)
```

パラメータ

パラメータの意味と設定可能な値は以下の通りです。

値	[IN/OUT]	意味	設定可能範囲
data	[IN]	送信データ	

説明

このメソッドは、バイトデータをそのままプリンターに送信するために使用します。
通常は必要ありませんが、プリンターの ESC コマンドを直接送信したい場合に使用してください。
ご使用の際は、他のメソッドに影響を与えない様に注意する必要があります。

戻り値

成功時は CMP_SUCCESS(0) を返します。それ以外のエラーコードは「[2.3.1 戻り値](#)」を参照してください。

使用例

```
// ブザーを鳴らす (ブザー対応プリンターが必要です)
byte[] data = { 0x1b, 0x1e };
printer.PrintData( data );
```

2.3.29. PrintNormal メソッド

形式

int PrintNormal (string data)

パラメータ

パラメータの意味と設定可能な値は以下の通りです。

値	[IN/OUT]	意味	設定可能範囲
Data	[IN]	印字データ (OPOS エスケープシーケンス対応)	

説明

このメソッドは、OPOS で定義されているエスケープシーケンスを使用して印字する場合に使用します。

OPOS に精通している場合に、ご使用ください。

本 SDK に対応するエスケープシーケンスは、下記の通りです。

詳細は OPOS の仕様書を参照ください。

エスケープシーケンス		注意事項
用紙カット	ESC #P	パーシャルカット(1-99)、フルカット(0,100)
フィードと用紙カット	ESC #fP	パーシャルカット(1-99)、フルカット(0,100)
ビットマップ印刷	ESC #B	1-20(プリンターに登録された Bitmap 番号を指定) Bitmap 印字後の印字位置は初期状態(左寄せ)にもどります
複数行フィード	ESC #fF	
単位フィード	ESC #uF	
バーコード印刷	ESC #R	
フォントタイプ指定	ESC #fT	
ボールド	ESC bC	
アンダーライン	ESC #uC	
カスタムカラー	ESC #rC	専用の 2 色紙を使用時のみ有効
赤色	ESC rC	専用の 2 色紙を使用時のみ有効
反転文字	ESC rvC	
縦横 1 倍角	ESC 1C	
横倍角	ESC 2C	
縦倍角	ESC 3C	
縦横倍角	ESC 4C	
横倍率	ESC #hC	1-8
縦倍率	ESC #vC	1-8
中央揃え	ESC cA	
右寄せ	ESC rA	
ノーマル	ESC N	

戻り値

成功時は CMP_SUCCESS(0) を返します。それ以外のエラーコードは「[2.3.1 戻り値](#)」を参照してください。

使用例

```
printer.PrintNormal( "¥u001b|4C- Receipt -¥n" );
```

2.3.30. WatermarkPrint メソッド

形式

```
int WatermarkPrint (int start, int nvImageNumber, int pass, int feed, int repeat);
```

パラメータ

パラメータの意味と設定可能な値は以下の通りです。

値	[IN/OUT]	意味	設定可能範囲
start	[IN]	開始／終了	CMP_WM_START: 透かし印刷を開始します CMP_WM_STOP: 透かし印刷を終了します。
nvImageNumber	[IN]	プリンターのフラッシュメモリ内に格納されている画像番号	1～20
pass	[IN]	透かし印刷初回開始位置(縦方向)	0～65,535(ドット単位時) 印刷開始位置を MapMode プロパティ で定義された単位(デフォルトはドット単位)で指定します。
feed	[IN]	透かし印刷間隔(縦方向)	0～65,535(ドット単位時) 印刷間隔を MapMode プロパティ で定義された単位(デフォルトはドット単位)で指定します。
repeat	[IN]	透かし印刷繰り返し回数	0: 無限 1～65,535: 繰り返し回数

説明

このメソッドは、透かし印刷を行うために使用します。

CT-S251/255/257/601II/651II/801II/851II 系のプリンターで利用可能です。

プリンターのフラッシュメモリに保存されているビットマップ画像を重ねて印刷します。

このメソッドを使用するためには、事前にロゴの登録が必要です。ロゴ登録は、[SetNVBitmap メソッド](#) で登録するか、プリンター用ユーティリティソフトウェアの「POS プリンターユーティリティ」を使用してください。

CMP_WM_STOP で透かし印刷終了を指定した時は、他の全ての引数は無視されます。

戻り値

成功時は CMP_SUCCESS(0) を返します。それ以外のエラーコードは「[2.3.1 戻り値](#)」を参照してください。

使用例

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

2.3.31. SearchCitizenPrinter メソッド

形式

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

パラメータ

パラメータの意味と設定可能な値は以下の通りです。

値	[IN/OUT]	意味	設定可能範囲
connectType	[IN]	接続タイプ	CMP_PORT_WiFi CMP_PORT_BLUETOOTH CMP_PORT_USB (iOS のみ)
searchTime	[IN]	検索時間 (秒)	0: CMP_PORT_Bluetooth の場合、ペアリング済 みアドレス取得 1~30: 指定時間検索
result	[OUT]	エラーコード	

result には成功時は CMP_SUCCESS(0) を返します。失敗時は下記のエラーコードの説明を確認してください。それ以外のエラーコードは「[2.3.1 戻り値](#)」を参照してください。

値	設定可能範囲
CMP_E_NOCONTEXT (1006)	コンテキストを指定していません。
CMP_E_ILLEGAL (1101)	無効なパラメータ値です。 ①未対応の接続タイプを指定された場合 ②範囲外の検索時間を指定された場合
CMP_E_FAILURE (1104)	検索処理の途中でエラーが発生し、検索に失敗しました。
CMP_E_NO_LIST (1106)	検索の結果、プリンターが発見出来ませんでした。

説明

このメソッドは、プリンターを検索し、プリンターの情報リストを取得するために使用します。接続タイプと検索時間を指定して実行してください。Android の場合、メソッド実行前に [SetContext メソッド](#) を実行する必要があります。このメソッドはシミュレーターには対応していません。

検索時間経過後に、結果を result パラメータにセットし、検索できたプリンターの情報を配列で返します。接続タイプが CMP_PORT_WiFi の場合、CT-E301/601、CT-S251/253/255/257/401/601/651/801/851/601II/651II/801II/851II/4500 系のプリンターのみ検索可能です。検索時間の推奨値は 3 秒以上です。これより短い時間の場合、ネットワークの状態によっては検索漏れが発生する場合があります。

接続タイプが CMP_PORT_Bluetooth の場合は、プラットフォームにより動作が異なります。Android の場合、検索時間に 0 を指定すると、ペアリング済みのアドレスを取得します。検索時間に 1~30 を指定すると、接続可能なアドレスを取得します。検索時間の推奨値は 10 秒以上です。これより短い時間の場合、Bluetooth の状態によっては検索漏れが発生する場合があります。iOS の場合、検索時間である searchTime に関係無く、ペアリング済みのアドレスを取得します。

接続タイプが CMP_PORT_USB の場合、iOS で CT-E601、CT-S251/255/257/4500 が検索可能です。検索時間である searchTime に関係無く、検索は直ぐに完了します。

戻り値

検索が成功した場合、プリンターの情報リストが返されます。失敗した場合、空のリストを返します。プリンターの情報リストは CitizenPrinterInfo 型で格納されており、接続タイプによって取得できる情報が異なります。

接続タイプ	CitizenPrinterInfo	取得出来る情報
CMP_PORT_WiFi	IpAddress	IP アドレス

	MacAddress	MAC アドレス
	DeviceName	(空)
	BdAddress	(空)
	usbSerialNo	(空)
CMP_PORT_BLUETOOTH	IpAddress	(空)
	MacAddress	(空)
	DeviceName	Bluetooth デバイス名
	BdAddress	Bluetooth デバイスアドレス
	usbSerialNo	(空)
CMP_PORT_USB	ipAddress	(空)
	macAddress	(空)
	deviceName	モデル名
	bdAddress	(空)
	usbSerialNo	USB シリアル番号

使用例

```
int[] errCode = new int[1];
CitizenPrinterInfo[] list = printer.SearchCitizenPrinter(
    ESCPOSConst.CMP_PORT_WiFi, 3, errCode );
for (int i = 0; i < list.Length; i++)
{
    Console.WriteLine("IP Address: " + list[i].IpAddress);
    Console.WriteLine("MAC Address: " + list[i].MacAddress);
}
```

2.3.32. SearchESCPOSPrinter メソッド

形式

