SII SDK for .NET MAUI
アプリケーションプログラマーズガイド

Rev.07

[対象製品]
SLP720RT シリーズ
SLP721RT シリーズ
RP-F10 シリーズ
RP-G10 シリーズ
RP-E10 シリーズ
RP-D10 シリーズ
MP-B30 シリーズ
MP-B30L シリーズ
MP-B21L シリーズ
MP-B20 シリーズ
MP-A40 シリーズ
DSP-A01 シリーズ

セイコーインスツル株式会社

Rev.01　2024 年　3 月
Rev.02　2024 年　5 月
Rev.03　2024 年　7 月
Rev.04　2024 年　11 月
Rev.05　2025 年　3 月
Rev.06　2026 年　3 月
Rev.07　2026 年　5 月

©セイコーインスツル株式会社　2024-2026
無断転載を禁じます。

Microsoft®、Windows®、および Microsoft Store は、米国およびその他の国における Microsoft Corporation の商標または登録商標です。
Android™は、Google LLC.の商標です。
Google Play および Google Play ロゴは、Google LLC の商標です。
App Store℠ は Apple Inc. のサービスマークです。
iPad®、iPad Air®、iPad mini®、iPhone®、iPod® は、Apple Inc. の米国およびその他の国および地域における商標です。
Bluetooth®ワードマークは、Bluetooth SIG, Inc. が所有する登録商標です。
IOS は、Cisco の米国およびその他の国における商標または登録商標であり、ライセンスに基づき使用されています。
その他の製品名および会社名は、各社の商標または登録商標です。

本書の内容は、断りなく変更することがあります。
本書及び本書に記載された製品の利用によって発生した
損害及びその回復に要する費用に対し、当社は一切の責任を負いかねます。
本書を第三者に無断で頒布することを禁じます。

はじめに

本書は、セイコーインスツル株式会社（以降、SII）が提供する、「SII SDK for .NET MAUI」（以降、SDK）について説明します。

対象デバイス

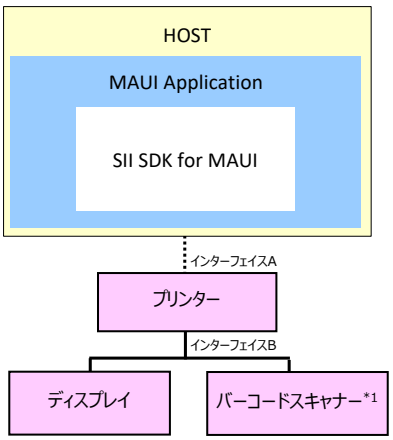
SDKでサポートするデバイスを下記に記載します。

iOS

プリンター	インターフェイスA	ディスプレイ	インターフェイスB
SLP720RTシリーズ	TCP/IP	–	–
SLP721RTシリーズ	Bluetooth	–	–
	TCP/IP		
RP-F10シリーズ	Bluetooth	DSP-A01シリーズ	USB
	USB		
	TCP/IP		
RP-E10シリーズ	Bluetooth	–	–
	TCP/IP*1		
RP-D10シリーズ	Bluetooth		
	TCP/IP*2		
MP-B30シリーズ MP-B30Lシリーズ	Bluetooth		
	TCP/IP		
MP-B21Lシリーズ	Bluetooth		
	TCP/IP		
MP-B20シリーズ	Bluetooth		
MP-A40シリーズ	Bluetooth		

*1： プリンターは、メインファームウェアVer. 1.11、LANインターフェースファームウェアVer. 1.13.01以降をご使用ください。

*2： プリンターは、メインファームウェアVer. 1.05、LANインターフェースファームウェアVer. 1.13.01以降をご使用ください。



構成イメージ

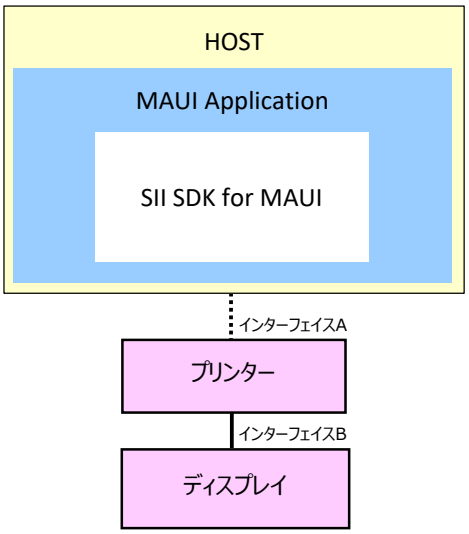
*1： バーコードスキャナーを含む周辺機器の組み合わせについては、「RP-F10シリーズ サーマルプリンタ取扱説明書」を参照してください。

Android

(1) ディスプレイを SII 製プリンターに接続して使用する場合

プリンター	インターフェイスA	ディスプレイ	インターフェイスB
SLP720RTシリーズ	USB	–	–
	TCP/IP		
SLP721RTシリーズ	Bluetooth	–	–
	USB		
	TCP/IP		
RP-F10シリーズ	Bluetooth	DSP-A01シリーズ*1	USB
	USB		
	TCP/IP		
RP-G10シリーズ	USB	–	–
RP-E10シリーズ	Bluetooth		
	USB		
	TCP/IP*2		
RP-D10シリーズ	Bluetooth		
	USB		
	TCP/IP*3		
MP-B30シリーズ MP-B30Lシリーズ	Bluetooth		
	USB		
	TCP/IP		
MP-B21Lシリーズ	Bluetooth		
	USB		
	TCP/IP		
MP-B20シリーズ	Bluetooth		
	USB		
MP-A40シリーズ	Bluetooth		
	USB		

*1: ディスプレイをSII製プリンターに接続して使用する場合は、"プリンター経由のDSP-A01"と記載します。
*2: プリンターは、メインファームウェアVer. 1.11、LANインターフェースファームウェアVer. 1.13.01以降をご使用ください。
*3: プリンターは、メインファームウェアVer. 1.05、LANインターフェースファームウェアVer. 1.13.01以降をご使用ください。

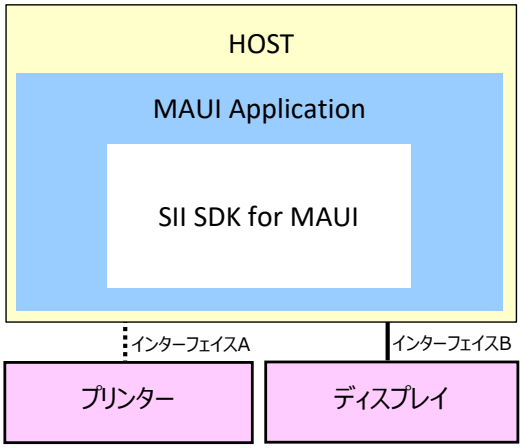


構成イメージ

(2) ディスプレイを単体で使用する場合

プリンター	インターフェイスA	ディスプレイ	インターフェイスB
–	–	DSP-A01シリーズ*1	USB

*1: ディスプレイを単体で使用する場合は、"DSP-A01単体"と記載します。



構成イメージ

用語

本書で使用する用語について下記に記載します。

プリンター

用語	内容
プリンター用技術説明書	下記の技術説明書。 ・SLP720RTシリーズ サーマルプリンタ 技術説明書 ・SLP721RTシリーズ サーマルプリンタ 技術説明書 ・RP-F10シリーズ サーマルプリンタ 技術説明書 ・RP-G10シリーズ サーマルプリンタ 技術説明書 ・RP-E10シリーズ サーマルプリンタ 技術説明書 ・RP-D10シリーズ サーマルプリンタ 技術説明書 ・MP-B30シリーズ サーマルプリンタ 技術説明書 ・MP-B30-x02JK2x サーマルプリンタ 技術説明書 ・MP-B30Lシリーズ サーマルプリンタ 技術説明書 ・MP-B30L-x46JK2x サーマルプリンタ 技術説明書 ・MP-B21Lシリーズ サーマルプリンタ 技術説明書 ・MP-B20-B02JK1 サーマルプリンタ 技術説明書 ・MP-A40シリーズ サーマルプリンタ 技術説明書
プリンターコマンド	「プリンター用技術説明書」に記載されている、プリンターを制御するための命令。
取扱説明書	下記の取扱説明書。 ・SLP720RTシリーズ サーマルプリンタ 取扱説明書 ・SLP721RTシリーズ サーマルプリンタ 取扱説明書 ・RP-F10シリーズ サーマルプリンタ 取扱説明書 ・RP-G10シリーズ サーマルプリンタ 取扱説明書 ・RP-E10シリーズ サーマルプリンタ 取扱説明書 ・RP-D10シリーズ サーマルプリンタ 取扱説明書 ・MP-B30シリーズ サーマルプリンタ 取扱説明書 ・MP-B30-x02JK2x サーマルプリンタ 取扱説明書 ・MP-B30Lシリーズ サーマルプリンタ 取扱説明書 ・MP-B30L-x46JK2x サーマルプリンタ 取扱説明書 ・MP-B21Lシリーズ サーマルプリンタ 取扱説明書 ・MP-B20-B02JK1 サーマルプリンタ 取扱説明書 ・MP-A40シリーズ サーマルプリンタ 取扱説明書

ディスプレイ

用語	内容
ディスプレイ用技術説明書	下記の技術説明書。 ・DSP-A01シリーズ カスタマーディスプレイ 技術説明書
ディスプレイコマンド	「ディスプレイ用技術説明書」に記載されている、ディスプレイを制御するための命令。
スライド	ディスプレイの画面サイズ（横480px×縦272px）のイメージデータ。スタンバイ画面としてディスプレイへの表示、及び背景画面としてテンプレートと重ね合わせてディスプレイに表示する。
テンプレート	描画領域や配置位置などの属性を設定できる要素を持つ定型フォーム。要素にはtext要素（テキストデータ）、img要素（イメージデータ）、barcode要素（バーコードデータ）、及びqr要素（QRコードデータ）がある。 テンプレートを登録する場合、配置する要素にそれぞれマップIDを定義する。 登録済みのイメージデータやテキストデータ等は、テンプレートを選択し、そのマップIDを指定し、画面更新することでディスプレイに表示ができる。 テンプレートのデータはXMLファイル形式で指定する必要がある。 登録できるテンプレートの最大サイズは横480px×縦272px。
マップID	テンプレートを登録する際に位置や修飾情報等を保持した要素に定義するID。
マクロ	複数のAPIを実行する順に登録し、イベントが発生した際に自動的に実行される機能。
イベント	ディスプレイコマンド「イベント通知」によって定義するイベント。

目次

1章　製品概要　1-1

1.1　SDKの提供機能 1-1
1.2　SDKの概要 1-1
1.2.1　SDKの構成 1-1
1.2.2　ライブラリの提供機能 1-2
1.2.3　SII製プリンターとBluetooth通信・USB通信を行う
iOSアプリケーションの開発 1-2
1.2.4　ディスプレイの工場出荷時の登録データ 1-2

2章　製品仕様　2-1

2.1　動作環境 2-1
2.1.1　iOS 2-1
2.1.2　Android 2-6
2.1.3　.NET MAUI 2-7
2.2　プリンターの設定 2-8
2.2.1　iOS 2-8
2.2.2　Android 2-15
2.3　注意事項 2-20
2.3.1　iOS 2-20
2.3.2　Android 2-20

3章　ライブラリの利用方法　3-1

3.1　iOS/Androidアプリケーションの開発環境 3-1
3.2　提供ファイル 3-2
3.3　Visual Studioへの組み込み 3-3
3.3.1　Windows 3-3
3.3.2　Mac 3-12
3.4　注意事項 3-20

4章　プリンターのライブラリ機能　4-1

4.1　スタンダードモードとページモード 4-1
4.1.1　基本動作 4-1
(1)　スタンダードモード 4-1
(2)　ページモード 4-2
4.1.2　スタンダードモードにおけるテキストデータの印字 4-3
4.1.3　ページモードにおける印字データの展開位置 4-4
(1)　印字領域 4-4
(2)　印字方向 4-4
(3)　基準点 4-5

4.1.4　ページモードの印字領域外の印字データ処理 4-6
4.2　ラベル印字機能 4-7
4.2.1　ラベルファイルの構造 4-7
(1)　オブジェクトの種類とライブラリでのサポート 4-8
(2)　ラベルファイルをライブラリで印字する際の注意点 4-8
①　全てのオブジェクト 4-8
②　テキストオブジェクト 4-8
③　イメージオブジェクト 4-8
④　バーコードオブジェクト 4-8
⑤　連絡帳オブジェクト 4-9
⑥　日付と時刻オブジェクト 4-9
⑦　図オブジェクト 4-9
4.2.2　ラベルファイルの利用手順 4-10
(1)　ラベルファイルをそのままライブラリから印字する場合 4-10
(2)　ラベルファイルのオブジェクトデータを置換して印字する場合 4-10
4.3　ログファイル出力機能 4-11
4.3.1　ログ出力の設定方法 4-11
4.3.2　ログ出力の設定内容 4-11
4.3.3　ログファイル 4-11
4.4　API リファレンス 4-12
Method or Property　メソッド名またはプロパティ名 4-13
4.4.1　列挙型定数一覧 4-14
①　デバイスモデル（DeviceModel） 4-14
②　ポート種別（PortType） 4-14
③　ドロワ番号（DrawerNum） 4-14
④　パルス幅（PulseWidth） 4-15
⑤　ブザーパターン（BuzzerPattern） 4-15
⑥　ディザリング（Dithering） 4-15
⑦　プリンター応答種別（PrinterResponseId） 4-16
⑧　デバイスタイプ（DeviceType） 4-16
⑨　一括処理選択（TransactionFunction） 4-16
⑩　位置揃え（PrintAlignment） 4-16
⑪　強調文字（CharacterBold） 4-17
⑫　アンダーライン（CharacterUnderline） 4-17
⑬　文字倍率（CharacterScale） 4-17
⑭　文字フォント（CharacterFont） 4-18
⑮　モジュールサイズ（ModuleSize） 4-18
⑯　エラー訂正レベル（ErrorCorrection） 4-22
⑰　白黒反転印字（CharacterReverse） 4-22
⑱　倒立印字(CharacterInversion) 4-22
⑲　バーコードシンボル（BarcodeSymbol） 4-23
⑳　HRI文字印字位置（HriPosition） 4-23
㉑　NW比（NwRatio） 4-24
㉒　PDF417シンボル（Pdf417Symbol） 4-24
㉓　QRコードモデル（QrModel） 4-24
㉔　Data Matrixモジュール（DataMatrixModule） 4-25
㉕　MaxiCodeモード（MaxiCodeMode） 4-26

㉖ Aztecシンボル（AztecSymbol） 4-26
㉗ カット方法（CuttingMethod） 4-26
㉘ 頭出し位置（FeedPosition） 4-27
㉙ イメージの回転方向(Rotate) 4-29
㉚ イメージの拡大縮小(ImageScale) 4-29
㉛ 印字方向(Direction) 4-29
㉜ 線種類(LineStyle) 4-30
㉝ 国際文字セット(InternationalCharacter) 4-30
㉞ コードページ(CodePage) 4-31
㉟ ラベルファイル印刷時マークあり・なし用紙選択 4-31
4.4.2 PrinterManagerクラス 4-32
(1) メソッド一覧 4-32
① スタンダードモード・ページモード共通メソッド 4-32
② スタンダードモード専用メソッド 4-33
③ ページモード専用メソッド 4-34
(2) スタンダードモード・ページモード共通プロパティ一覧 4-35
(3) 定数一覧 4-36
① バーコード、PDF417専用 4-36
② バッテリ残量レベル 4-36
(4) メソッド詳細 4-37
① スタンダードモード・ページモード共通メソッド 4-37
PrinterManager コンストラクタ 4-37
Connect デバイスとの通信開始 4-37
Disconnect デバイスとの通信切断 4-39
OpenDrawer キャッシュドロワを開く 4-39
Buzzer ブザーの鳴動 4-40
ExternalBuzzer 外部ブザーの鳴動 4-40
GetStatus プリンターステータスの取得 4-41
Abort プリンターのデータ待ち状態解除 4-42
RegisterLogo ロゴの登録 4-42
UnregisterLogo ロゴの登録削除 4-43
RegisterStyleSheet スタイルシートの登録 4-44
UnregisterStyleSheet スタイルシートの登録削除 4-45
ResetPrinter プリンターのリセット 4-45
GetPrinterResponse プリンターからの各種応答取得 4-46
StartDiscoveryDevice デバイスの探索開始 4-48
CancelDiscoveryDevice
デバイスの探索中断 4-49
GetFoundDevice 発見されたデバイス情報リストの取得 4-49
ControlTransaction 一括処理の開始・終了 4-49
SetStatusChangedEventHandler
プリンターステータス変化のコールバック開始・終了
..... 4-51
SetBarcodeScannerReadDataEventHandler
バーコードデータ受信のコールバック開始・終了
..... 4-52

SetBarcodeScannerChangedOnlineEventHandler
バーコードスキャナー接続のコールバック開始・終了 4-52
SetBarcodeScannerChangedOfflineEventHandler
バーコードスキャナー切断のコールバック開始・終了 4-53
② スタンダードモード専用メソッド 4-54
SendText テキストデータの送信 4-54
SendTextEx 書式指定テキストデータの送信 4-54
PrintBarcode バーコードの印字 4-56
PrintPDF417 PDF417の印字 4-60
PrintQRcode QRコードの印字 4-61
PrintDataMatrix Data Matrixの印字 4-62
PrintMaxiCode MaxiCodeの印字 4-62
PrintGS1DataBarStacked
GS1 Databar Stackedの印字 4-63
PrintGS1DataBarStackedOmnidirectional
GS1 Databar Stacked Omni-directionalの印字 .. 4-64
PrintGS1DataBarExpandedStacked
GS1 Databar Expanded Stackedの印字 4-64
PrintAztecCode Aztec Codeの印字 4-65
CutPaper 用紙のカット 4-66
FeedPosition 用紙の頭出し 4-66
SendBinary バイナリデータの送信 4-67
SendDataFile 指定ファイルの送信 4-67
SendImage イメージの送信 4-69
PrintPDF PDFページの印字 4-69
PrintLogo ロゴの印字 4-71
PrintSmartLabelImageData
ラベルの印字 4-72
③ ページモード専用メソッド 4-73
EnterPageMode ページモードの開始 4-74
ExitPageMode ページモードの終了 4-74
SetPageModeArea ページモードの印字領域の指定 4-74
SetPageModeDirection ページモードの印字方向の指定 4-76
SetPageModeLineSpacing
ページモードの改行量の指定 4-76
PrintPageMode ページモードの印字 4-76
PrintPageModeText ページモードのテキストデータの送信 4-77
PrintPageModeTextEx ページモードの書式指定テキストデータの送信 .. 4-77
PrintPageModeBarcode ページモードのバーコードの印字 4-78
PrintPageModePDF417 ページモードのPDF417の印字 4-83
PrintPageModeQRcode ページモードのQRコードの印字 4-84
PrintPageModeDataMatrix
ページモードのData Matrixの印字 4-85
PrintPageModeMaxiCode
ページモードのMaxiCodeの印字 4-86

PrintPageModeGS1DataBarStacked
ページモードのGS1 Databar Stackedの印字.....4-87
PrintPageModeGS1DataBarStackedOmnidirectional
ページモードのGS1 Databar Stacked
Omni-directionalの印字.....4-87
PrintPageModeGS1DataBarExpandedStacked
ページモードのGS1 Databar Expanded Stackedの印字
.....4-88
PrintPageModeAztecCode
ページモードのAztec Codeの印字.....4-89
SendPageModeBinary ページモードのバイナリデータの送信.....4-90
PrintPageModeImageFile
ページモードのイメージファイルの描画.....4-91
PrintPageModeRectangle
ページモードの矩形の描画.....4-92
PrintPageModeLine ページモードの罫線の印字.....4-93
PrintPageModeLogo ページモードのロゴの印字.....4-95
(5) スタンダードモード・ページモード共通プロパティ詳細 4-96
SendTimeout 送信タイムアウト時間の設定/取得.....4-96
ReceiveTimeout 受信タイムアウト時間の設定/取得.....4-96
InternationalCharacter
国際文字セットの設定/取得.....4-97
CodePage コードページの設定/取得.....4-97
DeviceModel デバイスモデルの取得.....4-97
PortType 接続中ポート種別の取得.....4-98
IsConnect プリンターとの通信状態の確認.....4-98
SocketKeepingTime ソケット維持時間の設定/取得.....4-98
Version SDKバージョンの取得.....4-99
PrintSmartLabelMode ラベルファイル印刷時の用紙の設定/取得.....4-99
4.4.3 DeviceInfoクラス 4-100
(1) メソッド一覧 4-100
(2) プロパティ一覧 4-100
(3) メソッド詳細 4-101
DeviceInfo コンストラクタ.....4-101
(4) プロパティ詳細 4-102
PortType 通信ポート種別の取得.....4-102
DeviceName デバイス名（プリンターモデル名）の取得.....4-102
BluetoothAddress Bluetoothアドレスの取得.....4-102
MacAddress MACアドレスの取得.....4-102
IpAddress IPアドレスの取得.....4-102
IsBonded ペアリング状態の取得.....4-103
DevicePath デバイスパスの取得.....4-103
4.4.4 PrinterIdentifierクラス 4-104
(1) プロパティ一覧 4-104
(2) プロパティ詳細 4-104
PrinterModelID 機種IDの取得.....4-104
TypeID タイプIDの取得.....4-104

RomVersionID ROMバージョンの取得.....4-105
MainFirmwareVersion ファームウェアバージョン(main)の取得.....4-105
Manufacturer メーカ名の取得.....4-105
ModelName モデル名の取得.....4-105
SerialNumber シリアル番号の取得.....4-105
BootFirmwareVersion ファームウェアバージョン(boot)の取得.....4-106
BootFirmwareChecksum ファームウェアチェックサム(boot)の取得.....4-106
MainFirmwareChecksum ファームウェアチェックサム(main)の取得.....4-106
FirmwareChecksum ファームウェアチェックサム(main + boot)の取得.....4-106
4.4.5 PrinterExceptionクラス.....4-107
(1) メソッド一覧.....4-107
(2) プロパティ一覧.....4-107
(3) 定数一覧.....4-107
① エラーコード.....4-107
(4) メソッド詳細.....4-109
PrinterException コンストラクタ.....4-109
(5) プロパティ詳細.....4-109
HResult エラーコードの取得.....4-109
Message エラーメッセージの取得.....4-109
4.4.6 SmartLabelManagerクラス.....4-110
(1) メソッド一覧.....4-110
(2) メソッド詳細.....4-111
SmartLabelManager コンストラクタ.....4-111
SelectSmartLabelFile ラベルファイルの指定.....4-111
ReplaceSmartLabelTextData ラベルのテキストデータの置換.....4-112
ReplaceSmartLabelImageData ラベルのイメージデータの置換.....4-113
ReplaceSmartLabelBarcodeData ラベルのバーコードデータの置換.....4-114

5章 ディスプレイのライブラリ機能 5-1

5.1 API リファレンス.....5-1
Method or Property メソッド名またはプロパティ名.....5-2
5.1.1 列挙型定数一覧.....5-3
① デバイスモデル（DeviceModel）.....5-3
② ポート種別（PortType）.....5-3
③ プリンター応答種別（PrinterResponseId）.....5-3
④ デバイスタイプ（DeviceType）.....5-4
⑤ メモリ領域（MemoryArea）.....5-4
⑥ 位置揃え（PrintAlignment）.....5-4
⑦ 強調文字（CharacterBold）.....5-4
⑧ アンダーライン（CharacterUnderline）.....5-4
⑨ 文字倍率（CharacterScale）.....5-5
⑩ 文字フォント（CharacterFont）.....5-5

⑪ 登録フォント（RegisteredFont）.....5-5
⑫ モジュールサイズ（ModuleSize）.....5-6
⑬ エラー訂正レベル（ErrorCorrection）.....5-6
⑭ QRデータモード（QrDataMode）.....5-6
⑮ QRクワイエットゾーン（QrQuietZone）.....5-7
⑯ マクロ登録処理（MacroRegistrationFunction）.....5-7
⑰ ディスプレイ応答種別（DisplayResponseId）.....5-7
⑱ 国際文字セット(InternationalCharacter).....5-8
⑲ コードページ(CodePage).....5-9
5.1.2 PrinterManagerクラス.....5-10
(1) メソッド一覧.....5-10
(2) プロパティ一覧.....5-11
(3) メソッド詳細.....5-12
PrinterManager コンストラクタ.....5-12
Connect デバイスとの通信開始.....5-12
Disconnect デバイスとの通信切断.....5-14
GetPrinterResponse プリンターからの各種応答取得.....5-14
StartDiscoveryDevice デバイスの探索開始.....5-16
CancelDiscoveryDevice
デバイスの探索中断.....5-17
GetFoundDevice 発見されたデバイス情報リストの取得.....5-17
Defragment メモリ領域の最適化.....5-17
InitializeMemoryArea メモリ領域の初期化.....5-17
ShowTemplate テンプレートの表示.....5-18
ShowSlide スライドの表示.....5-19
EnterStandbyMode スタンバイの表示.....5-20
ExecuteMacro マクロの実行.....5-20
TurnOnScreen 画面の表示オン・オフ.....5-21
SelectTemplate テンプレートの選択.....5-21
SetTemplateImageData イメージデータの設定.....5-22
SelectTemplateTextObject
text要素の選択.....5-23
SetTemplateTextAlignment
テキストデータの位置揃え.....5-23
SetTemplateTextLeftMargin
テキストデータの左マージンの設定.....5-24
SetTemplateTextLineSpacing
テキストデータの改行量の設定.....5-24
SetTemplateTextBold テキストデータの強調文字の設定.....5-25
SetTemplateTextUnderline
テキストデータのアンダーラインの設定.....5-25
SetTemplateTextSize テキストデータの文字サイズの設定.....5-26
SetTemplateTextFont テキストデータの文字フォントの設定.....5-26
SetTemplateTextRegisteredFont
テキストデータの登録フォントの設定.....5-27
SetTemplateTextRightSpacing
テキストデータの右スペース量の設定.....5-28

SetTemplateTextColor テキストデータの文字色の設定.....5-28
SetTemplateTextData テキストデータの入力.....5-29
SetTemplateBarcodeData
バーコードデータの入力.....5-29
SetTemplateQRCodeData
QRコードデータの入力.....5-30
RegisterTemplate テンプレートの登録.....5-32
UnregisterTemplate テンプレートの削除.....5-33
RegisterImageData イメージデータの登録.....5-34
UnregisterImageData イメージデータの削除.....5-35
RegisterSlideData スライドデータの登録.....5-35
UnregisterSlideData スライドデータの削除.....5-37
RegisterUserDefinedCharacter
外字の登録.....5-37
UnregisterUserDefinedCharacter
外字の削除.....5-38
RegisterOptionFont オプションフォントの登録.....5-39
UnregisterOptionFont オプションフォントの削除.....5-40
ControlMacroRegistration
マクロ登録の開始・終了.....5-40
GetDisplayResponse ディスプレイからの各種応答取得.....5-42
SendBinary バイナリデータの送信.....5-43
SendDataFile 指定ファイルの送信.....5-44
(4) プロパティ詳細.....5-45
SendTimeout 送信タイムアウト時間の設定/取得.....5-45
ReceiveTimeout 受信タイムアウト時間の設定/取得.....5-45
InternationalCharacter
国際文字セットの設定/取得.....5-46
CodePage コードページの設定/取得.....5-46
DeviceModel デバイスモデルの取得.....5-47
PortType 接続中ポート種別の取得.....5-48
IsConnect プリンターとの通信状態の確認.....5-48
SocketKeepingTime ソケット維持時間の設定/取得.....5-48
Version SDKバージョンの取得.....5-49
5.1.3 DeviceInfoクラス.....5-50
(1) メソッド一覧.....5-50
(2) プロパティ一覧.....5-50
(3) メソッド詳細.....5-51
DeviceInfo コンストラクタ.....5-51
(4) プロパティ詳細.....5-52
PortType 通信ポート種別の取得.....5-52
DeviceName デバイス名（プリンターモデル名）の取得.....5-52
BluetoothAddress Bluetoothアドレスの取得.....5-52
MacAddress MACアドレスの取得.....5-52
IpAddress IPアドレスの取得.....5-52
IsBonded ペアリング状態の取得.....5-53
DevicePath デバイスパスの取得.....5-53

5.1.4 PrinterIdentifierクラス 5-54
(1) プロパティ一覧 5-54
(2) プロパティ詳細 5-54
PrinterModelID 機種IDの取得..... 5-54
TypeID タイプIDの取得..... 5-54
RomVersionID ROMバージョンの取得..... 5-55
MainFirmwareVersion
ファームウェアバージョン(main)の取得..... 5-55
Manufacturer メーカ名の取得..... 5-55
ModelName モデル名の取得..... 5-55
SerialNumber シリアル番号の取得..... 5-55
BootFirmwareVersion ファームウェアバージョン(boot)の取得..... 5-56
BootFirmwareChecksum ファームウェアチェックサム(boot)の取得..... 5-56
MainFirmwareChecksum ファームウェアチェックサム(main)の取得..... 5-56
FirmwareChecksum ファームウェアチェックサム(main + boot)の取得...
..... 5-56
5.1.5 PrinterExceptionクラス 5-57
(1) メソッド一覧 5-57
(2) プロパティ一覧 5-57
(3) 定数一覧 5-57
① エラーコード 5-57
(4) メソッド詳細 5-59
PrinterException コンストラクタ..... 5-59
(5) プロパティ詳細 5-59
HResult エラーコードの取得..... 5-59
Message エラーメッセージの取得..... 5-59

6章 サンプルプログラム 6-1

6.1 画面構成 6-1
6.2 注意事項 6-2

付録A 文字セット A-1

A.1 コードページ表（文字コードテーブル） A-1
A.2 国際文字セット A-11

付録B バーコードサイズ一覧

B.1 バーコードサイズ一覧（SLP720RT、SLP721RT、RP-F10、RP-G10、MP-B30、
MP-B21L） B-1
B.1.1 PrintBarcode, PrintPageModeBarcode B-1
B.1.2 PrintPDF417, PrintPageModePDF417 ..... B-6
B.1.3 PrintQRCode, PrintPageModeQRCode B-7
B.1.4 PrintDataMatrix, PrintPageModeDataMatrix B-8

B.1.5 PrintMaxicode, PrintPageModeMaxicode B-10
B.1.6 PrintGS1DataBarStacked, PrintPageModeGS1DataBarStacked B-11
B.1.7 PrintGS1DataBarStackedOmnidirectional,
PrintPageModeGS1DataBarStackedOmnidirectional B-12
B.1.8 PrintGS1DataBarExpandedStacked,
PrintPageModeGS1DataBarExpandedStacked B-13
B.2 バーコードサイズ一覧（MP-B30L、MP-B21L） B-14
B.2.1 PrintAztecCode, PrintPageModeAztecCode B-14
B.3 バーコードサイズ一覧（RP-E10、RP-D10、MP-B20、MP-A40） B-16
B.3.1 PrintBarcode, PrintPageModeBarcode B-16
B.3.2 PrintPDF417, PrintPageModePDF417 ..... B-21
B.3.3 PrintQRCode, PrintPageModeQRCode B-22
B.3.4 PrintDataMatrix, PrintPageModeDataMatrix B-23
B.3.5 PrintMaxicode, PrintPageModeMaxicode B-25
B.3.6 PrintGS1DataBarStacked, PrintPageModeGS1DataBarStacked B-26
B.3.7 PrintGS1DataBarStackedOmnidirectional,
PrintPageModeGS1DataBarStackedOmnidirectional B-27
B.3.8 PrintGS1DataBarExpandedStacked,
PrintPageModeGS1DataBarExpandedStacked B-28

付録 C オープンソースソフトウェアライセンス

C.1 MIT License C-1
C.2 Apache License 2.0 ..... C-2

1章　製品概要

本章では、SDKの製品概要について説明します。

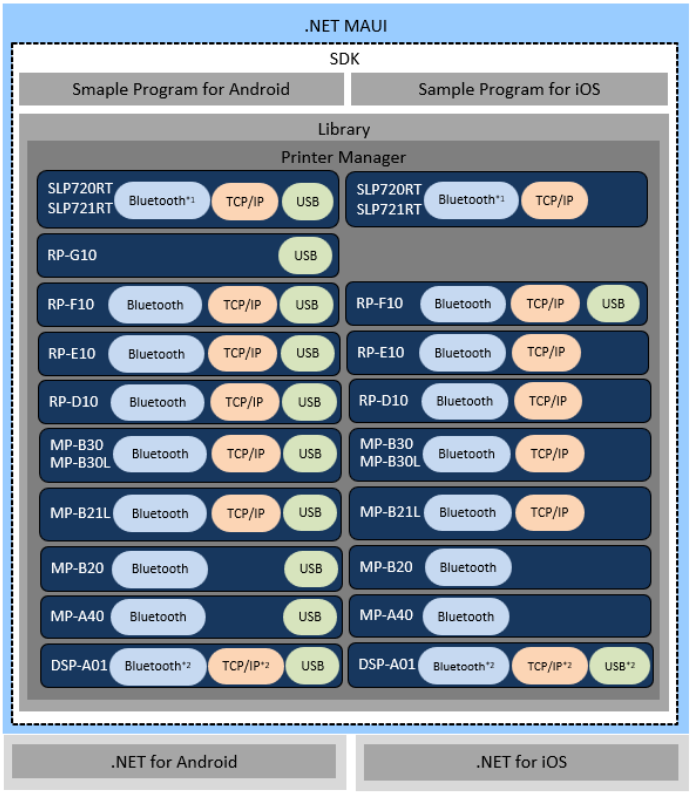
1.1　SDKの提供機能

ライブラリとサンプルプログラムを含むSDKは、「はじめに」の「対象デバイス」に記載のSII製プリンター（以降、プリンター）を利用するための機能を提供します。
またSDKは、ライブラリのサンプルプログラムをVisual Studioにおける.NET MAUIアプリのソリューションで提供します。

1.2　SDKの概要

1.2.1　SDKの構成

本SDKはSII Print Class Library for Android、SII Print Class Library for iOSを組み込んでいます。
SDKに含まれるライブラリとサンプルプログラムの範囲を、下図に破線で示します。



*1：SLP721RTのみサポート
*2：プリンター経由で使用する場合にサポート

1.2.2 ライブラリの提供機能

.NET MAUIにより開発するiOS/Androidアプリケーションは iOS/Androidデバイスの通信ポート（Bluetooth、USB、またはTCP/IP）を通じて、プリンターに印字データやプリンターコマンドを容易に送信できます。
また、プリンターステータスを取得できます。

ライブラリでは、下記の機能を提供します。
提供する機能は、デバイスにより異なります。提供する機能については、「4章 プリンターのライブラリ機能」または「5章 ディスプレイのライブラリ機能」を参照してください。

- デバイスとの接続及び切断
- デバイスへのデータ送信（印字データやプリンターコマンド*1）
- バーコードの印字、2次元コードの印字
- デバイスへのデータファイル送信（印字データやプリンターコマンド*1）
- 用紙のカット
- プリンターステータスの取得
- プリンターのデータ待ち状態解除
- デバイスからの各種応答取得
- 印字命令の一括登録
- プリンターステータスのコールバック関数登録
- Bluetooth、またはTCP/IPによるプリンターの探索
- プリンターのハードウェアリセット
- ドロワの動作制御
- ブザーの鳴動制御
- ディスプレイの表示制御
- バーコードスキャナーのコールバック関数登録
- ラベルファイルの印字
- ラベルファイルのオブジェクトデータの置き換え
- ログファイルの出力

*1: プリンターからの応答を読み取るコマンドには対応しておりません。
プリンターから応答を読み取るには、GetStatusまたはGetPrinterResponseを使用してください。

（注意） 推奨バーコードスキャナー、及びバーコードスキャナーの設定については、「RP-F10シリーズ サーマルプリンタ 取扱説明書」を参照してください。

1.2.3 SII製プリンターとBluetooth通信・USB通信を行うiOSアプリケーションの開発

プリンターとBluetooth通信、またはUSB通信を行うiOSアプリケーションをApp Storeに登録する場合、SIIからAppleに事前申請が必要になります。詳細は弊社までお問合せください。

1.2.4 ディスプレイの工場出荷時の登録データ

ディスプレイに工場出荷時に登録されるテンプレートなどのデータは、品質向上のため予告無く追加・変更される場合があります。
また、指定する言語設定、及び文字コードによっては適切なエンコードが指定されているテンプレートを利用する必要があります。工場出荷時に登録されるデータの詳細については、弊社Webサイトを参照してください。
https://www.sii-ps.com/dspa01/

2章　製品仕様

本章では、ライブラリの製品仕様について説明します。

2.1　動作環境

2.1.1　iOS

(1)　対応 iOS デバイス

ライブラリが動作するiOSデバイスを下記に示します。

①　SLP721RT

iPhoneモデル名

- iPhone 15 Pro Max
- iPhone 15 Pro
- iPhone 15 Plus
- iPhone 15
- iPhone 14 Pro Max
- iPhone 14 Pro
- iPhone 14 Plus
- iPhone 14
- iPhone SE (3rd generation)
- iPhone 13 Pro Max
- iPhone 13 Pro
- iPhone 13
- iPhone 13 mini
- iPhone 12 Pro Max
- iPhone 12 Pro
- iPhone 12
- iPhone 12 mini
- iPhone SE (2nd generation)
- iPhone 11 Pro Max
- iPhone 11 Pro
- iPhone 11
- iPhone XS Max
- iPhone XS

- iPhone XR

iPadモデル名

- iPad Pro 12.9-inch (6th generation)
- iPad Pro 11inch (4th generation)
- iPad (10th generation)
- iPad Air (5th generation)
- iPad mini (6th generation)
- iPad (9th generation)
- iPad Pro 12.9inch (5th generation)
- iPad Pro 11inch (3rd generation)
- iPad Air (4th generation)
- iPad (8th generation)
- iPad Pro 12.9inch (4th generation)
- iPad Pro 11inch (2nd generation)
- iPad (7th generation)
- iPad Air (3rd generation)
- iPad mini (5th generation)
- iPad Pro 12.9inch (3rd generation)
- iPad (6th generation)
- iPad Pro 12.9inch (2nd generation)

② RP-F10-x27J1-5（USB Type-C＋Bluetooth＋USB ホストモデル）、MP-B30-B02JK2-xx、MP-B30L-B46JK2-xx、MP-B20-B02Jx1-74、MP-A40-B06JK3x

iPhoneモデル名

- iPhone 16 Pro Max
- iPhone 16 Pro
- iPhone 16 Plus
- iPhone 16
- iPhone 15 Pro Max
- iPhone 15 Pro
- iPhone 15 Plus
- iPhone 15
- iPhone 14 Pro Max
- iPhone 14 Pro
- iPhone 14 Plus
- iPhone 14
- iPhone SE (3rd generation)
- iPhone SE (2nd generation)

iPadモデル名

- iPad mini (A17 Pro)
- iPad Pro 13-inch (M4)
- iPad Pro 11-inch (M4)
- iPad Air 13-inch (M2)
- iPad Air 11-inch (M2)
- iPad Pro 12.9-inch (6th generation)
- iPad Pro 11-inch (4th generation)
- iPad (10th generation)
- iPad Air (5th generation)
- iPad mini (6th generation)

③ RP-F10-x27J1-4(Bluetooth＋USB ホストモデル)、MP-B30-B02JK1-xx

iPhoneモデル名

- iPhone XR
- iPhone XS
- iPhone XS Max
- iPhone X
- iPhone 8
- iPhone 8 Plus
- iPhone 7
- iPhone 7 Plus
- iPhone SE
- iPhone 6s
- iPhone 6s Plus

iPadモデル名

- iPad Pro 11-inch
- iPad Pro 12.9-inch (3rd generation)
- iPad (6th generation)
- iPad Pro 12.9-inch (2nd generation)
- iPad Pro 10.5-inch
- iPad (5th generation)
- iPad Pro 9.7-inch
- iPad Pro 12.9-inch (1st generation)
- iPad mini 4

iPodモデル名

- iPod touch (6th generation)

④ MP-B30L-B46JK1-xx

iPhoneモデル名

- iPhone 12
- iPhone 12 Pro
- iPhone 12 Pro Max
- iPhone 12 mini
- iPhone SE (2nd generation)
- iPhone 11
- iPhone 11 Pro
- iPhone 11 Pro Max
- iPhone XR
- iPhone XS
- iPhone XS Max
- iPhone X
- iPhone 8
- iPhone 8 Plus
- iPhone 7
- iPhone 7 Plus
- iPhone SE
- iPhone 6s
- iPhone 6s Plus

iPadモデル名

- iPad (8th generation)
- iPad Pro 11-inch (2nd generation)
- iPad Pro 12.9-inch (4th generation)
- iPad (7th generation)
- iPad Pro 11-inch
- iPad Pro 12.9-inch (3rd generation)
- iPad (6th generation)
- iPad Pro 12.9-inch (2nd generation)
- iPad Pro 10.5-inch
- iPad (5th generation)
- iPad Pro 9.7-inch
- iPad Pro 12.9-inch (1st generation)
- iPad mini 4

iPodモデル名

- iPod touch (7th generation)
- iPod touch (6th generation)

⑤ MP-B21L

iPhoneモデル名

- iPhone 14 Pro Max
- iPhone 14 Pro
- iPhone 14 Plus
- iPhone 14
- iPhone SE (3rd generation)
- iPhone 13 Pro Max
- iPhone 13 Pro
- iPhone 13
- iPhone 13 mini
- iPhone 12 Pro Max
- iPhone 12 Pro
- iPhone 12
- iPhone 12 mini
- iPhone SE (2nd generation)
- iPhone 11 Pro Max
- iPhone 11 Pro
- iPhone 11
- iPhone XS Max
- iPhone XS
- iPhone XR

iPadモデル名

- iPad Pro 12.9-inch (6th generation)
- iPad Pro 11inch (4th generation)
- iPad (10th generation)
- iPad Air (5th generation)
- iPad mini (6th generation)
- iPad (9th generation)
- iPad Pro 12.9inch (5th generation)
- iPad Pro 11inch (3rd generation)
- iPad Air (4th generation)
- iPad (8th generation)
- iPad Pro 12.9inch (4th generation)
- iPad Pro 11inch (2nd generation)
- iPad (7th generation)
- iPad Air (3rd generation)
- iPad mini (5th generation)
- iPad Pro 12.9inch (3rd generation)
- iPad (6th generation)
- iPad Pro 12.9inch (2nd generation)

⑥ RP-E10、RP-D10

iPhoneモデル名

- iPhone X
- iPhone 8
- iPhone 8 Plus
- iPhone 7
- iPhone 7 Plus
- iPhone SE
- iPhone 6s
- iPhone 6s Plus
- iPhone 6
- iPhone 6 Plus

iPadモデル名

- iPad Pro 12.9-inch (2nd generation)
- iPad Pro 10.5-inch
- iPad (5th generation)
- iPad Pro 9.7-inch
- iPad Pro 12.9-inch (1st generation)
- iPad mini 4
- iPad Air 2
- iPad mini 3

iPodモデル名

- iPod touch (6th generation)

(2) 対応 iOS バージョン

ライブラリが動作するiOSバージョンを下記に示します。

- iOS 17-17.6.1
- iPadOS 17-17.6.1
- iOS 18-18.6
- iPadOS 18-18.6
- iOS 26-26.4
- iPadOS 26-26.4

2.1.2 Android

(1) 対応 OS

ライブラリが動作するOSを下記に示します。

Android 7.0 (API 24)-Android 16.0 (API 36)

2.1.3 .NET MAUI

(1) 動作環境

ライブラリが動作する.NET MAUI環境を以下に示します。

- .NET 10.0 (net10.0-android, net10.0-ios)
- .NET 9.0 (net9.0-android, net9.0-ios)

2.2 プリンターの設定

ライブラリの利用時には、プリンターのメモリスイッチを下記の[値]に設定してください。
メモリスイッチの詳細、及び工場出荷時の設定については、「取扱説明書」を参照してください。

2.2.1 iOS

(1) SLP720RT、SLP721RT

MS	機能	値
1-2	テイクンモード選択 (Taken Mode)	0 : 有効*1 1 : 無効*2
1-3	マークモード選択 (Mark Mode)	0 : 有効*3 1 : 無効*4
4-6	用紙自動検出選択 (Paper Auto Detection)	0 : 有効*3 1 : 無効*3*4
5-1	自動ステータス応答機能選択 (Auto Status Back)	0 : 有効
5-2	イニシャライズ完了応答選択 (Init. Response)	0 : 有効
5-3	エラー時データ破棄選択 (Error Through)	0 : 有効
5-4	出力バッファフル時データ破棄選択 (Response Data Discarding)	1 : 無効
7	感熱紙選択 (Thermal Paper)	00B : レシート 01B : ライナレスラベル紙 10B : SLPラベル紙
13-3	リアルタイムコマンド選択 (Realtime Command)	1 : 有効
17-3	用紙カット後の逆方向紙送り設定 (Backfeed After Cut)	0 : 有効 1 : 無効*5

*1: ライナレスラベル紙で連続印字する場合は、本設定を「有効」にしてください。
テイクンセンサのステータス応答は、本設定が「有効」の場合に応答します。
*2: レシート、SLPラベル紙で連続印刷する場合は、本設定を「無効」にしてください。
*3: FeedPositionを使用する場合は、下記のどちらかの設定が必要です。
・用紙を自動検出する場合は、プリンターのメモリスイッチMS4-6(用紙自動検出選択)を「有効」にしてください。
・用紙を指定する場合は、MS4-6(用紙自動検出選択)を「無効」、MS1-3(マークモード選択)を「有効」にしてください。
また、利用する用紙を下記の通りに選択してください。
・マーク付きライナレスラベル紙を使用する場合:
MS7(感熱紙選択)を「ライナレスラベル紙」にしてください。
・SLPラベル紙を使用する場合:
MS7(感熱紙選択)を「SLPラベル紙」にしてください。
*4: マークなし用紙またはマークなしライナレスラベル紙を利用する場合は、本設定を「無効」にし、プリンターのメモリスイッチMS7(感熱紙選択)で使用する用紙を選択してください。
*5: 下記の条件でCutPaperを実行し、直後にPrintSmartLabelImageDataを使用する場合は、本設定を「無効」にしてください。
・MS1-3(マークモード選択)が「無効」
・CutPaperのCuttingMethodに**CUT_FULL**を指定、またはMS1-2(テイクンモード選択)が「有効」かつCuttingMethodに**CUT_PARTIAL**を指定

(2) RP-F10

MS	機能	値
5-1	自動ステータス応答機能選択 (Auto Status Back)	0 : 有効
5-2	イニシャライズ完了応答選択 (Init. Response)	0 : 有効
5-3	エラー時データ破棄選択 (Error Through)	0 : 有効
5-4	出力バッファフル時データ破棄選択 (Response Data Discarding)	1 : 無効
13-3	リアルタイムコマンド選択 (Realtime Command)	1 : 有効
38-1	スキャナー自動ステータス応答機能選択*1 (Scanner Auto Status Back)	0 : 有効*2
39-1*3	iOS自動接続選択 (Auto Connection)	1 : 無効 0 : 有効*4

*1: バーコードスキャナーをサポートするプリンターのファームウェアバージョンは、Ver.1.10以降です。
*2: バーコードスキャナーを使用する場合は、「有効」を選択してください。
バーコードスキャナーを含む周辺機器の組み合わせについては、「取扱説明書」を参照してください。
*3: Bluetooth接続の場合のみ。
*4: ResetPrinterを使用する場合は、「有効」を選択してください。

(3) RP-E10、RP-D10

MS	機能	値
5-2	イニシャライズ完了応答選択 (Init. Response)	0 : 有効
5-3	エラー時データ破棄選択 (Error Through)	0 : 有効
39-1*1	iOS自動接続選択 (Auto Connection)	1 : 無効 0 : 有効*2

*1: Bluetooth接続の場合のみ。
*2: ResetPrinterを使用する場合は、「有効」を選択してください。

(4) MP-B30

- メモリスイッチ

MS	機能	値
1-1	通信選択 (Interface)	1 : USB/Wireless
1-2	マークモード選択 (Mark Mode)	1 : 無効*1 0 : 有効
1-6	エラー時データ破棄選択 (Error Through)	0 : 有効
1-8	出力バッファフル時データ破棄選択 (Response Data Discarding)	1 : 無効
2-2	リアルタイムコマンド選択 (Realtime Command)	0 : 有効
9-1	自動ステータス応答機能選択 (Auto Status Back)	0 : 有効
9-2	イニシャライズ完了応答選択 (Init. Response)	0 : 有効

*1: CutPaperを使用する場合は、「無効」を選択してください。

- Bluetooth通信設定

Bluetooth 通信設定については、「MP-B30 シリーズ サーマルプリンタ 技術説明書」または「MP-B30-x02JK2x サーマルプリンタ 技術説明書」に記載されている、プリンターコマンド「Bluetooth 通信の設定」を参照してください。

機能	値
iOS自動接続選択 (Auto Connection)	1 : 有効*1 0 : 無効

*1: ResetPrinterを使用する場合は、「有効」を選択してください。

(5) MP-B30L

- メモリスイッチ

MS	機能	値
1-1	通信選択 (Interface)	1 : USB/Wireless
1-2	マーク/ギャップモード選択 (Mark/Gap Mode)	1 : 無効*1 0 : 有効*2
1-3 ～ 1-5	コマンド体系選択 (Command System)	000B : ESC/POS
1-6	エラー時データ破棄選択 (Error Through)	0 : 有効
1-8	出力バッファフル時データ破棄選択 (Response Data Discarding)	1 : 無効
2-1	紙セット後の初期化動作選択 (Paper Set Handle)	0 : 頭出し
2-2	リアルタイムコマンド選択 (Realtime Command)	0 : 有効
2-7	フィードスイッチによる頭出し動作選択 (SW Paper Form Feed)	1 : カット位置頭出し
9-1	自動ステータス応答機能選択 (Auto Status Back)	0 : 有効
9-2	イニシャライズ完了応答選択 (Init. Response)	0 : 有効

*1: CutPaperを使用する場合は、「無効」を選択してください。
*2: FeedPositionを使用する場合は、「有効」を選択してください。

- Bluetooth通信設定

Bluetooth 通信設定については、「MP-B30L シリーズ サーマルプリンタ 技術説明書」または「MP-B30L-x46JK2x サーマルプリンタ 技術説明書」に記載されている、プリンターコマンド「Bluetooth 通信の設定」を参照してください。

機能	値
iOS自動接続選択 (Auto Connection)	1 : 有効*1 0 : 無効

*1: ResetPrinterを使用する場合は、「有効」を選択してください。

(6) MP-B21L

- メモリスイッチ

MS	機能	値
1-1	通信選択 (Interface)	1 : USB/Wireless
1-2	マーク・ギャップモード選択 (Mark/Gap Mode)	1 : 無効*1 0 : 有効*2
1-3 ～ 1-5	コマンド体系選択 (Command System)	000B : ESC/POS
1-6	エラー時データ破棄選択 (Error Through)	0 : 有効
1-8	出力バッファフル時データ破棄選択 (Response Data Discarding)	1 : 無効
2-2	リアルタイムコマンド選択 (Realtime Command)	0 : 有効
9-1	自動ステータス応答機能選択 (Auto Status Back)	0 : 有効
9-2	イニシャライズ完了応答選択 (Init. Response)	0 : 有効

*1: CutPaperを使用する場合は「無効」を選択してください。
*2: FeedPositionを使用する場合は「有効」を選択してください。

- Bluetooth通信設定

Bluetooth 通信設定については、「MP-B21 シリーズ サーマルプリンタ 技術説明書」に記載されている、プリンターコマンド「Bluetooth 通信の設定」を参照してください。

機能	値
iOS自動接続選択 (iOS Auto Connection)	1 : 有効*1 0 : 無効

*1: ResetPrinterを使用する場合は、「有効」を選択してください。

(7) MP-B20

- メモリスイッチ

MS	機能	値
1-1	通信選択 (Interface)	1 : Wireless
3-1	自動ステータス応答機能選択 (Auto Status Back)	0 : 有効
3-2	イニシャライズ完了応答選択 (Init. Response)	0 : 有効
3-3	リアルタイムコマンド選択 (Realtime Command)	0 : 有効
3-4	エラー時データ破棄選択 (Error Through)	0 : 有効

- Bluetooth通信設定

Bluetooth 通信設定については、「MP-B20-B02JK1 サーマルプリンタ 技術説明書」に記載されている、プリンターコマンド「Bluetooth 通信の設定」を参照してください。

機能	値
iOS自動接続選択 (iOS Auto Connection)	1 : 有効*1 0 : 無効

*1: ResetPrinterを使用する場合は、「有効」を選択してください。

(8) MP-A40

- メモリスイッチ

MS	機能	値
1-1	通信選択 (Interface)	1 : USB/Wireless
1-2	マークモード選択 (Mark Mode)	1 : 無効*1 0 : 有効*2
1-3 ～ 1-5	コマンド体系選択 (Command System)	000B : ESC/POS
1-6	エラー時データ破棄選択 (Error Through)	0 : 有効
2-2	リアルタイムコマンド選択 (Realtime Command)	0 : 有効
9-1	自動ステータス応答機能選択<ESC/POS> (Auto Status Back<ESC/POS>)	0 : 有効
9-2	イニシャライズ完了応答選択<ESC/POS> (Init. Response<ESC/POS>)	0 : 有効

*1: CutPaperを使用する場合は「無効」を選択してください。
*2: FeedPositionを使用する場合は「有効」を選択してください。

- Bluetooth通信設定

Bluetooth 通信設定については、「MP-A40シリーズ サーマルプリンタ 技術説明書」に記載されている、プリンターコマンド「Bluetooth 通信の設定」を参照してください。

機能	値
iOS自動接続選択 (iOS Auto Connection)	1 : 有効*1 0 : 無効

*1: ResetPrinterを使用する場合は、「有効」を選択してください。

2.2.2 Android

(1) SLP720RT、SLP721RT

MS	機能	値
1-2	テイクンモード選択 (Taken Mode)	0 : 有効[*1] 1 : 無効[*2]
1-3	マークモード選択 (Mark Mode)	0 : 有効[*3] 1 : 無効[*4]
4-6	用紙自動検出選択 (Paper Auto Detection)	0 : 有効[*3] 1 : 無効[*3*4]
5-1	自動ステータス応答機能選択 (Auto Status Back)	0 : 有効
5-2	イニシャライズ完了応答選択 (Init. Response)	0 : 有効
5-3	エラー時データ破棄選択 (Error Through)	0 : 有効
5-4	出力バッファフル時データ破棄選択 (Response Data Discarding)	1 : 無効
7	感熱紙選択 (Thermal Paper)	00B : レシート 01B : ライナレスラベル紙 10B : SLPラベル紙
13-3	リアルタイムコマンド選択 (Realtime Command)	1 : 有効
17-3	用紙カット後の逆方向紙送り設定 (Backfeed After Cut)	0 : 有効 1 : 無効[*5]

*1: ライナレスラベル紙で連続印字する場合は、本設定を「有効」にしてください。
テイクンセンサのステータス応答は、本設定が「有効」の場合に応答します。

*2: レシート、SLPラベル紙で連続印字する場合は、本設定を「無効」にしてください。

*3: FeedPositionを使用する場合は、下記のどちらかの設定が必要です。
・用紙を自動検出する場合は、プリンターのメモリスイッチMS4-6(用紙自動検出選択)を「有効」にしてください。
・用紙を指定する場合は、MS4-6(用紙自動検出選択)を「無効」、MS1-3(マークモード選択)を「有効」にしてください。
また、利用する用紙を下記の通りに選択してください。
・マーク付きライナレスラベル紙を使用する場合:
MS7(感熱紙選択)を「ライナレスラベル紙」にしてください。
・SLPラベル紙を使用する場合:
MS7(感熱紙選択)を「SLPラベル紙」にしてください。

*4: マークなし用紙またはマークなしライナレスラベル紙を利用する場合は、本設定を「無効」にし、プリンターのメモリスイッチMS7(感熱紙選択)で使用する用紙を選択してください。

*5: 下記の条件でCutPaperを実行し、直後にPrintSmartLabelImageDataを使用する場合は、本設定を「無効」にしてください。
・MS1-3(マークモード選択)が「無効」
・CutPaperのCuttingMethodに**CUT_FULL**を指定、またはMS1-2(テイクンモード選択)が「有効」かつCuttingMethodに**CUT_PARTIAL**を指定

(2) RP-F10、RP-G10

MS	機能	値
5-1	自動ステータス応答機能選択 (Auto Status Back)	0 : 有効
5-2	イニシャライズ完了応答選択 (Init. Response)	0 : 有効
5-3	エラー時データ破棄選択 (Error Through)	0 : 有効
5-4	出力バッファフル時データ破棄選択 (Response Data Discarding)	1 : 無効
13-3	リアルタイムコマンド選択 (Realtime Command)	1 : 有効
38-1	スキャナー自動ステータス応答機能選択*1 (Scanner Auto Status Back)	0 : 有効*2

*1: バーコードスキャナーをサポートするプリンターのファームウェアバージョンは、Ver.1.10以降です。
*2: バーコードスキャナーを使用する場合は、「有効」を選択してください。
バーコードスキャナーを含む周辺機器の組み合わせについては、「取扱説明書」を参照してください。

(3) RP-E10、RP-D10

MS	機能	値
5-2	イニシャライズ完了応答選択 (Init. Response)	0 : 有効

(4) MP-B30

MS	機能	値
1-1	通信選択 (Interface)	1 : USB/Wireless*1 0 : USB*2
1-2	マークモード選択 (Mark Mode)	1 : 無効*3 0 : 有効
1-6	エラー時データ破棄選択 (Error Through)	0 : 有効
1-8	出力バッファフル時データ破棄選択 (Response Data Discarding)	1 : 無効
2-2	リアルタイムコマンド選択 (Realtime Command)	0 : 有効
9-1	自動ステータス応答機能選択 (Auto Status Back)	0 : 有効
9-2	イニシャライズ完了応答選択 (Init. Response)	0 : 有効

*1: Bluetooth接続、及びTCP/IP接続の場合は、「USB/Wireless」を選択してください。
*2: USB接続の場合は、「USB」を選択してください。
*3: CutPaperを使用する場合は、「無効」を選択してください。

(5) MP-B30L

MS	機能	値
1-1	通信選択 (Interface)	1 : USB/Wireless*1 0 : USB*2
1-2	マーク/ギャップモード選択 (Mark/Gap Mode)	1 : 無効*3 0 : 有効*4
1-3 ～ 1-5	コマンド体系選択 (Command System)	000B : ESC/POS
1-6	エラー時データ破棄選択 (Error Through)	0 : 有効
1-8	出力バッファフル時データ破棄選択 (Response Data Discarding)	1 : 無効
2-1	紙セット後の初期化動作選択 (Paper Set Handle)	0 : 頭出し
2-2	リアルタイムコマンド選択 (Realtime Command)	0 : 有効
2-7	フィードスイッチによる頭出し動作選択 (SW Paper Form Feed)	1 : カット位置頭出し
9-1	自動ステータス応答機能選択 (Auto Status Back)	0 : 有効
9-2	イニシャライズ完了応答選択 (Init. Response)	0 : 有効

*1: Bluetooth接続、及びTCP/IP接続の場合は、「USB/Wireless」を選択してください。
*2: USB接続の場合は、「USB」を選択してください。
*3: CutPaperを使用する場合は、「無効」を選択してください。
*4: FeedPositionを使用する場合は、「有効」を選択してください。

(6) MP-B21L

MS	機能	値
1-1	通信選択 (Interface)	1 : USB/Wireless*1 0 : USB*2
1-2	マーク・ギャップモード選択 (Mark/Gap Mode)	1 : 無効*3 0 : 有効*4
1-3 ～ 1-5	コマンド体系選択 (Command System)	000B : ESC/POS
1-6	エラー時データ破棄選択 (Error Through)	0 : 有効
1-8	出力バッファフル時データ破棄選択 (Response Data Discarding)	1 : 無効
2-2	リアルタイムコマンド選択 (Realtime Command)	0 : 有効
9-1	自動ステータス応答機能選択 (Auto Status Back)	0 : 有効
9-2	イニシャライズ完了応答選択 (Init. Response)	0 : 有効

*1: Bluetooth及びTCP/IP接続の場合は、「USB/Wireless」を選択してください。
*2: USB接続の場合は、「USB」を選択してください。
*3: CutPaperを使用する場合は、「無効」を選択してください。
*4: FeedPositionを使用する場合は、「有効」を選択してください。

(7) MP-B20

MS	機能	値
1-1	通信選択 (Interface)	1 : Wireless*1 0 : USB*2
3-1	自動ステータス応答機能選択 (Auto Status Back)	0 : 有効
3-2	イニシャライズ完了応答選択 (Init. Response)	0 : 有効
3-3	リアルタイムコマンド選択 (Realtime Command)	0 : 有効
3-4	エラー時データ破棄選択 (Error Through)	0 : 有効
3-5	出力バッファフル時データ破棄選択 (Response Data Discarding)	1 : 無効

*1: Bluetooth接続の場合は、「Wireless」を選択してください。
*2: USB接続の場合は、「USB」を選択してください。

(8) MP-A40

MS	機能	値
1-1	通信選択 (Interface)	1 : USB/Wireless*1 0 : USB
1-2	マークモード選択 (Mark Mode)	1 : 無効*2 0 : 有効*3
1-3 ～ 1-5	コマンド体系選択 (Command System)	000B : ESC/POS
1-6	エラー時データ破棄選択 (Error Through)	0 : 有効
2-2	リアルタイムコマンド選択 (Realtime Command)	0 : 有効
9-1	自動ステータス応答機能選択<ESC/POS> (Auto Status Back<ESC/POS>)	0 : 有効
9-2	イニシャライズ完了応答選択<ESC/POS> (Init. Response<ESC/POS>)	0 : 有効

*1: Bluetooth及びTCP/IP接続の場合は、「USB/Wireless」を選択してください。
*2: CutPaperを使用する場合は、「無効」を選択してください。
*3: FeedPositionを使用する場合は、「有効」を選択してください。

2.3 注意事項

2.3.1 iOS

ライブラリはスレッドセーフではありません。マルチスレッドにおいてライブラリを使用すると、異常終了する可能性があります。

TCP/IP利用時はライブラリでプリンタードライバーや他のライブラリと通信ポートの共有はできません。
TCP/IP利用時はiOSデバイスが接続している無線LANアクセスポイントとプリンターが同じネットワークに接続されている必要があります。

iPadOSを搭載したiPadのマルチタスク機能を使った複数App同時起動において、複数のAppから1台のプリンターへの同時接続をサポートしていません。

2.3.2 Android

Bluetooth利用時はSPP（Serial Port Profile）で接続が確立されている必要があります。

USB利用時は、Androidデバイスが USBホスト機能をサポートしている必要があります。

TCP/IP利用時はライブラリでプリンタードライバーや他のライブラリと通信ポートの共有はできません。
TCP/IP利用時はAndroidデバイスが接続している無線LANアクセスポイントとプリンターが同じネットワークに接続されている必要があります。

3章　ライブラリの利用方法

本章では、iOSまたはAndroidアプリケーションの開発とライブラリの利用方法について説明します。

3.1　iOS/Androidアプリケーションの開発環境

iOS/Androidアプリケーションを開発するためには、下記のツールが必要です。

- iOS/Android

  Windows：　Visual Studio 2022 または Visual Studio 2026

  Mac：　Visual Studio Code

- iOS

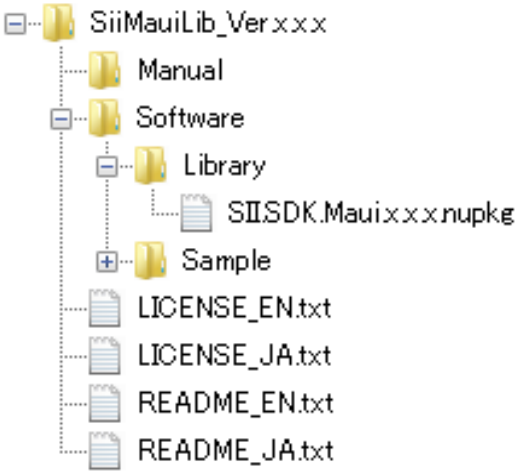
  Mac：　Xcode16.1以降

本書では、本章以降、各ツールを利用できる環境が整っていることを前提に説明します。

（注意）　iOSアプリをビルドするにはMacにXcodeがインストールされている必要があります。

3.2 提供ファイル

SDKのファイル構成は、下記の通りです。



ライブラリは、NuGetパッケージ形式です。ライブラリのファイル名は、SII.SDK.Maui.x.x.x.nupkgです。

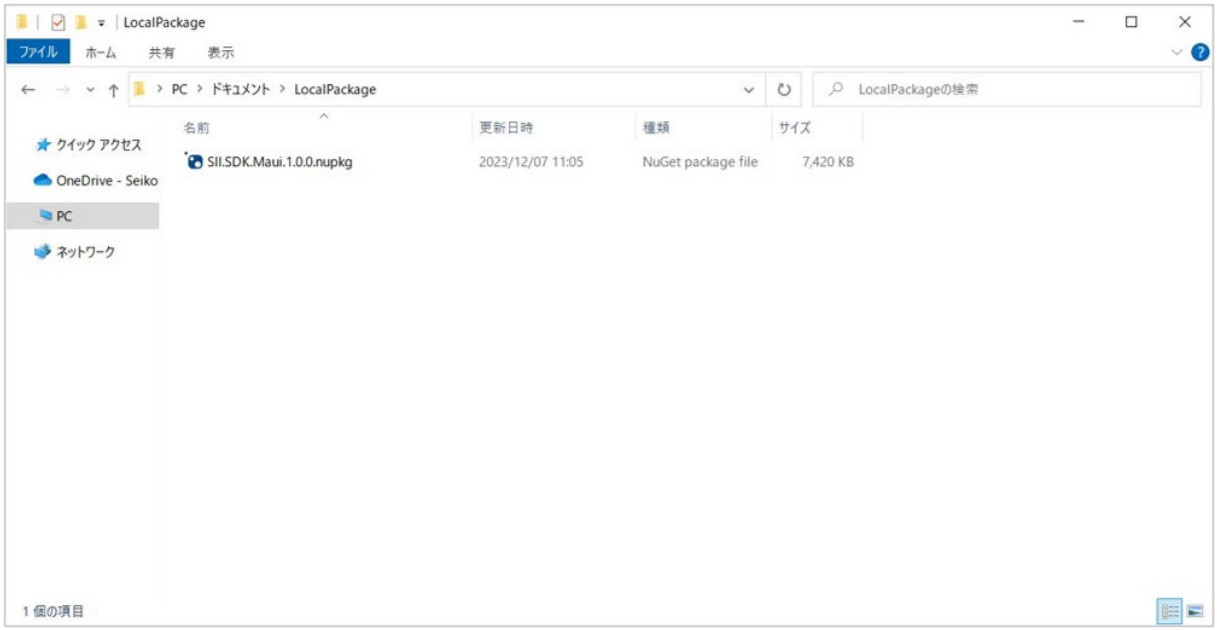
3.3 Visual Studioへの組み込み

SDKに含まれるサンプルプログラム（Sample）のプロジェクトを例にして、ライブラリをプロジェクトに組み込む手順を説明します。
SDKに含まれるサンプルプログラムについては、「6章 サンプルプログラム」を参照してください。

