MP-A40シリーズWindows向けSDK
アプリケーションプログラマーズガイド

U00135941505

セイコーインスツル株式会社

U00135941500　2016年　8月
U00135941501　2018年　1月
U00135941502　2019年　2月
U00135941503　2021年　9月
U00135941504　2022年　10月
U00135941505　2024年　11月

©セイコーインスツル株式会社　2016-2024
無断転載を禁じます。
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。
Bluetooth®はBluetooth SIG, Inc.の登録商標です。
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のSII Printer Driver for Windows上で動作するSDK（以後：SDK）について説明します。

対象プリンタードライバーについて

SDKでサポートするプリンタードライバーを下記に記載します。

- SII Printer Driver for Windows（以後：プリンタードライバー）

用語について

本書で使用されている用語について下記に記載します。

用語	内容
ASB設定コマンド （ASB: Automatic Status Back）	プリンターコマンドにおける「自動ステータス送信の有効・無効」。 詳細は、「MP-A40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。
プリンターステータス	SDKで取得できるプリンターのステータス情報。 プリンターコマンド「自動ステータス送信の有効・無効」に応答するステータスです。詳細は『6.1 プリンターステータス一覧』を参照してください。

免責

ご利用いただいた結果の影響については、一切の責任を負いかねます。

本製品がお客様により不適切に使用されたり、本書の内容に従わずに取り扱われたり、または弊社以外の第三者により修理・変更されたことなどに起因して発生した損害などにつきましては責任を負いかねます。

1章 概要 1-1

1.1 はじめに 1-1
1.2 動作条件 1-1

2章 インストール 2-1

3章 Win32 API 3-1

3.1 概要 3-1
3.2 開発対象言語 3-1
3.3 ライブラリファイル 3-1
3.4 API一覧 3-2
3.5 API詳細 3-4
MpOpen 3-4
MpClose 3-5
MpSetWriteTimeout 3-5
MpGetWriteTimeout 3-6
MpSetResponseTimeout 3-7
MpGetResponseTimeout 3-8
MpWrite 3-8
MpRead 3-9
MpWriteAndWaitResponse 3-10
MpReset 3-12
MpGetStatus 3-13
MpSetCallbackFunction 3-13
MpRegisterStyleSheet 3-14
MpDeleteStyleSheet 3-15
MpRegisterLogo 3-16
MpRegisterLogoEx 3-17
MpDeleteLogo 3-18
MpGetPrinterInformation 3-19
MpControlTransaction 3-20
MpSelectStandardMode 3-22
MpSelectPageMode 3-22
MpPrintPageModeData 3-23
MpSetPageModeVerticalPosition 3-24
MpSetStandardModeArea 3-24
MpSetStandardModeAlignment 3-25
MpSetHorizontalPosition 3-26
MpSetStandardModeBarcodeDirection 3-27
MpSetLineSpacing 3-27
MpSetCharacterRightSpace 3-28
MpSelectCharacterSet 3-29
MpSelectInternationalCharacterSet 3-30
MpSetCharacterFormatting 3-31
MpPrintText 3-33
MpPrintLogo 3-34
MpSendDataFile 3-34

MpSendDataFileEx 3-36
MpPrintBarcode 3-38
MpPrint2Dcode 3-41
MpPrintPageModeRectangle 3-43
MpFeedLine 3-45
MpFeedDotLine 3-45
MpFeedCutPosition 3-46
MpFeedMarkPosition 3-46

4章 .NET API 4-1
4.1 概要 4-1
4.2 開発対象言語 4-1
4.3 ライブラリファイル 4-1
4.4 API一覧 4-2
4.4.1 PrinterManagerクラス 4-2
4.4.2 PrinterExceptionクラス 4-3
4.5 API詳細（PrinterManagerクラス） 4-4
4.5.1 プロパティ 4-4
IsOpened 4-4
4.5.2 メソッド 4-5
Open 4-5
Close 4-5
SetWriteTimeout 4-6
GetWriteTimeout 4-6
SetResponseTimeout 4-7
GetResponseTimeout 4-7
Write 4-8
Read 4-8
GetReadSize 4-9
WriteAndWaitResponse 4-9
Reset 4-10
GetStatus 4-11
StartCallbackFunction 4-11
StopCallbackFunction 4-12
RegisterStyleSheet 4-12
DeleteStyleSheet 4-13
RegisterLogo 4-13
RegisterLogoEx 4-14
DeleteLogo 4-15
GetPrinterInformation 4-15
ControlTransaction 4-16
SelectStandardMode 4-16
SelectPageMode 4-17
PrintPageModeData 4-18
SetPageModeVerticalPosition 4-18
SetStandardModeArea 4-19
SetStandardModeAlignment 4-19
SetHorizontalPosition 4-20
SetStandardModeBarcodeDirection 4-20
SetLineSpacing 4-21
SetCharacterRightSpace 4-21

SelectCharacterSet 4-22
SelectInternationalCharacterSet 4-22
SetCharacterFormatting 4-23
PrintText 4-24
PrintLogo 4-24
SendDataFile 4-25
SendDataFileEx 4-25
PrintBarcode 4-26
Print2Dcode 4-27
PrintPageModeRectangle 4-28
FeedLine 4-29
FeedDotLine 4-30
FeedCutPosition 4-31
FeedMarkPosition 4-31
4.5.3 イベント 4-32
statusChanged 4-32
4.6 API詳細（PrinterExceptionクラス） 4-33
4.6.1 メソッド 4-33
GetErrorCode 4-33

5章 エラーコード一覧 5-1

5.1 エラーコード一覧 5-1

6章 引数情報 6-1

6.1 プリンターステータス一覧 6-1
6.2 プリンター情報 6-3
6.3 一括処理制御方法 6-4
6.4 印字方向 6-5
6.5 印字位置 6-5
6.6 文字セット 6-5
6.7 国際文字セット 6-6
6.8 文字フォント 6-7
6.9 文字サイズ 6-7
6.10 アンダーライン 6-7
6.11 強調印字 6-8
6.12 白黒反転印字 6-8
6.13 回転印字 6-9
6.14 バーコード種類 6-10
6.15 バーコードモジュール幅 6-10
6.16 バーコード高さ 6-11
6.17 バーコードHRI文字 6-12
6.18 バーコードN:W比 6-13
6.19 2次元バーコード種類 6-13
6.20 2次元バーコードモード 6-13
6.21 2次元バーコードモジュールサイズ 6-15
6.22 2次元バーコードモジュール高さ 6-16
6.23 2次元バーコードカラム数 6-16

6.24 2次元バーコード段数 6-16
6.25 2次元バーコードエラー訂正レベル 6-17

1章　概要

1.1 はじめに

本章ではSDKの概要について説明します。

SDKとは、開発者向けに提供されるプリンターを直接制御するためのダイナミックリンクライブラリです。

SDKは、プリンタードライバーに付属し、プリンタードライバーを利用して動作します。

SDKの利用により、アプリケーション開発においてプリンターを直接制御し、ポートの種類に依存しない設計ができます。

SDKの利用例として、各言語別のサンプルプログラムを提供していますので参照してください。

1.2 動作条件

プリンタードライバーの動作環境、メモリスイッチの利用条件及び制限事項に準じます。

詳細は「MP-A40シリーズ プリンタードライバー 取扱説明書」を参照してください。

また、上記以外に、下記の動作条件を満たす必要があります。

- .NET Framework Version 2.0以降のインストール
- 双方向サポートの機能が有効であること
  （設定方法は「MP-A40シリーズ プリンタードライバー 取扱説明書」を参照してください。）
- プリンタープールの機能が無効であること
  （設定方法は「MP-A40シリーズ プリンタードライバー 取扱説明書」を参照してください。）
- 下記のすべてに該当する場合は、共有プリンターを利用してください。（共有プリンターを利用しない場合、送信データが割り込む可能性があります。）
  ① TCP/IP接続で複数のホストから1台のプリンターを利用する場合
  ② 送信データを複数回に分割して送信する場合
  送信データを一括で送信する方法については、(Mp)ControlTransactionを参照してください。