```
string[] SearchESCPOSPrinter (int connectType, int serchTime, int[] result)
```

パラメータ

パラメータの意味と設定可能な値は以下の通りです。

値	[IN/OUT]	意味	設定可能範囲
connectType	[IN]	接続タイプ	CMP_PORT_WiFi CMP_PORT_Bluetooth CMP_PORT_USB (iOS のみ)
searchTime	[IN]	検索時間 (秒)	0: Android の CMP_PORT_Bluetooth の場合、 ペアリング済みアドレス取得 1~30: 指定時間検索
result	[OUT]	エラーコード	

result には成功時は CMP_SUCCESS(0) を返します。失敗時は下記のエラーコードの説明を確認してください。それ以外のエラーコードは「[2.3.1 戻り値](#)」を参照してください。

値	設定可能範囲
CMP_E_NOCONTEXT (1006)	コンテキストを指定していません。
CMP_E_ILLEGAL (1101)	無効なパラメータ値です。 ①未対応の接続タイプを指定された場合 ②範囲外の検索時間を指定された場合
CMP_E_FAILURE (1104)	検索処理の途中でエラーが発生し、検索に失敗しました。
CMP_E_NO_LIST (1106)	検索の結果、プリンターが発見出来ませんでした。

説明

このメソッドは、プリンターを検索するために使用します。接続タイプと検索時間を指定して実行してください。Android の場合、メソッド実行前に SetContext メソッドを実行する必要があります。このメソッドはシミュレーターには対応していません。

検索時間経過後に、結果を result パラメータにセットし、検索できたプリンターのアドレスを String 配列で返します。

接続タイプが CMP_PORT_WiFi の場合、CT-E301/601、CT-S251/253/255/257/401/601/651/801/851/601II/651II/801II/851II/4500 系のプリンターのみ検索可能です。検索時間の推奨値は 3 秒以上です。これより短い時間の場合、ネットワークの状態によっては検索漏れが発生する場合があります。

接続タイプが CMP_PORT_Bluetooth の場合は、動作させるプラットフォームにより動作が異なります。Android の場合、検索時間に 0 を指定すると、ペアリング済みのアドレスを取得します。検索時間に 1~30 を指定すると、接続可能なアドレスを取得します。検索時間の推奨値は 10 秒以上です。これより短い時間の場合、Bluetooth の状態によっては検索漏れが発生する場合があります。iOS の場合、検索時間である searchTime に関係無く、ペアリング済みのアドレスを取得します。

接続タイプが CMP_PORT_USB の場合は、iOS で CT-E601、CT-S251/255/257/4500 が検索可能です。検索時間である searchTime に関係無く、検索は直ぐに完了します。

戻り値

接続タイプが CMP_PORT_WiFi の場合、成功した時はプリンターの IP アドレスのリストが返されます。失敗した時は空のリストが返されます。

接続タイプが CMP_PORT_BLUETOOTH の場合、成功した時はプリンターの Bluetooth デバイスアドレスのリストが返されます。失敗した時は空のリストが返されます。

接続タイプが CMP_PORT_USB の場合、成功した時はプリンターのシリアル番号のリストが返されます。失敗した時は空のリストが返されます。

使用例

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

2.3.33. SetLog メソッド (Android のみ)

形式

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

パラメータ

パラメータの意味と設定可能な値は以下の通りです。

値	[IN/OUT]	意味	設定可能範囲
mode	[IN]	ログモード	0 : 記録なし 1 : アクセス履歴の記録 2 : エラーのみ記録
path	[IN]	格納フォルダ	SD カードのフォルダ
maxSize	[IN]	ログサイズ	0: サイズ制限なし 1~: 最大サイズ(MB)

説明

このメソッドは、ログ機能を設定するために使用します。ログ機能の詳細は「[2.4.4 ログ機能について](#)」を参照してください。

戻り値

ありません。

使用例

```
printer.SetLog(1, "/temp/citizen/Log", 10);
```

2.3.34. SetPrintCompletedTimeout メソッド

形式

int SetPrintCompletedTimeout (int timeout)

パラメータ

パラメータの意味と設定可能な値は以下の通りです。

値	[IN/OUT]	意味	設定可能範囲
timeout	[IN]	印刷完了通知確認タイムアウト (msec)	0: タイムアウトを自動調整します。 上記定数以外の 1 以上の値: タイムアウトを msec 単位で指定します。

説明

このメソッドは、印刷完了通知を確認するタイムアウトを設定するために使用します。タイムアウトは、インスタンス生成時に 0 に初期化します。0 を指定した場合は、印刷データに合わせてタイムアウトを自動的に調整されます。1 以上の値を指定した場合は、指定されたタイムアウトに固定されます。印刷完了確認処理の詳細は、「[2.4.1 印刷完了確認機能について](#)」を参照してください。

戻り値

成功時は CMP_SUCCESS(0)を返します。それ以外のエラーコードは「[2.3.1 戻り値](#)」を参照してください。

使用例

```
// 自動調整
printer.SetPrintCompletedTimeout(0);

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

2.3.35. GetVersionCode メソッド

形式

```
int GetVersionCode ()
```

パラメータ

ありません。

説明

このメソッドは、実行中のプラットフォームにおけるベースの SDK バージョン番号を数値で取得するために使用します。

戻り値

SDK のバージョン番号を数値 (Ver1.00 の場合: 100) で返します。

使用例

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

2.3.36. GetVersionName メソッド

形式

```
string GetVersionName ()
```

パラメータ

ありません。

説明

このメソッドは、動作させるプラットフォームにおけるベースの SDK バージョン番号を文字列で取得するために使用します。

戻り値

SDK のバージョン番号を文字列 (Ver1.00 の場合: "1.00") で返します。

使用例

```
string vname = printer.GetVersionCode ();
```

2.3.37. PageModeArea プロパティ

形式

string PageModeArea

属性

Read only

説明

このプロパティは、ページモード印刷領域で設定可能な最大値である、ページ領域を [MapMode プロパティ](#) で定義された単位 (デフォルトはドット単位) で保持します。

このプロパティの構成は、カンマ区切りの 2 つの ASCII 数字で構成され、幅と高さの順に列挙します。

ページ領域は、プリンターのハードウェア的な能力で決定されます。

[CT-S251 系]: "432,1662"

[CT-S281 系]: "384,938"

[CT-E301/601, CT-S253/255/257/401/601/651/801/851/601II/651II/801II/851II/2000 系]:
"576,1662"

[CT-S4000/4500 系]: "832,1662"

[PMU3300 系]: "576,938"

[CMP-20 系]: "384,938"

[CMP-30 系]: "576,938"

例えば、文字列が "384,938" であれば、ページモード領域は幅 384 単位、高さ 938 単位です。このページモード領域は、左上隅 (0,0) と右下隅 (383,937) で囲まれる長方形を示します。

このプロパティにアクセスする前に Connect メソッドを完了して下さい。このプロパティは、Connect メソッドで設定されます。

設定方法

なし。

取得方法

string GetPageModeArea()

戻り値として、ページ領域を返します。

2.3.38. PageModePrintArea プロパティ

形式

string PageModePrintArea

属性

Read/Write

説明

このプロパティは、ページモード印刷領域を [MapMode プロパティ](#) で定義された単位 (デフォルトはドット単位) で保持します。ページモード印刷領域のサイズは、ページ領域より大きく取ることはできません。このプロパティは、カンマ区切りの 4 つの ASCII 数字のみで構成され、空白文字を含めることはできません。構成は、始点の水平方向座標、始点の垂直方向座標、水平方向の幅、垂直方向の高さ、の順に列挙します。ページモード印刷領域の右端を超える文字は、次の行に印刷されます。ページモード印刷領域の下端を超える文字と画像は、印刷されません。

例えば文字列が、“50,100,200,400” である場合、ページモード印刷領域の座標は、左上隅 (50,100) と右下隅 (249,499) で囲まれる長方形を示します。

このプロパティにアクセスする前に Connect メソッドを完了して下さい。このプロパティは、Connect メソッドで “0,0,0,0” に初期化されます。

設定方法

int SetPageModePrintArea (string area)

パラメータに、設定したいプロパティ値を指定してください。

成功時は CMP_SUCCESS(0) を返します。それ以外のエラーコードは「[2.3.1 戻り値](#)」を参照してください。

取得方法

string GetPageModePrintArea ()

戻り値として、設定されているページモード印刷領域を返します。

2.3.39. PageModePrintDirection プロパティ

形式

int PageModePrintDirection

属性

Read/Write

説明

このプロパティは、現在編集集中のページモード印刷領域内の印刷方向を保持します。値は次の通りです。

値	意味
CMP_PD_LEFT_TO_RIGHT	ページモード印刷領域の左上隅を始点に、左から右方向へ印刷します。通常印刷方向です。
CMP_PD_BOTTOM_TO_TOP	ページモード印刷領域の左下隅を始点に、下から上方向へ印刷します。左 90 度回転印刷です。
CMP_PD_RIGHT_TO_LEFT	ページモード印刷領域の右下隅を始点に、右から左方向へ印刷します。180 度回転印刷です。
CMP_PD_TOP_TO_BOTTOM	ページモード印刷領域の右上隅を始点に、上から下方向へ印刷します。右 90 度回転印刷です。

このプロパティを変更すると、PageModeHorizontalPosition プロパティと PageModeVerticalPosition プロパティで示される印刷開始点の補正方向も変化します。

更にページモード印刷領域を切り替えることで、文字の回転方向を組み合わせたレシートや単票を印刷することができます。

このプロパティにアクセスする前に、Connect メソッドを完了して下さい。このプロパティは、Connect メソッドで CMP_PD_LEFT_TO_RIGHT に初期化されます。

設定方法

int SetPageModePrintDirection (int direction)

パラメータに、設定したいプロパティ値を指定してください。

成功時は CMP_SUCCESS(0) を返します。それ以外のエラーコードは「[2.3.1 戻り値](#)」を参照してください。

取得方法

int GetPageModePrintDirection ()

戻り値として、設定されているページモード印刷領域の印刷方向を返します。

2.3.40. PageModeHorizontalPosition プロパティ

形式

int PageModeHorizontalPosition

属性

Read/Write

説明

このプロパティは、ページモード印刷領域内の印刷開始位置を水平方向に補正するためのオフセット値を [MapMode プロパティ](#) で定義された単位 (デフォルトはドット単位) で保持します。

水平方向とは、PageModePrintDirection プロパティで設定した印刷方向と同じ方向を指します。

このプロパティは、現在位置ではなくて、最後に指定した水平方向のオフセット値の設定となります。

このプロパティにアクセスする前に、Connect メソッドを完了して下さい。このプロパティは、Connect メソッドでゼロ (0) に初期化されます。

設定方法

int SetPageModeHorizontalPosition (int position)