3.3.1 Windows

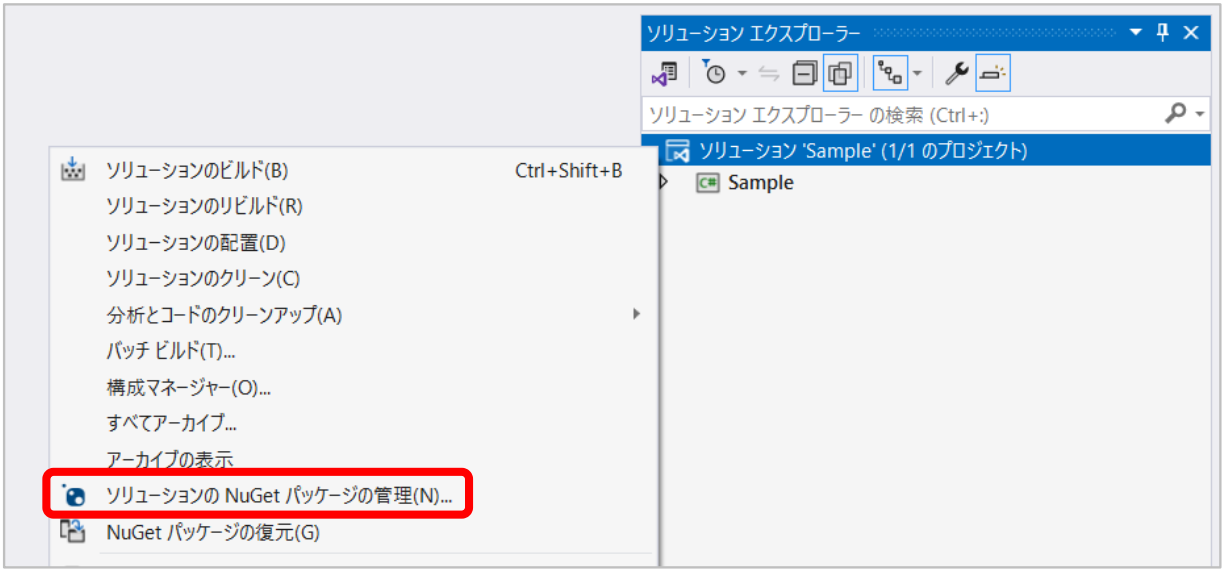
(1) iOS/Android共通

① 任意のフォルダにライブラリを配置してください。
例として「LocalPackage」フォルダを作成し、SII.SDK.Maui.x.x.x.nupkgを配置します。

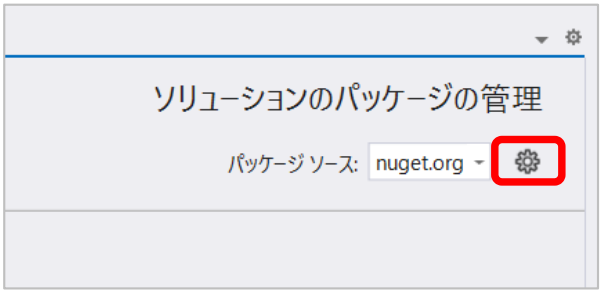


② Visual Studioを開いてください。

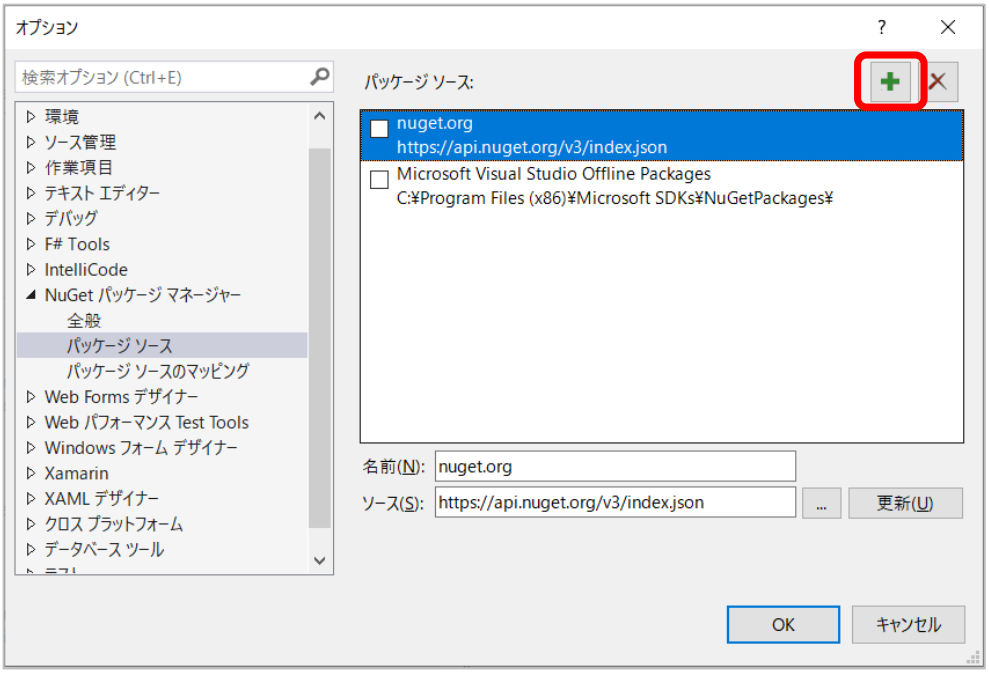
③ ソリューションエクスプローラーの[ソリューション'Sample']の右クリックメニューから[ソリューションのNuGetパッケージの管理(N)...]を選択してください。



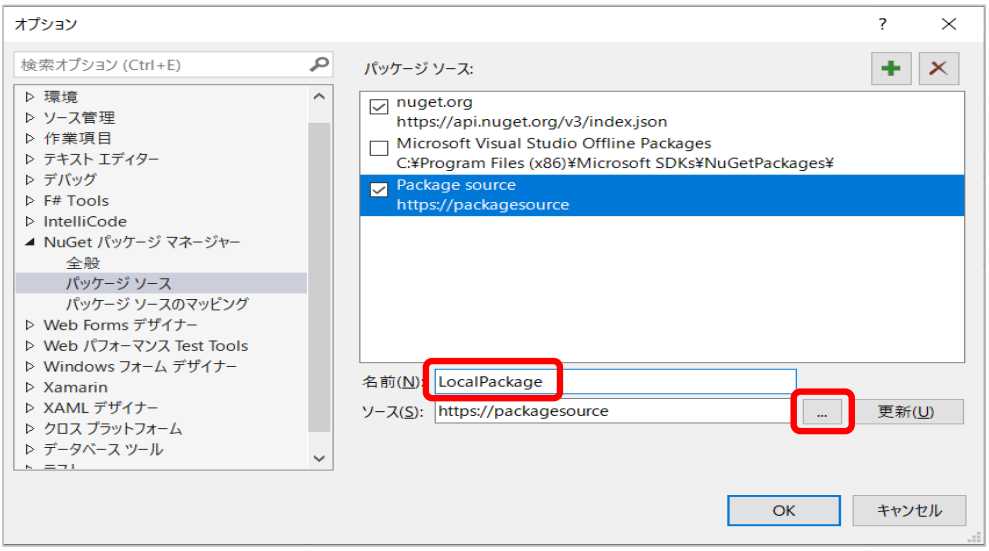
④　「ソリューションのパッケージの管理」から歯車アイコンをクリックしてください。



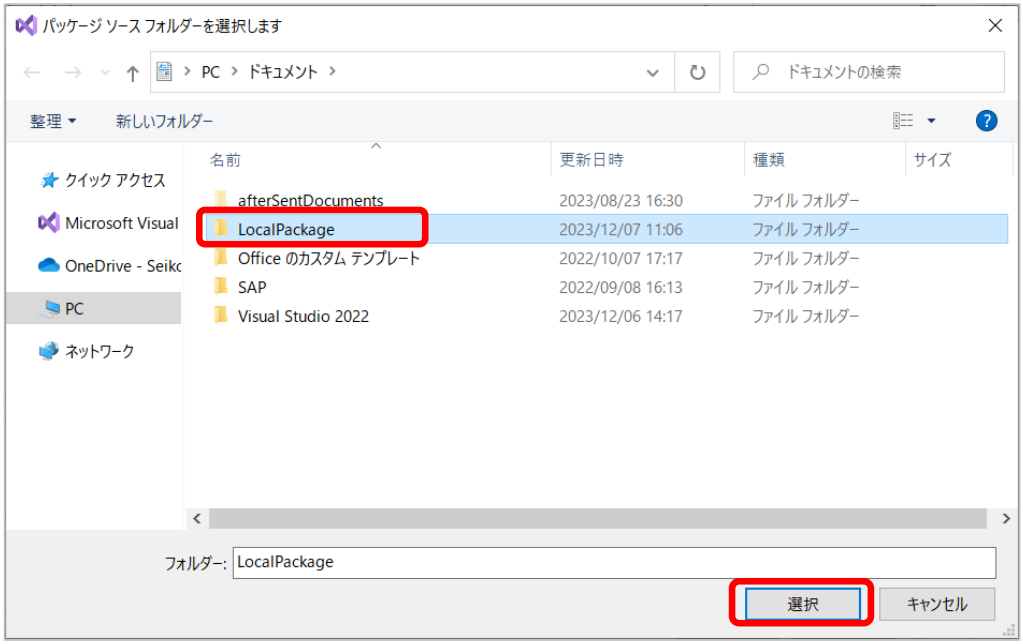
⑤　「オプション」ダイアログの[+]ボタンをクリックしてください。



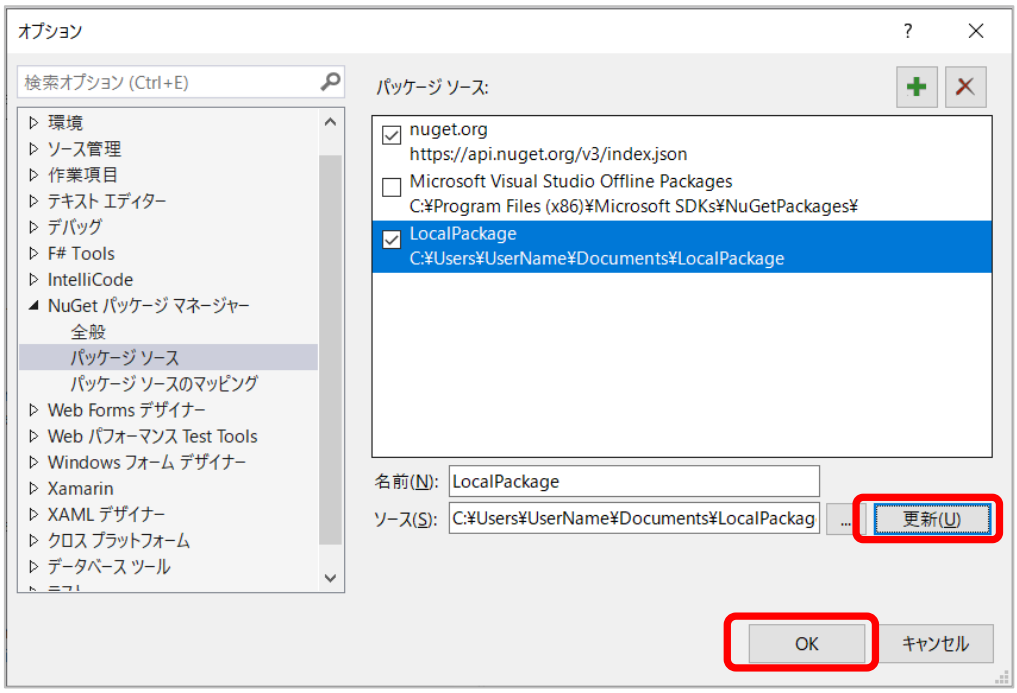
⑥　「名前(N)」をLocalPackageに変更し[...]ボタンをクリックしてください。



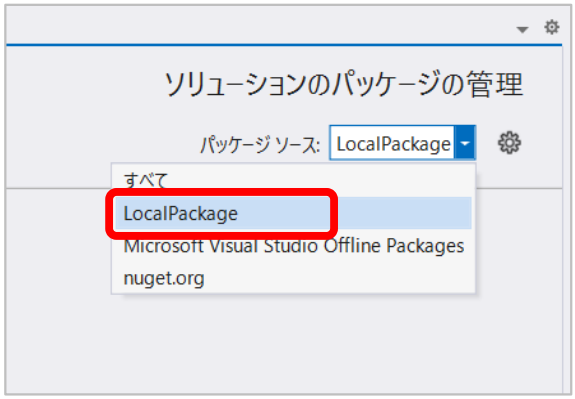
⑦ 「パッケージソースフォルダーを選択します」ダイアログで、手順①で作成した「LocalPackage」フォルダを選択し、[選択]ボタンをクリックしてください。



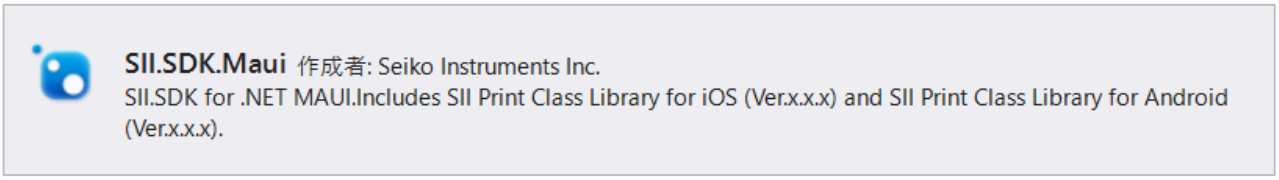
⑧ 「オプション」ダイアログに戻るので、[更新(U)]ボタンをクリックしてから[OK]ボタンをクリックしてください。



⑨ 「ソリューションのパッケージの管理」-「パッケージソース」のプルダウンメニューから[LocalPackage]を選択してください。



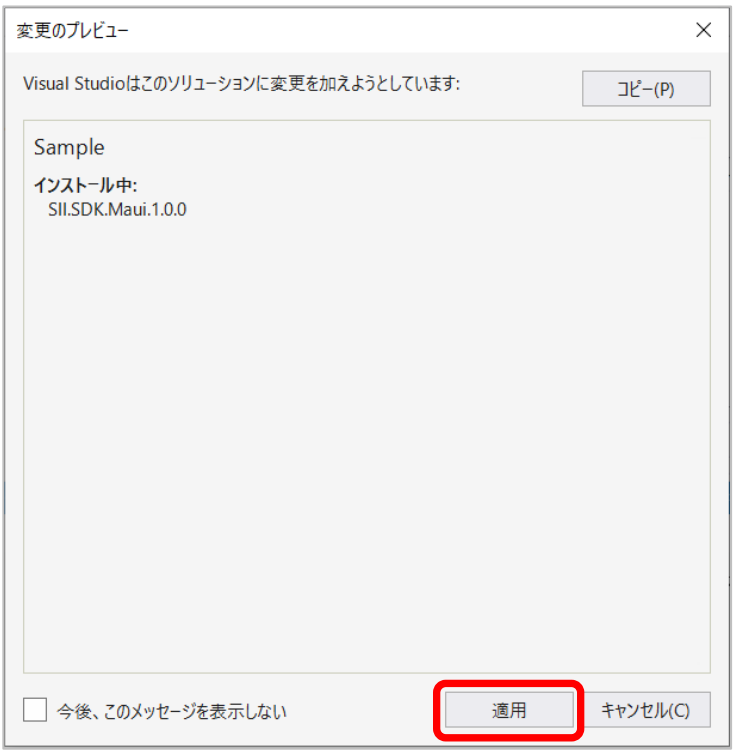
⑩ [SII.SDK.Maui]を選択してください。



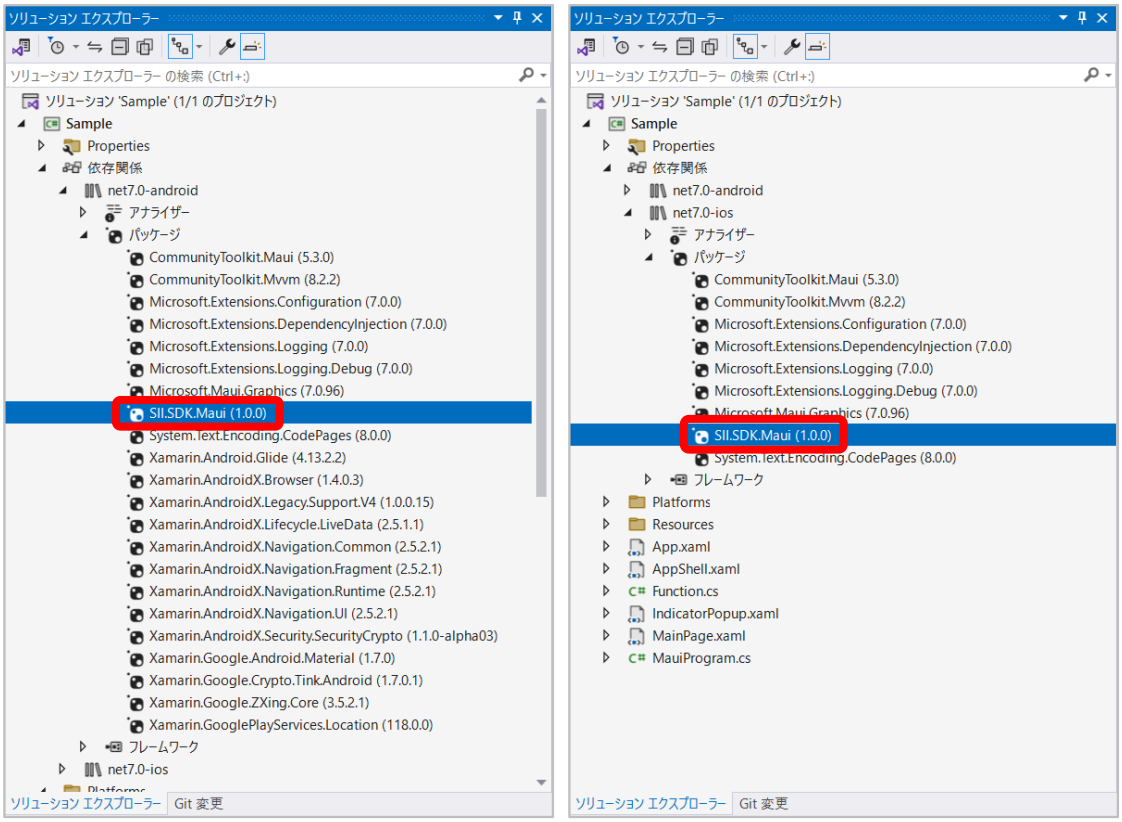
⑪ [SII.SDK.Maui]のプロパティパネルで追加したいプロジェクトのチェックボックスをオンにしてから、[インストール]ボタンをクリックしてください。



⑫　「変更のプレビュー」ダイアログが表示されたら、[適用]ボタンをクリックしてください。



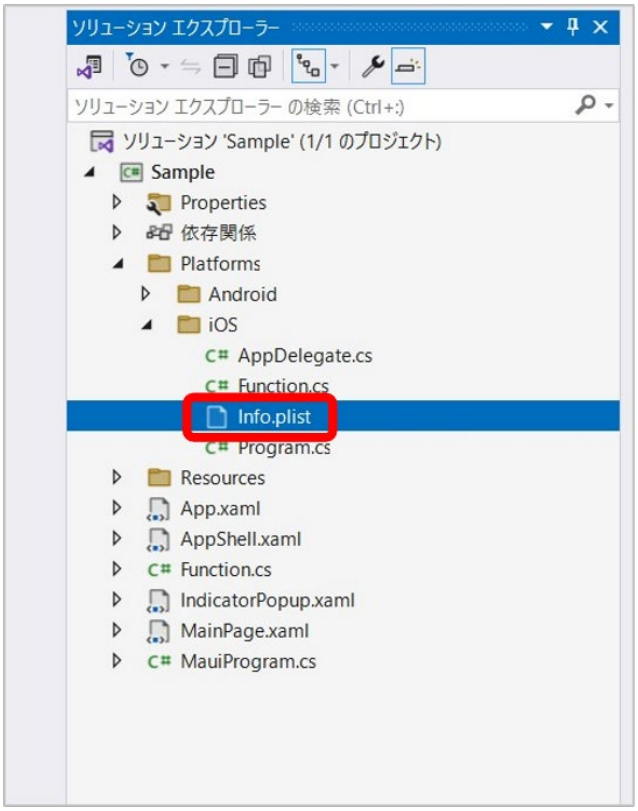
⑬　プロジェクトがインストールされたことを確認してください。



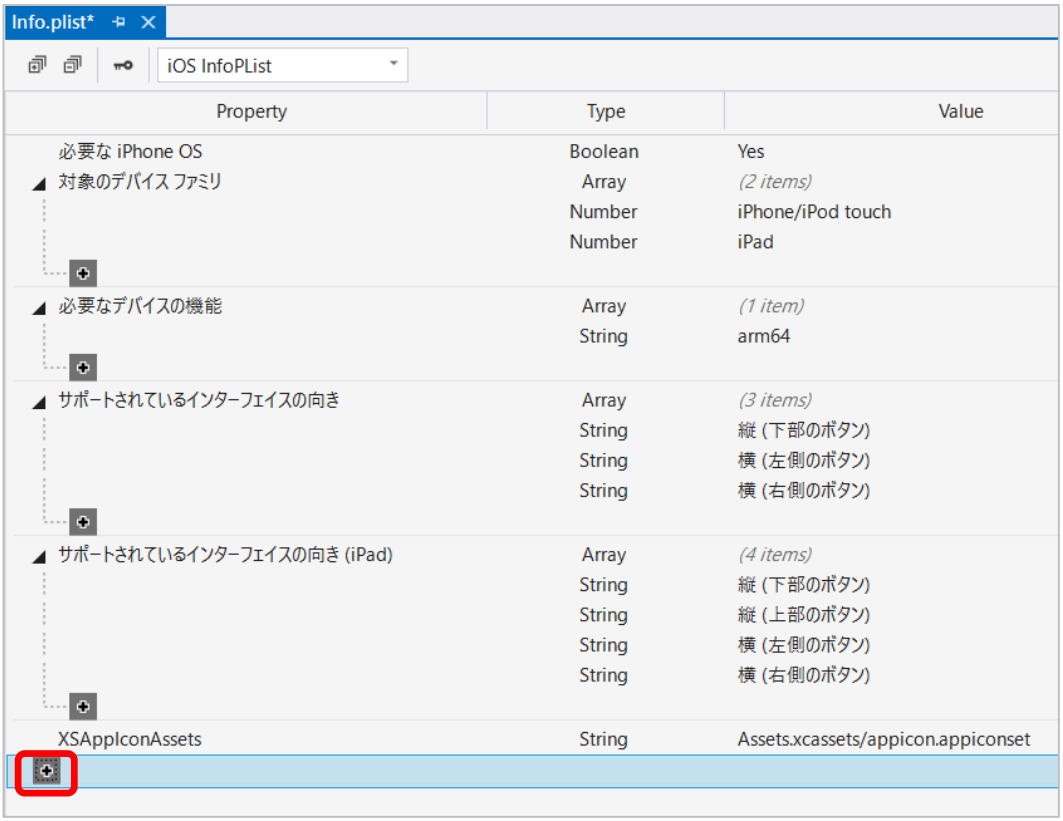
以降は、プラットフォームにより、手順が異なります。iOSの場合は、「3.3.1(2) iOS」を参照してください。Androidの場合は、「3.3.1(3) Android」を参照してください。

(2) iOS

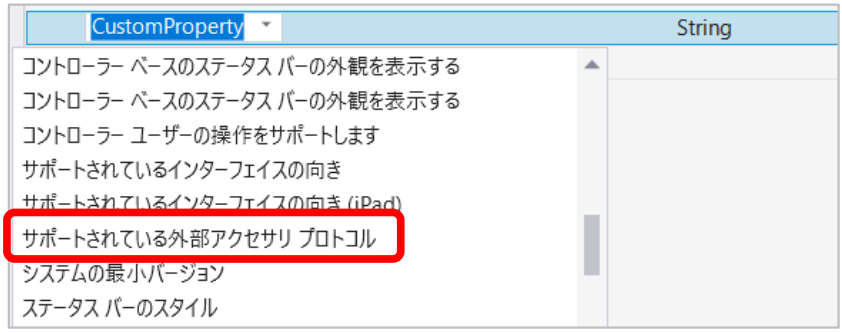
⑭ ソリューションエクスプローラーの[Sample]-[Info.plist]の右クリックメニューから[デザイナーの表示]を選択してください。



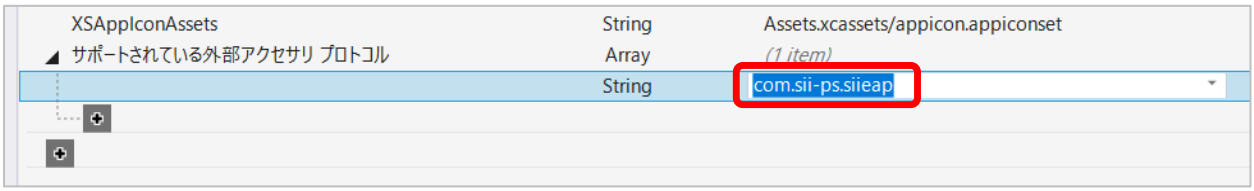
⑮ [Info.plist]の[+]を選択し、リストを展開してください。



⑯　リストから[サポートされている外部アクセサリプロトコル]を選択してください。



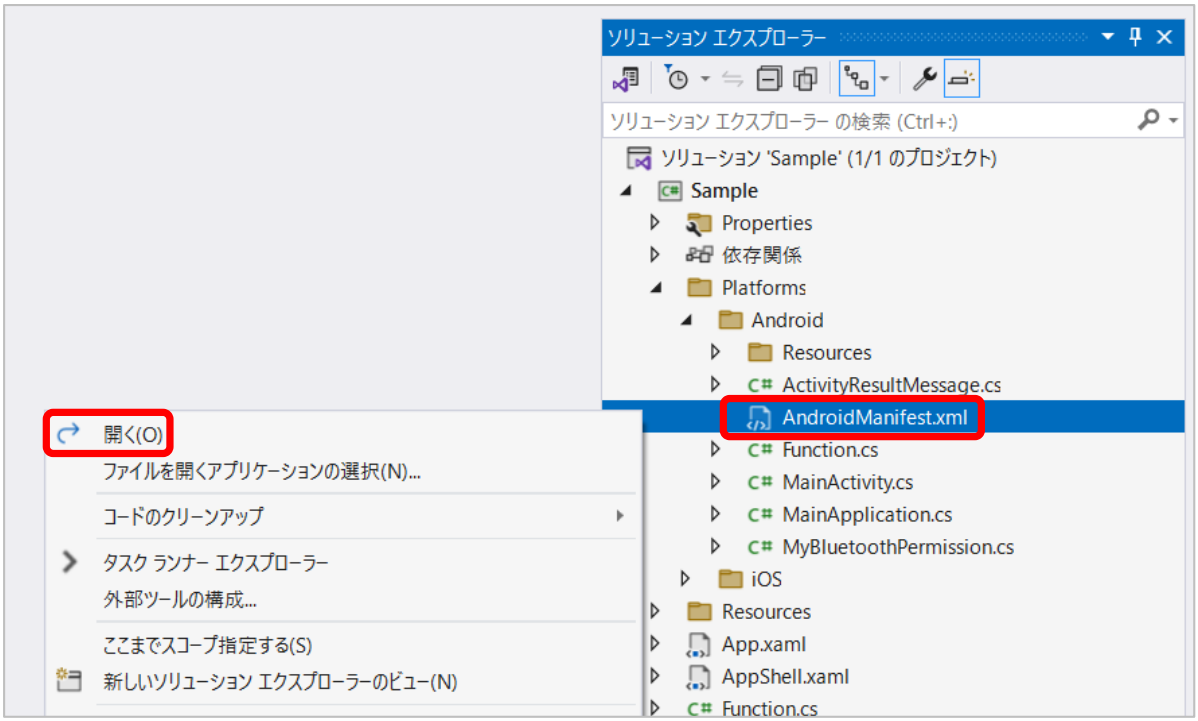
⑰　展開した[サポートされている外部アクセサリプロトコル]にValueとしてcom.sii-ps.siieapと入力してください。



以上で、ライブラリの機能が利用可能になります。

(3) Android

⑭ ソリューションエクスプローラーの[Sample]-[AndroidManifest.xml]の右クリックメニューから[開く(O)]を選択してください。



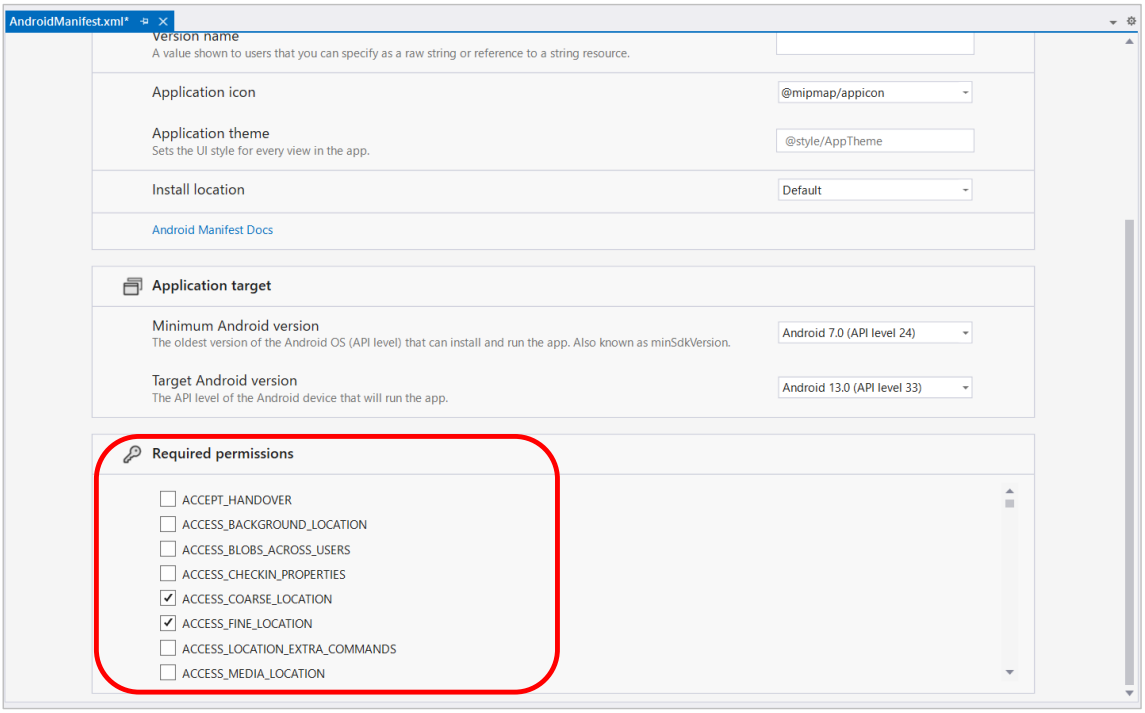
⑮ 「Required permissions」で下記のチェックボックスをオンにしてください。

[Bluetoothを利用する場合]

- ACCESS_FINE_LOCATION
- ACCESS_COARSE_LOCATION
- BLUETOOTH
- BLUETOOTH_ADMIN
- BLUETOOTH_CONNECT
- BLUETOOTH_SCAN

[TCP/IPを利用する場合]

- ACCESS_WIFI_STATE
- CHANGE_WIFI_STATE
- INTERNET



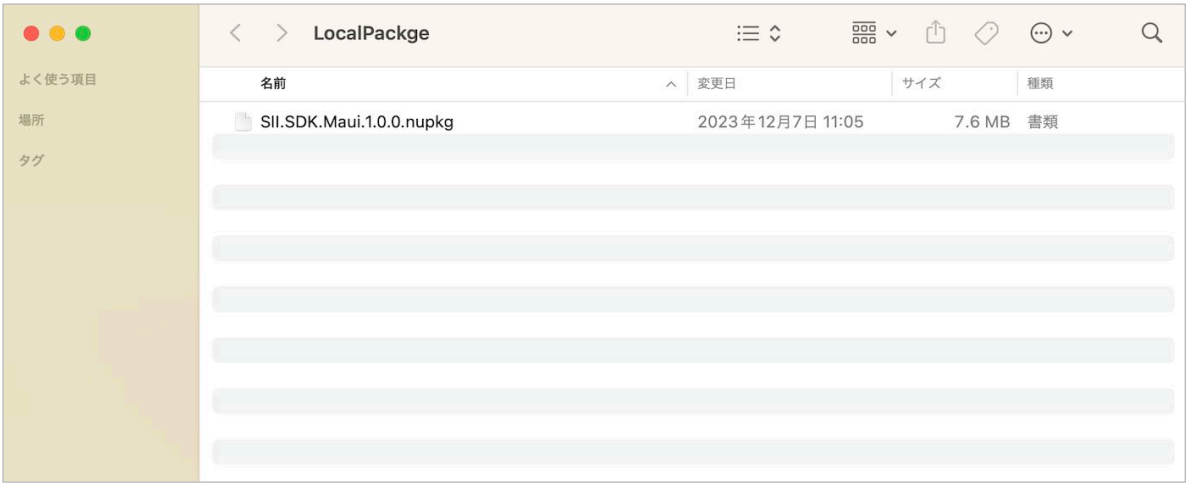
⑯ SmartLabelManagerクラスを利用する場合はSampleの「NuGetパッケージの管理...」から「Xamarin.Google.ZXing.Core」パッケージを追加してください。

以上で、ライブラリの機能が利用可能になります。

3.3.2 Mac

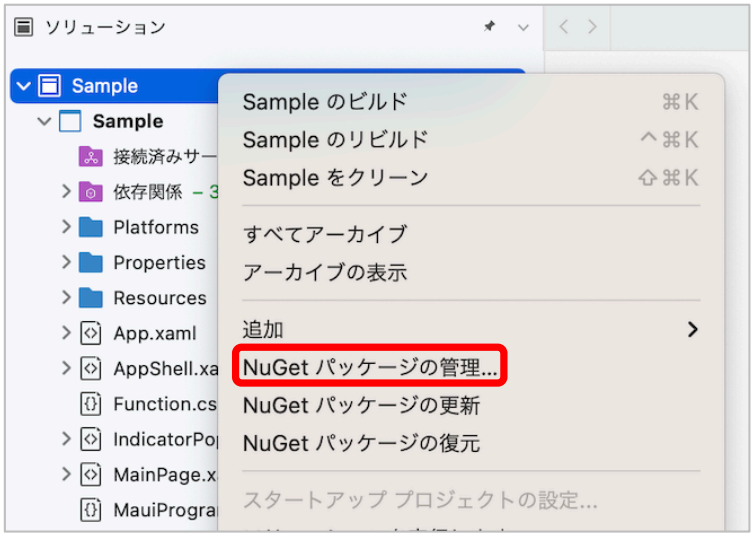
(1) iOS/Android共通

① 任意のフォルダにライブラリを配置してください。
例として「LocalPackage」フォルダを作成し、SII.SDK.Maui.x.x.x.nupkgを配置します。



② Visual Studioを開いてください。

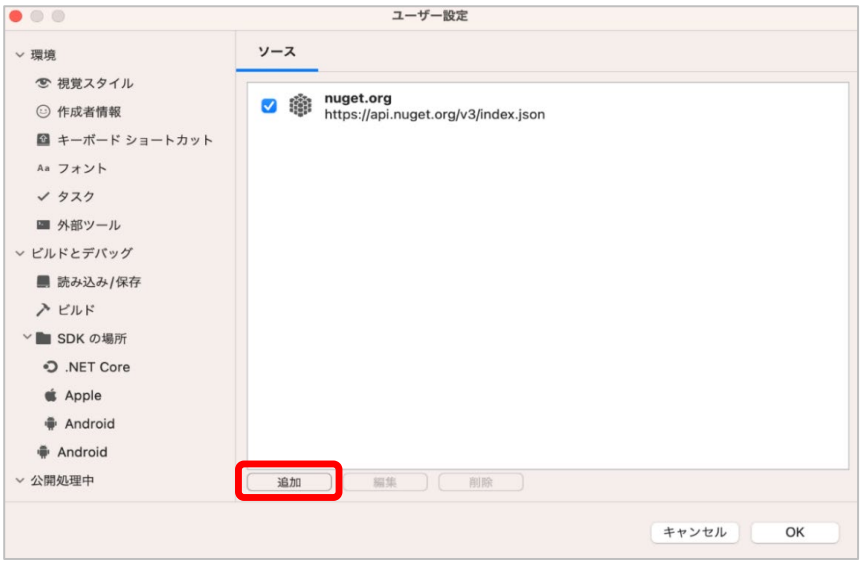
③ ソリューションエクスプローラーの「Sample」の右クリックメニューから[NuGetパッケージの管理...]を選択してください。



④ 「パッケージソース」のドロップダウンメニューから[ソースの構成...]を選択してください。

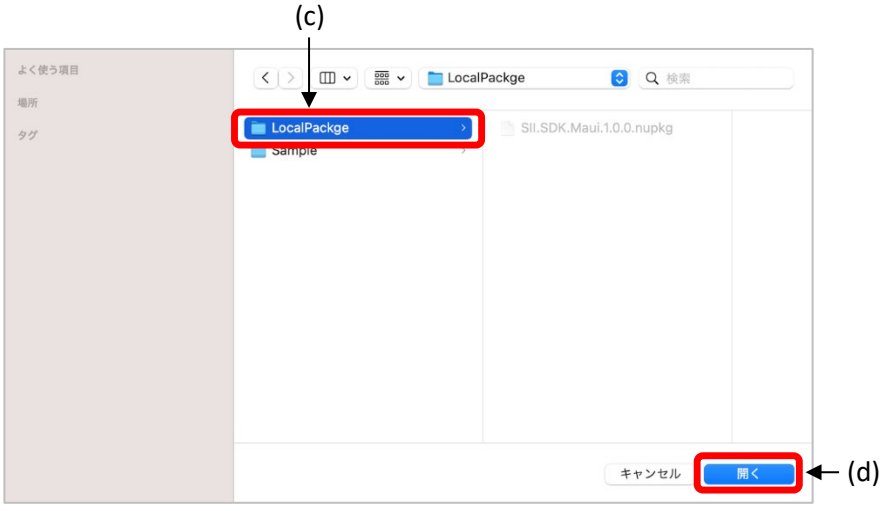


⑤ 「ユーザー設定」ダイアログで[追加]ボタンをクリックしてください。

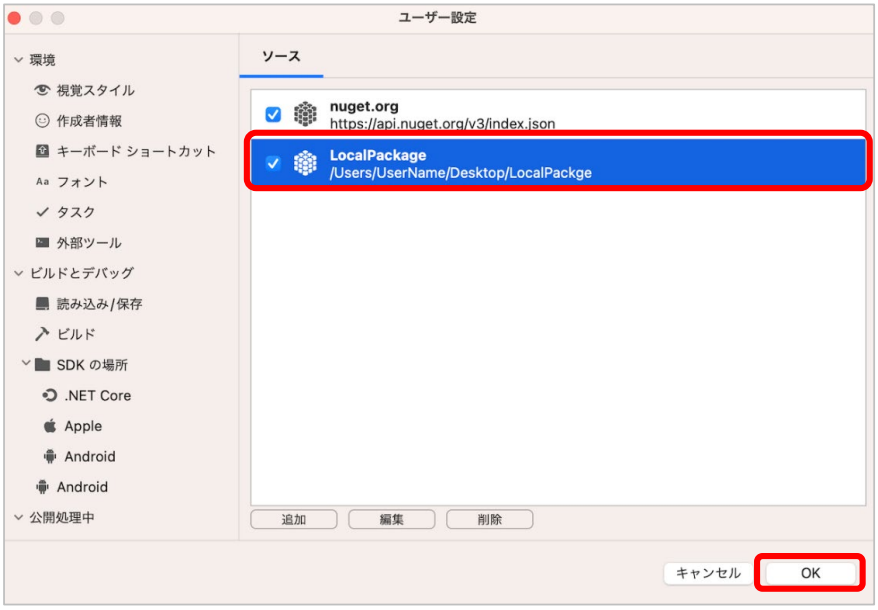


⑥ 「パッケージソースを追加する」ダイアログで「名前」にLocalPackageと入力し(a)、[...]ボタンをクリックしてください(b)。
「パッケージソースフォルダの選択」ダイアログで、手順①で作成した「LocalPackage」フォルダを選択し(c)、[開く]ボタンをクリックしてください(d)。
「パッケージソースを追加する」ダイアログに戻るので、[変換元の追加]ボタンをクリックしてください(e)。

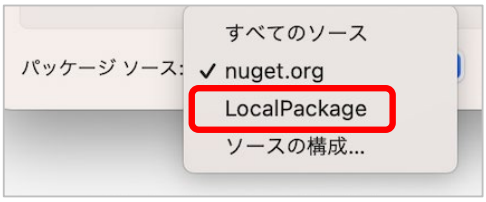




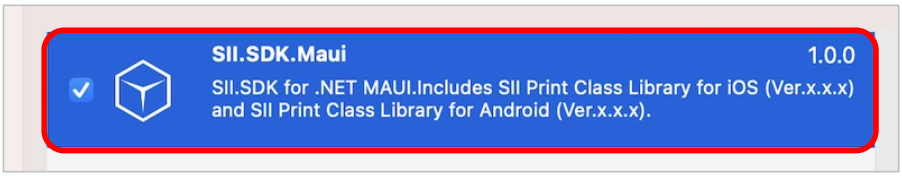
⑦ 「ユーザー設定」ダイアログで「LocalPackage」が追加されていることを確認し、[OK]ボタンをクリックしてください。



⑧ 「パッケージソース」のドロップダウンメニューから[LocalPackage]を選択してください。



⑨ [SII.SDK.Maui]を選択してください。



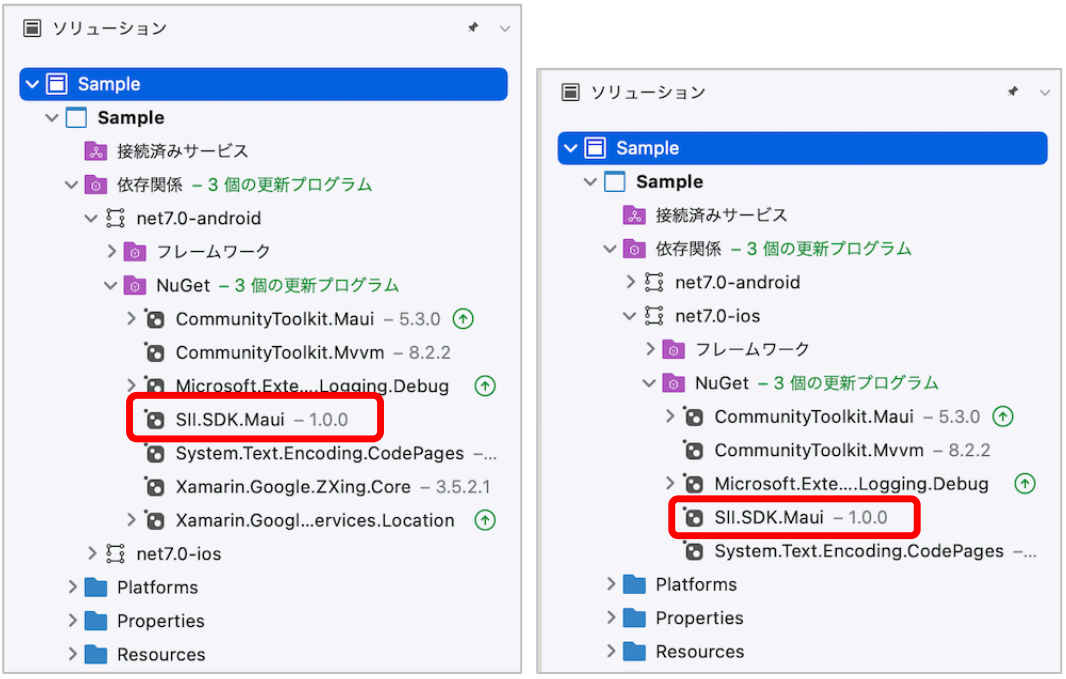
⑩ [パッケージを追加]ボタンをクリックしてください。



⑪ 「プロジェクトの選択」ダイアログで追加したいプロジェクトのチェックボックスをオンにしてから、[OK]ボタンをクリックしてください。



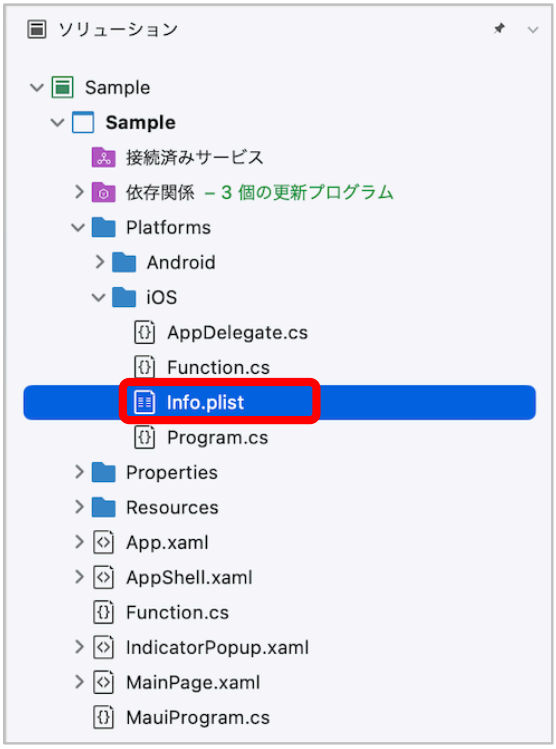
⑫ プロジェクトがインストールされたことを確認してください。



以降は、プラットフォームにより、手順が異なります。iOSの場合は、「3.3.2(2) iOSを参照してください。Androidの場合は、「3.3.2(3) Android」を参照してください。

(2) iOS

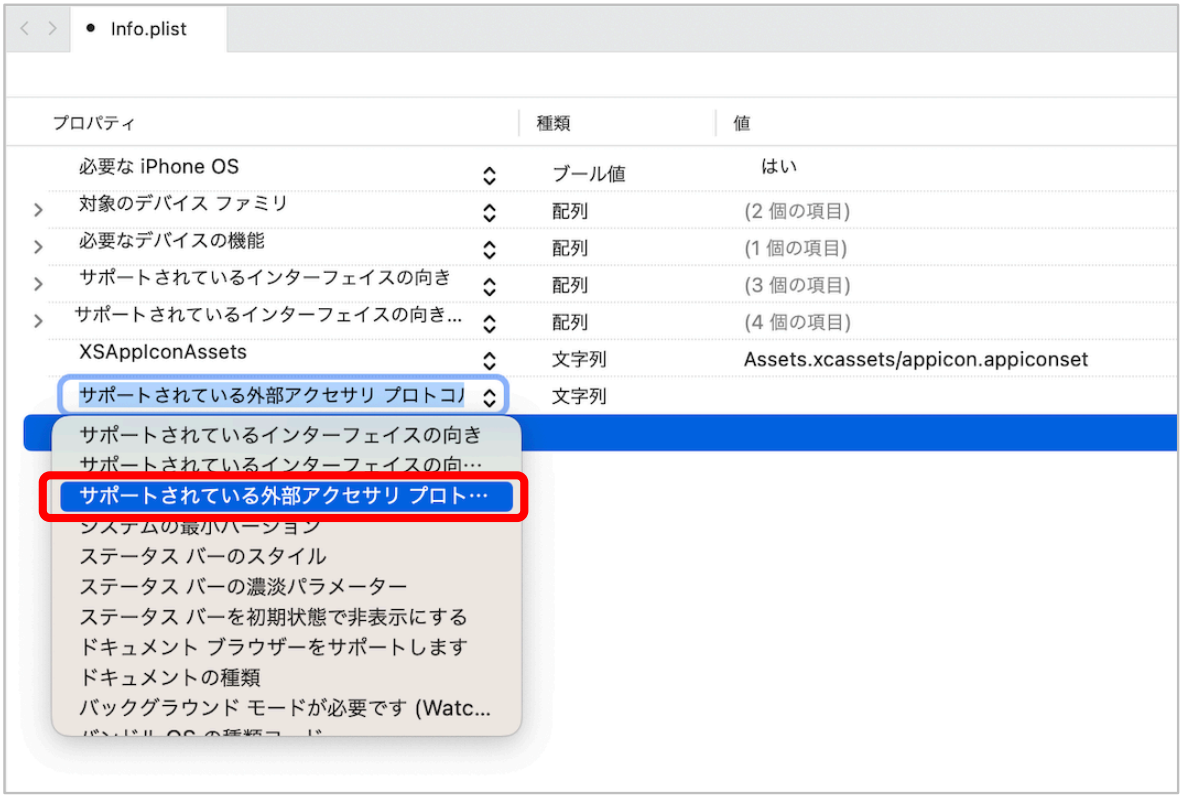
⑬ ソリューションエクスプローラーの[Sample]-[Info.plist]を選択してください。



⑭ 「プロパティ」－「新しいエントリの追加」の⊕を選択してください。



⑮　リストから[サポートされている外部アクセサリプロトコル]を選択してください。



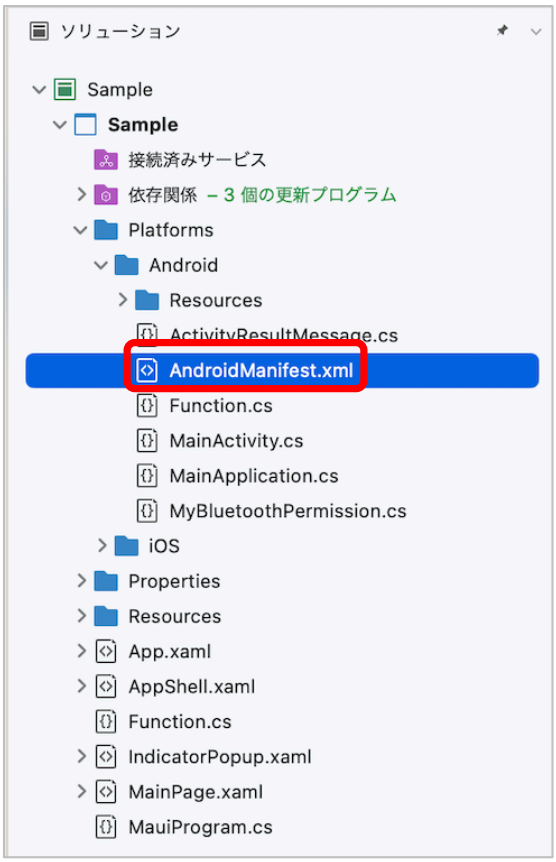
⑯　展開した[サポートされている外部アクセサリプロトコル]に値としてcom.sii-ps.siieapと入力してください。



以上で、ライブラリの機能が利用可能になります。

(3) Android

⑬ ソリューションエクスプローラーの[Sample]-[AndroidManifest.xml]を選択してください。



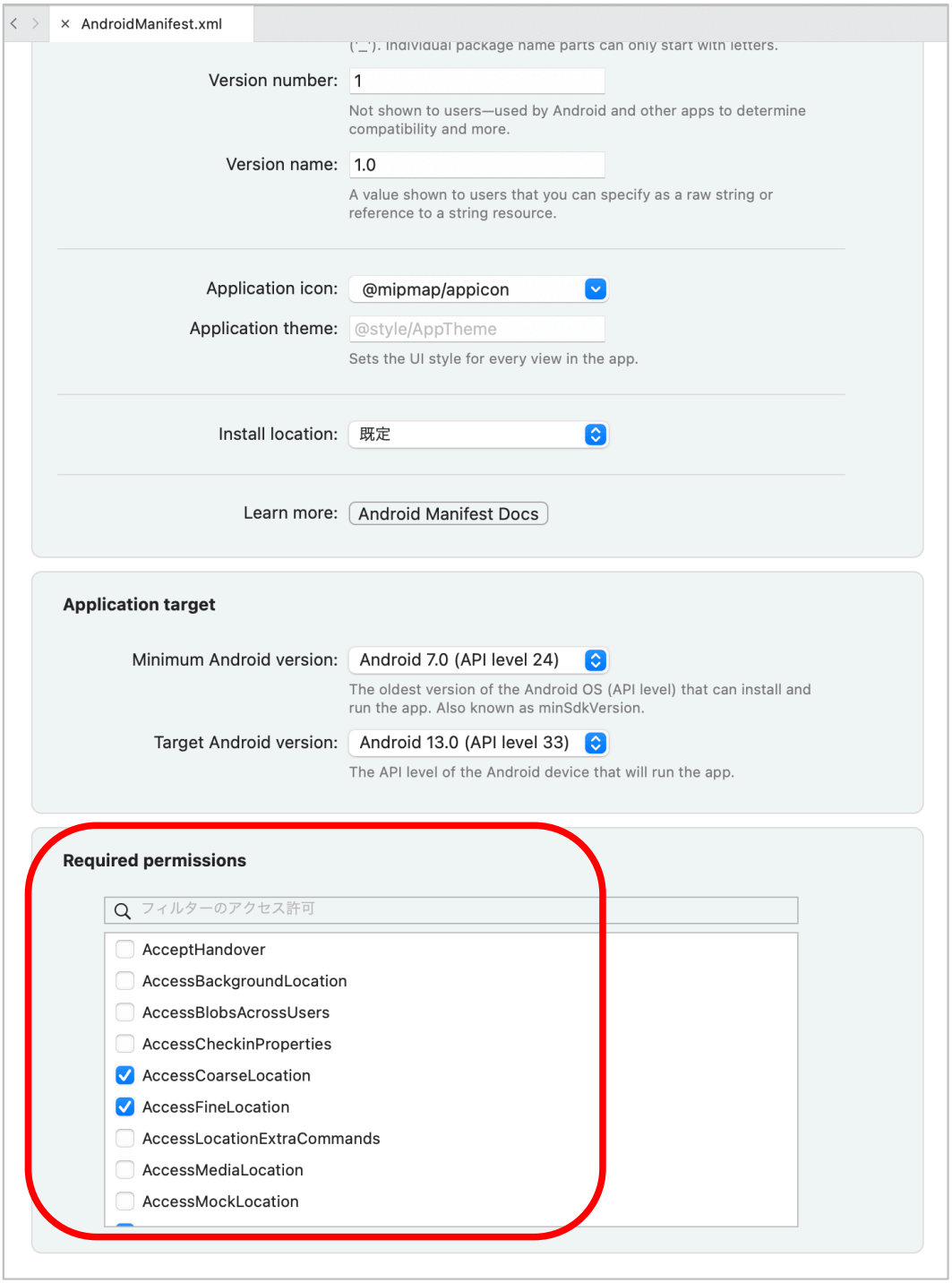
⑭ 「Required permissions」で下記のチェックボックスをオンにしてください。
チェックボックスをオンにしたら、[OK(O)]ボタンをクリックしてください。

[Bluetoothを利用する場合]

- AccessFineLocation
- AccessCoarseLocation
- Bluetooth
- BluetoothAdmin
- BluetoothConnect
- BluetoothScan

[TCP/IPを利用する場合]

- AccessWifiState
- ChangeWifiState
- Internet



⑮ SmartLabelManagerクラスを利用する場合はSampleの「NuGetパッケージの管理...」から「Xamarin.Google.ZXing.Core」パッケージを追加してください。

以上で、ライブラリの機能が利用可能になります。

3.4 注意事項

- **対象範囲別ストレージについて**

  Android 10から導入された"対象範囲別ストレージ"ではアプリ固有のストレージとアプリ外部のストレージが区別されます。
  Android 10 (API 29)以降をアプリのターゲットとする場合、アプリ外部のストレージ内にあるメディアファイルに該当しないファイルを直接扱うことができません。メディアファイルに該当しないファイルは"ストレージ アクセス フレームワーク"を利用すると扱うことができます。
  対象範囲別ストレージの詳細については、下記を参照してください。

  - データ ストレージとファイル ストレージの概要
    https://developer.android.com/training/data-storage

- **.csproj ファイル内で指定する TargetFrameworks について**

  TargetFrameworksには、ライブラリがサポートする.NETバージョンを指定してください。
  サポート範囲外のバージョンを指定した場合、ビルド時にエラー(例: NETSDK1202やNU1202)が発生することがあります。

4章　プリンターのライブラリ機能

本章では、ライブラリに実装されている各クラスのプリンター用APIについて説明します。

4.1　スタンダードモードとページモード

4.1.1　基本動作

ライブラリには、「スタンダードモード」と「ページモード」の2つの印字モードがあります。
下記で「スタンダードモード」と「ページモード」について説明します。

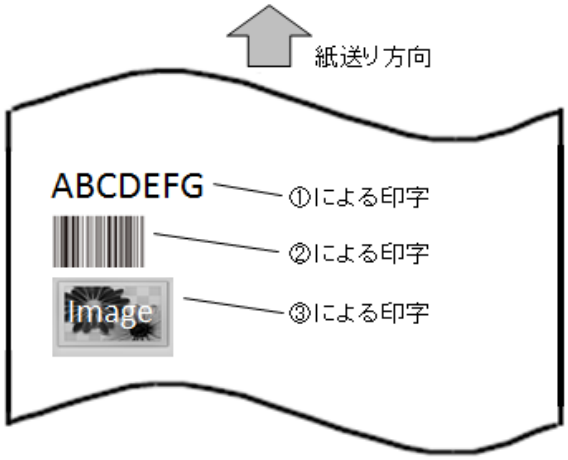
SLP720RT、SLP721RT及びMP-B20は「ページモード」をサポートしていません。

(1)　スタンダードモード

スタンダードモードは、順次印字を行うモードです。

印字命令例

① テキストデータの送信
② バーコードの印字
③ 指定ファイルの送信（イメージファイルを指定）



スタンダードモードは、レシートなどのように非定型の長さでの印字に適しています。

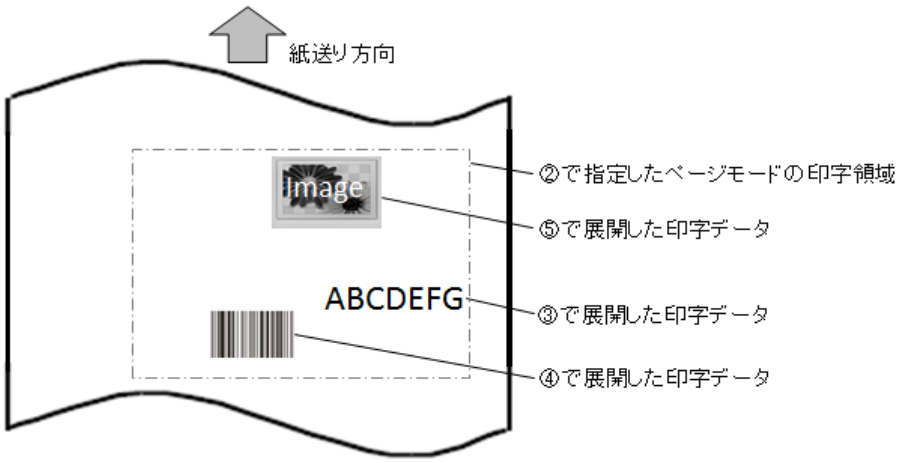
(2) ページモード

ページモードは、1ページ毎に印字を行うモードです。

ページモードでは、初めにページモードの印字領域を確保し、ページモードの印字領域上の任意の位置に印字データを展開していきます。
展開した印字データは、ページモードの印字命令により印字します。

印字命令例

① ページモードの開始
② ページモードの印字領域の指定
③ ページモードのテキストデータの送信
④ ページモードのバーコードの印字
⑤ ページモードのイメージファイルの描画
⑥ ページモードの印字（③④⑤の印字データを②の印字領域で印字）
⑦ ページモードの終了



ページモードは、下記の印字に適しています。

・定型の長さでの印字
・文字の書き出しや罫線の印字位置を座標指定したい印字

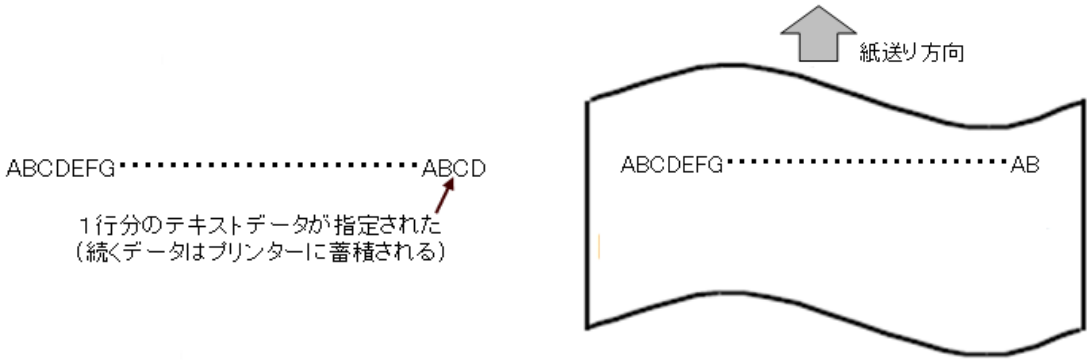
4.1.2　スタンダードモードにおけるテキストデータの印字

スタンダードモードにおけるテキストデータは1行毎に印字します。

1行分に満たないテキストデータを指定した場合は、テキストデータはプリンターに蓄積されます。
蓄積されたテキストデータは、下記のどちらかの条件で印字されます。

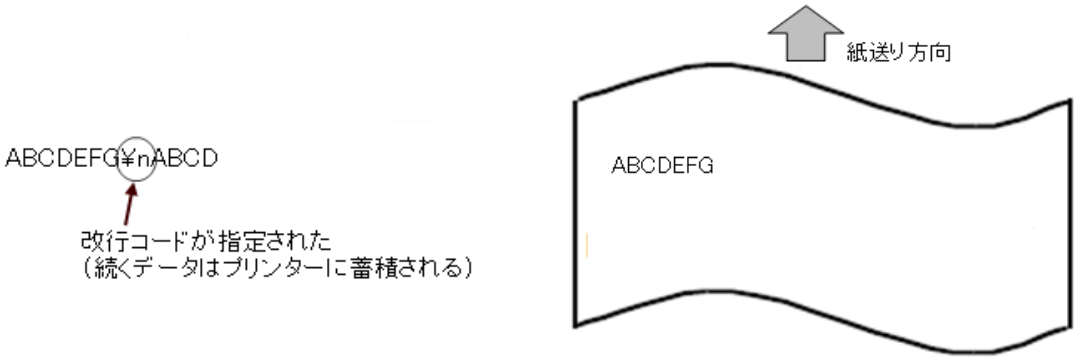
・1行分のテキストデータを指定した場合
・改行コードを指定した場合

- **1 行分のテキストデータを指定した場合の印字処理**



入力データ　　印字結果

- **改行コードを指定した場合の印字処理**



入力データ　　印字結果

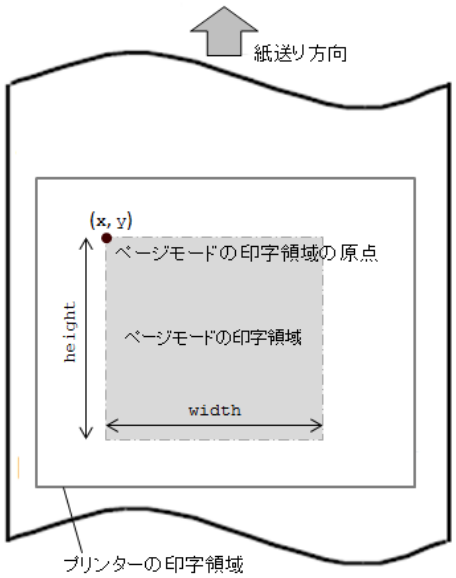
4.1.3　ページモードにおける印字データの展開位置

ページモードでは、印字領域、印字方向、基準点により印字データの展開位置が決まります。
印字領域、印字方向、基準点について説明します。

(1)　印字領域

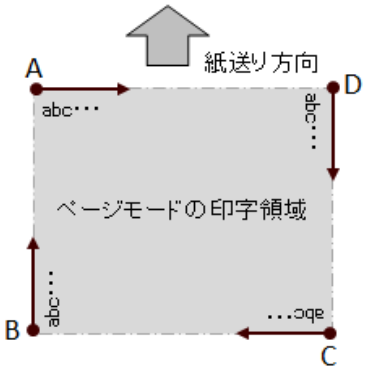
プリンターの印字領域に対して、ページモードの印字領域は、ページモードの印字領域の原点とその領域に対する幅と高さを指定します。下図に領域の考え方を示します。

ページモードの印字領域は、複数指定できます。



(2)　印字方向

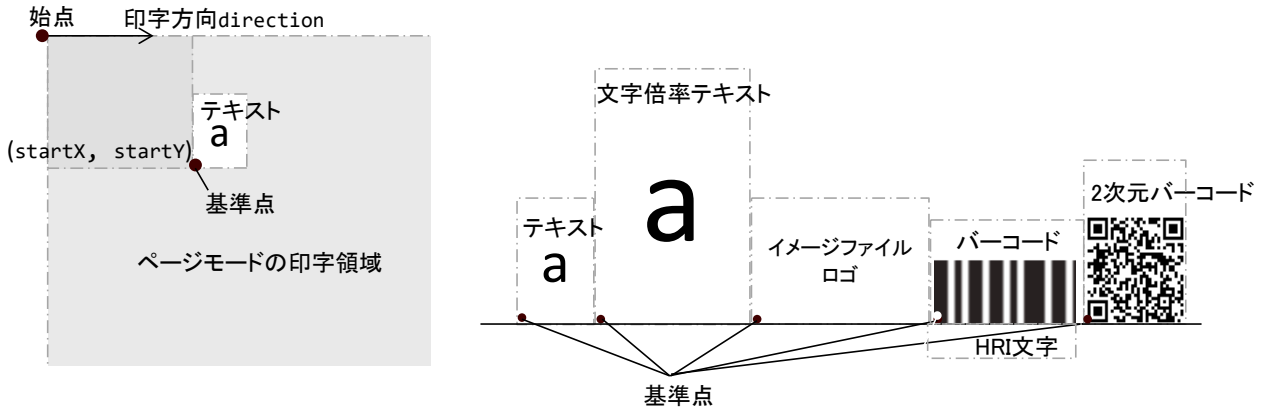
ページモードの印字領域を設定する際に、印字方向を指定します。
各方向の印字方向を指定すると、始点の位置も変わります。
印字方向と始点の関係を下図に示します。