2章　インストール

インストールについては、「SII Printer Software for Windows インストールガイド」を参照してください。

3章　Win32 API

3.1 概要

本章ではWin32開発環境用SDK（Win32 API）について説明します。

3.2 開発対象言語

開発対象言語は下記の通りです。

- Visual C++ ※1

※1：　Visual C++ 2008 以降を推奨

3.3 ライブラリファイル

ライブラリファイルのファイル名は下記の通りです。

- SiiMpa4Api.dll

ライブラリファイルの保管場所は、Windowsのシステムフォルダーです。

特殊な場合を除き、ライブラリファイルが保存されているフォルダーへパスを通す必要はありません。フォルダーから移動せずにそのまま使用してください。

ライブラリファイルを移動するとプリンタードライバーのバージョンアップの際にライブラリファイルが正しく更新されません。

3.4 API一覧

SDKに実装されているAPIは下記の通りです。

✓: 使用可
－: 使用不可

API	機能概略	スタンダードモード	ページモード
MpOpenA※1 ※2 MpOpenW※1 ※2	指定したプリンターに対しSDKの使用を開始し、セッションIDを返します。	✓	✓
MpClose	指定したセッションIDでのSDKの使用を終了します。	✓	✓
MpSetWriteTimeout	データ送信を行うAPIのタイムアウト値を設定します。	✓	✓
MpGetWriteTimeout	データ送信を行うAPIのタイムアウト値を取得します。	✓	✓
MpSetResponseTimeout	データ受信を行うAPIのタイムアウト値を設定します。	✓	✓
MpGetResponseTimeout	データ受信を行うAPIのタイムアウト値を取得します。	✓	✓
MpWrite※3	プリンターへバイナリデータを送信します。	✓	✓
MpRead	プリンターから受信したデータを取得します。	✓	✓
MpWriteAndWaitResponse※3	バイナリデータの送信後、応答データを受信します。	✓	✓
MpReset※4	プリンターをリセットします。	✓	✓
MpGetStatus	最新のプリンターステータスを取得します。	✓	✓
MpSetCallbackFunction	プリンターステータスの変化を検知した際に呼び出されるコールバック関数を登録します。	✓	✓
MpRegisterStyleSheetA※1 ※3 MpRegisterStyleSheetW※1 ※3	プリンターにスタイルシートを登録します。	✓	✓
MpDeleteStyleSheet※3	プリンターからスタイルシートを削除します。	✓	✓
MpRegisterLogoA※1 ※3 MpRegisterLogoW※1 ※3	プリンターにロゴを登録します。	✓	✓
MpDeleteLogo※3	プリンターから、登録済みのロゴを削除します。	✓	✓
MpGetPrinterInformation※3	指定したプリンター情報を取得します。	✓	✓
MpControlTransaction※5	対象のAPIを一括で処理します。	✓	✓
MpSelectStandardMode※3	スタンダードモードを開始します。	－	✓
MpSelectPageMode※3	ページモードの印字領域と印字方向を指定し、ページモードを開始します。	✓	－
MpPrintPageModeData※3	ページモードデータを印字します。	－	✓※6
MpSetPageModeVerticalPosition※3	ページモードの縦方向の絶対位置を設定します。	－	✓※6

API	機能概略	スタンダードモード	ページモード
MpSetStandardModeArea※3	スタンダードモードの印字領域を設定します。	✓※7	–
MpSetStandardModeAlignment※3	スタンダードモードでの印字位置を設定します。	✓※7	–
MpSetHorizontalPosition※3	横方向の絶対位置を設定します。	✓	✓
MpSetStandardModeBarcodeDirection※3	バーコードの印字方向を設定します。	✓※7	–
MpSetLineSpacing※3	改行量を設定します。	✓※8	✓※8
MpSetCharacterRightSpace※3	文字の右スペース量を設定します。	✓※8	✓※8
MpSelectCharacterSet	文字セットを設定します。	✓	✓
MpSelectInternationalCharacterSet	国際文字セットを設定します。	✓	✓
MpSetCharacterFormatting※3	文字修飾を設定します。	✓	✓
MpPrintTextA※1 ※3 MpPrintTextW※1 ※3	文字列を印字します。	✓	✓
MpPrintLogo※3	プリンターに登録されたイメージを印字します。	✓	✓
MpSendDataFileA※1 ※3 MpSendDataFileW※1 ※3	指定したファイルのデータを送信します。	✓	✓
MpPrintBarcode※3	バーコードを印字します。	✓	✓
MpPrint2Dcode※3	2次元バーコードを印字します。	✓	✓
MpPrintPageModeRectangle※3	ページモード選択時に、矩形を印字します。	–	✓※6
MpFeedLine※3	行単位で紙送りをします。	✓	✓
MpFeedDotLine※3	ドット単位で紙送りをします。	✓	✓
MpFeedCutPosition※3	カット位置まで紙送りをします。	✓	✓
MpFeedMarkPosition※3	マーク紙の頭出しによる紙送りをします。	✓	✓

※1: 文字列の引数はMBCS（マルチバイトコード）またはUNICODE（ユニコード）で指定します。MBCS（マルチバイトコード）を使用する場合は末尾に'A'、UNICODE（ユニコード）を使用する場合は末尾に'W'のついたAPIを使用してください。以降の説明ではAPI末尾の'A'または'W'を省略して説明します。

※2: プリンターと未接続または電源オフの場合でも、本APIは成功します。

※3: スプーラーに印刷ジョブが存在する場合や、プリンターと未接続または通信不能状態の場合、本APIは失敗します。

※4: スプーラーに印刷ジョブが存在する場合や、プリンターと未接続または電源オフの場合、本APIは失敗します。

※5: スプーラーに印刷ジョブが存在する場合や、プリンターと未接続または通信不能状態の場合、本APIを「一括印刷と一括処理の終了」で実行すると失敗します。

※6: 本APIを使用する場合、事前にMpSelectPageModeを実行し、ページモードを開始してください。

※7: 本APIを使用する場合、事前にMpSelectStandardModeを実行し、スタンダードモードに切り替えてください。

※8: スタンダードモードとページモードでそれぞれ独立した設定が可能です。

3.5 API詳細

注意

- ◆ Bluetooth接続の場合、接続が一時的に切断されてしまうと、応答データの一部が取得できない場合があります。
- ◆ Bluetooth接続の場合、プリンターと未接続中の応答データの取得はできません。

MpOpen

指定したプリンターに対しSDKの使用を開始し、セッションIDを返します。

INT **MpOpen**(
LPCTSTR *pszPrnName*,
LPDWORD *pdwSessionId*)

パラメーター

pszPrnName
SDKを使用するプリンター名
プリンター名（フレンドリ名）を指定します。

pdwSessionId
セッションID
プリンターを識別するセッションIDの変数を指定します。

戻り値

成功時は0を返します。

失敗時はエラーコードを返します。詳細は『5章 エラーコード一覧』を参照してください。

メモ

- 1プロセスについてのオープン数は最大8つです。
- 本APIで取得したセッションIDを使用しなくなった場合は、必ず**MpClose**を使って無効にしてください。
- プリンタードライバーの接続先がUSB、Bluetooth、またはTCP/IP以外の場合は、本APIは失敗します。

MpClose

指定したセッションIDでのSDKの使用を終了します。

INT **MpClose**(
DWORD *dwSessionId*)

パラメーター

dwSessionId
セッションID
MpOpenで取得したセッションIDを指定します。

戻り値

成功時は0を返します。

失敗時はエラーコードを返します。詳細は『5章 エラーコード一覧』を参照してください。

メモ

- 本APIで呼び出したセッションIDが他のAPIで使用中の場合、その処理が完了するまで本APIは実行されません。
- **MpControlTransaction**によって保持しているコマンドは破棄されます。
- 本APIで指定したセッションIDに関連付けられたすべての設定は破棄されます。

MpSetWriteTimeout

データ送信を行うAPIのタイムアウト値を設定します。

INT **MpSetWriteTimeout**(
DWORD *dwSessionId*,
DWORD *dwTimeout*)

パラメーター

dwSessionId
セッションID
MpOpenで取得したセッションIDを指定します。

dwTimeout
データ送信タイムアウト値
データ送信を行うAPIの送信待ち時間(ミリ秒単位)を指定します。
設定範囲は3000～90000ミリ秒です。3000ミリ秒未満の値を指定した場合は3000ミリ秒に、90000ミリ秒を超える値を指定した場合は90000ミリ秒に設定されます。
設定は下記のAPIで有効です。

- MpWrite
- MpWriteAndWaitResponse（送信処理部分）
- MpRegisterStyleSheet
- MpRegisterLogo
- MpControlTransaction（「一括印字と一括処理の終了」機能選択時）
- MpPrintText
- MpSendDataFile
- MpPrintBarcode
- MpPrint2Dcode

初期値

15000ミリ秒

戻り値

成功時は0を返します。

失敗時はエラーコードを返します。詳細は『5章　エラーコード一覧』を参照してください。

メモ

- 本APIで設定した値はMpGetWriteTimeoutで確認できます。

MpGetWriteTimeout

データ送信を行うAPIのタイムアウト値を取得します。

INT **MpGetWriteTimeout**(
DWORD *dwSessionId*,
LPDWORD *pdwTimeout*)

パラメーター

dwSessionId
セッションID
MpOpenで取得したセッションIDを指定します。

pdwTimeout
データ送信タイムアウト値
データ送信を行うAPIの送信待ち時間（ミリ秒単位）を取得する変数を指定します。

戻り値

成功時は0を返します。

失敗時はエラーコードを返します。詳細は『5章　エラーコード一覧』を参照してください。

メモ

- 本APIで取得する値は、MpSetWriteTimeoutで設定できます。

MpSetResponseTimeout

データ受信を行うAPIのタイムアウト値を設定します。

INT **MpSetResponseTimeout**(
DWORD *dwSessionId*,
DWORD *dwTimeout*)

パラメーター

dwSessionId
セッションID
MpOpenで取得したセッションIDを指定します。

dwTimeout
データ受信タイムアウト値
データ受信を行うAPIの受信待ち時間(ミリ秒単位)を指定します。
設定範囲は3000～90000ミリ秒です。3000ミリ秒未満の値を指定した場合は3000ミリ秒に、90000ミリ秒を超える値を指定した場合は90000ミリ秒に設定されます。
設定は下記のAPIで有効です。