パラメータに、設定したいプロパティ値を指定してください。

成功時は CMP_SUCCESS(0) を返します。それ以外のエラーコードは「[2.3.1 戻り値](#)」を参照してください。

取得方法

int GetPageModeHorizontalPosition ()

戻り値として、設定されているページモード印刷領域の水平方向オフセットを返します。

2.3.41. PageModeVerticalPosition プロパティ

形式

int PageModeVerticalPosition

属性

Read/Write

説明

このプロパティは、ページモード印刷領域内の印刷開始位置を垂直方向に補正するためのオフセット値を [MapMode プロパティ](#) で定義された単位 (デフォルトはドット単位) で保持します。

垂直方向とは、PageModePrintDirection プロパティで設定した印刷方向に対して垂直な方向です。

このプロパティは、現在位置ではなくて、最後に指定した垂直方向のオフセット値の設定となります。

このプロパティにアクセスする前に、Connect メソッドを完了して下さい。このプロパティは、Connect メソッドでゼロ (0) に初期化されます。

設定方法

int SetPageModeVerticalPosition (int position)

パラメータに、設定したいプロパティ値を指定してください。

成功時は CMP_SUCCESS(0) を返します。それ以外のエラーコードは「[2.3.1 戻り値](#)」を参照してください。

取得方法

int GetPageModeVerticalPosition ()

戻り値として、設定されているページモード印刷領域の垂直方向オフセットを返します。

2.3.42. RecLineSpacing プロパティ

形式

int RecLineSpacing

属性

Read/Write

説明

このプロパティは、通常文字の印刷行の高さを [MapMode プロパティ](#) で定義された単位 (デフォルトはドット単位) で保持します。すなわち、印字行の高さと行間スペースの高さの両方を加えたものです。

現在の行間の値によっては、縦倍角文字はこの値を超える場合があります。この場合の行間スペースは無しになります。

このプロパティにアクセスする前に、Connect メソッドを完了して下さい。このプロパティは、Connect メソッドで 34 に初期化されます。

設定方法

int SetRecLineSpacing (int spacing)

パラメータに、設定したいプロパティ値を指定してください。

成功時は CMP_SUCCESS(0) を返します。それ以外のエラーコードは「[2.3.1 戻り値](#)」を参照してください。

取得方法

int GetRecLineSpacing ()

戻り値として、設定されている通常文字の印刷行の高さを返します。

2.3.43. MapMode プロパティ

形式

int MapMode

属性

Read/Write

説明

このプロパティは、プリンターのマッピングモードを保持します。マッピングモードはメソッドや他のプロパティで使用されるラインの高さや行間を示す様な尺度の単位を定義します。値は次の通りです。

値	意味
CMP_MM_DOTS	POS プリンタのドット幅
CMP_MM_TWIPS	1 インチの 1/1440
CMP_MM_ENGLISH	0.001 インチ
CMP_MM_METRIC	0.01 ミリメートル

MapMode プロパティの影響を受けるメソッドの引数およびプロパティは以下の通りです。

[PrintBitmap メソッド](#)の width, alignment
[SetNVBitmap メソッド](#)の width
[PrintBarcode メソッド](#)の height, width, alignment
[PrintPDF417 メソッド](#)の moduleWidth, alignment
[PrintQRCode メソッド](#)の moduleSize, alignment
[PrintGS1DataBarStacked メソッド](#)の moduleSize, maxSize, alignment
[UnitFeed メソッド](#)の ufCount
[WatermarkPrint メソッド](#)の pass, feed
[PageModeArea](#) プロパティ
[PageModePrintArea](#) プロパティ
[PageModeHorizontalPosition](#) プロパティ
[PageModeVerticalPosition](#) プロパティ
[RecLineSpacing](#) プロパティ

このプロパティにアクセスする前に、Connect メソッドを完了して下さい。このプロパティは、Connect メソッドで CMP_MM_DOTS に初期化されます。

設定方法

int SetMapMode (int mode)

パラメータに、設定したいプロパティ値を指定してください。

成功時は CMP_SUCCESS(0) を返します。それ以外のエラーコードは「[2.3.1 戻り値](#)」を参照してください。

取得方法

int GetMapMode ()

戻り値として、設定されているマッピングモードを返します。

使用例

```

printer.SetMapMode( ESCPOSConst.CMP_MM_DOTS );
printer.UnitFeed( 200 ); // 200 ドット紙送り
printer.SetMapMode( ESCPOSConst.CMP_MM_METRIC );
printer.UnitFeed( 2500 ); // 25mm 紙送り

```

2.4. 注意事項

本 SDK の注意事項を以下に示します。

2.4.1. 印刷完了確認機能について

本 SDK は、プリンターへのデータ送信後、印刷完了通知を待ち、メソッドの成功／失敗を判断しています。印刷完了確認機能は次のタイミングで処理されます。

- (1) 一括処理 (TransactionPrint メソッド) の完了時
- (2) ページモード (PageModePrint メソッド) の完了時
- (3) 一括処理またはページモードのバッファリング中以外のメソッドのデータ送信時

印刷完了確認機能は、プリンターの応答を待つため時間がかかります。複数のメソッドを連続して印刷する場合は、一括処理 (TransactionPrint メソッド) を使用する事によりスムーズな印刷が可能です。

印刷完了通知を確認するタイムアウトは、印刷データに合わせて自動的に調整されます。印刷データによっては、毎回タイムアウトエラーが発生する場合があります。その場合は、実際の印刷時間に合わせて [SetPrintCompletedTimeout メソッド](#) でタイムアウトを設定してください。

2.4.2. UTF-8 エンコード文字列の印刷について

本 SDK は、UTF-8 でエンコードされた文字列の印刷をサポートします。

使用例

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

対応機種

機種	ファームウェアバージョン	制限事項
CT-S251	EM01-0304 以降	※1
CT-S401	DT00-1000 以降 DT10-1100 以降	
CT-S601II	EE00-0200 以降	※2
CT-S651II	EA00-0200 以降	
CT-S801II	ED00-0200 以降	
CT-S851II	DY00-0200 以降	
CT-E301/601	全バージョン	※3
CT-S253/255/257/4500		

注意

- ※1 プリンタは日本語、韓国語、簡体字中国語、繁体字中国語の同時印刷に未対応です。印刷可能な言語はプリンタの各仕向け毎に決められた1つの言語のみとなります。
- ※2 プリンタは日本語、韓国語、簡体字中国語、繁体字中国語の同時印刷に未対応です。印刷可能な言語はプリンタの MSW9-4 で指定される1つの言語のみとなります。
- ※3 プリンタは日本語、韓国語、簡体字中国語、繁体字中国語の同時印刷に対応します。複数の言語を印刷する場合、プリンタは MSW9-4 の設定に割り当てられた言語に基づいて、使用する文字を順次検索します。その際、字体、字形、書体が一様とならない場合があります。ご注意ください。

言語と書体 (CT-S253/255 系)

言語	書体
日本語フォント 韓国語フォント	ゴシック体

簡体中国語フォント 繁体中国語フォント	明朝体
------------------------	-----

言語と書体 (CT-E301/601、CT-S257/4500 系)

言語	書体
日本語フォント 韓国語フォント 簡体中国語フォント 繁体中国語フォント	ゴシック体

2.4.3. JIS 第3、第4水準漢字の印刷について

本 SDK は、JIS 第3、第4水準漢字の印刷をサポートしています。

JIS 第3、第4水準漢字を印刷には、プリンター送信データのエンコードを、“UTF-8”に設定する必要があります。詳しくは、「[2.4.2 UTF-8 エンコード文字列の印刷について](#)」を参照してください。

使用例

```
// エンコード設定
```

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

```
// テキスト印刷
```

```
printer.PrintText("啞焰鷗摑麴嘘俠頰軀俱繫妍鹼嚙攢繡蔣醬蟬搔瘦¥n",
    ESCPOSConst.CMP_ALIGNMENT_LEFT, ESCPOSConst.CMP_FNT_DEFAULT,
    ESCPOSConst.CMP_TXT_1WIDTH | ESCPOSConst.CMP_TXT_1WIDTH );
printer.PrintText("驛篔填顛鄧禱瀆吞囊剝潑醱屏并麵萊屢藹蠟汙¥n",
    ESCPOSConst.CMP_ALIGNMENT_LEFT, ESCPOSConst.CMP_FNT_DEFAULT,
    ESCPOSConst.CMP_TXT_1WIDTH | ESCPOSConst.CMP_TXT_1WIDTH );
```

2.4.4. ログ機能について(Androidのみ)

本 SDK は、メソッドの実行やプロパティの読み書きを記録するログ機能をサポートしています(Androidのみ)。ログ機能を設定する際は、SetLog メソッドを使用するか、次の書式の設定ファイル「CSJPOSLib.cfg」を SD カードの/temp/citizen に配置してください。

<CSJPOSLib.cfg の例>

[LogSetting]	・・・セクション名(固定)
LogMode=1	・・・ログモードを指定
LogPath=/temp/citizen	・・・ログファイルを格納する SD カードのフォルダを指定
LogMaxSize=10	・・・ログファイルの最大サイズを MB 単位で指定

設定項目

・ログモード

ログを記録するモードを指定します。

- 0: 記録なし
- 1: アクセス履歴の記録
- 2: エラーのみ記録

・格納フォルダ

ログファイルを格納するフォルダを指定します。本設定が指定されていない場合は、SD カードの/temp/citizen に格納されます。

・ログサイズ

ログファイルの最大容量を MB 単位で指定します。0 を指定した場合は容量制限が解除され可能な限り記録されます。

ログファイル名

ログファイルの拡張子は「.log」です。ファイル名は「CSJPOSLib」の後ろに曜日を表す数字が追加されます。曜日は日曜日を 0、月曜日を 1 として 0 から 6 のいずれかの数字が加えられます。

