



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

Rev.03

[対象製品]
MP-B30 シリーズ

セイコーインスツル株式会社

Rev.01	2019 年	1 月
Rev.02	2019 年	2 月
Rev.03	2019 年	12 月

©セイコーインスツル株式会社 2019

無断転載を禁じます。

Bluetooth® は、Bluetooth SIG, Inc.の登録商標です。

Microsoft®及び Windows®は米国 Microsoft Corporation の米国、日本及びその他の国における登録商標です。

文中の社名、商品名等は各社の商標または登録商標である場合があります。

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 Windows CE」(以降:SDK)について説明します。

本書の表記

本書の表記について説明します。

用語一覧

本書で使用する用語について以下の通りに定義します。

用語	内容
プリンターコマンド	「MP-B30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、プリンターを制御するための命令

記号

本書で使用する記号について以下に記載します。

注意

- ◆ 注意事項や制限事項を記載しています。

参考

- 補足説明や関連事項を記載しています。

目次

1章 概要 1-1

1.1	機能	1-1
1.2	構成	1-1
1.2.1	構成図	1-1
1.2.2	提供機能	1-2
1.2.3	サンプルプログラム	1-2
1.3	動作環境	1-2
1.3.1	システム環境	1-2
1.3.2	対象製品	1-3
1.4	プリンターの設定	1-3

2章 ライブラリの利用方法 2-1

2.1	提供ファイル	2-1
2.2	Microsoft Visual Studioのプロジェクトへの組み込み	2-2
3.3	WinCEデバイスでの実行条件	2-2

3章 ライブラリの機能 3-1

3.1	ライブラリのデータ送受信処理と制限事項	3-1
3.2	APIリファレンス	3-2
3.2.1	インターフェイス	3-4
3.2.2	クラス	3-6
3.2.3	列挙型	3-51
3.2.4	例外	3-65

4章 サンプルプログラム 4-1

4.1	利用方法	4-1
4.1.1	CAB ファイル	4-1
4.1.2	プロジェクト形式プログラム	4-1
4.2	画面構成	4-2
4.3	注意事項	4-3

付録A 文字セット（文字コード表） A-1

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

1章 概要

本章では、SDKの概要について説明します。

1.1 機能

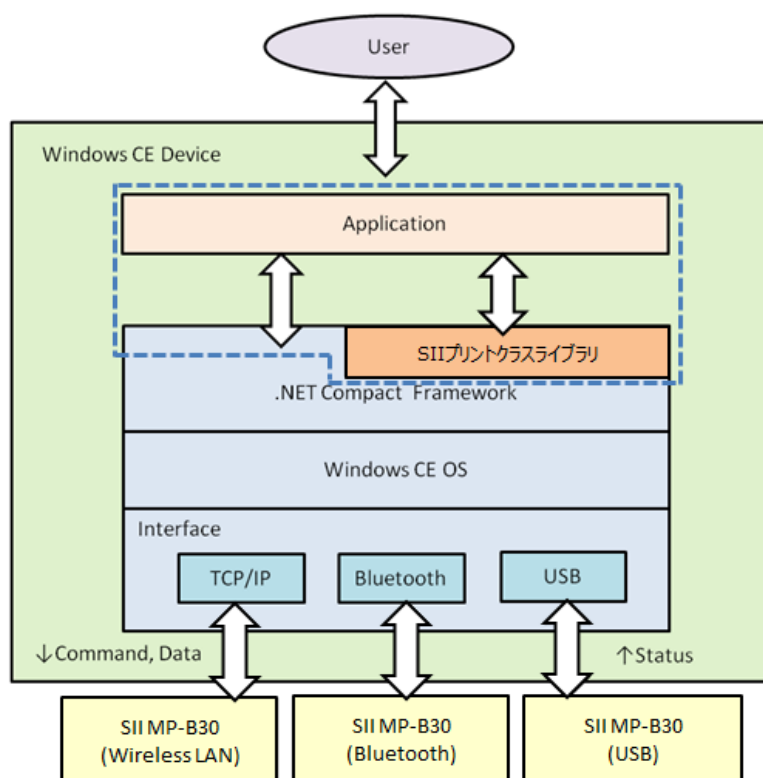
SDKに含まれるSIIプリントクラスライブラリは、Microsoft .NET Compact Framework(以降:.NET Compact Framework)環境で動作するWindows Embedded CE(以降:WinCE)アプリケーションにSII製プリンターMP-B30シリーズ(以降:プリンター)を利用するための機能を提供します。

また、SDKは、SIIプリントクラスライブラリ(以降:ライブラリ)のサンプルプログラムをMicrosoft Visual Studioのプロジェクトで提供します。

1.2 構成

1.2.1 構成図

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。



1.2.2 提供機能

ライブラリを使用することにより、WinCEアプリケーションは、WinCEデバイスの通信ポート(Bluetooth、USB、またはTCP/IP)を通じプリンターに対して印字データやプリンターコマンドを容易に送信することができます。また、プリンターステータスを取得することもできます。

ライブラリでは、下記の機能を提供します。

- プリンターとの接続及び切断
- プリンターへのデータ送信(印字データやプリンターコマンド)
- バーコードの印字、2次元バーコードの印字
- プリンター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)
- プリンターステータスの取得
- プリンターからの各種応答取得
- Bluetooth、またはTCP/IPによるプリンターの探索

1.2.3 サンプルプログラム

ライブラリを使用したWinCEアプリケーションの実装サンプルとして提供します。

1.3 動作環境

1.3.1 システム環境

ライブラリのシステム環境について記載します。

項目	仕様
オペレーティングシステム	Windows CE 5.0 Windows Embedded CE 6.0 Windows Embedded Compact 7 (Windows Mobile 6.1、Windows Mobile 6.5、Windows Embedded Handheld 6.5を含む)
.NET Framework	.NET Framework 3.5 以降 ^{*1}
サポート言語	日本語 英語
サポート開発言語	.NET Compact Frameworkを利用可能な開発言語 Microsoft C# Microsoft Visual Basic

^{*1}：事前にインストールしておく必要があります。

1.3.2 対象製品

プリンター	インターフェイス
MP-B30 シリーズ	USB* ¹
	Bluetooth* ²
	TCP/IP* ³

*1: Windows CE 5.x ベースの OS のみサポート対象となります。また、WinCE デバイスが USB ホスト機能をサポートしており、かつ対象 OS 上に Platform Builder にて 提供される USB Printer class driver (USBPRN.dll) が必要です。

USB Printer class driver は、ライブラリの対象 OS 上では LPT ポートとして扱えます。

*2: SPP (Serial Port Profile) により Bluetooth 接続が確立されている必要があります。

Bluetooth に関するドライバーはマイクロソフト社のシステム標準ドライバーをご利用ください。

*3: WinCE デバイス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。プリンターがクライアントモードのときはアクセスポイント経由で、シンプル AP モードのときは直接接続してください。

1.4 プリンターの設定

ライブラリの利用にあたり、プリンターのメモリスイッチを以下の[値]に設定してください。

メモリスイッチはMP-B30シリーズ用「SII Printer Setting Utility for Windows」により変更できます。

変更方法は、MP-B30シリーズ用「SII Printer Setting Utility for Windows 取扱説明書」の「3.3.2 メモリスイッチ」を参照してください。

• USB接続の場合

MS	機能	値
1-1	通信選択 (Interface)	0 : USB 1 : USB/Wireless
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

• Bluetooth接続の場合

MS	機能	値
1-1	通信選択 (Interface)	1 : USB/Wireless
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

- TCP/IP接続の場合

MS	機能	値
1-1	通信選択 (Interface)	1 : USB/Wireless
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

注意

- ◆ ライブラリはTCP/IP接続時、プリンターのポートを占有します。そのため、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用はできません。

2章 ライブラリの利用方法

本章では、本ライブラリの利用方法について説明します。

2.1 提供ファイル

SDKはライブラリとしてDLLファイルを提供します。SDKのフォルダ構成は、以下の通りです。

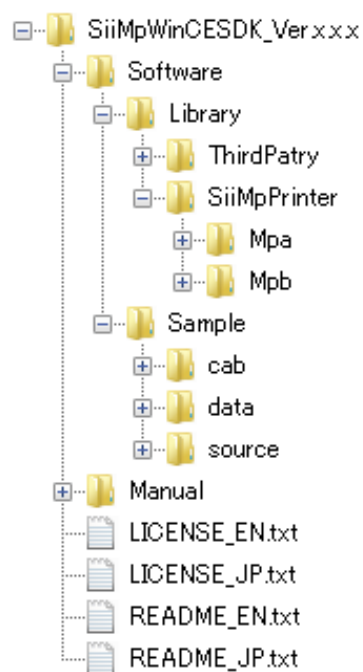


図 2-1

2.2 Microsoft Visual Studioのプロジェクトへの組み込み

SDKに含まれるライブラリをMicrosoft Visual Studioのプロジェクトに組み込む手順を説明します。
説明は、Microsoft Visual Studio C# 2008を基に記載します。

- (1) Microsoft Visual Studioを開き、任意のプロジェクトを作成してください。
- (2) Microsoft Visual Studio の[プロジェクト] – [参照の追加]を選択してください。
- (3) [参照の追加]ダイアログの[参照]に、¥WindowsCE¥Library¥SiiMpPrinter¥Mpx^{*1}のフォルダ下にあるDLLファイルを追加してください。

*1: x はシリーズを示すアルファベット(a または b)が表示されます。

以上で、ライブラリの機能が利用可能になります。

2.3 WinCEデバイスでの実行条件

Microsoft Visual Studioで作成したWinCEアプリケーションをWinCEデバイスで実行する場合は、同じフォルダに下記のファイルを入れて実行してください。

- WinCEアプリケーション (xxxxx.exe)
- ¥WindowsCE¥Library¥SiiMpPrinter¥Mpxフォルダ下にあるDLLファイル
- ¥WindowsCE¥Library¥ThirdPartyフォルダ下にあるDLLファイル

参考

- 開発するプログラムの種類によって、下記のSDKが必要となる場合があります。必要なSDKはMicrosoft社のウェブサイトより入手してください。
 - Windows Mobile 5 Pocket PC SDK
 - Windows Mobile 6 Standard SDK
 - Windows Mobile 6 Professional SDK

3章 ライブラリの機能

本章では、ライブラリに実装されている各クラスのAPIについて説明します。

3.1 ライブラリのデータ送受信処理と制限事項

ライブラリはOpenによるプリンターの利用開始前、SDK内部に送信バッファ用に10MB(10485760バイト)のメモリを確保します。メモリの確保はシステムに依存するので、メモリの確保ができない場合はエラーとなります。ライブラリは、プリンターからの受信データをSDK内部の受信バッファにバッファリングします。プリンターは状態が変化するたびに自動ステータス応答を返すため、受信データは受信バッファに順次バッファリングされます。バッファリングされる受信データのサイズは最大4096バイトです。プリンターが再接続された場合、接続前にプリンターにバッファリングされていたデータをまとめて受信する場合があります。

受信バッファにバッファリングされた受信データは、**Read**により取得が可能です。**Read**により取得した受信データは、受信バッファから削除されます。

バッファリングされた受信データが4096バイトを超えた場合、古いデータから順次破棄されますので、4096バイトを超えないように**Read**を実行してください。受信データのサイズは、**GetReadSize**で取得が可能です。

各メソッドの詳細は、「3.2 APIリファレンス」を参照してください。

各種応答の詳細は、「MP-B30シリーズ サーマルプリンタ技術説明書」を参照してください。

3.2 APIリファレンス

ライブラリのネームスペースは、SII.SPS.Windows.NetCompactFramework.PrintClassLibrary.MobilePrinterです。
SII.SPS.Windows.NetCompactFramework.PrintClassLibrary.MobilePrinterは、下記の機能を提供します。

- インターフェイス

名前	説明
StatusCallbackHandler	プリンターステータスの変化イベントを取得するためのインターフェイス
DiscoveryHandler	プリンター検索時のイベントを取得するためのインターフェイス

- クラス

名前	説明
PrinterManager	プリンターとの通信、印字を行うためのAPIを提供するクラス
PrinterStatus	プリンターステータスを格納するためのクラス
PrinterDiscovery	プリンター探索するためのクラス
DiscoveredPrinter	プリンター探索で発見されたプリンター情報を格納するためのクラス

- 列挙型

名前	説明
PrinterInterface	Openの通信インターフェイス指定で利用する列挙型定数
PrinterModel	Openのプリンターモデル指定で利用する列挙型定数
Dithering	RegisterLogo、SendDataFileのディザリングで利用する列挙型定数
PrinterInformation	GetPrinterInformation、GetPrinterInformationNumber、GetPrinterInformationStringのプリンター情報指定で利用する列挙型定数
TransactionFunction	ControlTransactionの一括処理制御方法指定で利用する列挙型定数
Direction	SelectPageModeの印字方向指定で利用する列挙型定数
Alignment	SetStandardModeAlignmentの位置揃え指定で利用する列挙型定数
CharacterSet	SelectCharacterSet、及びGetCharacterSetの文字セット選択で利用する列挙型定数
InternationalCharacterSet	SelectInternationalCharacterSet、及びGetInternationalCharacterの国際文字セット選択で利用する列挙型定数
CharacterType	SetCharacterFormattingの文字フォント指定で利用する列挙型定数
CharacterScale	SetCharacterFormattingの文字サイズ指定で利用する列挙型定数
Underline	SetCharacterFormattingのアンダーライン指定で利用する列挙型定数
Bold	SetCharacterFormattingの強調印字指定で利用する列挙型定数
Reverse	SetCharacterFormattingの白黒反転印字指定で利用する列挙型定数
Rotate	SetCharacterFormattingの回転印字指定、SetStandardModeBarcodeDirectionの印字方向指定で利用する列挙型定数
TypeBarcode	PrintBarcodeのバーコードの種類指定で利用する列挙型定数

名前	説明
ModuleWidthBarcode	PrintBarcode のバーコードのモジュール幅または細エレメント指定で利用する列挙型定数
HriPositionBarcode	PrintBarcode のバーコードのHRI文字指定で利用する列挙型定数
NwRatioBarcode	PrintBarcode のバーコードのN:W比指定で利用する列挙型定数
Type2Dcode	Print2Dcode の2次元バーコードの種類指定で利用する列挙型定数
Mode2Dcode	Print2Dcode の2次元バーコードのモードで利用する列挙型定数
ModuleSize2Dcode	Print2Dcode の2次元バーコードのモジュールサイズ指定で利用する列挙型定数
ErrorCorrect2Dcode	Print2Dcode の2次元バーコードのエラー訂正レベル指定で利用する列挙型定数
LogFileSize	SetLog のログファイルの最大サイズ指定で利用する列挙型定数
ErrorCode	PrinterException で取得できるエラーコードの列挙型定数

- 例外

名前	説明
PrinterException	API呼び出し時にスローされる例外クラス

3.2.1 インターフェイス

(1) StatusCallbackHandler

- 概要

プリンタステータスの変化によるイベントを取得するためのインターフェイスです。
下記の機能を提供します。

Public Methods

名前	説明
StatusChanged	プリンタステータス変化時の処理登録

- Public Methods

StatusChanged	プリンタステータス変化時の処理登録
---------------	-------------------

構文 `public void StatusChanged(PrinterStatus status)`

パラメーター *status*
 PrinterStatusクラスのインスタンス

説明 本メソッドは、StartCallbackFunctionによりプリンタステータス変化のコールバックが開始され、プリンタステータスが変化した場合に呼び出されます。

インターフェイスのメソッドであるため、実装はありません。プリンタステータス変化のコールバックを受け取るクラスにて、任意の実装を実装してください。

*status*には、変化した時点のプリンタステータスを含んだPrinterStatusクラスのインスタンスを代入します。プリンタステータスは、*status*からPrinterStatusクラスのメソッドにより取得できます。PrinterStatusクラスのメソッドは、「3.2.2(2) PrinterStatus」を参照してください。

(2) DiscoveryHandler

- 概要

プリンター検索時のイベントを取得するためのインターフェイスです。
下記の機能を提供します。

Public Methods

名前	説明
DiscoveryFinished	プリンターの探索終了処理
DiscoveryCancelled	プリンターの探索中断処理

- Public Methods

DiscoveryFinished プリンターの探索終了処理

構文 `public void DiscoveryFinished()`

説明 本メソッドは、**StartDiscoveryPrinter**による探索が終了した場合に呼び出されます。

インターフェイスのメソッドであるため、実装はありません。プリンター探索の終了通知を受け取るクラスにて、任意の処理を実装してください。

DiscoveryCancelled プリンターの探索中断処理

構文 `public void DiscoveryCancelled()`

説明 本メソッドは、**StartDiscoveryPrinter**による探索が中断された場合に呼び出されます。

インターフェイスのメソッドであるため、実装はありません。プリンター探索の中断通知を受け取るクラスにて、任意の処理を実装してください。

3.2.2 クラス

(1) PrinterManager

- 概要

プリンターでの印字やプリンター情報の取得などを行うためのAPIを提供するクラスです。
下記の機能を提供します。

Public Methods

名前	説明
PrinterManager	コンストラクタ
Open	プリンターの利用開始
Close	プリンターの利用終了
SetWriteTimeout	送信タイムアウト時間の設定
GetWriteTimeout	送信タイムアウト時間の取得
SetResponseTimeout	受信タイムアウト時間の設定
GetResponseTimeout	受信タイムアウト時間の取得
Write	バイナリデータの送信
Read	受信データの取得
GetReadSize	取得可能な受信データサイズの取得
WriteAndWaitResponse	バイナリデータの送受信
Reset	プリンターリセット
GetStatus	プリンターステータスの取得
StartCallbackFunction	プリンターステータス変化のコールバック開始
StopCallbackFunction	プリンターステータス変化のコールバック終了
RegisterStyleSheet	スタイルシートの登録 本APIはサポートしていません。
DeleteStyleSheet	スタイルシートの消去 本APIはサポートしていません。
RegisterLogo	ロゴの登録
DeleteLogo	ロゴの登録削除
GetPrinterInformation	プリンター情報の取得
GetPrinterInformationNumber	
GetPrinterInformationString	
ControlTransaction	一括処理の開始・終了
SelectStandardMode	スタンダードモードの選択
SelectPageMode	ページモードの選択
PrintPageModeData	ページモードのデータ印字
SetPageModeVerticalPosition	ページモードにおける縦方向絶対位置の指定
SetStandardModeArea	スタンダードモードの印字領域幅設定
SetStandardModeAlignment	位置揃え
SetHorizontalPosition	絶対位置の指定
SetStandardModeBarcodeDirection	バーコード、2次元バーコードの印字方向の選択