- **MpWriteAndWaitResponse**(受信処理部分)
- **MpGetPrinterInformation**

初期値

3000ミリ秒

戻り値

成功時は0を返します。

失敗時はエラーコードを返します。詳細は『5章　エラーコード一覧』を参照してください。

メモ

- 本APIで設定した値は**MpGetResponseTimeout**で確認できます。

MpGetResponseTimeout

データ受信を行うAPIのタイムアウト値を取得します。

INT **MpGetResponseTimeout**(
DWORD *dwSessionId*,
LPDWORD *pdwTimeout*)

パラメーター

dwSessionId
セッションID
MpOpenで取得したセッションIDを指定します。

pdwTimeout
データ受信タイムアウト値
データ受信を行うAPIの受信待ち時間(ミリ秒単位)を取得する変数を指定します。

戻り値

成功時は0を返します。

失敗時はエラーコードを返します。詳細は『5章 エラーコード一覧』を参照してください。

メモ

- 本APIで取得する値は、**MpSetResponseTimeout**で設定できます。

MpWrite

プリンターへバイナリデータを送信します。

INT **MpWrite**(
DWORD *dwSessionId*,
LPBYTE *pCmd*,
DWORD *cbCmd*,
DWORD *dwOffset*)

パラメーター

dwSessionId
セッションID
MpOpenで取得したセッションIDを指定します。

pCmd
送信データバッファー

プリンターへ送信するデータを格納したバッファーを指定します。

cbCmd

送信データサイズ

*pCmd*が指すバッファーのサイズをバイト単位で指定します。

dwOffset

送信データオフセット

送信するデータの開始位置を指定します。

戻り値

成功時は0を返します。

失敗時はエラーコードを返します。詳細は『5章　エラーコード一覧』を参照してください。

メモ

- 送信データサイズ（*cbCmd*）よりも送信データオフセット（*dwOffset*）が大きい場合、本APIは失敗します。
- **MpControlTransaction**を使用しない場合、本APIの送信タイムアウト時間は、**MpGetWriteTimeout**で設定した値となります。
- 本APIによるデータ送信は、プリンタードライバーのジョブには含まれません。
- 本APIは**MpReset**により中断されます。
- イメージデータなどの途中で別データの割り込みが許されないコマンドやデータの場合、一括で出力してください。分割して出力した場合、別プロセスからのデータが割り込む恐れがあります。全てのデータを1度の本APIの呼び出しにより出力するようにしてください。
- 本APIでASB設定コマンドを無効に設定するデータを使用しないでください。ASB設定コマンドを無効に設定するデータが含まれた場合、プリンターステータスを取得するAPIが正常に動作しなくなります。
- プリンターに書き込まれたデータのバイト数が*cbCmd*のサイズに満たない場合、本APIは失敗します。
- 送信するデータにプリンターコマンドの「ハードウェアリセット」及び「プリンタリセット」を含めないでください。
  プリンターをリセットする場合、**MpReset**を使用してください。

MpRead

プリンターから受信したデータを取得します。

INT **MpRead**(
DWORD *dwSessionId*,
LPBYTE *pData*,
DWORD *cbData*,
LPDWORD *pcbNeeded*)

パラメーター

dwSessionId

セッションID

MpOpenで取得したセッションIDを指定します。

pData

受信データバッファー

受信するデータを格納したバッファーを指定します。

cbData

最大受信データサイズ

*pData*が指すバッファーのサイズをバイト単位で指定します。
最大受信データサイズは、4096バイトです。最大受信データサイズを超えた分は、古いデータから破棄されます。

pcbNeeded

データサイズ変数

受信済みデータサイズ、または受信に必要なバッファーサイズを格納するための変数を指定します。

戻り値

成功時は0を返します。

失敗時はエラーコードを返します。詳細は『5章　エラーコード一覧』を参照してください。

メモ

- 本APIでは**MpOpen**の呼び出し後に受信した（未変換）データを取得します。
- 本APIでの受信データには、別プロセスによる操作に起因する応答データが含まれることがあります。
- プリンターに再接続した場合、プリンターに蓄積されていた過去のデータをまとめて受信する場合があります。
- 応答データサイズを取得するには、*cbData*に0を指定してください。本APIは失敗（ERR_WORKAREA_NO_MEMORY(-260)）し、*pcbNeeded*に指定した変数に応答データサイズが格納されます。

MpWriteAndWaitResponse

バイナリデータの送信後、応答データを受信します。

INT **MpWriteAndWaitResponse**(
DWORD *dwSessionId*,
LPBYTE *pCmd*,
DWORD *cbCmd*,
LPBYTE *pResp*,
DWORD *cbResp*,
LPDWORD *pcbNeeded*,
BOOL *bRespFlag*,
BOOL *bIncASBData*)

パラメーター

dwSessionId

セッションID

MpOpenで取得したセッションIDを指定します。

pCmd

送信データバッファー

送信するデータを格納したバッファーを指定します。

cbCmd

送信データサイズ

送信するデータのサイズを指定します。

pResp

受信データバッファー

取得するデータを格納するバッファーを指定します。

cbResp

最大受信データサイズ

プリンターから受信するデータの最大サイズを指定します。
データ取得が必要ない場合、0を指定します。
最大受信データサイズは、4096バイトです。最大受信データサイズを超えた分は、4096バイトに設定されます。

pcbNeeded

データサイズ変数

受信済みデータサイズ、または受信に必要なバッファーサイズを格納するための変数を指定します。

bRespFlag

受信動作フラグ

受信動作を指定します。

TRUE : 何らかのデータを受信するか、またはタイムアウトになるまで受信を続けます。

FALSE : 最大受信データサイズ分を受信するか、またはタイムアウトになるまで受信を続けます。

bIncASBData

受信データにASB設定コマンドの応答を含めるかを指定します。

受信対象のデータを指定します。

TRUE : ASB設定コマンドの応答を含めたデータを取得します。

FALSE : ASB設定コマンドの応答を除いたデータを取得します。

戻り値

成功時は0を返します。

失敗時はエラーコードを返します。詳細は『5章 エラーコード一覧』を参照してください。

メモ

- 本APIでは本API実行中に受信した（未変換）データを取得します。
- 本APIでの受信データには、別プロセスによる操作に起因する応答データが含まれることがあります。
- 本APIは**MpReset**により中断されます。
- プリンターに書き込まれたデータのバイト数が*cbCmd*のサイズに満たない場合、本APIは失敗します。
- 本APIの送信タイムアウト時間は、**MpGetWriteTimeout**で設定した値になります。
- 本APIの受信タイムアウト時間は、**MpGetResponseTimeout**で設定した値になります。
- イメージデータなどの途中で別データの割り込みが許されないコマンドやデータの場合、一括で出力してください。分割して出力した場合、別プロセスからのデータが割り込む恐れがあります。全てのデータを1度の本APIの呼び出しにより出力するようにしてください。

- 本APIでASB設定コマンドを無効に設定するデータを使用しないでください。ASB設定コマンドを無効に設定するデータが含まれた場合、プリンターステータスを取得するAPIが正常に動作しなくなります。
- 送信するバイナリデータにプリンターコマンドの「ハードウェアリセット」及び「プリンタリセット」を含めないでください。
  プリンターをリセットする場合、**MpReset**を使用してください。
- 応答データサイズを取得するには、*cbResp*に0を指定してください。本APIは失敗（ERR_WORKAREA_NO_MEMORY(-260)）し、*pcbNeeded*に指定した変数に応答データサイズが格納されます。

MpReset

プリンターをリセットします。

INT **MpReset**(
DWORD *dwSessionId*)

パラメーター

dwSessionId
セッションID
MpOpenで取得したセッションIDを指定します。

戻り値

成功時は0を返します。

失敗時はエラーコードを返します。詳細は『5章　エラーコード一覧』を参照してください。

メモ

- 本APIを呼び出した場合、下記のAPIは中断されます。
  - **MpWrite**
  - **MpWriteAndWaitResponse**
  - **MpControlTransaction**（「一括印字と一括処理の終了」機能選択時）
  - **MpPrintText**
  - **MpSendDataFile**
- 本APIを実行した後、数秒経過してからデータ送信してください。本API直後にデータ送信を行うと、データ抜けの原因になります。
- 本API実行中、プリンターステータスは未接続の応答になります。
- Bluetooth接続で、プリンターがデータを受け付けない状態で本APIを実行した場合、本API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。

MpGetStatus

最新のプリンターステータスを取得します。

INT **MpGetStatus**(
DWORD *dwSessionId*,
LPDWORD *pdwStatus*)

パラメーター

dwSessionId
セッションID
MpOpenで取得したセッションIDを指定します。

pdwStatus
プリンターステータス
プリンターステータスを格納する変数を指定します。

戻り値

成功時は0を返します。

失敗時はエラーコードを返します。詳細は『5章 エラーコード一覧』を参照してください。

メモ

- プリンターの再接続を検出した場合のプリンターステータスは、その時点で最後に受信している値になります。
- プリンターステータスの内容は『6.1 プリンターステータス一覧』を参照してください。