・始点：左上（図のA）、印字方向：左→右
・始点：左下（図のB）、印字方向：下→上
・始点：右下（図のC）、印字方向：右→左
・始点：右上（図のD）、印字方向：上→下

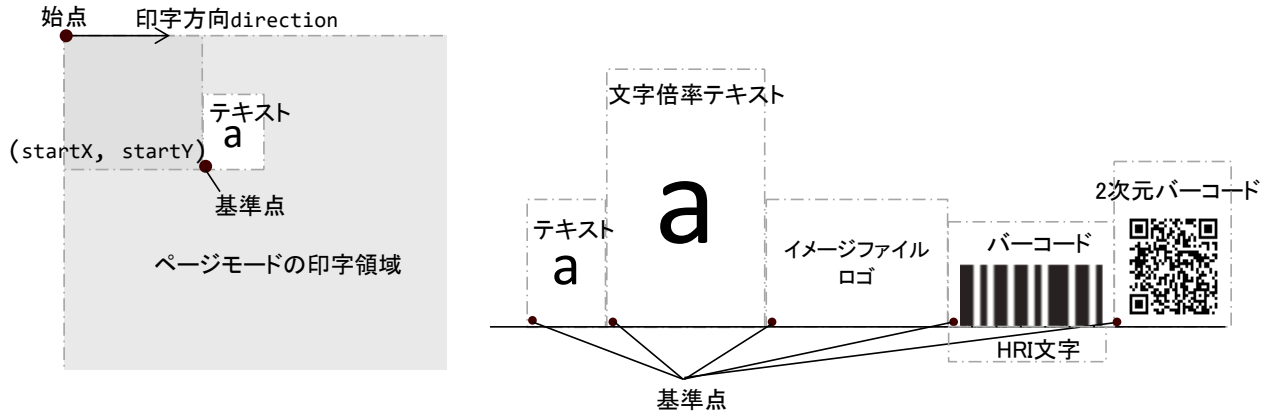
(3) 基準点

ページモードの印字を行う場合の、データを展開する基準点と各印字要素（テキスト、イメージファイル、ロゴ、バーコード等）の関係を下記に示します。



RP-F10、RP-G10、MP-B30、MP-B30L、MP-B21Lの基準点と各印字要素の関係

（注意） ・基準点は、ページモードの印字領域外に指定することはできません。
・バーコードの印字要素はクワイエットゾーンを含みません。



RP-E10、RP-D10、MP-A40の基準点と各印字要素の関係

（注意） ・基準点は、ページモードの印字領域外に指定することはできません。
・バーコードの印字要素はクワイエットゾーンを含みます。

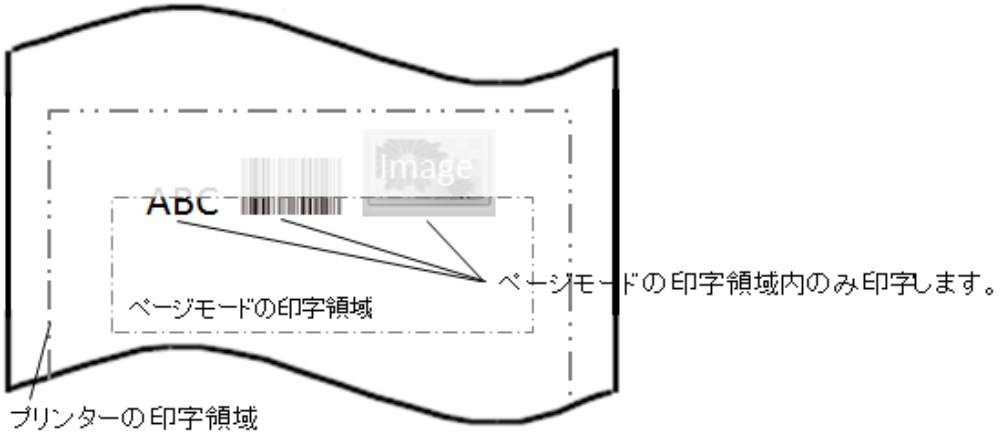
4.1.4　ページモードの印字領域外の印字データ処理

ページモードにおいて、展開した印字データがページモードの印字領域外になった場合の処理について説明します。

印字データ種類

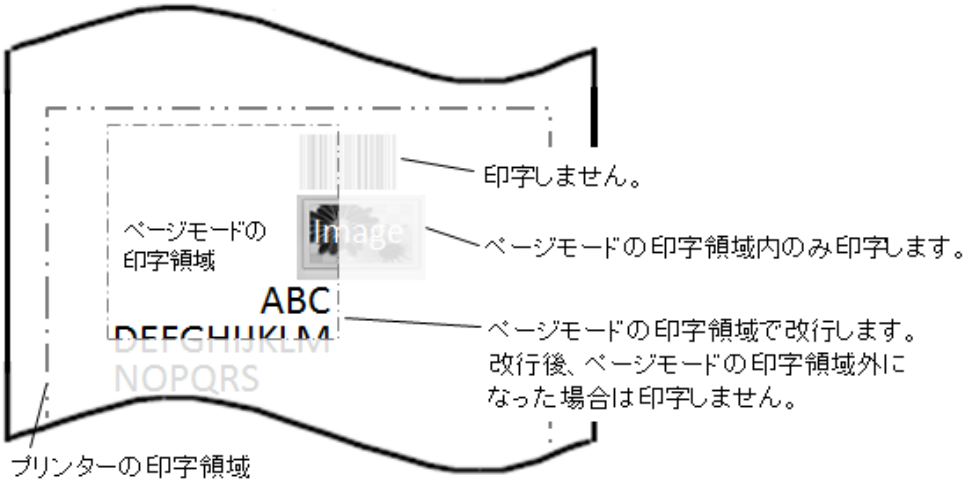
テキスト	バーコード、 2 次元バーコード	イメージファイル、ロゴ、 矩形、罫線
ABC		

(1)　ページモードの印字領域の上部に印字データが展開される場合



（注意）　**BARCODE_CUSTOMER_BAR_CODE_JP**で指定される印字データは印字しません。

(2)　ページモードの印字領域の右側に印字データが展開される場合



（注意）　展開したバーコード印字データの一部がページモードの印字領域外になった場合、読み取りエラーや誤読が発生することがあります。

4.2 ラベル印字機能

ライブラリでは、SII Layout Editorで作成したラベルファイル(*.sl、*.slex)の印字ができます。
また、ラベルファイルを利用し、テキスト、イメージ及びバーコードデータの置き換えを行い、印字できる機能を提供します。

SII Layout Editorはラベル作成ができるソフトウェアです。
本書では、SII Layout Editorの説明をする場合は、以降「アプリ」と称します。

下記のQRコードをスマートフォンで読み取ってストアへ移動し、アプリをインストールしてください。

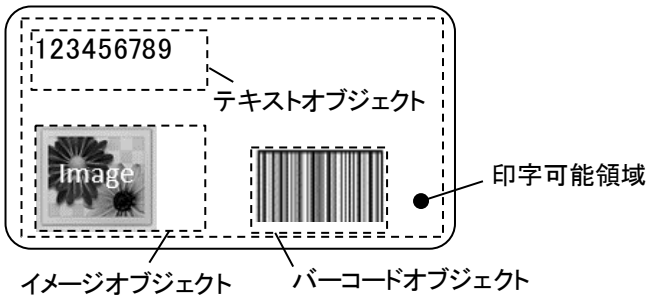
- iOS

- Android

参考　App StoreまたはGoogle Playに移動して、「SII Layout Editor」を検索、インストールすることも可能です。

4.2.1 ラベルファイルの構造

ラベルファイルは、ラベルの印字可能領域内にオブジェクトを配置したファイルです。



ラベルファイル(*.sl、*.slex)の例

(1) オブジェクトの種類とライブラリでのサポート

アプリのオブジェクトを下記に示します。

オブジェクト	説明	ライブラリでのサポート
テキストオブジェクト	テキストデータを扱うオブジェクト	✓
イメージオブジェクト	イメージデータを扱うオブジェクト	✓
バーコードオブジェクト	バーコードデータを扱うオブジェクト	✓
連絡帳オブジェクト	デバイスの連絡帳からデータを取り込むオブジェクト	✓
日付と時刻オブジェクト	日付と時刻のデータを扱うオブジェクト	✓
図(四角形)オブジェクト	図(四角形)の描画データを扱うオブジェクト	✓
図(円/楕円)オブジェクト	図(円/楕円)の描画データを扱うオブジェクト	✓
図(線)オブジェクト	図(線)の描画データを扱うオブジェクト	✓

(2) ラベルファイルをライブラリで印字する際の注意点

アプリを使用したラベルファイルの印字とライブラリを使用したラベルファイルの印字では印字結果が異なる場合がありますので、実機にて事前に確認してください。
ライブラリを使用してラベルファイルを印字する際は、下記の点にご注意ください。

① 全てのオブジェクト

- 印字可能領域外に配置されたオブジェクトの描画はサポートしていません。

② テキストオブジェクト

- 「連続番号」はサポートしていません。
- アプリで設定されているフォントがライブラリにない場合、テキストデータはシステム標準のフォントで印字します。

③ イメージオブジェクト

- ディザリングの設定が「バークス」または「ベイヤー」の場合、ライブラリでは「フロイド・スタンバーグ」となります。

④ バーコードオブジェクト

- アプリでサポートするバーコードのうち、ライブラリでは下記のバーコードをサポートしています。
  - CODE39
  - CODE128
  - ITF(Android のみサポート)
  - UPC-A
  - EAN13
  - CODABAR

    - UPC-E
    - EAN8
    - PDF417
    - Data Matrix
    - QR コード

- 「連続番号」はサポートしていません。
- 下記のバーコード設定は反映しません。
    - バー幅の比率

- アプリで生成されるバーコードイメージとライブラリで生成するバーコードイメージは同一バーコードイメージにはならない場合があります。
- アプリでバーコードオブジェクトの高さをバー高さより小さく指定した場合、ライブラリではバーコードはオブジェクト内に収まるように縮小して印字します。
- アプリでPDF417の「セキュリティ」を-1に設定した場合、ライブラリではセキュリティレベルを0固定で描画します。
- アプリで設定されているフォントがライブラリにない場合、テキストデータはシステム標準のフォントで印字します。

⑤　連絡帳オブジェクト

- アプリで設定されているフォントがライブラリにない場合、テキストデータはシステム標準のフォントで印字します。

⑥　日付と時刻オブジェクト

- アプリで設定されているフォントがライブラリにない場合、テキストデータはシステム標準のフォントで印字します。

⑦　図オブジェクト

- アプリの「線の太さ」が細い場合、破線・長破線・二重線の印字が潰れる可能性があります。
- アプリとライブラリでの図オブジェクトの描画位置が異なる場合があります。

4.2.2 ラベルファイルの利用手順

ラベルファイルを利用した印字手順を下記に示します。

(1) ラベルファイルをそのままライブラリから印字する場合

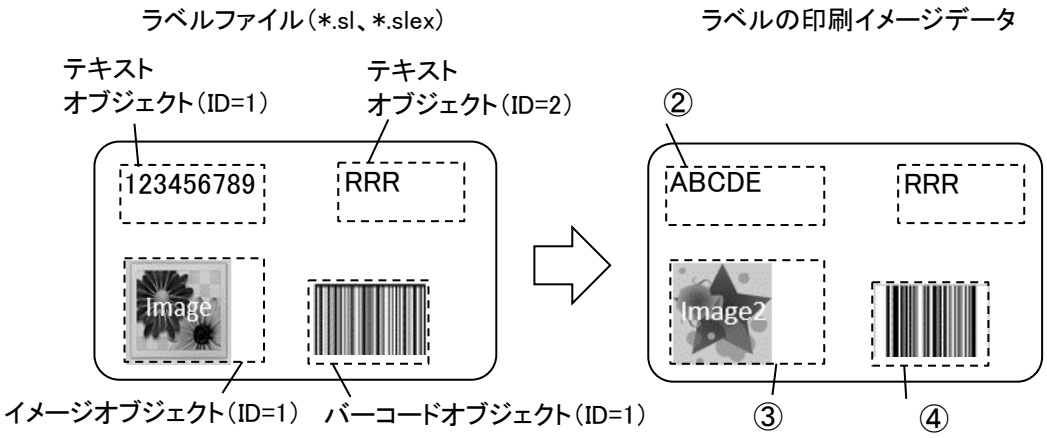
印字命令例

①ラベルファイルの指定
②ラベルの印字

(2) ラベルファイルのオブジェクトデータを置換して印字する場合

印字命令例

①ラベルファイルの指定
②ラベルのテキストデータの置換（テキストオブジェクト ID=1）
③ラベルのイメージデータの置換（イメージオブジェクト ID=1）
④ラベルのバーコードデータの置換（バーコードオブジェクト ID=1）
⑤ラベルの印字



4.3 ログファイル出力機能

ライブラリではログの取得とログファイルの出力ができます。

4.3.1 ログ出力の設定方法

- **iOS の場合**

  ライブラリを組み込んだiOSアプリケーションのDocumentsフォルダに、下記内容のconfig.iniファイルを追加することで、ログの出力設定が可能になります。

- **Android の場合**

  ライブラリを組み込んだAndroidアプリケーションの固有のディレクトリ(例:内部共有ストレージ¥Android¥data¥<package name>¥files)に、下記内容のconfig.iniファイルを追加することで、ログの出力設定が可能になります。

config.ini

```
LOGLEVEL=x
LOGSIZEMAX=xMB
LOGOUTPUT=x
```

参考 xの設定内容については、「4.3.2 ログ出力の設定内容」を参照してください。

4.3.2 ログ出力の設定内容

項目	説明	設定内容
LOGLEVEL	ログレベル	0 : ログを記録しません。 1 : SIIPrinterExceptionスロー時のエラーログを記録します。 2 : APIの実行履歴を記録します。
LOGSIZEMAX	ログファイル最大サイズ	1MB : ログファイル最大サイズは1MB 5MB : ログファイル最大サイズは5MB 10MB : ログファイル最大サイズは10MB 50MB : ログファイル最大サイズは50MB
LOGOUTPUT	システムログ出力有効・無効*1	0 : システムログ出力無効 1 : システムログ出力有効

*1: システムログ出力
・iOSの場合 : Console
・Androidの場合 : Logcat

4.3.3 ログファイル

ログファイルは、ライブラリを組み込んだアプリケーションのローカルファイルとして保存されます。

ログファイル名 : PrinterManagerX.log (Xの範囲は、0～4)

最初のログファイルはPrinterManager0.logで作成されます。ログファイルの最大サイズを超えた場合、ファイル名をPrinterManager1.logに変更し、新たにPrinterManager0.logを作成します。

作成できるログファイルは最大5つです。

4.4 APIリファレンス

ライブラリには、下記のクラスが含まれます。

✓:サポート、-:非サポート

名前	説明	SLP720RT SLP721RT MP-B30L MP-B21L	RP-F10 RP-G10	RP-E10 RP-D10	MP-B30 MP-B20 MP-A40
PrinterManager	プリンターとの通信、印字を行うためのAPIを提供します。 「**4.4.2 PrinterManagerクラス**」を参照してください。 **PrinterManager**クラスで提供するAPIで必要な列挙型定数については、「4.4.1 列挙型定数一覧」を参照してください。	✓	✓	✓	✓
DeviceInfo	StartDiscoveryDeviceで発見されたデバイス情報を格納します。 「**4.4.3 DeviceInfoクラス**」を参照してください。 **DeviceInfo**クラスで提供するAPIで必要な列挙型定数については、「4.4.1 列挙型定数一覧」を参照してください。	✓	✓	✓	✓
PrinterIdentifier	GetPrinterResponseで取得されたプリンターのプリンタIDを格納します。 「**4.4.4 PrinterIdentifierクラス**」を参照してください。 **PrinterIdentifier**クラスで提供するAPIで必要な列挙型定数については、「4.4.1 列挙型定数一覧」を参照してください。	✓	✓	✓	✓
PrinterException	API呼び出し時にスローされる例外クラスです。 「**4.4.5 PrinterExceptionクラス**を参照してください。	✓	✓	✓	✓
SmartLabelManager	ラベルファイルの指定、データの置き換えを行うためのAPIを提供します。 「**4.4.6 SmartLabelManagerクラス**」を参照してください。	✓	-	-	-

参考　プリンター経由でDSP-A01を使用する場合、またはDSP-A01単体で使用する場合のAPIリファレンスについては、「5章 ディスプレイのライブラリ機能」を参照してください。

各クラスにおけるメソッド及びプロパティの詳細説明について記載内容を説明します。

Method or Property　メソッド名またはプロパティ名

メソッドまたはプロパティの概略を記載します。

対象　メソッドをサポートするプラットフォーム、及びデバイスを記載します。

- プラットフォーム

	全てのプラットフォームをサポートする場合	プラットフォームによりサポートするデバイスが変わる場合
「対象」での記載	プラットフォーム共通	iOS
		Android

- デバイス

	複数のデバイスをサポートする場合		サポートするデバイスが個別の場合
「対象」での記載	全デバイス	全プリンター	SLP720RT
			SLP721RT
			RP-F10
			RP-G10
			RP-E10
			RP-D10
			MP-B30
			MP-B30L
			MP-B21L
			MP-B20
			MP-A40
		–	プリンター経由のDSP-A01
			DSP-A01単体

構文　構文を記載します。

パラメータ　パラメータと、パラメータの説明を記載します。

有効範囲　有効範囲がある場合は、有効範囲を記載します。

エラー　エラーが発生した場合の説明を記載します。

初期値　初期値がある場合は、初期値を記載します。

説明　メソッド説明を記載します。

戻り値　戻り値がある場合は、戻り値を記載します。

4.4.1 列挙型定数一覧

列挙型定数一覧を下記に示します。

① デバイスモデル(DeviceModel)

デバイスモデルで利用する列挙型定数を下記に示します。

定数名	説明
DEVICE_MODEL_RP_D10	RP-D10
DEVICE_MODEL_RP_E10	RP-E10
DEVICE_MODEL_MP_B20	MP-B20
DEVICE_MODEL_MP_B21L	MP-B21L
DEVICE_MODEL_MP_B30	MP-B30
DEVICE_MODEL_MP_B30L	MP-B30L
DEVICE_MODEL_MP_A40	MP-A40
DEVICE_MODEL_RP_F10	RP-F10/RP-G10*1
DEVICE_MODEL_SLP720RT	SLP720RT/SLP721RT

*1: Androidのみサポートします。

② ポート種別(PortType)

ポート種別で利用する列挙型定数を下記に示します。

定数名	説明
PORT_TYPE_BLUETOOTH	Bluetoothで接続
PORT_TYPE_USB	USBで接続
PORT_TYPE_TCP	TCP/IPで接続

③ ドロワ番号(DrawerNum)

ドロワ番号で利用する列挙型定数を下記に示します。

定数名	説明
DRAWER_1	ドロワ1
DRAWER_2	ドロワ2

④　パルス幅(PulseWidth)

パルス幅で利用する列挙型定数を下記に示します。ドロワの制御時間については、お客様がご利用されているドロワの仕様に従ってください。

定数名	説明
ON_OFF_TIME_100	ON/OFF時間 100ミリ秒
ON_OFF_TIME_200	ON/OFF時間 200ミリ秒
ON_OFF_TIME_300	ON/OFF時間 300ミリ秒
ON_OFF_TIME_400	ON/OFF時間 400ミリ秒
ON_OFF_TIME_500	ON/OFF時間 500ミリ秒
ON_OFF_TIME_600	ON/OFF時間 600ミリ秒
ON_OFF_TIME_700	ON/OFF時間 700ミリ秒
ON_OFF_TIME_800	ON/OFF時間 800ミリ秒

⑤　ブザーパターン(BuzzerPattern)

外部ブザーのブザーパターンで利用する列挙型定数を下記に示します。

定数名	説明
BUZZER_PATTERN_1	パターン1
BUZZER_PATTERN_2	パターン2
BUZZER_PATTERN_3	パターン3
BUZZER_PATTERN_4	パターン4

⑥　ディザリング(Dithering)

ディザリングで利用する列挙型定数を下記に示します。

定数名	説明
DITHERING_DISABLE	ディザリング無効
DITHERING_ERRORDIFFUSION	ディザリング有効

⑦　プリンター応答種別(PrinterResponseId)

プリンター応答種別で利用する列挙型定数を下記に示します。

定数名	説明
PRINTER_RESPONSE_REQUEST	実行応答リクエスト
PRINTER_RESPONSE_USER_AREA	ユーザ領域の残り容量の送信
PRINTER_RESPONSE_ARRANGE_USER_AREA	ユーザ領域の整理後の残り容量の送信
PRINTER_RESPONSE_NV_GRAPHICS	NVグラフィックスのメモリ容量の送信
PRINTER_RESPONSE_KEY_CODE	定義されているNVグラフィックスのキーコード一覧の送信
PRINTER_RESPONSE_BATTERY_STATUS[*1]	バッテリ残量レベル
PRINTER_RESPONSE_MAINTENANCE_COUNTER	メンテナンスカウンタの送信
PRINTER_RESPONSE_PRINTER_IDENTIFIER	プリンタIDの送信

*1: MP-B30、MP-B30L、MP-B21L、MP-B20及びMP-A40のみサポートします。

⑧　デバイスタイプ(DeviceType)

デバイスタイプで利用する列挙型定数を下記に示します。

定数名	説明
TYPE_PRINTER	プリンター
TYPE_DISPLAY	ディスプレイ

⑨　一括処理選択(TransactionFunction)

一括処理選択で利用する列挙型定数を下記に示します。

定数名	説明
TRANSACTION_CLEAR	一括処理の中止
TRANSACTION_START	一括処理の開始
TRANSACTION_PRINT	一括印字と一括処理の終了

⑩　位置揃え(PrintAlignment)

位置揃えで利用する列挙型定数を下記に示します。
改行前のテキストデータに位置揃えを追加することはできません。

定数名	説明
ALIGNMENT_LEFT	左揃え
ALIGNMENT_CENTER	中央揃え
ALIGNMENT_RIGHT	右揃え

⑪　強調文字(CharacterBold)

強調文字で利用する列挙型定数を下記に示します。

定数名	説明
BOLD_CANCEL	強調文字を解除
BOLD	強調文字を指定

⑫　アンダーライン(CharacterUnderline)

アンダーラインで利用する列挙型定数を下記に示します。

定数名	説明
UNDERLINE_CANCEL	アンダーライン印字を解除
UNDERLINE_1	1ドット幅アンダーライン印字を指定
UNDERLINE_2	2ドット幅アンダーライン印字を指定

⑬　文字倍率(CharacterScale)

文字倍率で利用する列挙型定数を下記に示します。

定数名	説明
VERTICAL_1_HORIZONTAL_1	縦1倍・横1倍
VERTICAL_1_HORIZONTAL_2	縦1倍・横2倍
VERTICAL_1_HORIZONTAL_3	縦1倍・横3倍
VERTICAL_1_HORIZONTAL_4	縦1倍・横4倍
VERTICAL_2_HORIZONTAL_1	縦2倍・横1倍
VERTICAL_2_HORIZONTAL_2	縦2倍・横2倍
VERTICAL_2_HORIZONTAL_3	縦2倍・横3倍
VERTICAL_2_HORIZONTAL_4	縦2倍・横4倍
VERTICAL_2_HORIZONTAL_6	縦2倍・横6倍
VERTICAL_3_HORIZONTAL_1	縦3倍・横1倍
VERTICAL_3_HORIZONTAL_2	縦3倍・横2倍
VERTICAL_3_HORIZONTAL_3	縦3倍・横3倍
VERTICAL_3_HORIZONTAL_4	縦3倍・横4倍
VERTICAL_4_HORIZONTAL_1	縦4倍・横1倍
VERTICAL_4_HORIZONTAL_2	縦4倍・横2倍
VERTICAL_4_HORIZONTAL_3	縦4倍・横3倍
VERTICAL_4_HORIZONTAL_4	縦4倍・横4倍
VERTICAL_4_HORIZONTAL_6	縦4倍・横6倍

定数名	説明
VERTICAL_4_HORIZONTAL_8	縦4倍・横8倍
VERTICAL_6_HORIZONTAL_2	縦6倍・横2倍
VERTICAL_6_HORIZONTAL_4	縦6倍・横4倍
VERTICAL_6_HORIZONTAL_6	縦6倍・横6倍
VERTICAL_6_HORIZONTAL_8	縦6倍・横8倍
VERTICAL_8_HORIZONTAL_4	縦8倍・横4倍
VERTICAL_8_HORIZONTAL_6	縦8倍・横6倍
VERTICAL_8_HORIZONTAL_8	縦8倍・横8倍

⑭　文字フォント（CharacterFont）

文字フォントで利用する列挙型定数を下記に示します。

定数名	説明
FONT_A	フォントA（24×12）
FONT_B	フォントB（16×8）

⑮　モジュールサイズ（ModuleSize）

バーコードの幅、公称細エレメント幅、及びモジュールサイズで利用する列挙型定数を下記に示します。

定数名	説明	使用メソッド
BARCODE_MODULE_WIDTH_2	細エレメント 2ドット モジュール幅 0.250 mm	● PrintBarcode ● PrintPageModeBarcode
BARCODE_MODULE_WIDTH_3	細エレメント 3ドット モジュール幅 0.375 mm	
BARCODE_MODULE_WIDTH_4	細エレメント 4ドット モジュール幅 0.500 mm	
BARCODE_MODULE_WIDTH_5	細エレメント 5ドット モジュール幅 0.625 mm	
BARCODE_MODULE_WIDTH_6	細エレメント 6ドット モジュール幅 0.750 mm	
CUSTOMERBARCODEJP_MODULE_4	カスタマバーコードサイズ 4ポイント*1	
CUSTOMERBARCODEJP_MODULE_5	カスタマバーコードサイズ 5ポイント*1	
CUSTOMERBARCODEJP_MODULE_6	カスタマバーコードサイズ 6ポイント*1	
CUSTOMERBARCODEJP_MODULE_7	カスタマバーコードサイズ 7ポイント*1	

定数名	説明	使用メソッド
CUSTOMERBARCODEJP_MODULE_8	カスタマバーコードサイズ 8ポイント*1	● PrintBarcode ● PrintPageModeBarcode
CUSTOMERBARCODEJP_MODULE_9	カスタマバーコードサイズ 9ポイント*1	
CUSTOMERBARCODEJP_MODULE_10	カスタマバーコードサイズ 10ポイント*1	

*1: カスタマバーコードのポイントによるバーの構成は下記になります。

カスタマバーコードサイズ	構成(mm)				
	ロングバー	タイミングバー	バーピッチ	バー幅	バースペース
CUSTOMERBARCODEJP_MODULE_4	1.500 (12ドット)	0.500 (4ドット)	0.500 (4ドット)	0.250 (2ドット)	0.250 (2ドット)
CUSTOMERBARCODEJP_MODULE_5	1.875 (15ドット)	0.625 (5ドット)	0.625 (5ドット)	0.375 (3ドット)	0.250 (2ドット)
CUSTOMERBARCODEJP_MODULE_6	2.250 (18ドット)	0.750 (6ドット)	0.750 (6ドット)	0.375 (3ドット)	0.375 (3ドット)
CUSTOMERBARCODEJP_MODULE_7	2.625 (21ドット)	0.875 (7ドット)	0.875 (7ドット)	0.500 (4ドット)	0.375 (3ドット)
CUSTOMERBARCODEJP_MODULE_8	3.000 (24ドット)	1.000 (8ドット)	1.000 (8ドット)	0.500 (4ドット)	0.500 (4ドット)
CUSTOMERBARCODEJP_MODULE_9	3.375 (27ドット)	1.125 (9ドット)	1.125 (9ドット)	0.625 (5ドット)	0.500 (4ドット)
CUSTOMERBARCODEJP_MODULE_10	3.750 (30ドット)	1.250 (10ドット)	1.250 (10ドット)	0.625 (5ドット)	0.625 (5ドット)

定数名	説明	使用メソッド
PDF417_MODULE_WIDTH_2	公称細エレメント幅 2ドット	● PrintPDF417 ● PrintPageModePDF417
PDF417_MODULE_WIDTH_3	公称細エレメント幅 3ドット	
PDF417_MODULE_WIDTH_4	公称細エレメント幅 ドット	
PDF417_MODULE_WIDTH_5*1	公称細エレメント幅 5ドット	
PDF417_MODULE_WIDTH_6*1	公称細エレメント幅 6ドット	
PDF417_MODULE_WIDTH_7*1	公称細エレメント幅 7ドット	
PDF417_MODULE_WIDTH_8*1	公称細エレメント幅 8ドット	

定数名	説明	使用メソッド
QR_MODULE_SIZE_2	2ドット	● PrintQRcode ● PrintPageModeQRcode
QR_MODULE_SIZE_3	3ドット	
QR_MODULE_SIZE_4	4ドット	
QR_MODULE_SIZE_5	5ドット	
QR_MODULE_SIZE_6	6ドット	
QR_MODULE_SIZE_7	7ドット	
QR_MODULE_SIZE_8	8ドット	
QR_MODULE_SIZE_9	9ドット	
QR_MODULE_SIZE_10	10ドット	
QR_MODULE_SIZE_11	11ドット	
QR_MODULE_SIZE_12[*1]	12ドット	
QR_MODULE_SIZE_13[*1]	13ドット	
QR_MODULE_SIZE_14[*1]	14ドット	
QR_MODULE_SIZE_15[*1]	15ドット	
QR_MODULE_SIZE_16[*1]	16ドット	
DATAMATRIX_MODULE_SIZE_2	2ドット	● PrintDataMatrix ● PrintPageModeDataMatrix
DATAMATRIX_MODULE_SIZE_3	3ドット	
DATAMATRIX_MODULE_SIZE_4	4ドット	
DATAMATRIX_MODULE_SIZE_5	5ドット	
DATAMATRIX_MODULE_SIZE_6	6ドット	
DATAMATRIX_MODULE_SIZE_7	7ドット	
DATAMATRIX_MODULE_SIZE_8	8ドット	
DATAMATRIX_MODULE_SIZE_9	9ドット	
DATAMATRIX_MODULE_SIZE_10	10ドット	
DATAMATRIX_MODULE_SIZE_11	11ドット	
DATAMATRIX_MODULE_SIZE_12[*1]	12ドット	
DATAMATRIX_MODULE_SIZE_13[*1]	13ドット	
DATAMATRIX_MODULE_SIZE_14[*1]	14ドット	
DATAMATRIX_MODULE_SIZE_15[*1]	15ドット	
DATAMATRIX_MODULE_SIZE_16[*1]	16ドット	

定数名	説明	使用メソッド
GS1DATABAR_MODULE_SIZE_2[*1]	2ドット	● PrintGS1DataBarStacked ● PrintGS1DataBarStackedOmnidirectional ● PrintGS1DataBarExpandedStacked ● PrintPageModeGS1DataBarStacked ● PrintPageModeGS1DataBarStackedOmnidirectional ● PrintPageModeGS1DataBarExpandedStacked
GS1DATABAR_MODULE_SIZE_3[*1]	3ドット	
GS1DATABAR_MODULE_SIZE_4[*1]	4ドット	
GS1DATABAR_MODULE_SIZE_5[*1]	5ドット	
GS1DATABAR_MODULE_SIZE_6[*1]	6ドット	
GS1DATABAR_MODULE_SIZE_7[*1]	7ドット	
GS1DATABAR_MODULE_SIZE_8[*1]	8ドット	
GS1DATABAR_MODULE_SIZE_9[*1]	9ドット	
GS1DATABAR_MODULE_SIZE_10[*1]	10ドット	
GS1DATABAR_MODULE_SIZE_11[*1]	11ドット	
GS1DATABAR_MODULE_SIZE_12[*1]	12ドット	
GS1DATABAR_MODULE_SIZE_13[*1]	13ドット	
GS1DATABAR_MODULE_SIZE_14[*1]	14ドット	
GS1DATABAR_MODULE_SIZE_15[*1]	15ドット	
GS1DATABAR_MODULE_SIZE_16[*1]	16ドット	
AZTECCODE_MODULE_SIZE_2[*2]	2ドット	● PrintAztecCode ● PrintPageModeAztecCode
AZTECCODE_MODULE_SIZE_3[*2]	3ドット	
AZTECCODE_MODULE_SIZE_4[*2]	4ドット	
AZTECCODE_MODULE_SIZE_5[*2]	5ドット	
AZTECCODE_MODULE_SIZE_6[*2]	6ドット	
AZTECCODE_MODULE_SIZE_7[*2]	7ドット	
AZTECCODE_MODULE_SIZE_8[*2]	8ドット	
AZTECCODE_MODULE_SIZE_9[*2]	9ドット	
AZTECCODE_MODULE_SIZE_10[*2]	10ドット	
AZTECCODE_MODULE_SIZE_11[*2]	11ドット	
AZTECCODE_MODULE_SIZE_12[*2]	12ドット	
AZTECCODE_MODULE_SIZE_13[*2]	13ドット	
AZTECCODE_MODULE_SIZE_14[*2]	14ドット	
AZTECCODE_MODULE_SIZE_15[*2]	15ドット	
AZTECCODE_MODULE_SIZE_16[*2]	16ドット	

*1: RP-E10及びRP-D10はサポートしていません。
*2: MP-B30L及びMP-B21Lのみサポートしします。

⑯　エラー訂正レベル(ErrorCorrection)

エラー訂正レベルで利用する列挙型定数を下記に示します。

定数名	説明	使用メソッド
PDF417_ERROR_CORRECTION_0	エラー訂正レベル 0	● PrintPDF417 ● PrintPageModePDF417
PDF417_ERROR_CORRECTION_1	エラー訂正レベル 1	
PDF417_ERROR_CORRECTION_2	エラー訂正レベル 2	
PDF417_ERROR_CORRECTION_3	エラー訂正レベル 3	
PDF417_ERROR_CORRECTION_4	エラー訂正レベル 4	
PDF417_ERROR_CORRECTION_5	エラー訂正レベル 5	
PDF417_ERROR_CORRECTION_6	エラー訂正レベル 6	
PDF417_ERROR_CORRECTION_7	エラー訂正レベル 7	
PDF417_ERROR_CORRECTION_8	エラー訂正レベル 8	
QR_ERROR_CORRECTION_L	エラー訂正レベル L	● PrintQRcode ● PrintPageModeQRcode
QR_ERROR_CORRECTION_M	エラー訂正レベル M	
QR_ERROR_CORRECTION_H	エラー訂正レベル H	
QR_ERROR_CORRECTION_Q	エラー訂正レベル Q	

⑰　白黒反転印字(CharacterReverse)

白黒反転印字で利用する列挙型定数を下記に示します。

定数名	説明
REVERSE_CANCEL	白黒反転印字を解除
REVERSE	白黒反転印字を指定

⑱　倒立印字(CharacterInversion)

倒立印字で利用する列挙型定数を下記に示します。
改行前のテキストデータに倒立印字を追加することはできません。

定数名	説明
INVERSION_CANCEL	倒立印字を解除
INVERSION	倒立印字を指定

⑲　バーコードシンボル(BarcodeSymbol)

バーコードシンボルで利用する列挙型定数を下記に示します。

定数名	説明	構文*1
BARCODE_UPC_A	UPC-A	(a)
BARCODE_UPC_E	UPC-E	(a)
BARCODE_EAN13	EAN13	(a)
BARCODE_JAN13	JAN13	(a)
BARCODE_EAN8	EAN8	(a)
BARCODE_JAN8	JAN8	(a)
BARCODE_CODE39	CODE39	(a), (b)
BARCODE_CODE93	CODE93	(c)
BARCODE_CODE128	CODE128	(c)
BARCODE_ITF	ITF	(a), (b)
BARCODE_CODABAR	CODABAR	(a), (b)
BARCODE_EAN13_ADDON	EAN13 add-on	(a)
BARCODE_JAN13_ADDON	JAN13 add-on	(a)
BARCODE_CUSTOMER_BAR_CODE_JP*2	カスタマバーコード	(d)
BARCODE_GS1_OMNI_DIRECTIONAL*3	GS1 Databar Omni-directional	(a)
BARCODE_GS1_TRUNCATED*3	GS1 Databar Truncated	(a)
BARCODE_GS1_LIMITED*3	GS1 Databar Limited	(a)
BARCODE_GS1_EXPANDED*3	GS1 Databar Expanded	(a)

*1:　構文の内容は、PrintBarcodeまたはPrintPageModeBarcodeを参照してください。
*2:　MP-B21Lのみサポートします。
*3:　RP-E10及びRP-D10はサポートしていません。

⑳　HRI文字印字位置(HriPosition)

HRI文字印字位置で利用する列挙型定数を下記に示します。

定数名	説明
HRI_NONE	印字しない
HRI_POSITION_ABOVE	バーコードの上
HRI_POSITION_BELOW	バーコードの下
HRI_POSITION_ABOVE_BELOW	バーコードの上と下(両方)

㉑ NW比（NwRatio）

NW比で利用する列挙型定数を下記に示します。

定数名	説明
NWRATIO_1TO2	1:2
NWRATIO_1TO2_5	1:2.5
NWRATIO_1TO3	1:3

㉒ PDF417シンボル（Pdf417Symbol）

PDF417シンボルで利用する列挙型定数を下記に示します。

定数名	説明
PDF417_STANDARD	PDF417
PDF417_COMPACT	コンパクトPDF417

㉓ QRコードモデル（QrModel）

QRコードモデルで利用する列挙型定数を下記に示します。

定数名	説明
QR_MODEL_1	QRコードモデル1
QR_MODEL_2	QRコードモデル2

㉔ Data Matrixモジュール(DataMatrixModule)

Data Matrixモジュールで利用する列挙型定数を下記に示します。

定数名	説明
DATA_MATRIX_AUTO	モジュール数自動
DATA_MATRIX_10_10	モジュール数10×10
DATA_MATRIX_12_12	モジュール数12×12
DATA_MATRIX_14_14	モジュール数14×14
DATA_MATRIX_16_16	モジュール数16×16
DATA_MATRIX_18_18	モジュール数18×18
DATA_MATRIX_20_20	モジュール数20×20
DATA_MATRIX_22_22	モジュール数22×22
DATA_MATRIX_24_24	モジュール数24×24
DATA_MATRIX_26_26	モジュール数26×26
DATA_MATRIX_32_32	モジュール数32×32
DATA_MATRIX_36_36	モジュール数36×36
DATA_MATRIX_40_40	モジュール数40×40
DATA_MATRIX_44_44	モジュール数44×44
DATA_MATRIX_48_48	モジュール数48×48
DATA_MATRIX_52_52	モジュール数52×52
DATA_MATRIX_64_64	モジュール数64×64
DATA_MATRIX_72_72	モジュール数72×72
DATA_MATRIX_80_80	モジュール数80×80
DATA_MATRIX_88_88	モジュール数88×88
DATA_MATRIX_96_96	モジュール数96×96
DATA_MATRIX_104_104	モジュール数104×104
DATA_MATRIX_120_120	モジュール数120×120
DATA_MATRIX_132_132	モジュール数132×132
DATA_MATRIX_144_144	モジュール数144×144
DATA_MATRIX_8_18	モジュール数8×18
DATA_MATRIX_8_32	モジュール数8×32
DATA_MATRIX_12_26	モジュール数12×26
DATA_MATRIX_12_36	モジュール数12×36
DATA_MATRIX_16_36	モジュール数16×36
DATA_MATRIX_16_48	モジュール数16×48

㉕ MaxiCodeモード(MaxiCodeMode)

MaxiCodeモードで利用する列挙型定数を下記に示します。

定数名	説明
MAXI_CODE_2	Mode2
MAXI_CODE_3	Mode3
MAXI_CODE_4	Mode4
MAXI_CODE_5	Mode5

㉖ Aztecシンボル(AztecSymbol)

Aztecシンボルで利用する列挙型定数を下記に示します。

定数名	説明
AZTECCODE_FULLRANGE	フルレンジモード
AZTECCODE_COMPACT	コンパクトモード

㉗ カット方法(CuttingMethod)

カット方法で利用する列挙型定数を下記に示します。

- SLP720RT、SLP721RT、RP-F10、RP-G10、RP-E10

定数名	説明	
	用紙カット位置までの紙送り	カット方法
CUT_FULL	あり	フルカット
CUT_FULL_NO_FEED	なし	
CUT_PARTIAL	あり	パーシャルカット
CUT_PARTIAL_NO_FEED	なし	
CUT_NONE*1	なし	カットなし

*1: PrintPageModeのみサポートします。

- MP-B30、MP-B30L、MP-B21L、MP-B20、MP-A40

定数名	説明
CUT_FULL	カットなし 用紙カット位置までの紙送り
CUT_PARTIAL	
CUT_NONE*1	カットなし

*1: PrintPageModeのみサポートします。

㉘　頭出し位置(FeedPosition)

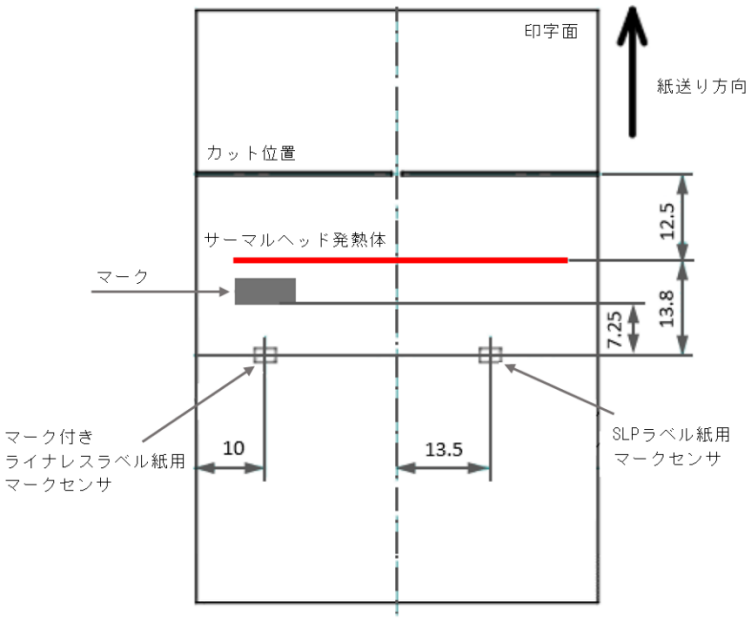
マーク紙またはラベル紙の頭出し位置で利用する列挙型定数を下記に示します。

定数名	説明
FEED_CUTTER[*1]	マークまたはギャップ検出後、カット位置まで紙送りを行います。 紙送り長さはプリンターのメモリスイッチ[*2*3*4]で指定された長さになります。
FEED_NEXT_TOF[*5]	次のマークまたはギャップ検出後、印字位置まで紙送りを行います。 紙送り長さはプリンターのメモリスイッチ[*6]で指定された長さになります。

*1:　SLP720RT、SLP721RT、MP-B30L、MP-B21L及びMP-A40のみサポートします。
*2:　SLP720RT及びSLP721RTの場合、メモリスイッチMS8～9(マーク位置補正)の長さになります。紙送り長さの初期値は58ドット(7.25mm)です。
*3:　MP-B30L及びMP-B21Lの場合、メモリスイッチMS21～22(マーク検出カット位置補正)の長さになります。
紙送り長さの初期値はMP-B30L:125ドット(15.6mm)、MP-B21L:122ドット(15.2mm)です。
*4:　MP-A40の場合、メモリスイッチMS21～22(マーク位置補正)の長さになります。
*5:　MP-B30L及びMP-B21Lのみサポートします。
*6:　MP-B30L及びMP-B21Lの場合、メモリスイッチMS26～27(マーク検出印字位置補正)の長さになります。
紙送り長さの初期値はMP-B30L:125ドット(15.6mm)、MP-B21L:122ドット(15.2mm)です。

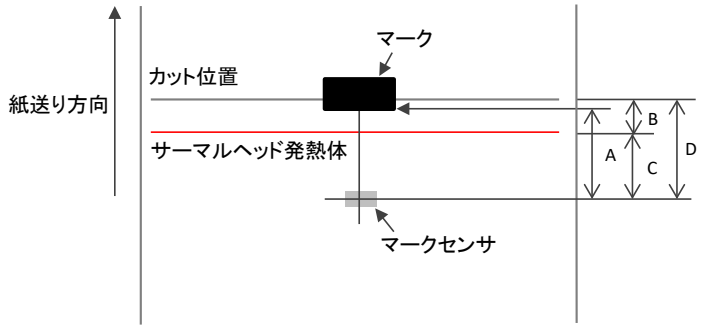
参考　　プリンターのメモリスイッチの詳細については、「取扱説明書」を参照してください。
プリンターのメモリスイッチは、iOSの場合はApp StoreのiOSアプリ「SII Printer Utility」、Androidの場合はGoogle PlayのAndroidアプリ「SII Printer Utility」で変更できます。

SLP720RT及びSLP721RTのセンサ位置とプリンターのメモリスイッチMS8～9(マーク位置補正)の初期値の関係を下図に示します。



単位:mm

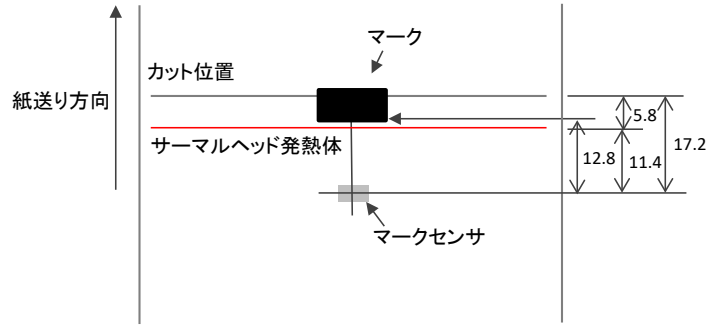
MP-B30L及びMP-B21Lのセンサ位置とプリンターのメモリスイッチMS21～22（マーク検出カット位置補正）、及びMS26～27（マーク検出印字位置補正）の初期値の関係を下図に示します。
プリンターのメモリスイッチが初期値の場合は、**FEED_CUTTER**のカット位置、及び**FEED_NEXT_TOF**の次の印字位置が同じ紙送り長さになります。



単位：mm

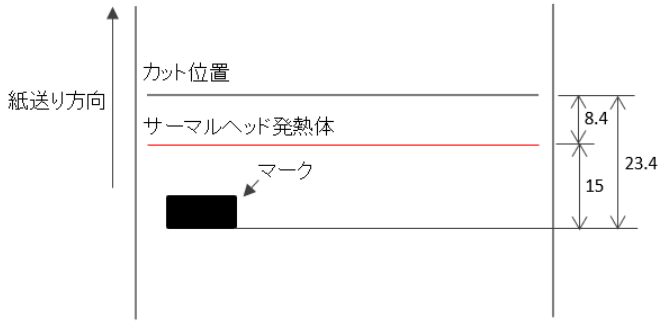
MP-B30L：A=15.6、B=5.8、C=11.4、D=17.2
MP-B21L：A=15.2、B=5.4、C=11.3、D=16.7

用紙の節約などで、次の印字位置をマークに対して短く設定したい場合は、プリンターのメモリスイッチMS26～27（マーク検出印字位置補正）の値を短く設定します。
例として、プリンターのメモリスイッチMS26～27の値を103ドット（12.8mm）に設定し、**FEED_NEXT_TOF**を指定して用紙の頭出しを実行した場合のセンサ位置の関係をMP-B30Lを例に下図に示します。



単位：mm

MP-A40のマーク位置とプリンターのメモリスイッチMS21～22（マーク位置補正）の初期値の関係を下図に示します。



単位：mm

注意　MP-B30L及びMP-B21Lでラベル紙を使用する場合は、印字位置がラベルの中になるようにプリンターのメモリスイッチMS26～27（マーク検出印字位置補正）の値を設定してください。

㉙　イメージの回転方向(Rotate)

イメージの回転方向で利用する列挙型定数を下記に示します。

定数名	説明
ROTATE_NONE	回転なし
ROTATE_180	180度回転

㉚　イメージの拡大縮小(ImageScale)

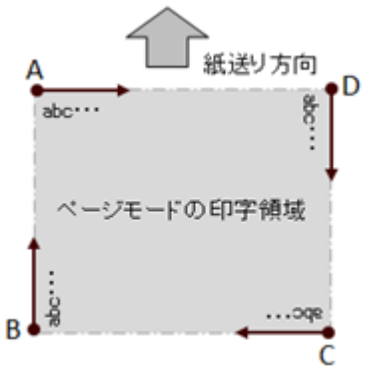
イメージの拡大縮小で利用する列挙型定数を下記に示します。

定数名	説明
IMAGE_SCALE_NONE	拡大縮小なし
IMAGE_SCALE_WIDTH_FIT	印字幅に合わせて拡大縮小

㉛　印字方向(Direction)

ページモードの印字方向で利用する列挙型定数を下記に示します。

定数名	説明
DIRECTION_LEFT_TO_RIGHT	始点:左上(図のA)、印字方向:左→右
DIRECTION_BOTTOM_TO_TOP	始点:左下(図のB)、印字方向:下→上
DIRECTION_RIGHT_TO_LEFT	始点:右下(図のC)、印字方向:右→左
DIRECTION_TOP_TO_BOTTOM	始点:右上(図のD)、印字方向:上→下



㉜ 線種類(LineStyle)

ページモードの線種類で利用する列挙型定数を下記に示します。

定数名	説明
LINESTYLE_THIN	細実線(2ドット)
LINESTYLE_MEDIUM	中太実線(4ドット)
LINESTYLE_THICK	太実線(8ドット)

㉝ 国際文字セット(InternationalCharacter)

国際文字セットで利用する列挙型定数を下記に示します。

定数名	説明
COUNTRY_USA	アメリカ(USA)
COUNTRY_FRANCE	フランス(France)
COUNTRY_GERMANY	ドイツ(Germany)
COUNTRY_ENGLAND	イギリス(United Kingdom)
COUNTRY_DENMARK_1	デンマークⅠ(Denmark I)
COUNTRY_SWEDEN	スウェーデン(Sweden)
COUNTRY_ITALY	イタリア(Italy)
COUNTRY_SPAIN	スペインⅠ(Spain I)
COUNTRY_JAPAN	日本(Japan)
COUNTRY_NORWAY	ノルウェー(Norway)
COUNTRY_DENMARK_2	デンマークⅡ(Denmark II)
COUNTRY_SPAIN_2	スペインⅡ(Spain II)
COUNTRY_LATIN_AMERICA	ラテンアメリカ(Latin America)
COUNTRY_ARABIA	アラビア(Arabia)

㉞ コードページ(CodePage)

コードページで利用する列挙型定数を下記に示します。

定数名	説明
CODE_PAGE_437	USA, Standard Europe (Code Page437)
CODE_PAGE_KATAKANA	Katakana
CODE_PAGE_850	Multilingual (Code Page850)
CODE_PAGE_860	Portuguese (Code Page860)
CODE_PAGE_863	Canadian-French (Code Page863)
CODE_PAGE_865	Nordic (Code Page865)
CODE_PAGE_857[*1*2]	Turkish (Code Page857)
CODE_PAGE_737[*2]	Greek (Code Page737)
CODE_PAGE_1252	Latin (Code Page1252)
CODE_PAGE_866[*2]	Russian (Code Page866)
CODE_PAGE_852	Eastern Europe (Code Page852)
CODE_PAGE_858	Euro (Code Page858)
CODE_PAGE_855[*2]	Cyrillic (Code Page855)
CODE_PAGE_864[*1*3]	Arabic (Code Page864)
CODE_PAGE_1250	Central European (Code Page1250)
CODE_PAGE_1251	Cyrillic (Code Page1251)
CODE_PAGE_1253[*4]	Greek (Code Page1253)
CODE_PAGE_1254	Turkish (Code Page1254)

*1: Unicodeの20AChは印字、または表示できません。
*2: RP-E10及びRP-D10はサポートしていません。
*3: フォントBは印字、または表示できません。
*4: Unicodeの00AAhは印字、または表示できません。

㉟ ラベルファイル印刷時マークあり・なし用紙選択(PrintSmartLabelMode)

ラベルファイル印刷時の用紙の選択で利用する定数を下記に示します。

定数名	説明
PRINTSMARTLABEL_MODE_MARK	マーク紙
PRINTSMARTLABEL_MODE_NONEMARK	マークなし用紙

4.4.2 PrinterManagerクラス

(1) メソッド一覧

PrinterManagerクラスで提供されるメソッドを下記に示します。
PrinterManagerクラスでは、「スタンダードモード」「ページモード」を選択できます。

メソッド	説明
スタンダードモード・ページモード共通メソッド	スタンダードモード及びページモードで有効なメソッドです。 メソッドについては、「4.4.2(1)① スタンダードモード・ページモード共通メソッド」を参照してください。
スタンダードモード専用メソッド	スタンダードモードで有効なメソッドです。 メソッドについては、「4.4.2(1)② スタンダードモード専用メソッド」を参照してください。
ページモード専用メソッド	ページモードで有効なメソッドです。 メソッドについては、「4.4.2(1)③ ページモード専用メソッド」を参照してください。

① スタンダードモード・ページモード共通メソッド

スタンダードモード・ページモード共通メソッドを下記に示します。共通メソッドの詳細については、「4.4.2(4)① スタンダードモード・ページモード共通メソッド」を参照してください。

✓:サポート、−:非サポート

名前	説明	SLP720RT SLP721RT	RP-F10 RP-G10	RP-E10 RP-D10	MP-B30 MP-B30L MP-B21L MP-B20 MP-A40
PrinterManager	コンストラクタ	✓	✓	✓	✓
Connect	デバイスとの通信開始	✓	✓	✓	✓
Disconnect	デバイスとの通信切断	✓	✓	✓	✓
OpenDrawer	キャッシュドロワを開く	✓*1	✓	✓	−
Buzzer	ブザーの鳴動	−	−	✓	−
ExternalBuzzer	外部ブザーの鳴動	✓*1	✓	−	−
GetStatus	プリンターステータスの取得	✓	✓	✓	✓
Abort	プリンターのデータ待ち状態解除	✓	✓	✓	✓
RegisterLogo	ロゴの登録	✓	✓	✓	✓
UnregisterLogo	ロゴの登録削除	✓	✓	✓	✓
RegisterStyleSheet	スタイルシートの登録	−	−	✓	✓*2
UnregisterStyleSheet	スタイルシートの登録削除	−	−	✓	✓*2
ResetPrinter	プリンターのリセット	✓	✓	✓	✓
GetPrinterResponse	プリンターからの各種応答取得	✓	✓	✓	✓
StartDiscoveryDevice	デバイスの探索開始	✓	✓	✓	✓
CancelDiscoveryDevice	デバイスの探索中断	✓	✓	✓	✓

名前	説明	SLP720RT SLP721RT	RP-F10 RP-G10	RP-E10 RP-D10	MP-B30 MP-B30L MP-B21L MP-B20 MP-A40
GetFoundDevice	発見されたデバイス情報リストの取得	✓	✓	✓	✓
ControlTransaction	一括処理の開始・終了	✓	✓	✓	✓
SetStatusChangedEventHandler	プリンターステータス変化のコールバック開始・終了	✓	✓	✓	✓
SetBarcodeScannerReadDataEventHandler	バーコードデータ受信のコールバック開始・終了	–	✓*3	–	–
SetBarcodeScannerChangedOnlineEventHandler	バーコードスキャナー接続のコールバック開始・終了	–	✓*3	–	–
SetBarcodeScannerChangedOfflineEventHandler	バーコードスキャナー切断のコールバック開始・終了	–	✓*3	–	–

*1: SLP721RTのみサポートします。
*2: MP-A40のみサポートします。
*3: RP-F10のみサポートします。

② スタンダードモード専用メソッド

スタンダードモード専用メソッドを下記に示します。専用メソッドの詳細については、「4.4.2(4)② スタンダードモード専用メソッド」を参照してください。

✓:サポート、-:非サポート

名前	説明	SLP720RT SLP721RT	RP-F10 RP-G10	RP-E10 RP-D10	MP-B30 MP-B30L MP-B21L MP-B20 MP-A40
SendText	テキストデータの送信	✓	✓	✓	✓
SendTextEx	書式指定テキストデータの送信	✓	✓	✓	✓
PrintBarcode	バーコードの印字	✓	✓	✓	✓
PrintPDF417	PDF417の印字	✓	✓	✓	✓
PrintQRcode	QRコードの印字	✓	✓	✓	✓
PrintDataMatrix	Data Matrixの印字	✓	✓	✓	✓
PrintMaxiCode	MaxiCodeの印字	✓	✓	✓	✓
PrintGS1DataBarStacked	GS1 Databar Stackedの印字	✓	✓	–	✓
PrintGS1DataBarStackedOmnidirectional	GS1 Databar Stacked Omni-directionalの印字	✓	✓	–	✓
PrintGS1DataBarExpandedStacked	GS1 Databar Expanded Stackedの印字	✓	✓	–	✓
PrintAztecCode	Aztec Codeの印字	–	–	–	✓*1
CutPaper	用紙のカット	✓	✓	✓	✓*2
FeedPosition	用紙の頭出し	✓	–	–	✓*3
SendBinary	バイナリデータの送信	✓	✓	✓	✓
SendDataFile	指定ファイルの送信	✓	✓	✓	✓
SendImage	イメージの送信	✓	✓	✓	✓
PrintPDF	PDFページの印字	✓	✓	✓	✓

名前	説明	SLP720RT SLP721RT	RP-F10 RP-G10	RP-E10 RP-D10	MP-B30 MP-B30L MP-B21L MP-B20 MP-A40
PrintLogo	ロゴの印字	✓	✓	✓	✓
PrintSmartLabelImageData	ラベルの印字	✓	–	–	–

*1: MP-B30L及びMP-B21Lのみサポートします。
*2: 用紙カット位置までの紙送り動作のみ行います。
*3: MP-B30L、MP-B21L及びMP-A40のみサポートします。

③ ページモード専用メソッド

ページモード専用メソッドを下記に示します。専用メソッドの詳細については、「4.4.2(4)③ ページモード専用メソッド」を参照してください。

✓:サポート、–:非サポート

名前	説明	SLP720RT SLP721RT	RP-F10 RP-G10	RP-E10 RP-D10	MP-B30 MP-B30L MP-B21L MP-A40	MP-B20
EnterPageMode	ページモードの開始	–	✓	✓	✓	–
ExitPageMode	ページモードの終了	–	✓	✓	✓	–
SetPageModeArea	ページモードの印字領域の指定	–	✓	✓	✓	–
SetPageModeDirection	ページモードの印字方向の指定	–	✓	✓	✓	–
SetPageModeLineSpacing	ページモードの改行量の指定	–	✓	✓	✓	–
PrintPageMode	ページモードの印字	–	✓	✓	✓	–
PrintPageModeText	ページモードのテキストデータの送信	–	✓	✓	✓	–
PrintPageModeTextEx	ページモードの書式指定テキストデータの送信	–	✓	✓	✓	–
PrintPageModeBarcode	ページモードのバーコードの印字	–	✓	✓	✓	–
PrintPageModePDF417	ページモードのPDF417の印字	–	✓	✓	✓	–
PrintPageModeQRcode	ページモードのQRコードの印字	–	✓	✓	✓	–
PrintPageModeDataMatrix	ページモードのData Matrixの印字	–	✓	✓	✓	–
PrintPageModeMaxiCode	ページモードのMaxiCodeの印字	–	✓	✓	✓	–
PrintPageModeGS1DataBar Stacked	ページモードのGS1 Databar Stackedの印字	–	✓	✓	✓	–
PrintPageModeGS1DataBar StackedOmnidirectional	ページモードのGS1 Databar Stacked Omni-directionalの印字	–	✓	✓	✓	–
PrintPageModeGS1DataBar ExpandedStacked	ページモードのGS1 Databar Expanded Stackedの印字	–	✓	✓	✓	–
PrintPageModeAztecCode	ページモードのAztec Codeの印字	–	–	–	✓*1	–
SendPageModeBinary	ページモードのバイナリデータの送信	–	✓	✓	✓	–
PrintPageModeImageFile	ページモードのイメージファイルの描画	–	✓	✓	✓	–
PrintPageModeRectangle	ページモードの矩形の描画	–	✓	✓	✓	–
PrintPageModeLine	ページモードの罫線の印字	–	✓	✓	✓	–
PrintPageModeLogo	ページモードのロゴの印字	–	✓	✓	✓	–

*1: MP-B30L及びMP-B21Lのみサポートします。

(2) スタンダードモード・ページモード共通プロパティ一覧

PrinterManagerクラスで提供されるプロパティを下記に示します。

✓:サポート、−:非サポート

名前	アクセス	説明	SLP720RT SLP721RT	RP-F10 RP-G10	RP-E10 RP-D10	MP-B30 MP-B30L MP-B21L MP-B20 MP-A40
SendTimeout	R/W	送信タイムアウト時間の設定/取得	✓	✓	✓	✓
ReceiveTimeout	R/W	受信タイムアウト時間の設定/取得	✓	✓	✓	✓
InternationalCharacter	R/W	国際文字セットの設定/取得	✓	✓	✓	✓
CodePage	R/W	コードページの設定/取得	✓	✓	✓	✓
DeviceModel	R	デバイスモデルの取得	✓	✓	✓	✓
PortType	R	接続中ポート種別の取得	✓	✓	✓	✓
IsConnect	R	プリンターとの通信状態の確認	✓	✓	✓	✓
SocketKeepingTime	R/W	ソケット維持時間の設定/取得	✓	✓	✓	✓*1
Version	R	SDKバージョンの取得	✓	✓	✓	✓
PrintSmartLabelMode	R/W	ラベルファイル印刷時の用紙の設定/取得	✓	−	−	✓*2

*1: MP-B20はサポートしていません。
*2: MP-B30L及びMP-B21Lのみサポートします。

(3) 定数一覧

① バーコード、PDF417専用

バーコードの印字と、PDF417の印字で利用する定数を下記に示します。

定数名	説明	値
BARCODE_HEIGHT_DEFAULT	バーコード高さの初期値	162
PDF417_MODULE_HEIGHT_DEFAULT	PDF417高さの初期値	10
PDF417_ROW_AUTO	行数自動選択	0
PDF417_COLUMN_AUTO	カラム数自動選択	0

② バッテリ残量レベル

プリンターから取得するバッテリ残量レベルの定数を下記に示します。

定数名	説明	値
BATTERY_STATUS_FULL	バッテリ残量目安:約80%	0
BATTERY_STATUS_MIDDLE	バッテリ残量目安:約40%	1
BATTERY_STATUS_LOW	バッテリ残量目安:約10%	2
BATTERY_STATUS_EMPTY	バッテリなし	3
BATTERY_STATUS_AC_POWERED	AC給電	4

(4) メソッド詳細

① スタンダードモード・ページモード共通メソッド

スタンダードモード及びページモードで有効なメソッドです。Connect直後はスタンダードモードになります。

PrinterManager　コンストラクタ

SII.SDK.Maui.Printer.PrinterManagerクラスのコンストラクタです。

対象　プラットフォーム共通　全デバイス

構文
```
public PrinterManager(object context);
```

パラメータ　context　コンテキスト
iOSの場合：
nullを指定してください。
Androidの場合：
本メソッドを呼び出すアプリケーションのコンテキストを指定してください。

Connect　デバイスとの通信開始

デバイスとの通信を開始します。

対象　プラットフォーム共通　全デバイス

構文
(a)
```
public void Connect(DeviceModel deviceModel,
                    DeviceInfo deviceInfo);
```
(b)
```
public void Connect(DeviceModel deviceModel,
                    PortType portType,
                    string param);
```

パラメータ　deviceModel　デバイスモデル
利用可能な定数は、「4.4.1① デバイスモデル（DeviceModel）」を参照してください。

deviceInfo　デバイス情報
DeviceInfoクラスのオブジェクトを指定してください。
DeviceInfoクラスのオブジェクトは、StartDiscoveryDeviceとGetFoundDeviceにて取得が可能です。

portType　ポート種別
利用可能な定数は、「4.4.1② ポート種別（PortType）」を参照してください。

param　　portTypeの設定により異なります。
対応するインターフェイスは「はじめに」の「対象デバイス」を参照してください。
・portTypeがPORT_TYPE_BLUETOOTH
Bluetoothアドレス、またはBluetoothデバイス名（Bluetoothアクセサリ）を指定してください。
例：″00:11:22:AA:BB:CC″、″RP-F10″
・portTypeがPORT_TYPE_USB
iOSの場合：
プリンタ名を指定してください。
例：″RP-F10″
Androidの場合：
nullを指定してください。
・portTypeがPORT_TYPE_TCP
プリンターのIPアドレスを指定してください。
例：″192.168.0.190″、″192.168.0.1″

エラー　　本メソッド呼び出し時にエラーが発生した場合は、PrinterExceptionをスローします。
エラーについての詳細は、「**4.4.5 PrinterExceptionクラス**」を参照してください。

説明　　本メソッドは、他のPrinterManagerクラスのメソッドを使用する前に呼び出してください。
本ライブラリを正しく動作させるために、本メソッドは接続時にプリンターの設定を変更する場合があります。

Bluetooth接続の場合：
iOSでは、iOSデバイスとペアリング済みのプリンターと、Bluetooth接続で通信を開始します。
paramで指定したペアリング済みのBluetoothアドレス、またはBluetoothデバイス（Bluetoothアクセサリ）に対して接続を行います。
Androidでは、paramで指定したBluetoothアドレスに対し、deviceModelで指定したプリンターを接続します。

USB接続の場合：
iOSでは、iOSデバイスとUSBケーブルで接続されたプリンターと、USB接続で通信を開始します。
Androidでは、deviceModelで指定したプリンターに接続します。

TCP/IP接続の場合：
デバイスと同じネットワークに接続されたプリンターと、TCP/IP接続で通信を開始します。paramで指定されたIPアドレスに対して接続を行います。通信にはTCPポート9100番、及び26100番を使用します。

• ライブラリの TCP/IP 接続中のソケットの作成・破棄について

ライブラリはConnect後、Disconnectするまで作成したソケットを維持します。また、Disconnectするまで別のアプリケーションから同じプリンターに接続することはできません。

プリンターへのデータ送信完了時を基準とし、SocketKeepingTimeで設定するソケット維持時間経過後、使用しているソケットをいったん破棄します。その後すぐに新規のソケットを作成し、次の接続に使用します。

ソケット破棄の時点でプリンターが同じネットワーク上の別のホストから接続要求を受けていた場合、プリンターはそちらのホストとの通信を確立するため、再接続に失敗する可能性があります。

注意　　複数のAppから1台のプリンターへの同時接続をサポートしていません。

Disconnect　　デバイスとの通信切断

デバイスとの通信を終了します。

対象　　プラットフォーム共通　　全デバイス

構文　　public void Disconnect();

エラー　　本メソッド呼び出し時にエラーが発生した場合は、PrinterExceptionをスローします。
エラーについての詳細は、「**4.4.5 PrinterExceptionクラス**」を参照してください。