名前	説明
SetLineSpacing	改行量の設定
SetCharacterRightSpace	文字の右スペース量の設定
SelectCharacterSet	文字セットの選択
GetCharacterSet	文字セットの取得
SelectInternationalCharacterSet	国際文字セットの選択
GetInternationalCharacter	国際文字セットの取得
SetCharacterFormatting	文字修飾
PrintText	テキストデータの送信
PrintLogo	ロゴの印字
SendDataFile	ファイルデータの送信
PrintBarcode	バーコードの印字
Print2Dcode	2次元バーコードの印字
PrintPageModeRectangle	ページモードの矩形の描画
FeedLine	行単位の紙送り
FeedDotLine	ドット単位の紙送り
FeedCutPosition	カット位置への紙送り
FeedMarkPosition	マーク紙の頭出し 本APIはサポートしていません。
SetLog	ログ出力の設定

Public Property

名前	説明
IsOpened	プリンターの利用状態取得

- Public Methods

PrinterManager

コンストラクタ

com.seikoinstruments.sdk.mobileprinter.PrinterManagerクラスのコンストラクタです。

構文 public **PrinterManager**()

例外 **PrinterException**

本メソッド呼び出し時にエラーが発生した場合**PrinterException**をスローします。
詳細は、「3.2.4(1) **PrinterException**」を参照してください。

説明 本メソッドはスタンダードモード及びページモードで利用が可能です。

Open

プリンターの利用開始

プリンターの利用を開始します。

構文 public void **Open**(PrinterInterface *prnIf*,
 PrinterModel *prnModel*,
 String *address*,
 Int32 *socketKeepingTime*)

パラメーター *prnIf*

通信インターフェイス定数

利用可能な設定は、「3.2.3(1) **PrinterInterface**」を参照してください。

prnModel

プリンターモデル定数

利用可能な設定は、「3.2.3(2) **PrinterModel**」を参照してください。

address

*prnIf*の設定により異なります。

- ・ **PRN_IF_TCP**を指定する場合

*address*には接続するプリンターのIPアドレスを指定してください。

例: "192.168.0.190"

- ・ **PRN_IF_BT**を指定する場合

*address*には接続するプリンターのBluetoothアドレスを指定してください。

例: "00:11:22:AA:BB:CC"

- ・ **PRN_IF_USB**を指定する場合

*address*にはLPT1: ~ LPT9:の範囲で指定してください。

socketKeepingTime

ソケット維持時間(秒:s)

*prnIf*の設定により異なります。

- ・ **PRN_IF_TCP**を指定する場合

有効範囲は60～300です。

60未満の値を設定した場合は60sに、300を超える値を指定した場合は300sに設定されます。*socketKeepingTime*には、プリンターコマンド「無線LAN通信設定の送信」で取得した受信タイムアウト時間設定と同じ値を指定してください。受信タイムアウト時間設定の初期値は300です。

プリンターコマンドの詳細は、「MP-B30シリーズ サーマルプリンタ技術説明書」を参照してください。

- ・ **PRN_IF_BT**を指定する場合

指定した値は無視されます。

- ・ **PRN_IF_USB**を指定する場合

指定した値は無視されます。

例外

PrinterException

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

詳細は、「3.2.4(1) **PrinterException**」を参照してください。

説明

本メソッドはスタンダードモード及びページモードで利用が可能です。

本メソッドは、*prnIf*で指定した通信インターフェイスにて、*prnModel*で指定したプリンターに接続します。

本メソッドにより、プリンターステータスのモニタリングを開始します。最新のプリンターステータスは**GetStatus**で取得が可能です。プリンターステータスの変化は、**StatusChanged**、**StartCallbackFunction**、及び**StopCallbackFunction**により、イベントとして通知できます。

注意

プリンターコマンド「自動ステータス送信の有効・無効」やプリンターのメモリスイッチの設定により、自動ステータス応答を無効にしないでください。プリンターステータスのモニタリングができず、関連する機能が動作しません。自動ステータス応答の詳細やプリンターのメモリスイッチの設定については「MP-B30シリーズ サーマルプリンタ技術説明書」を参照してください。

TCP/IP接続でのプリンターへのデータ送信後、ソケット維持時間が経過するとライブラリ内部の通信ソケットを破棄します。その後、再度、通信ソケットを作成し接続します。そのため、通信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。

プリンターの利用を終了し、プリンタステータスのモニタリングを終了します。

構文 `public void Close()`

例外 **PrinterException**

本メソッド呼び出し時にエラーが発生した場合**PrinterException**をスローします。
詳細は、「3.2.4(1) **PrinterException**」を参照してください。

説明 本メソッドはスタンダードモード及びページモードで利用が可能です。

ControlTransactionによって保持している送信データは、破棄されます。

SetWriteTimeout

送信タイムアウト時間の設定

送信タイムアウトの時間を設定します。

構文 `public void SetWriteTimeout(Int32 writeTimeout)`

パラメーター *writeTimeout*

送信タイムアウト時間(ミリ秒:ms)

有効範囲は1000～90000です。

1000未満の値を指定した場合は1000msに設定されます。

90000よりも大きい値を指定した場合は90000msに設定されます。

本メソッドで指定しない場合、送信タイムアウト時間は、初期値の10000になります。

説明 本メソッドはスタンダードモード及びページモードで利用が可能です。

設定した値は、**GetWriteTimeout**で取得できます。

本メソッドの設定は、下記のメソッドに有効です。

- ・ **Write**
- ・ **WriteAndWaitResponse** (送信処理部分)
- ・ **RegisterLogo**
- ・ **ControlTransaction** (*control*で**TRANSACTION_PRINT**選択時)
- ・ **PrintText**
- ・ **SendDataFile**
- ・ **PrintBarcode**
- ・ **Print2Dcode**

GetWriteTimeout

送信タイムアウト時間の取得

送信タイムアウトの時間を取得します。

構文 `public Int32 GetWriteTimeout()`

戻り値 送信タイムアウト時間(ミリ秒:ms)

説明 本メソッドはスタンダードモード及びページモードで利用が可能です。

受信タイムアウトの時間を設定します。

構文 `public void SetResponseTimeout(Int32 respTimeout)`

パラメーター *respTimeout*

受信タイムアウト時間(ミリ秒:ms)

有効範囲は1000～90000です。

1000未満の値を指定した場合は1000msに設定されます。

90000よりも大きい値を指定した場合は90000msに設定されます。

本メソッドで指定しない場合、受信タイムアウト時間は、初期値の10000になります。

説明 本メソッドはスタンダードモード及びページモードで利用が可能です。

設定した値は、`GetResponseTimeout`で取得できます。

本メソッドの設定は、下記のメソッドに有効です。

- `WriteAndWaitResponse`
- `GetPrinterInformation`
- `GetPrinterInformationNumber`
- `GetPrinterInformationString`

受信タイムアウトの時間を取得します。

構文 `public Int32 GetResponseTimeout()`

戻り値 受信タイムアウト時間(ミリ秒:ms)

説明 本メソッドはスタンダードモード及びページモードで利用が可能です。

バイナリデータを送信します。

構文 `public void Write(Byte[] binary, Int32 offset)`

パラメーター *binary*

プリンターに送信するバイナリデータ

指定可能なデータサイズは最大16KB(16384バイト)です。

offset

送信するデータの開始位置の指定

例外 **PrinterException**

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

詳細は、「3.2.4(1) **PrinterException**」を参照してください。

説明 本メソッドはスタンダードモード及びページモードで利用が可能です。

ControlTransactionを使用しない場合、本メソッドの送信タイムアウト時間は、**SetWriteTimeout**で設定した値になります。

送信データには、プリンターコマンド「プリンタの初期化」以外のプリンターを初期化するプリンターコマンドを含めないでください。プリンターの初期化については、「MP-B30シリーズ サーマルプリンタ技術説明書」を参照してください。

ハードウェアリセットを行う場合は、**Reset**を実行してください。

本メソッドは**Reset**により中断されます。

Read

受信データの取得

バッファにバッファリングされた受信データを取得します。

構文 `public Byte[] Read(Int32 bufferSize)`

パラメーター *bufferSize*

受信データサイズ(バイト)

有効範囲は1～4096です。

4096よりも大きい値を指定した場合、4096バイトに設定されます。

0以下の値を指定した場合、エラーを通知します。

戻り値 受信データ

例外 **PrinterException**

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

詳細は、「3.2.4(1) **PrinterException**」を参照してください。

説明 本メソッドはスタンダードモード及びページモードで利用が可能です。

バッファにバッファリングされた受信データを全て取得する場合は、*bufferSize*に**GetReadSize**で取得した値を指定してください。

本メソッドによる受信データの取得後、再度本メソッドを実行すると、バッファから続きの受信データを取得します。

受信データの処理と制限事項については、「3.1 ライブラリのデータ送受信処理と制限事項」を参照してください。

GetReadSize

取得可能な受信データサイズの取得

取得可能な受信データサイズを取得します。

構文 `public Int32 GetReadSize()`

戻り値 取得可能な受信データサイズ(バイト)

例外 **PrinterException**

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

詳細は、「3.2.4(1) **PrinterException**」を参照してください。

説明 本メソッドはスタンダードモード及びページモードで利用が可能です。

```
public Byte[] WriteAndWaitResponse(Byte[] sendBinary,
    Int32 bufferSize,
    Boolean respControl,
    Boolean incAshData)
```

bufferSize
受信データの最大サイズ(バイト)
有効範囲は1～10485760です。
10485760よりも大きい値を指定した場合、10485760バイトに設定されます。
0以下の値を指定した場合、エラーを通知します。

respControl

受信処理の動作選択

True: 何らかのデータを受信するか、タイムアウトになるまで受信を継続します。

False: *bufferSize*で指定したサイズのデータを受信するか、タイムアウトになるまで受信を継続します。

incAsbData
受信データに自動ステータス応答を含むか否か
True: 含む
False: 含まない

例外 **PrinterException**
本メソッド呼び出し時にエラーが発生した場合**PrinterException**をスローします。
詳細は、「3.2.4(1) **PrinterException**」を参照してください。

3-13

送信データには、プリンターコマンド「プリンタの初期化」以外のプリンターを初期化するプリンターコマンドを含めないでください。プリンターの初期化については、「MP-B30シリーズ サーマルプリンタ技術説明書」を参照してください。

ハードウェアリセットを行う場合は、**Reset**を実行してください。

本メソッドは**Reset**により中断されます。

Reset

プリンターのリセット

プリンターをリセットします。

構文 `public void Reset()`

例外 **PrinterException**

本メソッド呼び出し時にエラーが発生した場合**PrinterException**をスローします。
詳細は、「3.2.4(1) **PrinterException**」を参照してください。

説明 本メソッドはスタンダードモード及びページモードで利用が可能です。

以下のメソッド処理中に本メソッドを実行すると、処理が中断されます。以下のメソッドの未送信データ、未受信データは破棄されます。

- ・ **Write**
- ・ **WriteAndWaitResponse**
- ・ **ControlTransaction** (*control*で**TRANSACTION_PRINT**選択時)
- ・ **PrintText**
- ・ **SendDataFile**
- ・ **GetPrinterInformation**
- ・ **GetPrinterInformationNumber**
- ・ **GetPrinterInformationString**

本メソッド実行中、プリンターステータスはオフラインとなります。

本メソッド実行後、数秒経過してからデータ送信のメソッドを実行してください。リセット中にデータ送信を行うと、データ抜けの原因となります。

Bluetooth接続で、プリンターがデータを受け付けない状態で本メソッドを実行した場合、本メソッドは成功しますが、リセットはプリンターが印刷可能状態となるまで実行されません。また、それまでの間データ送信は行えません。

本メソッドは、USB接続をサポートしません。

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。

構文 `public PrinterStatus GetStatus()`

戻り値 プリンターステータスを**PrinterStatus**クラスで返します。
PrinterStatusクラスのメソッドにより、プリンターステータスの内容が取得できます。
PrinterStatusクラスのメソッドは、「3.2.2(2) **PrinterStatus**」を参照してください。

例外 **PrinterException**
本メソッド呼び出し時にエラーが発生した場合**PrinterException**をスローします。
詳細は、「3.2.4(1) **PrinterException**」を参照してください。

説明 本メソッドはスタンダードモード及びページモードで利用が可能です。

StartCallbackFunction プリンタステータス変化のコールバック開始

プリンタステータスの変化に応じて実行するコールバックを開始します。

構文 `public void StartCallbackFunction(StatusCallbackHandler handler)`

パラメーター *handler*
StatusCallbackHandlerのインスタンス

例外 **PrinterException**
本メソッド呼び出し時にエラーが発生した場合**PrinterException**をスローします。
詳細は、「3.2.4(1) **PrinterException**」を参照してください。

説明 本メソッドはスタンダードモード及びページモードで利用が可能です。
コールバックで実行する処理は、**StatusChanged**にて登録します。

StopCallbackFunction プリンタステータス変化のコールバック終了

StartCallbackFunctionで開始されたコールバックを終了します。

構文 `public void StopCallbackFunction()`

例外 **PrinterException**
本メソッド呼び出し時にエラーが発生した場合**PrinterException**をスローします。
詳細は、「3.2.4(1) **PrinterException**」を参照してください。

説明 本メソッドはスタンダードモード及びページモードで利用が可能です。

RegisterStyleSheet スタイルシートの登録

本APIはサポートしていません。
呼び出した場合は、**PrinterException**をスローし、**ERR_UNSUPPORTED_METHOD**を返します。

構文 `public void RegisterStyleSheet(String filePath, Int32 regNum)`

DeleteStyleSheet スタイルシートの消去

本APIはサポートしていません。
呼び出した場合は、**PrinterException**をスローし、**ERR_UNSUPPORTED_METHOD**を返します。

構文 `public void DeleteStyleSheet(Int32 regNum)`

イメージデータをロゴとしてプリンターに登録します。

構文(a)のメソッドは、ディザリングはディザリング有効に固定されます。

構文(b)のメソッドは、ディザリングを指定できます。

構文 (a) public void **RegisterLogo**(String *filePath*, Int32 *regNum*)

(b) public void **RegisterLogo**(String *filePath*, Int32 *regNum*, Dithering *dithering*)

パラメーター *filePath*

イメージデータのファイルパス

regNum

ロゴ番号

有効範囲は0～99です。

dithering

ディザリング

利用可能な設定は、「3.2.3(3) Dithering」を参照してください。

例外

PrinterException

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

詳細は、「3.2.4(1) **PrinterException**」を参照してください。

説明

本メソッドはスタンダードモード及びページモードで利用が可能です。

サポートするイメージデータのファイル拡張子は、.bmpです。

ビットマップデータはモノクロ(二値)のみサポートします。

ビットマップデータを、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。

本メソッドは、十分なメモリが実装されている環境で実行してください。

ビットマップデータの送信が何らかの理由で中断された場合でも、残りのデータは本メソッドで指定したビットマップファイルのファイルサイズまで、ビットマップデータとしての処理が継続されます。このような場合は、インターフェイスにより下記を実行してください。

- ・BluetoothまたはTCP/IPの場合

Resetを実行するか、プリンターを再起動してください。

- ・USBの場合

プリンターを再起動してください。

登録可能なファイルサイズは、最大1MB(1048576バイト)です。

登録可能な画像サイズは、最大幅8192ドット、最大高さ2304ドットです。

本メソッドで登録したロゴの登録状況は、**GetPrinterInformation**または

GetPrinterInformationStringで、*prnInfo*にGET_NV_MEM_KEYCODE_LISTを指定して実行することで確認できます。

既にロゴが登録されているロゴ番号を指定した場合、ロゴを上書きします。

本メソッドの送信タイムアウト時間は、**SetWriteTimeout**で設定した値になります。

プリンターに登録されているロゴを削除します。

構文 `public void DeleteLogo(Int32 regNum)`

パラメーター *regNum*
 ロゴ番号
 有効範囲は0～99です。

例外 **PrinterException**
 本メソッド呼び出し時にエラーが発生した場合**PrinterException**をスローします。
 詳細は、「3.2.4(1) **PrinterException**」を参照してください。

説明 本メソッドはスタンダードモード及びページモードで利用が可能です。

 本メソッドで削除したロゴの削除状況は、**GetPrinterInformation**または
GetPrinterInformationStringで、*prnInfo*にGET_NV_MEM_KEYCODE_LISTを指定して実行することで確認できます。