例) CSJPOSLib_1.log

ログファイルが既に存在し、それが当日以外の場合は、古いファイルを削除してからログを記録します。

ログフォーマット

ログ機能は、メソッド、プロパティの日付、時間、結果のアクセス情報を記録します。

--- メソッドの例1 (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)
```

--- メソッドの例2 (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)
```

--- プロパティの設定例 ---

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

--- プロパティの参照例 ---

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

※ログ機能を使用する場合、全てのメソッドとプロパティのアクセス時にログファイルが更新されますので、SDK の処理が低下してしまうことがあります。

※次のような理由などでファイルの書き込みができない場合はログファイルの記録が行われません。このような場合エラーメッセージなどは表示されませんので、ご注意ください。

- ・書き込み禁止デバイスを指定した場合（書き込み禁止のメモリーカードなど）
- ・出力先に十分な領域が残っていない場合
- ・書き込み禁止のログファイルがある場合
- ・ファイルやフォルダのアクセス権がない場合
- ・他のアプリケーションがログファイルを使用している場合

2.4.5. 定数定義一覧

No	項目	定数名	型	値	説明
1	処理結果	CMP_SUCCESS	int	0	正常終了
		CMP_E_CONNECTED	int	1001	接続済み
		CMP_E_DISCONNECT	int	1002	未接続
		CMP_E_NOTCONNECT	int	1003	接続失敗
		CMP_E_CONNECT_NOTFOUND	int	1004	未対応機種
		CMP_E_OFFLINE	int	1005	プリンター状態異常
		CMP_E_NOCONTEXT	int	1006	コンテキスト情報指定無し
		CMP_E_BT_DISABLE	int	1007	Bluetooth デバイス設定無効
		CMP_E_BT_NODEVICE	int	1008	Bluetooth 接続デバイスが無い
		CMP_E_ILLEGAL	int	1101	未対応処理または無効パラメータ
		CMP_E_OFFLINE	int	1102	オフライン
		CMP_E_NOEXIST	int	1103	ファイルが存在しない
		CMP_E_FAILURE	int	1104	処理異常
		CMP_E_TIMEOUT	int	1105	書き込みタイムアウト
		CMP_E_NO_LIST	int	1106	プリンターが見つからない
		CMP_EPTR_COVER_OPEN	int	1201	プリンターのカバーオープン
		CMP_EPTR_REC_EMPTY	int	1202	用紙切れ
CMP_EPTR_BADFORMAT	int	1203	画像フォーマット異常		
CMP_EPTR_CMP_EPTR_TOOBIG	int	1204	画像サイズが大きすぎる		
2	接続インターフェース	CMP_PORT_WiFi	int	0	ネットワーク接続
		CMP_PORT_Bluetooth	int	1	Bluetooth 接続
		CMP_PORT_Bluetooth_Insecure	int	2	Bluetooth 接続(Insecure 通信用)
		CMP_PORT_USB	int	3	USB 接続
3	ステータス	CMP_STS_NORMAL	int	0	ステータス正常
		CMP_STS_DRAWER_LEVEL_H	int	2	ドロワーキックコネクタ3番ピン状態 H
		CMP_STS_PAPER_NEAREMPTY	int	4	ニアエンプティ
		CMP_STS_BATTERY_LOW	int	8	バッテリー容量低下
		CMP_STS_COVER_OPEN	int	16	カバーオープン
		CMP_STS_PAPER_EMPTY	int	32	用紙切れ
		CMP_STS_MSR_READ	int	64	SR 読み取りモード状態
CMP_STS_PRINTEROFF	int	128	オフライン		
4	配置	CMP_ALIGNMENT_LEFT	int	0	左揃え
		CMP_ALIGNMENT_CENTER	Int	1	中揃え
		CMP_ALIGNMENT_RIGHT	int	2	右揃え
5	文字属性	CMP_FNT_DEFAULT	int	0	標準フォント
		CMP_FNT_FONTB	int	1	フォント B
		CMP_FNT_FONTC	int	2	フォント C
		CMP_FNT_BOLD	int	8	太字
		CMP_FNT_REVERSE	int	16	反転
		CMP_FNT_UNDERLINE	int	128	下線
		CMP_FNT_ITALIC	int	256	イタリック
CMP_FNT_STRIKEOUT	int	512	打ち消し		
6	文字サイズ	CMP_TXT_1WIDTH	int	0	幅 1 倍
		CMP_TXT_2WIDTH	int	16	幅 2 倍
		CMP_TXT_3WIDTH	int	32	幅 3 倍
		CMP_TXT_4WIDTH	int	48	幅 4 倍

		CMP_TXT_5WIDTH	int	64	幅 5 倍
		CMP_TXT_6WIDTH	int	80	幅 6 倍
		CMP_TXT_7WIDTH	int	96	幅 7 倍
		CMP_TXT_8WIDTH	int	112	幅 8 倍
		CMP_TXT_1HEIGHT	int	0	高さ 1 倍
		CMP_TXT_2HEIGHT	int	1	高さ 2 倍
		CMP_TXT_3HEIGHT	int	2	高さ 3 倍
		CMP_TXT_4HEIGHT	int	3	高さ 4 倍
		CMP_TXT_5HEIGHT	int	4	高さ 5 倍
		CMP_TXT_6HEIGHT	int	5	高さ 6 倍
		CMP_TXT_7HEIGHT	int	6	高さ 7 倍
		CMP_TXT_8HEIGHT	int	7	高さ 8 倍
7	サイド	CMP_SIDE_RIGHT	int	0	右側
		CMP_SIDE_LEFT	int	1	左側
8	画像幅	CMP_BM_ASIS	int	-11	1ドット当たり1ピクセル
9	画像モード	CMP_BM_MODE_CMD_RASTER	int	1	モノクロ印刷(ラスタコマンド)
		CMP_BM_MODE_CMD_BITIMAGE	int	2	モノクロ印刷(ビットイメージコマンド)
		CMP_BM_MODE_CMD_MONO	int	8	モノクロ登録
		CMP_BM_MODE_CMD_GRAY16	int	8	グレースケール印刷/登録
		CMP_BM_MODE_HT_THRESHOLD	int	16	ハーフトーン(しきい値)
		CMP_BM_MODE_HT_DITHER	int	32	ハーフトーン(ディザ)
		CMP_BM_MODE_CMD_GRAY16DOWNLOAD	int	256	グレースケール印刷(ダウンロードグラフィックスコマンド)
10	バーコード種類	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	Code30
		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	可視コード	CMP_HRI_TEXT_NONE	int	0	なし
		CMP_HRI_TEXT_ABOVE	int	1	バーコードの上
		CMP_HRI_TEXT_BELOW	int	2	バーコードの下
12	エラー修正レベル(PDF417)	CMP_PDF417_EC_LEVEL_0	int	48	レベル 0
		CMP_PDF417_EC_LEVEL_1	int	49	レベル 1
		CMP_PDF417_EC_LEVEL_2	int	50	レベル 2
		CMP_PDF417_EC_LEVEL_3	int	51	レベル 3
		CMP_PDF417_EC_LEVEL_4	int	52	レベル 4

		CMP_PDF417_EC_LEVEL_5	int	53	レベル 5
		CMP_PDF417_EC_LEVEL_6	int	54	レベル 6
		CMP_PDF417_EC_LEVEL_7	int	55	レベル 7
		CMP_PDF417_EC_LEVEL_8	int	56	レベル 8
13	エラー修正レベル(QR Code)	CMP_QRCODE_EC_LEVEL_L	int	48	レベル L(7%)
		CMP_QRCODE_EC_LEVEL_M	int	49	レベル M(15%)
		CMP_QRCODE_EC_LEVEL_Q	int	50	レベル Q(25%)
		CMP_QRCODE_EC_LEVEL_H	int	51	レベル H(30%)
14	カット種類	CMP_CUT_FULL	int	-1	フルカット
		CMP_CUT_PARTIAL	int	-2	パーシャルカット
		CMP_CUT_FULL_PREFEED	int	-3	カット位置送り後フルカット
		CMP_CUT_PARTIAL_PREFEED	int	-4	カット位置送り後パーシャルカット
15	マークフィード種類	CMP_MF_TO_CUTTER	int	2	フィードしてカット
		CMP_MF_TO_NEXT_TOF	int	8	次の用紙までフィード
16	ドロワー番号	CMP_DRAWER_1	int	1	ドロワー1
		CMP_DRAWER_2	int	2	ドロワー2
17	一括処理コントロール	CMP_TP_TRANSACTION	int	11	一括処理の開始
		CMP_TP_NORMAL	int	12	一括処理の印刷
18	回転方向コントロール	CMP_RT_NORMAL	int	0x0001	回転印刷の終了
		CMP_RT_ROTATE180	int	0x0103	倒立印刷の開始
		CMP_RP_BARCODE	int	0x1000	バーコード回転の開始
		CMP_RP_BITMAP	int	0x2000	画像回転の開始
19	ページモードコントロール	CMP_PM_PAGE_MODE	int	1	ページモードの開始
		CMP_PM_PRINT_SAVE	int	2	印刷とデータ保持
		CMP_PM_NORMAL	int	3	印刷とページモード終了
		CMP_PM_CANCEL	int	4	ページモードキャンセル
20	ページモード印刷方向	CMP_PD_LEFT_TO_RIGHT	int	1	通常印刷方向
		CMP_PD_BOTTOM_TO_TOP	int	2	左 90 度回転印刷
		CMP_PD_RIGHT_TO_LEFT	int	3	180 度回転印刷
		CMP_PD_TOP_TO_BOTTOM	int	4	右 90 度回転印刷
21	透かし印刷コントロール	CMP_WM_STOP	int	0	透かし印刷の終了
		CMP_WM_START	int	1	透かし印刷の開始
22	マップモード種類	CMP_MM_DOTS	int	1	POS プリンターのドット幅
		CMP_MM_TWIPS	int	2	1 インチの 1/1440
		CMP_MM_ENGLISH	int	3	0.001 インチ
		CMP_MM_METRIC	int	4	0.01 ミリメートル