注意　　本メソッドの直前に、GetPrinterResponseのPRINTER_RESPONSE_REQUEST(実行応答リクエスト)により実行応答を取得することを推奨します。取得しない場合は、下記の問題が発生する場合があります。

- デバイスの印字データが全てプリンターに送信される前に通信が切断され、印字データの一部が消失する場合があります。
- Bluetooth接続では、プリンターがバッファフルの状態*1でDisconnectとConnectを実行すると、デバイスとプリンターとの通信が切断される場合があります。

  *1: バッファフルの状態とは、印字データによりプリンターのバッファーが一杯になっている状態を指します。
  バッファフルの状態となるサイズは、約4KBです。

GetPrinterResponseを実行しない場合は、お客様のプログラムにおいて、問題が無いことを十分評価してからご使用ください。

OpenDrawer　　キャッシュドロワを開く

指定されたキャッシュドロワをオープンします。

対象　　プラットフォーム共通　　SLP721RT、RP-F10、RP-G10、RP-E10、RP-D10

構文　　public void OpenDrawer(DrawerNum drawerNum, PulseWidth onOffTime);

パラメータ　　drawerNum　　ドロワ番号
利用可能な定数は、「4.4.1③ ドロワ番号(DrawerNum)」を参照してください。

onOffTime　　パルス幅
利用可能な定数は、「4.4.1④ パルス幅(PulseWidth)」を参照してください。

エラー　　本メソッド呼び出し時にエラーが発生した場合は、PrinterExceptionをスローします。
エラーについての詳細は、「**4.4.5 PrinterExceptionクラス**」を参照してください。

Buzzer　ブザーの鳴動

ブザーを鳴動します。

対象　プラットフォーム共通　RP-E10、RP-D10

構文
```
public void Buzzer(int onTime, int offTime);
```

パラメータ

onTime	ブザーオン時間(ミリ秒:ms) プザーをオンする時間を設定します。 有効範囲は、0～510です。
offTime	ブザーオフ時間(ミリ秒:ms) プザーをオフする時間を設定します。 有効範囲は、0～510です。

エラー　本メソッド呼び出し時にエラーが発生した場合は、PrinterExceptionをスローします。
エラーについての詳細は、「**4.4.5 PrinterExceptionクラス**」を参照してください。

ExternalBuzzer　外部ブザーの鳴動

外部ブザーを鳴動します。

対象　プラットフォーム共通　SLP721RT、RP-F10、RP-G10

構文
```
public void ExternalBuzzer(BuzzerPattern buzzerPattern,
                           int buzzerCount);
```

パラメータ

buzzerPattern	ブザーパターン 利用可能な定数は、「4.4.1⑤ ブザーパターン(BuzzerPattern)」を参照してください。 外部ブザーの鳴動は下記のいずれかの条件で停止します。 ・ buzzerCountで設定したブザーの回数分の鳴動 ・ カバーオープン ・ プリンターコマンド「外付けブザーの停止」の実行
buzzerCount	鳴動回数(回) buzzerCountで設定した回数だけ外部ブザーを鳴動します。 有効範囲は、1～255です。

エラー　本メソッド呼び出し時にエラーが発生した場合は、PrinterExceptionをスローします。
エラーについての詳細は、「**4.4.5 PrinterExceptionクラス**」を参照してください。

GetStatus　　プリンターステータスの取得

最新のプリンターステータスを取得します。

対象　　プラットフォーム共通　　全プリンター

構文　　public void GetStatus(out int[] buf);

パラメータ　buf　　プリンターから取得したステータス

エラー　　本メソッド呼び出し時にエラーが発生した場合は、PrinterExceptionをスローします。
エラーについての詳細は、「**4.4.5 PrinterExceptionクラス**」を参照してください。

説明　　プリンターから取得したステータスは、int型の配列に格納されます。

プリンターステータスを下記に示します。
通信失敗時は、0x80000000を示します。

ビット	機能	値	
		0	1
0	電圧エラー*1	エラーなし	エラーあり
1	ハードウェアエラー*1	エラーなし	エラーあり
2	ヘッド温度エラー*1	エラーなし	エラーあり
3	オートカッタエラー*2*3*4	エラーなし*12	エラーあり
4	紙無しエラー*1	エラーなし	エラーあり
5	ニアエンドセンサエラー*3	エラーなし*12	エラーあり
6	マーク検出時紙ジャムエラー*3*5*8*9*11 マーク/ギャップ検出時紙ジャムエラー*6*10	エラーなし*12	エラーあり
7	カバーオープンエラー*2*3*4*5*6*8*9*10*11	エラーなし*12	エラーあり
8	フィードスイッチの状態*1	OFF	ON
9	予約済み	固定	ー
10	紙送り状態*1	停止	動作中
11	復帰待ち状態*1	復帰待ち状態でない	復帰待ち状態である
12	予約済み	固定	ー
13	テイクンセンサの状態*8*9	紙取り出し済み	紙取り出し待ち
13	予約済み	ー	固定
14	予約済み	ー	固定
15	ドロワスイッチの入力状態*2*3*4*9	Low*13	High*14
16	フラッシュメモリ書き換え中*1	書き換え中でない	書き換え中である
17	周辺機器選択*2*3*4	プリンター	その他
17	予約済み	固定*13	固定*14
18	予約済み	固定*15	固定*16
19	予約済み	ー	固定

ビット	機能	値	
		0	1
20～22*14	バッテリ残量レベル*5*6*7*10	000 : バッテリなし 001 : Low(残量目安:約10%) 011 : Middle(残量目安:約40%) 111 : Full(残量目安:約80%)	
20～22*14	バッテリ残量レベル*11	000 : バッテリなし 001 : レベル4、5(要充電) 010 : レベル3(残量目安:約10%) 011 : レベル2(残量目安:約40%) 100 : レベル1(残量目安:約80%)	
23	バッテリエラー*5*6*7*10*11	エラーなし	エラーあり*14
24～31	予約済み	ー	固定

*1: 全プリンターがサポートします。
*2: RP-F10、RP-G10がサポートします。
*3: RP-E10がサポートします。
*4: RP-D10がサポートします。
*5: MP-B30がサポートします。
*6: MP-B30Lがサポートします。
*7: MP-B20がサポートします。
*8: SLP720RTがサポートします。
*9: SLP721RTがサポートします。
*10: MP-B21Lがサポートします。
*11: MP-A40がサポートします。
*12: [機能]をサポートしないプリンターの[値]は0に固定されます。
*13: SLP720RTの[値]は0に固定されます。
*14: [機能]をサポートしないプリンターの[値]は1に固定されます。
*15: SLP720RT、SLP721RT、RP-F10、RP-G10の[値]は0に固定されます。
*16: RP-E10、RP-D10、MP-B30、MP-B30L、MP-B21L、MP-B20、MP-A40の[値]は1に固定されます。

Abort　プリンターのデータ待ち状態解除

プリンターのデータ待ち状態を解除します。

対象　プラットフォーム共通　全プリンター

構文
```
public void Abort();
```

エラー　本メソッド呼び出し時にエラーが発生した場合は、PrinterExceptionをスローします。
エラーについての詳細は、「**4.4.5 PrinterExceptionクラス**」を参照してください。

説明　SendDataFileによるイメージファイルの送信が中断した場合、プリンターは指定したイメージファイルの受信完了まで、他の処理を受け付けません(メソッドや送信データが誤解釈され、イメージファイルの続きと認識されます)。この状況を解消するために、本メソッドを利用し、プリンターのデータ待ち状態を解除します。
本メソッドを実行した場合、未印字のイメージファイルの一部が印字されることがあります。

RegisterLogo　ロゴの登録

イメージファイルをロゴとしてプリンターのNVグラフィックスメモリに登録します。

対象　プラットフォーム共通　全プリンター

構文
```
public void RegisterLogo(string fileName,
                         string logoId,
                         Dithering dithering);
```

パラメータ　fileName　　ロゴとして登録するイメージファイルのファイルパス
入力可能な形式について下記に説明します。

iOSの場合：
使用するファイルパス文字列

Androidの場合：

- Javaの標準的なクラス「java.io.File」が扱う絶対パス文字列
  Android 10 (API 29)以降をアプリのターゲットとする場合には、一部のファイルが直接扱えなくなりますのでご注意ください。
  詳細については、「3.4 注意事項 – 対象範囲別ストレージについて」を参照してください。

- Androidに用意されたクラス「android.net.Uri」が扱う下記スキーム名のURI文字列
  ・file://
  ・content://
  本パラメータには"ストレージ アクセス フレームワーク"から取得したURI文字列を指定する必要があります。"ストレージ アクセス フレームワーク"から取得せずに作成されたURIはファイルを開けない可能性がありますのでご注意ください。

サポートするイメージファイルのファイル拡張子は、.bmp、.jpg、.jpeg、.pngです。カラーイメージの場合は、2値化処理によりモノクロイメージに変換して登録します。

logoId　　登録するロゴのID（キーコード）
登録するロゴのIDを、2文字の文字列で指定してください。
有効な文字は、英数字（'0'～'9'、'A'～'Z'、'a'～'z'）などのASCII文字コード20h（スペース）～7Eh（チルダ）の文字です。

dithering　　ディザリング
利用可能な定数は、「4.4.1⑥ ディザリング（Dithering）」を参照してください。

エラー　本メソッド呼び出し時にエラーが発生した場合は、PrinterExceptionをスローします。
エラーについての詳細は、「**4.4.5 PrinterExceptionクラス**」を参照してください。

UnregisterLogo　ロゴの登録削除

登録済みのロゴを削除します。

対象　プラットフォーム共通　全プリンター

構文　`public void UnregisterLogo(string logoId);`

パラメータ　logoId　　削除するロゴのID（キーコード）
登録済みのロゴのIDを文字列で指定してください。

エラー　本メソッド呼び出し時にエラーが発生した場合は、PrinterExceptionをスローします。
エラーについての詳細は、「**4.4.5 PrinterExceptionクラス**」を参照してください。

RegisterStyleSheet　スタイルシートの登録

CSSファイルをスタイルシートとしてプリンターに登録します。

対象　プラットフォーム共通　RP-E10、RP-D10、MP-A40

構文　`public void RegisterStyleSheet(string fileName, int cssId);`

パラメータ　`fileName`　スタイルシートとして登録するCSSファイルパス
入力可能な形式について下記に説明します。

iOSの場合：
使用するファイルパス文字列

Androidの場合：

- Javaの標準的なクラス「java.io.File」が扱う絶対パス文字列
  Android 10 (API 29)以降をアプリのターゲットとする場合には、一部のファイルが直接扱えなくなりますのでご注意ください。
  詳細については、「3.4 注意事項 - 対象範囲別ストレージについて」を参照してください。

- Androidに用意されたクラス「android.net.Uri」が扱う下記スキーム名のURI文字列
  ・file://
  ・content://
  本パラメータには"ストレージ アクセス フレームワーク"から取得したURI文字列を指定する必要があります。"ストレージ アクセス フレームワーク"から取得せずに作成されたURIはファイルを開けない可能性がありますのでご注意ください。

サポートするスタイルシートのファイル拡張子は、.cssです。

`cssId`　登録するスタイルシートのID
登録するスタイルシートのIDを設定します。
有効範囲は、1～4です。

エラー　本メソッド呼び出し時にエラーが発生した場合は、PrinterExceptionをスローします。
エラーについての詳細は、「**4.4.5 PrinterExceptionクラス**」を参照してください。

説明　スタイルシートは、言語をCSS（cascading style sheets）で記述してください。
登録可能なスタイルシートの数は4シートです。
1つのCSSファイルに登録可能なスタイル数は、最大64です。

UnregisterStyleSheet　スタイルシートの登録削除

登録済みのスタイルシートを削除します。

対象　プラットフォーム共通　RP-E10、RP-D10、MP-A40

構文　`public void UnregisterStyleSheet(int cssId);`

パラメータ　`cssId`　削除するスタイルシートのID
登録済みのスタイルシートのIDを指定してください。

エラー　本メソッド呼び出し時にエラーが発生した場合は、PrinterExceptionをスローします。
エラーについての詳細は、「**4.4.5 PrinterExceptionクラス**」を参照してください。

ResetPrinter　プリンターのリセット

プリンターをハードウェアリセットします。

対象　プラットフォーム共通　全プリンター

構文　`public void ResetPrinter();`

エラー　本メソッド呼び出し時にエラーが発生した場合は、PrinterExceptionをスローします。
エラーについての詳細は、「**4.4.5 PrinterExceptionクラス**」を参照してください。

説明　Bluetooth接続の場合：
プリンターコマンド「プリンタリセット」によりプリンターをハードウェアリセットします。
iOS（RP-F10、RP-E10、RP-D10）の場合は、本メソッドを利用する前にプリンターのメモリスイッチ「iOS自動接続選択」を有効に設定してください。
iOS（MP-B30、MP-B30L、MP-B21L、MP-B20、MP-A40）の場合は、本メソッドを利用する前にプリンターコマンド「Bluetooth通信の設定」でiOS自動接続選択を有効に設定してください。
無効の場合はリセット後の再接続に失敗し、PrinterExceptionをスローします。
本メソッドはリセットを実行後、プリンターとの再接続が完了するまでに約10秒かかります。充分な受信タイムアウト時間を設定した上で本メソッドを使用してください。

USB接続の場合：
iOSの場合は、プリンターコマンド「プリンタリセット」によりプリンターをハードウェアリセットします。
Androidの場合は、USBプリンタークラスのSOFT_RESET機能によりプリンターをリセットします。

TCP/IP接続の場合：
接続中のプリンターに対して、TCPポート26100番への弊社独自コマンド（リセット要求）によりプリンターをリセットします。

本メソッド実行後も、プリンターとの接続は保持されます。

GetPrinterResponse　　プリンターからの各種応答取得

プリンターからの応答データを取得します。

対象　　プラットフォーム共通　　全プリンター

構文

```
public void GetPrinterResponse<T>(PrinterResponseId responseId,
                                  object param,
                                  out T[] response);
```

パラメータ

	responseId	プリンター応答種別定数 利用可能な定数は、 「4.4.1⑦ プリンター応答種別（PrinterResponseId）」を参照してください。
	param	コマンドパラメータ プリンター応答種別定数により指定する値が異なります。 指定する値の説明は、下表を参照してください。
	response	取得した応答データを格納するバッファー プリンター応答種別定数によりバッファーの型が異なります。 バッファーの型は、下表を参照してください。

応答種別定数

パラメータ	説明
PRINTER_RESPONSE_REQUEST（実行応答リクエスト）	
param	int型で0～15（00h～0Fh）を指定してください。
response	int型の配列を指定してください。 応答取得成功時には、実行応答リクエストの応答コードが128～143（80～8Fh）で格納されます。
PRINTER_RESPONSE_USER_AREA（ユーザ領域の残り容量の送信）	
param	nullを指定してください。
response	int型の配列を指定してください。 応答取得成功時には、ユーザ領域の残り容量（単位：バイト）が数値で格納されます。
PRINTER_RESPONSE_ARRANGE_USER_AREA（ユーザ領域の整理後の残り容量の送信）	
param	nullを指定してください。
response	int型の配列を指定してください。 応答取得成功時には、ユーザ領域の整理後の残り容量（単位：バイト）が数値で格納されます。
PRINTER_RESPONSE_NV_GRAPHICS（NVグラフィックスのメモリ容量の送信）	
param	nullを指定してください。
response	int型の配列を指定してください。 応答取得成功時には、NVグラフィックスメモリ容量（単位：バイト）が数値で格納されます。

応答種別定数		
	パラメータ	説明
PRINTER_RESPONSE_KEY_CODE（定義されているNVグラフィックスのキーコード一覧の送信）		
	param	nullを指定してください。
	response	string型の配列を指定してください。 応答取得成功時には、NVグラフィックスのキーコードが文字列の配列で格納されます。
PRINTER_RESPONSE_BATTERY_STATUS（バッテリ残量レベル）		
	param	nullを指定してください。
	response	MP-B30、MP-B30L、MP-B21L、MP-B20及びMP-A40のみサポートします。 int型の配列を指定してください。 応答取得成功時には、バッテリ残量レベルが値で格納されます。 値については、「4.4.2(3)② バッテリ残量レベル」を参照してください。 バッテリ残量レベル **BATTERY_STATUS_FULL:** Full(残量目安:約80%) **BATTERY_STATUS_MIDDLE:** Middle(残量目安:約40%) **BATTERY_STATUS_LOW:** Low(残量目安:約10%) **BATTERY_STATUS_EMPTY:** バッテリなし **BATTERY_STATUS_AC_POWERED:** AC給電*1
PRINTER_RESPONSE_MAINTENANCE_COUNTER（メンテナンスカウンタの送信）		
	param	nullを指定してください。
	response	int型の配列を指定してください。 応答取得成功時は、下記のように格納されます。 buf[0] :紙送り行数(単位:100ドットライン) buf[1] :サーマルヘッド通電回数(単位:100ドットライン) buf[2] :オートカッタ駆動回数*2 buf[3] :製品の駆動時間(単位:分) buf[4] :紙送り行数(積算)(単位:100ドットライン) buf[5] :サーマルヘッド通電回数(積算)(単位:100ドットライン) buf[6] :オートカッタ駆動回数(積算)*2 buf[7] :製品の駆動時間(積算)(単位:分)
PRINTER_RESPONSE_PRINTER_IDENTIFIER（プリンタIDの送信）		
	param	nullを指定してください。
	response	応答取得成功時には、bufの先頭要素にPrinterIdentifierクラスのオブジェクトが格納されます。 例:buf.size() = 1、 buf.get(0) = ″(PrinterIdentifierクラスのオブジェクト)″ PrinterIdentifierクラスのオブジェクトについては、 「4.4.4 PrinterIdentifierクラス」を参照してください。

*1: MP-A40のみサポートします。
*2: MP-B30、MP-B30L、MP-B21L、MP-B20、MP-A40はサポートしていません。常に0を返します。

エラー　本メソッド呼び出し時にエラーが発生した場合は、PrinterExceptionをスローします。
エラーについての詳細は、「**4.4.5 PrinterExceptionクラス**」を参照してください。

StartDiscoveryDevice　　デバイスの探索開始

デバイスを探索します。

対象　　プラットフォーム共通　　全デバイス

構文

```
public void StartDiscoveryDevice(DeviceType deviceType,
                                 PortType portType,
                                 int timeout,
                                 Action completion);
```

パラメータ

deviceType　　デバイスタイプ
利用可能な定数は、「4.4.1⑧ デバイスタイプ（DeviceType）」を参照してください。
iOSの場合は、TYPE_PRINTERのみ有効です。

portType　　ポート種別
利用可能な定数は、「4.4.1② ポート種別（PortType）」を参照してください。

timeout　　探索タイムアウト時間（ミリ秒：ms）
タイムアウト時間を設定します。
有効範囲は、3000～60000です。
3000未満の値を指定した場合は3000msに設定されます。
60000を超える値を指定した場合は60000msに設定されます。

- **PORT_TYPE_TCP**の場合
  ローカルブロードキャストパケットを送信し、timeoutで指定した時間が経過するまでプリンターからの応答を待ちます。
- **PORT_TYPE_BLUETOOTH**の場合
  デバイスを探索中の場合、timeoutで指定した時間が経過するまで探索を行います。デバイスの探索が終了した場合は、timeoutの時間を待たずに探索を終了します。
  iOSの場合は、timeoutの指定は無視されます。
- **PORT_TYPE_USB**の場合
  timeoutの指定は無視されます。

completion　　デバイス探索終了時に実行するブロック

エラー　　本メソッド呼び出し時にエラーが発生した場合は、PrinterExceptionをスローします。

説明　　本メソッドは、デバイスを探索します。
発見したデバイスのデバイス情報は、GetFoundDeviceで取得できます。

CancelDiscoveryDevice　デバイスの探索中断

実行中のStartDiscoveryDeviceを中断します。

対象　プラットフォーム共通　全プリンター

構文　public void CancelDiscoveryDevice();

説明　StartDiscoveryDeviceのcompletionで設定したブロックにイベントとして通知します。

GetFoundDevice　発見されたデバイス情報リストの取得

StartDiscoveryDeviceで発見されたデバイス情報リストを、格納先のDeviceInfoクラスから取得します。

対象　プラットフォーム共通　全プリンター

構文　public DeviceInfo[] GetFoundDevice();

説明　iOSでBluetooth接続の場合は、接続済みプリンターのデバイス情報のみ取得が可能です。

戻り値　「**4.4.3** DeviceInfoクラス」を参照してください。

ControlTransaction　一括処理の開始・終了

一括処理を開始、または終了します。

対象　プラットフォーム共通　全プリンター

構文　public void ControlTransaction(TransactionFunction control);

パラメータ　control　一括処理の選択
利用可能な定数は、「4.4.1⑨ 一括処理選択（TransactionFunction）」を参照してください。

エラー　本メソッド呼び出し時にエラーが発生した場合は、PrinterExceptionをスローします。
エラーについての詳細は、「**4.4.5 PrinterExceptionクラス**」を参照してください。

説明　一括処理の手順は下記の通りです。

① 一括処理を開始します。
TRANSACTION_STARTを指定してください。

② メソッドを実行します。
一括処理対象メソッドの場合、送信データのバッファリングを開始します。
バッファリング中に実行した一括処理対象メソッドの送信データは、プリンターへ送信せずに送信バッファーにバッファリングします。
バッファリングできる送信データの最大サイズはシステムに依存します。
バッファリングされた送信データが最大サイズを超える場合、超えた時点の一括対象メソッドがエラーとなります。エラーとなった場合は、エラー前までの送信データが保持されます。
保持された送信データは、手順③で一括処理を終了してください。
一括処理対象以外のメソッドの場合は、送信データはバッファリングされずに即実行されます。

③ 一括処理を終了します。
TRANSACTION_PRINTを指定するとバッファリングされた送信データをプリンターへ送信します。バッファリングされた送信データはプリンター送信後も保持されます。
保持されている送信データは下記のいずれかにより破棄されます。
・**TRANSACTION_CLEAR**の指定
・**TRANSACTION_START**の指定
・Disconnectの実行

一括処理対象メソッドを下記に示します。
・SendText
・SendTextEx
・PrintBarcode
・PrintPDF417
・PrintQRcode
・PrintDataMatrix
・PrintMaxiCode
・PrintGS1DataBarStacked
・PrintGS1DataBarStackedOmnidirectional
・PrintGS1DataBarExpandedStacked
・PrintAztecCode
・CutPaper
・FeedPosition
・OpenDrawer
・Buzzer
・ExternalBuzzer
・SendBinary
・SendDataFile
・SendImage
・PrintPDF
・PrintLogo*1
・PrintSmartLabelImageData
・EnterPageMode
・ExitPageMode
・SetPageModeArea
・SetPageModeDirection
・SetPageModeLineSpacing
・PrintPageMode
・PrintPageModeText
・PrintPageModeTextEx
・PrintPageModeBarcode
・PrintPageModePDF417
・PrintPageModeQRcode
・PrintPageModeDataMatrix
・PrintPageModeMaxiCode

・PrintPageModeGS1DataBarStacked
・PrintPageModeGS1DataBarStackedOmnidirectional
・PrintPageModeGS1DataBarExpandedStacked
・PrintPageModeAztecCode
・SendPageModeBinary
・PrintPageModeImageFile
・PrintPageModeRectangle
・PrintPageModeLine
・PrintPageModeLogo*1

*1: 一括処理中のメソッドは登録済みのロゴが存在しない場合でもエラーを通知しません。

SetStatusChangedEventHandler　プリンターステータス変化のコールバック開始・終了

プリンターステータス変化のコールバックを開始、または終了します。

対象　プラットフォーム共通　全プリンター

構文
```
public void SetStatusChangedEventHandler(
                                    Action<int> statusChangedEventHandler);
```

パラメータ　statusChangedEventHandler
プリンターステータス変化の通知を受け取るデリゲートオブジェクト

説明　本メソッドをstatusChangedEventHandlerにnull以外の有効なデリゲートオブジェクトを指定して実行することによりプリンターステータス変化の通知を開始します。

プリンターステータス変化はIsConnectがtrueの場合のみ通知されます。

最新のプリンターステータスは、下記のタイミングで通知されます。
・プリンターステータスが変化した場合
・本メソッドをstatusChangedEventHandlerにnull以外の有効なデリゲートオブジェクトを指定して実行した場合

プリンターステータス変化の通知は、下記のタイミングで終了します。
・Disconnectが実行された場合
・本メソッドをstatusChangedEventHandlerにnullを指定して実行した場合

プリンターとの通信が切断されると、0x80000000を通知します。ライブラリはプリンターとの通信切断後、Disconnectが実行されるまでプリンターとの通信の再開を試みます。プリンターとの通信が可能になると、最新のプリンターステータスを通知します。
プリンターステータスの内容は、GetStatusを参照してください。

statusChangedEventHandlerに指定するデリゲートオブジェクト内で、PrinterManagerクラスのAPIを実行しないでください。

SetBarcodeScannerReadDataEventHandler
バーコードデータ受信のコールバック開始・終了

バーコードデータ受信のコールバックを開始、または終了します。

対象　　プラットフォーム共通　　RP-F10

構文　　public void SetBarcodeScannerReadDataEventHandler(
　　Action<byte[]> barcodeScannerReadDataEventHandler);

パラメータ　barcodeScannerReadDataEventHandler
　　バーコードデータ受信の通知を受け取るデリゲートオブジェクト

説明　　barcodeScannerReadDataEventHandlerに指定したデリゲートオブジェクトは、プリンターに接続したバーコードスキャナーでバーコードをスキャンした時に呼び出されます。

バーコードデータの受信は、IsConnectがtrueでプリンターと通信中に通知されます。

バーコードデータ受信のコールバックは、下記のタイミングで終了します。
・Disconnectが実行された場合
・本メソッドをbarcodeScannerReadDataEventHandlerにnullを指定して実行した場合

barcodeScannerReadDataEventHandlerに指定するデリゲートオブジェクト内で
PrinterManagerクラスのAPIを実行しないでください。

SetBarcodeScannerChangedOnlineEventHandler
バーコードスキャナー接続のコールバック開始・終了

バーコードスキャナーの接続を通知します。

対象　　プラットフォーム共通　　RP-F10

構文　　public void SetBarcodeScannerChangedOnlineEventHandler(
　　Action barcodeScannerChangedOnlineEventHandler);

パラメータ　barcodeScannerChangedOnlineEventHandler
　　バーコードスキャナー接続の通知を受け取るデリゲートオブジェクト

説明　　本メソッドは、下記のタイミングで呼び出されます。
・バーコードスキャナーがプリンターと接続状態で本メソッドをbarcodeScannerChanged OnlineEventHandlerにnull以外の有効なデリゲートオブジェクトを指定して実行した場合
・プリンターにバーコードスキャナーが接続された場合

バーコードスキャナーの接続は、IsConnectがtrueでプリンターと通信中に通知されます。

バーコードスキャナー接続のコールバックは、下記のタイミングで終了します。
・Disconnectが実行された場合
・本メソッドをbarcodeScannerChangedOnlineEventHandlerにnullを指定して実行した場合

barcodeScannerChangedOnlineEventHandlerに指定するデリゲートオブジェクト内で
PrinterManagerクラスのAPIを実行しないでください。

SetBarcodeScannerChangedOfflineEventHandler
バーコードスキャナー切断のコールバック開始・終了

バーコードスキャナーの切断を通知します。

対象　　プラットフォーム共通　　RP-F10

構文　　public void SetBarcodeScannerChangedOfflineEventHandler(
　　Action barcodeScannerChangedOfflineEventHandler);

パラメータ　barcodeScannerChangedOfflineEventHandler
　　バーコードスキャナー切断の通知を受け取るデリゲートオブジェクト

説明　　本メソッドは、下記のタイミングで呼び出されます。
・バーコードスキャナーがプリンターと非接続状態で本メソッドをbarcodeScannerChanged OfflineEventHandlerにnull以外の有効なデリゲートオブジェクトを指定して実行した場合
・プリンターからバーコードスキャナーが切断された場合

バーコードスキャナーの切断は、IsConnectがtrueの場合に通知されます。

バーコードスキャナー切断のコールバックは、下記のタイミングで終了します。
・Disconnectが実行された場合
・本メソッドをbarcodeScannerChangedOfflineEventHandlerにnullを指定して実行した場合

barcodeScannerChangedOfflineEventHandlerに指定するデリゲートオブジェクト内で
PrinterManagerクラスのAPIを実行しないでください。

② スタンダードモード専用メソッド

スタンダードモードで有効なメソッドです。スタンダードモード専用メソッドをページモード中に実行すると、PrinterExceptionをスローします。

SendText　テキストデータの送信

テキストデータを送信します。

対象　プラットフォーム共通　全プリンター

構文
```
public void SendText(string text);
```

パラメータ　text　プリンターに送信するテキストデータ
一度に指定可能なデータサイズは16KB（16384バイト）です。

エラー　本メソッド呼び出し時にエラーが発生した場合は、PrinterExceptionをスローします。
エラーについての詳細は、「**4.4.5 PrinterExceptionクラス**」を参照してください。

説明　本メソッドは、指定されたテキストデータを、InternationalCharacter、及びCodePageを基にプリンターで印字可能なテキストデータにエンコードしてプリンターに送信します。

テキストデータの最後に改行コードは付加しません。最後まで印字する場合は、テキストデータの最後に改行コードを入れてください。

SendTextEx　書式指定テキストデータの送信

書式指定されたテキストデータをプリンターに送信します。

対象　プラットフォーム共通　全プリンター

構文
(a)
```
public void SendTextEx(string text,
                       CharacterBold bold,
                       CharacterUnderline underline,
                       CharacterReverse reverse,
                       CharacterFont font,
                       CharacterScale scale,
                       PrintAlignment alignment);
```

(b)
```
public void SendTextEx(string text,
                       CharacterBold bold,
                       CharacterUnderline underline,
                       CharacterReverse reverse,
                       CharacterInversion inversion,
                       CharacterFont font,
                       CharacterScale scale,
                       PrintAlignment alignment);
```

パラメータ　text　プリンターに送信するテキストデータ
一度に指定可能なデータサイズは16KB（16384バイト）です。

bold　強調文字
利用可能な定数は、「4.4.1⑪ 強調文字（CharacterBold）」を参照してください。

underline　アンダーライン
利用可能な定数は、「4.4.1⑫ アンダーライン（CharacterUnderline）」を参照してください。

reverse　白黒反転印字
利用可能な定数は、「4.4.1⑰ 白黒反転印字（CharacterReverse）」を参照してください。

inversion　倒立印字
利用可能な定数は、「4.4.1⑱ 倒立印字(CharacterInversion)」を参照してください。

font　文字フォント
利用可能な定数は、「4.4.1⑭ 文字フォント（CharacterFont）」を参照してください。

scale　文字倍率
利用可能な定数は、「4.4.1⑬ 文字倍率（CharacterScale）」を参照してください。

alignment　位置揃え
利用可能な定数は、「4.4.1⑩ 位置揃え（PrintAlignment）」を参照してください。

エラー　本メソッド呼び出し時にエラーが発生した場合は、PrinterExceptionをスローします。
エラーについての詳細は、「**4.4.5 PrinterExceptionクラス**」を参照してください。

説明　本メソッドは、指定されたテキストデータを、InternationalCharacter、及びCodePageを基にプリンターで印字可能なテキストデータにエンコードしてプリンターに送信します。

テキストデータの最後に改行コードは付加しません。最後まで印字する場合は、テキストデータの最後に改行コードを入れてください。

PrintBarcode　　バーコードの印字

バーコードを印字します。
構文(a)は、バーコードデータを文字列で指定できます。
構文(b)は、バーコードデータを文字列で指定でき、NW比を指定できます。
構文(c)は、バーコードデータをバイト配列で指定できます。
構文(d)は、カスタマバーコードを文字列で指定できます。

対象　　プラットフォーム共通　　構文(a)～構文(c) :全プリンター
　　構文(d) :MP-B21L

構文　　(a) public void PrintBarcode(BarcodeSymbol barcodeSymbol,
　　string text,
　　ModuleSize moduleSize,
　　int moduleHeight,
　　HriPosition hriPosition,
　　CharacterFont hriFont,
　　PrintAlignment alignment);

(b) public void PrintBarcode(BarcodeSymbol barcodeSymbol,
　　string text,
　　ModuleSize moduleSize,
　　int moduleHeight,
　　HriPosition hriPosition,
　　CharacterFont hriFont,
　　PrintAlignment alignment,
　　NwRatio nwRatio);

(c) public void PrintBarcode(BarcodeSymbol barcodeSymbol,
　　byte[] data,
　　ModuleSize moduleSize,
　　int moduleHeight,
　　HriPosition hriPosition,
　　CharacterFont hriFont,
　　PrintAlignment alignment);

(d) public void PrintBarcode(BarcodeSymbol barcodeSymbol,
　　string text,
　　ModuleSize moduleSize,
　　PrintAlignment alignment);

パラメータ　　barcodeSymbol　バーコードシンボル
　　利用可能な定数、及び対応する構文は、
　　「4.4.1⑲ バーコードシンボル(BarcodeSymbol)」を参照してください。

text(data)　　プリンターに送信するバーコードデータ

バーコードデータの入力条件は下記の通りです。

バーコード	データ数	入力可能データ 文字列（データ）	備考
UPC-A	11～12文字	'0' ～ '9'	
UPC-E	11～12文字	'0' ～ '9'	
EAN13 JAN13	12～13文字	'0' ～ '9'	
EAN8 JAN8	7～8文字	'0' ～ '9'	
CODE39	1～150文字	'0' ～ '9' 'A' ～ 'Z' ' '、'$'、'%'、'+'、'-'、'.'、'/'	スタートコード及びストップコード('*')は自動付加されます。
CODE93	1～150バイト	(0x00 ～ 0x2E)	末尾に0x2F以上のデータを入力してください。
CODE128	2～150バイト	(0x00 ～ 0x66)	CODE128コードセットのスタートコード(0x67～0x69)で入力する場合。 末尾に0x67以上のデータを入力してください。
		(0x00 ～ 0x7F)	CODE128特殊コードのスタートコード(″{A″, ″{B″, ″{C″)で開始する場合。
ITF	2～150文字 (ただし偶数個)	'0' ～ '9'	
CODABAR	1～150文字	'0' ～ '9' '$'、'+'、'-'、'.'、'/'、':'	先頭と末尾に'A'～'D'のいずれかを指定する必要があります。
EAN13 add-on JAN13 add-on	Add-on 2: 14～15文字 Add-on 5: 17～18文字	'0' ～ '9'	
カスタマバーコード	7～20文字	'0' ～ '9' 'A' ～ 'Z' '-'	スタートコード、チェックデジット、ストップコードは自動付加されます。 先頭から7文字は'0'～'9'のいずれかを指定する必要があります。'A'～'Z'は2文字として計算されます。
GS1 Databar Omni-directional	13文字	'0' ～ '9'	チェックデジットは自動計算されます。
GS1 Databar Truncated	13文字	'0' ～ '9'	チェックデジットは自動計算されます。
GS1 Databar Limited	13文字	'0' ～ '9'	チェックデジットは自動計算されます。
GS1 Databar Expanded	2～255文字	' ' ～ '″' '%' ～ '?' 'A' ～ 'Z' '_' 'a' ～ 'z' '{'	

moduleSize　　バーコードの幅
利用可能な定数は、「4.4.1⑮ モジュールサイズ（ModuleSize）」を参照してください。

moduleHeight　　バーコードの高さ(ドット)

- barcodeSymbolが下記の場合は、有効範囲は、1～255です。
  BARCODE_UPC_A
  BARCODE_UPC_E
  BARCODE_EAN13
  BARCODE_JAN13
  BARCODE_EAN8
  BARCODE_JAN8
  BARCODE_CODE39
  BARCODE_CODE93
  BARCODE_CODE128
  BARCODE_ITF
  BARCODE_CODABAR
  BARCODE_EAN13_ADDON
  BARCODE_JAN13_ADDON

- barcodeSymbolが下記の場合は、barcodeSymbolとmoduleSizeにより有効範囲が異なります。

barcodeSymbol	moduleSize	有効範囲
BARCODE_GS1_OMNI_DIRECTIONAL		
	BARCODE_MODULE_WIDTH_2	66 ～ 255
	BARCODE_MODULE_WIDTH_3	99 ～ 255
	BARCODE_MODULE_WIDTH_4	132 ～ 255
	BARCODE_MODULE_WIDTH_5	165 ～ 255
	BARCODE_MODULE_WIDTH_6	198 ～ 255
BARCODE_GS1_TRUNCATED		
	BARCODE_MODULE_WIDTH_2	26 ～ 255
	BARCODE_MODULE_WIDTH_3	39 ～ 255
	BARCODE_MODULE_WIDTH_4	52 ～ 255
	BARCODE_MODULE_WIDTH_5	65 ～ 255
	BARCODE_MODULE_WIDTH_6	78 ～ 255
BARCODE_GS1_LIMITED		
	BARCODE_MODULE_WIDTH_2	20 ～ 255
	BARCODE_MODULE_WIDTH_3	30 ～ 255
	BARCODE_MODULE_WIDTH_4	40 ～ 255
	BARCODE_MODULE_WIDTH_5	50 ～ 255
	BARCODE_MODULE_WIDTH_6	60 ～ 255

barcodeSymbol	
moduleSize	有効範囲
BARCODE_GS1_EXPANDED	
BARCODE_MODULE_WIDTH_2	68 ～ 255
BARCODE_MODULE_WIDTH_3	102 ～ 255
BARCODE_MODULE_WIDTH_4	136 ～ 255
BARCODE_MODULE_WIDTH_5	170 ～ 255
BARCODE_MODULE_WIDTH_6	204 ～ 255

hriPosition　HRI文字印字位置
利用可能な定数は、「4.4.1⑳ HRI文字印字位置(HriPosition)」を参照してください。

hriFont　HRI文字フォント
利用可能な定数は、「4.4.1⑭ 文字フォント(CharacterFont)」を参照してください。

alignment　位置揃え
利用可能な定数は、「4.4.1⑩ 位置揃え(PrintAlignment)」を参照してください。

nwRatio　NW比
利用可能な定数は、「4.4.1㉑ NW比(NwRatio)」を参照してください。
指定したnwRatioとmoduleSizeにより、太エレメントの幅が下表のように設定されます。

moduleSize	nwRatio		
	NWRATIO_1TO2	**NWRATIO_1TO2_5**	**NWRATIO_1TO3**
BARCODE_MODULE_WIDTH_2	0.500 mm (4ドット)	0.625 mm (5ドット)	0.750 mm (6ドット)
BARCODE_MODULE_WIDTH_3	0.750 mm (6ドット)	1.000 mm (8ドット)	1.125 mm (9ドット)
BARCODE_MODULE_WIDTH_4	1.000 mm (8ドット)	1.250 mm (10ドット)	1.500 mm (12ドット)
BARCODE_MODULE_WIDTH_5	1.250 mm (10ドット)	1.625 mm (13ドット)	1.875 mm (15ドット)
BARCODE_MODULE_WIDTH_6	1.500 mm (12ドット)	1.875 mm (15ドット)	2.250 mm (18ドット)

エラー　本メソッド呼び出し時にエラーが発生した場合は、PrinterExceptionをスローします。
エラーについての詳細は、「**4.4.5 PrinterExceptionクラス**」を参照してください。

注意　SLP720RT、SLP721RT、RP-F10、RP-G10、MP-B30、MP-B30L及びMP-B21Lは、クワイエットゾーンの確保を行いません。バーコードシンボルの規格に応じて、クワイエットゾーンの設定を行ってください。

参考　　バーコードサイズについての詳細は、「付録B バーコードサイズ一覧」を参照してください。

PrintPDF417　　PDF417の印字

PDF417を印字します。

対象　　プラットフォーム共通　　全プリンター

構文

```
public void PrintPDF417(string text,
                        ErrorCorrection errorCorrection,
                        int row,
                        int column,
                        ModuleSize moduleSize,
                        int moduleHeight,
                        PrintAlignment alignment,
                        Pdf417Symbol pdf417Symbol);
```

パラメータ

text　　プリンターに送信するバーコードデータ

errorCorrection
エラー訂正レベル
利用可能な定数は、「4.4.1⑯ エラー訂正レベル(ErrorCorrection)」を参照してください。

row　　段数(段)
有効範囲は、0、3～90です。
0を指定した場合は、段数が自動設定されます。

column　　データ領域のカラム数
有効範囲は、0～30です。
0を指定した場合は、データ領域のカラム数が自動設定されます。

moduleSize　　公称細エレメント幅
利用可能な定数は、「4.4.1⑮ モジュールサイズ(ModuleSize)」を参照してください。

moduleHeight　　段高さ(ドット)
有効範囲は、2～127です。
段高さの設定を小さくすると、バーコードスキャナーによっては読み取れない場合があります。通常の使用では、3以上を設定してください。

alignment　　位置揃え
利用可能な定数は、「4.4.1⑩ 位置揃え(PrintAlignment)」を参照してください。

pdf417Symbol　　PDF417シンボル
利用可能な定数は、「4.4.1㉒ PDF417シンボル(Pdf417Symbol)」を参照してください。

エラー　　本メソッド呼び出し時にエラーが発生した場合は、PrinterExceptionをスローします。
エラーについての詳細は、「**4.4.5 PrinterExceptionクラス**」を参照してください。

注意　　SLP720RT、SLP721RT、RP-F10、RP-G10、MP-B30、MP-B30L及びMP-B21Lは、クワイエットゾーンの確保を行いません。バーコードシンボルの規格に応じて、クワイエットゾーンの設定を行ってください。

参考　　バーコードサイズについての詳細は、「付録B バーコードサイズ一覧」を参照してください。

PrintQRcode　　QRコードの印字

QRコードを印字します。

対象　　プラットフォーム共通　　全プリンター

構文

```
public void PrintQRcode(string text,
                        ErrorCorrection errorCorrection,
                        ModuleSize moduleSize,
                        PrintAlignment alignment,
                        QrModel model);
```

パラメータ

text　　プリンターに送信するバーコードデータ
バージョンはtextで設定したデータバイト数に応じて自動設定されます。

errorCorrection
エラー訂正レベル
利用可能な定数は、「4.4.1⑯ エラー訂正レベル（ErrorCorrection）」を参照してください。

moduleSize　　モジュールサイズ
利用可能な定数は、「4.4.1⑮ モジュールサイズ（ModuleSize）」を参照してください。

alignment　　位置揃え
利用可能な定数は、「4.4.1⑩ 位置揃え（PrintAlignment）」を参照してください。

model　　QRコードモデル
利用可能な定数は、「4.4.1㉓　QRコードモデル（QrModel）」を参照してください。

エラー　　本メソッド呼び出し時にエラーが発生した場合は、PrinterExceptionをスローします。
エラーについての詳細は、「**4.4.5 PrinterExceptionクラス**」を参照してください。

注意　　SLP720RT、SLP721RT、RP-F10、RP-G10、MP-B30、MP-B30L及びMP-B21Lは、クワイエットゾーンの確保を行いません。バーコードシンボルの規格に応じて、クワイエットゾーンの設定を行ってください。

参考　　バーコードサイズについての詳細は、「付録B バーコードサイズ一覧」を参照してください。

PrintDataMatrix　Data Matrixの印字

Data Matrixを印字します。

対象　プラットフォーム共通　全プリンター

構文
```
public void PrintDataMatrix(string text,
                            DataMatrixModule dataMatrixModule,
                            ModuleSize moduleSize,
                            PrintAlignment alignment);
```

パラメータ

text　プリンターに送信するバーコードデータ

dataMatrixModule
Data Matrixモジュール数
利用可能な定数は、
「4.4.1㉔ Data Matrixモジュール（DataMatrixModule）」を参照してください。

moduleSize　モジュールサイズ
利用可能な定数は、「4.4.1⑮ モジュールサイズ（ModuleSize）」を参照してください。

alignment　位置揃え
利用可能な定数は、「4.4.1⑩ 位置揃え（PrintAlignment）」を参照してください。

エラー　本メソッド呼び出し時にエラーが発生した場合は、PrinterExceptionをスローします。
エラーについての詳細は、「**4.4.5 PrinterExceptionクラス**」を参照してください。

注意　SLP720RT、SLP721RT、RP-F10、RP-G10、MP-B30、MP-B30L及びMP-B21Lは、クワイエットゾーンの確保を行いません。バーコードシンボルの規格に応じて、クワイエットゾーンの設定を行ってください。

参考　バーコードサイズについての詳細は、「付録B バーコードサイズ一覧」を参照してください。

PrintMaxiCode　MaxiCodeの印字

MaxiCodeを印字します。

対象　プラットフォーム共通　全プリンター

構文
```
public void PrintMaxiCode(string text,
                          MaxiCodeMode maxiCodeMode,
                          PrintAlignment alignment);
```

パラメータ

text　プリンターに送信するバーコードデータ

- maxiCodeModeが**MAXI_CODE_2**の場合
  データの先頭にサービスクラス（3桁）、国コード（3桁）、郵便番号（9桁）を付加してください。

- maxiCodeModeが**MAXI_CODE_3**の場合
  データの先頭にサービスクラス(3桁)、国コード(3桁)、郵便番号(6桁)を付加してください。

maxiCodeMode　MaxiCodeモード
利用可能な定数は、「4.4.1㉕ MaxiCodeモード(MaxiCodeMode)」を参照してください。

alignment　位置揃え
利用可能な定数は、「4.4.1⑩ 位置揃え(PrintAlignment)」を参照してください。

エラー　本メソッド呼び出し時にエラーが発生した場合は、PrinterExceptionをスローします。
エラーについての詳細は、「**4.4.5 PrinterExceptionクラス**」を参照してください。

注意　SLP720RT、SLP721RT、RP-F10、RP-G10、MP-B30、MP-B30L及びMP-B21Lは、クワイエットゾーンの確保を行いません。バーコードシンボルの規格に応じて、クワイエットゾーンの設定を行ってください。

参考　バーコードサイズについての詳細は、「付録B バーコードサイズ一覧」を参照してください。

PrintGS1DataBarStacked　GS1 Databar Stackedの印字

GS1 Databar Stackedを印字します。

対象　プラットフォーム共通　SLP720RT、SLP721RT、RP-F10、RP-G10、MP-B30、MP-B30L、MP-B21L、MP-B20、MP-A40

構文
```
public void PrintGS1DataBarStacked(string text,
                                   ModuleSize moduleSize,
                                   PrintAlignment alignment);
```

パラメータ　text　プリンターに送信するバーコードデータ
'0'～'9'を13文字入力してください。先頭の'01'はプリンターにより自動付加されます。チェックデジットはプリンターにより自動計算されます。

moduleSize　モジュールサイズ
利用可能な定数は、「4.4.1⑮ モジュールサイズ(ModuleSize)」を参照してください。

alignment　位置揃え
利用可能な定数は、「4.4.1⑩ 位置揃え(PrintAlignment)」を参照してください。

エラー　本メソッド呼び出し時にエラーが発生した場合は、PrinterExceptionをスローします。
エラーについての詳細は、「**4.4.5 PrinterExceptionクラス**」を参照してください。

参考　バーコードサイズについての詳細は、「付録B バーコードサイズ一覧」を参照してください。

PrintGS1DataBarStackedOmnidirectional　GS1 Databar Stacked Omni-directionalの印字

GS1 Databar Stacked Omni-directionalを印字します。

対象　プラットフォーム共通　SLP720RT、SLP721RT、RP-F10、RP-G10、MP-B30、MP-B30L、MP-B21L、MP-B20、MP-A40

構文
```
public void PrintGS1DataBarStackedOmnidirectional(string text,
                                                  int moduleHeight,
                                                  ModuleSize moduleSize,
                                                  PrintAlignment alignment);
```

パラメータ

text	プリンターに送信するバーコードデータ '0'～'9'を13文字入力してください。先頭の'01'はプリンターにより自動付加されます。チェックデジットはプリンターにより自動計算されます。
moduleHeight	段高さ(モジュール数) 有効範囲は、33～255です。
moduleSize	モジュールサイズ 利用可能な定数は、「4.4.1⑮ モジュールサイズ(ModuleSize)」を参照してください。
alignment	位置揃え 利用可能な定数は、「4.4.1⑩ 位置揃え(PrintAlignment)」を参照してください。

エラー　本メソッド呼び出し時にエラーが発生した場合は、PrinterExceptionをスローします。
エラーについての詳細は、「**4.4.5 PrinterExceptionクラス**」を参照してください。

参考　バーコードサイズについての詳細は、「付録B バーコードサイズ一覧」を参照してください。

PrintGS1DataBarExpandedStacked　GS1 Databar Expanded Stackedの印字

GS1 Databar Expanded Stackedを印字します。

対象　プラットフォーム共通　SLP720RT、SLP721RT、RP-F10、RP-G10、MP-B30、MP-B30L、MP-B21L、MP-B20、MP-A40

構文
```
public void PrintGS1DataBarExpandedStacked(string text,
                                           int column,
                                           ModuleSize moduleSize,
                                           PrintAlignment alignment);
```

パラメータ

text	プリンターに送信するバーコードデータ ' '、'!'、'"'、'%'、'&'、'''、'('、')'、'*'、'+'、','、'-'、'.'、'/'、':'、';'、'<'、'='、'>'、'?'、'_'、'0'～'9'、'A'～'Z'、'a'～'z'を任意の文字数分入力してください。 FNC1には、'{1'を入力してください。

column　　カラム数
1行のカラム数を指定してください。
有効範囲は、2～20の偶数です。

moduleSize　　モジュールサイズ
利用可能な定数は、「4.4.1⑮ モジュールサイズ（ModuleSize）」を参照してください。

alignment　　位置揃え
利用可能な定数は、「4.4.1⑩ 位置揃え（PrintAlignment）」を参照してください。

エラー　本メソッド呼び出し時にエラーが発生した場合は、PrinterExceptionをスローします。
エラーについての詳細は、「**4.4.5 PrinterExceptionクラス**」を参照してください。

参考　バーコードサイズについての詳細は、「付録B バーコードサイズ一覧」を参照してください。

PrintAztecCode　Aztec Codeの印字

Aztec Codeを印字します。

対象　プラットフォーム共通　MP-B30L、MP-B21L

構文
```
public void PrintAztecCode(string text,
                           int layer,
                           int errorCorrection,
                           ModuleSize moduleSize,
                           AztecSymbol aztecSymbol,
                           PrintAlignment alignment);
```

パラメータ　text　　プリンターに送信するバーコードデータ
データをUTF-8でバイナリにエンコードし、プリンターに送信します。
送信するデータが文字列で指定できない場合は、下記のエスケープシーケンスで指定できます。

エスケープシーケンス	説明
\u00nn	コントロールコード（nnは16進数で指定）
\\	バックスラッシュ

FNC1を指定するには、データに"\u001b\u0030"を入力してください。
1Bhを指定するには、データに"\u001b\u001b"を入力してください。

layer　　レイヤー数
フルレンジモードの有効範囲は、0、4～32です。
コンパクトモードの有効範囲は、0、1～4です。
0を指定した場合はレイヤーが自動設定されます。

errorCorrection
エラー訂正レベル(%)
有効範囲は、0、5～95です。
0を指定した場合は23%になります。

moduleSize　モジュールサイズ
利用可能な定数は、「4.4.1⑮ モジュールサイズ（ModuleSize）」を参照してください。

aztecSymbol　Aztecシンボル
利用可能な定数は、「4.4.1㉖ Aztecシンボル（AztecSymbol）」を参照してください。

alignment　位置揃え
利用可能な定数は、「4.4.1⑩ 位置揃え（PrintAlignment）」を参照してください。

エラー　本メソッド呼び出し時にエラーが発生した場合は、PrinterExceptionをスローします。
エラーについての詳細は、「**4.4.5 PrinterExceptionクラス**」を参照してください。

参考　バーコードサイズについての詳細は、「付録B バーコードサイズ一覧」を参照してください。

CutPaper　用紙のカット

用紙カット位置までの紙送りあり／なしを選択し、用紙のカットを行います。

対象　プラットフォーム共通　全プリンター

構文
```
public void CutPaper(CuttingMethod cuttingMethod);
```

パラメータ　cuttingMethod　カット方法
利用可能な定数は、「4.4.1㉗ カット方法（CuttingMethod）」を参照してください。

エラー　本メソッド呼び出し時にエラーが発生した場合は、PrinterExceptionをスローします。
エラーについての詳細は、「**4.4.5 PrinterExceptionクラス**」を参照してください。

説明　MP-B30、MP-B30L、MP-B21L、MP-B20またはMP-A40の場合、カットは行わず用紙カット位置までの紙送りを行います。

FeedPosition　用紙の頭出し

マーク紙またはラベル紙の頭出しを行います。

対象　プラットフォーム共通　SLP720RT、SLP721RT、MP-B30L、MP-B21L、MP-A40

構文
```
public void FeedPosition(FeedPosition feedPosition);
```

パラメータ　feedPosition　頭出し位置
利用可能な定数は、「4.4.1㉘ 頭出し位置（FeedPosition）」を参照してください。

エラー　　本メソッド呼び出し時にエラーが発生した場合は、PrinterExceptionをスローします。
エラーについての詳細は、「**4.4.5 PrinterExceptionクラス**」を参照してください。

説明　　マーク紙またはラベル紙の頭出し位置で本メソッドを実行しても、頭出しは行いません。

SendBinary　バイナリデータの送信

バイナリデータをプリンターに送信します。

対象　　プラットフォーム共通　　全デバイス

構文　　
```
public void SendBinary(byte[] data);
```

パラメータ　data　　プリンターに送信するバイナリデータ
一度に指定可能なデータサイズは3MB（3145728バイト）です。

エラー　　本メソッド呼び出し時にエラーが発生した場合は、PrinterExceptionをスローします。
エラーについての詳細は、「**4.4.5 PrinterExceptionクラス**」を参照してください。

説明　　本メソッドは、指定されたバイナリデータを変換せずにプリンターに送信します。

本メソッドでプリンターコマンドをバイナリデータとして送信することにより、ライブラリではサポートされていないプリンター機能を利用できます。ただし、本メソッドではプリンターから応答を取得するコマンドには対応しておりません。

SendDataFile　指定ファイルの送信

ファイルデータを送信します。

対象　　プラットフォーム共通　　全デバイス

構文　　
```
public void SendDataFile(string fileName,
                         PrintAlignment alignment,
                         Dithering dithering);
```

パラメータ　fileName　　プリンターに送信するデータファイルパス
入力可能な形式について下記に説明します。

iOSの場合：
　使用するファイルパス文字列

Androidの場合：

- Javaの標準的なクラス「java.io.File」が扱う絶対パス文字列
  Android 10 (API 29)以降をアプリのターゲットとする場合には、一部のファイルが直接扱えなくなりますのでご注意ください。
  詳細については、「3.4 注意事項 – 対象範囲別ストレージについて」を参照してください。

- Androidに用意されたクラス「android.net.Uri」が扱う下記スキーム名のURI文字列
  - ・file://
  - ・content://

  本パラメータには"ストレージ アクセス フレームワーク"から取得したURI文字列を指定する必要があります。"ストレージ アクセス フレームワーク"から取得せずに作成されたURIはファイルを開けない可能性がありますのでご注意ください。

指定可能なファイルサイズは、最大16MB（16777216バイト）です。
送信可能なファイル拡張子とファイルの送信について下記に説明します。

- .bmp、.jpg、.jpeg、.png
  イメージファイルとしてプリンターに送信します。カラーイメージの場合は、2値化処理によりモノクロイメージに変換して登録します。イメージファイルをプリンターのメモリ上に展開してから一括で印字を行ないます。

- .txt
  テキストデータとしてプリンターに送信します。テキストデータのフォーマットはUTF-8をサポートしています。InternationalCharacter、及びCodePageの設定を基にプリンターで印字可能なテキストデータにエンコードしてプリンターに送信します。
  本メソッドではテキストデータの最後に改行コードは付加しません。最後まで印字する場合は、テキストデータの最後に改行コードを入れてください。

- .bin、.dat
  バイナリデータとして変換せずにプリンターに送信します。

- .htm、.html
  RP-E10、RP-D10及びMP-A40のみサポートします。
  htmlデータとして変換せずにプリンターに送信します。

alignment　位置揃え
fileNameで指定されたファイルの拡張子が.bmp、.jpg、.jpeg、.png、.txtの場合に有効です。
利用可能な定数は、「4.4.1⑩ 位置揃え（PrintAlignment）」を参照してください。

dithering　ディザリング
fileNameで指定されたファイルの拡張子が.bmp、.jpg、.jpeg、.pngの場合に有効です。
利用可能な定数は、「4.4.1⑥ ディザリング（Dithering）」を参照してください。

エラー　本メソッド呼び出し時にエラーが発生した場合は、PrinterExceptionをスローします。
エラーについての詳細は、「**4.4.5 PrinterExceptionクラス**」を参照してください。

SendImage　イメージの送信

指定したイメージを印字します。

対象　プラットフォーム共通　全デバイス

構文
```
public void SendImage(object image,
                      PrintAlignment alignment,
                      Dithering dithering);
```

パラメータ

image	プリンターに送信するイメージデータ iOSの場合： iOSに用意されたクラス「UIKit.UIImage」に準するイメージデータを指定してください。 Androidの場合： Androidに用意されたクラス「Android.Graphics.Bitmap」に準するイメージデータを指定してください。
alignment	位置揃え 利用可能な定数は、「4.4.1⑩ 位置揃え（PrintAlignment）」を参照してください。
dithering	ディザリング 利用可能な定数は、「4.4.1⑥ ディザリング（Dithering）」を参照してください。

エラー　本メソッド呼び出し時にエラーが発生した場合は、PrinterExceptionをスローします。
エラーについての詳細は、「**4.4.5 PrinterExceptionクラス**」を参照してください。

PrintPDF　PDFページの印字

選択したPDFファイルの指定したページを印字します。

対象　プラットフォーム共通　全プリンター

構文
```
public void PrintPDF(string fileName,
                     int startIndex,
                     int endIndex,
                     Rotate rotate,
                     ImageScale imageScale,
                     int bottomMargin,
                     Dithering dithering,
                     PrintAlignment alignment);
```

パラメータ

fileName	PDFファイルパス サポートするPDFのファイル拡張子は、.pdfです。 指定可能なファイルサイズは、最大16MB（16777216バイト）です。 入力可能な形式について下記に説明します。

iOSの場合：
使用するファイルパス文字列

Androidの場合：

- Javaの標準的なクラス「java.io.File」が扱う絶対パス文字列
  Android 10 (API 29)以降をアプリのターゲットとする場合には、一部のファイルが直接扱えなくなりますのでご注意ください。
  詳細については、「3.4 注意事項 － 対象範囲別ストレージについて」を参照してください。

- Androidに用意されたクラス「android.net.Uri」が扱う下記スキーム名のURI文字列
  ・file://
  ・content://
  本パラメータには"ストレージ アクセス フレームワーク"から取得したURI文字列を指定する必要があります。"ストレージ アクセス フレームワーク"から取得せずに作成されたURIはファイルを開けない可能性がありますのでご注意ください。

startIndex	印字するページの開始番号 有効範囲は、-1、及び1～PDFファイルのページ数です。 startIndexに-1を指定した場合、全てのページを印字します。 startIndexに-1を指定した場合、endIndexの値は無視されます。 startIndexに1以上を指定した場合、startIndexに指定したページ番号からendIndexに指定したページ番号のページまでを印字します。 startIndexにendIndexで指定した値より大きい値を指定した場合はエラーとなります。 startIndex及びendIndexに範囲外の値を指定した場合、エラーとなります。
endIndex	印字するページの終了番号 有効範囲は、1～2147483647です。 endIndexにPDFファイルページ数より大きい値を指定した場合、startIndexに指定したページ番号からPDFファイルの最後のページまでを印字します。
rotate	イメージの回転方向 利用可能な定数は、「4.4.1㉙ イメージの回転方向(Rotate)」を参照してください。
imageScale	イメージの拡大縮小 利用可能な定数は、「4.4.1㉚ イメージの拡大縮小(ImageScale)」を参照してください。 imageScaleに**IMAGE_SCALE_WIDTH_FIT**を指定した場合、アスペクト比を維持しながら画像の幅をプリンターの印字幅に変換します。

bottomMargin　下余白(ドット)
有効範囲は、-1、及び0～2400です。
bottomMarginに-1を指定した場合、下余白を維持してイメージを作成し印字します。
bottomMarginに0～2400の値を指定した場合、下余白を指定したサイズに変更します。
bottomMarginに0～2400の値を指定した場合、空白ページの印字は行いません。

dithering　ディザリング
利用可能な定数は、「4.4.1⑥ ディザリング(Dithering)」を参照してください。

alignment　位置揃え。
利用可能な定数は、「4.4.1⑩ 位置揃え(PrintAlignment)」を参照してください。

エラー　本メソッド呼び出し時にエラーが発生した場合は、PrinterExceptionをスローします。
エラーについての詳細は、「**4.4.5 PrinterExceptionクラス**」を参照してください。

説明　PDFデータを、プリンターで印字可能な形式に変換してプリンターに送信します。

カラーのイメージデータは、二値化処理によりモノクロイメージに変換します。

下記のメモリスイッチが「有効」の場合、PDFページの印字後マーク紙またはラベル紙の頭出しを行います。
- ・MP-B21L、MP-B30L　：MS1-2(マークギャップモード選択)
- ・MP-B30、MP-A40　：MS1-2(マークモード選択)
- ・SLP720RT、SLP721RT、RP-E10　：MS1-3(マークモード選択)

注意　一度に100枚より多く印字した場合の印字動作保証は行っておりません。

PrintLogo　ロゴの印字

登録済みのロゴを印字します。

対象　プラットフォーム共通　全プリンター

構文
```
public void PrintLogo(string logoId, PrintAlignment alignment);
```

パラメータ　logoId　印字するロゴのID(キーコード)
登録済みのロゴのIDを文字列で指定してください。

alignment　位置揃え
利用可能な定数は、「4.4.1⑩ 位置揃え(PrintAlignment)」を参照してください。

エラー　本メソッド呼び出し時にエラーが発生した場合は、PrinterExceptionをスローします。
エラーについての詳細は、「**4.4.5 PrinterExceptionクラス**」を参照してください。

PrintSmartLabelImageData　ラベルの印字

ラベルを印字します。

対象　プラットフォーム共通　SLP720RT、SLP721RT、MP-B30L、MP-B21L

構文　`public void PrintSmartLabelImageData(SmartLabelManager labelManager);`

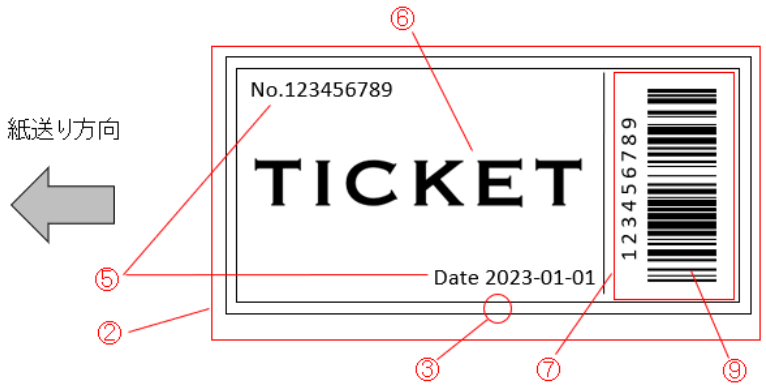
パラメータ　`labelManager`　ラベルファイル(*.sl、*.slex)をプリンターで印字できるデータに変換する機能を提供するクラス

エラー　本メソッド呼び出し時にエラーが発生した場合は、PrinterExceptionをスローします。
エラーについての詳細は、「**4.4.5 PrinterExceptionクラス**」を参照してください。

説明　本メソッドを利用した印字例については、「**4.4.6 SmartLabelManagerクラス**」を参照してください。

③　ページモード専用メソッド

ページモードを利用するための専用メソッドです。ページモードで印字する手順例を下記に示します。



① ページモードを開始します

```
printerManager.EnterPageMode();
```

② ページモードの印字領域を指定します

```
printerManager.SetPageModeArea(0, 0, 355, 576);
```

③ 矩形と罫線を指定します

```
printerManager.PrintPageModeRectangle(0, 0, 344, 575, LineStyle.LINESTYLE_THIN);
printerManager.PrintPageModeRectangle(7, 7, 336, 567, LineStyle.LINESTYLE_THIN);
printerManager.PrintPageModeLine(11, 404, 334, 404, LineStyle.LINESTYLE_THIN);
```

④ ページモードの印字方向を指定します

```
printerManager.SetPageModeDirection(Direction.DIRECTION_TOP_TO_BOTTOM );
```

⑤ 文字を指定します

```
printerManager.PrintPageModeText(21, 47, "NO.123456789");
printerManager.PrintPageModeText(212, 340, "Date 2023-01-01");
```

⑥ イメージファイルを指定します

```
printerManager.PrintPageModeImageFile(
            10,
            222,
            System.IO.Path.Combine(FileSystem.Current.AppDataDirectory, "TicketImage.jpg"),
            Dithering.DITHERING_DISABLE);
```

⑦ ページモードの印字領域を指定します

```
printerManager.SetPageModeArea(0, 404, 345, 163);
```

⑧ 印字方向を指定します

```
printerManager.SetPageModeDirection(Direction.DIRECTION_LEFT_TO_RIGHT);
```

⑨ バーコードを指定します

```
printerManager.PrintPageModeBarcode(
            20,
            132,
            BarcodeSymbol.BARCODE_SYMBOL_CODE128,
            new byte[]{0x67, 0x11, 0x12, 0x13, 0x14, 0x15, 0x16, 0x17, 0x18, 0x19x, 0x68},
            ModuleSize.BARCODE_MODULE_WIDTH_2,
            80,
            HriPosition.HRI_POSITION_ABOVE,
            CharacterFont.FONT_A);
```

⑩ ページモードを印字します

```
printerManager.PrintPageMode(CuttingMethod.CUT_PARTIAL);
```

⑪ ページモードを終了します

```
printerManager.ExitPageMode();
```

EnterPageMode　ページモードの開始

ページモードを開始します。

対象　プラットフォーム共通　RP-F10、RP-G10、RP-E10、RP-D10、MP-B30、MP-B30L、MP-B21L、MP-A40

構文　`public void EnterPageMode();`

エラー　本メソッド呼び出し時にエラーが発生した場合は、PrinterExceptionをスローします。
エラーについての詳細は、「**4.4.5 PrinterExceptionクラス**」を参照してください。

説明　本メソッドによりページモードが開始されます。以後は、ページモード専用メソッドとスタンダードモード・ページモード共通メソッドが使用できます。

ExitPageModeを実行すると、ページデータバッファに保持している印字データを破棄し、スタンダードモードに移行します。
PrintPageModeを実行すると、ページデータバッファに保持している印字データを印字します。

ExitPageMode　ページモードの終了

ページモードを終了し、スタンダードモードに戻ります。

対象　プラットフォーム共通　RP-F10、RP-G10、RP-E10、RP-D10、MP-B30、MP-B30L、MP-B21L、MP-A40

構文　`public void ExitPageMode();`

エラー　本メソッド呼び出し時にエラーが発生した場合は、PrinterExceptionをスローします。
エラーについての詳細は、「**4.4.5 PrinterExceptionクラス**」を参照してください。