GetPrinterInformation
GetPrinterInformationNumber
GetPrinterInformationString

プリンター情報の取得

プリンター情報を取得します。

構文 (a) `public Byte[] GetPrinterInformation(PrinterInformation prnInfo)`

 (b) `public Int32 GetPrinterInformationNumber(PrinterInformation prnInfo)`

 (c) `public String GetPrinterInformationString(PrinterInformation prnInfo)`

パラメーター *prnInfo*
 取得するプリンター情報の種類
 利用可能な設定、および取得できるプリンター情報の一覧は、
 「3.2.3(4) **PrinterInformation**」を参照してください。
 取得できるプリンター情報の詳細については、「MP-B30シリーズ サーマルプリンタ技術説明書」を参照してください。

戻り値 プリンター情報

例外 **PrinterException**
 本メソッド呼び出し時にエラーが発生した場合**PrinterException**をスローします。
 詳細は、「3.2.4(1) **PrinterException**」を参照してください。

説明 本メソッドはスタンダードモード及びページモードで利用が可能です。

*prnInfo*の指定に基づき、プリンター情報を応答させるプリンターコマンドをプリンターへ送信し、
プリンターからの応答データを応答識別子に基づき解析し、数値配列、数値、または文字列で返します。

 全ての**PrinterInformation**において構文(a)を用いて数値配列でプリンター情報を取得できます。

以下のPrinterInformationにおいて構文(b)を用いて数値でプリンター情報を取得できます。

- GET_NV_MEM_CAP
- GET_NV_MEM_REM_CAP
- GET_REM_USER_MEM_CAP
- GET_REM_USER_MEM_CAP_DEFRAG
- GET_PRN_ID_MODEL
- GET_PRN_ID_TYPE
- GET_PRN_ID_ROM_VER
- GET_PRN_ID_FIRM_CHECKSUM_BOOT
- GET_PRN_ID_FIRM_CHECKSUM_MAIN
- GET_PRN_ID_FIRM_CHECKSUM
- GET_MAINT_NUM_FEED_LINE
- GET_MAINT_NUM_HEAD_ACTIVE
- GET_MAINT_DRIVE_TIME
- GET_MAINT_NUM_FEED_LINE_INTEGRATION
- GET_MAINT_NUM_HEAD_ACTIVE_INTEGRATION
- GET_MAINT_DRIVE_TIME_INTEGRATION
- GET_HFONT_24_CHECKSUM
- GET_HFONT_16_CHECKSUM
- GET_FFONT_CHECKSUM
- GET_FFONT_DATA_SIZE

以下のPrinterInformationにおいて構文(c)を用いて文字列でプリンター情報を取得できます。

- GET_NV_MEM_KEYCODE_LIST
- GET_PRN_ID_FIRM_VER_MAIN
- GET_PRN_ID_MFR
- GET_PRN_ID_MODEL_NAME
- GET_PRN_ID_FIRM_VER_BOOT
- GET_HFONT_24_ID
- GET_HFONT_24_INT_CHAR
- GET_HFONT_16_ID
- GET_HFONT_16_INT_CHAR
- GET_FFONT_LANG
- GET_FFONT_STANDARD
- GET_FFONT_COMPANY

本メソッドの受信タイムアウト時間は、SetResponseTimeoutで設定した値になります。

本メソッドはResetにより中断されます。

一括処理対象メソッドの送信データをバッファリングし、送信バッファ内の送信データをプリンターへ送信します。

構文 `public void ControlTransaction(TransactionFunction control)`

パラメーター *control*

一括処理の動作選択

利用可能な設定は、「3.2.3(5) TransactionFunction」を参照してください。

例外 **PrinterException**

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

詳細は、「3.2.4(1) PrinterException」を参照してください。

説明 本メソッドはスタンダードモード及びページモードで利用が可能です。

一括処理の手順は以下の通りです。

①一括処理を開始します。

TRANSACTION_STARTを指定してください。

一括処理開始前に、**TRANSACTION_CLEAR**または**TRANSACTION_PRINT**を指定するとエラーとなります。

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サイズは最大10MB(10485760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、超えた時点の一括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、エラー前までの送信データが保持されますので、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を指定した場合は、バッファリングを終了し、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。

送信データの送信処理は以下の通りです。

・送信タイムアウト時間は、**SetWriteTimeout**で設定した値になります。

・エラーとなった場合は、未送信の送信データは破棄されます。

・**Reset**が実行された場合は、送信は中断され、未送信の送信データは破棄されます。

・別スレッドから**TRANSACTION_START**を実行した場合は、送信は継続され、新たに別スレッドでバッファリングを開始します。

TRANSACTION_STARTを指定した場合は、バッファリングは継続しますが、バッファリングされた送信データは破棄されます。

TRANSACTION_CLEARを指定した場合は、バッファリングを中止し、バッファリングされた送信データは破棄されます。

一括処理対象メソッドを以下に示します。

- ・ **Write**
- ・ **SelectStandardMode**
- ・ **SelectPageMode**
- ・ **PrintPageModeData**
- ・ **SetPageModeVerticalPosition**
- ・ **SetStandardModeArea**
- ・ **SetStandardModeAlignment**
- ・ **SetHorizontalPosition**
- ・ **SetStandardModeBarcodeDirection**
- ・ **SetLineSpacing**
- ・ **SetCharacterRightSpace**
- ・ **SetCharacterFormatting**
- ・ **PrintText**
- ・ **PrintLogo**
- ・ **SendDataFile**
- ・ **PrintBarcode**
- ・ **Print2Dcode**
- ・ **PrintPageModeRectangle**
- ・ **FeedLine**
- ・ **FeedDotLine**
- ・ **FeedCutPosition**

SelectStandardMode

スタンダードモードの選択

スタンダードモードを選択します。

構文 `public void SelectStandardMode()`

例外 **PrinterException**

本メソッド呼び出し時にエラーが発生した場合**PrinterException**をスローします。
詳細は、「3.2.4(1) **PrinterException**」を参照してください。

説明 本メソッドはページモードでのみ利用が可能です。

Openが実行された直後はスタンダードモードです。

SelectPageModeによりページモードが選択されていた場合、ページモードを終了し、プリンターのページモードデータを破棄します。

ページモードを選択し、印字領域と印字方向を指定します。

構文 `public void SelectPageMode(Int32 startX,
 Int32 startY,
 Int32 width,
 Int32 height,
 Direction direction)`

パラメーター *startX*
 横方向始点(ドット)
 有効範囲は0～2399です。

startY
 縦方向始点(ドット)
 有効範囲は0～2399です。

width
 印字領域幅(ドット)
 有効範囲は1～2400です。

height
 印字領域高さ(ドット)
 有効範囲は1～2400です。

direction
 印字方向
 利用可能な設定は、「3.2.3(6) **Direction**」を参照してください。

例外 **PrinterException**
 本メソッド呼び出し時にエラーが発生した場合**PrinterException**をスローします。
 詳細は、「3.2.4(1) **PrinterException**」を参照してください。

説明 本メソッドはスタンダードモードでのみ利用が可能です。

*startX*が印字幅を超える場合、本メソッドによる印字領域の設定は無効となります。
(*startX* + *width*)が印字幅を超える場合、*startX*は指定した値に設定され、*width*は
(印字幅 - *startX*)に設定されます。
(*startY* + *height*)が2400以上の場合、*startY*は指定した値に設定され、*height*は
(2400 - *startY*)に設定されます。

指定した印字領域内において、**SetHorizontalPosition**、及び**SetPageModeVerticalPosition**によりキャラクタ、イメージ、バーコード、2次元バーコードを任意の印字位置に配置し、ページモードデータを構成してください。

SelectStandardModelによりページモードが中断された場合は、ページモードのデータは破棄されます。

プリンターが初期化された場合、スタンダードモードに戻ります。
プリンターの初期化については、「MP-B30シリーズ サーマルプリンタ技術説明書」を参照してください。

ページモードデータを印字します。

構文 `public void PrintPageModeData()`

例外 **PrinterException**

本メソッド呼び出し時にエラーが発生した場合**PrinterException**をスローします。
詳細は、「3.2.4(1) **PrinterException**」を参照してください。

説明 本メソッドはページモードでのみ利用が可能です。スタンダードモードでは無視されます。
本メソッドを使用する場合、事前に**SelectPageMode**を実行し、ページモードを選択してください。

本メソッドは、**SelectPageMode**によりページモードを選択し、印字領域内にページモードデータを構成した後、実行してください。

ページモードの印字領域内で、印字データの縦方向の絶対位置を指定します。

構文 `public void SetPageModeVerticalPosition(Int32 verticalPosition)`

パラメーター *verticalPosition*

縦方向のデータ展開開始位置(ドット)
有効範囲は0～2399です。

例外 **PrinterException**

本メソッド呼び出し時にエラーが発生した場合**PrinterException**をスローします。
詳細は、「3.2.4(1) **PrinterException**」を参照してください。

説明 本メソッドはページモードでのみ利用が可能です。スタンダードモードでは無視されます。
本メソッドを使用する場合、事前に**SelectPageMode**を実行し、ページモードを選択してください。

始点は、**SelectPageMode**の*direction*(印字方向)設定により変わります。印字方向と始点については、「3.2.3(6) **Direction**」を参照してください。

印字領域を超える指定は無視されます。

プリンターが初期化された場合、本メソッドの設定は無効になります。

プリンターの初期化については、「MP-B30シリーズ サーマルプリンタ技術説明書」を参照してください。

スタンダードモードの印字領域幅を設定します。

構文 `public void SetStandardModeArea(Int32 leftMargin, Int32 prnAreaWidth)`

パラメーター *leftMargin*
 左マージン位置(ドット)
 有効範囲は0～575です。

prnAreaWidth
 印字領域幅(ドット)
 有効範囲は1～576です。

例外 **PrinterException**
 本メソッド呼び出し時にエラーが発生した場合**PrinterException**をスローします。
 詳細は、「3.2.4(1) **PrinterException**」を参照してください。

説明 本メソッドはスタンダードモードでのみ利用が可能です。本メソッドをページモードで実行した場合、設定はスタンダードモード開始後の印字に反映されます。
 本メソッドを使用する場合、事前に**SelectStandardMode**を実行し、スタンダードモードを選択してください。

左マージン位置と印字領域幅の関係を図 3-1に示します。印字データは、網掛けで示す印字領域に展開されます。

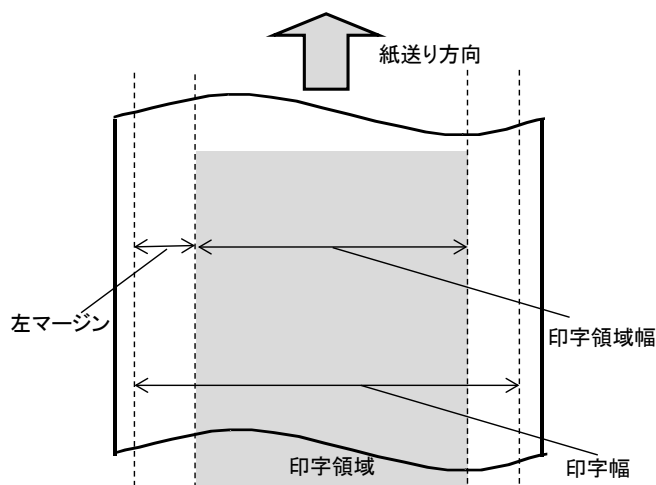


図 3-1

プリンターに印字データが残っていると、本メソッドは実行されません。印字を全て終わらせてから本メソッドを実行してください。

*leftMargin*に印字幅を超える値を指定した場合、印字幅が設定されます。*prnAreaWidth*に印字幅を超える値を指定した場合、印字幅と左マージンの差が設定されます。

プリンターが初期化された場合、*leftMargin*は0、*prnAreaWidth*は印字幅に設定されます。プリンターの初期化については、「MP-B30シリーズ サーマルプリンタ技術説明書」を参照してください。

スタンダードモードの位置揃えを設定します。

構文 `public void SetStandardModeAlignment(Alignment align)`

パラメーター *align*
 位置揃え
 利用可能な設定は、「3.2.3(7) Alignment」を参照してください。

例外 **PrinterException**
 本メソッド呼び出し時にエラーが発生した場合**PrinterException**をスローします。
 詳細は、「3.2.4(1) PrinterException」を参照してください。

説明 本メソッドはスタンダードモードでのみ利用が可能です。本メソッドをページモードで実行した場合、設定はスタンダードモード開始後の印字に反映されます。
 本メソッドを使用する場合、事前にSelectStandardModeを実行し、スタンダードモードを選択してください。

SetStandardModeAreaで印字領域を設定している場合、設定している印字領域内で位置を揃えます。

プリンターに印字データが残っていると、本メソッドは実行されません。印字を全て終わらせてから本メソッドを実行してください。

プリンターが初期化された場合、本メソッドのパラメーターは初期値に設定されます。
 プリンターの初期化については、「MP-B30シリーズ サーマルプリンタ技術説明書」を参照してください。

横方向の絶対位置を指定します。

構文 `public void SetHorizontalPosition(Int32 horizontalPosition)`

パラメーター *horizontalPosition*
 横方向の印字開始位置(ドット)
 有効範囲は0～575です。

例外 **PrinterException**
 本メソッド呼び出し時にエラーが発生した場合**PrinterException**をスローします。
 詳細は、「3.2.4(1) PrinterException」を参照してください。

説明 本メソッドはスタンダードモード及びページモードで利用が可能です。
 スタンダードモードの場合、SetStandardModeAreaで設定した左マージン位置を基準とします。

ページモードの場合、始点はSelectPageModeの*direction*で設定する印字方向により変わります。印字方向と始点については、「3.2.3(6) Direction」を参照してください。

SetStandardModeArea、SelectPageModeで設定した印字領域を超える指定は無視されます。

プリンターが初期化された場合、本メソッドの設定は無効になります。

プリンターの初期化については、「MP-B30シリーズ サーマルプリンタ技術説明書」を参照してください。

SetStandardModeBarcodeDirection

バーコード、2次元バーコードの印字方向の設定

スタンダードモードでのバーコード、2次元バーコードの印字方向を設定します。

構文 `public void SetStandardModeBarcodeDirection(Rotate rotate)`

パラメーター *rotate*

印字方向

ROTATE_90_TO_RIGHT、**ROTATE_90_TO_LEFT**を指定した場合、印字可能なバーコード、2次元バーコードの幅は最大300mmです。300mmを超える場合、バーコード、2次元バーコードは印字されません。

利用可能な設定は、「3.2.3(15) Rotate」を参照してください。

例外 **PrinterException**

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

詳細は、「3.2.4(1) PrinterException」を参照してください。

説明 本メソッドはスタンダードモードでのみ利用が可能です。本メソッドをページモードで実行した場合、設定はスタンダードモード開始後の印字に反映されます。

本メソッドを使用する場合、事前に**SelectStandardMode**を実行し、スタンダードモードを選択してください。

PrintBarcode、または**Print2Dcode**を実行すると、本メソッドの設定は、初期値に戻ります。

プリンターが初期化された場合、本メソッドのパラメーターは初期値に設定されます。

プリンターの初期化については、「MP-B30シリーズ サーマルプリンタ技術説明書」を参照してください。

注意 **SetCharacterFormatting**の*rotate*によってバーコード、2次元バーコードの印字位置と向きが変わる場合があります。印字位置と向きについては、「3.2.3(15) Rotate」を参照してください。

SetLineSpacing

改行量の設定

改行量を設定します。

構文 `public void SetLineSpacing(Int32 lineSpacing)`

パラメーター *lineSpacing*

改行量(ドット)

有効範囲は0~255です。

例外 **PrinterException**

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

詳細は、「3.2.4(1) PrinterException」を参照してください。

説明 本メソッドはスタンダードモードとページモードにそれぞれ独立した改行量の利用が可能です。

プリンターが初期化された場合、*lineSpacing*は34に設定されます。

プリンターの初期化については、「MP-B30シリーズ サーマルプリンタ技術説明書」を参照してください。

SetCharacterRightSpace

文字の右スペース量の設定

文字の右スペース量を設定します。

構文 `public void SetCharacterRightSpace(Int32 space)`

パラメーター *space*
 右スペース量(ドット)
 有効範囲は0～255です。

例外 **PrinterException**
 本メソッド呼び出し時にエラーが発生した場合**PrinterException**をスローします。
 詳細は、「3.2.4(1) **PrinterException**」を参照してください。

説明 本メソッドはスタンダードモードとページモードにそれぞれ独立した右スペース量の利用が可能です。

半角文字及び全角文字に有効です。全角文字の場合、左スペースは0ドットになります。

プリンターが初期化された場合、*space*は0に設定されます。

プリンターの初期化については、「MP-B30シリーズ サーマルプリンタ技術説明書」を参照してください。

SelectCharacterSet

文字セットの選択

文字セットを選択します。

構文 `public void SelectCharacterSet(CharacterSet charSet)`

パラメーター *charSet*
 文字セット
 利用可能な選択は、「3.2.3(8) **CharacterSet**」を参照してください。

例外 **PrinterException**
 本メソッド呼び出し時にエラーが発生した場合**PrinterException**をスローします。
 詳細は、「3.2.4(1) **PrinterException**」を参照してください。