3. ラインディスプレイ制御

3.1. プログラム構造

本 SDK を使用する場合のプログラム構造は、以下の通りです。

```

// インスタンス生成
ILineDisplay display = CrossCSJPOSLib.CreateLineDisplay();

// ディスプレイへ接続
int result = display.Connect(LineDisplayConst.CDP_PORT_WiFi,
    "192.168.0.10");
if (LineDisplayConst.CDP_SUCCESS == result)
{
    // エンコード設定
    display.SetEncoding("Shift_JIS");

    // 文字を消去
    display.ClearDisplay();

    // テキスト表示
    display.DisplayText("123456");

    // カーソルの位置指定
    display.SetCursorPosotion(1,2);

    // テキスト表示 (反転)
    display.DisplayText("123456",true);

    // 切断処理
    display.Disconnect();
}
else
{
    // 接続エラー
    DisplayAlert("Citizen_POS_sample1", "Connect or LineDisplay Error : "
        + result.ToString(), "OK");
}

```

クラス定義

接続処理

ディスプレイ処理

切断処理

3.2. 機能一覧

本 SDK は以下の機能を提供します。

メソッド一覧

No	機能	詳細
1	接続 (Connect メソッド)	ディスプレイと接続します。
2	切断 (Disconnect メソッド)	ディスプレイとの接続を切断します。
3	文字列表示 (DisplayText メソッド)	文字を表示します。
4	表示クリア (ClearDisplay メソッド)	表示文字を消去します。
5	点滅表示 (BlinkDisplay メソッド)	点滅表示させます。
6	スクリーンモード設定 (SetDisplayMode メソッド)	ディスプレイモードを設定します。
7	ディスプレイ設定 (SetDisplayConfig メソッド)	各種設定を行います。
8	カーソル設定 (SetCursorPosition メソッド)	カーソルの位置指定をします。
9	カーソル移動 (MoveCursor メソッド)	カーソルを移動します。
10	カーソル型の設定 (SetCursorType メソッド)	カーソル位置を表示します。
11	初期化 (InitializeDisplay メソッド)	ディスプレイを初期化します。
12	コマンド送信 (DisplayData メソッド)	コマンドを送信します
13	文字エンコード指定 (SetEncoding メソッド)	文字エンコードを指定します。
14	コードページ指定 (SetCodePage メソッド)	コードページを指定します。
15	国際文字指定 (SetInternationalCharacterSet メソッド)	国際文字を指定します。
16	ディスプレイ状態確認 (DisplayCheck メソッド)	ディスプレイの接続状態を確認します。
17	SDK バージョン番号取得 (GetVersionCode メソッド)	バージョン番号を取得します。
18	SDK バージョン文字列取得 (GetVersionName メソッド)	バージョン文字列を取得します。

3.3. ライブラリインターフェース

本 SDK のインターフェースを以下に示します。

3.3.1. 戻り値

以降に示すメソッドは、下記の値を返します。

戻り値	説明
CDP_SUCCESS (0)	正常終了
CDP_E_CONNECTED (1001)	デバイスへ既に接続済みです。
CDP_E_DISCONNECT (1002)	デバイスへ接続していません。
CDP_E_NOTCONNECT (1003)	デバイスへ接続できませんでした。
CDP_E_CONNECT_NOTFOUND (1004)	デバイス接続後の対応機種確認に失敗しました。
CDP_E_CONNECT_OFFLINE (1005)	デバイス接続後のプリンター状態確認に失敗しました。
CDP_E_NOCONTEXT (1006)	コンテキストを指定していません。
CDP_E_BT_DISABLE (1007)	Bluetooth デバイス設定が無効です。
CDP_E_BT_NODEVICE (1008)	Bluetooth デバイスが見つかりませんでした。
CDP_E_ILLEGAL (1101)	サポートされていない処理または無効なパラメータ値です。
CDP_E_OFFLINE (1102)	デバイスがオフラインです。
CDP_E_FAILURE (1104)	要求された処理が実行できません。

3.3.2. コンストラクタ

形式

ILineDisplay

パラメータ

ありません。

説明

ライブラリのコンストラクタです。インスタンスを生成します。

戻り値

ありません。

使用例

```
ILineDisplay display = CrossCSJPOSLib.CreateLineDisplay();
```

3.3.3. Connect メソッド

形式

- 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)

パラメータ

パラメータの意味と設定可能な値は以下の通りです。

値	[IN/OUT]	意味	設定可能範囲
connectType	[IN]	接続されているプリンターの接続タイプ	CDP_PORT_WiFi CDP_PORT_Bluetooth CDP_PORT_Bluetooth_Insecure (Androidのみ) CDP_PORT_USB (iOSのみ)
addr	[IN]	接続先されているプリンターのIP アドレス、 または Bluetooth デバイスアドレス または Bluetooth デバイス名 またはシリアル番号	WiFi : 0.0.0.0~255.255.255.255 Bluetooth : 00:00:00:00:00:00~FF:FF:FF:FF:FF:FF デバイス名(自動検出) USB : 空白またはシリアル番号
port	[IN]	接続先ポート番号	WiFi : ポート番号 USB : 1~3
timeout	[IN]	タイムアウト (msec)	

説明

このメソッドは、ラインディスプレイと接続するために使用します。ラインディスプレイが接続されているプリンターの接続タイプとアドレスを指定してください。

Bluetooth デバイスアドレスの英字は、大文字で指定してください。

Bluetooth デバイス名が指定された場合は、一致するペアリング済みのデバイスを自動検出します。

アドレスが省略された場合は、ペアリング済みの対応機種デバイスを自動検出します。

Android 2.3.3 以降に用意される Bluetooth デバイスの Insecure 通信を使用する場合は、接続タイプに CDP_PORT_Bluetooth_Insecure を指定してください。

USB 接続の場合、アドレスに空白を指定した場合は自動的に接続されます。プリンターのシリアル番号を指定して特定のプリンターへ接続する事も可能です。

接続先ポート番号は、接続タイプに Wi-Fi または USB を指定した場合のみ有効です。省略された場合は、WiFi の場合はポートの 9200 番、USB の場合はストリームの 1 番で接続します。

タイムアウトは、ラインディスプレイへの接続の最大時間(ミリ秒単位)を指定します。省略された場合は、WiFi 使用時に 4000 ミリ秒、Bluetooth 使用時に 8000 ミリ秒で接続します。

ラインディスプレイとの通信が不要になった場合は、必ず [Disconnect メソッド](#) を実行し、ラインディスプレイとの接続を切断してください。切断しなかった場合は、次の接続がエラーとなります。

戻り値

成功時は CDP_SUCCESS(0) を返します。失敗時は下記のエラーコードの説明を確認してください。

それ以外のエラーコードは「[3.3.1.戻り値](#)」を参照してください。

エラーコード	説明
CDP_E_NOTCONNECT (1003)	ラインディスプレイへ接続できませんでした。 ①ラインディスプレイが未接続

	②プリンターの電源が入っていない ③インターフェースポートのハンドルを取得できない
CDP_E_BT_DISABLE (1007)	Bluetooth デバイス設定が無効です。
CDP_E_BT_NODEVICE (1008)	Bluetooth デバイスが見つかりませんでした。

使用例

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

3.3.4. Disconnect メソッド

形式

```
int Disconnect ()
```

パラメータ

ありません。

説明

このメソッドは、ラインディスプレイとの接続を切断するために使用します。
ラインディスプレイの終了、あるいは、何らかのエラーが発生した場合は、本メソッドを実行して接続を切断してください。

戻り値

成功時は CDP_SUCCESS(0) を返します。それ以外のエラーコードは「[3.3.1.戻り値](#)」を参照してください。

使用例

```
display.Disconnect ();
```

3.3.5. DisplayText メソッド

形式

int DisplayText (string data, boolean reverseFlag)

パラメータ

パラメータの意味と設定可能な値は以下の通りです。

値	意味	設定可能範囲
data	テキストデータ	string
reverseFlag	反転指定フラグ	false: 標準 true: 反転 引数省略時には、false として扱います。

説明

このメソッドは、現在のカーソル位置からテキストを表示するために使用します。
テキスト属性は、反転を指定可能です。

戻り値

成功時は CDP_SUCCESS(0) を返します。それ以外のエラーコードは「[3.3.1.戻り値](#)」を参照してください。

使用例

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

3.3.6. ClearDisplay メソッド

形式

int ClearDisplay (int displayArea)

パラメータ

パラメータの意味と設定可能な値は以下の通りです。

値	意味	設定可能範囲
displayArea	消去領域	CDP_AREA_ALL(0): 全領域 CDP_AREA_CURSORLINE(1): カーソル行 引数省略時には、CDP_AREA_ALL として扱います。