説明　ページデータバッファに保持している印字データを破棄し、スタンダードモードに移行します。

SetPageModeArea　ページモードの印字領域の指定

ページモードの印字領域を指定します。

対象　プラットフォーム共通　RP-F10、RP-G10、RP-E10、RP-D10、MP-B30、MP-B30L、MP-B21L、MP-A40

構文　`public void SetPageModeArea(int x, int y, int width, int height);`

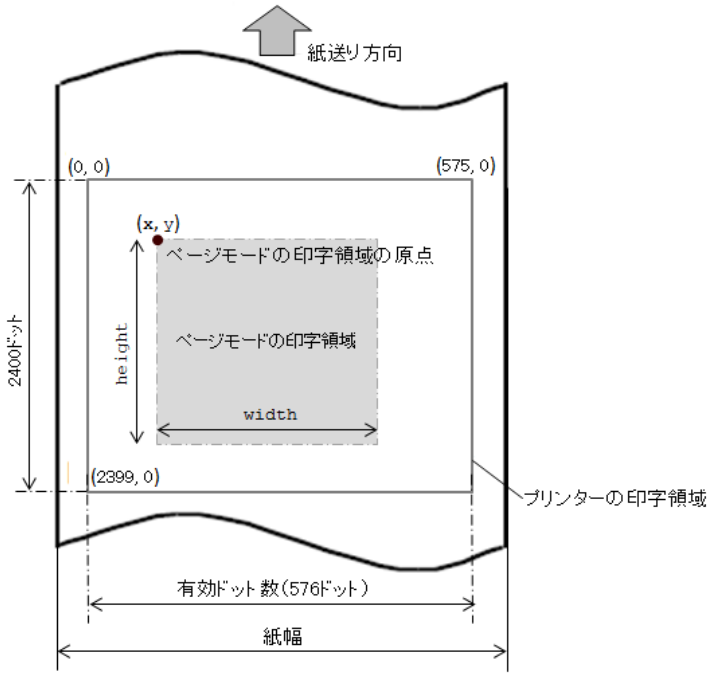
パラメータ　x　ページモードの印字領域の横方向の原点(ドット)
有効範囲は、0～(印字幅-1)です。
0はプリンターの印字領域の左端になります。

y　ページモードの印字領域の縦方向の原点(ドット)
有効範囲は、0～2399です。
0は紙送りをしていない位置になります。

width　　ページモードの印字領域幅(ドット)
有効範囲は、1～(印字幅-x)です。

height　　ページモードの印字領域高さ(ドット)
有効範囲は、1～(2400-y)です。

ページモードの印字領域とプリンターの印字領域の関係を、印字幅576ドットの場合を例として下記に示します。



エラー　　本メソッド呼び出し時にエラーが発生した場合は、PrinterExceptionをスローします。
エラーについての詳細は、「**4.4.5 PrinterExceptionクラス**」を参照してください。

説明　　本メソッドの実行前にEnterPageModeでページモードを開始してください。

EnterPageModeでページモードを開始し、ページモード専用メソッドを実行した後に本メソッドを実行すると、追加してページモードの印字領域を指定できます。それまでに展開されているデータは保持されます。
本メソッド実行後は本メソッドで追加したページモードの印字領域に対して、ページモード専用メソッドのデータが展開されます。

EnterPageModeを実行すると、ページモードの印字領域は、x=0、y=0、width=印字幅、height=2400になります。

MP-B30の印字幅は576です。
MP-B30L、MP-B21Lの印字幅はプリンターのメモリスイッチMS3(印字幅)の幅になります。
MP-A40の印字幅はプリンターのメモリスイッチMS3-1～6(紙幅選択)の印字幅になります。
RP-F10、RP-G10、RP-E10、RP-D10の印字幅はプリンターのメモリスイッチMS4-5(有効ドット数選択)に適用された有効ドット数になります。
メモリスイッチの詳細、及び工場出荷時の設定については、「取扱説明書」を参照してください。

SetPageModeDirection　ページモードの印字方向の指定

ページモードの印字方向を指定します。

対象　プラットフォーム共通　RP-F10、RP-G10、RP-E10、RP-D10、MP-B30、MP-B30L、MP-B21L、MP-A40

構文　`public void SetPageModeDirection(Direction direction);`

パラメータ　`direction`　印字方向
利用可能な定数は、「4.4.1㉛ 印字方向(Direction)」を参照してください。

エラー　本メソッド呼び出し時にエラーが発生した場合は、PrinterExceptionをスローします。
エラーについての詳細は、「**4.4.5 PrinterExceptionクラス**」を参照してください。

説明　本メソッドの実行前にEnterPageModeでページモードを開始してください。

EnterPageModeを実行すると、印字方向は左→右になります。

SetPageModeLineSpacing　ページモードの改行量の指定

ページモードの改行量を指定します。

対象　プラットフォーム共通　RP-F10、RP-G10、RP-E10、RP-D10、MP-B30、MP-B30L、MP-B21L、MP-A40

構文　`public void SetPageModeLineSpacing(int lineSpacing);`

パラメータ　`lineSpacing`　ページモードの改行量(ドット)
有効範囲は、0～255です。

エラー　本メソッド呼び出し時にエラーが発生した場合は、PrinterExceptionをスローします。
エラーについての詳細は、「**4.4.5 PrinterExceptionクラス**」を参照してください。

説明　本メソッドの実行前にEnterPageModeでページモードを開始してください。

EnterPageModeを実行すると、改行量は34ドットになります。

PrintPageMode　ページモードの印字

ページデータバッファに保持している印字データを印字します。

対象　プラットフォーム共通　RP-F10、RP-G10、RP-E10、RP-D10、MP-B30、MP-B30L、MP-B21L、MP-A40

構文　`public void PrintPageMode(CuttingMethod cuttingMethod);`

パラメータ　`cuttingMethod`　カット方法
利用可能な定数は、「4.4.1㉗ カット方法(CuttingMethod)」を参照してください。

エラー　本メソッド呼び出し時にエラーが発生した場合は、PrinterExceptionをスローします。
エラーについての詳細は、「**4.4.5 PrinterExceptionクラス**」を参照してください。

説明　印字後も印字データは保持されます。印字データは下記のタイミングで破棄されます。
・EnterPageModeを実行した場合
・Disconnectを実行した場合
・ExitPageModeを実行した場合

PrintPageModeText　ページモードのテキストデータの送信

テキストデータをページモードの印字領域に展開します。

対象　プラットフォーム共通　RP-F10、RP-G10、RP-E10、RP-D10、MP-B30、MP-B30L、MP-B21L、MP-A40

構文
```
public void PrintPageModeText(int startX, int startY, string text);
```

パラメータ

startX	始点からの横方向の基準点(ドット) 有効範囲は、0～2399です。
startY	始点からの縦方向の基準点(ドット) 有効範囲は、0～2399です。
text	テキストデータ 一度に指定可能なデータサイズは16KB(16384バイト)です。

エラー　本メソッド呼び出し時にエラーが発生した場合は、PrinterExceptionをスローします。
エラーについての詳細は、「**4.4.5 PrinterExceptionクラス**」を参照してください。

説明　本メソッドは、指定されたテキストデータを、InternationalCharacter、及びCodePageを基にプリンターで印字可能なテキストデータにエンコードします。

本メソッドの実行前にEnterPageModeでページモードを開始してください。

PrintPageModeTextEx　ページモードの書式指定テキストデータの送信

書式設定されたテキストデータをページモードの印字領域に展開します。

対象　プラットフォーム共通　RP-F10、RP-G10、RP-E10、RP-D10、MP-B30、MP-B30L、MP-B21L、MP-A40

構文
```
public void PrintPageModeTextEx(int startX,
                                int startY,
                                string text,
                                CharacterBold bold,
                                CharacterUnderline underline,
                                CharacterReverse reverse,
                                CharacterFont font,
                                CharacterScale scale);
```

パラメータ　startX　始点からの横方向の基準点(ドット)
有効範囲は、0～2399です。

startY　始点からの縦方向の基準点(ドット)
有効範囲は、0～2399です。

text　テキストデータ
一度に指定可能なデータサイズは16KB(16384バイト)です。

bold　強調文字
利用可能な定数は、「4.4.1⑪ 強調文字(CharacterBold)」を参照してください。

underline　アンダーライン
利用可能な定数は、「4.4.1⑫ アンダーライン(CharacterUnderline)」を参照してください。

reverse　白黒反転印字
利用可能な定数は、「4.4.1⑰ 白黒反転印字(CharacterReverse)」を参照してください。

font　文字フォント
利用可能な定数は、「4.4.1⑭ 文字フォント(CharacterFont)」を参照してください。

scale　文字倍率
利用可能な定数は、「4.4.1⑬ 文字倍率(CharacterScale)」を参照してください。

エラー　本メソッド呼び出し時にエラーが発生した場合は、PrinterExceptionをスローします。
エラーについての詳細は、「**4.4.5 PrinterExceptionクラス**」を参照してください。

説明　本メソッドは、指定されたテキストデータを、InternationalCharacter、及びCodePageを基にプリンターで印字可能なテキストデータにエンコードします。

本メソッドの実行前にEnterPageModeでページモードを開始してください。

PrintPageModeBarcode　ページモードのバーコードの印字

バーコードをページモードの印字領域に展開します。
構文(a)は、バーコードデータを文字列で指定します。
構文(b)は、バーコードデータを文字列で指定でき、バーコードのNW比を指定できます。
構文(c)は、バーコードデータをバイト配列で指定できます。
構文(d)は、カスタマバーコードを文字列で指定できます。

対象　プラットフォーム共通　構文(a)～構文(c)　:RP-F10、RP-G10、RP-E10、RP-D10、MP-B30、MP-B30L、MP-B21L、MP-A40
構文(d)　:MP-B21L

構文　(a) public void PrintPageModeBarcode(int startX,
int startY,
BarcodeSymbol barcodeSymbol,
string text,
ModuleSize moduleSize,
int moduleHeight,
HriPosition hriPosition,
CharacterFont hriFont);

(b) public void PrintPageModeBarcode(int startX,
int startY,
BarcodeSymbol barcodeSymbol,
string text,
ModuleSize moduleSize,
int moduleHeight,
HriPosition hriPosition,
CharacterFont hriFont,
NwRatio nwRatio);

(c) public void PrintPageModeBarcode(int startX,
int startY,
BarcodeSymbol barcodeSymbol,
byte[] data,
ModuleSize moduleSize,
int moduleHeight,
HriPosition hriPosition,
CharacterFont hriFont);

(d)public void PrintPageModeBarcode(int startX,
int startY,
BarcodeSymbol barcodeSymbol,
string text,
ModuleSize moduleSize);

パラメータ　startX　始点からの横方向の基準点(ドット)
有効範囲は、0～2399です。

startY　始点からの縦方向の基準点(ドット)
有効範囲は、0～2399です。

barcodeSymbol　バーコードシンボル
利用可能な定数、及び対応する構文は、
「4.4.1⑲ バーコードシンボル(BarcodeSymbol)」を参照してください。

text(data)　　プリンターに送信するバーコードデータ
バーコードデータの入力条件は下記の通りです。

バーコード	データ数	入力可能データ 文字列（データ）	備考
UPC-A	11～12文字	'0' ～ '9'	
UPC-E	11～12文字	'0' ～ '9'	
EAN13 JAN13	12～13文字	'0' ～ '9'	
EAN8 JAN8	7～8文字	'0' ～ '9'	
CODE39	1～150文字	'0' ～ '9'、'A' ～ 'Z' ' '、'$'、'%'、'+'、'–'、'.'、'/'	スタートコード及びストップコード('*')は自動付加されます。
CODE93	1～150バイト	(0x00 ～ 0x2E)	末尾に0x2F以上のデータを入力してください。
CODE128	2～150バイト	(0x00 ～ 0x66)	CODE128コードセットのスタートコード(0x67～0x69)で入力する場合。 末尾に0x67以上のデータを入力してください。
		(0x00 ～ 0x7F)	CODE128特殊コードのスタートコード(″{A″, ″{B″, ″{C″)で開始する場合。
ITF	2～150文字 (ただし偶数個)	'0' ～ '9'	
CODABAR	1～150文字	'0' ～ '9' '$'、'+'、'–'、'.'、'/'、':'	
EAN13 add-on JAN13 add-on	Add-on 2: 14～15文字 Add-on 5: 17～18文字	'0' ～ '9'	
カスタマバーコード	7～20文字	'0' ～ '9' 'A' ～ 'Z' '–'	スタートコード、チェックデジット、ストップコードは自動付加されます。 先頭から7文字は'0'～'9'のいずれかを指定する必要があります。'A'～'Z'は2文字として計算されます。
GS1 Databar Omni-directional	13文字	'0' ～ '9'	チェックデジットは自動計算されます。
GS1 Databar Truncated	13文字	'0' ～ '9'	チェックデジットは自動計算されます。
GS1 Databar Limited	13文字	'0' ～ '9'	チェックデジットは自動計算されます。
GS1 Databar Expanded	2～255文字	' ' ～ '″' '%' ～ '?' 'A' ～ 'Z' ' ' 'a' ～ 'z' '{'	

moduleSize　　バーコードの幅
利用可能な定数は、「4.4.1⑮ モジュールサイズ（ModuleSize）」を参照してください。

moduleHeight　バーコードの高さ(ドット)

- barcodeSymbolが下記の場合は、有効範囲は1～255です。
  BARCODE_UPC_A
  BARCODE_UPC_E
  BARCODE_EAN13
  BARCODE_JAN13
  BARCODE_EAN8
  BARCODE_JAN8
  BARCODE_CODE39
  BARCODE_CODE93
  BARCODE_CODE128
  BARCODE_ITF
  BARCODE_CODABAR
  BARCODE_EAN13_ADDON
  BARCODE_JAN13_ADDON

- barcodeSymbolが下記の場合は、barcodeSymbolとmoduleSizeにより有効範囲が異なります。

barcodeSymbol	moduleSize	有効範囲
BARCODE_GS1_OMNI_DIRECTIONAL		
	BARCODE_MODULE_WIDTH_2	66 ～ 255
	BARCODE_MODULE_WIDTH_3	99 ～ 255
	BARCODE_MODULE_WIDTH_4	132 ～ 255
	BARCODE_MODULE_WIDTH_5	165 ～ 255
	BARCODE_MODULE_WIDTH_6	198 ～ 255
BARCODE_GS1_TRUNCATED		
	BARCODE_MODULE_WIDTH_2	26 ～ 255
	BARCODE_MODULE_WIDTH_3	39 ～ 255
	BARCODE_MODULE_WIDTH_4	52 ～ 255
	BARCODE_MODULE_WIDTH_5	65 ～ 255
	BARCODE_MODULE_WIDTH_6	78 ～ 255
BARCODE_GS1_LIMITED		
	BARCODE_MODULE_WIDTH_2	20 ～ 255
	BARCODE_MODULE_WIDTH_3	30 ～ 255
	BARCODE_MODULE_WIDTH_4	40 ～ 255
	BARCODE_MODULE_WIDTH_5	50 ～ 255
	BARCODE_MODULE_WIDTH_6	60 ～ 255

barcodeSymbol	
moduleSize	有効範囲
BARCODE_GS1_EXPANDED	
BARCODE_MODULE_WIDTH_2	68 ～ 255
BARCODE_MODULE_WIDTH_3	102 ～ 255
BARCODE_MODULE_WIDTH_4	136 ～ 255
BARCODE_MODULE_WIDTH_5	170 ～ 255
BARCODE_MODULE_WIDTH_6	204 ～ 255

hriPosition　HRI文字印字位置
利用可能な定数は、「4.4.1⑳ HRI文字印字位置(HriPosition)」を参照してください。

hriFont　HRI文字フォント
利用可能な定数は、「4.4.1⑭ 文字フォント(CharacterFont)」を参照してください。

nwRatio　NW比
利用可能な定数は、「4.4.1㉑ NW比(NwRatio)」を参照してください。
指定したnwRatioとmoduleSizeにより、太エレメントの幅が下表のように設定されます。

moduleSize	nwRatio		
	NWRATIO_1TO2	**NWRATIO_1TO2_5**	**NWRATIO_1TO3**
BARCODE_MODULE_WIDTH_2	0.500 mm (4ドット)	0.625 mm (5ドット)	0.750 mm (6ドット)
BARCODE_MODULE_WIDTH_3	0.750 mm (6ドット)	1.000 mm (8ドット)	1.125 mm (9ドット)
BARCODE_MODULE_WIDTH_4	1.000 mm (8ドット)	1.250 mm (10ドット)	1.500 mm (12ドット)
BARCODE_MODULE_WIDTH_5	1.250 mm (10ドット)	1.625 mm (13ドット)	1.875 mm (15ドット)
BARCODE_MODULE_WIDTH_6	1.500 mm (12ドット)	1.875 mm (15ドット)	2.250 mm (18ドット)

エラー　本メソッド呼び出し時にエラーが発生した場合は、PrinterExceptionをスローします。
エラーについての詳細は、「**4.4.5 PrinterExceptionクラス**」を参照してください。

説明　本メソッドの実行前にEnterPageModeでページモードを開始してください。

注意　バーコードの印字データは、その他の印字データと重ならないように展開してください。
RP-F10、RP-G10、MP-B30、MP-B30L及びMP-B21Lは、クワイエットゾーンの確保を行いません。バーコードシンボルの規格に応じて、クワイエットゾーンの設定を行ってください。

参考　バーコードサイズについての詳細は、「付録B バーコードサイズ一覧」を参照してください。

PrintPageModePDF417　ページモードのPDF417の印字

PDF417をページモードの印字領域に展開します。

対象　プラットフォーム共通　RP-F10、RP-G10、RP-E10、RP-D10、MP-B30、MP-B30L、MP-B21L、MP-A40

構文

```
public void PrintPageModePDF417(int startX,
                                 int startY,
                                 string text,
                                 ErrorCorrection errorCorrection,
                                 int row,
                                 int column,
                                 ModuleSize moduleSize,
                                 int moduleHeight,
                                 Pdf417Symbol pdf417Symbol);
```

パラメータ

startX	始点からの横方向の基準点(ドット) 有効範囲は、0～2399です。
startY	始点からの縦方向の基準点(ドット) 有効範囲は、0～2399です。
text	バーコードデータ
errorCorrection	エラー訂正レベル 利用可能な定数は、「4.4.1⑯ エラー訂正レベル(ErrorCorrection)」を参照してください。
row	段数(段) 有効範囲は、0、3～90です。 0を指定した場合は段数が自動設定されます。
column	データ領域のカラム数 有効範囲は、0～30です。 0を指定した場合はデータ領域のカラム数が自動設定されます。
moduleSize	バーコードの幅 利用可能な定数は、「4.4.1⑮ モジュールサイズ(ModuleSize)」を参照してください。
moduleHeight	段高さ(ドット) 有効範囲は、2～127です。 段高さの設定を小さくすると、バーコードスキャナーによっては読み取れない場合があります。通常の使用では、3以上を設定してください。
pdf417Symbol	PDF417シンボル 利用可能な定数は、「4.4.1㉒ PDF417シンボル(Pdf417Symbol)」を参照してください。

エラー　本メソッド呼び出し時にエラーが発生した場合は、PrinterExceptionをスローします。
エラーについての詳細は、「**4.4.5 PrinterExceptionクラス**」を参照してください。

説明　本メソッドの実行前にEnterPageModeでページモードを開始してください。

注意　バーコードの印字データは、その他の印字データと重ならないように展開してください。
RP-F10、RP-G10、MP-B30、MP-B30L及びMP-B21Lは、クワイエットゾーンの確保を行いません。バーコードシンボルの規格に応じて、クワイエットゾーンの設定を行ってください。

参考　バーコードサイズについての詳細は、「付録B バーコードサイズ一覧」を参照してください。

PrintPageModeQRcode　ページモードのQRコードの印字

QRコードをページモードの印字領域に展開します。

対象　プラットフォーム共通　RP-F10、RP-G10、RP-E10、RP-D10、MP-B30、MP-B30L、MP-B21L、MP-A40

構文
```
public void PrintPageModeQRcode(int startX,
                                 int startY,
                                 string text,
                                 ErrorCorrection errorCorrection,
                                 ModuleSize moduleSize,
                                 QrModel model);
```

パラメータ

startX　始点からの横方向の基準点（ドット）
有効範囲は、0～2399です。

startY　始点からの縦方向の基準点（ドット）
有効範囲は、0～2399です。

text　バーコードデータ
バージョンはtextで設定したデータバイト数に応じて自動設定されます。

errorCorrection
エラー訂正レベル
利用可能な定数は、「4.4.1⑯ エラー訂正レベル（ErrorCorrection）」を参照してください。

moduleSize　バーコードの幅
利用可能な定数は、「4.4.1⑮ モジュールサイズ（ModuleSize）」を参照してください。

model　QRコードモデル
利用可能な定数は、「4.4.1㉓　QRコードモデル（QrModel）」を参照してください。

エラー　本メソッド呼び出し時にエラーが発生した場合は、PrinterExceptionをスローします。
エラーについての詳細は、「**4.4.5 PrinterExceptionクラス**」を参照してください。

説明　本メソッドの実行前にEnterPageModeでページモードを開始してください。

注意　　バーコードの印字データは、その他の印字データと重ならないように展開してください。
RP-F10、RP-G10、MP-B30、MP-B30L及びMP-B21Lは、クワイエットゾーンの確保を行いません。バーコードシンボルの規格に応じて、クワイエットゾーンの設定を行ってください。

参考　　バーコードサイズについての詳細は、「付録B バーコードサイズ一覧」を参照してください。

PrintPageModeDataMatrix　　ページモードのData Matrixの印字

Data Matrixをページモードの印字領域に展開します。

対象　　プラットフォーム共通　　RP-F10、RP-G10、RP-E10、RP-D10、MP-B30、MP-B30L、MP-B21L、MP-A40

構文
```
public void PrintPageModeDataMatrix(int startX,
                                    int startY,
                                    string text,
                                    DataMatrixModule dataMatrixModule,
                                    ModuleSize moduleSize);
```

パラメータ

startX　　始点からの横方向の基準点(ドット)
有効範囲は、0～2399です。

startY　　始点からの縦方向の基準点(ドット)
有効範囲は、0～2399です。

text　　バーコードデータ

dataMatrixModule
Data Matrixモジュール数
利用可能な定数は、
「4.4.1㉔ Data Matrixモジュール(DataMatrixModule)」を参照してください。

moduleSize　　バーコードの幅
利用可能な定数は、「4.4.1⑮ モジュールサイズ(ModuleSize)」を参照してください。

エラー　　本メソッド呼び出し時にエラーが発生した場合は、PrinterExceptionをスローします。
エラーについての詳細は、「**4.4.5 PrinterExceptionクラス**」を参照してください。

説明　　本メソッドの実行前にEnterPageModeでページモードを開始してください。

注意　　バーコードの印字データは、その他の印字データと重ならないように展開してください。
RP-F10、RP-G10、MP-B30、MP-B30L及びMP-B21Lは、クワイエットゾーンの確保を行いません。バーコードシンボルの規格に応じて、クワイエットゾーンの設定を行ってください。

参考　　バーコードサイズについての詳細は、「付録B バーコードサイズ一覧」を参照してください。

PrintPageModeMaxiCode　　ページモードのMaxiCodeの印字

MaxiCodeをページモードの印字領域に展開します。

対象　　プラットフォーム共通　　RP-F10、RP-G10、RP-E10、RP-D10、MP-B30、MP-B30L、MP-B21L、MP-A40

構文

```
public void PrintPageModeMaxiCode(int startX,
                                  int startY,
                                  string text,
                                  MaxiCodeMode maxiCodeMode);
```

パラメータ

startX	始点からの横方向の基準点(ドット) 有効範囲は、0～2399です。
startY	始点からの縦方向の基準点(ドット) 有効範囲は、0～2399です。
text	バーコードデータ ● maxiCodeModeが**MAXI_CODE_2**の場合 データの先頭にサービスクラス(3桁)、国コード(3桁)、郵便番号(9桁)を付加してください。 ● maxiCodeModeが**MAXI_CODE_3**の場合 データの先頭にサービスクラス(3桁)、国コード(3桁)、郵便番号(6桁)を付加してください。
maxiCodeMode	MaxiCodeモード 利用可能な定数は、「4.4.1㉕ MaxiCodeモード(MaxiCodeMode)」を参照してください。

エラー　　本メソッド呼び出し時にエラーが発生した場合は、PrinterExceptionをスローします。
エラーについての詳細は、「**4.4.5 PrinterExceptionクラス**」を参照してください。

説明　　本メソッドの実行前にEnterPageModeでページモードを開始してください。

注意　　バーコードの印字データは、その他の印字データと重ならないように展開してください。
RP-F10、RP-G10、MP-B30、MP-B30L及びMP-B21Lは、クワイエットゾーンの確保を行いません。バーコードシンボルの規格に応じて、クワイエットゾーンの設定を行ってください。

参考　　バーコードサイズについての詳細は、「付録B バーコードサイズ一覧」を参照してください。

PrintPageModeGS1DataBarStacked
ページモードのGS1 Databar Stackedの印字

GS1 Databar Stackedをページモードの印字領域に展開します。

対象　　プラットフォーム共通　　RP-F10、RP-G10、MP-B30、MP-B30L、MP-B21L、MP-A40

構文
```
public void PrintPageModeGS1DataBarStacked(int startX,
                                           int startY,
                                           string text,
                                           ModuleSize moduleSize);
```

パラメータ

startX	始点からの横方向の基準点(ドット) 有効範囲は、0～2399です。
startY	始点からの縦方向の基準点(ドット) 有効範囲は、0～2399です。
text	バーコードデータ '0'～'9'を13文字入力してください。先頭の'01'はプリンターにより自動付加されます。チェックデジットはプリンターにより自動計算されます。
moduleSize	バーコードの幅 利用可能な定数は、「4.4.1⑮ モジュールサイズ(ModuleSize)」を参照してください。

エラー　　本メソッド呼び出し時にエラーが発生した場合は、PrinterExceptionをスローします。
エラーについての詳細は、「**4.4.5 PrinterExceptionクラス**」を参照してください。

説明　　本メソッドの実行前にEnterPageModeでページモードを開始してください。

注意　　バーコードの印字データは、その他の印字データと重ならないように展開してください。

参考　　バーコードサイズについての詳細は、「付録B バーコードサイズ一覧」を参照してください。

PrintPageModeGS1DataBarStackedOmnidirectional
ページモードのGS1 Databar Stacked Omni-directionalの印字

GS1 Databar Stacked Omni-directionalをページモードの印字領域に展開します。

対象　　プラットフォーム共通　　RP-F10、RP-G10、MP-B30、MP-B30L、MP-B21L、MP-A40

構文
```
public void PrintPageModeGS1DataBarStackedOmnidirectional(int startX,
                                                          int startY,
                                                          string text,
                                                          int moduleHeight,
                                                          ModuleSize moduleSize);
```

パラメータ

startX	始点からの横方向の基準点(ドット) 有効範囲は、0～2399です。

	startY	始点からの縦方向の基準点(ドット) 有効範囲は、0～2399です。
	text	バーコードデータ '0'～'9'を13文字入力してください。先頭の'01'はプリンターにより自動付加されます。チェックデジットはプリンターにより自動計算されます。
	moduleHeight	段高さ(モジュール数) 有効範囲は、33～255です。
	moduleSize	バーコードの幅 利用可能な定数は、「4.4.1⑮ モジュールサイズ(ModuleSize)」を参照してください。

エラー　本メソッド呼び出し時にエラーが発生した場合は、PrinterExceptionをスローします。
エラーについての詳細は、「**4.4.5 PrinterExceptionクラス**」を参照してください。

説明　本メソッドの実行前にEnterPageModeでページモードを開始してください。

注意　バーコードの印字データは、その他の印字データと重ならないように展開してください。

参考　バーコードサイズについての詳細は、「付録B バーコードサイズ一覧」を参照してください。

PrintPageModeGS1DataBarExpandedStacked
ページモードのGS1 Databar Expanded Stackedの印字

GS1 Databar Expanded Stackedをページモードの印字領域に展開します。

対象　プラットフォーム共通　RP-F10、RP-G10、MP-B30、MP-B30L、MP-B21L、MP-A40

構文
```
public void PrintPageModeGS1DataBarExpandedStacked(int startX,
                                                   int startY,
                                                   string text,
                                                   int column,
                                                   ModuleSize moduleSize);
```

パラメータ	startX	始点からの横方向の基準点(ドット) 有効範囲は、0～2399です。
	startY	始点からの縦方向の基準点(ドット) 有効範囲は、0～2399です。
	text	バーコードデータ ' '、'!'、'"'、'%'、'&'、"'"、'('、')'、'*'、'+'、','、'-'、'.'、'/'、':'、';'、'<'、'='、'>'、'?'、'_'、'0'～'9'、'A'～'Z'、'a'～'z'を任意の文字数分入力してください。 FNC1には、'{1'を入力してください。
	column	カラム数 1行のカラム数を指定してください。 有効範囲は、2～20の偶数です。

moduleSize　　バーコードの幅
利用可能な定数は、「4.4.1⑮ モジュールサイズ（ModuleSize）」を参照してください。

エラー　　本メソッド呼び出し時にエラーが発生した場合は、PrinterExceptionをスローします。
エラーについての詳細は、「**4.4.5 PrinterExceptionクラス**」を参照してください。

説明　　本メソッドの実行前にEnterPageModeでページモードを開始してください。

注意　　バーコードの印字データは、その他の印字データと重ならないように展開してください。

参考　　バーコードサイズについての詳細は、「付録B バーコードサイズ一覧」を参照してください。

PrintPageModeAztecCode　　ページモードのAztec Codeの印字

Aztec Codeをページモードの印字領域に展開します。

対象　　プラットフォーム共通　　MP-B30L、MP-B21L

構文
```
public void PrintPageModeAztecCode(int startX,
                                   int startY,
                                   string text,
                                   int layer,
                                   int errorCorrection,
                                   ModuleSize moduleSize,
                                   AztecSymbol aztecSymbol);
```

パラメータ　　startX　　始点からの横方向の基準点（ドット）
有効範囲は、0～2399です。

startY　　始点からの縦方向の基準点（ドット）
有効範囲は、0～2399です。

text　　バーコードデータ
データをUTF-8でバイナリにエンコードし、プリンターに送信します。
送信するデータが文字列で指定できない場合は、下記のエスケープシーケンスで指定できます。

エスケープシーケンス	説明
\u00nn	コントロールコード（nnは16進数で指定）
\\	バックスラッシュ

FNC1を指定するには、データに"\u001b\u0030"を入力してください。
1Bhを指定するには、データに"\u001b\u001b"を入力してください。

layer　　レイヤー数
フルレンジモードの有効範囲は、0、4～32です。
コンパクトモードの有効範囲は、0、1～4です。
0を指定した場合はレイヤーが自動設定されます。

errorCorrection
エラー訂正レベル(%)
有効範囲は、0、5～95です。
0を指定した場合は23%になります。

moduleSize　モジュールサイズ
利用可能な定数は、「4.4.1⑮ モジュールサイズ（ModuleSize）」を参照してください。

aztecSymbol　Aztecシンボル
利用可能な定数は、「4.4.1㉖ Aztecシンボル（AztecSymbol）」を参照してください。

エラー　本メソッド呼び出し時にエラーが発生した場合は、PrinterExceptionをスローします。
エラーについての詳細は、「**4.4.5 PrinterExceptionクラス**」を参照してください。

説明　本メソッドの実行前にEnterPageModeでページモードを開始してください。

注意　バーコードの印字データは、その他の印字データと重ならないように展開してください。

参考　バーコードサイズについての詳細は、「付録B バーコードサイズ一覧」を参照してください

SendPageModeBinary　ページモードのバイナリデータの送信

バイナリデータをページモードの印字領域に展開します。

対象　プラットフォーム共通　RP-F10、RP-G10、RP-E10、RP-D10、MP-B30、MP-B30L、MP-B21L、MP-A40

構文
```
public void SendPageModeBinary(byte[] data);
```

パラメータ　data　バイナリデータ
一度に指定可能なデータサイズは3MB（3145728バイト）です。

エラー　本メソッド呼び出し時にエラーが発生した場合は、PrinterExceptionをスローします。
エラーについての詳細は、「**4.4.5 PrinterExceptionクラス**」を参照してください。

説明　本メソッドの実行前にEnterPageModeでページモードを開始してください。

本メソッドは、指定されたバイナリデータを変換せずにプリンターに送信します。

本メソッドでプリンターコマンドをバイナリデータとして送信することにより、ライブラリではサポートされていないプリンター機能を利用できます。

注意　送信するデータによっては期待しない動作をする場合があります。実機にて事前に確認してください。

PrintPageModeImageFile　　ページモードのイメージファイルの描画

イメージファイルをページモードの印字領域に展開します。

対象　　プラットフォーム共通　　RP-F10、RP-G10、RP-E10、RP-D10、MP-B30、MP-B30L、MP-B21L、MP-A40

構文

```
public void PrintPageModeImageFile(int startX,
                                   int startY,
                                   string fileName,
                                   Dithering dithering);
```

パラメータ

startX　　始点からの横方向の基準点(ドット)
有効範囲は、0～2399です。

startY　　始点からの縦方向の基準点(ドット)
有効範囲は、0～2399です。

fileName　　データのファイルパス
入力可能な形式について下記に説明します。

iOSの場合:
使用するファイルパス文字列

Androidの場合:

- Javaの標準的なクラス「java.io.File」が扱う絶対パス文字列
  Android 10 (API 29)以降をアプリのターゲットとする場合には、一部のファイルが直接扱えなくなりますのでご注意ください。
  詳細については、「3.4 注意事項 - 対象範囲別ストレージについて」を参照してください。

- Androidに用意されたクラス「android.net.Uri」が扱う下記スキーム名のURI文字列
  ・file://
  ・content://
  本パラメータには"ストレージ アクセス フレームワーク"から取得したURI文字列を指定する必要があります。"ストレージ アクセス フレームワーク"から取得せずに作成されたURIはファイルを開けない可能性がありますのでご注意ください。

指定可能なファイルサイズは、最大16MB(16777216バイト)です。
送信可能なイメージファイルは.bmp、.jpg、.jpeg、.pngです。
カラーイメージの場合は、2値化処理によりモノクロイメージに変換して登録します。

dithering　　ディザリング
利用可能な定数は、「4.4.1⑥ ディザリング(Dithering)」を参照してください。

エラー　　本メソッド呼び出し時にエラーが発生した場合は、PrinterExceptionをスローします。
エラーについての詳細は、「**4.4.5 PrinterExceptionクラス**」を参照してください。

説明　　本メソッドの実行前にEnterPageModeでページモードを開始してください。

PrintPageModeRectangle　　ページモードの矩形の描画

ページモードの印字領域に矩形の描画を展開します。

対象　　プラットフォーム共通　　RP-F10、RP-G10、RP-E10、RP-D10、MP-B30、MP-B30L、MP-B21L、MP-A40

構文

```
public void PrintPageModeRectangle(int startX,
                                   int startY,
                                   int endX,
                                   int endY,
                                   LineStyle lineStyle);
```

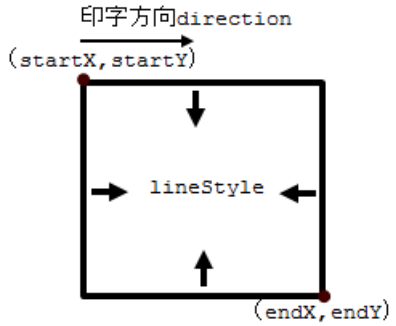
パラメータ

startX	始点からの横方向の描画開始位置(ドット) 有効範囲は、0～2399です。
startY	始点からの縦方向の描画開始位置(ドット) 有効範囲は、0～2399です。
endX	始点からの横方向の描画終了位置(ドット) 有効範囲は、0～2399です。
endY	始点からの縦方向の描画終了位置(ドット) 有効範囲は、0～2399です。
lineStyle	線種類 利用可能な定数は、「4.4.1㉜ 線種類(LineStyle)」を参照してください。

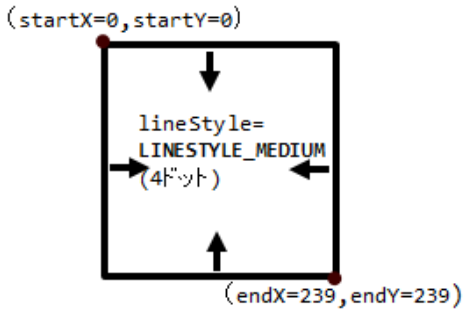
エラー　　本メソッド呼び出し時にエラーが発生した場合は、PrinterExceptionをスローします。
エラーについての詳細は、「**4.4.5 PrinterExceptionクラス**」を参照してください。

説明　　本メソッドの実行前にEnterPageModeでページモードを開始してください。

矩形はSetPageModeDirectionのdirectionに対して下記のように展開します。



描画イメージに対するパラメータの設定例を下記に示します。
例:始点から240ドット(30mm)、線種類を中太実線(4ドット)の正四角形を描画する場合

描画イメージ	パラメータ
	startX 0 startY 0 endX 239 endY 239 lineStyle **LINESTYLE_MEDIUM**

PrintPageModeLine　ページモードの罫線の印字

ページモードの印字領域に罫線を展開します。

対象　プラットフォーム共通　RP-F10、RP-G10、RP-E10、RP-D10、MP-B30、MP-B30L、MP-B21L、MP-A40

構文
```
public void PrintPageModeLine(int startX,
                              int startY,
                              int endX,
                              int endY,
                              LineStyle lineStyle);
```

パラメータ

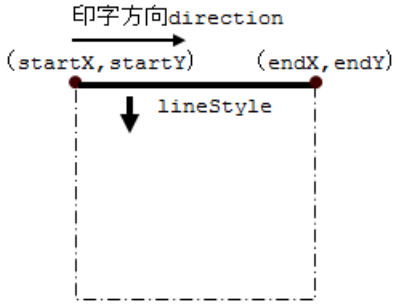
startX	始点からの横方向の罫線開始位置(ドット) 有効範囲は、0～2399です。
startY	始点からの縦方向の罫線開始位置(ドット) 有効範囲は、0～2399です。
endX	始点からの横方向の罫線終了位置(ドット) 有効範囲は、0～2399です。
endY	始点からの縦方向の罫線終了位置(ドット) 有効範囲は、0～2399です。
lineStyle	線種類 利用可能な定数は、「4.4.1㉜ 線種類(LineStyle)」を参照してください。

エラー　本メソッド呼び出し時にエラーが発生した場合は、PrinterExceptionをスローします。
エラーについての詳細は、「**4.4.5 PrinterExceptionクラス**」を参照してください。

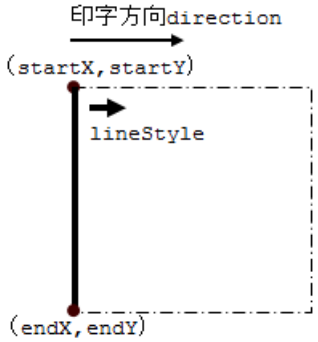
説明　本メソッドの実行前にEnterPageModeでページモードを開始してください。

本メソッドでは、斜線は描画できません。

罫線はSetPageModeDirectionのdirectionに対して下記のように展開します。



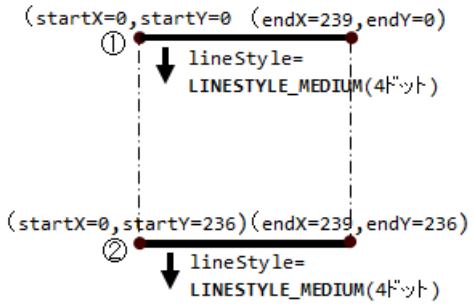
横罫線の展開方向



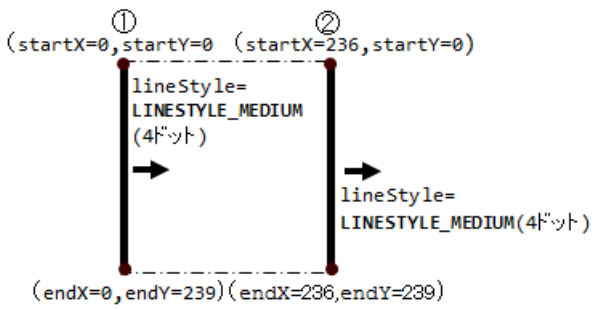
縦罫線の展開方向

描画イメージに対するパラメータの設定例を下記に示します。

例：始点から240ドット（30mm）、線種類を中太実線（4ドット）の正四角形の横罫線を描画する場合

描画イメージ	パラメータ
	① startX 0 startY 0 endX 239 endY 0 lineStyle **LINESTYLE_MEDIUM** ② startX 0 startY 236 endX 239 endY 236 lineStyle **LINESTYLE_MEDIUM**

例：始点から240ドット（30mm）、線種類を中太実線（4ドット）の正四角形の縦罫線を描画する場合

描画イメージ	パラメータ
	① startX 0 startY 0 endX 0 endY 239 lineStyle **LINESTYLE_MEDIUM** ② startX 236 startY 0 endX 236 endY 239 lineStyle **LINESTYLE_MEDIUM**

PrintPageModeLogo　　ページモードのロゴの印字

登録済みのロゴをページモードの印字領域に展開します。

対象　　プラットフォーム共通　　RP-F10、RP-G10、RP-E10、RP-D10、MP-B30、MP-B30L、MP-B21L、MP-A40

構文

```
public void PrintPageModeLogo(int startX,
                              int startY,
                              string logoId);
```

パラメータ

startX	始点からの横方向の基準点(ドット) 有効範囲は、0～2399です。
startY	始点からの縦方向の基準点(ドット) 有効範囲は、0～2399です。
logoId	印字するロゴのID(キーコード) 登録済みのロゴのIDを文字列で指定してください。

エラー　　本メソッド呼び出し時にエラーが発生した場合は、PrinterExceptionをスローします。
エラーについての詳細は、「**4.4.5 PrinterExceptionクラス**」を参照してください。

説明　　本メソッドの実行前にEnterPageModeでページモードを開始してください。

(5) スタンダードモード・ページモード共通プロパティ詳細

SendTimeout　送信タイムアウト時間の設定/取得

データ送信時のタイムアウト時間を設定または取得します。

対象　プラットフォーム共通　全デバイス

構文　`public int SendTimeout {get; set;}`

有効範囲　iOSの場合：
100～300000（ミリ秒：ms）
100未満の値を指定した場合は、100msに設定されます。
300000を超える値を指定した場合は、300000msに設定されます。
Androidの場合：
100～90000（ミリ秒：ms）
有効範囲外の値を指定した場合は、10000msに設定されます。

初期値　10000

説明　本プロパティはIsConnectがtrue/falseに関わらず設定または取得が可能です。

設定したタイムアウト時間が有効になるのは、次回データ送信時となります。

ReceiveTimeout　受信タイムアウト時間の設定/取得

データ受信時のタイムアウト時間を設定または取得します。

対象　プラットフォーム共通　全デバイス

構文　`public int ReceiveTimeout {get; set;}`

有効範囲　iOSの場合：
100～300000（ミリ秒：ms）
100未満の値を指定した場合は、100msに設定されます。
300000を超える値を指定した場合は、300000msに設定されます。
Androidの場合：
100～90000（ミリ秒：ms）
有効範囲外の値を指定した場合は、10000msに設定されます。

初期値　10000

説明　本プロパティはIsConnectがtrue/falseに関わらず設定または取得が可能です。

設定したタイムアウト時間が有効になるのは、次回データ受信時となります。

InternationalCharacter　国際文字セットの設定/取得

国際文字セットの値を設定または取得します。

対象　プラットフォーム共通　全デバイス

構文　public InternationalCharacter InternationalCharacter {get; set;}

説明　利用可能な定数は、「4.4.1㉝ 国際文字セット(InternationalCharacter)」を参照してください。無効な値を指定した場合は無視されます。

本プロパティが未設定の場合は、デバイスの言語設定により下記の国際文字セットとなります。
デバイスの言語設定が日本の場合:　COUNTRY_JAPAN
デバイスの言語設定が日本以外の場合:　COUNTRY_USA

SendText、SendTextEx、SendDataFile、PrintPageModeText及びPrintPageModeTextExにてテキストデータを送信した場合は、下記に示す文字コードの印字結果が異なります。印刷される文字に関しては、「付録A 文字セット」を参照してください。

国際文字セット設定により印字結果が異なる文字コード
0x23、0x24、0x40、0x5B、0x5C、0x5D、0x5E、0x60、0x7B、0x7C、0x7D、0x7E

CodePage　コードページの設定/取得

コードページの値を設定または取得します。

対象　プラットフォーム共通　全デバイス

構文　public CodePage CodePage {get; set;}

説明　利用可能な定数は、「4.4.1㉞ コードページ(CodePage)」を参照してください。無効な値を指定した場合は無視されます。

本プロパティが未設定の場合は、デバイスの言語設定により下記のコードページとなります。
デバイスの言語設定が日本の場合:　CODE_PAGE_KATAKANA
デバイスの言語設定が日本以外の場合:　CODE_PAGE_1252

SendText、SendTextEx、SendDataFile、PrintPageModeText及びPrintPageModeTextExにてテキストデータを送信する場合に利用するエンコーダが変更されます。印刷される文字に関しては、「付録A 文字セット」を参照してください。

DeviceModel　デバイスモデルの取得

通信中のプリンターのモデル値を取得します。

対象　プラットフォーム共通　全デバイス

構文　public DeviceModel DeviceModel {get;}

初期値　DeviceModel.DEVICE_MODEL_RP_E10

戻り値　「4.4.1① デバイスモデル（DeviceModel）」を参照してください。

PortType　接続中ポート種別の取得

プリンターとの接続に利用しているポート種別の値を取得します。

対象　プラットフォーム共通　全プリンター

構文　`public PortType PortType {get;}`

初期値　PortType.PORT_TYPE_BLUETOOTH

戻り値　「4.4.1② ポート種別（PortType）」を参照してください。

IsConnect　プリンターとの通信状態の確認

プリンターとの通信状態を確認します。

対象　プラットフォーム共通　全デバイス

構文　`public bool IsConnect {get;}`

戻り値　true　プリンターと通信中
false　プリンターと未接続

説明　本プロパティは、Connect状態をBOOL値で保持します。
Connectに成功すると、本プロパティはtrueになります。Connect後、Disconnectに成功すると、本プロパティはfalseになります。

SocketKeepingTime　ソケット維持時間の設定/取得

ソケット維持時間の設定または取得します。

対象　プラットフォーム共通　SLP720RT、SLP721RT、RP-F10、RP-G10、RP-E10、RP-D10、MP-B30、MP-B30L、MP-B21L、MP-A40

構文　`public int SocketKeepingTime {get; set;}`

有効範囲　60000～300000（ミリ秒：ms）
60000未満の値を指定した場合は、60000msに設定されます。
300000を超える値を指定した場合は、300000msに設定されます。

初期値　300000

説明　本プロパティはIsConnectがtrue/falseに関わらず設定または取得が可能です。

ソケット維持時間には、接続するプリンターの受信タイムアウト時間設定と等しい時間を指定してください。受信タイムアウト時間設定は、iOSの場合はApp StoreのiOSアプリ「SII Printer Utility」、Androidの場合はGoogle PlayのAndroidアプリ「SII Printer Utility」で変更できます。

設定したソケット維持時間が有効になるのは、次回Connect実行時となります。

Version　SDKバージョンの取得

SDKのバージョンを文字列で取得します。

対象　プラットフォーム共通　全デバイス

構文　`public string Version {get;}`

戻り値　2つのSDKバージョン文字列を返します。(例:"1.0.0 / 1.0.0")
1つ目:SII SDK for .NET MAUIのバージョン
2つ目:動作環境に対応したSII Print Class Libraryのバージョン

説明　本メソッドはIsConnectがtrue/falseに関わらず取得が可能です。

PrintSmartLabelMode　ラベルファイル印刷時の用紙の設定/取得

ラベルファイル印刷時の用紙を設定または取得します。

対象　プラットフォーム共通　SLP720RT、SLP721RT、MP-B30L、MP-B21L

構文　`public PrintSmartLabelMode PrintSmartLabelMode {get; set;}`

説明　設定可能な定数は、
「4.4.1ラベルファイル印刷時マークあり・なし用紙選択(PrintSmartLabelMode)」を参照してください。無効な値を設定した場合は無視されます。
本プロパティが未設定の場合は、PRINTSMARTLABEL_MODE_MARKとなります。

本プロパティはIsConnectがtrue/falseに関わらず設定または取得が可能です。

マーク紙を指定した場合、PrintSmartLabelImageData実行時に印字開始位置への紙送りを行います。

マークなし用紙を指定した場合、PrintSmartLabelImageData実行時に印字開始位置への紙送りは行いません。

4.4.3 DeviceInfoクラス

プリンターの探索メソッドで発見されたデバイス情報を格納するクラスです。取得できるデバイス情報については、「4.4.3(2) プロパティー覧」を参照してください。

(1) メソッド一覧

DeviceInfoクラスで提供されるメソッドの一覧を下記に示します。

名前	説明
DeviceInfo	デバイス情報クラスのコンストラクタ

(2) プロパティー覧

DeviceInfoクラスで提供されるプロパティの一覧を下記に示します。

名前	アクセス	説明
PortType	R	通信ポート種別の取得
DeviceName	R	デバイス名（プリンターモデル名）の取得
BluetoothAddress	R	Bluetoothアドレスの取得
MacAddress	R	MACアドレスの取得
IpAddress	R	IPアドレスの取得
IsBonded	R	ペアリング状態の取得
DevicePath	R	デバイスパスの取得

(3) メソッド詳細

DeviceInfo　コンストラクタ

構文(a)は、Bluetooth接続で使用するデバイス情報クラスのコンストラクタです。
構文(b)は、TCP/IP接続で使用するデバイス情報クラスのコンストラクタです。
構文(c)は、USB接続で使用するデバイス情報クラスのコンストラクタです。

対象　プラットフォーム共通　全デバイス

構文

(a)
```
public DeviceInfo(PortType portType,
                  string deviceName,
                  string bluetoothAddress,
                  bool isBonded);
```

(b)
```
public DeviceInfo(PortType portType,
                  string deviceName,
                  string macAddress,
                  string ipAddress);
```

(c)
```
public DeviceInfo(PortType portType,
                  string deviceName,
                  string devicePath);
```

説明　本メソッドは、StartDiscoveryDeviceで発見されたデバイス情報を格納します。

(4) プロパティ詳細

PortType　通信ポート種別の取得

対象　プラットフォーム共通　全デバイス

構文　public PortType PortType {get;}

説明　本プロパティは、StartDiscoveryDeviceで発見されたデバイス情報から通信ポート種別を取得します。

DeviceName　デバイス名（プリンターモデル名）の取得

対象　プラットフォーム共通　全デバイス

構文　public string DeviceName {get;}

説明　本プロパティは、StartDiscoveryDeviceで発見されたデバイス情報からデバイス名（プリンターモデル名）を取得します。

BluetoothAddress　Bluetoothアドレスの取得

対象　プラットフォーム共通　全プリンター

構文　public string BluetoothAddress {get;}

説明　本プロパティは、StartDiscoveryDeviceで発見されたプリンター情報からBluetoothアドレスを取得します。

MacAddress　MACアドレスの取得

対象　プラットフォーム共通　全プリンター

構文　public string MacAddress {get;}

説明　本プロパティは、StartDiscoveryDeviceで発見されたデバイス情報からMACアドレスを取得します。

IpAddress　IPアドレスの取得

対象　プラットフォーム共通　全プリンター

構文　public string IpAddress {get;}

説明　本プロパティは、StartDiscoveryDeviceで発見されたデバイス情報からIPアドレスを取得します。

IsBonded　ペアリング状態の取得

対象　プラットフォーム共通　全プリンター

構文　public bool IsBonded {get;}

説明　本プロパティは、StartDiscoveryDeviceで発見されたデバイス情報からペアリング状態を取得します。

iOSでは、常にtrueを返します。

DevicePath　デバイスパスの取得

対象　Android　全プリンター

構文　public string DevicePath {get;}

説明　本プロパティは、StartDiscoveryDeviceで発見されたデバイス情報からUSBデバイスファイルのパスの文字列を取得します。

4.4.4 PrinterIdentifierクラス

PrinterIdentifierクラスは、GetPrinterResponseで取得されたプリンターのプリンタIDを格納するクラスです。

(1) プロパティー覧

PrinterIdentifierクラスで提供されるプロパティの一覧を下記に示します。

名前	アクセス	説明
PrinterModelID	R	機種IDの取得
TypeID	R	タイプIDの取得
RomVersionID	R	ROMバージョンIDの取得
MainFirmwareVersion	R	ファームウェアバージョン(main)の取得
Manufacturer	R	メーカ名の取得
ModelName	R	モデル名の取得
SerialNumber	R	シリアル番号の取得
BootFirmwareVersion	R	ファームウェアバージョン(boot)の取得
BootFirmwareChecksum	R	ファームウェアチェックサム(boot)の取得
MainFirmwareChecksum	R	ファームウェアチェックサム(main)の取得
FirmwareChecksum	R	ファームウェアチェックサム(main + boot)の取得

プロパティ詳細

PrinterModelID　機種IDの取得

対象　プラットフォーム共通　全プリンター

構文　`public byte PrinterModelID {get;}`

説明　本プロパティは、GetPrinterResponseで取得されたプリンターのプリンタIDから機種IDを取得します。

TypeID　タイプIDの取得

対象　プラットフォーム共通　全プリンター

構文　`public byte TypeID {get;}`

説明　本プロパティは、GetPrinterResponseで取得されたプリンターのプリンタIDからタイプIDを取得します。

RomVersionID　ROMバージョンIDの取得

対象　プラットフォーム共通　全プリンター

構文　public byte RomVersionID {get;}

説明　本プロパティは、GetPrinterResponseで取得されたプリンターのプリンタIDからROMバージョンIDを取得します。

MainFirmwareVersion　ファームウェアバージョン(main)の取得

対象　プラットフォーム共通　全プリンター

構文　public String MainFirmwareVersion {get;}

説明　本プロパティは、GetPrinterResponseで取得されたプリンターのプリンタIDからファームウェアバージョン(main)の文字列を取得します。

Manufacturer　メーカ名の取得

対象　プラットフォーム共通　全プリンター

構文　public String Manufacturer {get;}

説明　本プロパティは、GetPrinterResponseで取得されたプリンターのプリンタIDからメーカ名の文字列を取得します。

ModelName　モデル名の取得

対象　プラットフォーム共通　全プリンター

構文　public String ModelName {get;}

説明　本プロパティは、GetPrinterResponseで取得されたプリンターのプリンタIDからモデル名の文字列を取得します。

SerialNumber　シリアル番号の取得

対象　プラットフォーム共通　全プリンター

構文　public String SerialNumber {get;}

説明　本プロパティは、GetPrinterResponseで取得されたプリンターのプリンタIDからシリアル番号の文字列を取得します。

BootFirmwareVersion　ファームウェアバージョン(boot)の取得

対象　プラットフォーム共通　全プリンター

構文　public String BootFirmwareVersion {get;}

説明　本プロパティは、GetPrinterResponseで取得されたプリンターのプリンタIDからファームウェアバージョン(boot)の文字列を取得します。

BootFirmwareChecksum　ファームウェアチェックサム(boot)の取得

対象　プラットフォーム共通　全プリンター

構文　public byte[] BootFirmwareChecksum {get;}

説明　本プロパティは、GetPrinterResponseで取得されたプリンターのプリンタIDからファームウェアチェックサム(boot)を取得します。

MainFirmwareChecksum　ファームウェアチェックサム(main)の取得

対象　プラットフォーム共通　全プリンター

構文　public byte[] MainFirmwareChecksum {get;}

説明　本プロパティは、GetPrinterResponseで取得されたプリンターのプリンタIDからファームウェアチェックサム(main)を取得します。

FirmwareChecksum　ファームウェアチェックサム(main + boot)の取得

対象　プラットフォーム共通　全プリンター

構文　public byte[] FirmwareChecksum {get;}

説明　本プロパティは、GetPrinterResponseで取得されたプリンターのプリンタIDからファームウェアチェックサム(main + boot)を取得します。

4.4.5 PrinterExceptionクラス

(1) メソッド一覧

PrinterExceptionクラスで提供されるメソッドの一覧を下記に示します。

名前	説明
PrinterException	コンストラクタ

(2) プロパティ一覧

PrinterExceptionクラスで提供されるプロパティの一覧を下記に示します。

名前	アクセス	説明
HResult	R	エラーコードの取得
Message	R	エラーメッセージの取得

(3) 定数一覧

① エラーコード

エラーコードの取得で利用される定数を下記に示します。

定数名	説明	値
ERROR_ACCESS_DENIED	ハンドルの取得に失敗しました。*1	-1
	利用できないポートが指定されました。	
	サポートされていないメソッドが指定されました。	
ERROR_SHARING_VIOLATION	すでにオープン済みのポートが指定されました。	-11
ERROR_PORT_NOT_OPENED	ポートがオープンされていません。	-12
ERROR_DEVICE_NOT_CONNECTED	デバイスとプリンターの接続に問題があります。	-21
ERROR_OFFLINE	切断状態またはプリンターがオフラインです。	-22
ERROR_EXTERNAL_DEVICE_NOT_CONNECTED	ディスプレイが接続されていません。	-23
ERROR_DEVICE_INITIALIZE_FAILED	プリンターの設定変更に失敗しました。プリンターへのデータ送信が送信タイムアウト時間内に完了していないか、プリンターからのデータ受信が受信タイムアウト時間内に完了していません。	-31
ERROR_DATA_SIZE_ZERO	データサイズが0バイトのデータが指定されました。	-101
ERROR_OVER_MAX_DATA_SIZE	データサイズが最大値を超過しています。	-102
ERROR_DATA_SIZE_INVALID	既定のデータサイズではありません。	-103

定数名	説明	値
ERROR_ENCODE_FAILED	テキストデータのエンコードでエラーが発生しました。*1	-111
ERROR_TIMEOUT	送信タイムアウトが発生しました。 受信タイムアウトが発生しました。	-201
ERROR_FILE_NOT_FOUND	指定されたファイルが存在しません。	-301
ERROR_FILE_USED	指定されたファイルは別プロセスで使用中です。	-302
ERROR_FILE_INVALID	指定されたファイルは無効です。	-303
ERROR_LOW_MEMORY	イメージファイルの読み込みでメモリ不足が発生しました。	-311
ERROR_OVER_MAX_IMAGE	イメージファイルの幅と高さのいずれか、または両方が印字可能な最大ドット数を超過しています。	-312
ERROR_LOGO_NOT_DEFINED	ロゴが登録されていません。	-313
ERROR_LOW_USER_AREA	ユーザ領域の残り容量が不足しています。	-401
ERROR_LOW_EXTERNAL_RAM	RAMの残り容量が不足しています。	-402
ERROR_NOT_REGISTERD	テンプレートが登録されていません。 イメージファイルが登録されていません。 スライドデータが登録されていません。 オプションフォントが登録されていません。 外字が登録されていません。	-403
ERROR_NOT_UNREGISTERD	テンプレートが削除されていません。 イメージファイルが削除されていません。 スライドデータが削除されていません。 オプションフォントが削除されていません。 外字が削除されていません。	-404
ERROR_INVALID_NO	指定されたロゴID、またはスタイルシートの値は無効です。	-501
ERROR_OVER_STYLE_NUM	指定されたファイルに登録されているスタイルの数が規定値(64)を超過しています。	-502
ERROR_INVALID_DATA	指定されたデータは無効です。	-503
ERROR_PAGE_MODE_SPECIFIED	ページモードが指定されています。	-511
ERROR_PAGE_MODE_NOT_SPECIFIED	ページモードが指定されていません。	-512
ERROR_LABEL_FILE_NOT_SELECTED	ラベルファイルが指定されていません。	-521
ERROR_GET_LABEL_IMAGE	ラベルイメージの作成に失敗しました。	-522
ERROR_INVALID_PARAM	指定されたパラメータは無効です。	-9999

*1: 異常処理の可能性があります。

(4) メソッド詳細

PrinterException　コンストラクタ

PrinterManagerクラスのAPI呼び出し時にスローされる例外クラスです。

構文　　public PrinterException(int code, string message);

(5) プロパティ詳細

HResult　エラーコードの取得

対象　　プラットフォーム共通　　全プリンター

構文　　public int HResult {get;}

説明　　本プロパティは、エラーコードをint型で取得します。

戻り値　　「4.4.5(3) 定数一覧」を参照してください。

Message　エラーメッセージの取得

対象　　プラットフォーム共通　　全プリンター

構文　　public string Message {get;}

説明　　本プロパティは、HResultの内容を補足する文字列が取得できます。

4.4.6 SmartLabelManagerクラス

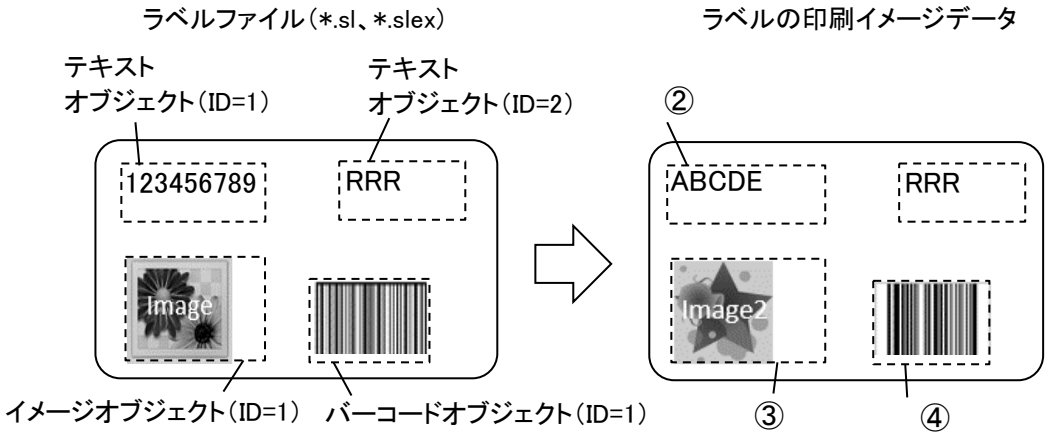
SmartLabelManagerクラスは、アプリで作成したラベルファイル(*.sl、*.slex)を、プリンターで印字可能なデータに変換する機能を提供するクラスです。

(1) メソッド一覧

SmartLabelManagerクラスで提供されるメソッドの一覧を下記に示します。

名前	説明
SmartLabelManager	コンストラクタ
SelectSmartLabelFile	ラベルファイルの指定
ReplaceSmartLabelTextData	ラベルのテキストデータの置換
ReplaceSmartLabelImageData	ラベルのイメージデータの置換
ReplaceSmartLabelBarcodeData	ラベルのバーコードデータの置換

ラベルファイルを利用してデータを置き換えし、印字する手順例を下記に示します。



① 印字またはデータの置き換えをするラベルファイルを指定します。

```
smartLabelManager.SelectSmartLabelFile(labelfilePath);
```

② テキストデータを置き換えます。

```
smartLabelManager.ReplaceSmartLabelTextData(1, "ABCDE");
```

③ イメージデータを置き換えます。

```
smartLabelManager.ReplaceSmartLabelImageData(1, image);
```

④ バーコードデータを置き換えます。

```
smartLabelManager.ReplaceSmartLabelBarcodeData(1, "123456789");
```

⑤ ラベルを印字します。

```
printerManager.PrintSmartLabelImageData(smartLabelManager);
```

(2) メソッド詳細

SmartLabelManager　コンストラクタ

SII.SDK.Maui.Printer.SmartLabelManagerクラスのコンストラクタです。

対象　プラットフォーム共通　SLP720RT、SLP721RT、MP-B30L、MP-B21L

構文　`public SmartLabelManager(Context context);`

パラメータ　`context`　コンテキスト

iOSの場合：

nullを指定してください。

Androidの場合：

本メソッドを呼び出すアプリケーションのコンテキストを指定してください。

SelectSmartLabelFile　ラベルファイルの指定

ラベルファイル（*.sl、*.slex）を指定します。

対象　プラットフォーム共通　SLP720RT、SLP721RT、MP-B30L、MP-B21L

構文　`public void SelectSmartLabelFile(string filePath);`

パラメータ　`filePath`　使用するラベルファイル（*.sl、*.slex）のファイルパス（*.slは非推奨）
入力可能な形式について下記に説明します。

iOSの場合：

使用するラベルファイル（*.sl、*.slex）名
アプリで作成したラベルファイル（*.sl、*.slex）を指定してください。

Androidの場合：

- Javaの標準的なクラス「java.io.File」が扱う絶対パス文字列
  Android 10 (API 29)以降をアプリのターゲットとする場合には、一部のファイルが直接扱えなくなりますのでご注意ください。
  詳細については、「3.4 注意事項 - 対象範囲別ストレージについて」を参照してください。

- Androidに用意されたクラス「android.net.Uri」が扱う下記スキーム名のURI文字列
  ・file://
  ・content://
  本パラメータには"ストレージ アクセス フレームワーク"から取得したURI文字列を指定する必要があります。"ストレージ アクセス フレームワーク"から取得せずに作成されたURIはファイルを開けない可能性がありますのでご注意ください。

エラー　本メソッド呼び出し時にエラーが発生した場合は、PrinterExceptionをスローします。
エラーについての詳細は、「**4.4.5 PrinterExceptionクラス**」を参照してください。

説明　　指定したラベルファイル（*.sl、*.slex）を内部で保持します。
ラベルファイル指定後は、各オブジェクトのデータの置き換えができます。

使用できるラベルファイルには制限があります。制限については、「4.2.1 ラベルファイルの構造」を参照してください。

注意　　ラベルファイル（*.sl）は、将来的に非サポートとなるラベルファイルです。
SII Layout Editor（アプリバージョン1.5.0以降）で作成できるラベルファイル（*.slex）をお使いください。

ReplaceSmartLabelTextData　　ラベルのテキストデータの置換

ラベルファイル（*.sl、*.slex）のテキストオブジェクトの値を置き換えます。

対象　　プラットフォーム共通　　SLP720RT、SLP721RT、MP-B30L、MP-B21L

構文　　`public void ReplaceSmartLabelTextData(int mapID, string text);`

パラメータ　`mapID`　　テキストオブジェクトのID
アプリのラベルファイル（*.sl、*.slex）に配置したテキストオブジェクトのIDを指定してください。テキストオブジェクトのIDはアプリのUI画面で確認できます。
指定したmapIDが選択中のラベルに定義されていない場合は無視されます。



アプリのUI画面

`text`　　置き換えるテキストデータ

エラー　　本メソッド呼び出し時にエラーが発生した場合は、PrinterExceptionをスローします。
エラーについての詳細は、「**4.4.5 PrinterExceptionクラス**」を参照してください。