説明 本メソッドはスタンダードモード及びページモードで利用が可能です。

文字セットの初期値は、WinCEデバイスの言語設定により下記の通りです。

日本語の場合： **CODEPAGE_KATAKANA**

日本語以外の場合： **CODEPAGE_1252**

本メソッドの実行後、**Reset**の実行、及びプリンターが初期化されても本メソッドの設定は有効です。プリンターの初期化については、「MP-B30シリーズ サーマルプリンタ技術説明書」を参照してください。

設定されている文字セットを取得します。

構文 `public CharacterSet GetCharacterSet()`

戻り値 文字セット

説明 本メソッドはスタンダードモード及びページモードで利用が可能です。

取得可能な設定は、「3.2.3(8) CharacterSet」を参照してください。

国際文字セットを選択します。

構文 `public void SelectInternationalCharacterSet(InternationalCharacterSet intCharSet)`

パラメーター *intCharSet*

国際文字セット

利用可能な選択は、「3.2.3(9) InternationalCharacterSet」を参照してください。

例外 **PrinterException**

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

詳細は、「3.2.4(1) PrinterException」を参照してください。

説明 本メソッドはスタンダードモード及びページモードで利用が可能です。

国際文字セットの初期値は、WinCEデバイスの言語設定により下記の通りです。

日本語の場合: **INT_CHAR_SET_JAPAN**

日本語以外の場合: **INT_CHAR_SET_USA**

本メソッドの実行後、**Reset**の実行、及びプリンターが初期化されても本メソッドの設定は有効です。プリンターの初期化については、「MP-B30シリーズ サーマルプリンタ技術説明書」を参照してください。

設定されている国際文字セットを取得します。

構文 `public InternationalCharacterSet GetInternationalCharacter()`

戻り値 国際文字セット

説明 本メソッドはスタンダードモード及びページモードで利用が可能です。

取得可能な設定は、「3.2.3(9) InternationalCharacterSet」を参照してください。

構文

```
public void SetCharacterFormatting(CharacterType type,  
    CharacterScale verticalScale,  
    CharacterScale horizontalScale,  
    Underline underline,  
    Bold bold,  
    Reverse reverse,  
    Rotate rotate)
```

rotate

回転印字

ページモードでは、設定は無効です。設定した場合、スタンダードモード開始後の印字に反映されます。

スタンダードモードの場合、プリンターに印字データが残っていると、本メソッドは実行されません。印字を全て終わらせてから本メソッドを実行してください。

スタンダードモードでROTATE_90_TO_RIGHT、ROTATE_90_TO_LEFTを指定した場合、下記の動作となります。

- ・アンダーライン印字は行われません。
- ・*verticalScale*と*horizontalScale*の縦横の倍率は逆になります。

*rotate*の指定は、バーコード、2次元バーコードの印字位置と向きに影響する場合があります。

利用可能な設定、及びバーコード、2次元バーコードの印字位置と向きについては、「3.2.3(15) Rotate」を参照してください。

例外	PrinterException 本メソッド呼び出し時にエラーが発生した場合 PrinterException をスローします。 詳細は、「3.2.4(1) PrinterException 」を参照してください。
説明	本メソッドはスタンダードモード及びページモードで利用が可能です。 プリンターが初期化された場合、本メソッドのパラメーターは初期値に設定されます。 プリンターの初期化については、「MP-B30シリーズ サーマルプリンタ技術説明書」を参照してください。

PrintText

テキストデータの送信

テキストデータを送信します。

構文 `public void PrintText(String text)`

パラメーター *text*
 プリンターに送信するテキストデータ

例外 **PrinterException**
 本メソッド呼び出し時にエラーが発生した場合**PrinterException**をスローします。
 詳細は、「3.2.4(1) **PrinterException**」を参照してください。

説明 本メソッドはスタンダードモード及びページモードで利用が可能です。

本メソッドは、*text*で指定したテキストデータを、**SelectCharacterSet**で指定した文字セット設定を基に、プリンターが認識可能なテキストデータへエンコードを行った後、プリンターへ送信します。

送信データにヘッダとしてプリンターコマンド「文字コードテーブルの選択」「国際文字の選択」「漢字コード体系の選択」「漢字モードの解除」を付加します。
 プリンターコマンド「文字コードテーブルの選択」及び「国際文字の選択」は、**SelectCharacterSet**と**SelectInternationalCharacterSet**の設定に基づき送信されます。
 プリンターコマンド「漢字コード体系の選択」は、**SelectInternationalCharacterSet**の設定に基づき、下記の漢字コード体系が選択されます。

SelectInternationalCharacterSetが**INT_CHAR_SET_JAPAN**の場合: Shift-JIS
SelectInternationalCharacterSetが**INT_CHAR_SET_JAPAN**以外の場合: JIS

プリンターコマンドの詳細は、「MP-B30シリーズ サーマルプリンタ技術説明書」を参照してください。

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。

指定可能なデータサイズは最大16KB(16384バイト)です。

ControlTransactionを使用しない場合、本メソッドの送信タイムアウト時間は、**SetWriteTimeout**で設定した値になります。

本メソッドは**Reset**により中断されます。

プリンターに登録したロゴを印字します。

構文 `public void PrintLogo(Int32 regNum)`

パラメーター *regNum*
 ロゴ番号
 有効範囲は0～99です。

例外 **PrinterException**
 本メソッド呼び出し時にエラーが発生した場合**PrinterException**をスローします。
 詳細は、「3.2.4(1) **PrinterException**」を参照してください。

説明 本メソッドはスタンダードモード及びページモードで利用が可能です。

登録は**RegisterLogo**により行います。

指定された*regNum*にロゴが登録されていない場合、無視されます。

SendDataFile

ファイルデータの送信

プリンターにファイルデータを送信します。

構文(a)のメソッドは、ディザリングはディザリング有効に固定されます。

構文(b)のメソッドは、ディザリングを指定できます。

構文 (a) `public void SendDataFile(String filePath)`
 (b) `public void SendDataFile(String filePath, Dithering dithering)`

パラメーター *filePath*
 プリンターに送信するファイルのパス

 dithering
 ディザリング
 *filePath*で指定されたファイルの拡張子が.bmpの場合に有効です。
 利用可能な設定は、「3.2.3(3) **Dithering**」を参照してください。

例外 **PrinterException**
 本メソッド呼び出し時にエラーが発生した場合**PrinterException**をスローします。
 詳細は、「3.2.4(1) **PrinterException**」を参照してください。

説明 本メソッドはスタンダードモード及びページモードで利用が可能です。

指定したファイルの拡張子に応じて、下記の処理を行います。

- ・ ファイル拡張子が.bmpの場合

ビットマップデータはモノクロ(二値)のみサポートします。

ビットマップデータを、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。

本メソッドは、十分なメモリが実装されている環境で実行してください。

ビットマップデータの送信が何らかの理由で中断された場合でも、残りのデータは本メソッドで指定したビットマップファイルのファイルサイズまで、ビットマップデータとしての処理が継続されます。このような場合は、インターフェイスにより下記を実行してください。

- ・BluetoothまたはTCP/IPの場合

Resetを実行するか、プリンターを再起動してください。

- ・USBの場合

プリンターを再起動してください。

- ・ ファイル拡張子が.txtの場合

テキストデータのフォーマットはUTF-8をサポートします。

テキストデータを、**SelectCharacterSet**で指定した文字セット設定を基に、プリンターが認識可能なテキストデータへエンコードを行った後、プリンターへ送信します。

送信データにヘッダとしてプリンターコマンド「文字コードテーブルの選択」「国際文字の選択」「漢字コード体系の選択」「漢字モードの解除」を付加します。

プリンターコマンド「文字コードテーブルの選択」及び「国際文字の選択」は、

SelectCharacterSetと**SelectInternationalCharacterSet**の設定に基づき送信されます。

プリンターコマンド「漢字コード体系の選択」は、**SelectInternationalCharacterSet**の設定に基づき、下記の漢字コード体系が選択されます。

SelectInternationalCharacterSetが**INT_CHAR_SET_JAPAN**の場合 : Shift-JIS

SelectInternationalCharacterSetが**INT_CHAR_SET_JAPAN**以外の場合 : JIS

プリンターコマンドの詳細は、「MP-B30シリーズ サーマルプリンタ技術説明書」を参照してください。

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。

- ・ ファイル拡張子が.bin、.datの場合

無変換のままプリンターに送信します。

指定可能なファイルサイズは、最大1MB(1048576バイト)です。

ControlTransactionを使用しない場合、本メソッドの送信タイムアウト時間は、**SetWriteTimeout**で設定した値になります。

送信データには、プリンターコマンド「プリンタの初期化」以外のプリンターを初期化するプリンターコマンドを含めないでください。プリンターの初期化については、「MP-B30シリーズ サーマルプリンタ技術説明書」を参照してください。

ハードウェアリセットを行う場合は、**Reset**を実行してください。

本メソッドは**Reset**により中断されます。

バーコードを印字します。

構文

```
(a) public void PrintBarcode(TypeBarcode type,
    String text,
    ModuleWidthBarcode moduleWidth,
    Int32 moduleHeight,
    HriPositionBarcode hri,
    NwRatioBarcode nwRatio)

(b) public void PrintBarcode(TypeBarcode type,
    Byte[] binary,
    ModuleWidthBarcode moduleWidth,
    Int32 moduleHeight,
    HriPositionBarcode hri,
    NwRatioBarcode nwRatio)
```

パラメーター *type*
 バーコードの種類
 利用可能な設定は、「3.2.3(16) **TypeBarcode**」を参照してください。

(a) *text*
 バーコードのテキストデータ

(b) *binary*
 バーコードのバイナリデータ

text、*binary*で利用可能な値を下記に示します。

バーコードの種類	データサイズ	入力可能なバーコードデータ	
UPC-A	11 ～ 12バイト	'0' ～ '9'	(0x30 ～ 0x39)
UPC-E	11 ～ 12バイト	'0' ～ '9'	(0x30 ～ 0x39)
JAN13(EAN13)	12 ～ 13バイト	'0' ～ '9'	(0x30 ～ 0x39)
JAN8(EAN8)	7 ～ 8バイト	'0' ～ '9'	(0x30 ～ 0x39)
CODE39	1 ～ 150バイト	' '	(0x20)
		'\$'	(0x24)
		'%'	(0x25)
		'+'	(0x2B)
		'-'	(0x2D)
		'.'	(0x2E)
		'/'	(0x2F)
		'0' ～ '9'	(0x30 ～ 0x39)
		'A' ～ 'Z'	(0x41 ～ 0x54)
ITF	2 ～ 150バイト (ただし偶数個)	'0' ～ '9'	(0x30 ～ 0x39)

バーコードの種類	データサイズ	入力可能なバーコードデータ	
CODABAR	1 ~ 150バイト	'\$'	(0x24)
		'+'	(0x2B)
		'_'	(0x2D)
		':'	(0x2E)
		'/'	(0x2F)
		'0' ~ '9'	(0x30 ~ 0x39)
		':'	(0x3A)
		'A' ~ 'D'	(0x41 ~ 0x44)
CODE93	1 ~ 150バイト		(0x00 ~ 0x7F)
CODE128	2 ~ 150バイト		(0x00 ~ 0x7F)*1
JAN13アドオン2(EAN13アドオン2)	14 ~ 15バイト	'0' ~ '9'	(0x30 ~ 0x39)
JAN13アドオン5(EAN13アドオン5)	17 ~ 18バイト	'0' ~ '9'	(0x30 ~ 0x39)
GS1 Databar Omni-directional	13バイト	'0' ~ '9'	(0x30 ~ 0x39)
GS1 Databar Truncated	13バイト	'0' ~ '9'	(0x30 ~ 0x39)
GS1 Databar Limited	13バイト	'0' ~ '9'	(0x30 ~ 0x39)
GS1 Databar Expanded	2 ~ 150バイト	' ' ~ ' "'	(0x20 ~ 0x22)
		'%' ~ ' ?'	(0x25 ~ 0x3F)
		'A' ~ 'Z'	(0x41 ~ 0x5A)
		'_'	(0x5F)
		'a' ~ 'z'	(0x61 ~ 0x7A)
		'{'	(0x7B)

*1 : 指定したコードセットにより、入力可能なバーコードデータが異なります。

CODE A : 0x00 ~ 0x50

CODE B : 0x00 ~ 0x7F

CODE C : 00 ~ 99 (0x00 ~ 0x63) の2桁の数字

moduleWidth

バーコードのモジュール幅または細エレメント

利用可能な設定は、「3.2.3(17) ModuleWidthBarcode」を参照してください。

moduleHeight

バーコードのモジュール高さ(ドット)

バーコード種類が下記の場合の利用可能な設定を示します。

バーコード種類	<i>moduleHeight</i>
UPC-A	0*1、1 ~ 255
UPC-E	
JAN13(EAN13)	
JAN8(EAN8)	
CODE39	
ITF	
CODABAR	
CODE93	
CODE128	
JAN13アドオン2(EAN13アドオン2)	
JAN13アドオン5(EAN13アドオン5)	

*1 : 0を指定した場合、高さは自動的にバーコードの初期値 (162ドット) となります。

バーコード種類が下記の場合は、バーコードモジュール幅により、バーコード高さ範囲が異なります。

*moduleHeight*の値を最小値未満に指定した場合は、値は自動的に最小値へ変更されます。

バーコード種類	<i>moduleHeight</i>
GS1 Databar Omni-directional	
MODULE_WIDTH_BARCODE_2	0*1、66 ~ 255
MODULE_WIDTH_BARCODE_3	0*1、99 ~ 255
MODULE_WIDTH_BARCODE_4	0*1、132 ~ 255
MODULE_WIDTH_BARCODE_5	0*2、165 ~ 255
MODULE_WIDTH_BARCODE_6	0*3、198 ~ 255
GS1 Databar Truncated	
MODULE_WIDTH_BARCODE_2	0*1、26 ~ 255
MODULE_WIDTH_BARCODE_3	0*1、39 ~ 255
MODULE_WIDTH_BARCODE_4	0*1、52 ~ 255
MODULE_WIDTH_BARCODE_5	0*1、65 ~ 255
MODULE_WIDTH_BARCODE_6	0*1、78 ~ 255
GS1 Databar Limited	
MODULE_WIDTH_BARCODE_2	0*1、20 ~ 255
MODULE_WIDTH_BARCODE_3	0*1、30 ~ 255
MODULE_WIDTH_BARCODE_4	0*1、40 ~ 255
MODULE_WIDTH_BARCODE_5	0*1、50 ~ 255
MODULE_WIDTH_BARCODE_6	0*1、60 ~ 255

バーコード種類	<i>moduleHeight</i>
GS1 Databar Expanded	
MODULE_WIDTH_BARCODE_2	0*1、68 ~ 255
MODULE_WIDTH_BARCODE_3	0*1、102 ~ 255
MODULE_WIDTH_BARCODE_4	0*1、136 ~ 255
MODULE_WIDTH_BARCODE_5	0*4、170 ~ 255
MODULE_WIDTH_BARCODE_6	0*5、204 ~ 255

- *1 : 0を指定した場合、高さは自動的にバーコードの初期値（162ドット）となります。
 *2 : 0を指定した場合、高さは自動的に165ドットとなります。
 *3 : 0を指定した場合、高さは自動的に198ドットとなります。
 *4 : 0を指定した場合、高さは自動的に170ドットとなります。
 *5 : 0を指定した場合、高さは自動的に204ドットとなります。

hri

バーコードのHRI文字設定

利用可能な設定は、「3.2.3(18) HriPositionBarcode」を参照してください。

nwRatio

バーコードのNW比

利用可能な設定は、*type*の設定により異なります。

詳細は、「3.2.3(19)NwRatioBarcode」を参照してください。

例外

PrinterException

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

詳細は、「3.2.4(1) PrinterException」を参照してください。

説明

本メソッドはスタンダードモード及びページモードで利用が可能です。

スタンダードモードの場合、プリンターに印字データが残っていると、本メソッドは実行されません。印字を全て終わらせてから本メソッドを実行してください。

印字領域を超えた場合は、バーコードは印字されません。

本メソッドは、構文(a)の*text*で指定したテキストデータを、プリンターが認識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。

構文(b)の*binary*で指定したバイナリデータは、そのままプリンターに送信します。

CODE128で指定する特殊コードを下記に示します。

入力データ		コードセット毎の特殊コード		
		CODE A	CODE B	CODE C
{S}	(0x7B, 0x53)	SHIFT	SHIFT	SHIFT
{A}	(0x7B, 0x41)	–	CODE A	CODE A
{B}	(0x7B, 0x42)	CODE B	–	CODE B
{C}	(0x7B, 0x43)	CODE C	CODE C	–
{1}	(0x7B, 0x31)	FNC 1	FNC 1	FNC 1
{2}	(0x7B, 0x32)	FNC 2	FNC 2	FNC 2
{3}	(0x7B, 0x33)	FNC 3	FNC 3	FNC 3
{4}	(0x7B, 0x34)	FNC 4	FNC 4	FNC 4
{I}	(0x7B, 0x7B)	{I	{I	–

下記のバーコードのチェックデジットは自動計算されます。

- ・ UPC-A
- ・ UPC-E
- ・ JAN13(EAN13)
- ・ JAN8(EAN8)
- ・ JAN13アドオン2(EAN13アドオン2)
- ・ JAN13アドオン5(EAN13アドオン5)
- ・ GS1 Databar Omni-directional
- ・ GS1 Databar Truncated
- ・ GS1 Databar Limited

下記のバーコードでチェックデジットを含むデータを指定した場合、チェックデジットデータは無視され、プリンター側で再計算します。

- ・ UPC-A
- ・ UPC-E
- ・ JAN13(EAN)
- ・ JAN8(EAN)