説明

このメソッドは、表示中の文字を消去します。

戻り値

成功時は CDP_SUCCESS(0) を返します。それ以外のエラーコードは「[3.3.1.戻り値](#)」を参照してください。

使用例

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

3.3.7. BlinkDisplay メソッド

形式

int BlinkDisplay (int intervalBlink)

パラメータ

パラメータの意味と設定可能な値は以下の通りです。

値	意味	設定可能範囲
intervalBlink	点滅間隔(ミリ秒)	0~

説明

このメソッドは、表示画面全体を点滅させます。

点滅間隔(ミリ秒)は、点灯と消灯の間隔を指定します。点滅間隔に、0 を指定すると、点滅は解除されます。

戻り値

成功時は CDP_SUCCESS(0) を返します。それ以外のエラーコードは「[3.3.1.戻り値](#)」を参照してください。

使用例

```
display.BlinkDisplay(1000);
```

3.3.8. SetDisplayMode メソッド

形式

```
int SetDisplayMode (int displayMode)
```

パラメータ

パラメータの意味と設定可能な値は以下の通りです。

値	意味	設定可能範囲
displayMode	ディスプレイモード	CDP_OVERWRITE(1): 上書きモード CDP_VERTICALSCROLL(2): 垂直スクロールモード CDP_HORIZONTALSCROLL(3): 水平スクロールモード

説明

このメソッドは、以下のディスプレイのモードを設定します。

DisplayMode	概要
Overwrite	カーソル位置の文字を上書きし、カーソルを右に移動。 (カーソルが上端右端のときの入力は、カーソルを下端左端に移動、 カーソルが下端右端のときの文字入力は、カーソルを上端左端に移動)
VerticalScroll	カーソルが上端のときのカーソル上移動(または、左端での左移動)で、上端の表示行を下端にスクロール カーソルが下端のときのカーソル下移動(または、右端での右移動)で、下端の表示行を上端にスクロール
HorizontalScroll	カーソルが右端でのカーソル右移動(または、文字入力)で、現在のカーソル行に対して、左方向に文字をスクロール カーソル左端のカーソル左移動で、現在のカーソル行に対して、右方向に文字をスクロール

戻り値

成功時は CDP_SUCCESS(0) を返します。それ以外のエラーコードは「[3.3.1.戻り値](#)」を参照してください。

使用例

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

3.3.9. SetDisplayConfig メソッド

形式

int SetDisplayConfig (int brightness)

パラメータ

パラメータの意味と設定可能な値は以下の通りです。

値	意味	設定可能範囲
brightness	輝度 (%)	0~100

説明

このメソッドは、表示画面の輝度を変更します。

輝度は、数値が大きいくほど明るくなります。0を指定すると、画面を消灯します(表示内容は保持されます)。

設定後、表示画面全体の点滅は解除されます。

戻り値

成功時は CDP_SUCCESS(0) を返します。それ以外のエラーコードは「[3.3.1.戻り値](#)」を参照してください。

使用例

```
display.SetDisplayConfig(40);
```

3.3.10. SetCursorPosition メソッド

形式

```
int SetCursorPosition(int x, int y)
```

パラメータ

パラメータの意味と設定可能な値は以下の通りです。

値	意味	設定可能範囲
x	桁位置	1～
y	行位置	1～

説明

このメソッドは、カーソル位置を設定するために使用します。

カーソル位置は、カーソルの移動座標で、桁位置と行位置を指定します。

戻り値

成功時は CDP_SUCCESS(0) を返します。それ以外のエラーコードは「[3.3.1.戻り値](#)」を参照してください。

使用例

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

3.3.11. MoveCursor メソッド

形式

int MoveCursor (int dx, int dy)

パラメータ

パラメータの意味と設定可能な値は以下の通りです。

値	意味	設定可能範囲
dx	左右方向移動量	-128～127
dy	上下方向移動量	-128～127

説明

このメソッドは、カーソルを移動するために使用します。

現在のカーソル位置からの移動となります。カーソル移動量は、左右方向移動量(-:左方向, +:右方向)と上下方向移動量(-:上方向, +:下方向)を指定します。

戻り値

成功時は CDP_SUCCESS(0) を返します。それ以外のエラーコードは「[3.3.1.戻り値](#)」を参照してください。

使用例

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

3.3.12. SetCursorType メソッド

形式

int SetCursorType (int cursorType)

パラメータ

パラメータの意味と設定可能な値は以下の通りです。

要素	意味	設定可能範囲
cursorType	カーソル型指定	CDP_TYPE_NONE(0): カーソル非表示 CDP_TYPE_UNDERLINE(1): カーソル表示 (省略可能要素, 省略時は TYPE_UNDERLINE)

説明

現在のカーソル位置をディスプレイに表示します。

戻り値

成功時は CDP_SUCCESS(0) を返します。それ以外のエラーコードは「[3.3.1.戻り値](#)」を参照してください。

使用例

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

3.3.13. InitializeDisplay メソッド

形式

```
int InitializeDisplay ()
```

パラメータ

ありません。

説明

デバイスを初期化します。

戻り値

成功時は CDP_SUCCESS(0) を返します。それ以外のエラーコードは「[3.3.1.戻り値](#)」を参照してください。

使用例

```
display.InitializeDisplay();
```

3.3.14. DisplayData メソッド

形式

int DisplayData (byte[] data)

パラメータ

パラメータの意味と設定可能な値は以下の通りです。

要素	意味	設定可能範囲
data	送信データ	

説明

バイトデータをそのままデバイスに送信するために使用します。
ご使用の際は、他のメソッドに影響を与えない様に注意する必要があります。

戻り値

成功時は CDP_SUCCESS(0) を返します。それ以外のエラーコードは「[3.3.1.戻り値](#)」を参照してください。

使用例

```
// セルフテスト実行  
byte[] data = new byte[] {0x1f, 0x40};  
res = display.DisplayData(data);
```

3.3.15. SetEncoding メソッド

形式

int SetEncoding (string charset)

パラメータ

パラメータの意味と設定可能な値は以下の通りです。

値	意味	設定可能範囲
charset	文字セット名	サポートされる文字セット名は、動作環境により異なります。

説明

このメソッドは、プリンター送信データのエンコードを設定するために使用します。
インスタンスを生成時に OS のデフォルト文字セットに初期化します。

戻り値

成功時は CDP_SUCCESS(0) を返します。それ以外のエラーコードは「[3.3.1.戻り値](#)」を参照してください。

使用例

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

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

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

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

3.3.16. SetCodePage メソッド

形式

```
int SetCodePage (int codePage)
```

パラメータ

パラメータの意味と設定可能な値は以下の通りです。

要素	意味	設定可能範囲
codePage	コードページ指定	0~255

説明

設定値につきましては、利用デバイスのコマンドリファレンス”ESC t” コマンドを参照してください。

戻り値

成功時は CDP_SUCCESS(0) を返します。それ以外のエラーコードは「[3.3.1.戻り値](#)」を参照してください。

使用例

```
display.SetCodePage (1);
```

3.3.17. SetInternationalCharacterSet メソッド

形式

int SetInternationalCharacterSet (int characterSet)

パラメータ

パラメータの意味と設定可能な値は以下の通りです。

値	意味	設定可能範囲
characterSet	国際文字指定	0～16

説明

以下の国際文字セットを設定します。

CharacterSet	国際文字セット	CharacterSet	国際文字セット
0	アメリカ	9	ノルウェー
1	フランス	10	デンマークⅡ
2	ドイツ	11	スペインⅡ
3	イギリス	12	ラテンアメリカ
4	デンマークⅠ	13	韓国
5	スウェーデン	14	クロアチア
6	イタリア	15	中国
7	スペインⅠ	16	ベトナム
8	日本		

戻り値

成功時は CDP_SUCCESS(0) を返します。それ以外のエラーコードは「[3.3.1.戻り値](#)」を参照してください。

使用例

```
display.SetInternationalCharacterSet(8);
display.DisplayText("Total:¥¥1,010");
```

3.3.18. DisplayCheck メソッド

形式

```
int DisplayCheck ()
```

パラメータ

ありません。

説明

このメソッドは、ディスプレイの接続状態を確認するために使用します。

本メソッドの実行結果が成功の場合は、ラインディスプレイが接続されている事が確認できます。

本メソッドの実行結果が失敗の場合は、通信異常やデバイスの異常が発生した可能性があります。この場合、[Disconnect メソッド](#)および [Connect メソッド](#)を使用して再接続してください。

ネットワーク接続の場合、長時間放置すると自動的に切断されます。接続を保持する場合は、定期的に本メソッドを実行してください。

戻り値

成功時は CDP_SUCCESS(0) を返します。それ以外のエラーコードは「[3.3.1.戻り値](#)」を参照してください。

使用例

```
display.DisplayCheck();
```

3.3.19. GetVersionCode メソッド

形式

```
int GetVersionCode ()
```

パラメータ

ありません。

説明

このメソッドは、実行中のプラットフォームにおけるベースの SDK バージョン番号を数値で取得するために使用します。

戻り値

SDK のバージョン番号を数値 (Ver1.00 の場合:100) で取得します。

使用例

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

3.3.20. GetVersionName メソッド

形式

```
string GetVersionName ()
```

パラメータ

ありません。

説明

このメソッドは、実行中のプラットフォームにおけるベースの SDK バージョン番号を文字列で取得するために使用します。

戻り値

SDK のバージョン番号を文字列 (Ver1.00 の場合: "1.00") で取得します。

使用例

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

3.3.21. SetLog メソッド (Android のみ)

形式

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

パラメータ

パラメータの意味と設定可能な値は以下の通りです。

値	[IN/OUT]	意味	設定可能範囲
mode	[IN]	ログモード	0 : 記録なし 1 : アクセス履歴の記録 2 : エラーのみ記録
path	[IN]	格納フォルダ	SD カードのフォルダ
maxSize	[IN]	ログサイズ	0: サイズ制限なし 1~: 最大サイズ(MB)