ReplaceSmartLabelImageData　　ラベルのイメージデータの置換

ラベルファイル(*.sl、*.slex)のイメージオブジェクトの値を置き換えます。

対象　　プラットフォーム共通　　SLP720RT、SLP721RT、MP-B30L、MP-B21L

構文　　`public void ReplaceSmartLabelImageData(int mapID, object image);`

パラメータ　　mapID　　イメージオブジェクトのID

アプリのラベルファイル(*.sl、*.slex)に配置したイメージオブジェクトのIDを指定してください。イメージオブジェクトのIDはアプリのUI画面で確認できます。

指定したmapIDが選択中のラベルに定義されていない場合は無視されます。



アプリのUI画面

image　　置き換えるイメージデータ

iOSの場合：

iOSに用意されたクラス「UIKit.UIImage」に準するイメージデータを指定してください。

Androidの場合：

Androidに用意されたクラス「Android.Graphics.Bitmap」に準するイメージデータを指定してください。

エラー　　本メソッド呼び出し時にエラーが発生した場合は、PrinterExceptionをスローします。
エラーについての詳細は、「**4.4.5 PrinterExceptionクラス**」を参照してください。

ReplaceSmartLabelBarcodeData　ラベルのバーコードデータの置換

ラベルファイル(*.sl、*.slex)のバーコードオブジェクトの値を置き換えます。

対象　プラットフォーム共通　SLP720RT、SLP721RT、MP-B30L、MP-B21L

構文　`public void ReplaceSmartLabelBarcodeData(int mapID, string text);`

パラメータ　`mapID`　バーコードオブジェクトのID
アプリのラベルファイル(*.sl、*.slex)に配置したバーコードオブジェクトのIDを指定してください。バーコードオブジェクトのIDはアプリのUI画面で確認できます。
指定したmapIDが選択中のラベルに定義されていない場合は無視されます。



アプリのUI画面

`text`　置き換えるテキストデータ
置き換えるテキストデータが無効なバーコードデータの場合でも、エラーにはなりません。有効なバーコードデータをご確認の上、指定してください。

エラー　本メソッド呼び出し時にエラーが発生した場合は、PrinterExceptionをスローします。
エラーについての詳細は、「**4.4.5 PrinterExceptionクラス**」を参照してください。

5章　ディスプレイのライブラリ機能

本章では、ライブラリに実装されている各クラスのディスプレイ用APIについて説明します。

5.1　APIリファレンス

ライブラリには、下記のクラスが含まれます。

✓：サポート、－：非サポート

名前	説明	サポート
PrinterManager	デバイスとの通信、表示を行うためのAPIを提供します。 「**5.1.2 PrinterManagerクラス**」を参照してください。 PrinterManagerクラスで提供するAPIで必要な列挙型定数については、「**5.1.1 列挙型定数一覧**」を参照してください。	✓
DeviceInfo	StartDiscoveryDeviceで発見されたデバイス情報を格納します。 「**5.1.3 DeviceInfoクラス**」を参照してください。 DeviceInfoクラスで提供するAPIで必要な列挙型定数については、「**5.1.1 列挙型定数一覧**」を参照してください。	✓
PrinterIdentifier	GetPrinterResponseで取得されたプリンターのプリンタIDを格納します。 「**5.1.4** PrinterIdentifierクラス」を参照してください。 PrinterIdentifierクラスで提供するAPIで必要な列挙型定数については、「**5.1.1 列挙型定数一覧**」を参照してください。	✓
PrinterException	API呼び出し時にスローされる例外クラスです。 「**5.1.5 PrinterExceptionクラス**」を参照してください。	✓

（注意）「はじめに」の「対象デバイス」を参照し、プラットフォーム別でのデバイスの組み合わせをご確認ください。

各クラスにおけるメソッド及びプロパティの詳細説明について記載内容を説明します。

Method or Property　メソッド名またはプロパティ名

メソッドまたはプロパティの概略を記載します。

対象　メソッドをサポートするプラットフォーム、及びデバイスを記載します。

- プラットフォーム

	全てのプラットフォームを サポートする場合	サポートするデバイスが 個別の場合
「対象」の記載	プラットフォーム共通	iOS
		Android

- デバイス

	複数のデバイスを サポートする場合		サポートするデバイスが 個別の場合
「対象」の記載	全デバイス	全プリンター	SLP720RT
			SLP721RT
			RP-F10
			RP-G10
			RP-E10
			RP-D10
			MP-B30
			MP-B30L
			MP-B21L
			MP-B20
			MP-A40
		－	プリンター経由のDSP-A01
			DSP-A01単体

構文　構文を記載します。

パラメータ　パラメータと、パラメータの説明を記載します。

有効範囲　有効範囲がある場合は、有効範囲を記載します。

エラー　エラーが発生した場合の説明を記載します。

初期値　初期値がある場合は、初期値を記載します。

説明　メソッド説明を記載します。

戻り値　戻り値がある場合は、戻り値を記載します。

5.1.1 列挙型定数一覧

列挙型定数一覧を下記に示します。

① デバイスモデル(DeviceModel)

デバイスモデルで利用する列挙型定数を下記に示します。

定数名	説明
DEVICE_MODEL_RP_F10[*1]	RP-F10
DEVICE_MODEL_DSP_A01	DSP-A01

*1: プリンター経由で使用している場合のみ有効。

② ポート種別(PortType)

ポート種別で利用する列挙型定数を下記に示します。

定数名	説明
PORT_TYPE_BLUETOOTH[*1]	Bluetoothで接続
PORT_TYPE_USB	USBで接続
PORT_TYPE_TCP[*1]	TCP/IPで接続

*1: Androidの場合で、プリンター経由で使用している場合のみ有効。

③ プリンター応答種別(PrinterResponseId)

プリンター応答種別で利用する列挙型定数を下記に示します。

定数名	説明
PRINTER_RESPONSE_REQUEST	実行応答リクエスト
PRINTER_RESPONSE_USER_AREA	ユーザ領域の残り容量の送信
PRINTER_RESPONSE_ARRANGE_USER_AREA	ユーザ領域の整理後の残り容量の送信
PRINTER_RESPONSE_NV_GRAPHICS	NVグラフィックスのメモリ容量の送信
PRINTER_RESPONSE_KEY_CODE	定義されているNVグラフィックスのキーコード一覧の送信
PRINTER_RESPONSE_MAINTENANCE_COUNTER	メンテナンスカウンタの送信
PRINTER_RESPONSE_PRINTER_IDENTIFIER	プリンタIDの送信

④　デバイスタイプ(DeviceType)

デバイスタイプで利用する列挙型定数を下記に示します。

定数名	説明
TYPE_PRINTER	プリンター
TYPE_DISPLAY	ディスプレイ

⑤　メモリ領域(MemoryArea)

メモリ領域の操作で利用する列挙型定数を下記に示します。

定数名	説明
MEMORY_DISPLAY_USERMEMORY	ユーザ領域

⑥　位置揃え(PrintAlignment)

位置揃えで利用する列挙型定数を下記に示します。

定数名	説明
ALIGNMENT_LEFT	左揃え
ALIGNMENT_CENTER	中央揃え
ALIGNMENT_RIGHT	右揃え

⑦　強調文字(CharacterBold)

強調文字で利用する列挙型定数を下記に示します。

定数名	説明
BOLD_CANCEL	強調文字を解除
BOLD	強調文字を指定

⑧　アンダーライン(CharacterUnderline)

アンダーラインで利用する列挙型定数を下記に示します。

定数名	説明
UNDERLINE_CANCEL	アンダーライン表示を解除
UNDERLINE_1	1ドット幅アンダーライン表示を指定

⑨　文字倍率(CharacterScale)

文字倍率で利用する列挙型定数を下記に示します。

定数名	説明
VERTICAL_1_HORIZONTAL_1	縦1倍・横1倍
VERTICAL_1_HORIZONTAL_2	縦1倍・横2倍
VERTICAL_1_HORIZONTAL_3	縦1倍・横3倍
VERTICAL_1_HORIZONTAL_4	縦1倍・横4倍
VERTICAL_2_HORIZONTAL_1	縦2倍・横1倍
VERTICAL_2_HORIZONTAL_2	縦2倍・横2倍
VERTICAL_2_HORIZONTAL_3	縦2倍・横3倍
VERTICAL_2_HORIZONTAL_4	縦2倍・横4倍
VERTICAL_3_HORIZONTAL_1	縦3倍・横1倍
VERTICAL_3_HORIZONTAL_2	縦3倍・横2倍
VERTICAL_3_HORIZONTAL_3	縦3倍・横3倍
VERTICAL_3_HORIZONTAL_4	縦3倍・横4倍
VERTICAL_4_HORIZONTAL_1	縦4倍・横1倍
VERTICAL_4_HORIZONTAL_2	縦4倍・横2倍
VERTICAL_4_HORIZONTAL_3	縦4倍・横3倍
VERTICAL_4_HORIZONTAL_4	縦4倍・横4倍

⑩　文字フォント(CharacterFont)

文字フォントで利用する列挙型定数を下記に示します。

定数名	説明
FONT_A	フォントA(24×12)
FONT_B	フォントB(16×8)

⑪　登録フォント(RegisteredFont)

登録フォントで利用する列挙型定数を下記に示します。

定数名	説明
FONT_STANDARD	標準フォント
FONT_OPTION	オプションフォント

⑫　モジュールサイズ(ModuleSize)

バーコードのモジュールサイズで利用する列挙型定数を下記に示します。

定数名	説明	使用メソッド
QR_MODULE_SIZE_2	2ドット	SetTemplateQRCodeData
QR_MODULE_SIZE_3	3ドット	
QR_MODULE_SIZE_4	4ドット	
QR_MODULE_SIZE_5	5ドット	
QR_MODULE_SIZE_6	6ドット	
QR_MODULE_SIZE_7	7ドット	
QR_MODULE_SIZE_8	8ドット	
QR_MODULE_SIZE_9	9ドット	
QR_MODULE_SIZE_10	10ドット	
QR_MODULE_SIZE_11	11ドット	

⑬　エラー訂正レベル(ErrorCorrection)

エラー訂正レベルで利用する列挙型定数を下記に示します。

定数名	説明	使用メソッド
QR_ERROR_CORRECTION_L	エラー訂正レベル L	SetTemplateQRCodeData
QR_ERROR_CORRECTION_M	エラー訂正レベル M	
QR_ERROR_CORRECTION_H	エラー訂正レベル H	
QR_ERROR_CORRECTION_Q	エラー訂正レベル Q	

⑭　QRデータモード(QrDataMode)

QRデータモードで利用する列挙型定数を下記に示します。

定数名	説明
QRDATAMODE_NUMERIC	数字モード
QRDATAMODE_ALPHANUMERIC	英数字モード
QRDATAMODE_8BITBYTE	8ビットバイトモード
QRDATAMODE_KANJI	漢字モード
QRDATAMODE_MIXTURE	混在モード

⑮　QRクワイエットゾーン(QrQuietZone)

QRクワイエットゾーンで利用する列挙型定数を下記に示します。

定数名	説明
QRQUIETZONE_EXIST	QRクワイエットゾーンあり
QRQUIETZONE_NONE	QRクワイエットゾーンなし

⑯　マクロ登録処理(MacroRegistrationFunction)

マクロ登録処理で利用する列挙型定数を下記に示します。

定数名	説明
MACRO_REGISTRATION_CLEAR	マクロ登録処理の中止
MACRO_REGISTRATION_START	マクロ登録処理の開始
MACRO_REGISTRATION_REGIST	マクロ登録とマクロ登録処理の終了

⑰　ディスプレイ応答種別(DisplayResponseId)

ディスプレイ応答種別で利用する列挙型定数を下記に示します。

定数名	説明
DISPLAY_RESPONSE_REQUEST	実行応答リクエスト
DISPLAY_RESPONSE_USER_AREA	ユーザ領域残り容量の送信
DISPLAY_RESPONSE_TEMPLATE_ID_LIST	テンプレートIDの送信
DISPLAY_RESPONSE_IMAGE_ID_LIST	イメージIDの送信
DISPLAY_RESPONSE_SLIDE_ID_LIST	スライドIDの送信
DISPLAY_RESPONSE_TEMPLATE_LABEL	テンプレート名の送信
DISPLAY_RESPONSE_IMAGE_LABEL	イメージ名の送信
DISPLAY_RESPONSE_SLIDE_LABEL	スライド名の送信

⑱　国際文字セット(InternationalCharacter)

国際文字セットで利用する列挙型定数を下記に示します。

定数名	説明
COUNTRY_USA	アメリカ(USA)
COUNTRY_FRANCE	フランス(France)
COUNTRY_GERMANY	ドイツ(Germany)
COUNTRY_ENGLAND	イギリス(United Kingdom)
COUNTRY_DENMARK_1	デンマークⅠ(Denmark I)
COUNTRY_SWEDEN	スウェーデン(Sweden)
COUNTRY_ITALY	イタリア(Italy)
COUNTRY_SPAIN	スペインⅠ(Spain I)
COUNTRY_JAPAN	日本(Japan)
COUNTRY_NORWAY	ノルウェー(Norway)
COUNTRY_DENMARK_2	デンマークⅡ(Denmark II)
COUNTRY_SPAIN_2	スペインⅡ(Spain II)
COUNTRY_LATIN_AMERICA	ラテンアメリカ(Latin America)
COUNTRY_ARABIA	アラビア(Arabia)

⑲　コードページ(CodePage)

コードページで利用する列挙型定数を下記に示します。

定数名	説明
CODE_PAGE_437	USA, Standard Europe (Code Page437)
CODE_PAGE_KATAKANA	Katakana
CODE_PAGE_850	Multilingual (Code Page850)
CODE_PAGE_860	Portuguese (Code Page860)
CODE_PAGE_863	Canadian-French (Code Page863)
CODE_PAGE_865	Nordic (Code Page865)
CODE_PAGE_857[*1]	Turkish (Code Page857)
CODE_PAGE_737	Greek (Code Page737)
CODE_PAGE_1252	Latin (Code Page1252)
CODE_PAGE_866	Russian (Code Page866)
CODE_PAGE_852	Eastern Europe (Code Page852)
CODE_PAGE_858	Euro (Code Page858)
CODE_PAGE_855	Cyrillic (Code Page855)
CODE_PAGE_864[*1*2]	Arabic (Code Page864)
CODE_PAGE_1250	Central European (Code Page1250)
CODE_PAGE_1251	Cyrillic (Code Page1251)
CODE_PAGE_1253[*3]	Greek (Code Page1253)
CODE_PAGE_1254	Turkish (Code Page1254)

*1：Unicodeの20AChは表示できません。
*2：フォントBは表示できません。
*3：Unicodeの00AAhは表示できません。

5.1.2 PrinterManagerクラス

(1) メソッド一覧

PrinterManagerクラスで提供されるメソッドを下記に示します。

メソッドの詳細については、「5.1.2(3) メソッド詳細」を参照してください。

✓:サポート、-:非サポート

名前	説明	サポート
PrinterManager	コンストラクタ	✓
Connect	デバイスとの通信開始	✓
Disconnect	デバイスとの通信切断	✓
GetPrinterResponse	プリンターからの各種応答取得	✓
StartDiscoveryDevice	デバイスの探索開始	✓
CancelDiscoveryDevice	デバイスの探索中断	✓
GetFoundDevice	発見されたデバイス情報リストの取得	✓
Defragment	メモリ領域の最適化	✓
InitializeMemoryArea	メモリ領域の初期化	✓
ShowTemplate	テンプレートの表示	✓
ShowSlide	スライドの表示	✓
EnterStandbyMode	スタンバイの表示	✓
ExecuteMacro	マクロの実行	✓
TurnOnScreen	画面の表示オン・オフ	✓
SelectTemplate	テンプレートの選択	✓
SetTemplateImageData	イメージデータの設定	✓
SelectTemplateTextObject	text要素の選択	✓
SetTemplateTextAlignment	テキストデータの位置揃え	✓
SetTemplateTextLeftMargin	テキストデータの左マージンの設定	✓
SetTemplateTextLineSpacing	テキストデータの改行量の設定	✓
SetTemplateTextBold	テキストデータの強調文字の設定	✓
SetTemplateTextUnderline	テキストデータのアンダーラインの設定	✓
SetTemplateTextSize	テキストデータの文字サイズの設定	✓
SetTemplateTextFont	テキストデータの文字フォントの設定	✓
SetTemplateTextRegisteredFont	テキストデータの登録フォントの設定	✓
SetTemplateTextRightSpacing	テキストデータの右スペース量の設定	✓
SetTemplateTextColor	テキストデータの文字色の設定	✓
SetTemplateTextData	テキストデータの入力	✓
SetTemplateBarcodeData	バーコードデータの入力	✓
SetTemplateQRCodeData	QRコードデータの入力	✓
RegisterTemplate	テンプレートの登録	✓
UnregisterTemplate	テンプレートの削除	✓
RegisterImageData	イメージデータの登録	✓
UnregisterImageData	イメージデータの削除	✓
RegisterSlideData	スライドデータの登録	✓

名前	説明	サポート
UnregisterSlideData	スライドデータの削除	✓
RegisterUserDefinedCharacter	外字の登録	✓
UnregisterUserDefinedCharacter	外字の削除	✓
RegisterOptionFont	オプションフォントの登録	✓
UnregisterOptionFont	オプションフォントの削除	✓
ControlMacroRegistration	マクロ登録の開始・終了	✓
GetDisplayResponse	ディスプレイからの各種応答取得	✓
SendBinary	バイナリデータの送信	✓
SendDataFile	指定ファイルの送信	✓

(2) プロパティー覧

PrinterManagerクラスで提供されるプロパティを下記に示します。

✓:サポート、-:非サポート

名前	アクセス	説明	サポート
SendTimeout	R/W	送信タイムアウト時間の設定/取得	✓
ReceiveTimeout	R/W	受信タイムアウト時間の設定/取得	✓
InternationalCharacter	R/W	国際文字セットの設定/取得	✓
CodePage	R/W	コードページの設定/取得	✓
DeviceModel	R	デバイスモデルの取得	✓
PortType	R	接続中ポート種別の取得	✓
IsConnect	R	プリンターとの通信状態の確認	✓
SocketKeepingTime	R/W	ソケット維持時間の設定/取得	✓
Version	R	SDKバージョンの取得	✓

(3) メソッド詳細

PrinterManager　コンストラクタ

SII.SDK.Maui.Printer.PrinterManagerクラスのコンストラクタです。

対象　プラットフォーム共通　全デバイス

構文
```
public PrinterManager(object context);
```

パラメータ　context　コンテキスト
iOSの場合：
nullを指定してください。
Androidの場合：
本メソッドを呼び出すアプリケーションのコンテキストを指定してください。

Connect　デバイスとの通信開始

デバイスとの通信を開始します。

対象　プラットフォーム共通　全デバイス

構文
(a)
```
public void Connect(DeviceModel deviceModel,
                    DeviceInfo deviceInfo);
```
(b)
```
public void Connect(DeviceModel deviceModel,
                    PortType portType,
                    string param);
```

パラメータ　deviceModel　デバイスモデル
利用可能な定数は、「5.1.1① デバイスモデル（DeviceModel）」を参照してください。

deviceInfo　デバイス情報
DeviceInfoクラスのオブジェクトを指定してください。
DeviceInfoクラスのオブジェクトは、StartDiscoveryDeviceとGetFoundDeviceにて取得が可能です。

portType　ポート種別
利用可能な定数は、「5.1.1② ポート種別（PortType）」を参照してください。

param　　portTypeの設定により異なります。
対応するインターフェイスは「はじめに」の「対象デバイス」を参照してください。
・portTypeが**PORT_TYPE_BLUETOOTH**
Bluetoothアドレス、またはBluetoothデバイス名（Bluetoothアクセサリ）を指定してください。
例："00:11:22:AA:BB:CC"、"RP-F10"
・portTypeが**PORT_TYPE_USB**
iOSの場合：
プリンタ名を指定してください。
例："RP-F10"
Androidの場合：
nullを指定してください。
・portTypeが**PORT_TYPE_TCP**
プリンターのIPアドレスを指定してください。
例："192.168.0.190"、"192.168.0.1"

エラー　　本メソッド呼び出し時にエラーが発生した場合は、PrinterExceptionをスローします。
エラーについての詳細は、「**5.1.5 PrinterExceptionクラス**」を参照してください。

説明　　本メソッドは、他のPrinterManagerクラスのメソッドを使用する前に呼び出してください。
本ライブラリを正しく動作させるために、本メソッドは接続時にプリンターの設定を変更する場合があります。

Bluetooth接続の場合：
プリンター経由でディスプレイを使用する場合のみ、paramで指定したペアリング済みのBluetoothアドレス、またはBluetoothデバイス（Bluetoothアクセサリ）に対して接続を行います。
iOSでは、iOSデバイスとペアリング済みのプリンターと、Bluetooth接続で通信を開始します。
Androidでは、paramで指定したBluetoothアドレスに対し、deviceModelで指定したプリンターを接続します。

USB接続の場合：
iOSでは、iOSデバイスとUSBケーブルで接続されたプリンターと、USB接続で通信を開始します。
Androidでは、プリンター経由でディスプレイを使用する場合は、deviceModelで指定したプリンターに接続します。ディスプレイ単体で使用する場合は、deviceModelで指定したディスプレイに接続します。

TCP/IP接続の場合：
プリンター経由でディスプレイを使用する場合のみ、デバイスと同じネットワークに接続されたプリンターと、TCP/IP接続で通信を開始します。
iOS、Androidのどちらもparamで指定されたIPアドレスに対し、deviceModelで指定したプリンターを接続します。

- **ライブラリのTCP/IP接続中のソケットの作成・破棄について**

  ライブラリはConnect後、Disconnectするまで作成したソケットを維持します。また、Disconnectするまで別のアプリケーションから同じプリンターに接続することはできません。

プリンターへのデータ送信完了時を基準とし、SocketKeepingTimeで設定するソケット維持時間経過後、使用しているソケットをいったん破棄します。その後すぐに新規のソケットを作成し、次の接続に使用します。
ソケット破棄の時点でプリンターが同じネットワーク上の別のホストから接続要求を受けていた場合、プリンターはそちらのホストとの通信を確立するため、再接続に失敗する可能性があります。

注意　　複数のAppから1台のプリンターへの同時接続をサポートしていません。

Disconnect　　デバイスとの通信切断

デバイスとの通信を終了します。

対象　　プラットフォーム共通　　全デバイス

構文　　`public void Disconnect();`

エラー　　本メソッド呼び出し時にエラーが発生した場合は、PrinterExceptionをスローします。
エラーについての詳細は、「**5.1.5 PrinterExceptionクラス**」を参照してください。

注意　　本メソッドの直前に、GetPrinterResponseの**PRINTER_RESPONSE_REQUEST**(実行応答リクエスト)により実行応答を取得することを推奨します。取得しない場合は、iOS/Androidデバイスのデータがすべてデバイスに送信される前に通信が切断され、データの一部が消失する場合があります。GetPrinterResponse、またはGetDisplayResponseを実行しない場合は、お客様のプログラムにおいて、問題が無いことを十分評価してからご使用ください。

GetPrinterResponse　　プリンターからの各種応答取得

プリンターからの応答データを取得します。

対象　　iOS　　全プリンター、プリンター経由のDSP-A01
Android　　全デバイス

構文
```
public void GetPrinterResponse<T>(PrinterResponseId responseId,
                                  object param,
                                  out T[] response);
```

パラメータ　　responseId　　プリンター応答種別定数
利用可能な定数は、
「5.1.1③ プリンター応答種別(PrinterResponseId)」を参照してください。

param　　コマンドパラメータ
プリンター応答種別定数により指定する値が異なります。
指定する値の説明は、下表を参照してください。

response　　取得した応答データを格納するバッファー
プリンター応答種別定数によりバッファーの型が異なります。
バッファーの型は、下表を参照してください。

応答種別定数	
パラメータ	説明
PRINTER_RESPONSE_REQUEST（実行応答リクエスト）	
param	int型で0～15（00h～0Fh）を指定してください。
response	int型の配列を指定してください。 応答取得成功時には、実行応答リクエストの応答コードが128～143（80h～8Fh）で格納されます。
PRINTER_RESPONSE_USER_AREA（ユーザ領域の残り容量の送信）	
param	nullを指定してください。
response	int型の配列を指定してください。 応答取得成功時には、ユーザ領域の残り容量（単位：バイト）が数値で格納されます。
PRINTER_RESPONSE_ARRANGE_USER_AREA（ユーザ領域の整理後の残り容量の送信）	
param	nullを指定してください。
response	int型の配列を指定してください。 応答取得成功時には、ユーザ領域の整理後の残り容量（単位：バイト）が数値で格納されます。
PRINTER_RESPONSE_NV_GRAPHICS（NVグラフィックスのメモリ容量の送信）	
param	nullを指定してください。
response	int型の配列を指定してください。 応答取得成功時には、NVグラフィックスメモリ容量（単位：バイト）が数値で格納されます。
PRINTER_RESPONSE_KEY_CODE（定義されているNVグラフィックスのキーコード一覧の送信）	
param	nullを指定してください。
response	string型の配列を指定してください。 応答取得成功時には、NVグラフィックスのキーコードが文字列の配列で格納されます。
PRINTER_RESPONSE_MAINTENANCE_COUNTER（メンテナンスカウンタの送信）	
param	nullを指定してください。
response	int型の配列を指定してください。 応答取得成功時は、下記のように格納されます。 buf[0] ：紙送り行数(単位：100ドットライン) buf[1] ：サーマルヘッド通電回数(単位：100ドットライン) buf[2] ：オートカッタ駆動回数*1 buf[3] ：製品の駆動時間(単位：分) buf[4] ：紙送り行数(積算)(単位：100ドットライン) buf[5] ：サーマルヘッド通電回数(積算)(単位：100ドットライン) buf[6] ：オートカッタ駆動回数（積算）*1 buf[7] ：製品の駆動時間(積算)(単位：分)
PRINTER_RESPONSE_PRINTER_IDENTIFIER（プリンタIDの送信）	
param	nullを指定してください。
response	応答取得成功時には、bufの先頭要素にPrinterIdentifierクラスのオブジェクトが格納されます。 例：buf.size() = 1、 buf.get(0) = ″(PrinterIdentifierクラスのオブジェクト)″ PrinterIdentifierクラスのオブジェクトについては、 「5.1.4 PrinterIdentifierクラス」を参照してください。

*1: MP-B30、MP-B30L、MP-B21L、MP-B20、MP-A40はサポートしていません。常に0を返します。

エラー　本メソッド呼び出し時にエラーが発生した場合は、PrinterExceptionをスローします。
エラーについての詳細は、「**5.1.5 PrinterExceptionクラス**」を参照してください。

StartDiscoveryDevice　デバイスの探索開始

デバイスを探索します。

対象　iOS　全プリンター、プリンター経由のDSP-A01
Android　全デバイス

構文
```
public void StartDiscoveryDevice(DeviceType deviceType,
                                  PortType portType,
                                  int timeout,
                                  Action completion);
```

パラメータ　deviceType　デバイスタイプ
利用可能な定数は、「5.1.1④ デバイスタイプ（DeviceType）」を参照してください。
iOSの場合は、**TYPE_PRINTER**のみ有効です。

portType　ポート種別
利用可能な定数は、「5.1.1② ポート種別（PortType）」を参照してください。

timeout　探索タイムアウト時間（ミリ秒：ms）
タイムアウト時間を設定します。
有効範囲は、3000～60000です。
3000未満の値を指定した場合は3000msに設定されます。
60000を超える値を指定した場合は60000msに設定されます。

- **PORT_TYPE_TCP**の場合
  ローカルブロードキャストパケットを送信し、timeoutで指定した時間が経過するまでプリンターからの応答を待ちます。
- **PORT_TYPE_BLUETOOTH**の場合
  デバイスを探索中の場合、timeoutで指定した時間が経過するまで探索を行います。デバイスの探索が終了した場合は、timeoutの時間を待たずに探索を終了します。
  iOSの場合は、timeoutの指定は無視されます。
- **PORT_TYPE_USB**の場合
  timeoutの指定は無視されます。

completion　デバイス探索終了時に実行するブロック

エラー　本メソッド呼び出し時にエラーが発生した場合は、PrinterExceptionをスローします。

説明　本メソッドは、デバイスを探索します。
発見したデバイスのデバイス情報は、GetFoundDeviceで取得できます。

CancelDiscoveryDevice　デバイスの探索中断

実行中のStartDiscoveryDeviceを中断します。

対象　プラットフォーム共通　全プリンター、プリンター経由のDSP-A01

構文　public void CancelDiscoveryDevice();

説明　StartDiscoveryDeviceのcompletionで設定したブロックにイベントとして通知します。

GetFoundDevice　発見されたデバイス情報リストの取得

StartDiscoveryDeviceで発見されたデバイス情報リストを、格納先のDeviceInfoクラスから取得します。

対象　iOS　全プリンター、プリンター経由のDSP-A01
Android　全デバイス

構文　public DeviceInfo[] GetFoundDevice();

説明　「**5.1.3 DeviceInfoクラス**」を参照してください。

Defragment　メモリ領域の最適化

メモリ領域を最適化します。

対象　iOS　プリンター経由のDSP-A01
Android　プリンター経由のDSP-A01、DSP-A01単体

構文　public void Defragment(MemoryArea memoryArea);

パラメータ　memoryArea　メモリ領域
利用可能な定数は、「5.1.1⑤ メモリ領域（MemoryArea）」を参照してください。

エラー　本メソッド呼び出し時にエラーが発生した場合は、PrinterExceptionをスローします。
エラーについての詳細は、「**5.1.5 PrinterExceptionクラス**」を参照してください。

説明　最適化には数分かかることがあります。
最適化中にプリンターの電源をオフにしないでください。
本メソッドを実行するとディスプレイはスタンバイモードになります。選択中のテンプレートは選択が解除されます。

InitializeMemoryArea　メモリ領域の初期化

メモリ領域を初期化します。

対象　iOS　プリンター経由のDSP-A01
Android　プリンター経由のDSP-A01、DSP-A01単体

構文　public void InitializeMemoryArea(MemoryArea memoryArea);

パラメータ　memoryArea　メモリ領域
利用可能な定数は、「5.1.1⑤ メモリ領域（MemoryArea）」を参照してください。

エラー　本メソッド呼び出し時にエラーが発生した場合は、PrinterExceptionをスローします。
エラーについての詳細は、「**5.1.5 PrinterExceptionクラス**」を参照してください。

説明　初期化には数分かかることがあります。
初期化中にプリンターの電源をオフにしないでください。
本メソッドを実行するとディスプレイはスタンバイモードになります。選択中のテンプレートは選択が解除されます。

注意　**MEMORY_DISPLAY_USERMEMORY**を指定して初期化すると、下記のメソッドで登録したデータは削除されます。

- ・RegisterTemplate
- ・RegisterImageData
- ・ControlMacroRegistration
- ・RegisterUserDefinedCharacter
- ・RegisterOptionFont

工場出荷時に登録されるシステムで利用する一部のデータも削除されます。
それにより、プリンターでエラーなどが発生した際にディスプレイに表示されるガイド画面の表示ができなくなります。
使用済みのメモリ領域を再度利用する場合は、Defragmentをご利用ください。

ShowTemplate　テンプレートの表示

ディスプレイの画面をテンプレート表示にします。

対象　iOS　プリンター経由のDSP-A01
Android　プリンター経由のDSP-A01、DSP-A01単体

構文　`public void ShowTemplate(int time_ms);`

パラメータ　time_ms　表示時間（ms:ミリ秒）
画面に表示する時間をtime_ms （ms)で指定してください。
有効範囲は、0～25500です。
0を超える値、かつ100未満の値を指定した場合は100msに設定されます。
25500を超える値を指定した場合は25500msに設定されます。

マクロ登録の場合:
time_msに0を指定した場合は、テンプレート表示を続けます。
time_msに0以外を指定した場合は、表示時間経過後は次のテンプレートに移ります。

マクロ登録以外の場合:
time_msに0を指定した場合は、テンプレート表示を続けます。
time_msに0以外を指定した場合は、表示時間経過後は直前のテンプレートに戻ります。直前に表示時間に0以外を指定して画面更新を行っていた場合は、0を指定して画面更新を行ったテンプレートまで戻ります。

エラー　本メソッド呼び出し時にエラーが発生した場合は、PrinterExceptionをスローします。
エラーについての詳細は、「**5.1.5 PrinterExceptionクラス**」を参照してください。

説明　画面を更新し、下記のメソッドで指定されたデータを表示します。
・SelectTemplate
・SetTemplateImageData
・SelectTemplateTextObject
・SetTemplateTextAlignment
・SetTemplateTextLeftMargin
・SetTemplateTextLineSpacing
・SetTemplateTextBold
・SetTemplateTextUnderline
・SetTemplateTextSize
・SetTemplateTextFont
・SetTemplateTextRightSpacing
・SetTemplateTextColor
・SetTemplateTextData
・SetTemplateBarcodeData
・SetTemplateQRCodeData

ShowSlide　スライドの表示

ディスプレイの画面をスライド表示にします。

対象　iOS　プリンター経由のDSP-A01
Android　プリンター経由のDSP-A01、DSP-A01単体

構文
```
public void ShowSlide(int slideID, int time_ms);
```

パラメータ

slideID　スライドID
RegisterSlideDataで登録したスライドデータのスライドIDを指定してください。
有効範囲は、0～91です。
指定したスライドIDにスライドデータが登録されていない場合は無視されます。

time_ms　表示時間（ms:ミリ秒）
画面に表示する時間をtime_ms （ms）で指定してください。
有効範囲は、0～25500です。
0を超える値、かつ100未満の値を指定した場合は100msに設定されます。
25500を超える値を指定した場合は25500msに設定されます。

マクロ登録の場合：
time_msに0を指定した場合は、スライド表示を続けます。
time_msに0以外を指定した場合は、スライド表示時間経過後は次のスライドに移ります。
マクロ登録以外の場合：
time_msに0を指定した場合は、スライド表示を続けます。
time_msに0以外を指定した場合は、スライド表示時間経過後は直前の画面に戻ります。直前に表示時間に0以外を指定して画面更新を行っていた場合は、0を指定して画面更新を行ったスライドまで戻ります。

エラー　本メソッド呼び出し時にエラーが発生した場合は、PrinterExceptionをスローします。
エラーについての詳細は、「**5.1.5 PrinterExceptionクラス**」を参照してください。

説明　選択中のテンプレートは選択が解除されます。

EnterStandbyMode　スタンバイの表示

ディスプレイをスタンバイモードにします。

対象　iOS　プリンター経由のDSP-A01
Android　プリンター経由のDSP-A01、DSP-A01単体

構文　`public void EnterStandbyMode();`

エラー　本メソッド呼び出し時にエラーが発生した場合は、PrinterExceptionをスローします。
エラーについての詳細は、「**5.1.5 PrinterExceptionクラス**」を参照してください。

説明　本メソッドはスタンバイモード中、及びガイドモード中は無視されます。
選択中のテンプレートは選択が解除されます。

ExecuteMacro　マクロの実行

マクロを実行します。

対象　iOS　プリンター経由のDSP-A01
Android　プリンター経由のDSP-A01、DSP-A01単体

構文　`public void ExecuteMacro(int macroID, int repeatCount);`

パラメータ　macroID　マクロID
ControlMacroRegistrationで登録したマクロのマクロIDを指定してください。
有効範囲は、0～127です。
指定したマクロIDにマクロが登録されていない場合は無視されます。

repeatCount　実行回数
マクロを実行する回数を指定してください。
有効範囲は、0～255です。
0を指定した場合は、繰り返しを継続します。

エラー　本メソッド呼び出し時にエラーが発生した場合は、PrinterExceptionをスローします。
エラーについての詳細は、「**5.1.5 PrinterExceptionクラス**」を参照してください。

説明　選択中のテンプレートは選択が解除されます。

TurnOnScreen　画面の表示オン・オフ

ディスプレイの画面のバックライトのオン・オフを設定します。

対象　iOS　プリンター経由のDSP-A01
Android　プリンター経由のDSP-A01、DSP-A01単体

構文
```
public void TurnOnScreen(bool isOn);
```

パラメータ　isOn　画面状態
画面状態を下記から指定してください。
true：　バックライトオン
false：　バックライトオフ

エラー　本メソッド呼び出し時にエラーが発生した場合は、PrinterExceptionをスローします。
エラーについての詳細は、「**5.1.5 PrinterExceptionクラス**」を参照してください。

SelectTemplate　テンプレートの選択

ディスプレイで表示するテンプレートを選択します。
構文(a)は、テンプレートとテンプレートの背景に使用するスライドデータを選択できます。
構文(b)は、テンプレートを選択できます。

対象　iOS　プリンター経由のDSP-A01
Android　プリンター経由のDSP-A01、DSP-A01単体

構文
(a)
```
public void SelectTemplate(int templateID, int slideID);
```
(b)
```
public void SelectTemplate(int templateID);
```

パラメータ　templateID　テンプレートID
選択するテンプレートのテンプレートIDを指定してください。
有効範囲は、0～127です。
指定したテンプレートIDにテンプレートが登録されていない場合は無視されます。

slideID　スライドID
テンプレートの背景に使用するスライドデータのスライドIDを指定してください。
有効範囲は、0～91です。
指定したスライドIDにスライドデータが登録されていない場合は無視されます。

エラー　本メソッド呼び出し時にエラーが発生した場合は、PrinterExceptionをスローします。
エラーについての詳細は、「**5.1.5 PrinterExceptionクラス**」を参照してください。

説明　本メソッドを実行するとテンプレート上のデータはクリアされます。
選択したテンプレートはShowTemplateを実行すると画面に表示されます。

選択中のテンプレートはShowSlide、EnterStandbyMode、ExecuteMacroで選択が解除されます。
選択中のテンプレートはShowTemplateで時間指定表示を実行すると選択が解除されます。

SetTemplateTextDataで20h～7EhのASCII文字以外を入力する場合、CodePageの値に応じて下記のテンプレートを使用してください。
CodePageが**CODE_PAGE_KATAKANA**の場合：
エンコード指定がShift_JISのテンプレートを使用してください。
CodePageが**CODE_PAGE_KATAKANA**以外：
エンコード指定がISO-2022-JPのテンプレートを使用してください。

SetTemplateImageData　イメージデータの設定

選択中のテンプレートに表示するイメージデータを設定します。

対象　iOS　プリンター経由のDSP-A01
Android　プリンター経由のDSP-A01、DSP-A01単体

構文
```
public void SetTemplateImageData(int mapID, int imageID);
```

パラメータ
mapID　マップID
有効範囲は、0～63です。
指定したマップIDがテンプレートに定義されていない場合は無視されます。

imageID　イメージID
RegisterImageDataで登録したイメージデータのイメージIDを指定してください。
有効範囲は、0～63です。
指定したイメージIDにイメージデータが登録されていない場合は無視されます。

エラー　本メソッド呼び出し時にエラーが発生した場合は、PrinterExceptionをスローします。
エラーについての詳細は、「**5.1.5 PrinterExceptionクラス**」を参照してください。

説明　選択中のテンプレートのマップIDを本メソッドで指定してから、配置するイメージIDを指定してください。
指定したイメージデータはShowTemplateを実行すると画面に表示されます。

本メソッドの設定は下記の条件でクリアされます。
・SelectTemplateを実行した場合
・ShowTemplateのtime_msで0以外を指定し、指定した表示時間を経過した場合
・ExecuteMacroに登録されているShowTemplateを実行した場合

テンプレートが未選択の場合は無視されます。

SelectTemplateTextObject　text要素の選択

編集を開始するtext要素を選択します。

対象　iOS　プリンター経由のDSP-A01
Android　プリンター経由のDSP-A01、DSP-A01単体

構文　`public void SelectTemplateTextObject(int mapID);`

パラメータ　mapID　マップID
有効範囲は、0～63です。
指定したマップIDがテンプレートに定義されていない場合は無視されます。

エラー　本メソッド呼び出し時にエラーが発生した場合は、PrinterExceptionをスローします。
エラーについての詳細は、「**5.1.5 PrinterExceptionクラス**」を参照してください。

説明　選択中のテンプレートのマップIDを本メソッドで指定してから、文字の編集を開始してください。

指定したマップIDのtext要素にスクロールを設定している場合は、ShowTemplateの後に本メソッドを実行した場合、スクロールを実行します。
指定したマップIDのtext要素にスクロールを設定していない場合は、ShowTemplateの後に本メソッドを実行した場合、入力済みのテキストデータを破棄します。

本メソッドの指定は下記の条件でクリアされます。
・SelectTemplateを実行した場合
・ShowTemplateを実行した場合

テンプレートが未選択の場合は無視されます。

SetTemplateTextAlignment　テキストデータの位置揃え

ディスプレイで表示するテキストデータに位置揃えを指定します。

対象　iOS　プリンター経由のDSP-A01
Android　プリンター経由のDSP-A01、DSP-A01単体

構文　`public void SetTemplateTextAlignment(PrintAlignment alignment);`

パラメータ　alignment　位置揃え
利用可能な定数は、「5.1.1⑥ 位置揃え（PrintAlignment）」を参照してください。

エラー　本メソッド呼び出し時にエラーが発生した場合は、PrinterExceptionをスローします。
エラーについての詳細は、「**5.1.5 PrinterExceptionクラス**」を参照してください。

説明　位置揃えは下記の場合のみ有効です。
・SelectTemplateTextObjectで指定したマップIDにテキストデータが未入力時
・SelectTemplateTextObjectで指定したマップIDに入力したテキストデータが改行直後

本メソッドを実行する前にSelectTemplateTextObjectでマップIDを指定してください。
本メソッドを実行した後にSetTemplateTextDataでテキストデータを入力してください。
入力したテキストデータはShowTemplateを実行すると画面に表示されます。

本メソッドの設定は下記の条件でクリアされます。
・SelectTemplateを実行した場合
・ShowTemplateのtime_msで0以外を指定し、指定した表示時間を経過した場合
・ExecuteMacroに登録されているShowTemplateを実行した場合

SetTemplateTextLeftMargin　テキストデータの左マージンの設定

ディスプレイで表示するテキストデータに左マージンを設定します。

対象　iOS　プリンター経由のDSP-A01
Android　プリンター経由のDSP-A01、DSP-A01単体

構文　public void SetTemplateTextLeftMargin(int margin);

パラメータ　margin　左マージン(ピクセル:px)
有効範囲は、0～479です。

エラー　本メソッド呼び出し時にエラーが発生した場合は、PrinterExceptionをスローします。
エラーについての詳細は、「**5.1.5 PrinterExceptionクラス**」を参照してください。

説明　左マージンの設定は下記の場合のみ有効です。
・SelectTemplateTextObjectで指定したマップIDにテキストデータが未入力時
・SelectTemplateTextObjectで指定したマップIDに入力したテキストデータが改行直後

本メソッドを実行する前にSelectTemplateTextObjectでマップIDを指定してください。
本メソッドを実行した後にSetTemplateTextDataでテキストデータを入力してください。
入力したテキストデータはShowTemplateを実行すると画面に表示されます。

本メソッドの設定は下記の条件でクリアされます。
・SelectTemplateを実行した場合
・ShowTemplateのtime_msで0以外を指定し、指定した表示時間を経過した場合
・ExecuteMacroに登録されているShowTemplateを実行した場合

SetTemplateTextLineSpacing　テキストデータの改行量の設定

ディスプレイで表示するテキストデータに改行量を設定します。

対象　iOS　プリンター経由のDSP-A01
Android　プリンター経由のDSP-A01、DSP-A01単体

構文　public void SetTemplateTextLineSpacing(int spacing);

パラメータ　spacing　改行量(ピクセル:px)
有効範囲は、0～255です。

エラー　本メソッド呼び出し時にエラーが発生した場合は、PrinterExceptionをスローします。
エラーについての詳細は、「**5.1.5 PrinterExceptionクラス**」を参照してください。

説明　改行量の設定は下記の場合のみ有効です。
・SelectTemplateTextObjectで指定したマップIDにテキストデータが未入力時
・SelectTemplateTextObjectで指定したマップIDに入力したテキストデータが改行直後

指定したマップIDのtext要素にスクロールを設定している場合は、本メソッドの指定は表示されません。

本メソッドを実行する前にSelectTemplateTextObjectでマップIDを指定してください。
本メソッドを実行した後にSetTemplateTextDataでテキストデータを入力してください。
入力したテキストデータはShowTemplateを実行すると画面に表示されます。

本メソッドの設定は下記の条件でクリアされます。
・SelectTemplateを実行した場合
・ShowTemplateのtime_msで0以外を指定し、指定した表示時間を経過した場合
・ExecuteMacroに登録されているShowTemplateを実行した場合

SetTemplateTextBold　テキストデータの強調文字の設定

ディスプレイで表示するテキストデータに強調文字を設定します。

対象　iOS　プリンター経由のDSP-A01
Android　プリンター経由のDSP-A01、DSP-A01単体

構文　`public void SetTemplateTextBold(CharacterBold bold);`

パラメータ　bold　強調文字
利用可能な定数は、「5.1.1⑦ 強調文字（CharacterBold）」を参照してください。

エラー　本メソッド呼び出し時にエラーが発生した場合は、PrinterExceptionをスローします。
エラーについての詳細は、「**5.1.5 PrinterExceptionクラス**」を参照してください。

説明　本メソッド実行後のテキストデータから強調文字は適用されます。1文字ずつ強調文字の設定ができます。

本メソッドを実行する前にSelectTemplateTextObjectでマップIDを指定してください。
本メソッドを実行した後にSetTemplateTextDataでテキストデータを入力してください。
入力したテキストデータはShowTemplateを実行すると画面に表示されます。

本メソッドの設定は下記の条件でクリアされます。
・SelectTemplateを実行した場合
・ShowTemplateのtime_msで0以外を指定し、指定した表示時間を経過した場合
・ExecuteMacroに登録されているShowTemplateを実行した場合

SetTemplateTextUnderline　テキストデータのアンダーラインの設定

ディスプレイで表示するテキストデータにアンダーラインを設定します。

対象　iOS　プリンター経由のDSP-A01
Android　プリンター経由のDSP-A01、DSP-A01単体

構文　`public void SetTemplateTextUnderline(CharacterUnderline underline);`

パラメータ　underline　アンダーライン
利用可能な定数は、「5.1.1⑧ アンダーライン（CharacterUnderline）」を参照してください。

エラー　本メソッド呼び出し時にエラーが発生した場合は、PrinterExceptionをスローします。
エラーについての詳細は、「**5.1.5 PrinterExceptionクラス**」を参照してください。

説明　本メソッド実行後のテキストデータからアンダーラインは適用されます。1文字ずつアンダーラインの設定ができます。

本メソッドを実行する前にSelectTemplateTextObjectでマップIDを指定してください。
本メソッドを実行した後にSetTemplateTextDataでテキストデータを入力してください。
入力したテキストデータはShowTemplateを実行すると画面に表示されます。

本メソッドの設定は下記の条件でクリアされます。
・SelectTemplateを実行した場合
・ShowTemplateのtime_msで0以外を指定し、指定した表示時間を経過した場合
・ExecuteMacroに登録されているShowTemplateを実行した場合

SetTemplateTextSize　テキストデータの文字サイズの設定

ディスプレイで表示するテキストデータに文字サイズを設定します。

対象　iOS　プリンター経由のDSP-A01
Android　プリンター経由のDSP-A01、DSP-A01単体

構文
```
public void SetTemplateTextSize(CharacterScale scale);
```

パラメータ　scale　文字倍率
利用可能な定数は、「5.1.1⑨ 文字倍率（CharacterScale）」を参照してください。

エラー　本メソッド呼び出し時にエラーが発生した場合は、PrinterExceptionをスローします。
エラーについての詳細は、「**5.1.5 PrinterExceptionクラス**」を参照してください。

説明　本メソッド実行後のテキストデータから文字サイズは適用されます。1文字ずつ文字サイズの設定ができます。

本メソッドを実行する前にSelectTemplateTextObjectでマップIDを指定してください。
本メソッドを実行した後にSetTemplateTextDataでテキストデータを入力してください。
入力したテキストデータはShowTemplateを実行すると画面に表示されます。

本メソッドの設定は下記の条件でクリアされます。
・SelectTemplateを実行した場合
・ShowTemplateのtime_msで0以外を指定し、指定した表示時間を経過した場合
・ExecuteMacroに登録されているShowTemplateを実行した場合

SetTemplateTextFont　テキストデータの文字フォントの設定

ディスプレイで表示するテキストデータに文字フォントを設定します。

対象　iOS　プリンター経由のDSP-A01
Android　プリンター経由のDSP-A01、DSP-A01単体

構文
```
public void SetTemplateTextFont(CharacterFont font);
```

パラメータ　font　　文字フォント
利用可能な定数は、「5.1.1⑩ 文字フォント（CharacterFont）」を参照してください。

エラー　本メソッド呼び出し時にエラーが発生した場合は、PrinterExceptionをスローします。
エラーについての詳細は、「**5.1.5 PrinterExceptionクラス**」を参照してください。

説明　本メソッド実行後のテキストデータから文字フォントは適用されます。1文字ずつ文字フォントの設定ができます。

本メソッドを実行する前にSelectTemplateTextObjectでマップIDを指定してください。
本メソッドを実行した後にSetTemplateTextDataでテキストデータを入力してください。
入力したテキストデータはShowTemplateを実行すると画面に表示されます。

本メソッドの設定は下記の条件でクリアされます。
・SelectTemplateを実行した場合
・ShowTemplateのtime_msで0以外を指定し、指定した表示時間を経過した場合
・ExecuteMacroに登録されているShowTemplateを実行した場合

SetTemplateTextRegisteredFont　テキストデータの登録フォントの設定

ディスプレイで表示するテキストデータで使用する登録フォントを設定します。

対象　iOS　プリンター経由のDSP-A01
Android　プリンター経由のDSP-A01、DSP-A01単体

構文　`public void SetTemplateTextRegisteredFont(RegisteredFont font);`

パラメータ　font　　登録フォント
利用可能な定数は、「5.1.1⑪ 登録フォント（RegisteredFont）」を参照してください。
オプションフォントが登録されていない場合は無視されます。

エラー　本メソッド呼び出し時にエラーが発生した場合は、PrinterExceptionをスローします。
エラーについての詳細は、「**5.1.5 PrinterExceptionクラス**」を参照してください。

説明　本メソッド実行後のテキストデータから登録フォントは適用されます。1文字ずつ登録フォントの設定ができます。

本メソッドを実行した後にSetTemplateTextDataでテキストデータを入力してください。
入力したテキストデータはShowTemplateを実行すると画面に表示されます。

本メソッドの設定は下記の条件でクリアされます。
・SelectTemplateを実行した場合
・ShowTemplateのtime_msで0以外を指定し、指定した表示時間を経過した場合
・ExecuteMacroに登録されているShowTemplateを実行した場合

SetTemplateTextRightSpacing　テキストデータの右スペース量の設定

ディスプレイで表示するテキストデータに右スペース量を設定します。

対象　iOS　プリンター経由のDSP-A01
Android　プリンター経由のDSP-A01、DSP-A01単体

構文　`public void SetTemplateTextRightSpacing(int spacing);`

パラメータ　spacing　文字の右スペース量（ピクセル：px）
有効範囲は、0～255です。

エラー　本メソッド呼び出し時にエラーが発生した場合は、PrinterExceptionをスローします。
エラーについての詳細は、「**5.1.5 PrinterExceptionクラス**」を参照してください。

説明　本メソッド実行後のテキストデータから文字の右スペース量は適用されます。1文字ずつ右スペース量の設定ができます。

本メソッドを実行する前にSelectTemplateTextObjectでマップIDを指定してください。
本メソッドを実行した後にSetTemplateTextDataでテキストデータを入力してください。
入力したテキストデータはShowTemplateを実行すると画面に表示されます。

本メソッドの設定は下記の条件でクリアされます。
- SelectTemplateを実行した場合
- ShowTemplateのtime_msで0以外を指定し、指定した表示時間を経過した場合
- ExecuteMacroに登録されているShowTemplateを実行した場合

SetTemplateTextColor　テキストデータの文字色の設定

ディスプレイで表示するテキストデータで使用する文字色を設定します。

対象　iOS　プリンター経由のDSP-A01
Android　プリンター経由のDSP-A01、DSP-A01単体

構文　`public void SetTemplateTextColor(int color);`

パラメータ　color　文字色
有効範囲は、0～0xffffffです。

エラー　本メソッド呼び出し時にエラーが発生した場合は、PrinterExceptionをスローします。
エラーについての詳細は、「**5.1.5 PrinterExceptionクラス**」を参照してください。

説明　本メソッドでテキストデータに文字色を設定できます。
文字色はRGB24bitカラーで設定できます。設定した文字色はRGB555の16bitカラーに減色して表示されます。

本メソッド実行後のテキストデータから文字色は適用されます。1文字ずつ文字色の設定ができます。

本メソッドを実行する前にSelectTemplateTextObjectでマップIDを指定してください。
本メソッドを実行した後にSetTemplateTextDataでテキストデータを入力してください。
入力したテキストデータはShowTemplateを実行すると画面に表示されます。

本メソッドの設定は下記の条件でクリアされます。
・SelectTemplateを実行した場合
・ShowTemplateのtime_msで0以外を指定し、指定した表示時間を経過した場合
・ExecuteMacroに登録されているShowTemplateを実行した場合

SetTemplateTextData　テキストデータの入力

ディスプレイで表示するテキストデータを入力します。

対象　iOS　プリンター経由のDSP-A01
Android　プリンター経由のDSP-A01、DSP-A01単体

構文
```
public void SetTemplateTextData(string text);
```

パラメータ　text　ディスプレイで表示するテキストデータ
一度に指定可能なデータサイズは1～1020バイトです。

エラー　本メソッド呼び出し時にエラーが発生した場合は、PrinterExceptionをスローします。
エラーについての詳細は、「**5.1.5 PrinterExceptionクラス**」を参照してください。

説明　本メソッドは、入力されたテキストデータを、InternationalCharacter、及びCodePageの設定を基に表示可能なテキストデータにエンコードして画面に表示します。

選択中のテンプレートのマップIDをSelectTemplateTextObjectで指定してから、本メソッドでテキストデータを入力してください。
SelectTemplateTextObjectでマップIDが指定されていない場合は、無視されます。
入力したテキストデータはShowTemplateを実行すると画面に表示されます。

本メソッドで入力したテキストデータは下記の条件でクリアされます。
・SelectTemplateを実行した場合
・ShowTemplateのtime_msで0以外を指定し、指定した表示時間を経過した場合
・ExecuteMacroに登録されているShowTemplateを実行した場合

SetTemplateBarcodeData　バーコードデータの入力

選択中のテンプレートのバーコード要素のマップIDを指定し、バーコードデータを入力します。
構文(a)は、表示するバーコードにデータを文字列で入力できます。
構文(b)は、表示するバーコードにデータをバイト配列で入力できます。

対象　iOS　プリンター経由のDSP-A01
Android　プリンター経由のDSP-A01、DSP-A01単体

構文
(a)
```
public void SetTemplateBarcodeData(int mapID, string text);
```
(b)
```
public void SetTemplateBarcodeData(int mapID, byte[] data);
```

パラメータ　mapID　マップID
有効範囲は、0～7です。
指定したマップIDが選択中のテンプレートに定義されていない場合は無視されます。

text　　バーコードデータ
入力可能な文字は、英数字('0'～'9'、'A' ～'Z'、'a'～'z')などの
ASCII文字コード20h(スペース)～7Eh(チルダ)の文字です。
入力可能な文字数は1～150文字です。
バーコードの仕様に沿わないバーコードデータは無視されます。

data　　バーコードデータ
入力可能な値は00h～7Fhです。
入力可能なデータ数は1～150です。
バーコードの仕様に沿わないバーコードデータは無視されます。

エラー　本メソッド呼び出し時にエラーが発生した場合は、PrinterExceptionをスローします。
エラーについての詳細は、「**5.1.5 PrinterExceptionクラス**」を参照してください。

説明　選択中のテンプレートのマップIDを本メソッドで指定してから、バーコードデータを入力してください。
入力したバーコードデータはShowTemplateを実行すると画面に表示されます。

本メソッドで入力したバーコードデータは下記の条件でクリアされます。
・SelectTemplateを実行した場合
・ShowTemplateのtime_msで0以外を指定し、指定した表示時間を経過した場合
・ExecuteMacroに登録されているShowTemplateを実行した場合

テンプレートが未選択の場合は無視されます。

SetTemplateQRCodeData　QRコードデータの入力

選択中のテンプレートのqr要素のマップIDを指定し、QRコードデータを入力します。
構文(a)は、QRコードデータを文字列で入力できます。
構文(b)は、QRコードデータを文字列で入力できます。moduleSize、errorCorrection、mode、及びqrQuietZoneは選択中のテンプレートの設定が反映されます。
構文(c)は、QRコードデータをバイト配列で入力できます。
構文(d)は、QRコードデータをバイト配列で入力できます。moduleSize、errorCorrection、mode、及びqrQuietZoneは選択中のテンプレートの設定が反映されます。

対象　iOS　プリンター経由のDSP-A01
Android　プリンター経由のDSP-A01、DSP-A01単体

構文　(a)
```
public void SetTemplateQRCodeData(int mapID,
                                   ModuleSize moduleSize,
                                   ErrorCorrection errorCorrection,
                                   QrDataMode mode,
                                   QrQuietZone qrQuietZone,
                                   string text);
```

(b)
```
public void SetTemplateQRCodeData(int mapID, string text);
```

(c) public void SetTemplateQRCodeData(int mapID,
ModuleSize moduleSize,
ErrorCorrection errorCorrection,
QrDataMode mode,
QrQuietZone qrQuietZone,
byte[] data);

(d) public void SetTemplateQRCodeData(int mapID, byte[] data);

パラメータ

mapID　マップID
有効範囲は、0～7です。
指定したマップIDが選択中のテンプレートに定義されていない場合は無視されます。

moduleSize　モジュールサイズ
利用可能な定数は、「5.1.1⑫ モジュールサイズ(ModuleSize)」を参照してください。

errorCorrection
エラー訂正レベル
利用可能な定数は、「5.1.1⑬ エラー訂正レベル(ErrorCorrection)」を参照してください。

mode　データモード
利用可能な定数は、「5.1.1⑭ QRデータモード(QrDataMode)」を参照してください。

qrQuietZone　クワイエットゾーン
利用可能な定数は、「5.1.1⑮ QRクワイエットゾーン(QrQuietZone)」を参照してください。

text　QRコードデータ
入力可能な文字は下記の通りです。
・英数字('0'～'9'、'A' ～'Z'、'a'～'z')などのASCII文字コード20h(スペース)～7Eh(チルダ)の文字です。
・JIS X 0201に基づく8ビットラテン/カナ文字
・JIS X 0208に基づくシフトJISコード
入力可能なデータサイズは1～3909バイトです。
QRコードの仕様に沿わないQRコードデータは無視されます。

data　QRコードデータ
ディスプレイで表示するQRコードデータ
入力可能な値は00h～FFhです。
入力可能なデータ数は1～3909です。
QRコードの仕様に沿わないQRコードデータは無視されます。

エラー　本メソッド呼び出し時にエラーが発生した場合は、PrinterExceptionをスローします。
エラーについての詳細は、「**5.1.5 PrinterExceptionクラス**」を参照してください。

説明　　選択中のテンプレートのマップIDを本メソッドで指定してから、QRコードデータを入力してください。
入力したQRコードデータはShowTemplateを実行すると画面に表示されます。

本メソッドで入力したQRコードデータは下記の条件でクリアされます。
・SelectTemplateを実行した場合
・ShowTemplateのtime_msで0以外を指定し、指定した表示時間を経過した場合
・ExecuteMacroに登録されているShowTemplateを実行した場合

テンプレートが未選択の場合は無視されます。

RegisterTemplate　テンプレートの登録

ディスプレイにテンプレートを登録します。

対象　　iOS　　プリンター経由のDSP-A01
Android　　プリンター経由のDSP-A01、DSP-A01単体

構文
```
public void RegisterTemplate(int templateID,
                             string label,
                             string fileName);
```

パラメータ　templateID　　テンプレートID
登録するテンプレートのテンプレートIDを指定してください。
有効範囲は、0～127です。
テンプレートID127はシステムで利用していますので、指定しないでください。

label　　テンプレート名
登録するテンプレートに識別用の名前を付けることができます。
指定可能な文字は、英数字('0'～'9'、'A'～'Z'、'a'～'z')などのASCII文字コード20h(スペース)～7Eh(チルダ)の文字です。
ユニコードの00A5h('¥')は用いないでください。
指定可能な文字数は0～32文字です。
labelは省略することも可能です。省略する場合はnullを指定してください。
指定したテンプレート名はGetDisplayResponseで取得できます。

fileName　　ディスプレイに登録するテンプレートデータのファイルパス
入力可能な形式について下記に説明します。

iOSの場合:
使用するファイルパス文字列

Androidの場合:

- Javaの標準的なクラス「java.io.File」が扱う絶対パス文字列
  Android 10 (API 29)以降をアプリのターゲットとする場合には、一部のファイルが直接扱えなくなりますのでご注意ください。
  詳細については、「3.4 注意事項 – 対象範囲別ストレージについて」を参照してください。

- Androidに用意されたクラス「android.net.Uri」が扱う下記スキーム名のURI文字列
  ・file://
  ・content://
  本パラメータには"ストレージ アクセス フレームワーク"から取得したURI文字列を指定する必要があります。"ストレージ アクセス フレームワーク"から取得せずに作成されたURIはファイルを開けない可能性がありますのでご注意ください。

サポートするファイルの拡張子は.xmlです。
登録可能な最大データサイズは8192バイトです。
登録するテンプレートデータの詳細については「ディスプレイ用技術説明書」を参照してください。

エラー　本メソッド呼び出し時にエラーが発生した場合は、PrinterExceptionをスローします。
エラーについての詳細は、「**5.1.5 PrinterExceptionクラス**」を参照してください。

説明　本メソッドを実行するとディスプレイはスタンバイモードになります。選択中のテンプレートは選択が解除されます。

注意　工場出荷時に登録されるデータは、品質向上のため予告無く追加・変更される場合があります。

UnregisterTemplate　テンプレートの削除

ディスプレイに登録済みのテンプレートを削除します。

対象　iOS　プリンター経由のDSP-A01
Android　プリンター経由のDSP-A01、DSP-A01単体

構文
```
public void UnregisterTemplate(int templateID);
```

パラメータ　templateID　テンプレートID
削除するテンプレートのテンプレートIDを指定してください。
有効範囲は、0～127です。
指定したテンプレートIDにテンプレートが登録されていない場合は無視されます。

エラー　本メソッド呼び出し時にエラーが発生した場合は、PrinterExceptionをスローします。
エラーについての詳細は、「**5.1.5 PrinterExceptionクラス**」を参照してください。

説明　本メソッドを実行するとディスプレイはスタンバイモードになります。選択中のテンプレートは選択が解除されます。

テンプレートを削除しても使用済みのメモリは解放されません。使用済みのメモリは、Defragmentを実行すると再度利用できます。

RegisterImageData　　イメージデータの登録

ディスプレイにイメージデータを登録します。

対象　　iOS　　プリンター経由のDSP-A01
　　Android　　プリンター経由のDSP-A01、DSP-A01単体

構文
```
public void RegisterImageData(int imageID,
                              string label,
                              string fileName);
```

パラメータ　　imageID　　イメージID
有効範囲は、0～63です。
イメージID49～63はシステムで利用していますので、指定しないでください。

label　　イメージ名
登録するイメージデータに識別用の名前を付けることができます。
指定可能な文字は、英数字('0'～'9'、'A'～'Z'、'a'～'z')などの
ASCII文字コード20h(スペース)～7Eh(チルダ)の文字です。
ユニコードの00A5h('¥')は用いないでください。
指定可能な文字数は0～32文字です。
labelは省略することも可能です。省略する場合はnullを指定してください。
指定したイメージ名はGetDisplayResponseで取得できます。

fileName　　ファイルパス
登録するイメージデータのファイル名を指定してください。
入力可能な形式について下記に説明します。

iOSの場合:
　使用するファイルパス文字列

Androidの場合:

- Javaの標準的なクラス「java.io.File」が扱う絶対パス文字列
  Android 10 (API 29)以降をアプリのターゲットとする場合には、一部のファイルが直接扱えなくなりますのでご注意ください。
  詳細については、「3.4 注意事項 - 対象範囲別ストレージについて」を参照してください。

- Androidに用意されたクラス「android.net.Uri」が扱う下記スキーム名のURI文字列
  ・file://
  ・content://
  本パラメータには"ストレージ アクセス フレームワーク"から取得したURI文字列を指定する必要があります。"ストレージ アクセス フレームワーク"から取得せずに作成されたURIはファイルを開けない可能性がありますのでご注意ください。

サポートするファイルの拡張子は.jpg、.jpeg、及び.pngです。
ただし、サポートする拡張子であっても、フォーマットによっては登録できない場合があります。
指定可能な最大ファイルサイズは786,432バイトです。
登録できる最大データサイズは横480×縦272ピクセル(px)です。

エラー　本メソッド呼び出し時にエラーが発生した場合は、PrinterExceptionをスローします。
エラーについての詳細は、「**5.1.5 PrinterExceptionクラス**」を参照してください。

説明　本メソッドを実行するとディスプレイはスタンバイモードになります。選択中のテンプレートは選択が解除されます。

注意　工場出荷時に登録されるデータは、品質向上のため予告無く追加・変更される場合があります。

UnregisterImageData　イメージデータの削除

ディスプレイに登録済みのイメージデータを削除します。

対象　iOS　プリンター経由のDSP-A01
Android　プリンター経由のDSP-A01、DSP-A01単体

構文
```
public void UnregisterImageData(int imageID);
```

パラメータ　imageID　イメージID
有効範囲は、0～63です。
指定したイメージIDにイメージデータが登録されていない場合は無視されます。

エラー　本メソッド呼び出し時にエラーが発生した場合は、PrinterExceptionをスローします。
エラーについての詳細は、「**5.1.5 PrinterExceptionクラス**」を参照してください。

説明　本メソッドを実行するとディスプレイはスタンバイモードになります。選択中のテンプレートは選択が解除されます。

イメージデータを削除しても使用済みのメモリは解放されません。使用済みのメモリは、Defragmentを実行すると再度利用できます。

RegisterSlideData　スライドデータの登録

ディスプレイにスライドデータを登録します。

対象　iOS　プリンター経由のDSP-A01
Android　プリンター経由のDSP-A01、DSP-A01単体

構文
```
public void RegisterSlideData(int slideID,
                               string label,
                               string fileName);
```