CODE39のスタートコード及びストップコード('*')は自動で付加されますので指定する必要はありません。

ITFのバーコードデータに奇数個のデータを指定した場合、最後のデータが切り捨てられます。

CODABARのバーコードデータにおいて、'A'~'D'はスタートコード及びストップコードとして入力してください。

GS1 Databar Expandedのバーコードデータにおいて、'f'は、FUNC1を指定する場合にのみ使用してください。FUNC1を指定する場合、'f'(0x7B, 0x31)を入力します。

text、*binary*、*moduleWidth*、*moduleHeight*、*nwRatio*のいずれか一つでも、*type*で指定したバーコード種類に対応する値以外を設定した場合、エラーになります。

ControlTransactionを使用しない場合、本メソッドの送信タイムアウト時間は、**SetWriteTimeout**で設定した値になります。

Print2Dcode

2次元バーコードの印字

2次元バーコードを印字します。

構文 (a) public void **Print2Dcode**(Type2Dcode *type*,
String *text*,
Mode2Dcode *mode*,
ModuleSize2Dcode *moduleSize*,
Int32 *moduleHeight*,
Int32 *column*,
Int32 *row*,
ErrorCorrect2Dcode *errorCorrect*)

```
(b) public void Print2Dcode(Type2Dcode type,
    Byte[] binary,
    Mode2Dcode mode,
    ModuleSize2Dcode moduleSize,
    Int32 moduleHeight,
    Int32 column,
    Int32 row,
    ErrorCorrect2Dcode errorCorrect)
```

パラメーター *type*

2次元バーコードの種類

利用可能な設定は、「3.2.3(20) **Type2Dcode**」を参照してください。

(a) *text*

バーコードのテキストデータ

(b) *binary*

バーコードのバイナリデータ

mode

2次元バーコードのモード

利用可能な設定は、*type*の設定により異なります。

詳細は、「3.2.3(21) **Mode2Dcode**」を参照してください。

MODE_2DCODE_MAXI_CODE_2を選択する場合、*text*、または*binary*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)のデータをそれぞれ付加してください。

MODE_2DCODE_MAXI_CODE_3を選択する場合、*text*、または*binary*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)のデータをそれぞれ付加してください。

moduleSize

2次元バーコードのモジュールサイズ

利用可能な設定は、*type*の設定により異なります。

詳細は、「3.2.3(22) **ModuleSize2Dcode**」を参照してください。

moduleHeight

2次元バーコードのモジュール高さ(ドット)

PDF417、及びGS1 Databar Stacked Omni-directionalで有効です。

それ以外の2次元バーコード種類を使用する場合は、0を指定してください。

2次元バーコード種類	<i>moduleHeight</i>	
	PDF段高さ	1段の高さ
PDF417	2 ~ 127	–
GS1 Databar Stacked Omni-directional	–	33 ~ 255

column

2次元バーコードのカラム数

PDF417、及びGS1 Databar Expanded Stackedで有効です。

それ以外の2次元バーコード種類を使用する場合は、0を指定してください。

2次元バーコード種類	<i>column</i>	
	データ領域のカラム数	1行のセグメント数 ^{*1}
PDF417	0 ^{*2} 、1 ～ 30	–
GS1 Databar Expanded Stacked	–	2 ～ 20

*1：1行のセグメント数には偶数値を指定してください。

*2：0を指定した場合、カラム数は自動となります。

row

2次元バーコードの段数

PDF417のみで有効です。

それ以外の2次元バーコード種類を使用する場合は、0を指定してください。

2次元バーコード種類 ^{*1}	<i>row</i>
PDF417	0 ^{*1} 、3 ～ 90

*1：0を指定した場合、段数は自動となります。

errorCorrect

2次元バーコードのエラー訂正レベル

利用可能な設定は、*type*の設定により異なります。

詳細は、「3.2.3(23) **ErrorCorrect2Dcode**」を参照してください。

例外

PrinterException

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

詳細は、「3.2.4(1) **PrinterException**」を参照してください。

説明

本メソッドはスタンダードモード及びページモードで利用が可能です。

スタンダードモードの場合、プリンターに印字データが残っていると、本メソッドは実行されません。印字を全て終わらせてから本メソッドを実行してください。

本メソッドは、構文(a)の*text*で指定したテキストデータを、プリンターが認識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。

*text*には、*type*で指定する2次元バーコード種類の規格に従い、対応する文字データをASCII形式で入力してください。

構文(b)の*binary*で指定したバイナリデータは、そのままプリンターに送信します。

*binary*には、*type*で指定する2次元バーコード種類の規格に従い、対応する文字データをASCII形式の文字コードで入力してください。

text、*binary*、*mode*、*moduleSize*、*moduleHeight*、*column*、*errorCorrect*のいずれか一つでも、*type*で指定した2次元バーコード種類に対応する値以外を設定した場合、エラーになります。

2次元バーコードの幅が、印字領域幅を超える場合、本メソッドは無視されます。

ControlTransactionを使用しない場合、本メソッドの送信タイムアウト時間は、**SetWriteTimeout**で設定した値になります。

ページモードで矩形を描画します。

構文 `public void PrintPageModeRectangle(Int32 startX,
 Int32 startY,
 Int32 width,
 Int32 height,
 Int32 thickness)`

パラメーター *startX*
 横方向始点(ドット)
 有効範囲は0～2395です。

startY
 縦方向始点(ドット)
 有効範囲は0～2395です。

width
 幅(ドット)
 有効範囲は4～2400です。

height
 高さ(ドット)
 有効範囲は4～2400です。

thickness
 線幅(ドット)
 有効範囲は2～40です。

例外 **PrinterException**
 本メソッド呼び出し時にエラーが発生した場合**PrinterException**をスローします。
 詳細は、「3.2.4(1) **PrinterException**」を参照してください。

説明 本メソッドはページモードでのみ利用が可能です。
 スタンダードモードでは実行しないでください。
 本メソッドを使用する場合、事前に**SelectPageMode**を実行し、ページモードを選択してください。

 始点は、**SelectPageMode**の*direction*で設定するページモードの印字方向により変わります。
 印字方向と始点については、「3.2.3(6) **Direction**」を参照してください。

startX、*startY*、*width*、*height*に、**SelectPageModel**により設定するページモードの印字領域を超える指定をした場合、印字領域を超える部分は描画されません。

注意 本メソッドではプリンターの罫線コマンドを使用し、矩形描画を実現しています。
 本メソッドの終了時にはプリンターコマンド「罫線OFFの設定」及び「罫線のクリア」を送信し、プリンターの罫線バッファをクリアしますのでご注意ください。罫線コマンドについては、「MP-B30 シリーズ サーマルプリンタ技術説明書」を参照してください。

行単位で紙送りをします。

構文 `public void FeedLine(Int32 lines)`

パラメーター *lines*

紙送りする行数(行)
有効範囲は0～255です。

例外 **PrinterException**

本メソッド呼び出し時にエラーが発生した場合**PrinterException**をスローします。
詳細は、「3.2.4(1) **PrinterException**」を参照してください。

説明

本メソッドはスタンダードモード及びページモードで利用が可能です。

スタンダードモードでは、指定した行数の紙送りを行います。

プリンターの行バッファ内にデータがある場合には、一行印字した後に紙送りを行います。

ページモードでは、縦方向のデータ展開開始位置を指定した行数分、移動します。

一行の紙送り量(改行量)は、**SetLineSpacing**で設定が可能です。

ドット単位で紙送りをします。

構文 `public void FeedDotLine(Int32 dotLines)`

パラメーター *dotLines*

紙送りするドット数(ドット)
有効範囲は0～8192です。

例外 **PrinterException**

本メソッド呼び出し時にエラーが発生した場合**PrinterException**をスローします。
詳細は、「3.2.4(1) **PrinterException**」を参照してください。

説明

本メソッドはスタンダードモード及びページモードで利用が可能です。

スタンダードモードでは、指定したドット数の紙送りを行います。

プリンターの行バッファ内にデータがある場合には、一行印字した後に紙送りを行います。

ページモードでは、縦方向のデータ展開開始位置を指定したドット数分、移動します。

用紙カット位置までの紙送りを行います。

構文 `public void FeedCutPosition()`

例外 **PrinterException**

本メソッド呼び出し時にエラーが発生した場合**PrinterException**をスローします。
詳細は、「3.2.4(1) **PrinterException**」を参照してください。

説明 本メソッドはスタンダードモード及びページモードで利用が可能です。

スタンダードモードの場合、プリンターに印字データが残っていると、本メソッドは実行されません。印字を全て終わらせてから本メソッドを実行してください。

FeedMarkPosition マーク紙の頭出し

本APIはサポートしていません。
呼び出した場合は、**PrinterException**をスローし、**ERR_UNSUPPORTED_METHOD**を返します。

構文 `public void FeedMarkPosition(Int32 dotLines)`

SetLog ログ出力の設定

ログ出力を設定します。

構文 `public void SetLog(Int32 logLevel, LogFileSize logFileSize)`

パラメーター *logLevel*
 ログの出力レベル
 0を指定してください。0を指定した場合、エラーログを出力します。
 0以外は指定しないでください。

logFileSize
 ログファイルの最大サイズ
 利用可能な設定は、「3.2.3(24) LogFileSize」を参照してください。

例外 **PrinterException**
 本メソッド呼び出し時にエラーが発生した場合**PrinterException**をスローします。
 詳細は、「3.2.4(1) PrinterException」を参照してください。

説明 本メソッドはスタンダードモード及びページモードで利用が可能です。

ログファイルは、ライブラリを組み込んだWinCEアプリケーションが存在するフォルダ下に保存されます。

ログファイル名 : PrinterManager.log.x (xの範囲は、0～4です)

最初にログファイルはPrinterManager.log.0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PrinterManager.log.1に変更し、新たにPrinterManager.log.0を作成します。
 作成されるログファイル数は最大5つです。

本メソッドを実行しない場合、1MBのエラーログを出力します。ログファイルのファイル名や作成数は、前記と同様となります。

- Public Property

IsOpened

プリンターの利用状態取得

Openによるプリンターの利用開始状態を取得します。

構文 `public Boolean IsOpened{get;}`

初期値 `False`

戻り値 `True` : プリンターの利用が開始されている。
 `False` : プリンターの利用が開始されていない。

説明 本プロパティはスタンダードモード及びページモードで利用が可能です。
 本プロパティは読み取り専用のため、値を設定することはできません。

(2) PrinterStatus

- 概要

プリンタステータスを格納するクラスです。

下記の機能を提供します。

Public Methods

名前	説明
GetErrOffline	オフラインエラー状態の取得
GetErrHardware	ハードウェアエラー状態の取得
GetErrVoltage	電圧エラー状態の取得
GetErrHeadTemperature	ヘッド温度エラー状態の取得
GetErrOutOfPaper	紙無しエラー状態の取得
GetErrCoverOpen	ペーパーカバーオープンエラー状態の取得
GetErrBattery	バッテリーエラー状態の取得
GetStateFeedSwitch	フィードスイッチ状態の取得
GetStatePaperFeed	紙送り状態の取得
GetStateReturnWaiting	復帰待ち状態の取得
GetStateFlashMemoryRewriting	フラッシュメモリ書き換え状態の取得
GetStateBattery	バッテリー残量状態の取得

- Public Methods

GetErrOffline オフラインエラー状態の取得

オフラインエラー状態を取得します。

構文 `public Boolean GetErrOffline()`

戻り値 True : オフライン
 False : オンライン

GetErrHardware ハードウェアエラー状態の取得

ハードウェアエラー状態を取得します。

構文 `public Boolean GetErrHardware()`

戻り値 True : エラーあり
 False : エラーなし

GetErrVoltage

電圧エラー状態の取得

電圧エラー状態を取得します。

構文 `public Boolean GetErrVoltage()`

戻り値 `True` : エラーあり
 `False` : エラーなし

GetErrHeadTemperature

ヘッド温度エラー状態の取得

ヘッド温度エラー状態を取得します。

構文 `public Boolean GetErrHeadTemperature()`

戻り値 `True` : エラーあり
 `False` : エラーなし

GetErrOutOfPaper

紙無しエラー状態の取得

紙無しエラー状態を取得します。

構文 `public Boolean GetErrOutOfPaper()`

戻り値 `True` : エラーあり
 `False` : エラーなし

GetErrCoverOpen

ペーパーカバーオープンエラー状態の取得

ペーパーカバーオープンエラー状態を取得します。

構文 `public Boolean GetErrCoverOpen()`

戻り値 `True` : エラーあり
 `False` : エラーなし

GetErrBattery

バッテリーエラー状態の取得

バッテリーエラー状態を取得します。

構文 `public Boolean GetErrBattery()`

戻り値 `True` : エラーあり
 `False` : エラーなし

フィードスイッチ状態を取得します。

構文 public Boolean **GetStateFeedSwitch()**

戻り値 True : ON
False : OFF

紙送り状態を取得します。

構文 public Boolean **GetStatePaperFeed()**

戻り値 True : 紙送中
False : 停止

復帰待ち状態を取得します。

構文 public Boolean **GetStateReturnWaiting()**

戻り値 True : 復帰待ち状態である
False : 復帰待ち状態でない

フラッシュメモリ書き換え状態を取得します。

構文 public Boolean **GetStateFlashMemoryRewriting()**

戻り値 True : フラッシュメモリ書き換え中である
False : フラッシュメモリ書き換え中でない

バッテリーの残量状態を取得します。

構文 public Int32 **GetStateBattery()**

戻り値 0: バッテリーなし (要充電、または残量0%)
1: バッテリー残量Low (残量目安: 約10%)
3: バッテリー残量Middle (残量目安: 約40%)
7: バッテリー残量Full (残量目安: 約80%)

(3) PrinterDiscovery

- 概要

プリンターの探索機能を提供するクラスです。

下記の機能を提供します。

Public Methods

名前	説明
PrinterDiscovery	コンストラクタ
StartDiscoveryPrinter	プリンターの探索開始
CancelDiscoveryPrinter	プリンターの探索中断
GetFoundPrinter	発見されたプリンター情報リストの取得

- Public Methods

PrinterDiscovery	コンストラクタ
-------------------------	----------------

com.seikoinstruments.sdk.mobileprinter.PrinterDiscoveryクラスのコンストラクタです。

構文 public **PrinterDiscovery** ()

例外 **PrinterException**
 本メソッド呼び出し時にエラーが発生した場合**PrinterException**をスローします。
 詳細は、「3.2.4(1) **PrinterException**」を参照してください。

StartDiscoveryPrinter	プリンターの探索開始
------------------------------	-------------------

プリンターの探索を開始します。

構文 public void **StartDiscoveryPrinter**(DiscoveryHandler *handler*,
 PrinterInterface *prnIf*,
 Int32 *retry*,
 Int32 *timeout*)

パラメーター *handler*
 DiscoveryHandlerのインスタンス

prnIf
 通信インターフェイス定数
 利用可能な設定は、「3.2.3(1) **PrinterInterface**」を参照してください。
 PRN_IF_USBはサポートしていません。

retry

リトライ回数(回)

*prnlf*の設定により、動作が異なります。

- ・ **PRN_IF_TCP**の場合

*retry*で設定した回数だけローカルブロードキャストパケットを送信します。

有効範囲は1～5です。

1未満の値を指定した場合は1回に設定されます。

5を超える値を指定した場合は5回に設定されます。

- ・ **PRN_IF_BT**の場合

指定した値は無視され、探索は1回のみ行います。

timeout

探索1回のタイムアウト時間(秒:s)

*prnlf*の設定により、動作が異なります。

- ・ **PRN_IF_TCP**の場合

ローカルブロードキャストパケットの送信毎に、*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。

有効範囲は3～60です。

3未満の値を指定した場合は3sに設定されます。

60を超える値を指定した場合は60sに設定されます。

- ・ **PRN_IF_BT**の場合

指定した値は無視されます。

例外

PrinterException

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

詳細は、「3.2.4(1) **PrinterException**」を参照してください。

説明

CancelDiscoveryPrinterで中断が可能です。探索を中断、または完了すると、*handler*で指定したインスタンスの**DiscoveryFinished**を実行します。

探索結果は、**GetFoundPrinter**で取得してください。

CancelDiscoveryPrinter

プリンターの探索中断

StartDiscoveryPrinterで開始したプリンターの探索を中断します。

構文

public void **CancelDiscoveryPrinter**()

例外

PrinterException

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

詳細は、「3.2.4(1) **PrinterException**」を参照してください。

説明

探索を中断すると、**StartDiscoveryPrinter**の*handler*で指定したインスタンスの**DiscoveryFinished**を実行します。

中断時までの探索結果は**GetFoundPrinter**にて取得できます。

StartDiscoveryPrinterで発見された全てのプリンター情報を取得します。

構文 `public ArrayList<DiscoveredPrinter> GetFoundPrinter()`

戻り値 発見されたプリンター情報を**DiscoveredPrinter**クラスのArrayListで返します。
DiscoveredPrinterクラスのメソッドにより、プリンターモデル名、Bluetoothアドレス、IPアドレス、MACアドレス及びシリアルナンバーを取得できます。
DiscoveredPrinterクラスのメソッドは、「3.2.2(4) **DiscoveredPrinter**」を参照してください。