説明

このメソッドは、ログ機能を設定するために使用します。ログ機能の詳細は「[3.4.1 ログ機能について](#)」を参照してください。

戻り値

ありません。

使用例

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

3.4. 注意事項

本 SDK の注意事項を以下に示します。

3.4.1. ログ機能について(Androidのみ)

本 SDK は、メソッドの実行やプロパティの読み書きを記録するログ機能をサポートしています(Androidのみ)。ログ機能を設定する際は、SetLog メソッドを使用するか、次の書式の設定ファイル「CSJPOSLibD.cfg」を SD カードの/temp/citizen に配置してください。

<CSJPOSLibD.cfg の例>

```
[LogSetting]          ...セクション名(固定)
LogMode=1            ...ログモードを指定
LogPath=/temp/citizen ...ログファイルを格納する SD カードのフォルダを指定
LogMaxSize=10        ...ログファイルの最大サイズを MB 単位で指定
```

設定項目

・ログモード

ログを記録するモードを指定します。

- 0: 記録なし
- 1: アクセス履歴の記録
- 2: エラーのみ記録

・格納フォルダ

ログファイルを格納するフォルダを指定します。本設定が指定されていない場合は、SD カードの /temp/citizen に格納されます。

・ログサイズ

ログファイルの最大容量を MB 単位で指定します。0 を指定した場合は容量制限が解除され可能な限り記録されます。

ログファイル名

ログファイルの拡張子は「.log」です。ファイル名は「CSJPOSLibD」の後ろに曜日を表す数字が追加されます。曜日は日曜日を 0、月曜日を 1 として 0 から 6 のいずれかの数字が加えられます。

例) CSJPOSLibD_1.log

ログファイルが既に存在し、それが当日以外の場合は、古いファイルを削除してからログを記録します。

ログフォーマット

ログ機能は、メソッド、プロパティの日付、時間、結果のアクセス情報を記録します。

--- メソッドの例1 (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)
```

--- メソッドの例2 (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)
```

※ログ機能を使用する場合、全てのメソッドとプロパティのアクセス時にログファイルが更新されますので、SDK の処理が低下してしまうことがあります。

※次のような理由などでファイルの書き込みができない場合はログファイルの記録が行われません。このような場合エラーメッセージなどは表示されませんので、ご注意ください。

- ・書き込み禁止デバイスを指定した場合（書き込み禁止のメモリーカードなど）
- ・出力先に十分な領域が残っていない場合
- ・書き込み禁止のログファイルがある場合
- ・ファイルやフォルダのアクセス権がない場合
- ・他のアプリケーションがログファイルを使用している場合

3.4.2. 定数定義一覧

No	項目	定数名	型	値	説明
1	処理結果	CDP_SUCCESS	int	0	正常終了
		CDP_E_CONNECTED	int	1001	接続済み
		CDP_E_DISCONNECT	int	1002	未接続
		CDP_E_NOTCONNECT	int	1003	接続失敗
		CDP_E_CONNECT_NOTFOUND	int	1004	未対応機種
		CDP_E_OFFLINE	int	1005	プリンター状態異常
		CDP_E_NOCONTEXT	int	1006	コンテキスト情報指定無し
		CDP_E_ILLEGAL	int	1101	未対応処理または無効パラメータ
		CDP_E_OFFLINE	int	1102	オフライン
		CDP_E_FAILURE	int	1104	処理異常
2	接続インターフェース	CDP_PORT_WiFi	int	0	ネットワーク接続
		CDP_PORT_Bluetooth	int	1	Bluetooth 接続
		CDP_PORT_Bluetooth_Insecure	int	2	Bluetooth 接続(Insecure 通信用)
		CDP_PORT_USB	int	3	USB 接続
3	消去領域	CDP_AREA_ALL	int	0	全領域
		CDP_AREA_CURSORLINE	int	1	カーソル行
4	ディスプレイモード	CDP_OVERWRITE	int	1	上書きモード
		CDP_VERTICALSCROLL	int	2	垂直スクロールモード
		CDP_HORIZONTALSCROLL	int	3	水平スクロールモード
5	カーソル型指定	CDP_TYPE_NONE	int	0	カーソル非表示
		CDP_TYPE_UNDERLINE	int	1	カーソル表示

4. バーコードスキャナー制御

4.1. プログラム構造

本 SDK を使用する場合のプログラム構造は、以下の通りです。

```
// インスタンス生成
IScanner scanner = CrossCSJPOSLib.CreateScanner();

// データイベント定義
void DataEventHandler(byte[] data)
{
    Console.WriteLine("Data call back:" + Encoding.UTF8.GetString(data));
}

// ステータスイベント定義
void StatusUpdateEventHandler(int status)
{
    Console.WriteLine("Status update call back:" + status);
}

// スキャン開始
void StartScan()
{
    // イベントハンドラ設定
    scanner.SetDataEventCallback(DataEventHandler);
    scanner.SetStatusUpdateEventCallback(StatusUpdateEventHandler);

    // スキャナー接続
    int result = scanner.Connect(ScannerConst.CSC_PORT_ WiFi,
        "192.168.0.10");
}

// スキャン終了
void StopScan()
{
    // スキャナー切断
    scanner.Disconnect();
}
```

クラス定義

コールバック処理

接続処理

切断処理

4.2. 機能一覧

本 SDK は以下の機能を提供します。

メソッド一覧

No	機能	詳細
1	接続 (Connect メソッド)	スキャナーと接続します。
2	切断 (Disconnect メソッド)	スキャナーとの接続を切断します。
3	データ読み取りコールバックメソッド登録 (SetDataEventCallback メソッド)	読取データコールバックメソッドを登録します。
4	ステータス変化コールバックメソッド登録 (SetStatusUpdateEventCallback メソッド)	ステータス変化コールバックメソッドを登録します。
5	SDK バージョン番号取得 (GetVersionCode メソッド)	バージョン番号を取得します。
6	SDK バージョン文字列取得 (GetVersionName メソッド)	バージョン文字列を取得します。

4.3. ライブラリインターフェース

本 SDK のインターフェースを以下に示します。

4.3.1. 戻り値

以降に示すメソッドは、下記の値を返します。

戻り値	説明
CSC_SUCCESS (0)	正常終了
CSC_E_CONNECTED (1001)	デバイスへ既に接続済みです。
CSC_E_DISCONNECT (1002)	デバイスへ接続していません。
CSC_E_NOTCONNECT (1003)	デバイスへ接続できませんでした。
CSC_E_CONNECT_NOTFOUND (1004)	デバイス接続後の対応機種確認に失敗しました。
CSC_E_CONNECT_OFFLINE (1005)	デバイス接続後のプリンター状態確認に失敗しました。
CSC_E_NOCONTEXT (1006)	コンテキストを指定していません。
CSC_E_BT_DISABLE (1007)	Bluetooth デバイス設定が無効です。
CSC_E_BT_NODEVICE (1008)	Bluetooth デバイスが見つかりませんでした。
CSC_E_ILLEGAL (1101)	サポートされていない処理または無効なパラメータ値です。
CSC_E_OFFLINE (1102)	デバイスがオフラインです。
CSC_E_FAILURE (1104)	要求された処理が実行できません。

4.3.2. コンストラクタ

形式

IScanner

パラメータ

ありません。

説明

ライブラリのコンストラクタです。インスタンスを生成します。

戻り値

ありません。

使用例

```
IScanner scanner = CrossCSJPOSLib.CreateScanner();
```

4.3.3. Connect メソッド

形式

- 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)

パラメータ

パラメータの意味と設定可能な値は以下の通りです。

値	[IN/OUT]	意味	設定可能範囲
connectType	[IN]	接続先されているプリンターの接続タイプ	CSC_PORT_WiFi CSC_PORT_Bluetooth CSC_PORT_Bluetooth_Insecure (Androidのみ) CDP_PORT_USB (iOSのみ)
addr	[IN]	接続先されているプリンターのIP アドレス、 または Bluetooth デバイスアドレス または Bluetooth デバイス名 またはシリアル番号	WiFi : 0.0.0.0~255.255.255.255 Bluetooth : 00:00:00:00:00:00~FF:FF:FF:FF:FF:FF デバイス名(自動検出)
port	[IN]	接続先ポート番号	
timeout	[IN]	タイムアウト (msec)	

説明

このメソッドは、バーコードスキャナーと接続するために使用します。バーコードスキャナーが接続されているプリンターの接続タイプとアドレスを指定してください。

Bluetooth デバイスアドレスの英字は、大文字で指定してください。

Bluetooth デバイス名が指定された場合は、一致するペアリング済みのデバイスを自動検出します。

アドレスが省略された場合は、ペアリング済みの対応機種デバイスを自動検出します。

Android 2.3.3 以降に用意される Bluetooth デバイスの Insecure 通信を使用する場合は、接続タイプに CSC_PORT_Bluetooth_Insecure を指定してください。

USB 接続の場合、アドレスに空白を指定した場合は自動的に接続されます。プリンターのシリアル番号を指定して特定のプリンターへ接続する事も可能です。

接続先ポート番号は、接続タイプに CSC_PORT_WiFi を指定した場合のみ有効です。省略された場合は、9210 番で接続します。

タイムアウトは、バーコードスキャナーへの接続の最大時間(ミリ秒単位)を指定します。省略された場合は、WiFi 使用時に 4000 ミリ秒、Bluetooth 使用時に 8000 ミリ秒で接続します。