MpSetCallbackFunction

プリンターステータスの変化を検知した際に呼び出されるコールバック関数を登録します。

INT **MpSetCallbackFunction**(
DWORD *dwSessionId*,
INT (CALLBACK EXPORT **lpfnCallBackStatus*) (DWORD *dwStatus*))

パラメーター

dwSessionId
セッションID
MpOpenで取得したセッションIDを指定します。

lpfnCallBackStatus

コールバック関数アドレス

プリンターステータスを受け取るアプリケーション定義のコールバック関数のアドレスを指定します。
NULLを指定した場合、プリンターステータスの監視を中断します。

dwStatus

プリンターステータス

プリンターステータスが格納される変数を指定します。

戻り値

成功時は0を返します。

失敗時はエラーコードを返します。詳細は『5章　エラーコード一覧』を参照してください。

メモ

- 登録されたコールバック関数内から同一のセッションIDでSDKのAPIを呼び出すことはできません。
- プリンターの再接続を検出した場合のプリンターステータスは、その時点で最後に受信している値になります。
- プリンターステータスを受信した場合でも、直前に受信したプリンターステータスに対して変化がない場合、コールバック関数は呼び出されません。
- 本APIによりコールバック関数を登録すると、現在のプリンターステータスでコールバック関数を呼び出します。
- コールバック関数を登録した状態で本APIを呼び出した場合、登録済みの関数は無効となり、新しいコールバック関数が登録されます。
- コールバック関数の戻り値は無視されます。
- プリンターステータスの受信からコールバック関数の呼び出しまでの時間は保証されません。
- プリンターステータスの内容は『6.1　プリンターステータス一覧』を参照してください。

MpRegisterStyleSheet

プリンターにスタイルシートを登録します。

INT **MpRegisterStyleSheet**(
DWORD *dwSessionId*,
LPCTSTR *pszFilePath*,
BYTE *byRegNum*)

パラメーター

dwSessionId

セッションID

MpOpenで取得したセッションIDを指定します。

pszFilePath

スタイルシートのファイルパス

プリンターに登録するスタイルシートのファイルパスを指定します。
ファイルパスには、CSSファイル(*.css)を指定してください。

byRegNum

スタイルシート番号

プリンターに登録するスタイルシートの番号を指定します。
1から4の範囲で値を指定してください。
範囲外の値を指定した場合、本APIは失敗(ERR_PARAM(-90))します。

戻り値

成功時は0を返します。

失敗時はエラーコードを返します。詳細は『5章 エラーコード一覧』を参照してください。

メモ

- 1つのスタイルシートの中で登録できるスタイル数は最大64です。
- 登録できるスタイルシートの作成の詳細は、「MP-A40シリーズ サーマルプリンタ 技術説明書」の「スタイルシート登録」コマンドを参照してください。
- 読み込み可能なファイルサイズは最大4294967295バイトです。
  ただしシステム上にメモリを確保できなかった場合、本APIは失敗します。
- 本APIの送信タイムアウト時間は、**MpGetWriteTimeout**で設定した値になります。

MpDeleteStyleSheet

プリンターから、登録済みのスタイルシートを削除します。

INT **MpDeleteStyleSheet**(
DWORD *dwSessionId*,
BYTE *byRegNum*)

パラメーター

dwSessionId

セッションID

MpOpenで取得したセッションIDを指定します。

byRegNum

スタイルシート番号

プリンターから削除するスタイルシートの番号を指定します。
1から4の範囲で値を指定してください。
範囲外の値を指定した場合、本APIは失敗(ERR_PARAM(-90))します。

戻り値

成功時は0を返します。

失敗時はエラーコードを返します。詳細は『5章 エラーコード一覧』を参照してください。

MpRegisterLogo

プリンターにロゴを登録します。

INT **MpRegisterLogo**(
DWORD *dwSessionId*,
LPCTSTR *pszFilePath*,
BYTE *byRegNum*)

パラメーター

dwSessionId
セッションID
MpOpenで取得したセッションIDを指定します。

pszFilePath
ロゴイメージデータのファイルパス
プリンターに登録するロゴのファイルパスを指定します。
ファイルパスには、Windowsビットマップファイル(*.bmp)またはJPEGファイル(*.jpg / *.jpeg)を指定してください。
本APIでロゴイメージデータを登録する場合は、ディザリング有効に固定されます。

byRegNum
ロゴ番号
プリンターに登録するロゴ番号を指定します。
0から99の範囲で値を指定してください。
範囲外の値を指定した場合、本APIは失敗(ERR_PARAM(-90))します。

戻り値

成功時は0を返します。

失敗時はエラーコードを返します。詳細は『5章 エラーコード一覧』を参照してください。

メモ

- 読み込み可能なファイルサイズは最大4294967295バイトです。
  ただしシステム上にメモリを確保できなかった場合、本APIは失敗します。
- 登録可能なロゴイメージサイズは、最大幅8192ドット、最大高さ2304ドットです。
- 本APIで登録したロゴの登録状況は、**MpGetPrinterInformation**で確認できます。*dwPrnInfo* に GET_NV_MEM_KEYCODE_LISTを指定してください。
- 同じロゴ番号に対して登録を行った場合、後から登録したロゴイメージデータが、有効になります。
- 本APIの送信タイムアウト時間は、**MpGetWriteTimeout**で設定した値になります。

MpRegisterLogoEx

プリンターにロゴを登録します。

INT **MpRegisterLogoEx**(
DWORD *dwSessionId*,
LPCTSTR *pszFilePath*,
BYTE *byRegNum*,
BYTE *byOption*)

パラメーター

dwSessionId

セッションID

MpOpenで取得したセッションIDを指定します。

pszFilePath

ロゴイメージデータのファイルパス

プリンターに登録するロゴのファイルパスを指定します。
ファイルパスには、Windowsビットマップファイル(*.bmp)またはJPEGファイル(*.jpg / *.jpeg)を指定してください。

byRegNum

ロゴ番号

プリンターに登録するロゴ番号を指定します。
0から99の範囲で値を指定してください。
範囲外の値を指定した場合、本APIは失敗(ERR_PARAM(-90))します。

byOption

オプション

プリンターに登録する際のオプションを指定します。
下記の値を指定してください。

byOption	内容
0	ディザリング有効
1	ディザリング無効

戻り値

成功時は0を返します。

失敗時はエラーコードを返します。詳細は『5章 エラーコード一覧』を参照してください。

メモ

- 読み込み可能なファイルサイズは最大4294967295バイトです。
  ただしシステム上にメモリを確保できなかった場合、本APIは失敗します。
- 登録可能なロゴイメージサイズは、最大幅8192ドット、最大高さ2304ドットです。

- 本APIで登録したロゴの登録状況は、**MpGetPrinterInformation**で確認できます。*dwPrnInfo* に GET_NV_MEM_KEYCODE_LISTを指定してください。
- 同じロゴ番号に対して登録を行った場合、後から登録したロゴイメージデータが、有効になります。
- 本APIの送信タイムアウト時間は、**MpGetWriteTimeout**で設定した値になります。

MpDeleteLogo

プリンターから、登録済みのロゴを削除します。

INT **MpDeleteLogo**(
DWORD *dwSessionId*,
BYTE *byRegNum*)

パラメーター

dwSessionId
セッションID
MpOpenで取得したセッションIDを指定します。

byRegNum
ロゴ番号
プリンターから削除するロゴ番号を指定します。
0から99の範囲で値を指定してください。
範囲外の値を指定した場合、本APIは失敗(ERR_PARAM(-90))します。

戻り値

成功時は0を返します。

失敗時はエラーコードを返します。詳細は『5章 エラーコード一覧』を参照してください。

メモ

- 本API実行後のロゴの登録状況は、**MpGetPrinterInformation**で確認できます。*dwPrnInfo*に GET_NV_MEM_KEYCODE_LISTを指定してください。

MpGetPrinterInformation

指定したプリンター情報を取得します。

INT **MpGetPrinterInformation**(
DWORD *dwSessionId*,
DWORD *dwPrnInfo*,
LPBYTE *pData*,
DWORD *cbData*,
LPDWORD *pdwData*,
LPDWORD *pcbNeeded*)

パラメーター

dwSessionId
セッションID
MpOpenで取得したセッションIDを指定します。

dwPrnInfo
プリンター情報
取得するプリンター情報を指定します。
指定するプリンター情報の応答形式が数値の場合、*pdwData*で指定した変数に*pData*の内容を数値化して格納します。それ以外の場合、*pdwData*は不定になります。
無効な値を指定した場合、本APIは失敗(ERR_PARAM(-90))します。
プリンター情報の内容と応答形式は『6.2 プリンター情報』を参照してください。

pData
受信データバッファー
取得するデータを格納するバッファーを指定します。

cbData
受信データサイズ
プリンターから受信するデータの最大サイズを指定します。

pdwData
受信データ(数値)
*dwPrnInfo*の応答形式が数値の場合、*pData*を数値化した値を格納する変数を指定します。
*dwPrnInfo*の応答形式が数値以外の場合、NULLを指定してください。

pcbNeeded
データサイズ変数
受信済みデータサイズ、または受信に必要なバッファーサイズを格納するための変数を指定します。

戻り値

成功時は0を返します。

失敗時はエラーコードを返します。詳細は『5章 エラーコード一覧』を参照してください。

メモ

- 本APIの受信タイムアウト時間は、**MpGetResponseTimeout**で設定した値になります。
- 本APIが失敗した場合、*pData*及び*pcbNeeded*の値は不定となります。
- 応答データサイズを取得するには、*cbResp*に0を指定してください。本APIは失敗（ERR_WORKAREA_NO_MEMORY(-260)）し、*pcbNeeded*に指定した変数に応答データサイズが格納されます。