(4) DiscoveredPrinter

- 概要

StartDiscoveryPrinterで発見されたプリンター情報を格納するクラスです。発見されたプリンター情報からプリンターモデル名、Bluetoothアドレス、IPアドレス、MACアドレス及びシリアルナンバーを取得できます。下記の機能を提供します。

Public Methods

名前	説明
GetPrinterModel	プリンターモデル名の取得
GetBluetoothAddress	Bluetoothアドレスの取得
GetIpAddress	IPアドレスの取得
GetMacAddress	MACアドレスの取得
GetSerialNumber	シリアルナンバーの取得

- Public Methods

GetPrinterModel プリンターモデル名の取得

プリンターモデル名の文字列を取得します。

構文 `public String GetPrinterModel()`

戻り値 プリンターモデル名

GetBluetoothAddress Bluetoothアドレスの取得

Bluetoothアドレスの文字列を取得します。

構文 `public String GetBluetoothAddress()`

戻り値 Bluetoothアドレス
Bluetoothアドレスは、“00:11:22:AA:BB:CC”形式の文字列を取得します。

GetIpAddress IPアドレスの取得

IPアドレスの文字列を取得します。

構文 `public String GetIpAddress()`

戻り値 IPアドレス

GetMacAddress MACアドレスの取得

MACアドレスの文字列を取得します。

構文 `public String GetMacAddress()`

戻り値 MACアドレス

シリアルナンバーの文字列を取得します。

構文 `public String GetSerialNumber()`

戻り値 シリアルナンバー

3.2.3 列挙型

(1) PrinterInterface

Openの通信インターフェイス指定で利用する列挙型定数を示します。

名前	説明
PRN_IF_TCP	TCP/IP
PRN_IF_BT	Bluetooth
PRN_IF_USB	USB

(2) PrinterModel

Openのプリンターモデル指定で利用する列挙型定数を示します。

名前	説明
PRN_MODEL_MP_B30	MP-B30

(3) Dithering

RegisterLogo、SendDataFileのディザリングで利用する列挙型定数を示します。

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

(4) PrinterInformation

GetPrinterInformation、GetPrinterInformationNumber、GetPrinterInformationStringのプリンター情報指定で利用する列挙型定数を示します。取得できるプリンター情報の詳細は、「MP-B30シリーズ サーマルプリンタ技術説明書」を参照してください。

名前	説明（プリンター情報）
GET_NV_MEM_CAP	NVグラフィックスのメモリ容量の送信
GET_NV_MEM_REM_CAP	NVグラフィックスのメモリ残り容量の送信
GET_NV_MEM_KEYCODE_LIST	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
GET_REM_USER_MEM_CAP_DEFRAG	ユーザ領域の整理後の残り容量の送信
GET_REM_USER_MEM_CAP	ユーザ領域の残り容量の送信
GET_FUNC_SET_RESP	機能設定の送信
GET_PRN_ID_MODEL	プリンタIDの送信(機種ID)
GET_PRN_ID_TYPE	プリンタIDの送信(タイプID)
GET_PRN_ID_ROM_VER	プリンタIDの送信(ROMバージョンID)
GET_PRN_ID_FIRM_VER_MAIN	プリンタIDの送信 (ファームウェアバージョン(main))
GET_PRN_ID_MFR	プリンタIDの送信(メーカー名)
GET_PRN_ID_MODEL_NAME	プリンタIDの送信(モデル名)

名前	説明（プリンター情報）
GET_PRN_ID_FIRM_VER_BOOT	プリンタIDの送信 (ファームウェアバージョン(boot))
GET_PRN_ID_FIRM_CHECKSUM_BOOT	プリンタIDの送信 (ファームウェアチェックサム(boot))
GET_PRN_ID_FIRM_CHECKSUM_MAIN	プリンタIDの送信 (ファームウェアチェックサム(main))
GET_PRN_ID_FIRM_CHECKSUM	プリンタIDの送信 (ファームウェアチェックサム(main+boot))
GET_MAINT_NUM_FEED_LINE	メンテナンスカウンタの送信 (紙送り行数(100ドットライン単位))
GET_MAINT_NUM_HEAD_ACTIVE	メンテナンスカウンタの送信 (サーマルヘッド通電回数(100ドットライン単位))
GET_MAINT_DRIVE_TIME	メンテナンスカウンタの送信 (製品の駆動時間(分単位))
GET_MAINT_NUM_FEED_LINE_INTEGRATION	メンテナンスカウンタの送信 (紙送り行数(100ドットライン単位)(積算))
GET_MAINT_NUM_HEAD_ACTIVE_INTEGRATION	メンテナンスカウンタの送信 (サーマルヘッド通電回数(100ドットライン単位)(積算))
GET_MAINT_DRIVE_TIME_INTEGRATION	メンテナンスカウンタの送信 (製品の駆動時間(分単位)(積算))
GET_HFONT_24_CHECKSUM	半角フォントIDの送信 (24ドットフォント、チェックサム)
GET_HFONT_24_ID	半角フォントIDの送信 (24ドットフォント、ID)
GET_HFONT_24_INT_CHAR	半角フォントIDの送信 (24ドットフォント、登録済み国際文字)
GET_HFONT_16_CHECKSUM	半角フォントIDの送信 (16ドットフォント、チェックサム)
GET_HFONT_16_ID	半角フォントIDの送信 (16ドットフォント、ID)
GET_HFONT_16_INT_CHAR	半角フォントIDの送信 (16ドットフォント、登録済み国際文字)
GET_FFONT_LANG	全角フォントIDの送信(言語名)
GET_FFONT_STANDARD	全角フォントIDの送信(規格名)
GET_FFONT_COMPANY	全角フォントIDの送信(社名)
GET_FFONT_CHECKSUM	全角フォントIDの送信(チェックサム)
GET_FFONT_DATA_SIZE	全角フォントIDの送信(データサイズ)

(5) TransactionFunction

ControlTransactionの一括処理制御方法指定で利用する列挙型定数を示します。

名前	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

(6) Direction

SelectPageModeの印字方向指定で利用する列挙型定数を示します。

始点が“左上”または“右下”の場合、紙送り方向と垂直方向に印字データを展開します。

始点が“右上”または“左下”の場合、紙送り方向に印字データを展開します。

名前	説明(" " : 初期値)
DIRECTION_LEFT_TO_RIGHT	印字方向: 左→右、始点: 左上(図 3-2 A)
DIRECTION_BOTTOM_TO_TOP	印字方向: 下→上、始点: 左下(図 3-2 B)
DIRECTION_RIGHT_TO_LEFT	印字方向: 右→左、始点: 右下(図 3-2 C)
DIRECTION_TOP_TO_BOTTOM	印字方向: 上→下、始点: 右上(図 3-2 D)

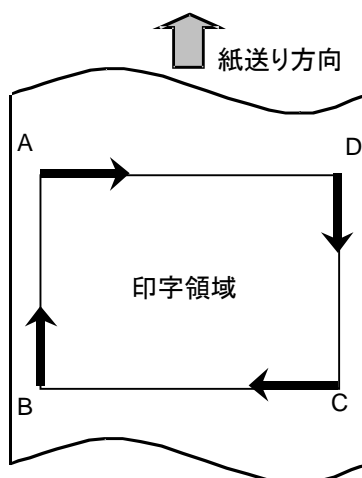


図 3-2

(7) Alignment

SetStandardModeAlignmentの位置揃え指定で利用する列挙型定数を示します。

名前	説明(" " : 初期値)
ALIGN_LEFT	左揃え
ALIGN_CENTER	中央揃え
ALIGN_RIGHT	右揃え

(8) CharacterSet

SelectCharacterSet、及びGetCharacterSetの文字セット選択で利用する列挙型定数を示します。

初期値は、WinCEデバイスの言語設定により下記の通りです。

日本語の場合: CODEPAGE_KATAKANA

日本語以外の場合: CODEPAGE_1252

名前	説明(" " : 初期値)
CODEPAGE_437	コードページ437(USA, Standard Europe)
CODEPAGE_KATAKANA	カタカナ文字セット
CODEPAGE_850	コードページ850(Multilingual)
CODEPAGE_860	コードページ860(Portuguese)
CODEPAGE_863	コードページ863(Canadian-French)
CODEPAGE_865	コードページ865(Nordic)
CODEPAGE_857	コードページ857(Turkish)
CODEPAGE_737	コードページ737(Greek)
CODEPAGE_1252	コードページ1252(Latin)
CODEPAGE_866	コードページ866(Russian)
CODEPAGE_852	コードページ852(Eastern Europe)
CODEPAGE_858	コードページ858(Euro)
CODEPAGE_855	コードページ855(Cyrillic)
CODEPAGE_864	コードページ864(Arabic)
CODEPAGE_1250	コードページ1250(Central European)
CODEPAGE_1251	コードページ1251(Cyrillic)
CODEPAGE_1253	コードページ1253(Greek)
CODEPAGE_1254	コードページ1254(Turkish)

(9) InternationalCharacterSet

SelectInternationalCharacterSet、及びGetInternationalCharacterの国際文字セット選択で利用する列挙型定数を示します。

初期値は、WinCEデバイスの言語設定により下記の通りです。

日本語の場合： INT_CHAR_SET_JAPAN

日本語以外の場合： INT_CHAR_SET_USA

名前	説明(" "：初期値)
INT_CHAR_SET_USA	アメリカ
INT_CHAR_SET_FRANCE	フランス
INT_CHAR_SET_GERMANY	ドイツ
INT_CHAR_SET_UNITED_KINGDOM	イギリス
INT_CHAR_SET_DENMARK_1	デンマーク I
INT_CHAR_SET_SWEDEN	スウェーデン
INT_CHAR_SET_ITALY	イタリア
INT_CHAR_SET_SPAIN_1	スペイン I
INT_CHAR_SET_JAPAN	日本
INT_CHAR_SET_NORWAY	ノルウェー
INT_CHAR_SET_DENMARK_2	デンマーク II
INT_CHAR_SET_SPAIN_2	スペイン II
INT_CHAR_SET_LATIN_AMERICA	ラテンアメリカ
INT_CHAR_SET_ARABIA	アラビア

(10) CharacterType

SetCharacterFormattingの文字フォント指定で利用する列挙型定数を示します。

名前	説明(" "：初期値)
CHAR_TYPE_FONT_CURRENT ^{*1}	現在の設定(設定するプリンターコマンドの送信なし)
CHAR_TYPE_FONT_A	フォントA(24×12)・漢字フォントA(24×24)
CHAR_TYPE_FONT_B ^{*1}	フォントB(16×8)・漢字フォントB(16×16)

*1: SelectCharacterSetでCODEPAGE_864を選択した場合、SetCharacterFormattingの文字フォント指定に関わらず、フォントA(24×12)で印字します。

(11) CharacterScale

SetCharacterFormattingの文字サイズ指定で利用する列挙型定数を示します。

名前	説明(" " : 初期値)
CHAR_SCALE_CURRENT	現在の設定(設定するプリンターコマンドの送信なし)
CHAR_SCALE_X1	1倍
CHAR_SCALE_X2	2倍
CHAR_SCALE_X3	3倍
CHAR_SCALE_X4	4倍
CHAR_SCALE_X5	5倍
CHAR_SCALE_X6	6倍
CHAR_SCALE_X7	7倍
CHAR_SCALE_X8	8倍

(12) Underline

SetCharacterFormattingのアンダーライン指定で利用する列挙型定数を示します。

名前	説明(" " : 初期値)
CHAR_UNDERLINE_CURRENT	現在の設定(設定するプリンターコマンドの送信なし)
CHAR_UNDERLINE_NONE	アンダーラインなし
CHAR_UNDERLINE_1DOT	1ドット幅のアンダーラインあり
CHAR_UNDERLINE_2DOT	2ドット幅のアンダーラインあり

(13) Bold

SetCharacterFormattingの強調印字指定で利用する列挙型定数を示します。

名前	説明(" " : 初期値)
CHAR_BOLD_CURRENT	現在の設定(設定するプリンターコマンドの送信なし)
CHAR_BOLD_OFF	強調印字なし
CHAR_BOLD_ON	強調印字あり

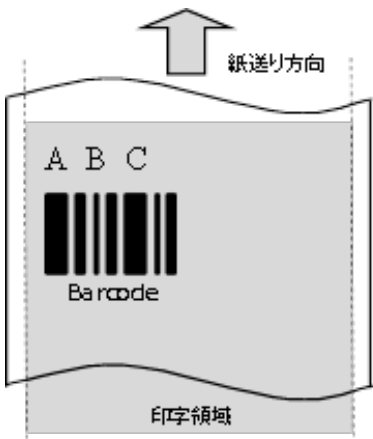
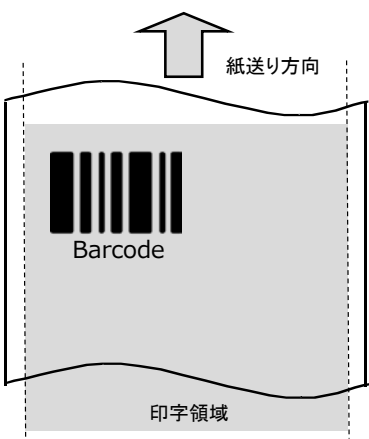
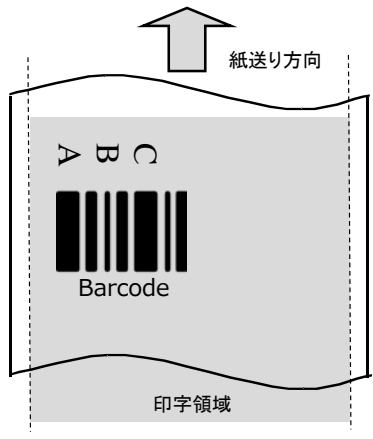
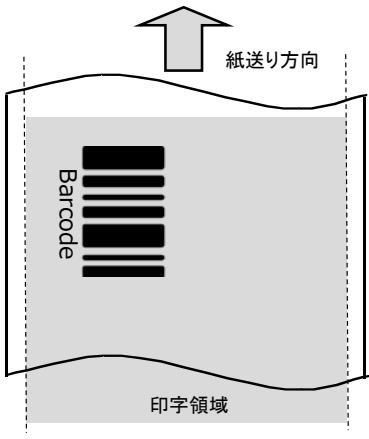
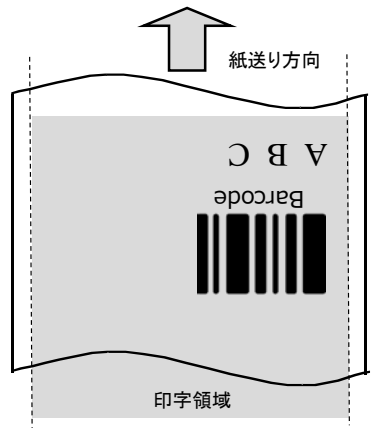
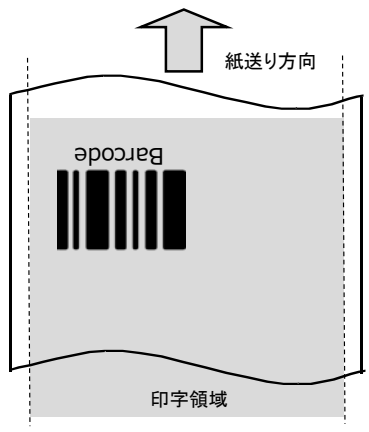
(14) Reverse

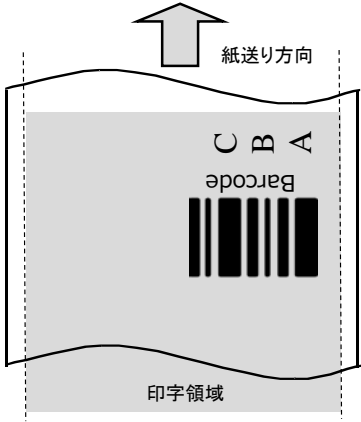
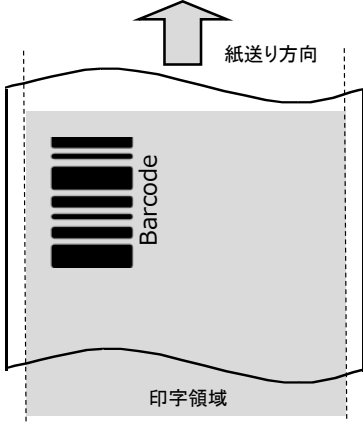
SetCharacterFormattingの白黒反転印字指定で利用する列挙型定数を示します。

名前	説明(" " : 初期値)
CHAR_REVERSE_CURRENT	現在の設定(設定するプリンターコマンドの送信なし)
CHAR_REVERSE_OFF	白黒反転印字なし
CHAR_REVERSE_ON	白黒反転印字あり

(15) Rotate

SetStandardModeBarcodeDirectionの印字方向指定、SetCharacterFormattingの回転印字指定で利用する列挙型定数を示します。文字、バーコード及び2次元バーコードの印字位置と向きを下記に示します。

名前	説明(" " : 初期値)	
	SetCharacterFormatting	SetStandardModeBarcodeDirection
ROTATE_CURRENT	現在の設定(設定するプリンターコマンドの送信なし)	
ROTATE_NONE	回転なし 	回転なし 
ROTATE_90_TO_RIGHT		
ROTATE_180		

名前	説明(" " : 初期値)	
	SetCharacterFormatting	SetStandardModeBarcodeDirection
ROTATE_90_TO_LEFT		