バーコードスキャナーとの通信が不要になった場合は、必ず [Disconnect メソッド](#) を実行し、接続を切断してください。切断しなかった場合は、次の接続がエラーとなります。

戻り値

成功時は CSC_SUCCESS(0) を返します。失敗時は下記のエラーコードの説明を確認してください。

それ以外のエラーコードは「[4.3.1 戻り値](#)」を参照してください。

エラーコード	説明
CSC_E_NOTCONNECT (1003)	バーコードスキャナーへ接続できませんでした。 ①バーコードスキャナーが未接続 ②プリンターの電源が入っていない ③インターフェースポートのハンドルを取得できない
CSC_E_BT_DISABLE (1007)	Bluetooth デバイス設定が無効です。
CSC_E_BT_NODEVICE (1008)	Bluetooth デバイスが見つかりませんでした。

使用例

```
scanner.Connect(ScannerConst.CMP_PORT_WiFi, "192.168.0.10");
```

4.3.4. Disconnect メソッド

形式

```
int Disconnect ()
```

パラメータ

ありません。

説明

このメソッドは、バーコードスキャナーとの接続を切断するために使用します。
スキャンの終了、あるいは、何らかのエラーが発生した場合は、本メソッドを実行して接続を切断してください。

戻り値

成功時は CSC_SUCCESS(0) を返します。それ以外のエラーコードは「[4.3.1 戻り値](#)」を参照してください。

使用例

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

4.3.5. SetDataEventCallback メソッド

形式

```
int SetDataEventCallback(DataEventHandler listener)
```

パラメータ

パラメータの意味と設定可能な値は以下の通りです。

値	[IN/OUT]	意味	設定可能範囲
listener	[IN]	コールバックメソッド	

説明

バーコードスキャナーがバーコードから読み取った情報を受け取るコールバックメソッドを登録します。コールバックメソッドは、バーコードスキャナーがバーコードから読み取った情報として、byte 配列型の引き数を受け取ります。SetDataEventCallback メソッドを複数回実行した場合は、最後に指定されたコールバックメソッドに上書きされます。オブジェクト指定に null が設定された場合は、コールバックを解除します。

[コールバックインターフェイス]

```
public delegate void DataEventHandler(byte[] data);
```

戻り値

成功時は CSC_SUCCESS(0) を返します。それ以外のエラーコードは「[4.3.1 戻り値](#)」を参照してください。

使用例

```
// データイベント定義
void DataEventHandler (byte[] data)
{
    Console.WriteLine("Data call back:" + Encoding.UTF8.GetString(data));
}

// イベントハンドラ設定
scanner.SetDataEventCallback(DataEventHandler);
```

4.3.6. SetStatusUpdateEventCallback メソッド

形式

```
int SetStatusUpdateEventCallback(StatusUpdateEventHandler listener)
```

パラメータ

パラメータの意味と設定可能な値は以下の通りです。

値	[IN/OUT]	意味	設定可能範囲
listener	[IN]	コールバックメソッド	

説明

デバイスのステータス変化情報を受け取るコールバックメソッドを登録します。コールバックメソッドは、デバイスのステータス変化の情報として、ステータスを示す int 型の引き数を受け取ります。

SetStatusUpdateEventCallback メソッドを複数回実行した場合は、最後に指定されたコールバックメソッドに上書きされます。

オブジェクト指定に null が設定された場合は、コールバックを解除します。

[コールバックインターフェイス]

```
public delegate void StatusUpdateEventHandler(int status);
```

ステータスコード	説明
CSC_SUE_POWER_ONLINE (2001)	デバイスはレディ状態です
CSC_SUE_POWER_OFF (2002)	通信異常または本体に接続されていません

戻り値

成功時は CSC_SUCCESS(0) を返します。それ以外のエラーコードは「[4.3.1 戻り値](#)」を参照してください。

使用例

```
// ステータスイベント定義
void StatusUpdateEventHandler(int status)
{
    Console.WriteLine("Status update call back:" + status);
}

// イベントハンドラ設定
scanner.SetStatusUpdateEventCallback(StatusUpdateEventHandler);
```

4.3.7. GetVersionCode メソッド

形式

GetVersionCode ()

戻り値

バージョン番号 : Number

パラメータ

ありません。

説明

このメソッドは、実行中のプラットフォームにおけるベースの SDK バージョン番号を数値で取得するために使用します。

戻り値

SDK のバージョン番号を数値 (Ver1.23 の場合:123) で取得します。

使用例

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

4.3.8. GetVersionName メソッド

形式

GetVersionName ()

戻り値

バージョン文字列 : string

パラメータ

ありません。

説明

このメソッドは、実行中のプラットフォームにおけるベースの SDK バージョン番号を文字列で取得するために使用します。

戻り値

SDK のバージョン番号を文字列 (Ver1.00 の場合 : "1.00") で取得します。

使用例

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

4.3.9. SetLog メソッド (Android のみ)

形式

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

パラメータ

パラメータの意味と設定可能な値は以下の通りです。

値	[IN/OUT]	意味	設定可能範囲
mode	[IN]	ログモード	0 : 記録なし 1 : アクセス履歴の記録 2 : エラーのみ記録
path	[IN]	格納フォルダ	SD カードのフォルダ
maxSize	[IN]	ログサイズ	0: サイズ制限なし 1~: 最大サイズ(MB)

説明

このメソッドは、ログ機能を設定するために使用します。ログ機能の詳細は「[4.4.1 ログ機能について](#)」を参照してください。

戻り値

ありません。

使用例

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

4.4. 注意事項

本 SDK の注意事項を以下に示します。

4.4.1. ログ機能について(Androidのみ)

本 SDK は、メソッドの実行やプロパティの読み書きを記録するログ機能をサポートしています。ログ機能を設定する際は、[SetLog メソッド](#)を使用するか、次の書式の設定ファイル「CSJPOSLibS.cfg」を SD カードの /temp/citizen に配置してください。

<CSJPOSLibS.cfg の例>

```
[LogSetting]          ...セクション名(固定)
LogMode=1            ...ログモードを指定
LogPath=/temp/citizen ...ログファイルを格納する SD カードのフォルダを指定
LogMaxSize=10        ...ログファイルの最大サイズを MB 単位で指定
```

設定項目

・ログモード

ログを記録するモードを指定します。

- 0: 記録なし
- 1: アクセス履歴の記録
- 2: エラーのみ記録

・格納フォルダ

ログファイルを格納するフォルダを指定します。本設定が指定されていない場合は、SD カードの /temp/citizen に格納されます。

・ログサイズ

ログファイルの最大容量を MB 単位で指定します。0 を指定した場合は容量制限が解除され可能な限り記録されます。

ログファイル名

ログファイルの拡張子は「.log」です。ファイル名は「CSJPOSLibS」の後ろに曜日を表す数字が追加されます。曜日は日曜日を 0、月曜日を 1 として 0 から 6 のいずれかの数字が加えられます。

例) CSJPOSLibD_1.log

ログファイルが既に存在し、それが当日以外の場合は、古いファイルを削除してからログを記録します。

ログフォーマット

ログ機能は、メソッド、プロパティの日付、時間、結果のアクセス情報を記録します。

```
--- メソッドの例1 (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)
```

```
--- メソッドの例2 (DataEvent) ---
```

```
2018/06/29 12:58:57.956 4481 EVENT      DataEvent : 31 32 33 34 35 36 37 38 39 30 31 32
```

※ログ機能を使用する場合、全てのメソッドとプロパティのアクセス時にログファイルが更新されますので、SDK の処理が低下してしまうことがあります。

※次のような理由などでファイルの書き込みができない場合はログファイルの記録が行われません。このような場合エラーメッセージなどは表示されませんので、ご注意ください。

- ・書き込み禁止デバイスを指定した場合（書き込み禁止のメモリーカードなど）
- ・出力先に十分な領域が残っていない場合
- ・書き込み禁止のログファイルがある場合
- ・ファイルやフォルダのアクセス権がない場合
- ・他のアプリケーションがログファイルを使用している場合

4.4.2. 定数定義一覧

No	項目	定数名	型	値	説明
1	処理結果	CSC_SUCCESS	int	0	正常終了
		CSC_E_CONNECTED	int	1001	接続済み
		CSC_E_DISCONNECT	int	1002	未接続
		CSC_E_NOTCONNECT	int	1003	接続失敗
		CSC_E_CONNECT_NOTFOUND	int	1004	未対応機種
		CSC_E_OFFLINE	int	1005	プリンター状態異常
		CSC_E_NOCONTEXT	int	1006	コンテキスト情報指定無し
		CSC_E_ILLEGAL	int	1101	未対応処理または無効パラメータ
		CSC_E_OFFLINE	int	1102	オフライン
		CSC_E_FAILURE	int	1104	処理異常
2	接続インターフェース	CSC_PORT_WiFi	int	0	ネットワーク接続
		CSC_PORT_Bluetooth	int	1	Bluetooth 接続
		CSC_PORT_Bluetooth_Insecure	int	2	Bluetooth 接続(Insecure 通信用)
		CSC_PORT_USB	int	3	USB 接続
3	ステータス	CSC_SUE_POWER_ONLINE	int	2001	レディ状態
		CSC_SUE_POWER_OFF	int	2002	通信異常または本体に接続されていない

CITIZEN Xamarin POS Print SDK プログラムマニュアル

2021/04/08 Ver.1.0.1.0 用

シチズン・システムズ株式会社

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