パラメータ　slideID　スライドID
有効範囲は、0～91です。
スライドID80～90はシステムで利用していますので、指定しないでください。

label　スライド名
登録するスライドデータに識別用の名前を付けることができます。
指定可能な文字は、英数字（'0'～'9'、'A'～'Z'、'a'～'z'）などのASCII文字コード20h（スペース）～7Eh（チルダ）の文字です。
ユニコードの00A5h（'¥'）は用いないでください。
指定可能な文字数は0～32文字です。
labelは省略することも可能です。省略する場合はnullを指定してください。
指定したスライド名はGetDisplayResponseで取得できます。

fileName　ファイルパス
登録するスライドデータのファイル名を指定してください。
入力可能な形式について下記に説明します。

iOSの場合：
使用するファイルパス文字列

Androidの場合：

- Javaの標準的なクラス「java.io.File」が扱う絶対パス文字列
  Android 10 (API 29)以降をアプリのターゲットとする場合には、一部のファイルが直接扱えなくなりますのでご注意ください。
  詳細については、「3.4 注意事項 – 対象範囲別ストレージについて」を参照してください。

- Androidに用意されたクラス「android.net.Uri」が扱う下記スキーム名のURI文字列
  ・file://
  ・content://
  本パラメータには"ストレージ アクセス フレームワーク"から取得したURI文字列を指定する必要があります。"ストレージ アクセス フレームワーク"から取得せずに作成されたURIはファイルを開けない可能性がありますのでご注意ください。

サポートするファイルの拡張子は.jpg、.jpeg、及び.pngです。
ただし、サポートする拡張子であっても、フォーマットによっては登録できない場合があります。
指定可能な最大ファイルサイズは786,432バイトです。
登録できるデータサイズは横480×縦272ピクセル(px)固定です。

エラー　本メソッド呼び出し時にエラーが発生した場合は、PrinterExceptionをスローします。
エラーについての詳細は、「**5.1.5 PrinterExceptionクラス**」を参照してください。

説明　本メソッドを実行するとディスプレイはスタンバイモードになります。選択中のテンプレートは選択が解除されます。

登録したスライドデータを表示する場合はShowSlideを実行してください。
登録したスライドデータをテンプレートの背景として使用する場合はSelectTemplateでスライドIDを指定してください。

注意　工場出荷時に登録されるデータは、品質向上のため予告無く追加・変更される場合があります。

UnregisterSlideData　スライドデータの削除

ディスプレイに登録済みのスライドデータを削除します。

対象　iOS　プリンター経由のDSP-A01
Android　プリンター経由のDSP-A01、DSP-A01単体

構文　`public void UnregisterSlideData(int slideID);`

パラメータ　slideID　スライドID
有効範囲は、0～91です。
指定したスライドIDにスライドデータが登録されていない場合は無視されます。

エラー　本メソッド呼び出し時にエラーが発生した場合は、PrinterExceptionをスローします。
エラーについての詳細は、「**5.1.5 PrinterExceptionクラス**」を参照してください。

説明　本メソッドを実行するとディスプレイはスタンバイモードになります。選択中のテンプレートは選択が解除されます。

RegisterUserDefinedCharacter　外字の登録

ディスプレイに外字を登録します。

対象　iOS　プリンター経由のDSP-A01
Android　プリンター経由のDSP-A01、DSP-A01単体

構文　`public void RegisterUserDefinedCharacter(string fileName);`

パラメータ　fileName　ファイルパス
登録する外字のファイル名を指定してください。
入力可能な形式について下記に説明します。

iOSの場合：
使用するファイルパス文字列

Androidの場合：

- Javaの標準的なクラス「java.io.File」が扱う絶対パス文字列
  Android 10 (API 29)以降をアプリのターゲットとする場合には、一部のファイルが直接扱えなくなりますのでご注意ください。
  詳細については、「3.4 注意事項 - 対象範囲別ストレージについて」を参照してください。

- Androidに用意されたクラス「android.net.Uri」が扱う下記スキーム名のURI文字列
  - ・file://
  - ・content://

  本パラメータには"ストレージ アクセス フレームワーク"から取得したURI文字列を指定する必要があります。"ストレージ アクセス フレームワーク"から取得せずに作成されたURIはファイルを開けない可能性がありますのでご注意ください。

サポートするファイルの拡張子は.binです。
外字データについては、「ディスプレイ用技術説明書」のディスプレイコマンド「外字の登録」を参照してください。

エラー　本メソッド呼び出し時にエラーが発生した場合は、PrinterExceptionをスローします。
エラーについての詳細は、「**5.1.5 PrinterExceptionクラス**」を参照してください。

説明　本メソッドを実行するとディスプレイはスタンバイモードになります。選択中のテンプレートは選択が解除されます。
外字が登録済みの状態で本メソッドを実行した場合は、外字は上書きされます。

外字の表示にはエンコード指定がShift_JISのテンプレートを使用してください。
外字を表示する前にCodePageに**CODE_PAGE_KATAKANA**を指定してください。
外字を表示する際はSetTemplateTextDataのtextに指定可能な文字コードを指定してください。指定可能な文字コードはE000h～E05Dhです。

UnregisterUserDefinedCharacter　外字の削除

ディスプレイに登録済みの外字を削除します。

対象　iOS　プリンター経由のDSP-A01
Android　プリンター経由のDSP-A01、DSP-A01単体

構文
```
public void UnregisterUserDefinedCharacter();
```

エラー　本メソッド呼び出し時にエラーが発生した場合は、PrinterExceptionをスローします。
エラーについての詳細は、「**5.1.5 PrinterExceptionクラス**」を参照してください。

説明　本メソッドを実行するとディスプレイはスタンバイモードになります。選択中のテンプレートは選択が解除されます。

登録されている外字は全て削除されます。
外字を削除しても使用済みのメモリは解放されません。使用済みのメモリは、Defragmentを実行すると再度利用できます。

RegisterOptionFont　　オプションフォントの登録

ディスプレイにオプションフォントを登録します。

対象　　iOS　　プリンター経由のDSP-A01
　　Android　　プリンター経由のDSP-A01、DSP-A01単体

構文
```
public void RegisterOptionFont(int startCode,
                               int endCode,
                               int width,
                               int height,
                               string fileName);
```

パラメータ

startCode　　登録開始文字コード
有効範囲は、20h～FFhのASCII文字コードです。

endCode　　登録終了文字コード
有効範囲は、20h～FFhのASCII文字コードです。

width　　文字幅(ピクセル:px)
有効範囲は、1～255です。

height　　文字高さ(ピクセル:px)
有効範囲は、1～255です。

fileName　　ファイルパス
登録するオプションフォントのファイル名を指定してください。
入力可能な形式について下記に説明します。

iOSの場合:
使用するファイルパス文字列

Androidの場合:

- Javaの標準的なクラス「java.io.File」が扱う絶対パス文字列
  Android 10 (API 29)以降をアプリのターゲットとする場合には、一部のファイルが直接扱えなくなりますのでご注意ください。
  詳細については、「3.4 注意事項 - 対象範囲別ストレージについて」を参照してください。

- Androidに用意されたクラス「android.net.Uri」が扱う下記スキーム名のURI文字列
  ・file://
  ・content://
  本パラメータには"ストレージ アクセス フレームワーク"から取得したURI文字列を指定する必要があります。"ストレージ アクセス フレームワーク"から取得せずに作成されたURIはファイルを開けない可能性がありますのでご注意ください。

サポートするファイルの拡張子は.binです。
オプションフォントデータについては、「ディスプレイ用技術説明書」のディスプレイコマンド「オプションフォントの登録」を参照してください。

エラー　本メソッド呼び出し時にエラーが発生した場合は、PrinterExceptionをスローします。
エラーについての詳細は、「**5.1.5 PrinterExceptionクラス**」を参照してください。

説明　本メソッドを実行するとディスプレイはスタンバイモードになります。選択中のテンプレートは選択が解除されます。

オプションフォントが登録済みの状態で本メソッドを実行した場合は、登録済みのオプションフォントは削除され、新たにメモリ領域を確保してオプションフォントを登録します。
登録済みのオプションフォントが削除されても使用済みのメモリは解放されません。使用済みのメモリは、Defragmentを実行すると再度利用できます。

UnregisterOptionFont　オプションフォントの削除

ディスプレイに登録済みのオプションフォントを削除します。

対象　iOS　プリンター経由のDSP-A01
Android　プリンター経由のDSP-A01、DSP-A01単体

構文
```
public void UnregisterOptionFont();
```

エラー　本メソッド呼び出し時にエラーが発生した場合は、PrinterExceptionをスローします。
エラーについての詳細は、「**5.1.5 PrinterExceptionクラス**」を参照してください。

説明　本メソッドを実行するとディスプレイはスタンバイモードになります。選択中のテンプレートは選択が解除されます。

登録されているオプションフォントは全て削除されます。
オプションフォントを削除しても使用済みのメモリは解放されません。使用済みのメモリは、Defragmentを実行すると再度利用できます。

ControlMacroRegistration　マクロ登録の開始・終了

ディスプレイで使用するマクロ登録の開始・終了を指定します。

対象　iOS　プリンター経由のDSP-A01
Android　プリンター経由のDSP-A01、DSP-A01単体

構文
```
public void ControlMacroRegistration(int macroID,
                                      MacroRegistrationFunction control);
```

パラメータ　macroID　マクロID
有効範囲は、-1～127です。
マクロID120～126はシステムで利用していますので、選択しないでください。

control　マクロ登録処理
利用可能な定数は、
「5.1.1⑯ マクロ登録処理(MacroRegistrationFunction)」を参照してください。

エラー　本メソッド呼び出し時にエラーが発生した場合は、PrinterExceptionをスローします。
エラーについての詳細は、「**5.1.5 PrinterExceptionクラス**」を参照してください。

説明　　マクロ登録処理の手順は下記の通りです。

① マクロ登録処理を開始します。
macroIDに-1を指定してください。
controlに**MACRO_REGISTRATION_START**を指定してください。

② メソッドを実行します。
マクロ登録処理対象メソッドの場合、送信データのバッファリングを開始します。
バッファリング中に実行したマクロ登録処理対象メソッドの送信データは、プリンターへ送信せずにマクロデータバッファーにバッファリングします。バッファリングできる最大送信データのサイズは1024バイトです。
バッファリングされた送信データが最大サイズを超える場合、超えた時点のマクロ登録処理対象メソッドがエラーとなります。
エラーとなった場合は登録中のデータは破棄され、マクロモードが解除されます。
保持された送信データは、手順③でマクロ登録処理を終了してください。
マクロ登録処理対象以外のメソッドの場合は、送信データはバッファリングされずに即実行されます。

マクロ登録処理対象メソッドを下記に示します。
・ShowTemplate
・ShowSlide
・SelectTemplate
・SetTemplateImageData
・SelectTemplateTextObject
・SetTemplateTextAlignment
・SetTemplateTextLeftMargin
・SetTemplateTextLineSpacing
・SetTemplateTextBold
・SetTemplateTextUnderline
・SetTemplateTextSize
・SetTemplateTextFont
・SetTemplateTextRegisteredFont
・SetTemplateTextRightSpacing
・SetTemplateTextColor
・SetTemplateTextData
・SetTemplateBarcodeData
・SetTemplateQRCodeData

③ マクロ登録処理を終了します。
macroIDに登録したいマクロID(0～127)を指定してください。
controlに**MACRO_REGISTRATION_REGIST**を指定するとバッファリングされた送信データをプリンターへ送信します。バッファリングされた送信データはプリンター送信後も保持されます。本メソッドを実行するとディスプレイはスタンバイモードになります。
選択中のテンプレートは選択が解除されます。
保持されている送信データは下記の手順により破棄されます。
・**MACRO_REGISTRATION_CLEAR**の指定
・**MACRO_REGISTRATION_START**の指定
・Disconnectの実行

登録したマクロはExecuteMacroで実行できます。

登録済みのマクロを削除する手順は下記の通りです。

controlに**MACRO_REGISTRATION_START**を指定し、macroIDに-1を指定して本メソッドを呼び出してください。

controlに**MACRO_REGISTRATION_REGIST**を指定し、macroIDに削除するマクロIDを指定して本メソッドを呼び出してください。

本メソッドを実行するとディスプレイはスタンバイモードになります。選択中のテンプレートは選択が解除されます。

注意　工場出荷時に登録されるデータは、品質向上のため予告無く追加・変更される場合があります。

GetDisplayResponse　ディスプレイからの各種応答取得

ディスプレイからの応答データを取得します。

対象　iOS　プリンター経由のDSP-A01
Android　プリンター経由のDSP-A01、DSP-A01単体

構文
```
public void GetDisplayResponse<T>(DisplayResponseId responseId,
                                  object param,
                                  out T[] response);
```

パラメータ

responseId　ディスプレイ応答種別定数
利用可能な定数は、
「5.1.1⑰ ディスプレイ応答種別（DisplayResponseId）」を参照してください。

param　コマンドパラメータ
ディスプレイ応答種別定数により指定する値が異なります。
指定する値の説明は、下表を参照してください。

response　取得した応答データを格納するバッファー
ディスプレイ応答種別定数によりバッファーの型が異なります。
バッファーの型は、下表を参照してください。

応答種別定数	
パラメータ	説明
DISPLAY_RESPONSE_REQUEST（実行応答リクエスト）	
param	int型で0～15（00h～0Fh）を指定してください。
response	int型の配列を指定してください。 応答取得成功時には、実行応答リクエストの応答コードが64～79（40h～4Fh）で格納されます。
DISPLAY_RESPONSE_USER_AREA（ユーザ領域の残り容量の送信）	
param	nullを指定してください。
response	int型の配列を指定してください。 応答取得成功時には、ユーザ領域の残り容量（単位：バイト）が数値で格納されます。

応答種別定数		
	パラメータ	説明
DISPLAY_RESPONSE_TEMPLATE_ID_LIST（テンプレートIDの送信）		
	param	nullを指定してください。
	response	int型の配列を指定してください。 応答取得成功時には、登録済みのテンプレートIDがNSIntegerの配列で格納されます。
DISPLAY_RESPONSE_IMAGE_ID_LIST（イメージIDの送信）		
	param	nullを指定してください。
	response	int型の配列を指定してください。 応答取得成功時には、登録済みのイメージIDがNSIntegerの配列で格納されます。
DISPLAY_RESPONSE_SLIDE_ID_LIST（スライドIDの送信）		
	param	nullを指定してください。
	response	int型の配列を指定してください。 応答取得成功時には、登録済みのスライドIDがNSIntegerの配列で格納されます。
DISPLAY_RESPONSE_TEMPLATE_LABEL（テンプレート名の送信）		
	param	int型で0～127（00h～7Fh）を指定してください。
	response	string型の配列を指定してください。 応答取得成功時には、テンプレートの登録時に指定したテンプレート名が文字列で格納されます。
DISPLAY_RESPONSE_IMAGE_LABEL（イメージ名の送信）		
	param	int型で0～63（00h～3Fh）を指定してください。
	response	string型の配列を指定してください。 応答取得成功時には、イメージファイルの登録時に指定したイメージ名が文字列で格納されます。
DISPLAY_RESPONSE_SLIDE_LABEL（スライド名の送信）		
	param	int型で0～91（00h～5bh）を指定してください。
	response	string型の配列を指定してください。 応答取得成功時には、スライドデータの登録時に指定したスライド名が文字列で格納されます。

エラー　本メソッド呼び出し時にエラーが発生した場合は、PrinterExceptionをスローします。
エラーについての詳細は、「**5.1.5 PrinterExceptionクラス**」を参照してください。

SendBinary　バイナリデータの送信

バイナリデータをプリンターに送信します。

対象　プラットフォーム共通　全デバイス

構文　`public void SendBinary(byte[] data);`

パラメータ　data　デバイスに送信するバイナリデータ
一度に指定可能なデータサイズは3MB（3145728バイト）です。

エラー　本メソッド呼び出し時にエラーが発生した場合は、PrinterExceptionをスローします。
エラーについての詳細は、「**5.1.5 PrinterExceptionクラス**」を参照してください。

説明　　本メソッドは、指定されたバイナリデータを変換せずにデバイスに送信します。

本メソッドでディスプレイコマンドをバイナリデータとして送信することにより、ライブラリではサポートされていないディスプレイ機能を利用できます。ただし、本メソッドではディスプレイから応答を取得するコマンドには対応しておりません。

SendDataFile　　指定ファイルの送信

ファイルデータを送信します。

対象　　プラットフォーム共通　　全デバイス

構文
```
public void SendDataFile(string fileName,
                         PrintAlignment alignment,
                         Dithering dithering);
```

パラメータ　fileName　　デバイスに送信するデータファイルパス
入力可能な形式について下記に説明します。

iOSの場合：
使用するファイルパス文字列

Androidの場合：

- Javaの標準的なクラス「java.io.File」が扱う絶対パス文字列
  Android 10 (API 29)以降をアプリのターゲットとする場合には、一部のファイルが直接扱えなくなりますのでご注意ください。
  詳細については、「3.4 注意事項 – 対象範囲別ストレージについて」を参照してください。

- Androidに用意されたクラス「android.net.Uri」が扱う下記スキーム名のURI文字列
  ・file://
  ・content://
  本パラメータには"ストレージ アクセス フレームワーク"から取得したURI文字列を指定する必要があります。"ストレージ アクセス フレームワーク"から取得せずに作成されたURIはファイルを開けない可能性がありますのでご注意ください。

指定可能なファイルサイズは、最大16MB（16777216バイト）です。
送信可能なファイル拡張子とファイルの送信について下記に説明します。

- bin、.dat
  バイナリデータとして変換せずにデバイスに送信します。

alignment　　位置揃え
ディスプレイの機能に反映されません。

dithering　　ディザリング
ディスプレイの機能に反映されません。

エラー　　本メソッド呼び出し時にエラーが発生した場合は、PrinterExceptionをスローします。
エラーについての詳細は、「**5.1.5 PrinterExceptionクラス**」を参照してください。

(4) プロパティ詳細

SendTimeout　送信タイムアウト時間の設定/取得

データ送信時のタイムアウト時間を設定または取得します。

対象　プラットフォーム共通　全デバイス

構文　`public int SendTimeout {get; set;}`

有効範囲　iOSの場合：
100～300000（ミリ秒：ms）
100未満の値を指定した場合は、100msに設定されます。
300000を超える値を指定した場合は、300000msに設定されます。
Androidの場合：
100～90000（ミリ秒：ms）
有効範囲外の値を指定した場合は、10000msに設定されます。

初期値　10000

説明　本プロパティは`IsConnect`がtrue/falseに関わらず設定または取得が可能です。

設定したタイムアウト時間が有効になるのは、次回データ送信時となります。

ReceiveTimeout　受信タイムアウト時間の設定/取得

データ受信時のタイムアウト時間を設定または取得します。

対象　プラットフォーム共通　全デバイス

構文　`public int ReceiveTimeout {get; set;}`

有効範囲　iOSの場合：
100～300000（ミリ秒：ms）
100未満の値を指定した場合は、100msに設定されます。
300000を超える値を指定した場合は、300000msに設定されます。
Androidの場合：
100～90000（ミリ秒：ms）
有効範囲外の値を指定した場合は、10000msに設定されます。

初期値　10000

説明　本プロパティは`IsConnect`がtrue/falseに関わらず設定または取得が可能です。

設定したタイムアウト時間が有効になるのは、次回データ受信時となります。

InternationalCharacter　国際文字セットの設定/取得

国際文字セットの値を設定または取得します。

対象　プラットフォーム共通　全デバイス

構文　`public InternationalCharacter InternationalCharacter {get; set;}`

説明　利用可能な定数は、「5.1.1⑱ 国際文字セット(InternationalCharacter)」を参照してください。無効な値を指定した場合は無視されます。

本プロパティが未設定の場合は、デバイスの言語設定により下記の国際文字セットとなります。
デバイスの言語設定が日本の場合:　**COUNTRY_JAPAN**
デバイスの言語設定が日本以外の場合:　**COUNTRY_USA**

SendDataFileまたはSetTemplateTextDataにてテキストデータを送信した場合は、下記に示す文字コードの表示結果が異なります。表示される文字に関しては、「付録A 文字セット」を参照してください。

国際文字セット設定により表示結果が異なる文字コード
0x23、0x24、0x40、0x5B、0x5C、0x5D、0x5E、0x60、0x7B、0x7C、0x7D、0x7E

CodePage　コードページの設定/取得

コードページの値を設定または取得します。

対象　プラットフォーム共通　全デバイス

構文　`public CodePage CodePage {get; set;}`

説明　利用可能な定数は、「5.1.1⑲ コードページ(CodePage)」を参照してください。無効な値を指定した場合は無視されます。

本プロパティが未設定の場合は、デバイスの言語設定により下記のコードページとなります。
デバイスの言語設定が日本の場合:　**CODE_PAGE_KATAKANA**
デバイスの言語設定が日本以外の場合:　**CODE_PAGE_1252**

SendDataFileまたはSetTemplateTextDataにてテキストデータを送信する場合に利用するエンコーダが変更されます。表示される文字に関しては、「付録A 文字セット」を参照してください。

DeviceModel　デバイスモデルの取得

通信中のプリンターのモデル値を取得します。

対象　プラットフォーム共通　全デバイス

構文　`public DeviceModel DeviceModel {get;}`

初期値　DeviceModel.DEVICE_MODEL_RP_E10

戻り値　　「5.1.1① デバイスモデル（DeviceModel）」を参照してください。

PortType　接続中ポート種別の取得

プリンターとの接続に利用しているポート種別の値を取得します。

対象　　プラットフォーム共通　　全デバイス

構文　　public PortType PortType {get;}

初期値　　PortType.PORT_TYPE_BLUETOOTH

戻り値　　「5.1.1② ポート種別（PortType）」を参照してください。

IsConnect　プリンターとの通信状態の確認

プリンターとの通信状態を確認します。

対象　　プラットフォーム共通　　全デバイス

構文　　public bool IsConnect {get;}

戻り値　　true　プリンターと通信中
false　プリンターと未接続

説明　　本プロパティは、Connect状態をBOOL値で保持します。
Connectに成功すると、本プロパティはtrueになります。Connect後、Disconnectに成功すると、本プロパティはfalseになります。

SocketKeepingTime　ソケット維持時間の設定/取得

ソケット維持時間の設定または取得します。

対象　　iOS　　プリンター経由のDSP-A01
Android　　プリンター経由のDSP-A01

構文　　public int SocketKeepingTime {get; set;}

有効範囲　　60000～300000（ミリ秒：ms）
60000未満の値を指定した場合は、60000msに設定されます。
300000を超える値を指定した場合は、300000msに設定されます。

初期値　　300000

説明　　本プロパティはIsConnectがtrue/falseに関わらず設定または取得が可能です。

ソケット維持時間には、接続するプリンターの受信タイムアウト時間設定と等しい時間を指定してください。受信タイムアウト時間設定は、iOSの場合はApp StoreのiOSアプリ「SII Printer Utility」、Androidの場合はGoogle PlayのAndroidアプリ「SII Printer Utility」で変更できます。

設定したソケット維持時間が有効になるのは、次回Connect実行時となります。

Version　　SDKバージョンの取得

SDKのバージョンを文字列で取得します。

対象　　プラットフォーム共通　　全デバイス

構文　　`public string Version {get;}`

戻り値　　2つのSDKバージョン文字列を返します。(例:"1.0.0 / 1.0.0")
1つ目:SII SDK for .NET MAUIのバージョン
2つ目:動作環境に対応したSII Print Class Libraryのバージョン

説明　　本メソッドは`IsConnect`がtrue/falseに関わらず取得が可能です。

5.1.3 DeviceInfoクラス

プリンターの探索メソッドで発見されたデバイス情報を格納するクラスです。取得できるデバイス情報については、「5.1.3(2) プロパティー覧」を参照してください。

(1) メソッド一覧

DeviceInfoクラスで提供されるメソッドの一覧を下記に示します。

名前	説明
DeviceInfo	デバイス情報クラスのコンストラクタ

(2) プロパティー覧

DeviceInfoクラスで提供されるプロパティの一覧を下記に示します。

名前	アクセス	説明
PortType	R	通信ポート種別の取得
DeviceName	R	デバイス名（プリンターモデル名）の取得
BluetoothAddress	R	Bluetoothアドレスの取得
MacAddress	R	MACアドレスの取得
IpAddress	R	IPアドレスの取得
IsBonded	R	ペアリング状態の取得
DevicePath	R	デバイスパスの取得

(3) メソッド詳細

DeviceInfo　コンストラクタ

構文(a)は、Bluetooth接続で使用するデバイス情報クラスのコンストラクタです。
構文(b)は、TCP/IP接続で使用するデバイス情報クラスのコンストラクタです。
構文(c)は、USB接続で使用するデバイス情報クラスのコンストラクタです。

対象　プラットフォーム共通　全デバイス

構文

(a)
```
public DeviceInfo(PortType portType,
                  string deviceName,
                  string bluetoothAddress,
                  bool isBonded);
```

(b)
```
public DeviceInfo(PortType portType,
                  string deviceName,
                  string macAddress,
                  string ipAddress);
```

(c)
```
public DeviceInfo(PortType portType,
                  string deviceName,
                  string devicePath);
```

説明　本メソッドは、StartDiscoveryDeviceで発見されたデバイス情報を格納します。

iOSでは、構文(a)、(b)が有効です。

Androidでは、プリンター経由でDSP-A01を使用している場合は構文(a)、(b)、(c)が有効です。
DSP-A01単体で使用している場合は、構文(c)のみ有効です。

(4) プロパティ詳細

PortType　通信ポート種別の取得

対象　プラットフォーム共通　全デバイス

構文　public PortType PortType {get;}

説明　本プロパティは、StartDiscoveryDeviceで発見されたデバイス情報から通信ポート種別を取得します。

DeviceName　デバイス名（プリンターモデル名）の取得

対象　プラットフォーム共通　全デバイス

構文　public string DeviceName {get;}

説明　本プロパティは、StartDiscoveryDeviceで発見されたデバイス情報からデバイス名（プリンターモデル名）を取得します。

BluetoothAddress　Bluetoothアドレスの取得

対象　プラットフォーム共通　全プリンター、プリンター経由のDSP-A01

構文　public string BluetoothAddress {get;}

説明　本プロパティは、StartDiscoveryDeviceで発見されたプリンター情報からBluetoothアドレスを取得します。

MacAddress　MACアドレスの取得

対象　プラットフォーム共通　全プリンター、プリンター経由のDSP-A01

構文　public string MacAddress {get;}

説明　本プロパティは、StartDiscoveryDeviceで発見されたデバイス情報からMACアドレスを取得します。

IpAddress　IPアドレスの取得

対象　プラットフォーム共通　全プリンター、プリンター経由のDSP-A01

構文　public string IpAddress {get;}

説明　本プロパティは、StartDiscoveryDeviceで発見されたデバイス情報からIPアドレスを取得します。

IsBonded　ペアリング状態の取得

対象　プラットフォーム共通　全プリンター、プリンター経由のDSP-A01

構文　`public bool IsBonded {get;}`

説明　本プロパティは、StartDiscoveryDeviceで発見されたデバイス情報からペアリング状態を取得します。

iOSでは、常にtrueを返します。

DevicePath　デバイスパスの取得

対象　Android　プリンター経由のDSP-A01、DSP-A01単体

構文　`public string DevicePath {get;}`

説明　本プロパティは、StartDiscoveryDeviceで発見されたデバイス情報からUSBデバイスファイルのパスの文字列を取得します。

5.1.4 PrinterIdentifierクラス

PrinterIdentifierクラスは、GetPrinterResponseで取得されたプリンターのプリンタIDを格納するクラスです。

(1) プロパティー覧

PrinterIdentifierクラスで提供されるプロパティの一覧を下記に示します。

名前	アクセス	説明
PrinterModelID	R	機種IDの取得
TypeID	R	タイプIDの取得
RomVersionID	R	ROMバージョンIDの取得
MainFirmwareVersion	R	ファームウェアバージョン(main)の取得
Manufacturer	R	メーカ名の取得
ModelName	R	モデル名の取得
SerialNumber	R	シリアル番号の取得
BootFirmwareVersion	R	ファームウェアバージョン(boot)の取得
BootFirmwareChecksum	R	ファームウェアチェックサム(boot)の取得
MainFirmwareChecksum	R	ファームウェアチェックサム(main)の取得
FirmwareChecksum	R	ファームウェアチェックサム(main + boot)の取得

(2) プロパティ詳細

PrinterModelID　機種IDの取得

対象　プラットフォーム共通　全プリンター、プリンター経由のDSP-A01

構文　`public byte PrinterModelID {get;}`

説明　本プロパティは、GetPrinterResponseで取得されたプリンターのプリンタIDから機種IDを取得します。

TypeID　タイプIDの取得

対象　プラットフォーム共通　全プリンター、プリンター経由のDSP-A01

構文　`public byte TypeID {get;}`

説明　本プロパティは、GetPrinterResponseで取得されたプリンターのプリンタIDからタイプIDを取得します。

RomVersionID　ROMバージョンIDの取得

対象　プラットフォーム共通　全プリンター、プリンター経由のDSP-A01

構文　public byte RomVersionID {get;}

説明　本プロパティは、GetPrinterResponseで取得されたプリンターのプリンタIDからROMバージョンIDを取得します。

MainFirmwareVersion　ファームウェアバージョン(main)の取得

対象　プラットフォーム共通　全プリンター、プリンター経由のDSP-A01

構文　public String MainFirmwareVersion {get;}

説明　本プロパティは、GetPrinterResponseで取得されたプリンターのプリンタIDからファームウェアバージョン(main)の文字列を取得します。

Manufacturer　メーカ名の取得

対象　プラットフォーム共通　全プリンター、プリンター経由のDSP-A01

構文　public String Manufacturer {get;}

説明　本プロパティは、GetPrinterResponseで取得されたプリンターのプリンタIDからメーカ名の文字列を取得します。

ModelName　モデル名の取得

対象　プラットフォーム共通　全プリンター、プリンター経由のDSP-A01

構文　public String ModelName {get;}

説明　本プロパティは、GetPrinterResponseで取得されたプリンターのプリンタIDからモデル名の文字列を取得します。

SerialNumber　シリアル番号の取得

対象　プラットフォーム共通　全プリンター、プリンター経由のDSP-A01

構文　public String SerialNumber {get;}

説明　本プロパティは、GetPrinterResponseで取得されたプリンターのプリンタIDからシリアル番号の文字列を取得します。

BootFirmwareVersion　ファームウェアバージョン(boot)の取得

対象　プラットフォーム共通　全プリンター、プリンター経由のDSP-A01

構文　public String BootFirmwareVersion {get;}

説明　本プロパティは、GetPrinterResponseで取得されたプリンターのプリンタIDからファームウェアバージョン(boot)の文字列を取得します。

BootFirmwareChecksum　ファームウェアチェックサム(boot)の取得

対象　プラットフォーム共通　全プリンター、プリンター経由のDSP-A01

構文　public byte[] BootFirmwareChecksum {get;}

説明　本プロパティは、GetPrinterResponseで取得されたプリンターのプリンタIDからファームウェアチェックサム(boot)を取得します。

MainFirmwareChecksum　ファームウェアチェックサム(main)の取得

対象　プラットフォーム共通　全プリンター、プリンター経由のDSP-A01

構文　public byte[] MainFirmwareChecksum {get;}

説明　本プロパティは、GetPrinterResponseで取得されたプリンターのプリンタIDからファームウェアチェックサム(main)を取得します。

FirmwareChecksum　ファームウェアチェックサム(main + boot)の取得

対象　プラットフォーム共通　全プリンター、プリンター経由のDSP-A01

構文　public byte[] FirmwareChecksum {get;}

説明　本プロパティは、GetPrinterResponseで取得されたプリンターのプリンタIDからファームウェアチェックサム(main + boot)を取得します。

5.1.5 PrinterExceptionクラス

(1) メソッド一覧

PrinterExceptionクラスで提供されるメソッドの一覧を下記に示します。

名前	説明
PrinterException	コンストラクタ

(2) プロパティ一覧

PrinterExceptionクラスで提供されるプロパティの一覧を下記に示します。

名前	アクセス	説明
HResult	R	エラーコードの取得
Message	R	エラーメッセージの取得

(3) 定数一覧

① エラーコード

エラーコードの取得で利用される定数を下記に示します。

定数名	説明	値
ERROR_ACCESS_DENIED	ハンドルの取得に失敗しました。*1	-1
	利用できないポートが指定されました。	
	サポートされていないメソッドが指定されました。	
ERROR_SHARING_VIOLATION	すでにオープン済みのポートが指定されました。	-11
ERROR_PORT_NOT_OPENED	ポートがオープンされていません。	-12
ERROR_DEVICE_NOT_CONNECTED	Bluetooth接続に問題があります。	-21
	USB接続に問題があります。	
	指定されたIPアドレスのプリンターが存在しません。	
ERROR_OFFLINE	切断状態またはプリンターがオフラインです。	-22
ERROR_EXTERNAL_DEVICE_NOT_CONNECTED	ディスプレイが接続されていません。	-23

定数名	説明	値
ERROR_DEVICE_INITIALIZE_FAILED	プリンターの設定変更に失敗しました。プリンターへのデータ送信が送信タイムアウト時間内に完了していないか、プリンターからのデータ受信が受信タイムアウト時間内に完了していません。	-31
ERROR_DATA_SIZE_ZERO	データサイズが0バイトのデータが指定されました。	-101
ERROR_OVER_MAX_DATA_SIZE	データサイズが最大値を超過しています。	-102
ERROR_DATA_SIZE_INVALID	既定のデータサイズではありません。	-103
ERROR_ENCODE_FAILED	テキストデータのエンコードでエラーが発生しました。*1	-111
ERROR_TIMEOUT	送信タイムアウトが発生しました。 受信タイムアウトが発生しました。	-201
ERROR_FILE_NOT_FOUND	指定されたファイルが存在しません。	-301
ERROR_FILE_USED	指定されたファイルは別プロセスで使用中です。	-302
ERROR_FILE_INVALID	指定されたファイルは無効です。	-303
ERROR_LOW_MEMORY	イメージファイルの読み込みでメモリ不足が発生しました。	-311
ERROR_OVER_MAX_IMAGE	イメージファイルの幅と高さのいずれか、または両方が表示可能な最大ドット数を超過しています。	-312
ERROR_LOW_USER_AREA	ユーザ領域の残り容量が不足しています。	-401
ERROR_LOW_EXTERNAL_RAM	RAMの残り容量が不足しています。	-402
ERROR_NOT_REGISTERD	テンプレートが登録されていません。 イメージファイルが登録されていません。 スライドデータが登録されていません。 オプションフォントが登録されていません。 外字が登録されていません。	-403
ERROR_NOT_UNREGISTERD	テンプレートが削除されていません。 イメージファイルが削除されていません。 スライドデータが削除されていません。 オプションフォントが削除されていません。 外字が削除されていません。	-404
ERROR_INVALID_DATA	指定されたデータは無効です。	-503
ERROR_INVALID_PARAM	指定されたパラメータは無効です。	-9999

*1: 異常処理の可能性があります。

(4) メソッド詳細

PrinterException　コンストラクタ

PrinterManagerクラスのAPI呼び出し時にスローされる例外クラスです。

構文　　public PrinterException(int code, string message);

(5) プロパティ詳細

HResult　エラーコードの取得

対象　　iOS　　プリンター経由のDSP-A01
　　Android　　プリンター経由のDSP-A01、DSP-A01単体

構文　　public int HResult {get;}

説明　　本プロパティは、エラーコードをint型で取得します。

戻り値　　「5.1.5(3) 定数一覧」を参照してください。

Message　エラーメッセージの取得

対象　　iOS　　プリンター経由のDSP-A01
　　Android　　プリンター経由のDSP-A01、DSP-A01単体

構文　　public string Message {get;}

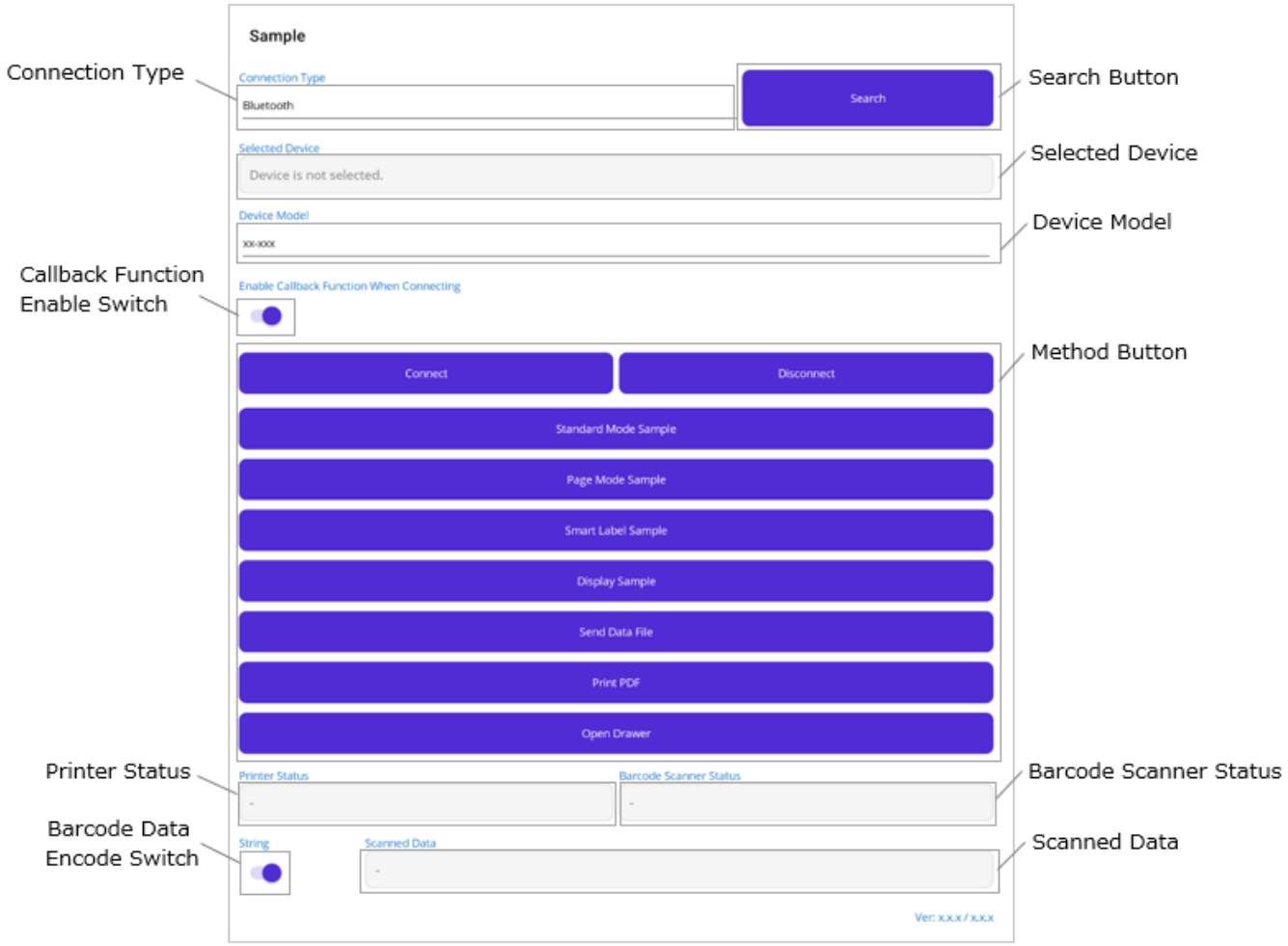
説明　　本プロパティは、HResultの内容を補足する文字列が取得できます。

6章　サンプルプログラム

本章では、SDKで提供するサンプルプログラムについて説明します。

6.1　画面構成

SDKには、.NET MAUIアプリプロジェクト形式のサンプルプログラムであるSampleが含まれています。
Sampleの画面について説明します。



項目	説明
Connection Type	プリンターへの接続種別を選択します。
Search Button	選択した接続種別のデバイスの探索を開始します。探索終了後、見つかったデバイス一覧が表示されます。 一覧の中からデバイス名をタップすると、デバイスを選択できます。 Bluetooth接続の場合： Bluetooth接続のデバイス一覧が表示されます。 iOSの場合は、ペアリング済みのBluetoothデバイスのみ表示されます。 USB接続の場合： USB接続中のデバイスが表示されます。 TCP/IP接続の場合： 接続可能なTCP/IP接続のデバイス一覧が表示されます。
Selected Device	選択中のデバイスを表示します。
Device Model	接続するデバイスの機種を表示します。 タップして接続するデバイスの機種を選択してください。
Callback Function Enable Switch	デバイスとの接続時にコールバック関数を有効にするか選択します。 On： 接続時にコールバック関数を開始します。 Off： コールバック関数は応答しません。
Method Button*1	Connect及びDisconnectを行うメソッドボタンの他、各メソッドを組み合わせたサンプルの印字やステータスの取得ができます。
Printer Status	プリンターのステータスを表示します。 [Callback Function Enable Switch]がOnの場合は、最新のプリンターステータスが表示されます。
Barcode Scanner Status	バーコードスキャナーの接続状態を表示します。 [Callback Function Enable Switch]がOnの場合は、バーコードスキャナーの接続状態が表示されます。 バーコードスキャナーをサポートしていないデバイスではOfflineが表示されます。
Barcode Data Encode Switch	バーコードスキャナーで読み取ったバーコードデータの変換を選択します。 On： 読み取ったバイナリデータを文字列に変換して表示します。 Off： 読み取ったバイナリデータをそのまま表示します。
Scanned Data	バーコードスキャナーで読み取ったバーコードデータを表示します。 [Callback Function Enable Switch]がOnの場合は、バーコードスキャナーで読み取ったバーコードデータを表示します。 バーコードスキャナーをサポートしていないデバイスでは表示されません。

*1: サポートする機能は機種により異なります。サポートしている機能以外は動作しません。

6.2 注意事項

サンプルプログラムは予告なしに変更される場合があります。

サンプルプログラムに対しての動作保証及びサポートは行っておりません。

付録 A　文字セット

A.1 コードページ表（文字コードテーブル）

国際文字セットで**COUNTRY_USA**設定時のコードページを下記に示します。
設定する国際文字セットにより、特定の文字コードの印字、または表示結果が異なります。
特定の文字コードについては「A.2 国際文字セット」を参照してください。

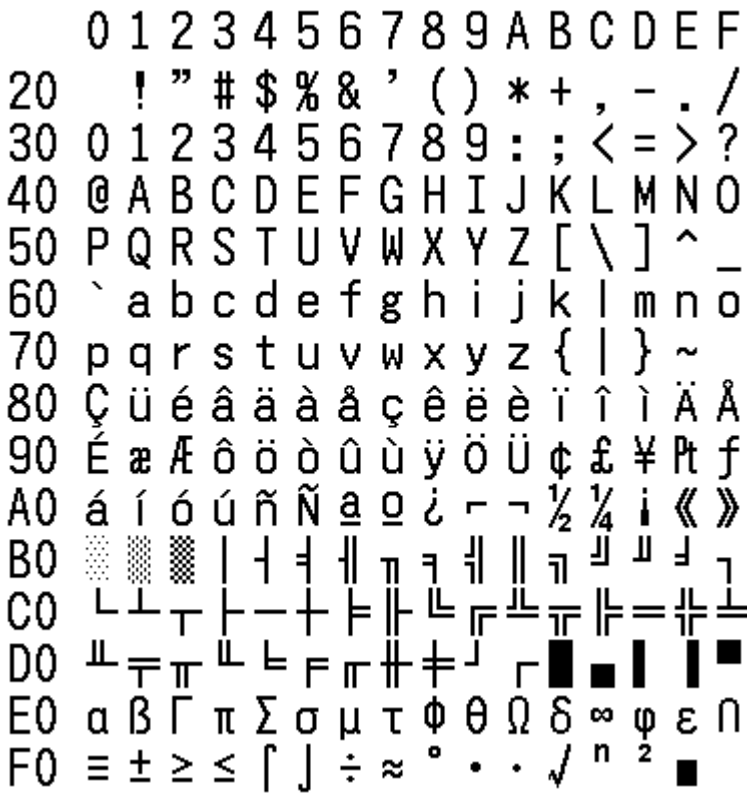


図 A-1 **CODE_PAGE_437** (USA, Standard Europe)

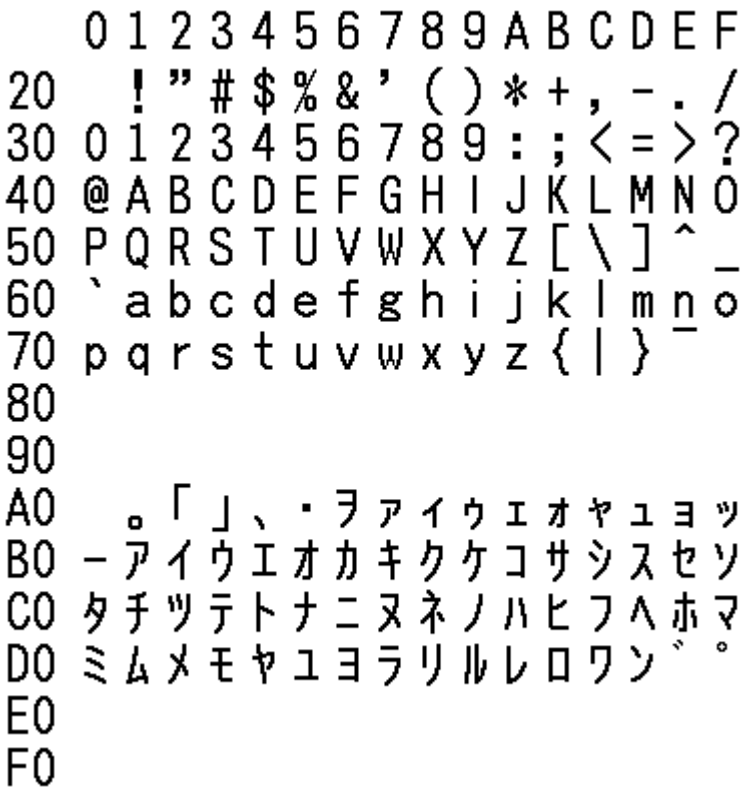


図 A-2 **CODE_PAGE_KATAKANA**

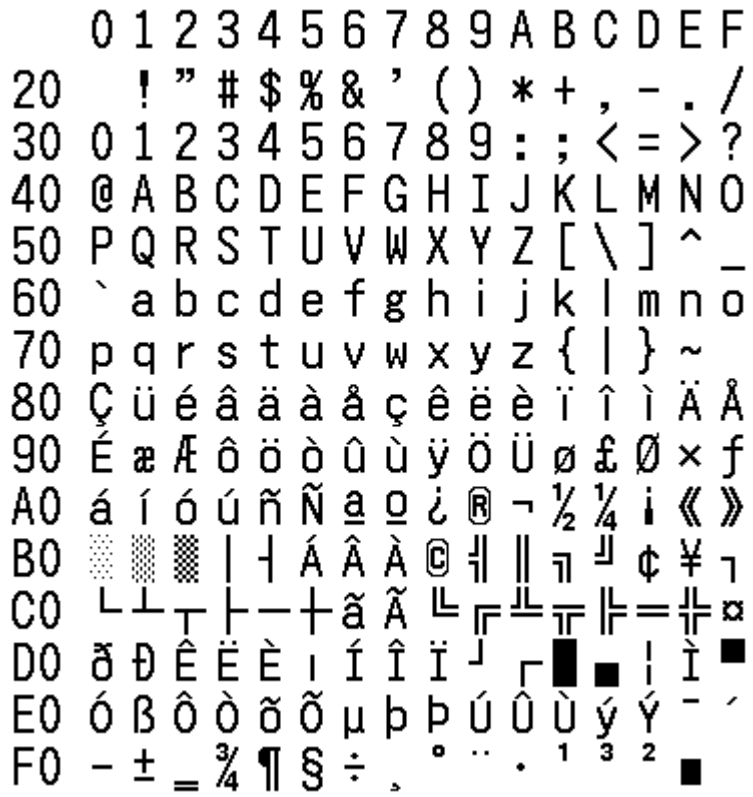


図 A-3 **CODE_PAGE_850** (Multilingual)

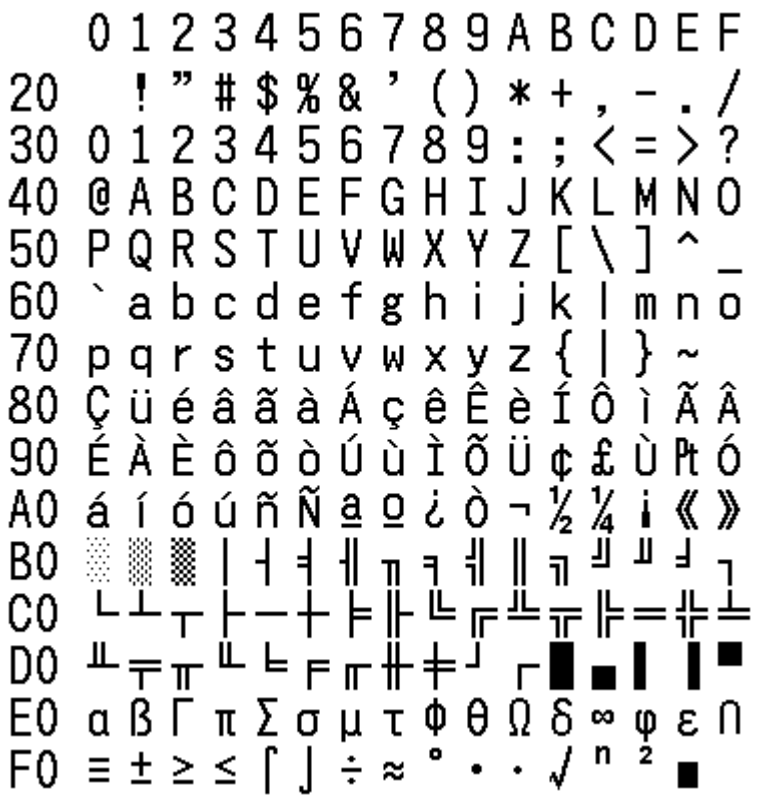


図 A-4 **CODE_PAGE_860** (Portuguese)

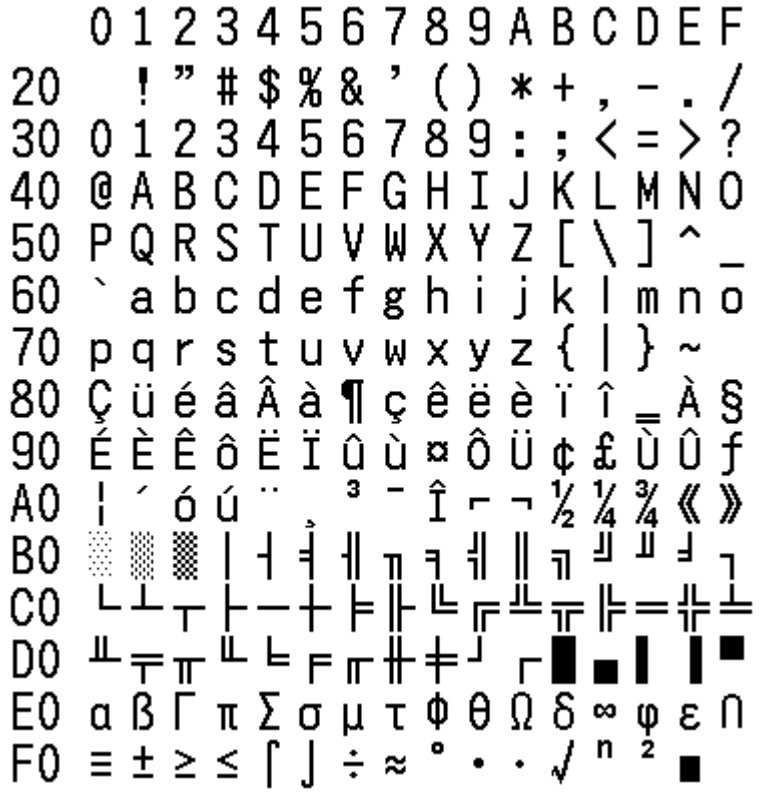


図 A-5 **CODE_PAGE_863** (Canadian-French)

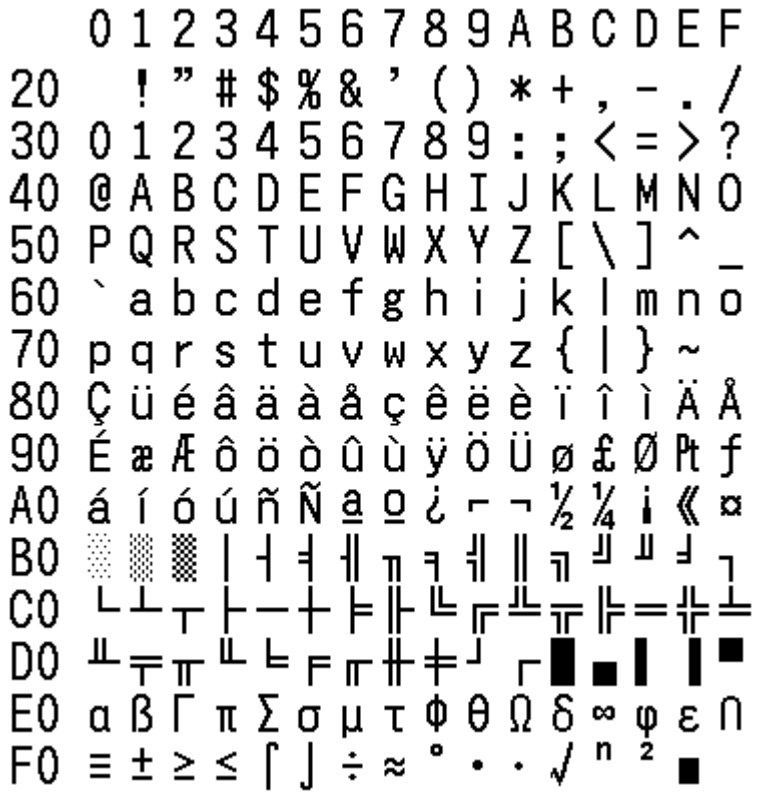


図 A-6 **CODE_PAGE_865** (Nordic)

RP-E10、RP-D10は**CODE_PAGE_857**をサポートしていません。

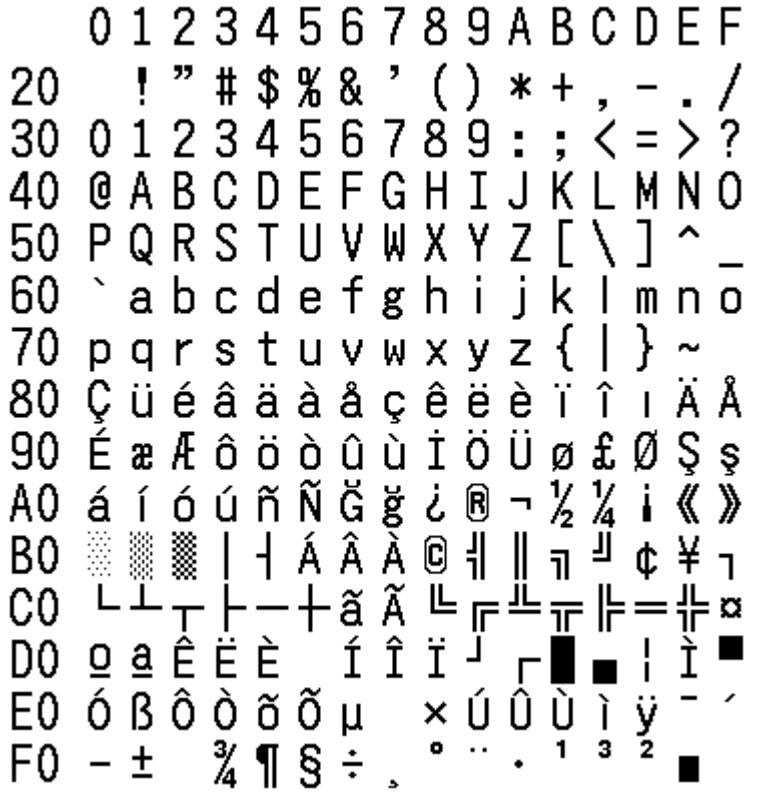
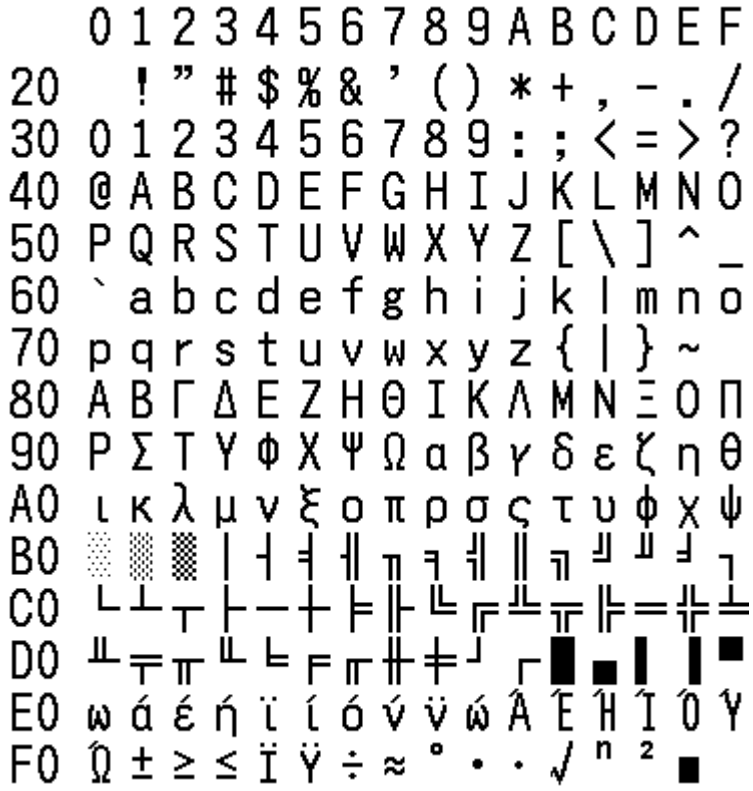


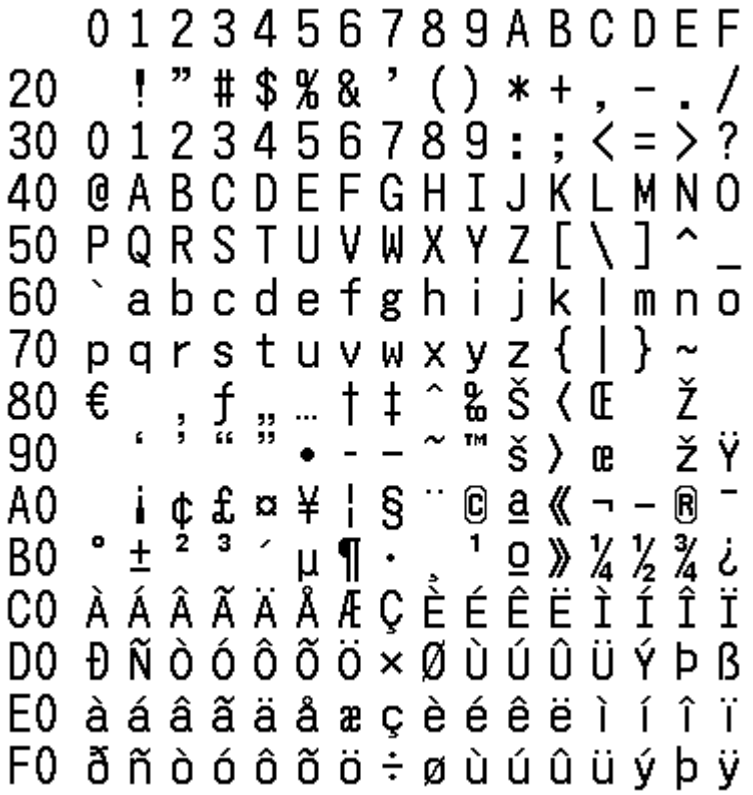
図 A-7 **CODE_PAGE_857** (Turkish)

RP-E10、RP-D10は**CODE_PAGE_737**をサポートしていません。



	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
20		!	”	#	$	%	&	’	(	)	*	+	,	-	.	/
30	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
60	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	
80	Α	Β	Γ	Δ	Ε	Ζ	Η	Θ	Ι	Κ	Λ	Μ	Ν	Ξ	Ο	Π
90	Ρ	Σ	Τ	Υ	Φ	Χ	Ψ	Ω	α	β	γ	δ	ε	ζ	η	θ
A0	ι	κ	λ	μ	ν	ξ	ο	π	ρ	σ	ς	τ	υ	φ	χ	ψ
B0	░	▒	▓	│	┤	╡	╢	╖	╕	╣	║	╗	╝	╜	╛	┐
C0	└	┴	┬	├	─	┼	╞	╟	╚	╔	╩	╦	╠	═	╬	╧
D0	╨	╤	╥	╙	╘	╒	╓	╫	╪	┘	┌	█	▄	▌	▐	▀
E0	ω	ά	έ	ή	ϊ	ί	ό	ύ	ϋ	ώ	Ά	Έ	Ή	Ί	Ό	Ύ
F0	Ώ	±	≥	≤	Ϊ	Ϋ	÷	≈	°	∙	·	√	^{n}	^{2}	■	

図 A-8 **CODE_PAGE_737** (Greek)



	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
20		!	”	#	$	%	&	’	(	)	*	+	,	-	.	/
30	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
60	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	
80	€		‚	ƒ	„	…	†	‡	ˆ	‰	Š	‹	Œ		Ž	
90		‘	’	“	”	•	-	–	˜	™	š	›	œ		ž	Ÿ
A0		¡	¢	£	¤	¥	¦	§	¨	©	ª	«	¬	-	®	¯
B0	°	±	²	³	´	µ	¶	·	¸	¹	º	»	¼	½	¾	¿
C0	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
D0	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
E0	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
F0	ð	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ

図 A-9 **CODE_PAGE_1252** (Latin)

RP-E10、RP-D10は**CODE_PAGE_866**をサポートしていません。

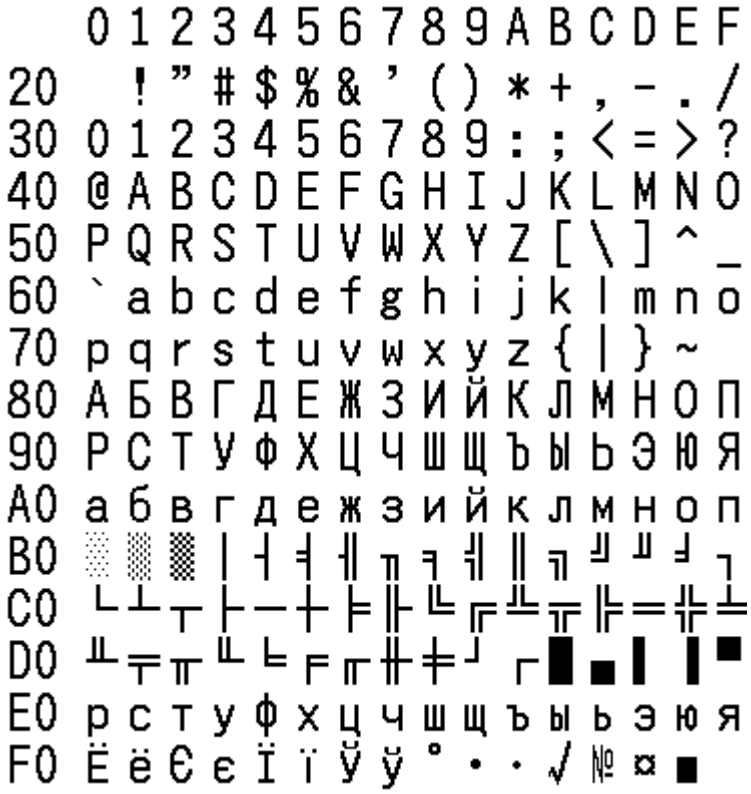


図 A-10 **CODE_PAGE_866** (Russian)

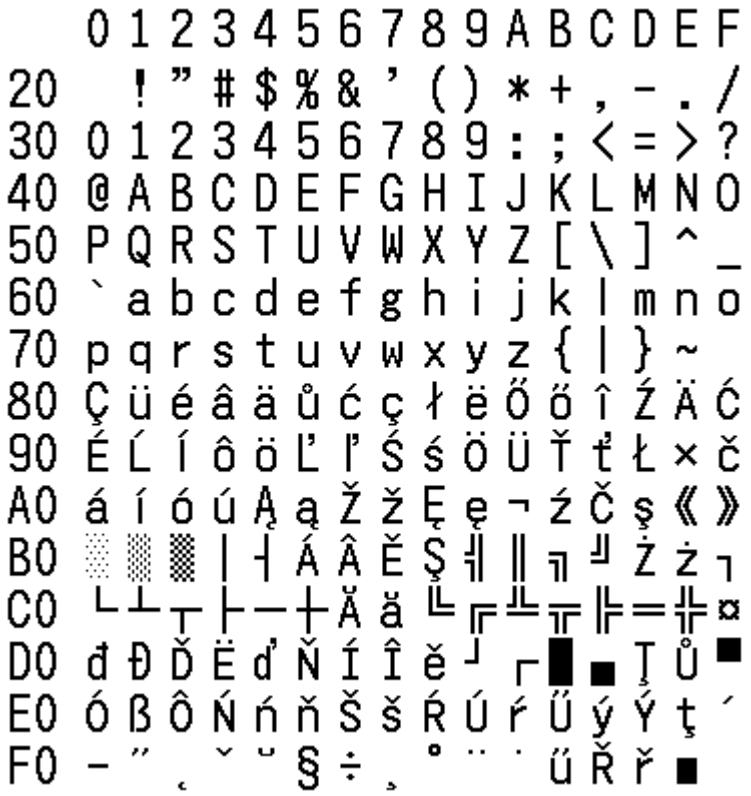


図 A-11 **CODE_PAGE_852** (Eastern Europe)

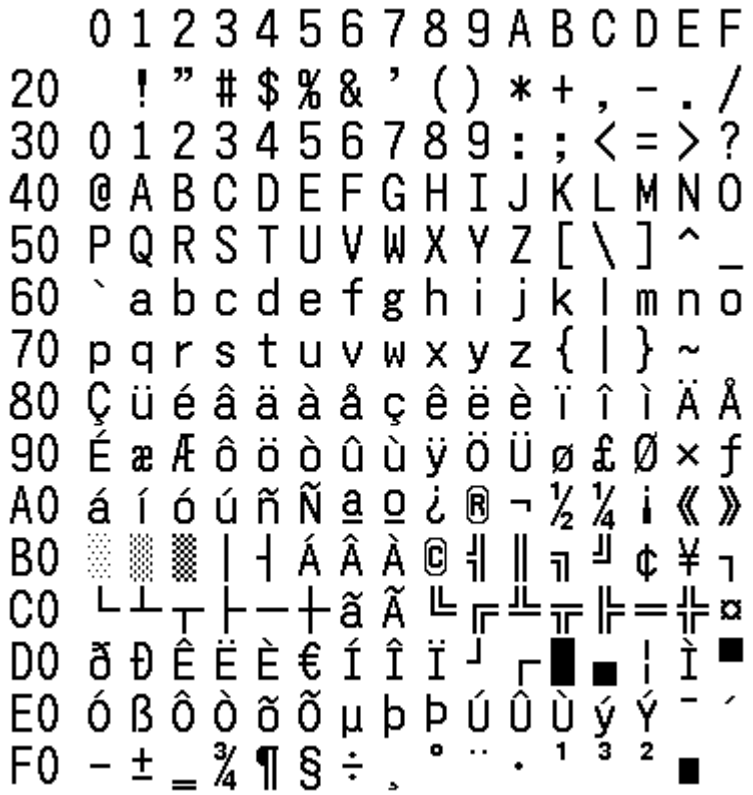


図 A-12 **CODE_PAGE_858** (Euro)