MpControlTransaction

対象のAPIを一括で処理します。

INT **MpControlTransaction**(
DWORD *dwSessionId*,
BYTE *byControl*)

パラメーター

dwSessionId
セッションID
MpOpenで取得したセッションIDを指定します。

byControl
制御方法
一括処理の制御方法を指定します。
無効な値を指定した場合、本APIは失敗（ERR_PARAM(-90)）します。
制御方法の内容は『6.3 一括処理制御方法』を参照してください。

戻り値

成功時は0を返します。

失敗時はエラーコードを返します。詳細は『5章 エラーコード一覧』を参照してください。

メモ

- 本APIによる一括処理の対象は、下記のAPIになります。
  - **MpWrite**
  - **MpSelectStandardMode**
  - **MpSelectPageMode**
  - **MpPrintPageModeData**
  - **MpSetPageModeVerticalPosition**
  - **MpSetStandardModeArea**
  - **MpSetStandardModeAlignment**
  - **MpSetHorizontalPosition**
  - **MpSetStandardModeBarcodeDirection**
  - **MpSetLineSpacing**
  - **MpSetCharacterRightSpace**
  - **MpSetCharacterFormatting**
  - **MpPrintText**
  - **MpPrintLogo**

- MpSendDataFile
- MpPrintBarcode
- MpPrint2Dcode
- MpPrintPageModeRectangle
- MpFeedLine
- MpFeedDotLine
- MpFeedCutPosition
- MpFeedMarkPosition

- 登録された処理は、本APIを「一括印字と一括処理の終了」で実行すると、一括で印字した後に破棄されます。
  登録された処理を使用しない場合、本APIを「一括処理の終了」で実行すると、登録された処理を印字せずに破棄します。
- 「一括処理の開始」を実行せずに「一括処理の終了」を実行した場合でも、本APIは成功します。
- 「一括処理の開始」を実行せずに「一括印字と一括処理の終了」を実行した場合、本APIは失敗します。
- 「一括処理の開始」を実行し、その後一度も処理を登録しない状態で「一括処理の終了」または「一括印字と一括処理の終了」を実行した場合でも、本APIは成功します。
- 「一括処理の開始」を実行し、「一括処理の終了」または「一括印字と一括処理の終了」を実行せずに、再度「一括処理の開始」を実行した場合、これまでに登録された処理を破棄し、新たな一括処理を開始します。
- 本APIを「一括印字と一括処理の終了」で実行中の場合、MpResetにより中断されます。
- 「一括印字と一括処理の終了」を実行中に、別スレッドから「一括処理の開始」を実行した場合、下記の動作となります。
  - 実行中の「一括印字と一括処理の終了」は、そのまま継続します。
  - 別スレッドの「一括処理の開始」は、別の「一括処理の開始」として新たにバッファリングを開始します。

- 「一括印字と一括処理の終了」を実行時の本APIの送信タイムアウト時間は、MpGetWriteTimeoutで設定した値になります。

MpSelectStandardMode

スタンダードモードを開始します。

INT **MpSelectStandardMode**(
DWORD *dwSessionId*)

パラメーター

dwSessionId
セッションID
MpOpenで取得したセッションIDを指定します。

戻り値

成功時は0を返します。

失敗時はエラーコードを返します。詳細は『5章 エラーコード一覧』を参照してください。

メモ

- ページモード実行中に本APIを実行した場合、ページモードデータを破棄し、スタンダードモードを開始します。

MpSelectPageMode

ページモードの印字領域と印字方向を指定し、ページモードを開始します。

INT **MpSelectPageMode**(
DWORD *dwSessionId*,
WORD *wStartX*,
WORD *wStartY*,
WORD *wWidth*,
WORD *wHeight*,
BYTE *byDirection*)

パラメーター

dwSessionId
セッションID
MpOpenで取得したセッションIDを指定します。

wStartX
横方向始点
ページモードの印字領域の横方向の始点をドット単位で指定します。

wStartY

縦方向始点

ページモードの印字領域の縦方向の始点をドット単位で指定します。

wWidth

横方向長さ

ページモードの印字領域の横方向の長さをドット単位で指定します。
1以上を指定してください。

wHeight

縦方向長さ

ページモードの印字領域の縦方向の長さをドット単位で指定します。
1以上を指定してください。

byDirection

印字方向

ページモードの印字方向を指定します。
無効な値を指定した場合、本APIは失敗(ERR_PARAM(-90))します。
印字方向の内容は『6.4 印字方向』を参照してください。

戻り値

成功時は0を返します。

失敗時はエラーコードを返します。詳細は『5章 エラーコード一覧』を参照してください。

メモ

- 横方向始点が印字幅を超える場合、または、縦方向始点が2400以上の場合、本APIは成功しますが、設定はプリンターの状態に依存します。
- 指定された領域が印字可能領域を超える場合、超えた値には印字可能領域が指定されます。

MpPrintPageModeData

ページモードデータを印字します。

INT **MpPrintPageModeData**(
DWORD *dwSessionId*)

パラメーター

dwSessionId

セッションID

MpOpenで取得したセッションIDを指定します。

戻り値

成功時は0を返します。

失敗時はエラーコードを返します。詳細は『5章 エラーコード一覧』を参照してください。

MpSetPageModeVerticalPosition

ページモードの縦方向の絶対位置を設定します。

INT **MpSetPageModeVerticalPosition**(
DWORD *dwSessionId*,
WORD *wVerticalPosition*)

パラメーター

dwSessionId
セッションID
MpOpenで取得したセッションIDを指定します。

wVerticalPosition
縦方向のデータ展開開始の絶対位置
ページモードの縦方向のデータ展開開始の絶対位置をドット単位で指定します。
MpSelectPageModeで指定したページモードの印字領域を超える値を指定した場合、APIは成功しますが、設定は無視されます。

戻り値

成功時は0を返します。

失敗時はエラーコードを返します。詳細は『5章 エラーコード一覧』を参照してください。

MpSetStandardModeArea

スタンダードモードの印字領域を設定します。

INT **MpSetStandardModeArea**(
DWORD *dwSessionId*,
WORD *wLeftMargin*,
WORD *wPrintAreaWidth*)

パラメーター

dwSessionId
セッションID
MpOpenで取得したセッションIDを指定します。

wLeftMargin
左マージン位置
左マージンの位置をドット単位で指定します。
1行の印字可能領域を超える値を入力した場合、印字可能領域の最大値を左マージンとして設定します。

wPrintAreaWidth

印字領域幅

印字領域幅をドット単位で指定します。
1行の印字可能領域を超える値を入力した場合、左マージンを除いた全領域を印字領域幅として設定します。

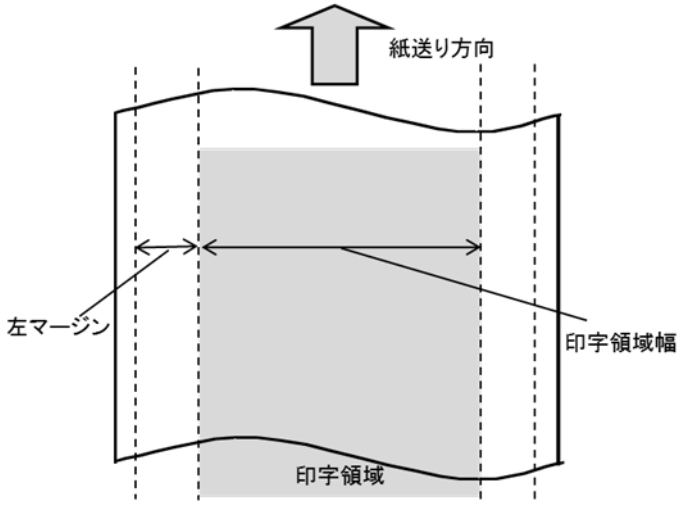
戻り値

成功時は0を返します。

失敗時はエラーコードを返します。詳細は『5章 エラーコード一覧』を参照してください。

メモ

- 印字データを送信する前に、本APIを実行してください。
- 左マージンの値は印字領域幅に含まれません。



MpSetStandardModeAlignment

スタンダードモードでの印字位置を設定します。

INT **MpSetStandardModeAlignment**(
DWORD *dwSessionId*,
BYTE *byAlign*)

パラメーター

dwSessionId

セッションID

MpOpenで取得したセッションIDを指定します。

byAlign

印字位置

スタンダードモードで印字する場合の印字位置を指定します。
無効な値を指定した場合、本APIは失敗(ERR_PARAM(-90))します。
印字位置の内容は『6.5 印字位置』を参照してください。