(16) TypeBarcode

PrintBarcodeのバーコードの種類指定で利用する列挙型定数を示します。

名前	説明	
TYPE_BARCODE_UPC_A	UPC-A	マルチレベルバーコード
TYPE_BARCODE_UPC_E	UPC-E	マルチレベルバーコード
TYPE_BARCODE_JAN13	JAN13(EAN13)	マルチレベルバーコード
TYPE_BARCODE_JAN8	JAN8(EAN8)	マルチレベルバーコード
TYPE_BARCODE_CODE39	CODE39	2値レベルバーコード
TYPE_BARCODE_ITF	ITF	2値レベルバーコード
TYPE_BARCODE_CODABAR	CODABAR	2値レベルバーコード
TYPE_BARCODE_CODE128	CODE128	マルチレベルバーコード
TYPE_BARCODE_CODE93	CODE93	マルチレベルバーコード
TYPE_BARCODE_JAN13_ADDON2	JAN13アドオン2	マルチレベルバーコード
TYPE_BARCODE_JAN13_ADDON5	JAN13アドオン5	マルチレベルバーコード
TYPE_BARCODE_GS1_OMNI_DIRECTIONAL	GS1 Databar Omni-directional	2値レベルバーコード
TYPE_BARCODE_GS1_TRUNCATED	GS1 Databar Truncated	2値レベルバーコード
TYPE_BARCODE_GS1_LIMITED	GS1 Databar Limited	2値レベルバーコード
TYPE_BARCODE_GS1_EXPANDED	GS1 Databar Expanded	2値レベルバーコード

(17) ModuleWidthBarcode

PrintBarcodeのバーコードのマルチレベルバーコードにおけるモジュール幅または2値レベルバーコードにおける細エレメント指定で利用する列挙型定数を示します。

名前	説明
MODULE_WIDTH_BARCODE_2	2ドット
MODULE_WIDTH_BARCODE_3	3ドット
MODULE_WIDTH_BARCODE_4	4ドット
MODULE_WIDTH_BARCODE_5	5ドット
MODULE_WIDTH_BARCODE_6	6ドット

(18) HriPositionBarcode

PrintBarcodeのバーコードのHRI文字指定で利用する列挙型定数を示します。

名前	説明
HRI_BARCODE_NONE	HRI文字なし
HRI_BARCODE_TOP_FONT_A	バーコードの上(フォントA)
HRI_BARCODE_BOTTOM_FONT_A	バーコードの下(フォントA)
HRI_BARCODE_FONT_A	バーコードの上下(フォントA)
HRI_BARCODE_TOP_FONT_B	バーコードの上(フォントB)
HRI_BARCODE_BOTTOM_FONT_B	バーコードの下(フォントB)
HRI_BARCODE_FONT_B	バーコードの上下(フォントB)

(19) NwRatioBarcode

PrintBarcodeの2値レベルバーコードのN:W比指定で利用する列挙型定数を示します。

名前	説明
NWRATIO_BARCODE_1TO2	1:2
NWRATIO_BARCODE_1TO2_5	1:2.5
NWRATIO_BARCODE_1TO3	1:3

(20) Type2Dcode

Print2Dcodeの2次元バーコードの種類指定で利用する列挙型定数を示します。

名前	説明
TYPE_2DCODE_QR_CODE	QRコード
TYPE_2DCODE_PDF417	PDF417
TYPE_2DCODE_DATA_MATRIX	Data Matrix
TYPE_2DCODE_MAXI_CODE	MaxiCode
TYPE_2DCODE_GS1_STACKED	GS1 Databar Stacked
TYPE_2DCODE_GS1_OMNI_DIRECTIONAL	GS1 Databar Stacked Omni-directional
TYPE_2DCODE_GS1_EXPANDED_STACKED	GS1 Databar Expanded Stacked

(21) Mode2Dcode

Print2Dcodeの2次元バーコードのモードで利用する列挙型定数を示します。

名前	説明	
MODE_2DCODE_QR_CODE_MODEL1	QRコード	Model1
MODE_2DCODE_QR_CODE_MODEL2	QRコード	Model2
MODE_2DCODE_PDF417_STANDARD	PDF417	通常モード
MODE_2DCODE_PDF417_COMPACT	PDF417	簡易モード
MODE_2DCODE_DATA_MATRIX_AUTO	Data Matrix	モジュール数自動
MODE_2DCODE_DATA_MATRIX_10_10	Data Matrix	モジュール数 10×10
MODE_2DCODE_DATA_MATRIX_12_12	Data Matrix	モジュール数 12×12
MODE_2DCODE_DATA_MATRIX_14_14	Data Matrix	モジュール数 14×14
MODE_2DCODE_DATA_MATRIX_16_16	Data Matrix	モジュール数 16×16
MODE_2DCODE_DATA_MATRIX_18_18	Data Matrix	モジュール数 18×18
MODE_2DCODE_DATA_MATRIX_20_20	Data Matrix	モジュール数 20×20
MODE_2DCODE_DATA_MATRIX_22_22	Data Matrix	モジュール数 22×22
MODE_2DCODE_DATA_MATRIX_24_24	Data Matrix	モジュール数 24×24
MODE_2DCODE_DATA_MATRIX_26_26	Data Matrix	モジュール数 26×26
MODE_2DCODE_DATA_MATRIX_32_32	Data Matrix	モジュール数 32×32
MODE_2DCODE_DATA_MATRIX_36_36	Data Matrix	モジュール数 36×36
MODE_2DCODE_DATA_MATRIX_40_40	Data Matrix	モジュール数 40×40
MODE_2DCODE_DATA_MATRIX_44_44	Data Matrix	モジュール数 44×44
MODE_2DCODE_DATA_MATRIX_48_48	Data Matrix	モジュール数 48×48
MODE_2DCODE_DATA_MATRIX_52_52	Data Matrix	モジュール数 52×52
MODE_2DCODE_DATA_MATRIX_64_64	Data Matrix	モジュール数 64×64
MODE_2DCODE_DATA_MATRIX_72_72	Data Matrix	モジュール数 72×72
MODE_2DCODE_DATA_MATRIX_80_80	Data Matrix	モジュール数 80×80
MODE_2DCODE_DATA_MATRIX_88_88	Data Matrix	モジュール数 88×88
MODE_2DCODE_DATA_MATRIX_96_96	Data Matrix	モジュール数 96×96
MODE_2DCODE_DATA_MATRIX_104_104	Data Matrix	モジュール数 104×104
MODE_2DCODE_DATA_MATRIX_120_120	Data Matrix	モジュール数 120×120
MODE_2DCODE_DATA_MATRIX_132_132	Data Matrix	モジュール数 132×132
MODE_2DCODE_DATA_MATRIX_144_144	Data Matrix	モジュール数 144×144
MODE_2DCODE_DATA_MATRIX_8_18	Data Matrix	モジュール数 8×18
MODE_2DCODE_DATA_MATRIX_8_32	Data Matrix	モジュール数 8×32
MODE_2DCODE_DATA_MATRIX_12_26	Data Matrix	モジュール数 12×26
MODE_2DCODE_DATA_MATRIX_12_36	Data Matrix	モジュール数 12×36
MODE_2DCODE_DATA_MATRIX_16_36	Data Matrix	モジュール数 16×36
MODE_2DCODE_DATA_MATRIX_16_48	Data Matrix	モジュール数 16×48
MODE_2DCODE_MAXI_CODE_2	MaxiCode	Mode2
MODE_2DCODE_MAXI_CODE_3	MaxiCode	Mode3
MODE_2DCODE_MAXI_CODE_4	MaxiCode	Mode4
MODE_2DCODE_MAXI_CODE_5	MaxiCode	Mode5

名前	説明	
MODE_2DCODE_NONE	GS1 Databar Stacked GS1 Databar Stacked Omni-directional GS1 Databar Expanded Stacked	設定なし

(22) **ModuleSize2Dcode**

Print2Dcodeの2次元バーコードのモジュールサイズ指定で利用する列挙型定数を示します。

名前	説明	
MODULE_SIZE_2DCODE_DEFAULT	QRコード	初期値 (6ドット)
	PDF417	初期値 (4ドット)
	Data Matrix	初期値 (6ドット)
	MaxiCode	設定なし
	GS1 Databar Stacked	初期値 (6ドット)
	GS1 Databar Stacked Omni-directional	初期値 (6ドット)
	GS1 Databar Expanded Stacked	初期値 (6ドット)
MODULE_SIZE_2DCODE_2	QRコード	2ドット
	PDF417	
	Data Matrix	
	GS1 Databar Stacked	
	GS1 Databar Stacked Omni-directional	
	GS1 Databar Expanded Stacked	
MODULE_SIZE_2DCODE_3	QRコード	3ドット
	PDF417	
	Data Matrix	
	GS1 Databar Stacked	
	GS1 Databar Stacked Omni-directional	
	GS1 Databar Expanded Stacked	
MODULE_SIZE_2DCODE_4	QRコード	4ドット
	Data Matrix	
	GS1 Databar Stacked	
	GS1 Databar Stacked Omni-directional	
	GS1 Databar Expanded Stacked	
MODULE_SIZE_2DCODE_5	QRコード	5ドット
	Data Matrix	
	GS1 Databar Stacked	
	GS1 Databar Stacked Omni-directional	
	GS1 Databar Expanded Stacked	

名前	説明	
MODULE_SIZE_2DCODE_6	QRコード	6ドット
	Data Matrix	
	GS1 Databar Stacked	
	GS1 Databar Stacked Omni-directional	
	GS1 Databar Expanded Stacked	
MODULE_SIZE_2DCODE_7	QRコード	7ドット
	Data Matrix	
	GS1 Databar Stacked	
	GS1 Databar Stacked Omni-directional	
	GS1 Databar Expanded Stacked	
MODULE_SIZE_2DCODE_8	QRコード	8ドット
	Data Matrix	
	GS1 Databar Stacked	
	GS1 Databar Stacked Omni-directional	
	GS1 Databar Expanded Stacked	
MODULE_SIZE_2DCODE_9	QRコード	9ドット
	Data Matrix	
	GS1 Databar Stacked	
	GS1 Databar Stacked Omni-directional	
	GS1 Databar Expanded Stacked	
MODULE_SIZE_2DCODE_10	QRコード	10ドット
	Data Matrix	
	GS1 Databar Stacked	
	GS1 Databar Stacked Omni-directional	
	GS1 Databar Expanded Stacked	
MODULE_SIZE_2DCODE_11	QRコード	11ドット
	Data Matrix	
	GS1 Databar Stacked	
	GS1 Databar Stacked Omni-directional	
	GS1 Databar Expanded Stacked	
MODULE_SIZE_2DCODE_12	QRコード	12ドット
	Data Matrix	
	GS1 Databar Stacked	
	GS1 Databar Stacked Omni-directional	
	GS1 Databar Expanded Stacked	
MODULE_SIZE_2DCODE_13	QRコード	13ドット
	Data Matrix	
	GS1 Databar Stacked	
	GS1 Databar Stacked Omni-directional	
	GS1 Databar Expanded Stacked	

名前	説明	
MODULE_SIZE_2DCODE_14	QRコード	14ドット
	Data Matrix	
	GS1 Databar Stacked	
	GS1 Databar Stacked Omni-directional	
	GS1 Databar Expanded Stacked	
MODULE_SIZE_2DCODE_15	QRコード	15ドット
	Data Matrix	
	GS1 Databar Stacked	
	GS1 Databar Stacked Omni-directional	
	GS1 Databar Expanded Stacked	
MODULE_SIZE_2DCODE_16	QRコード	16ドット
	Data Matrix	
	GS1 Databar Stacked	
	GS1 Databar Stacked Omni-directional	
	GS1 Databar Expanded Stacked	

(23) ErrorCorrect2Dcode

Print2Dcodeの2次元バーコードのエラー訂正レベル指定で利用する列挙型定数を示します。

名前	説明	
ERR_CORRECTION_2DCODE_QR_CODE_L	QRコード	L
ERR_CORRECTION_2DCODE_QR_CODE_M	QRコード	M
ERR_CORRECTION_2DCODE_QR_CODE_Q	QRコード	Q
ERR_CORRECTION_2DCODE_QR_CODE_H	QRコード	H
ERR_CORRECTION_2DCODE_PDF417_0	PDF417	0
ERR_CORRECTION_2DCODE_PDF417_1	PDF417	1
ERR_CORRECTION_2DCODE_PDF417_2	PDF417	2
ERR_CORRECTION_2DCODE_PDF417_3	PDF417	3
ERR_CORRECTION_2DCODE_PDF417_4	PDF417	4
ERR_CORRECTION_2DCODE_PDF417_5	PDF417	5
ERR_CORRECTION_2DCODE_PDF417_6	PDF417	6
ERR_CORRECTION_2DCODE_PDF417_7	PDF417	7
ERR_CORRECTION_2DCODE_PDF417_8	PDF417	8
ERR_CORRECTION_2DCODE_NONE	Data Matrix MaxiCode GS1 Databar Stacked GS1 Databar Stacked Omni-directional GS1 Databar Expanded Stacked	設定なし

(24) LogFileSize

SetLogのログファイルの最大サイズ指定で利用する列挙型定数を示します。

名前	説明(" " : 初期値)
LOG_FILE_SIZE_1MB	1 MB (1048576バイト)
LOG_FILE_SIZE_5MB	5 MB (5242880バイト)
LOG_FILE_SIZE_10MB	10 MB (10485760バイト)
LOG_FILE_SIZE_50MB	50 MB (52428800バイト)

(25) ErrorCode

GetErrorCodeで取得するエラー内容を示す列挙型定数を示します。

名前	説明
ERR_PARAM	パラメーターが不正です。
ERR_OPENED	指定したプリンターはすでにオープン済みです。
ERR_NOT_OPENED	指定したプリンターはオープンされていません。
ERR_TIMEOUT	タイムアウト、またはビジー状態です。
ERR_OFFLINE	切断状態またはプリンターがオフラインです。
ERR_CLOSE_FAIL	プリンターの切断に失敗しました。
ERR_NOT_MONITORING	接続状態が監視されていません。
ERR_INIT_FAILED	初期化に失敗しました。
ERR_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。
ERR_OVER_MAX_DATA_SIZE	データサイズが最大値を超過しています。
ERR_INVALID_DATA	不正なデータが指定されています。
ERR_INVALID_STATE	コールバックにPrinterManagerオブジェクトへのアクセスが指定されました。
ERR_ACCESS	プリンターにアクセスできません。
ERR_CANCELED	機能が取り消されました。
ERR_WRITE_FAULT	プリンターにデータを送信できません。
ERR_WORKAREA_NO_MEMORY	メモリの領域指定が不足しています。
ERR_FILE_INVALID	指定されたファイルは無効です。
ERR_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。
ERR_NOT_FOUND	指定したファイルが見つかりません。
ERR_TRANSACTION_STOPPED	一括処理が開始されていません。
ERR_PRINTER_STATUS_ERROR	プリンターステータスが正常ではありません。
ERR_UNSUPPORTED_METHOD	サポートしていないメソッドです。

3.2.4 例外

(1) PrinterException

- 概要

PrinterExceptionクラスは、例外状況が発生したことを示すクラスです。
下記の機能を提供します。

Public Methods

名前	説明
GetErrorCode	エラーコードの取得

- Public Methods

GetErrorCode	エラーコードの取得
---------------------	------------------

例外のエラーコードを取得します。

構文 `public ErrorCode GetErrorCode()`

戻り値 エラーコード

説明 エラー内容詳細は、「3.2.3(25) **ErrorCode**」を参照してください。

4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。

SDKでは、サンプルプログラムとしてCABファイル及びMicrosoft Visual Studio C# 2008のプロジェクト形式のプログラムを提供します。

WinCEアプリケーションの開発環境が整っていることを確認してください。必要な開発環境については、「2章 ライブラリの利用方法」を参照してください。

4.1 利用方法

サンプルプログラムの利用方法について説明します。

4.1.1 CABファイル

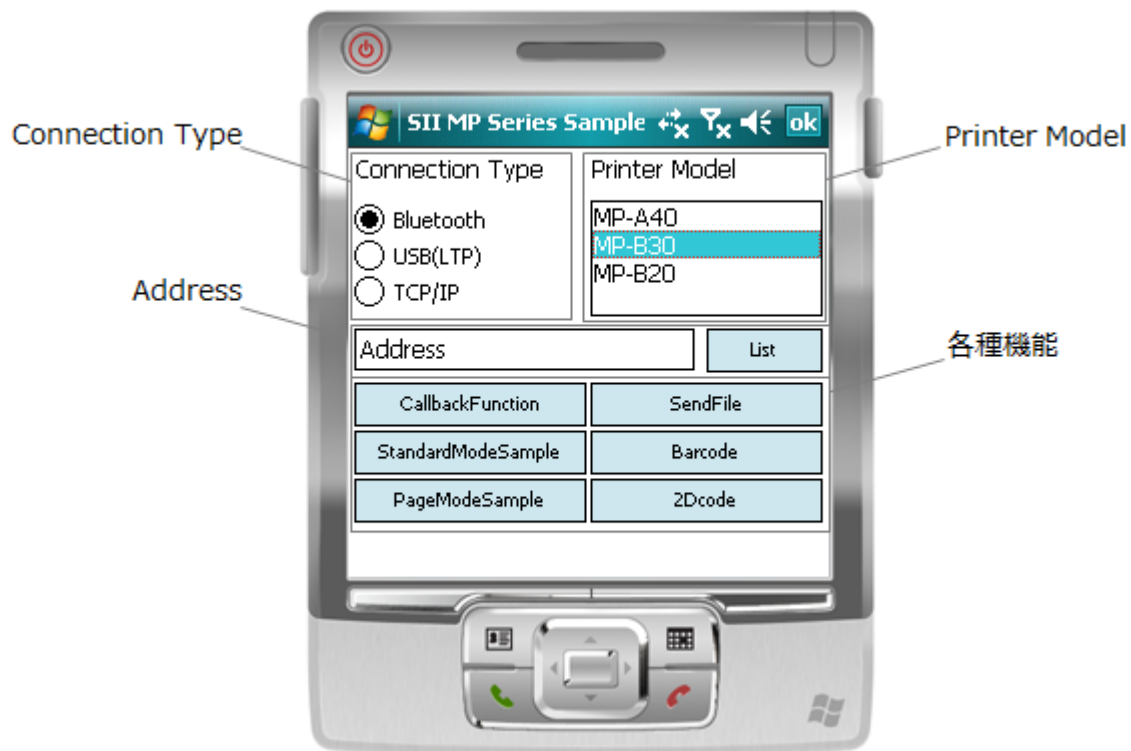
- (1) [システムドライブ]:¥WindowsCE¥Sample¥cabフォルダ下にあるCABファイルをWinCEデバイス上の適当なフォルダにコピーしてください。
- (2) ファイルエクスプローラからCABファイルを実行してください。
- (3) WinCEデバイスの[プログラム]からサンプルプログラムを実行してください。