RP-E10、RP-D10は**CODE_PAGE_855**をサポートしていません。



図 A-13 **CODE_PAGE_855** (Cyrillic)



図 A-14 **CODE_PAGE_864** (Arabic)



図 A-15 **CODE_PAGE_1250** (Central European)



図 A-16 **CODE_PAGE_1251** (Cyrillic)

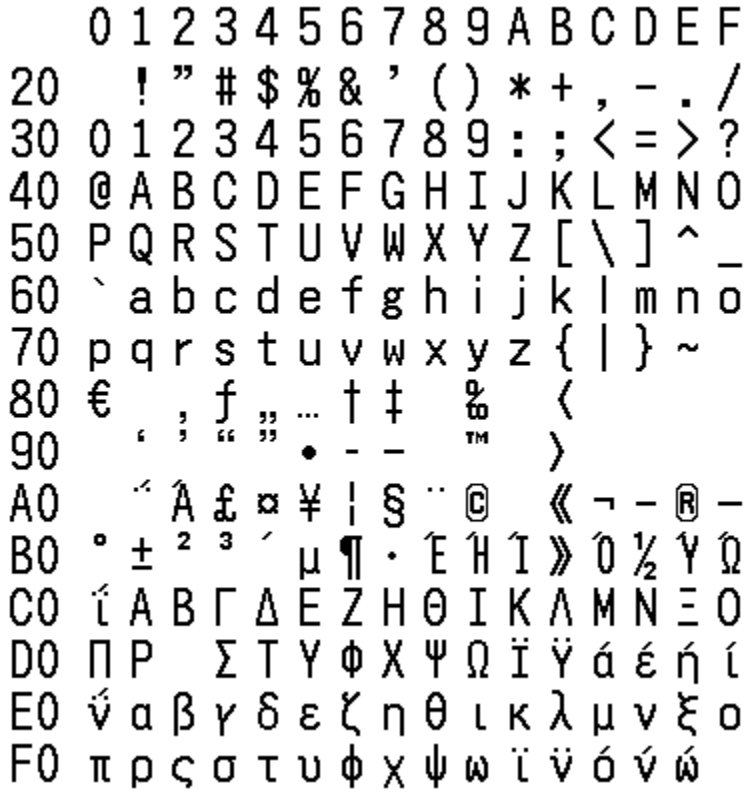


図 A-17 **CODE_PAGE_1253** (Greek)



	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
20		!	”	#	$	%	&	’	(	)	*	+	,	-	.	/
30	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
60	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	
80	€		‚	ƒ	„	…	†	‡	ˆ	‰	Š	‹	Œ			
90		‘	’	“	”	•	-	—	~	™	š	›	œ			Ÿ
A0		¡	¢	£	¤	¥	¦	§	¨	©	ª	«	¬	-	®	¯
B0	°	±	²	³	´	µ	¶	·	¸	¹	º	»	¼	½	¾	¿
C0	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
D0	Ğ	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	İ	Ş	ß
E0	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
F0	ğ	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ı	ş	ÿ

図 A-18 **CODE_PAGE_1254** (Turkish)

A.2 国際文字セット

設定する国際文字セットにより、特定の文字コードの印字、または表示結果が異なります。
特定の文字コード、及びその結果を下記に示します。

	23	24	40	5B	5C	5D	5E	60	7B	7C	7D	7E
COUNTRY_USA	#	$	@	[	\	]	^	`	{	|	}	~
COUNTRY_FRANCE	#	$	à	°	ç	§	^	`	é	ù	è	¨
COUNTRY_GERMANY	#	$	§	Ä	Ö	Ü	^	`	ä	ö	ü	ß
COUNTRY_ENGLAND	£	$	@	[	\	]	^	`	{	|	}	~
COUNTRY_DENMARK_1	#	$	@	Æ	Ø	Å	^	`	æ	ø	å	~
COUNTRY_SWEDEN	#	¤	É	Ä	Ö	Å	Ü	é	ä	ö	å	ü
COUNTRY_ITALY	#	$	@	°	\	é	^	ù	à	ò	è	ì
COUNTRY_SPAIN	₧	$	@	¡	Ñ	¿	^	`	¨	ñ	}	~
COUNTRY_JAPAN	#	$	@	[	¥	]	^	`	{	|	}	~
COUNTRY_NORWAY	#	¤	É	Æ	Ø	Å	Ü	é	æ	ø	å	ü
COUNTRY_DENMARK_2	#	$	É	Æ	Ø	Å	Ü	é	æ	ø	å	ü
COUNTRY_SPAIN_2	#	$	á	¡	Ñ	¿	é	`	í	ñ	ó	ú
COUNTRY_LATIN_AMERICA	#	$	á	¡	Ñ	¿	é	ü	í	ñ	ó	ú
COUNTRY_ARABIA	#	$	@	[	\	]	^	`	{	|	}	~

図 A-19 国際文字セット

付録 B　バーコードサイズ一覧

B.1 バーコードサイズ一覧(SLP720RT、SLP721RT、RP-F10、RP-G10、MP-B30、MP-B30L、MP-B21L)

B.1.1 PrintBarcode, PrintPageModeBarcode



(1) バーコードイメージの高さ

hriFont	hriPosition	バーコードの上端から 基準点までの長さ	バーコードイメージの高さ
FONT_A	**HRI_NONE**	moduleHeight	moduleHeight
	HRI_POSITION_ABOVE	moduleHeight+32	moduleHeight+32
	HRI_POSITION_BELOW	moduleHeight	moduleHeight+32
	HRI_POSITION_ABOVE_BELOW	moduleHeight+64	moduleHeight+64
FONT_B	**HRI_NONE**	moduleHeight	moduleHeight
	HRI_POSITION_ABOVE	moduleHeight+24	moduleHeight+24
	HRI_POSITION_BELOW	moduleHeight	moduleHeight+24
	HRI_POSITION_ABOVE_BELOW	moduleHeight+48	moduleHeight+48

(2) バーコードイメージの幅

barcodeSymbol	moduleSize	バーコードイメージの幅
BARCODE_UPC_A	**BARCODE_MODULE_WIDTH_2**	190
	BARCODE_MODULE_WIDTH_3	285
	BARCODE_MODULE_WIDTH_4	380
	BARCODE_MODULE_WIDTH_5	475
	BARCODE_MODULE_WIDTH_6	570
BARCODE_UPC_E	**BARCODE_MODULE_WIDTH_2**	102
	BARCODE_MODULE_WIDTH_3	153
	BARCODE_MODULE_WIDTH_4	204
	BARCODE_MODULE_WIDTH_5	255
	BARCODE_MODULE_WIDTH_6	306
BARCODE_EAN13	**BARCODE_MODULE_WIDTH_2**	190
	BARCODE_MODULE_WIDTH_3	285
	BARCODE_MODULE_WIDTH_4	380
	BARCODE_MODULE_WIDTH_5	475
	BARCODE_MODULE_WIDTH_6	570
BARCODE_JAN13	**BARCODE_MODULE_WIDTH_2**	190
	BARCODE_MODULE_WIDTH_3	285
	BARCODE_MODULE_WIDTH_4	380
	BARCODE_MODULE_WIDTH_5	475
	BARCODE_MODULE_WIDTH_6	570
BARCODE_EAN8	**BARCODE_MODULE_WIDTH_2**	134
	BARCODE_MODULE_WIDTH_3	201
	BARCODE_MODULE_WIDTH_4	268
	BARCODE_MODULE_WIDTH_5	335
	BARCODE_MODULE_WIDTH_6	402
BARCODE_JAN8	**BARCODE_MODULE_WIDTH_2**	134
	BARCODE_MODULE_WIDTH_3	201
	BARCODE_MODULE_WIDTH_4	268
	BARCODE_MODULE_WIDTH_5	335
	BARCODE_MODULE_WIDTH_6	402
BARCODE_CODE93	**BARCODE_MODULE_WIDTH_2**	18×バーコードデータ数＋56
	BARCODE_MODULE_WIDTH_3	27×バーコードデータ数＋84
	BARCODE_MODULE_WIDTH_4	36×バーコードデータ数＋112
	BARCODE_MODULE_WIDTH_5	45×バーコードデータ数＋140
	BARCODE_MODULE_WIDTH_6	54×バーコードデータ数＋168

barcodeSymbol	moduleSize	バーコードイメージの幅
BARCODE_CODE128	**BARCODE_MODULE_WIDTH_2**	22×バーコードデータ数＋26
	BARCODE_MODULE_WIDTH_3	33×バーコードデータ数＋39
	BARCODE_MODULE_WIDTH_4	44×バーコードデータ数＋52
	BARCODE_MODULE_WIDTH_5	55×バーコードデータ数＋65
	BARCODE_MODULE_WIDTH_6	66×バーコードデータ数＋78
BARCODE_GS1_OMNI_DIRECTIONAL	**BARCODE_MODULE_WIDTH_2**	192
	BARCODE_MODULE_WIDTH_3	288
	BARCODE_MODULE_WIDTH_4	384
	BARCODE_MODULE_WIDTH_5	480
	BARCODE_MODULE_WIDTH_6	576
BARCODE_GS1_TRUNCATED	**BARCODE_MODULE_WIDTH_2**	192
	BARCODE_MODULE_WIDTH_3	288
	BARCODE_MODULE_WIDTH_4	384
	BARCODE_MODULE_WIDTH_5	480
	BARCODE_MODULE_WIDTH_6	576
BARCODE_GS1_LIMITED	**BARCODE_MODULE_WIDTH_2**	158
	BARCODE_MODULE_WIDTH_3	237
	BARCODE_MODULE_WIDTH_4	316
	BARCODE_MODULE_WIDTH_5	395
	BARCODE_MODULE_WIDTH_6	474
BARCODE_GS1_EXPANDED[*1]	**BARCODE_MODULE_WIDTH_2**	バーコードモジュール数×2
	BARCODE_MODULE_WIDTH_3	バーコードモジュール数×3
	BARCODE_MODULE_WIDTH_4	バーコードモジュール数×4
	BARCODE_MODULE_WIDTH_5	バーコードモジュール数×5
	BARCODE_MODULE_WIDTH_6	バーコードモジュール数×6

*1:　バーコードモジュール数は指定するバーコードデータによって決まります。

barcodeSymbol	nwRatio	moduleSize	バーコードイメージの幅
BARCODE_CODE39	NWRATIO_1TO2	BARCODE_MODULE_WIDTH_2	26×バーコードデータ数＋50
		BARCODE_MODULE_WIDTH_3	39×バーコードデータ数＋75
		BARCODE_MODULE_WIDTH_4	52×バーコードデータ数＋100
		BARCODE_MODULE_WIDTH_5	65×バーコードデータ数＋125
		BARCODE_MODULE_WIDTH_6	78×バーコードデータ数＋150
	NWRATIO_1TO2_5	BARCODE_MODULE_WIDTH_2	29×バーコードデータ数＋56
		BARCODE_MODULE_WIDTH_3	45×バーコードデータ数＋87
		BARCODE_MODULE_WIDTH_4	58×バーコードデータ数＋112
		BARCODE_MODULE_WIDTH_5	74×バーコードデータ数＋143
		BARCODE_MODULE_WIDTH_6	87×バーコードデータ数＋168
	NWRATIO_1TO3	BARCODE_MODULE_WIDTH_2	32×バーコードデータ数＋62
		BARCODE_MODULE_WIDTH_3	48×バーコードデータ数＋93
		BARCODE_MODULE_WIDTH_4	64×バーコードデータ数＋124
		BARCODE_MODULE_WIDTH_5	80×バーコードデータ数＋155
		BARCODE_MODULE_WIDTH_6	96×バーコードデータ数＋186
BARCODE_ITF	NWRATIO_1TO2	BARCODE_MODULE_WIDTH_2	14×バーコードデータ数＋16
		BARCODE_MODULE_WIDTH_3	21×バーコードデータ数＋24
		BARCODE_MODULE_WIDTH_4	28×バーコードデータ数＋32
		BARCODE_MODULE_WIDTH_5	35×バーコードデータ数＋40
		BARCODE_MODULE_WIDTH_6	42×バーコードデータ数＋48
	NWRATIO_1TO2_5	BARCODE_MODULE_WIDTH_2	16×バーコードデータ数＋17
		BARCODE_MODULE_WIDTH_3	25×バーコードデータ数＋26
		BARCODE_MODULE_WIDTH_4	32×バーコードデータ数＋34
		BARCODE_MODULE_WIDTH_5	41×バーコードデータ数＋43
		BARCODE_MODULE_WIDTH_6	48×バーコードデータ数＋51
	NWRATIO_1TO3	BARCODE_MODULE_WIDTH_2	18×バーコードデータ数＋18
		BARCODE_MODULE_WIDTH_3	27×バーコードデータ数＋27
		BARCODE_MODULE_WIDTH_4	36×バーコードデータ数＋36
		BARCODE_MODULE_WIDTH_5	45×バーコードデータ数＋45
		BARCODE_MODULE_WIDTH_6	54×バーコードデータ数＋54
BARCODE_CODABAR*1	NWRATIO_1TO2	BARCODE_MODULE_WIDTH_2	20×データ数＋2×(2＋ワイドデータ数)－2
		BARCODE_MODULE_WIDTH_3	30×データ数＋3×(2＋ワイドデータ数)－3
		BARCODE_MODULE_WIDTH_4	40×データ数＋4×(2＋ワイドデータ数)－4
		BARCODE_MODULE_WIDTH_5	50×データ数＋5×(2＋ワイドデータ数)－5
		BARCODE_MODULE_WIDTH_6	60×データ数＋6×(2＋ワイドデータ数)－6

barcodeSymbol	nwRatio	moduleSize	バーコードイメージの幅
BARCODE_CODABAR*1	**NWRATIO_1TO2_5**	**BARCODE_MODULE_WIDTH_2**	22×データ数＋3×(2＋ワイドデータ数)－2
		BARCODE_MODULE_WIDTH_3	34×データ数＋5×(2＋ワイドデータ数)－3
		BARCODE_MODULE_WIDTH_4	44×データ数＋6×(2＋ワイドデータ数)－4
		BARCODE_MODULE_WIDTH_5	56×データ数＋8×(2＋ワイドデータ数)－5
		BARCODE_MODULE_WIDTH_6	66×データ数＋9×(2＋ワイドデータ数)－6
	NWRATIO_1TO3	**BARCODE_MODULE_WIDTH_2**	24×データ数＋4×(2＋ワイドデータ数)－2
		BARCODE_MODULE_WIDTH_3	36×データ数＋6×(2＋ワイドデータ数)－3
		BARCODE_MODULE_WIDTH_4	48×データ数＋8×(2＋ワイドデータ数)－4
		BARCODE_MODULE_WIDTH_5	60×データ数＋10×(2＋ワイドデータ数)－5
		BARCODE_MODULE_WIDTH_6	72×データ数＋12×(2＋ワイドデータ数)－6

*1: データ数はスタートキャラクタ、ストップキャラクタを除いた全てのキャラクタ数です。
ワイドデータ数は「：／．＋」の数です。

barcodeSymbol	データ数	moduleSize	バーコード イメージの幅
BARCODE_EAN13_ADDON	14 または 15	**BARCODE_MODULE_WIDTH_2**	244
		BARCODE_MODULE_WIDTH_3	366
		BARCODE_MODULE_WIDTH_4	488
		BARCODE_MODULE_WIDTH_5	610
		BARCODE_MODULE_WIDTH_6	732
	17 または 18	**BARCODE_MODULE_WIDTH_2**	298
		BARCODE_MODULE_WIDTH_3	447
		BARCODE_MODULE_WIDTH_4	596
		BARCODE_MODULE_WIDTH_5	745
		BARCODE_MODULE_WIDTH_6	894
BARCODE_JAN13_ADDON	14 または 15	**BARCODE_MODULE_WIDTH_2**	244
		BARCODE_MODULE_WIDTH_3	366
		BARCODE_MODULE_WIDTH_4	488
		BARCODE_MODULE_WIDTH_5	610
		BARCODE_MODULE_WIDTH_6	732
	17 または 18	**BARCODE_MODULE_WIDTH_2**	298
		BARCODE_MODULE_WIDTH_3	447
		BARCODE_MODULE_WIDTH_4	596
		BARCODE_MODULE_WIDTH_5	745
		BARCODE_MODULE_WIDTH_6	894

B.1.2 PrintPDF417, PrintPageModePDF417



(1) バーコードイメージの高さ

バーコードイメージの高さ[*1] ＝ moduleHeight × row[*2]

*1: バーコードイメージの高さ ＝ バーコードの上端から基準点までの長さ
*2: row≠0

(2) バーコードイメージの幅

pdf417Symbolが**PDF417_STANDARD**の場合

バーコードイメージの幅 ＝（17 × column[*1] ＋ 69）× モジュールサイズ値

*1: column≠0

pdf417Symbolが**PDF417_COMPACT**の場合

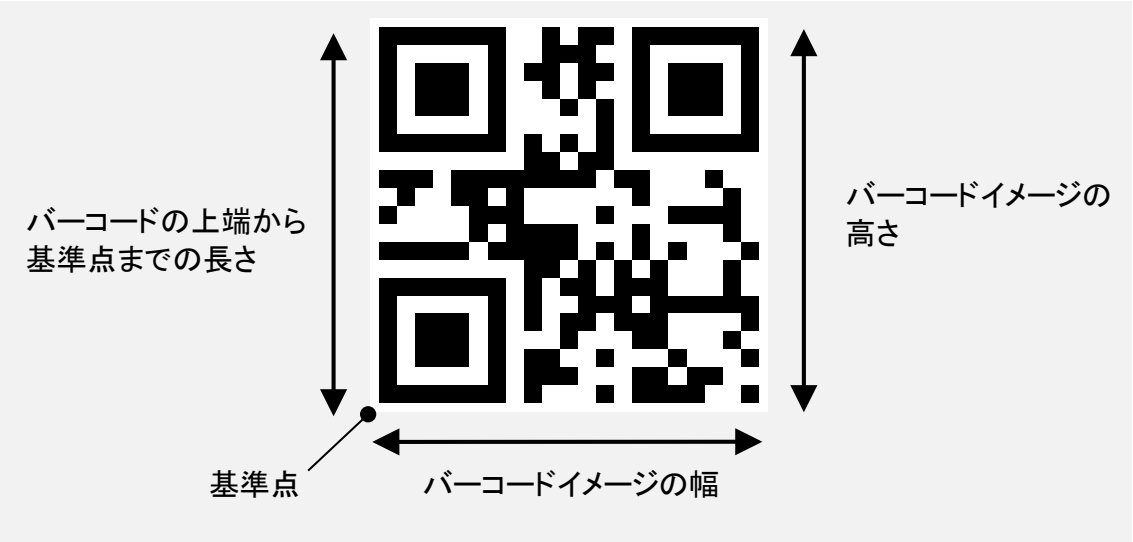
バーコードイメージの幅 ＝（17 × column[*1] ＋ 35）× モジュールサイズ値

*1: column≠0

モジュールサイズ値

moduleSize	モジュールサイズ値
PDF417_MODULE_WIDTH_2	2
PDF417_MODULE_WIDTH_3	3
PDF417_MODULE_WIDTH_4	4
PDF417_MODULE_WIDTH_5	5
PDF417_MODULE_WIDTH_6	6
PDF417_MODULE_WIDTH_7	7
PDF417_MODULE_WIDTH_8	8

B.1.3 PrintQRCode, PrintPageModeQRCode



(1) バーコードイメージの高さ及び幅

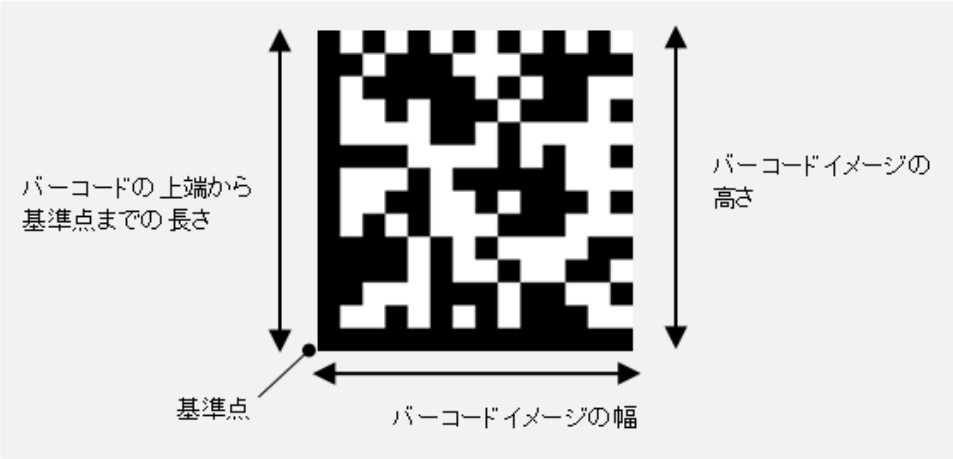
バーコードイメージの高さ*1及び幅 ＝（4 × バージョン*2 ＋ 17）× モジュールサイズ値

*1: バーコードイメージの高さ ＝ バーコードの上端から基準点までの長さ
*2: バージョンはバーコードデータの内容及びエラー訂正レベルにより決まります。

モジュールサイズ値

moduleSize	モジュールサイズ値
QR_MODULE_SIZE_2	2
QR_MODULE_SIZE_3	3
QR_MODULE_SIZE_4	4
QR_MODULE_SIZE_5	5
QR_MODULE_SIZE_6	6
QR_MODULE_SIZE_7	7
QR_MODULE_SIZE_8	8
QR_MODULE_SIZE_9	9
QR_MODULE_SIZE_10	10
QR_MODULE_SIZE_11	11
QR_MODULE_SIZE_12	12
QR_MODULE_SIZE_13	13
QR_MODULE_SIZE_14	14
QR_MODULE_SIZE_15	15
QR_MODULE_SIZE_16	16

B.1.4 PrintDataMatrix, PrintPageModeDataMatrix



(1) バーコードイメージの高さ及び幅

バーコードイメージの高さ ＝ 縦方向モジュール数 × モジュールサイズ値

バーコードイメージの幅 ＝ 横方向モジュール数 × モジュールサイズ値

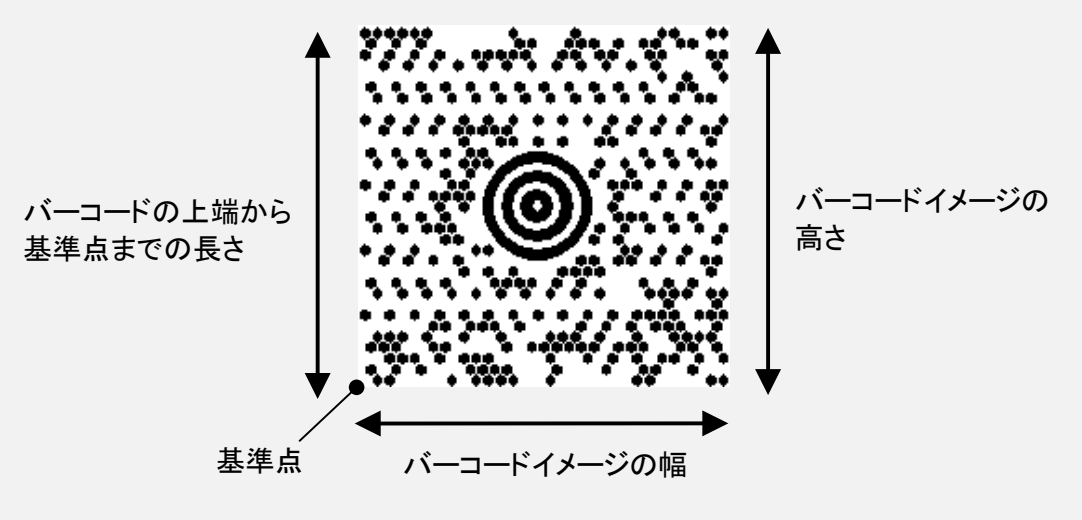
dataMatrixModule	縦方向モジュール数	横方向モジュール数
DATA_MATRIX_10_10	10	10
DATA_MATRIX_12_12	12	12
DATA_MATRIX_14_14	14	14
DATA_MATRIX_16_16	16	16
DATA_MATRIX_18_18	18	18
DATA_MATRIX_20_20	20	20
DATA_MATRIX_22_22	22	22
DATA_MATRIX_24_24	23	23
DATA_MATRIX_26_26	26	26
DATA_MATRIX_32_32	32	32
DATA_MATRIX_36_36	36	36
DATA_MATRIX_40_40	40	40
DATA_MATRIX_44_44	44	44
DATA_MATRIX_48_48	48	48
DATA_MATRIX_52_52	52	52
DATA_MATRIX_64_64	64	64
DATA_MATRIX_72_72	72	72
DATA_MATRIX_80_80	80	80
DATA_MATRIX_88_88	88	88
DATA_MATRIX_96_96	96	96
DATA_MATRIX_104_104	104	104

dataMatrixModule	縦方向モジュール数	横方向モジュール数
DATA_MATRIX_120_120	120	120
DATA_MATRIX_132_132	132	132
DATA_MATRIX_144_144	144	144
DATA_MATRIX_8_18	8	18
DATA_MATRIX_8_32	8	32
DATA_MATRIX_12_26	12	26
DATA_MATRIX_12_36	12	36
DATA_MATRIX_16_36	16	36
DATA_MATRIX_16_48	16	48

モジュールサイズ値

moduleSize	モジュールサイズ値
DATAMATRIX_MODULE_SIZE_2	2
DATAMATRIX_MODULE_SIZE_3	3
DATAMATRIX_MODULE_SIZE_4	4
DATAMATRIX_MODULE_SIZE_5	5
DATAMATRIX_MODULE_SIZE_6	6
DATAMATRIX_MODULE_SIZE_7	7
DATAMATRIX_MODULE_SIZE_8	8
DATAMATRIX_MODULE_SIZE_9	9
DATAMATRIX_MODULE_SIZE_10	10
DATAMATRIX_MODULE_SIZE_11	11
DATAMATRIX_MODULE_SIZE_12	12
DATAMATRIX_MODULE_SIZE_13	13
DATAMATRIX_MODULE_SIZE_14	14
DATAMATRIX_MODULE_SIZE_15	15
DATAMATRIX_MODULE_SIZE_16	16

B.1.5 PrintMaxicode, PrintPageModeMaxicode



(1) バーコードイメージの高さ

バーコードイメージの高さ*1 ＝ 200

*1: バーコードイメージの高さ ＝ バーコードの上端から基準点までの長さ

(2) バーコードイメージの幅

バーコードイメージの幅 ＝ 210

B.1.6 PrintGS1DataBarStacked, PrintPageModeGS1DataBarStacked



(1) バーコードイメージの高さ及び幅

バーコードイメージの高さ*1 ＝ 13 × モジュールサイズ値

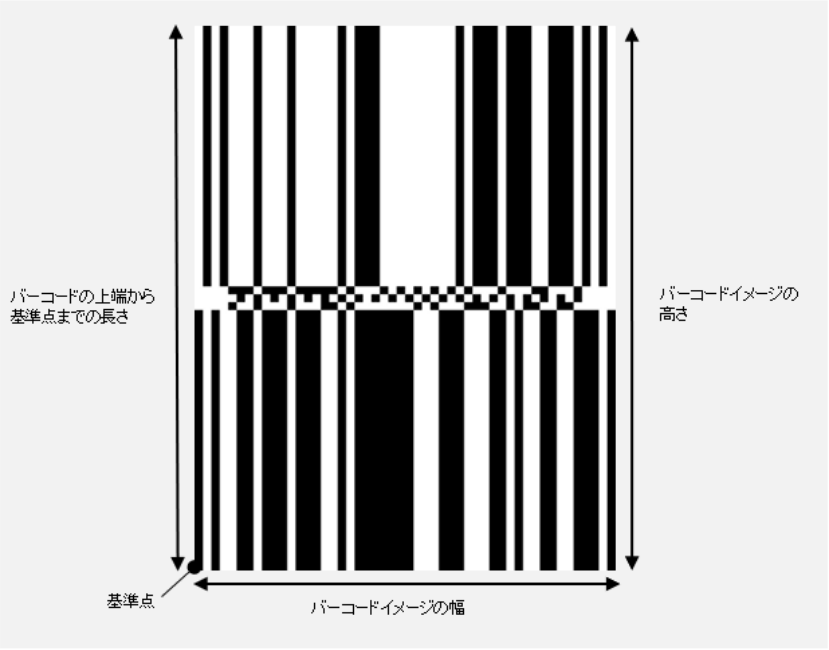
*1: バーコードイメージの高さ ＝ バーコードの上端から基準点までの長さ

バーコードイメージの幅 ＝ 50 × モジュールサイズ値

モジュールサイズ値

moduleSize	モジュールサイズ値
GS1DATABAR_MODULE_SIZE_2	2
GS1DATABAR_MODULE_SIZE_3	3
GS1DATABAR_MODULE_SIZE_4	4
GS1DATABAR_MODULE_SIZE_5	5
GS1DATABAR_MODULE_SIZE_6	6
GS1DATABAR_MODULE_SIZE_7	7
GS1DATABAR_MODULE_SIZE_8	8
GS1DATABAR_MODULE_SIZE_9	9
GS1DATABAR_MODULE_SIZE_10	10
GS1DATABAR_MODULE_SIZE_11	11
GS1DATABAR_MODULE_SIZE_12	12
GS1DATABAR_MODULE_SIZE_13	13
GS1DATABAR_MODULE_SIZE_14	14
GS1DATABAR_MODULE_SIZE_15	15
GS1DATABAR_MODULE_SIZE_16	16

B.1.7 PrintGS1DataBarStackedOmnidirectional, PrintPageModeGS1DataBarStackedOmnidirectional



(1) バーコードイメージの高さ及び幅

バーコードイメージの高さ[*1] ＝（moduleHeight × 2 ＋ 3）× モジュールサイズ値

*1: バーコードイメージの高さ ＝ バーコードの上端から基準点までの長さ

バーコードイメージの幅 ＝ 50 × モジュールサイズ値

モジュールサイズ値

moduleSize	モジュールサイズ値
GS1DATABAR_MODULE_SIZE_2	2
GS1DATABAR_MODULE_SIZE_3	3
GS1DATABAR_MODULE_SIZE_4	4
GS1DATABAR_MODULE_SIZE_5	5
GS1DATABAR_MODULE_SIZE_6	6
GS1DATABAR_MODULE_SIZE_7	7
GS1DATABAR_MODULE_SIZE_8	8
GS1DATABAR_MODULE_SIZE_9	9
GS1DATABAR_MODULE_SIZE_10	10
GS1DATABAR_MODULE_SIZE_11	11
GS1DATABAR_MODULE_SIZE_12	12
GS1DATABAR_MODULE_SIZE_13	13
GS1DATABAR_MODULE_SIZE_14	14
GS1DATABAR_MODULE_SIZE_15	15
GS1DATABAR_MODULE_SIZE_16	16

B.1.8 PrintGS1DataBarExpandedStacked, PrintPageModeGS1DataBarExpandedStacked



(1) バーコードイメージの高さ及び幅

バーコードイメージの高さ*1 ＝((34 ＋ 3) × 段数*2 ＋ 34) × モジュールサイズ値

*1: バーコードイメージの高さ＝バーコードの上端から基準点までの長さ
*2: 段数はバーコードデータによって決まります。

バーコードイメージの幅 ＝(4 ＋ 49 × column / 2) × モジュールサイズ値

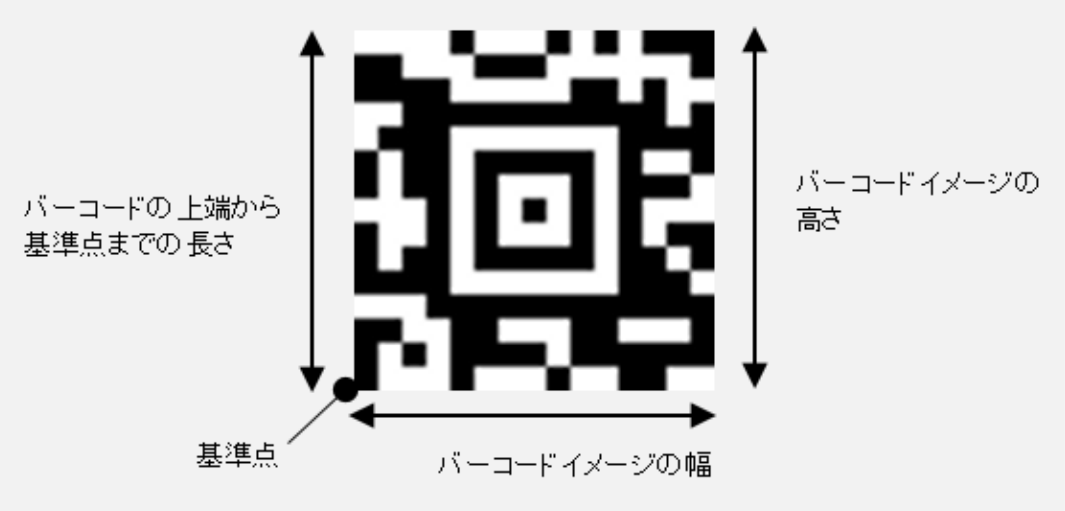
モジュールサイズ値

moduleSize	モジュールサイズ値
GS1DATABAR_MODULE_SIZE_2	2
GS1DATABAR_MODULE_SIZE_3	3
GS1DATABAR_MODULE_SIZE_4	4
GS1DATABAR_MODULE_SIZE_5	5
GS1DATABAR_MODULE_SIZE_6	6
GS1DATABAR_MODULE_SIZE_7	7
GS1DATABAR_MODULE_SIZE_8	8
GS1DATABAR_MODULE_SIZE_9	9
GS1DATABAR_MODULE_SIZE_10	10
GS1DATABAR_MODULE_SIZE_11	11
GS1DATABAR_MODULE_SIZE_12	12
GS1DATABAR_MODULE_SIZE_13	13
GS1DATABAR_MODULE_SIZE_14	14
GS1DATABAR_MODULE_SIZE_15	15
GS1DATABAR_MODULE_SIZE_16	16

B.2 バーコードサイズ一覧(MP-B30L、MP-B21L)

MP-B30L及びMP-B21Lは「B.1 バーコードサイズ一覧(SLP720RT、SLP721RT、RP-F10、RP-G10、MP-B30、MP-B30L、MP-B21L)」のバーコードに加えて、Aztec Codeが利用できます。

B.2.1 PrintAztecCode, PrintPageModeAztecCode



(1) バーコードイメージの高さ及び幅

バーコードイメージの高さ*1及び幅 ＝ モジュール数 × モジュールサイズ値

*1: バーコードイメージの高さ ＝ バーコードの上端から基準点までの長さ

例: aztecSymbolが**AZTECCODE_COMPACT**、layerが1、moduleSizeが**AZTECCODE_MODULE_SIZE_6**の場合

バーコードイメージの高さ及び幅 ＝ 15 × 6 ＝ 90

モジュール数

aztecSymbol	layer	モジュール数
AZTECCODE_FULLRANGE	4	31
	5	37
	6	41
	7	45
	8	49
	9	53
	10	57
	11	61
	12	67
	13	71
	14	75
	15	79
	16	83

aztecSymbol	layer	モジュール数
AZTECCODE_FULLRANGE	17	87
	18	91
	19	95
	20	101
	21	105
	22	109
	23	113
	24	117
	25	121
	26	125
	27	131
	28	135
	29	139
	30	143
	31	147
	32	151
AZTECCODE_COMPACT	1	15
	2	19
	3	23
	4	27

モジュールサイズ値

moduleSize	モジュールサイズ値
AZTECCODE_MODULE_SIZE_2	2
AZTECCODE_MODULE_SIZE_3	3
AZTECCODE_MODULE_SIZE_4	4
AZTECCODE_MODULE_SIZE_5	5
AZTECCODE_MODULE_SIZE_6	6
AZTECCODE_MODULE_SIZE_7	7
AZTECCODE_MODULE_SIZE_8	8
AZTECCODE_MODULE_SIZE_9	9
AZTECCODE_MODULE_SIZE_10	10
AZTECCODE_MODULE_SIZE_11	11
AZTECCODE_MODULE_SIZE_12	12
AZTECCODE_MODULE_SIZE_13	13
AZTECCODE_MODULE_SIZE_14	14
AZTECCODE_MODULE_SIZE_15	15
AZTECCODE_MODULE_SIZE_16	16

B.3 バーコードサイズ一覧(RP-E10、RP-D10、MP-B20、MP-A40)

B.3.1 PrintBarcode, PrintPageModeBarcode



(1) バーコードイメージの高さ

hriFont	hriPosition	バーコードの上端から 基準点までの長さ	バーコードイメージの高さ
FONT_A	**HRI_NONE**	moduleHeight	moduleHeight
	HRI_POSITION_ABOVE	moduleHeight+32	moduleHeight+32
	HRI_POSITION_BELOW	moduleHeight	moduleHeight+32
	HRI_POSITION_ABOVE_BELOW	moduleHeight+64	moduleHeight+64
FONT_B	**HRI_NONE**	moduleHeight	moduleHeight
	HRI_POSITION_ABOVE	moduleHeight+24	moduleHeight+24
	HRI_POSITION_BELOW	moduleHeight	moduleHeight+24
	HRI_POSITION_ABOVE_BELOW	moduleHeight+48	moduleHeight+48

(2) バーコードイメージの幅

barcodeSymbol	moduleSize	バーコードイメージの幅
BARCODE_UPC_A	**BARCODE_MODULE_WIDTH_2**	226
	BARCODE_MODULE_WIDTH_3	339
	BARCODE_MODULE_WIDTH_4	452
	BARCODE_MODULE_WIDTH_5	565
	BARCODE_MODULE_WIDTH_6	678
BARCODE_UPC_E	**BARCODE_MODULE_WIDTH_2**	130
	BARCODE_MODULE_WIDTH_3	195
	BARCODE_MODULE_WIDTH_4	260
	BARCODE_MODULE_WIDTH_5	325
	BARCODE_MODULE_WIDTH_6	390
BARCODE_EAN13	**BARCODE_MODULE_WIDTH_2**	226
	BARCODE_MODULE_WIDTH_3	339
	BARCODE_MODULE_WIDTH_4	452
	BARCODE_MODULE_WIDTH_5	565
	BARCODE_MODULE_WIDTH_6	678
BARCODE_JAN13	**BARCODE_MODULE_WIDTH_2**	226
	BARCODE_MODULE_WIDTH_3	339
	BARCODE_MODULE_WIDTH_4	452
	BARCODE_MODULE_WIDTH_5	565
	BARCODE_MODULE_WIDTH_6	678
BARCODE_EAN8	**BARCODE_MODULE_WIDTH_2**	162
	BARCODE_MODULE_WIDTH_3	243
	BARCODE_MODULE_WIDTH_4	324
	BARCODE_MODULE_WIDTH_5	405
	BARCODE_MODULE_WIDTH_6	486
BARCODE_JAN8	**BARCODE_MODULE_WIDTH_2**	162
	BARCODE_MODULE_WIDTH_3	243
	BARCODE_MODULE_WIDTH_4	324
	BARCODE_MODULE_WIDTH_5	405
	BARCODE_MODULE_WIDTH_6	486
BARCODE_CODE93	**BARCODE_MODULE_WIDTH_2**	18×バーコードデータ数＋96
	BARCODE_MODULE_WIDTH_3	27×バーコードデータ数＋144
	BARCODE_MODULE_WIDTH_4	36×バーコードデータ数＋192
	BARCODE_MODULE_WIDTH_5	45×バーコードデータ数＋240
	BARCODE_MODULE_WIDTH_6	54×バーコードデータ数＋288

barcodeSymbol	moduleSize	バーコードイメージの幅
BARCODE_CODE128	**BARCODE_MODULE_WIDTH_2**	22×バーコードデータ数＋66
	BARCODE_MODULE_WIDTH_3	33×バーコードデータ数＋99
	BARCODE_MODULE_WIDTH_4	44×バーコードデータ数＋132
	BARCODE_MODULE_WIDTH_5	55×バーコードデータ数＋165
	BARCODE_MODULE_WIDTH_6	66×バーコードデータ数＋198
BARCODE_GS1_OMNI_DIRECTIONAL[*1]	**BARCODE_MODULE_WIDTH_2**	192
	BARCODE_MODULE_WIDTH_3	288
	BARCODE_MODULE_WIDTH_4	384
	BARCODE_MODULE_WIDTH_5	480
	BARCODE_MODULE_WIDTH_6	576
BARCODE_GS1_TRUNCATED[*1]	**BARCODE_MODULE_WIDTH_2**	192
	BARCODE_MODULE_WIDTH_3	288
	BARCODE_MODULE_WIDTH_4	384
	BARCODE_MODULE_WIDTH_5	480
	BARCODE_MODULE_WIDTH_6	576
BARCODE_GS1_LIMITED[*1]	**BARCODE_MODULE_WIDTH_2**	158
	BARCODE_MODULE_WIDTH_3	237
	BARCODE_MODULE_WIDTH_4	316
	BARCODE_MODULE_WIDTH_5	395
	BARCODE_MODULE_WIDTH_6	474
BARCODE_GS1_EXPANDED[*1*2]	**BARCODE_MODULE_WIDTH_2**	バーコードモジュール数×2
	BARCODE_MODULE_WIDTH_3	バーコードモジュール数×3
	BARCODE_MODULE_WIDTH_4	バーコードモジュール数×4
	BARCODE_MODULE_WIDTH_5	バーコードモジュール数×5
	BARCODE_MODULE_WIDTH_6	バーコードモジュール数×6

*1: MP-B20 及び MP-A40 のみサポートします。
*2: バーコードモジュール数は指定するバーコードデータによって決まります。

barcodeSymbol	nwRatio	moduleSize	バーコードイメージの幅
BARCODE_CODE39	**NWRATIO_1TO2**	**BARCODE_MODULE_WIDTH_2**	26×バーコードデータ数＋90
		BARCODE_MODULE_WIDTH_3	39×バーコードデータ数＋135
		BARCODE_MODULE_WIDTH_4	52×バーコードデータ数＋180
		BARCODE_MODULE_WIDTH_5	65×バーコードデータ数＋225
		BARCODE_MODULE_WIDTH_6	78×バーコードデータ数＋270
	NWRATIO_1TO2_5	**BARCODE_MODULE_WIDTH_2**	29×バーコードデータ数＋96
		BARCODE_MODULE_WIDTH_3	45×バーコードデータ数＋147
		BARCODE_MODULE_WIDTH_4	58×バーコードデータ数＋192
		BARCODE_MODULE_WIDTH_5	74×バーコードデータ数＋243
		BARCODE_MODULE_WIDTH_6	87×バーコードデータ数＋288

barcodeSymbol	nwRatio	moduleSize	バーコードイメージの幅
BARCODE_CODE39	**NWRATIO_1TO3**	**BARCODE_MODULE_WIDTH_2**	32×バーコードデータ数＋102
		BARCODE_MODULE_WIDTH_3	48×バーコードデータ数＋153
		BARCODE_MODULE_WIDTH_4	64×バーコードデータ数＋204
		BARCODE_MODULE_WIDTH_5	80×バーコードデータ数＋255
		BARCODE_MODULE_WIDTH_6	96×バーコードデータ数＋306
BARCODE_ITF	**NWRATIO_1TO2**	**BARCODE_MODULE_WIDTH_2**	14×バーコードデータ数＋56
		BARCODE_MODULE_WIDTH_3	21×バーコードデータ数＋84
		BARCODE_MODULE_WIDTH_4	28×バーコードデータ数＋112
		BARCODE_MODULE_WIDTH_5	35×バーコードデータ数＋140
		BARCODE_MODULE_WIDTH_6	42×バーコードデータ数＋168
	NWRATIO_1TO2_5	**BARCODE_MODULE_WIDTH_2**	16×バーコードデータ数＋57
		BARCODE_MODULE_WIDTH_3	25×バーコードデータ数＋86
		BARCODE_MODULE_WIDTH_4	32×バーコードデータ数＋114
		BARCODE_MODULE_WIDTH_5	41×バーコードデータ数＋143
		BARCODE_MODULE_WIDTH_6	48×バーコードデータ数＋171
	NWRATIO_1TO3	**BARCODE_MODULE_WIDTH_2**	18×バーコードデータ数＋58
		BARCODE_MODULE_WIDTH_3	27×バーコードデータ数＋87
		BARCODE_MODULE_WIDTH_4	36×バーコードデータ数＋116
		BARCODE_MODULE_WIDTH_5	45×バーコードデータ数＋145
		BARCODE_MODULE_WIDTH_6	54×バーコードデータ数＋174

barcodeSymbol	nwRatio	moduleSize	バーコードイメージの幅
BARCODE_CODABAR[*1]	**NWRATIO_1TO2**	**BARCODE_MODULE_WIDTH_2**	20×データ数＋2×(2＋ワイドデータ数)＋38
		BARCODE_MODULE_WIDTH_3	30×データ数＋3×(2＋ワイドデータ数)＋57
		BARCODE_MODULE_WIDTH_4	40×データ数＋4×(2＋ワイドデータ数)＋76
		BARCODE_MODULE_WIDTH_5	50×データ数＋5×(2＋ワイドデータ数)＋95
		BARCODE_MODULE_WIDTH_6	60×データ数＋6×(2＋ワイドデータ数)＋114
	NWRATIO_1TO2_5	**BARCODE_MODULE_WIDTH_2**	22×データ数＋3×(2＋ワイドデータ数)＋38
		BARCODE_MODULE_WIDTH_3	34×データ数＋5×(2＋ワイドデータ数)＋57
		BARCODE_MODULE_WIDTH_4	44×データ数＋6×(2＋ワイドデータ数)＋76
		BARCODE_MODULE_WIDTH_5	56×データ数＋8×(2＋ワイドデータ数)＋95
		BARCODE_MODULE_WIDTH_6	66×データ数＋9×(2＋ワイドデータ数)＋114

barcodeSymbol	nwRatio	moduleSize	バーコードイメージの幅
BARCODE_CODABAR*1	**NWRATIO_1TO3**	**BARCODE_MODULE_WIDTH_2**	24×データ数＋4×(2＋ワイドデータ数)＋38
		BARCODE_MODULE_WIDTH_3	36×データ数＋6×(2＋ワイドデータ数)＋57
		BARCODE_MODULE_WIDTH_4	48×データ数＋8×(2＋ワイドデータ数)＋76
		BARCODE_MODULE_WIDTH_5	60×データ数＋10×(2＋ワイドデータ数)＋95
		BARCODE_MODULE_WIDTH_6	72×データ数＋12×(2＋ワイドデータ数)＋114

*1: データ数はスタートキャラクタ、ストップキャラクタを除いた全てのキャラクタ数です。
ワイドデータ数は「：／．＋」の数です。

barcodeSymbol	データ数	moduleSize	バーコードイメージの幅
BARCODE_EAN13_ADDON	14 または 15	**BARCODE_MODULE_WIDTH_2**	276
		BARCODE_MODULE_WIDTH_3	414
		BARCODE_MODULE_WIDTH_4	552
		BARCODE_MODULE_WIDTH_5	690
		BARCODE_MODULE_WIDTH_6	828
	17 または 18	**BARCODE_MODULE_WIDTH_2**	330
		BARCODE_MODULE_WIDTH_3	495
		BARCODE_MODULE_WIDTH_4	660
		BARCODE_MODULE_WIDTH_5	825
		BARCODE_MODULE_WIDTH_6	990
BARCODE_JAN13_ADDON	14 または 15	**BARCODE_MODULE_WIDTH_2**	276
		BARCODE_MODULE_WIDTH_3	414
		BARCODE_MODULE_WIDTH_4	552
		BARCODE_MODULE_WIDTH_5	690
		BARCODE_MODULE_WIDTH_6	828
	17 または 18	**BARCODE_MODULE_WIDTH_2**	330
		BARCODE_MODULE_WIDTH_3	495
		BARCODE_MODULE_WIDTH_4	660
		BARCODE_MODULE_WIDTH_5	825
		BARCODE_MODULE_WIDTH_6	990

B.3.2 PrintPDF417, PrintPageModePDF417



(1) バーコードイメージの高さ

moduleSize	バーコードイメージの高さ[*1]
PDF417_MODULE_WIDTH_2	moduleHeight×row[*2]+8
PDF417_MODULE_WIDTH_3	moduleHeight×row[*2]+12
PDF417_MODULE_WIDTH_4	moduleHeight×row[*2]+16
PDF417_MODULE_WIDTH_5[*3]	moduleHeight×row[*2]+20
PDF417_MODULE_WIDTH_6[*3]	moduleHeight×row[*2]+24
PDF417_MODULE_WIDTH_7[*3]	moduleHeight×row[*2]+28
PDF417_MODULE_WIDTH_8[*3]	moduleHeight×row[*2]+32

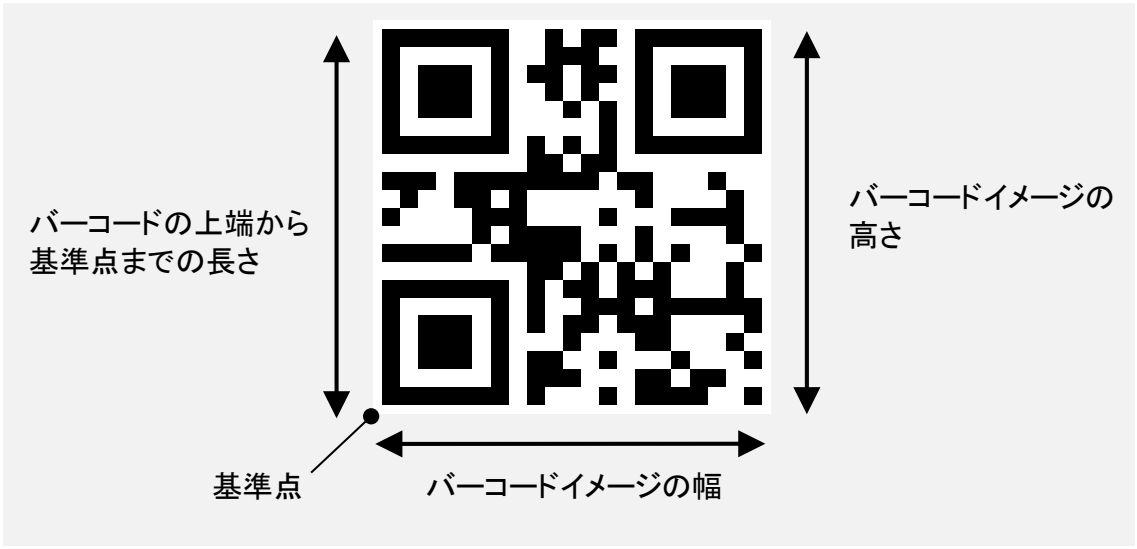
*1: バーコードイメージの高さ ＝ バーコードの上端から基準点までの長さ
*2: パラメータ row≠0
*3: MP-B20 及び MP-A40 のみサポートします。

(2) バーコードイメージの幅

pdf417Symol	moduleSize	バーコードイメージの幅
PDF417_STANDARD	**PDF417_MODULE_WIDTH_2**	(17×column[*1]+69)×2+8
	PDF417_MODULE_WIDTH_3	(17×column[*1]+69)×3+12
	PDF417_MODULE_WIDTH_4	(17×column[*1]+69)×4+16
	PDF417_MODULE_WIDTH_5[*2]	(17×column[*1]+69)×5+20
	PDF417_MODULE_WIDTH_6[*2]	(17×column[*1]+69)×6+24
	PDF417_MODULE_WIDTH_7[*2]	(17×column[*1]+69)×7+28
	PDF417_MODULE_WIDTH_8[*2]	(17×column[*1]+69)×8+32
PDF417_COMPACT	**PDF417_MODULE_WIDTH_2**	(17×column[*1]+35)×2+8
	PDF417_MODULE_WIDTH_3	(17×column[*1]+35)×3+12
	PDF417_MODULE_WIDTH_4	(17×column[*1]+35)×4+16
	PDF417_MODULE_WIDTH_5[*2]	(17×column[*1]+35)×5+20
	PDF417_MODULE_WIDTH_6[*2]	(17×column[*1]+35)×6+24
	PDF417_MODULE_WIDTH_7[*2]	(17×column[*1]+35)×7+28
	PDF417_MODULE_WIDTH_8[*2]	(17×column[*1]+35)×8+32

*1: パラメータ column≠0
*2: MP-B20 及び MP-A40 のみサポートします。

B.3.3 PrintQRCode, PrintPageModeQRCode



(1) バーコードイメージの高さ及び幅

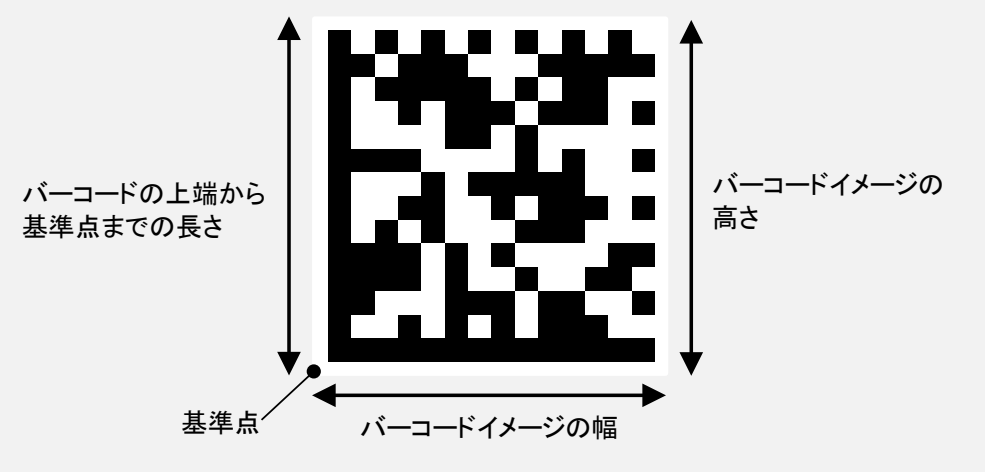
バーコードイメージの高さ[*1] 及び幅 ＝（4 × バージョン[*2] ＋ 17 + 8）× モジュールサイズ値

*1: バーコードイメージの高さ ＝ バーコードの上端から基準点までの長さ
*2: バージョンはバーコードデータの内容及びエラー訂正レベルにより決まります。

moduleSize	モジュールサイズ値
QR_MODULE_SIZE_2	2
QR_MODULE_SIZE_3	3
QR_MODULE_SIZE_4	4
QR_MODULE_SIZE_5	5
QR_MODULE_SIZE_6	6
QR_MODULE_SIZE_7	7
QR_MODULE_SIZE_8	8
QR_MODULE_SIZE_9	9
QR_MODULE_SIZE_10	10
QR_MODULE_SIZE_11	11
QR_MODULE_SIZE_12[*1]	12
QR_MODULE_SIZE_13[*1]	13
QR_MODULE_SIZE_14[*1]	14
QR_MODULE_SIZE_15[*1]	15
QR_MODULE_SIZE_16[*1]	16

*1: MP-B20 及び MP-A40 のみサポートします。

B.3.4 PrintDataMatrix, PrintPageModeDataMatrix



(1) バーコードイメージの高さ及び幅

バーコードイメージの高さ ＝（縦方向モジュール数 + 2）× モジュールサイズ値

バーコードイメージの幅 ＝（横方向モジュール数 + 2）× モジュールサイズ値

dataMatrixModule	縦方向モジュール数	横方向モジュール数
DATA_MATRIX_10_10	10	10
DATA_MATRIX_12_12	12	12
DATA_MATRIX_14_14	14	14
DATA_MATRIX_16_16	16	16
DATA_MATRIX_18_18	18	18
DATA_MATRIX_20_20	20	20
DATA_MATRIX_22_22	22	22
DATA_MATRIX_24_24	23	23
DATA_MATRIX_26_26	26	26
DATA_MATRIX_32_32	32	32
DATA_MATRIX_36_36	36	36
DATA_MATRIX_40_40	40	40
DATA_MATRIX_44_44	44	44
DATA_MATRIX_48_48	48	48
DATA_MATRIX_52_52	52	52
DATA_MATRIX_64_64	64	64
DATA_MATRIX_72_72	72	72
DATA_MATRIX_80_80	80	80
DATA_MATRIX_88_88	88	88
DATA_MATRIX_96_96	96	96
DATA_MATRIX_104_104	104	104

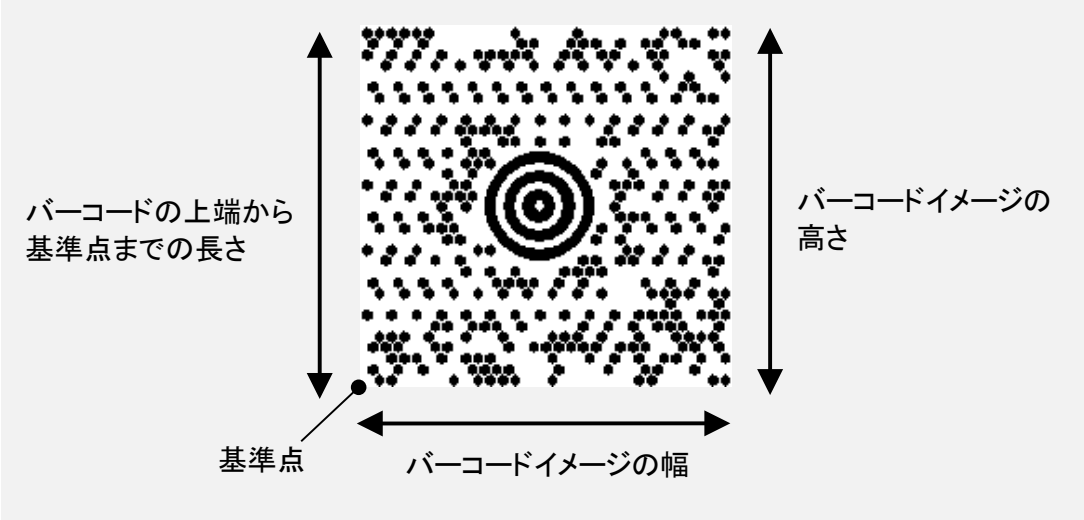
dataMatrixModule	縦方向モジュール数	横方向モジュール数
DATA_MATRIX_120_120	120	120
DATA_MATRIX_132_132	132	132
DATA_MATRIX_144_144	144	144
DATA_MATRIX_8_18	8	18
DATA_MATRIX_8_32	8	32
DATA_MATRIX_12_26	12	26
DATA_MATRIX_12_36	12	36
DATA_MATRIX_16_36	16	36
DATA_MATRIX_16_48	16	48

モジュールサイズ値

moduleSize	モジュールサイズ値
DATAMATRIX_MODULE_SIZE_2	2
DATAMATRIX_MODULE_SIZE_3	3
DATAMATRIX_MODULE_SIZE_4	4
DATAMATRIX_MODULE_SIZE_5	5
DATAMATRIX_MODULE_SIZE_6	6
DATAMATRIX_MODULE_SIZE_7	7
DATAMATRIX_MODULE_SIZE_8	8
DATAMATRIX_MODULE_SIZE_9	9
DATAMATRIX_MODULE_SIZE_10	10
DATAMATRIX_MODULE_SIZE_11	11
DATAMATRIX_MODULE_SIZE_12[*1]	12
DATAMATRIX_MODULE_SIZE_13[*1]	13
DATAMATRIX_MODULE_SIZE_14[*1]	14
DATAMATRIX_MODULE_SIZE_15[*1]	15
DATAMATRIX_MODULE_SIZE_16[*1]	16

*1: MP-B20 及び MP-A40 のみサポートします。

B.3.5 PrintMaxicode, PrintPageModeMaxicode



(1) バーコードイメージの高さ

バーコードイメージの高さ*1 ＝ 215

*1: バーコードイメージの高さ ＝ バーコードの上端から基準点までの長さ

(2) バーコードイメージの幅

バーコードイメージの幅 ＝ 225

B.3.6 PrintGS1DataBarStacked, PrintPageModeGS1DataBarStacked

MP-B20、MP-A40のみサポートします。



(1) バーコードイメージの高さ及び幅

バーコードイメージの高さ*1 ＝ 13 × モジュールサイズ値

*1: バーコードイメージの高さ ＝ バーコードの上端から基準点までの長さ

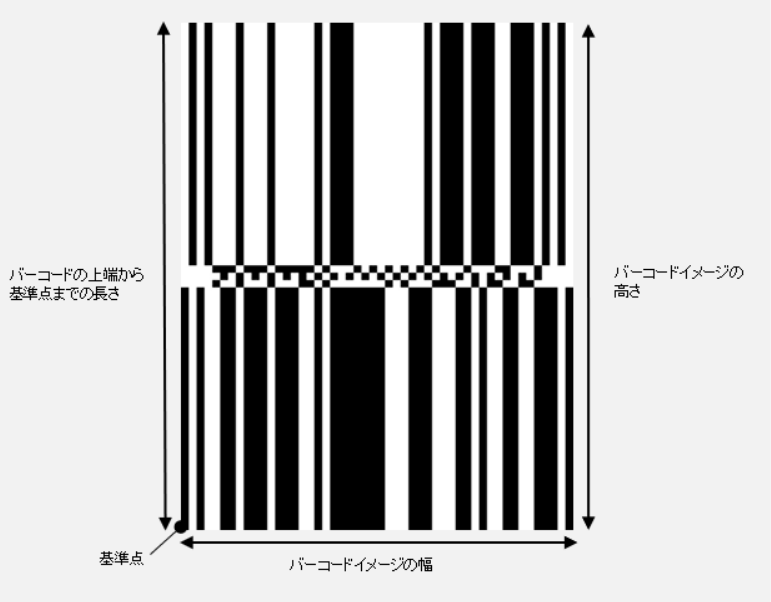
バーコードイメージの幅 ＝ 50 × モジュールサイズ値

モジュールサイズ値

moduleSize	モジュールサイズ値
GS1DATABAR_MODULE_SIZE_2	2
GS1DATABAR_MODULE_SIZE_3	3
GS1DATABAR_MODULE_SIZE_4	4
GS1DATABAR_MODULE_SIZE_5	5
GS1DATABAR_MODULE_SIZE_6	6
GS1DATABAR_MODULE_SIZE_7	7
GS1DATABAR_MODULE_SIZE_8	8
GS1DATABAR_MODULE_SIZE_9	9
GS1DATABAR_MODULE_SIZE_10	10
GS1DATABAR_MODULE_SIZE_11	11
GS1DATABAR_MODULE_SIZE_12	12
GS1DATABAR_MODULE_SIZE_13	13
GS1DATABAR_MODULE_SIZE_14	14
GS1DATABAR_MODULE_SIZE_15	15
GS1DATABAR_MODULE_SIZE_16	16

B.3.7 PrintGS1DataBarStackedOmnidirectional, PrintPageModeGS1DataBarStackedOmnidirectional

MP-B20及びMP-A40のみサポートします。



(1) バーコードイメージの高さ及び幅

バーコードイメージの高さ*1 ＝（moduleHeight × 2 ＋ 3）× モジュールサイズ値

*1: バーコードイメージの高さ ＝ バーコードの上端から基準点までの長さ

バーコードイメージの幅 ＝ 50 × モジュールサイズ値

モジュールサイズ値

moduleSize	モジュールサイズ値
GS1DATABAR_MODULE_SIZE_2	2
GS1DATABAR_MODULE_SIZE_3	3
GS1DATABAR_MODULE_SIZE_4	4
GS1DATABAR_MODULE_SIZE_5	5
GS1DATABAR_MODULE_SIZE_6	6
GS1DATABAR_MODULE_SIZE_7	7
GS1DATABAR_MODULE_SIZE_8	8
GS1DATABAR_MODULE_SIZE_9	9
GS1DATABAR_MODULE_SIZE_10	10
GS1DATABAR_MODULE_SIZE_11	11
GS1DATABAR_MODULE_SIZE_12	12
GS1DATABAR_MODULE_SIZE_13	13
GS1DATABAR_MODULE_SIZE_14	14
GS1DATABAR_MODULE_SIZE_15	15
GS1DATABAR_MODULE_SIZE_16	16

B.3.8 PrintGS1DataBarExpandedStacked, PrintPageModeGS1DataBarExpandedStacked

MP-B20及びMP-A40のみサポートします。



(1) バーコードイメージの高さ及び幅

バーコードイメージの高さ[*1] ＝ ((34 ＋ 3) × 段数[*2] ＋ 34) × モジュールサイズ値

*1: バーコードイメージの高さ＝バーコードの上端から基準点までの長さ
*2: 段数はバーコードデータによって決まります。

バーコードイメージの幅 ＝ (4 ＋ 49 × column / 2) × モジュールサイズ値

モジュールサイズ値

moduleSize	モジュールサイズ値
GS1DATABAR_MODULE_SIZE_2	2
GS1DATABAR_MODULE_SIZE_3	3
GS1DATABAR_MODULE_SIZE_4	4
GS1DATABAR_MODULE_SIZE_5	5
GS1DATABAR_MODULE_SIZE_6	6
GS1DATABAR_MODULE_SIZE_7	7
GS1DATABAR_MODULE_SIZE_8	8
GS1DATABAR_MODULE_SIZE_9	9
GS1DATABAR_MODULE_SIZE_10	10
GS1DATABAR_MODULE_SIZE_11	11
GS1DATABAR_MODULE_SIZE_12	12
GS1DATABAR_MODULE_SIZE_13	13
GS1DATABAR_MODULE_SIZE_14	14
GS1DATABAR_MODULE_SIZE_15	15
GS1DATABAR_MODULE_SIZE_16	16

付録 C　オープンソースソフトウェアライセンス

本章ではライブラリで使用しているオープンソースソフトウェアのライセンスを記載します。

C.1 MIT License

- **SSZipArchive**

  Copyright (c) 2010-2012 Sam Soffes

  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.

C.2 Apache License 2.0

- **zxingify-objc**

  Copyright 2012 ZXing authors

  Licensed under the Apache License, Version 2.0 (the "License");
  you may not use this file except in compliance with the License.
  You may obtain a copy of the License at

  https://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and limitations under the License.

- **zxing-android-embedded**

  Copyright (C) 2012-2022 ZXing authors, Journey Mobile

  Licensed under the Apache License, Version 2.0 (the "License");
  you may not use this file except in compliance with the License.
  You may obtain a copy of the License at

  https://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and limitations under the License.