戻り値

成功時は0を返します。

失敗時はエラーコードを返します。詳細は『5章　エラーコード一覧』を参照してください。

メモ

- 印字データを送信する前に、本APIを実行してください。
- 行の途中で印字位置を変えることはできません。

MpSetHorizontalPosition

横方向の絶対位置を設定します。

INT **MpSetHorizontalPosition**(
DWORD *dwSessionId*,
WORD *wHorizontalPosition*)

パラメーター

dwSessionId
セッションID
MpOpenで取得したセッションIDを指定します。

wHorizontalPosition
横方向の印字開始位置
横方向の印字開始位置をドット単位で指定します。
MpSelectPageModeで指定したページモードの印字領域を超える値を指定した場合、APIは成功しますが、設定は無視されます。

戻り値

成功時は0を返します。

失敗時はエラーコードを返します。詳細は『5章　エラーコード一覧』を参照してください。

メモ

- スタンダードモードの場合、本APIは**MpSetStandardModeArea**で設定した左マージンの設定を基準として実行されます。

MpSetStandardModeBarcodeDirection

スタンダードモードのバーコードの印字方向を設定します。

INT **MpSetStandardModeBarcodeDirection**(
DWORD *dwSessionId*,
BYTE *byRotate*)

パラメーター

dwSessionId
セッションID
MpOpenで取得したセッションIDを指定します。

byRotate
印字方向
バーコードの印字方向を指定します。
90度右回転または90度左回転を指定した場合、印字可能なバーコードの横幅は最大300mmまでです。300mmを超える場合、バーコードの印字は行われません。
無効な値を指定した場合、本APIは失敗(ERR_PARAM(-90))します。
バーコードの印字方向は『6.13 回転印字』を参照してください。

戻り値

成功時は0を返します。

失敗時はエラーコードを返します。詳細は『5章 エラーコード一覧』を参照してください。

メモ

- ページモードの印字では使用しないでください。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。

MpSetLineSpacing

改行量を設定します。

INT **MpSetLineSpacing**(
DWORD *dwSessionId*,
BYTE *byLineSpacing*)

パラメーター

dwSessionId
セッションID
MpOpenで取得したセッションIDを指定します。

byLineSpacing
改行量
改行量を指定します。

戻り値

成功時は0を返します。

失敗時はエラーコードを返します。詳細は『5章 エラーコード一覧』を参照してください。

メモ

- 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設定が可能です。

MpSetCharacterRightSpace

文字の右スペース量を設定します。

INT **MpSetCharacterRightSpace**(
DWORD *dwSessionId*,
BYTE *bySpace*)

パラメーター

dwSessionId
セッションID
MpOpenで取得したセッションIDを指定します。

bySpace
文字の右スペース量
文字の右スペース量をドット単位で指定します。

戻り値

成功時は0を返します。

失敗時はエラーコードを返します。詳細は『5章 エラーコード一覧』を参照してください。

メモ

- 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設定が可能です。
- 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に設定されます。

MpSelectCharacterSet

文字セットを設定します。

INT **MpSelectCharacterSet**(
DWORD *dwSessionId*,
BYTE *byCharSet*)

パラメーター

dwSessionId
セッションID
MpOpenで取得したセッションIDを指定します。

byCharSet
文字セット
使用する文字セットを指定します。
無効な値を指定した場合、本APIは失敗(ERR_PARAM(-90))します。
文字セットの内容は『6.6 文字セット』を参照してください。

初期値

Windows OSの言語設定により異なります。

日本語の場合 : Katakana (CODEPAGE_KATAKANA)

日本語以外の場合 : Latin (CODEPAGE_1252)

戻り値

成功時は0を返します。

失敗時はエラーコードを返します。詳細は『5章 エラーコード一覧』を参照してください。

メモ

- **MpReset**によるリセット及び、プリンターの電源オフを行った場合でも、本APIで設定した内容は保持されます。
- 本APIで設定した内容は下記のAPI呼び出し時のプリンターの設定に使用されます。
  - **MpPrintText**
  - **MpSendDataFile**(テキストファイル指定)
- 本APIで文字セットにArabic (CODEPAGE_864)を指定した場合、**MpSetCharacterFormatting**の文字フォントの指定にかかわらず、文字フォントA(24×12)で印字されます。

MpSelectInternationalCharacterSet

国際文字セットを設定します。

INT **MpSelectInternationalCharacterSet**(
DWORD *dwSessionId*,
BYTE *byIntCharSet*)

パラメーター

dwSessionId
セッションID
MpOpenで取得したセッションIDを指定します。

byIntCharSet
国際文字セット
使用する国際文字セットを指定します。
無効な値を指定した場合、本APIは失敗(ERR_PARAM(-90))します。
国際文字セットの内容は『6.7 国際文字セット』を参照してください。

初期値

Windows OSの言語設定により異なります。

日本語の場合 : 日本 (INT_CHAR_SET_JAPAN)

日本語以外の場合 : アメリカ (INT_CHAR_SET_USA)

戻り値

成功時は0を返します。

失敗時はエラーコードを返します。詳細は『5章 エラーコード一覧』を参照してください。

メモ

- **MpReset**によるリセット及び、プリンターの電源オフを行った場合でも、本APIで設定した内容は保持されます。
- 本APIで設定した内容は下記のAPI呼び出し時のプリンターの設定に使用されます。
  - **MpPrintText**
  - **MpSendDataFile**(テキストファイル指定)

MpSetCharacterFormatting

文字修飾を設定します。

INT **MpSetCharacterFormatting**(
DWORD *dwSessionId*,
BYTE *byType*,
BYTE *byVerticalScale*,
BYTE *byHorizontalScale*,
BYTE *byUnderline*,
BYTE *byBold*,
BYTE *byReverse*,
BYTE *byRotate*)

パラメーター

dwSessionId
セッションID
MpOpenで取得したセッションIDを指定します。

byType
文字フォント
文字フォントを指定します。
『6.8 文字フォント』を参照してください。

byVerticalScale
文字サイズ（縦方向）
文字サイズ（縦方向）を指定します。
『6.9 文字サイズ』を参照してください。

byHorizontalScale
文字サイズ（横方向）
文字サイズ（横方向)を指定します。
『6.9 文字サイズ』を参照してください。

byUnderline
アンダーライン
アンダーラインを指定します。
『6.10 アンダーライン』を参照してください。

byBold
強調印字
強調印字を指定します。
『6.11 強調印字』を参照してください。

byReverse

白黒反転印字

白黒反転印字を指定します。
値が「CHAR_REVERSE_ON（2：白黒反転印字あり）」の場合、下記の動作となります。

- アンダーラインと強調印字は行われません。
- 文字の右スペース量も白黒反転印字の対象となります。

『6.12 白黒反転印字』を参照してください。

byRotate

回転印字

回転印字を指定します。
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。
値が「ROTATE_NONE（1：回転なし）」以外を指定する場合、印字データを送信する前に、本APIを実行してください。
値が「ROTATE_90_TO_RIGHT（2：90度右回転）」または、「ROTATE_90_TO_LEFT（4：90度左回転）」の場合、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。
- *byVerticalScale*の値は文字サイズ（横方向）、*byHorizontalScale*の値は文字サイズ（縦方向）となります。

『6.13 回転印字』を参照してください。

戻り値

成功時は0を返します。

失敗時はエラーコードを返します。詳細は『5章 エラーコード一覧』を参照してください。

メモ

- *byVerticalScale*または*byHorizontalScale*のいずれかの値が「CHAR_SCALE_CURRENT（0：現在の設定）」の場合、両方の値とも「CHAR_SCALE_CURRENT（0：現在の設定）」として動作します。
- *byType*, *byVerticalScale*, *byHorizontalScale*, *byUnderline*, *byBold*, *byReverse*, *byRotate*のいずれか1つでも無効な値を指定した場合、本APIは失敗（ERR_PARAM(-90)）します。

MpPrintText

文字列を印字します。

INT **MpPrintText**(
DWORD *dwSessionId*,
LPCTSTR *pszText*)

パラメーター

dwSessionId
セッションID
MpOpenで取得したセッションIDを指定します。

pszText
送信するテキストデータ
プリンターに送信するテキストデータを指定します。
指定する文字列の末尾は必ずNULLとしてください。
指定できる文字列のバイト数は、NULLを含めて最大65535バイトです。
末尾にNULLが指定されなかった場合、本APIは失敗します。

戻り値

成功時は0を返します。

失敗時はエラーコードを返します。詳細は『5章 エラーコード一覧』を参照してください。

メモ

- 本APIは**MpReset**により中断されます。
- 本API呼び出し時は送信するデータの先頭に下記のプリンターコマンドを送信します。
  - 国際文字の選択
  - 文字コードテーブルの選択
  - 漢字コード体系の選択
  - 漢字モード解除
- 漢字コード体系は、使用している文字コードテーブルによって異なります。

Windows OS	文字コードテーブル	漢字コード体系
日本語	Katakana	シフトJISコード
	Katakana以外	JISコード
日本語以外	Katakana	シフトJISコード
	Katakana以外	JISコード

- 本APIによる文字列のエンコードは、**MpSelectCharacterSet**で選択されたコードページでエンコードします。
- **MpControlTransaction**を使用しない場合、本APIの送信タイムアウト時間は、**MpGetWriteTimeout**で設定した値になります。