4.1.2 プロジェクト形式プログラム

- (1) [システムドライブ]:¥WindowsCE¥Sample¥sourceフォルダをローカルコンピュータにコピーしてください。
- (2) ファイルエクスプローラからコピーしたsourceフォルダ下にあるSLNファイルを実行してください。

4.2 画面構成

「4.1.1 CABファイル」でインストールしたサンプルプログラムの画面構成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。
Printer Model	プリンターのモデルを選択します。
Address	プリンターのアドレスを指定します。 手動入力の場合： Bluetoothで接続する場合は、Bluetoothアドレス、TCP/IPで接続する場合は、MACアドレスを入力してください。 自動入力の場合： [List]をタップすると、プリンターの探索が開始されます。 表示された一覧の中からプリンターを選択すると、選択したプリンターのアドレスが自動入力されます。
各種機能	各種機能を実行できるボタンです。 スクロールすると、表示されないメソッド及びプロパティを見ることができます。メソッド及びプロパティの詳細については、「3章 ライブラリの機能」を参照してください。

4.3 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（文字コード表）

A.1 文字コード表（文字コードテーブル）

国際文字設定で INT_CHAR_SET_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。

	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	Ç	ü	é	â	ä	à	å	ç	ê	ë	è	ï	î	ì	Ä	Å
90	É	æ	Æ	ô	ö	ò	û	ù	ÿ	Ö	Ü	φ	£	¥	℔	ƒ
A0	á	í	ó	ú	ñ	Ñ	ä	ö	¿	¬	½	¼	¿	»	»	»
B0	☐	☐	☐													
C0	L	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥
D0	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥
E0	α	β	Γ	π	Σ	σ	μ	τ	φ	θ	Ω	δ	∞	φ	ε	Π
F0	≡	±	≥	≤	∫	∫	÷	≈	°	•	•	√	n	2	■	■

表 A-1 CODEPAGE_437 (USA, Standard Europe)

	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																
90																
A0	。	「	」	、	・	ヲ	ア	イ	ウ	エ	オ	ヤ	ユ	ヨ	ッ	
B0	ー	ア	イ	ウ	エ	オ	カ	キ	ク	ケ	コ	サ	シ	ス	セ	ソ
C0	タ	チ	ツ	テ	ト	ナ	ニ	ヌ	ネ	ノ	ハ	ヒ	フ	ヘ	ホ	マ
D0	ミ	ム	メ	モ	ヤ	ユ	ヨ	ラ	リ	ル	レ	ロ	ワ	ン	ゝ	。
E0																
F0																

表 A-2 CODEPAGE_KATAKANA

	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	Ç	ü	é	â	ä	à	å	ç	ê	ë	è	ï	î	ì	Ä	Å
90	É	æ	Æ	ô	ö	ò	û	ü	ÿ	Ö	Ü	ø	£	Ø	×	f
A0	á	í	ó	ú	ñ	Ñ	ä	ö	¿	®	¬	½	¼	¡	«	»
B0	☐	☐	☐			Á	Â	À	©	¶	¶	¶	¶	¢	¥	₱
C0	⌞	⌞	⌞	⌞	⌞	⌞	⌞	⌞	⌞	⌞	⌞	⌞	⌞	⌞	⌞	⌞
D0	ð	Đ	Ê	Ë	È	Í	Î	Ï	⌞	⌞	⌞	⌞	⌞	⌞	⌞	⌞
E0	Ó	ß	Ô	Ò	Õ	µ	þ	þ	Ú	Û	Ü	ý	Ý	-	'	
F0	-	±	=	¾	¶	§	÷	,	°	…	.	¹	³	²	■	

表 A-3 CODEPAGE_850 (Multilingual)

	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	Ç	ü	é	â	ã	à	Á	ç	ê	Ê	è	Í	ô	ì	Ã	Â
90	É	À	È	ô	õ	ò	Ú	ù	Ì	Õ	Ü	¢	£	Ù	Þ	Ó
A0	á	í	ó	ú	ñ	Ñ	ä	ö	ï	ò	¬	½	¼	¡	«	»
B0	☐	☐	☐													
C0	L	L	T	T	T	T	T	T	T	T	T	T	T	T	T	T
D0	⌌	⌌	⌌	⌌	⌌	⌌	⌌	⌌	⌌	⌌	⌌	⌌	⌌	⌌	⌌	⌌
E0	α	β	Γ	π	Σ	σ	μ	τ	φ	θ	Ω	δ	∞	φ	ε	Π
F0	≡	±	≥	≤		J	÷	≈	°	•	•	√	n	2	■	

表 A-4 CODEPAGE_860 (Portuguese)

	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	Ç	ü	é	â	Â	à	¶	ç	ê	ë	è	ï	î	≡	À	§
90	É	È	Ê	ô	Ë	Ï	Ô	ù	æ	ô	Ü	¢	£	Ù	û	f
A0		'	ó	ú	..	³	-	î	¬	¬	½	¼	¾	«	»	
B0	☐	☐	☐													
C0	L	L	T	T	T	T	T	T	T	T	T	T	T	T	T	T
D0	⌌	⌌	⌌	⌌	⌌	⌌	⌌	⌌	⌌	⌌	⌌	⌌	⌌	⌌	⌌	⌌
E0	α	β	Γ	π	Σ	σ	μ	τ	φ	θ	Ω	δ	∞	φ	ε	Π
F0	≡	±	≥	≤		J	÷	≈	°	•	•	√	n	2	■	

表 A-5 CODEPAGE_863 (Canadian-French)

	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	Ç	ü	é	â	ä	à	å	ç	ê	ë	è	ï	î	ì	Ä	Å
90	É	æ	Æ	ô	ö	ò	û	ù	ÿ	Ö	Ü	ø	£	Ø	Pt	f
A0	á	í	ó	ú	ñ	Ñ	ä	ö	¿	¬	½	¼	¡	«	»	
B0	☐	☐	☐													
C0	L	L	T	T	T	T	T	T	T	T	T	T	T	T	T	T
D0	L	T	T	L	L	F	π	π	π	π	π	π	π	π	π	π
E0	α	β	Γ	π	Σ	σ	μ	τ	φ	θ	Ω	δ	∞	φ	ε	Π
F0	≡	±	≥	≤		J	÷	≈	°	•	•	√	n	2	■	

表 A-6 CODEPAGE_865 (Nordic)

	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	Ç	ü	é	â	ä	à	å	ç	ê	ë	è	ï	î	ì	Ä	Å
90	É	æ	Æ	ô	ö	ò	û	ù	ÿ	Ö	Ü	ø	£	Ø	Ş	ş
A0	á	í	ó	ú	ñ	Ñ	Ğ	ğ	¿	®	¬	½	¼	¡	«	»
B0	☐	☐	☐			Á	Â	À	©							
C0	L	L	T	T	T	ã	Ã	L	π	π	π	π	π	π	π	π
D0	o	a	Ê	Ë	È	€	Í	Î	Ï	J	Γ	■	■		ì	■
E0	ó	β	ô	ò	õ	Ö	μ	×	Ú	Û	Ü	ì	ÿ	-	'	
F0	-	±	¾	¶	§	÷	,	°	..	•	1	3	2	■		

表 A-7 CODEPAGE_857 (Turkish)

	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	Υ
F0	Ω	±	≥	≤	İ	ÿ	÷	≈	°	•	•	√	n	²	■	

表 A-8 CODEPAGE_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	„	…	†	‡	^	‰	Š	‹	Œ		Ž		
90	‘	’	“	”	•	-	-	~	™	š	›	œ		ž	ÿ	
A0	ı	ϕ	£	¤	¥	¦	§	¨	©	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

表 A-9 CODEPAGE_1252 (Latin)

	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	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
90	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я
A0	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
B0	␣	␣	␣													
C0	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣
D0	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣
E0	р	с	т	у	ф	х	ц	ч	ш	щ	ъ	ы	ь	э	ю	я
F0	Ё	ё	Є	є	İ	ı	Ÿ	ÿ	°	•	•	√	№	α	■	

表 A-10 CODEPAGE_866 (Russian)

	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	Ç	ü	é	â	ä	û	ç	ı	ë	Ö	ö	î	ž	Ä	Ć	
90	É	Í	í	ô	ö	Ĺ	ĺ	Š	š	Ö	Ü	ř	ť	Ł	×	č
A0	á	í	ó	ú	À	à	Ž	ž	Ę	ę	¬	ž	Č	š	«	»
B0	␣	␣	␣			Á	Â	Ě	Š					Ž	ž	ı
C0	␣	␣	␣	␣	␣	Ä	ä	Ľ	ľ	␣	␣	␣	␣	␣	␣	␣
D0	ď	Đ	Ď	Ě	ď	Ň	í	î	ě	ı	ı	ı	ı	ı	ı	ı
E0	ó	ß	ô	ń	ň	š	š	ř	ú	ř	ú	ý	ý	ı	ı	ı
F0	-	"	˘	˘	˘	Š	÷	˘	˘	˘	˘	Ů	Ř	ř	■	

表 A-11 CODEPAGE_852 (Eastern Europe)

	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	Ç	ü	é	â	ä	à	å	ç	ê	ë	è	ï	î	ì	Ä	Å
90	É	æ	Æ	ô	ö	ò	û	ü	Ö	Ü	ø	£	Ø	×	ƒ	
A0	á	í	ó	ú	ñ	Ñ	á	ó	í	®	¬	½	¼	¡	«	»
B0	☐	☐	☐			Á	Â	À	©	¶	¶	¶	¶	¢	¥	₱
C0	L	L	T	T	—	+	ã	Ã	ℓ	ℓ	ℓ	ℓ	ℓ	ℓ	ℓ	α
D0	ð	Ð	Ê	Ë	È	€	Í	Î	Ï	Ј	Г	■	■	І	İ	■
E0	ó	β	ô	ò	õ	õ	μ	þ	þ	ú	û	ü	ý	ý	-	'
F0	-	±	=	¾	¶	§	÷	,	°	…	.	¹	³	²	■	

表 A-12 CODEPAGE_858 (Euro)

	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	ђ	Ђ	ѓ	Ѓ	ё	Ё	є	Є	ѕ	Ѕ	і	І	ї	Ї	ј	Ј
90	љ	Љ	њ	Њ	ћ	Ћ	ќ	Ќ	џ	Џ	џ	џ	џ	џ	џ	џ
A0	а	А	б	Б	в	В	г	Г	д	Д	е	Е	ф	Ф	г	Г
B0	☐	☐	☐			x	X	и	И	¶	¶	¶	¶	¶	¶	¶
C0	L	L	T	T	—	+	к	К	ℓ	ℓ	ℓ	ℓ	ℓ	ℓ	ℓ	α
D0	л	Л	м	М	н	Н	о	О	п	П	Г	■	■	П	Я	■
E0	Я	р	Р	с	С	т	Т	у	У	ж	Ж	в	В	ь	Ь	№
F0	-	ы	Ы	э	Э	ш	Ш	э	Э	щ	Щ	ч	Ч	§	■	

表 A-13 CODEPAGE_855 (Cyrillic)

	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	°	•	√	☒	-		+	+	+	+	+	+	+	+	+	+
90	β	∞	φ	±	½	¼	≈	«	»	لأ	لأ					
A0	-	ĩ	£	¤	£	€	ل	ب	ث	ج	ح	خ	خ	خ	خ	خ
B0	•	١	٢	٣	٤	٥	٦	٧	٨	٩	ف	س	ش	ص	ض	ظ
C0	¢	ء	آ	أ	ؤ	ع	ئ	ب	ا	ة	ث	ج	ح	خ	د	ذ
D0	ذ	ر	ز	س	ش	ص	ض	ط	ظ	ع	ا	÷	×	ع	ع	ع
E0	-	ف	ق	ك	ل	م	ن	ه	و	ي	ض	ي	غ	غ	غ	غ
F0	-	"	ن	ه	ي	ي	ي	ق	ل	ل	ل	ل	ل	ل	ل	ل

表 A-14 CODEPAGE_864 (Arabic)

	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	€	;	„	„	...	†	†	‰	š	<	š	ť	ž	ž	ž	ž
90	„	;	„	„	•	-	-	™	š	>	š	ť	ž	ž	ž	ž
A0	˘	˘	ł	¤	Å	ı	§	•	©	§	«	¬	-	®	ž	ž
B0	°	±	ı	ı	μ	¶	•	q	§	»	Ł	”	ł	ž	ž	ž
C0	Ř	Á	Â	Ä	Ä	Ł	Ć	Ç	Č	É	Ę	Ě	Ě	Í	Î	Ď
D0	Đ	Ň	Ň	Ó	Ô	Ö	×	Ř	Ů	Ú	Ú	Ü	Ý	Ť	ß	ß
E0	ř	á	â	ä	ä	í	ć	ç	č	é	ę	ě	ě	í	î	ď
F0	đ	ň	ň	ó	ô	ö	÷	ř	ů	ú	ú	ü	ý	ť	·	·

表 A-15 CODEPAGE_1250 (Central European)

	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	ъ	ѓ	;	ѓ	”	...	†	‡	€	‰	Љ	<	Њ	ќ	ћ	џ
90	ђ	‘	;	“	”	•	-	-	™	љ	>	њ	ќ	ћ	џ	
A0	ѣ	ѝ	Ј	Ѡ	Г	І	Ѕ	Ё	Є	«	¬	-	®	İ		
B0	°	±	І	і	г	μ	¶	•	ё	№	є	»	ј	ѕ	ѕ	ї
C0	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
D0	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я
E0	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
F0	р	с	т	у	ф	х	ц	ч	ш	щ	ъ	ы	ь	э	ю	я

表 A-16 CODEPAGE_1251 (Cyrillic)

	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	”	...	†	‡	‰	<						
90	‚	‘	;	“	”	•	-	-	™	>						
A0	“	À	£	¤	¥	¦	§	¨	©	ª	«	¬	-	®	-	
B0	°	±	²	³	´	μ	¶	·	¸	¹	º	»	¼	½	¾	¸
C0	ı	Α	Β	Γ	Δ	Ε	Ζ	Η	Θ	Ι	Κ	Λ	Μ	Ν	Ξ	Ο
D0	Π	Ρ	Σ	Τ	Υ	Φ	Χ	Ψ	Ω	İ	ÿ	ά	έ	ή	ί	
E0	ΐ	α	β	γ	δ	ε	ζ	η	θ	ι	κ	λ	μ	ν	ξ	ο
F0	π	ρ	ς	σ	τ	υ	φ	χ	ψ	ω	ï	ÿ	ό	ύ	ώ	

表 A-17 CODEPAGE_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	€	‘	’	“	”	…	†	‡	^	%	Š	<	£			
90		•	-	-	~	™	š	>	œ						ÿ	
A0	ı	¢	£	¤	¥	¦	§	¨	©	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

表 A-18 CODEPAGE_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結果が異なります。
特定の文字コードおよびその印字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	#	α	É	Ä	Ö	Å	Ü	é	ä	ö	å	ü
COUNTRY_ITALY	#	\$	@	°	\	é	^	ù	à	ò	è	ì
COUNTRY_SPAIN	Pl	\$	@	ì	Ñ	¿	^	`	..	ñ	}	~
COUNTRY_JAPAN	#	\$	@	[¥]	^	`	{		}	~
COUNTRY_NORWAY	#	α	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	ì	Ñ	¿	é	`	í	ñ	ó	ú
COUNTRY_LATIN_AMERICA	#	\$	á	ì	Ñ	¿	é	ü	í	ñ	ó	ú
COUNTRY_ARABIA	#	\$	@	[\]	^	`	{		}	~

表 A-19 国際文字セット



セイコーインスツル株式会社

プリントシステム事業部

千葉県千葉市美浜区中瀬 1-8 〒261-8507

電話番号：043-211-1212（直通） ファクシミリ：043-211-8037

ホームページ：<https://www.sii-ps.com>

大阪営業所

大阪府大阪市北区豊崎 3-2-1 淀川 5 番館 〒531-0072

電話番号：06-7711-0858（直通） ファクシミリ：06-7711-0856

◎本書の内容は、製品の改良に伴い、予告なしに変更することがあります。