MpPrintLogo

プリンターに登録したロゴを印字します。

INT **MpPrintLogo**(
DWORD *dwSessionId*,
BYTE *byRegNum*)

パラメーター

dwSessionId
セッションID
MpOpenで取得したセッションIDを指定します。

byRegNum
ロゴ番号
印字するロゴ番号を指定します。
0から99までの値を指定してください。
範囲外の値を指定した場合、本APIは失敗(ERR_PARAM(-90))します。
指定したロゴ番号にロゴイメージデータが登録されていない場合、本APIは無視されます。

戻り値

成功時は0を返します。

失敗時はエラーコードを返します。詳細は『5章 エラーコード一覧』を参照してください。

MpSendDataFile

指定したファイルデータを送信します。

INT **MpSendDataFile**(
DWORD *dwSessionId*,
LPCTSTR *pszFilePath*)

パラメーター

dwSessionId
セッションID
MpOpenで取得したセッションIDを指定します。

pszFilePath

ファイルパス

送信するファイルを指定します。
下記のファイルを指定してください。

pszFilePath	内容
イメージファイル*1	Windowsビットマップファイル（*.bmp） JPEGファイル（*.jpg / *.jpeg）
テキストファイル	テキストドキュメント（*.txt）
バイナリデータファイル	バイナリデータファイル（*.bin、*.dat、*.prn）
HTMLファイル	HTMLドキュメント（*.htm、*.html）

*1: 本 API でイメージファイルを登録する場合は、ディザリング有効に固定されます。

戻り値

成功時は0を返します。

失敗時はエラーコードを返します。詳細は『5章 エラーコード一覧』を参照してください。

メモ

- 本APIは**MpReset**により中断されます。
- イメージファイルを指定した場合、イメージをプリンターコマンド化しラスタビットイメージ印字を行います。
- イメージの高さが4095ドット以上となるイメージファイルを指定した場合、本APIは失敗します。
- テキストファイルを指定した場合、ファイル内の文字列を印字します。
- テキストファイル、HTMLファイルはANSI形式及びUTF-8形式のファイルを使用できます。
- テキストファイルを指定して本APIを呼び出した場合、送信するデータの先頭に下記のプリンターコマンドを送信します。
  - 国際文字の選択
  - 文字コードテーブルの選択
  - 漢字コード体系の選択
  - 漢字モード解除
- 漢字コード体系は、使用している文字コードテーブルによって異なります。

Windows OS	文字コードテーブル	漢字コード体系
日本語	Katakana	シフトJISコード
	Katakana以外	JISコード
日本語以外	Katakana	シフトJISコード
	Katakana以外	JISコード

- テキストファイルを指定して本APIを呼び出した場合、文字列のエンコードは、**MpSelectCharacterSet**で選択されたコードページでエンコードします。
- HTMLファイルを指定して本APを呼び出した場合、送信するコマンドの先頭に下記のコマンドを送信します。
  - タグ処理開始
- バイナリデータファイルを指定した場合、ファイル内のデータの変換は行わずプリンターへ送信します。
- **MpControlTransaction**を使用しない場合、本APIの送信タイムアウト時間は、**MpGetWriteTimeout**で設定した値になります。

- 読み込み可能なファイルサイズは最大4294967295バイトです。
  ただしシステム上にメモリを確保できなかった場合、本APIは失敗します。
- 送信するデータにプリンターコマンドの「ハードウェアリセット」及び「プリンタリセット」を含めないでください。
  プリンターをリセットする場合、**MpReset**を使用してください。

MpSendDataFileEx

指定したファイルデータを送信します。

INT **MpSendDataFileEx**(
DWORD *dwSessionId*,
LPCTSTR *pszFilePath*,
BYTE *byOption*)

パラメーター

dwSessionId
セッションID
MpOpenで取得したセッションIDを指定します。

pszFilePath
ファイルパス
送信するファイルを指定します。
下記のファイルを指定してください。

pszFilePath	内容
イメージファイル	Windowsビットマップファイル（*.bmp） JPEGファイル（*.jpg / *.jpeg）
テキストファイル	テキストドキュメント（*.txt）
バイナリデータファイル	バイナリデータファイル（*.bin、*.dat、*.prn）
HTMLファイル	HTMLドキュメント（*.htm、*.html）

byOption
オプション
イメージファイル送信時のオプションを指定します。
下記の値を指定してください。

byOption	内容
0	ディザリング有効
1	ディザリング無効

戻り値

成功時は0を返します。

失敗時はエラーコードを返します。詳細は『5章　エラーコード一覧』を参照してください。

メモ

- 本APIは**MpReset**により中断されます。
- イメージファイルを指定した場合、イメージをプリンターコマンド化しラスタビットイメージ印字を行います。
- イメージの高さが4095ドット以上となるイメージファイルを指定した場合、本APIは失敗します。
- テキストファイルを指定した場合、ファイル内の文字列を印字します。
- テキストファイル、HTMLファイルはANSI形式及びUTF-8形式のファイルを使用できます。
- テキストファイルを指定して本APIを呼び出した場合、送信するデータの先頭に下記のプリンターコマンドを送信します。
  - 国際文字の選択
  - 文字コードテーブルの選択
  - 漢字コード体系の選択
  - 漢字モード解除
- 漢字コード体系は、使用している文字コードテーブルによって異なります。

Windows OS	文字コードテーブル	漢字コード体系
日本語	Katakana	シフトJISコード
	Katakana以外	JISコード
日本語以外	Katakana	シフトJISコード
	Katakana以外	JISコード

- テキストファイルを指定して本APIを呼び出した場合、文字列のエンコードは、**MpSelectCharacterSet**で選択されたコードページでエンコードします。
- HTMLファイルを指定して本APIを呼び出した場合、送信するコマンドの先頭に下記のコマンドを送信します。
  - タグ処理開始
- バイナリデータファイルを指定した場合、ファイル内のデータの変換は行わずプリンターへ送信します。
- **MpControlTransaction**を使用しない場合、本APIの送信タイムアウト時間は、**MpGetWriteTimeout**で設定した値になります。
- 読み込み可能なファイルサイズは最大4294967295バイトです。
  ただしシステム上にメモリを確保できなかった場合、本APIは失敗します。
- 送信するデータにプリンターコマンドの「ハードウェアリセット」及び「プリンタリセット」を含めないでください。
  プリンターをリセットする場合、**MpReset**を使用してください。

MpPrintBarcode

バーコードを印字します。

INT **MpPrintBarcode**(
DWORD *dwSessionId*,
BYTE *byType*,
LPBYTE *pData*,
DWORD *cbData*,
BYTE *byModuleWidth*,
BYTE *byModuleHeight*,
BYTE *byHRI*,
BYTE *byNWRatio*)

パラメーター

dwSessionId
セッションID
MpOpenで取得したセッションIDを指定します。

byType
バーコード種類
印字するバーコードの種類を指定します。
『6.14 バーコード種類』を参照してください。

pData
バーコードデータ
印字するバーコードデータを指定します。
文字列データはマルチバイトコードで指定してください。

cbData
バーコードデータサイズ
指定したバーコードデータのサイズを指定します。

byModuleWidth
モジュール幅または細エレメント
バーコードのモジュール幅を指定します。
『6.15 バーコードモジュール幅』を参照してください。

byModuleHeight
バーコード高さ
バーコードの高さをドット単位で指定します。
入力値の範囲はバーコード種類によって異なります。
『6.16 バーコード高さ』を参照してください。

byHRI
HRI文字
バーコードHRI文字の文字フォントと位置を指定します。
『6.17 バーコードHRI文字』を参照してください。

byNWRatio

N:W比

バーコードのN:W比を指定します。
下記のバーコード種類の場合に指定が必要です。次のバーコード種類以外は、0を指定してください。

- CODE39
- ITF
- CODABAR

『6.18 バーコードN:W比』を参照してください。

戻り値

成功時は0を返します。

失敗時はエラーコードを返します。詳細は『5章 エラーコード一覧』を参照してください。

メモ

- スタンダードモードでは、印字データを送信する前に、本APIを実行してください。
- *byType, byModuleWidth*, *byModuleHeight*, *byHRI*, *byNWRatio*のいずれか1つでも無効な値を指定した場合、本APIは失敗(ERR_PARAM(-90))します。
- 印字領域を超えた場合、バーコードは印字されません。
- バーコードデータの入力条件は下記の通りです。
  入力条件に合わない値を指定した場合、本APIは失敗(ERR_INVALID_DATA(-1020))します。

種別	データ数	使用可能なデータ
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	'0' ～ '9' (0x30 ～ 0x39)
		'A' ～ 'Z' (0x41 ～ 0x54)
		' ' (0x20)
		'$' (0x24)
		'%' (0x25)
		'+' (0x2B)
		'–' (0x2D)
		'.' (0x2E)
		'/' (0x2F)
ITF	2 ～ 150バイト (ただし偶数個)	'0' ～ '9' (0x30 ～ 0x39)

種別	データ数	使用可能なデータ
CODABAR	1 ～ 150バイト	'0' ～ '9' (0x30 ～ 0x39)
		'A' ～ 'D' (0x41 ～ 0x44)
		'$' (0x24)
		'+' (0x2B)
		'–' (0x2D)
		'.' (0x2E)
		'/' (0x2F)
		':' (0x3A)
CODE128	2バイト以上	(0x00 ～ 0x7F)※1
CODE93	1バイト以上	(0x00 ～ 0x7F)
JAN13 (EAN13) add-on 2	12 ～ 13 + 2バイト	'0' ～ '9' (0x30 ～ 0x39)
JAN13 (EAN13) add-on 5	12 ～ 13 + 5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バイト以上	' '～ '"' (0x20 ～ 0x22)
		'%' ～ '?' (0x25 ～ 0x3F)
		'A' ～ 'Z' (0x41 ～ 0x5A)
		'_' (0x5F)
		'a' ～ 'z' (0x61 ～ 0x7A)
		'{' (0x7B)

※1: CODEの選択によって使用できるデータが異なります。

- CODE A:0x00～0x5F
- CODE B:0x20～0x7F
- CODE C:00～99(0x00～0x63)の2桁の数字

下記の特殊キャラクターによりCODE、FNC、SHIFTの選択を行います。

コード	CODE A	CODE B	CODE C
{S	SHIFT	SHIFT	SHIFT
{A	–	CODE A	CODE A
{B	CODE B	–	CODE B
{C	CODE C	CODE C	–
{1	FNC1	FNC1	FNC1
{2	FNC2	FNC2	FNC2
{3	FNC3	FNC3	FNC3
{4	FNC4	FNC4	FNC4
{{	'{'	'{'	–

- 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 (EAN13) add-on 2
  - JAN13 (EAN13) add-on 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 (EAN13)
  - JAN8 (EAN8)
- 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、本APIは失敗します。
- 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、'{'は、FNCを指定する場合にのみ使用してください。FNC1を指定する場合、'{', '1'(0x7B, 0x31)を入力します。
- **MpControlTransaction**を使用しない場合、本APIの送信タイムアウト時間は、**MpGetWriteTimeout**で設定した値になります。

MpPrint2Dcode

2次元バーコードを印字します。

INT **MpPrint2Dcode**(
DWORD *dwSessionId*,
BYTE *byType*,
LPBYTE *pData*,
DWORD *cbData*,
BYTE *byMode*,
BYTE *byModuleSize*,
BYTE *byModuleHeight*,
BYTE *byColumn*,
BYTE *byRow*,
BYTE *byErrCorrect*)

パラメーター

dwSessionId

セッションID

MpOpenで取得したセッションIDを指定します。

byType

2次元バーコード種類

印字する2次元バーコードの種類を指定します。
『6.19 2次元バーコード種類』を参照してください。

pData

2次元バーコードデータ

印字するバーコードデータを指定します。
文字列データはマルチバイトコードで指定してください。

cbData

2次元バーコードデータサイズ

指定したバーコードデータのサイズを指定します。

byMode

各2次元バーコードのモード

2次元バーコードのモードを指定します。
下記の2次元バーコードの種類の場合に指定が必要です。記載のない2次元バーコード種類を使用する場合、0を指定してください。

- QR Code
- PDF417
- Data Matrix
- Maxi Code

2次元バーコードの種類により入力値の範囲が異なります。
『6.20 2次元バーコードモード』を参照してください。

byModuleSize

2次元バーコードのモジュールサイズ

2次元バーコードのモジュールサイズを指定します。
下記の2次元バーコードの種類の場合に指定が必要です。記載のない2次元バーコード種類を使用する場合、0を指定してください。

- QR Code
- PDF417
- Data Matrix
- GS1 Databar Stacked
- GS1 Databar Stacked Omni-directional
- GS1 Databar Expanded Stacked

2次元バーコードの種類により入力値の範囲が異なります。
『6.21 2次元バーコードモジュールサイズ』を参照してください。

byModuleHeight

2次元バーコードのモジュール高さ

2次元バーコードのモジュール高さを指定します。
下記の2次元バーコードの種類の場合に指定が必要です。記載のない2次元バーコード種類を使用する場合、0を指定してください。

- PDF417
- GS1 Databar Stacked Omni-directional

2次元バーコードの種類により入力値の範囲が異なります。
『6.22 2次元バーコードモジュール高さ』を参照してください。

byColumn

2次元バーコードのカラム数または1行のエレメント数

2次元バーコードのカラム数または1行のエレメント数を指定します。
下記の2次元バーコードの種類の場合に指定が必要です。記載のない2次元バーコード種類を使用する場合、0を指定してください。

- PDF417
- GS1 Databar Expanded Stacked

2次元バーコードの種類により入力値の範囲が異なります。
『6.23 2次元バーコードカラム数』を参照してください。

byRow

2次元バーコードの段数

2次元バーコードの段数を指定します。
下記の2次元バーコードの種類の場合に指定が必要です。記載のない2次元バーコード種類を使用する場合、0を指定してください。

- PDF417

『6.24 2次元バーコード段数』を参照してください。

byErrCorrect

2次元バーコードのエラー訂正レベル

2次元バーコードのエラー訂正レベルを指定します。
下記の2次元バーコードの種類の場合に指定が必要です。記載のない2次元バーコード種類を使用する場合、0を指定してください。

- QR Code
- PDF417

2次元バーコードの種類により入力値の範囲が異なります。
『6.25 2次元バーコードエラー訂正レベル』を参照してください。

戻り値

成功時は0を返します。

失敗時はエラーコードを返します。詳細は『5章 エラーコード一覧』を参照してください。

メモ

- *byType*, *byMode*, *byModuleSize*, *byModuleHeight*, *byColumn*, *byRow*, *byErrCorrect*のいずれか1つでも無効な値を指定した場合、本APIは失敗(ERR_PARAM(-90))します。
- 印字領域を超えた場合、バーコードは印字されません。
- **MpControlTransaction**を使用しない場合、本APIの送信タイムアウト時間は、**MpGetWriteTimeout**で設定した値になります。

MpPrintPageModeRectangle

ページモード選択時に、矩形を印字します。

INT **MpPrintPageModeRectangle**(
DWORD *dwSessionId*,
WORD *wStartX*,
WORD *wStartY*,
WORD *wWidth*,
WORD *wHeight*,
BYTE *byThickness*)

パラメーター

dwSessionId

セッションID

MpOpenで取得したセッションIDを指定します。

wStartX

横方向始点

矩形の横方向の始点をドット単位で指定します。

wStartY

縦方向始点

矩形の縦方向の始点をドット単位で指定します。

wWidth

横方向長さ

矩形の横方向の長さをドット単位で指定します。
値は1以上を指定してください。

wHeight

縦方向長さ

矩形の縦方向長さをドット単位で指定します。
値は1以上を指定してください。

byThickness

線幅

矩形の線幅をドット単位で指定します。
設定範囲は2～40です。
2未満の値を指定した場合は2に、40を超える値を指定した場合は40に設定されます。

戻り値

成功時は0を返します。

失敗時はエラーコードを返します。詳細は『5章　エラーコード一覧』を参照してください。

メモ

- 本APIをスタンダードモードで使用した場合、動作は不定となります。
- 矩形の印字方向は、**MpSelectPageMode**の設定に依存します。
- 本APIを使用する場合、自動的にプリンターコマンドの「罫線クリア」及び「罫線OFF」を実行します。

MpFeedLine

行単位で紙送りをします。

INT **MpFeedLine**(
DWORD *dwSessionId*,
BYTE *byLines*)

パラメーター

dwSessionId
セッションID
MpOpenで取得したセッションIDを指定します。

byLines
行単位の紙送り量
紙送量を行単位で指定します。
設定範囲は0～255です。

戻り値

成功時は0を返します。

失敗時はエラーコードを返します。詳細は『5章 エラーコード一覧』を参照してください。

MpFeedDotLine

ドット単位で紙送りをします。

INT **MpFeedDotLine**(
DWORD *dwSessionId*,
INT *nDotLines*)

パラメーター

dwSessionId
セッションID
MpOpenで取得したセッションIDを指定します。

nDotLines
ドット単位の紙送り量
紙送量をドット単位で指定します。
設定範囲は-48～8192です。
-48未満の値を指定した場合は-48に、8192を超える値を指定した場合は8192に設定されます。

戻り値

成功時は0を返します。

失敗時はエラーコードを返します。詳細は『5章　エラーコード一覧』を参照してください。

メモ

- ページモードで、本APIを*nDotLines*に負の値を指定した場合、紙送りはされません。

MpFeedCutPosition

カット位置まで紙送りをします。

INT **MpFeedCutPosition**(
DWORD *dwSessionId*)

パラメーター

dwSessionId
セッションID
MpOpenで取得したセッションIDを指定します。

戻り値

成功時は0を返します

失敗時はエラーコードを返します。詳細は『5章　エラーコード一覧』を参照してください。

メモ

- スタンダードモードでは、印字データを送信する前に、本APIを実行してください。

MpFeedMarkPosition

マークによる頭出しをします。

INT **MpFeedMarkPosition**(
DWORD *dwSessionId*,
INT *nDotLines*)

パラメーター

dwSessionId
セッションID
MpOpenで取得したセッションIDを指定します。

nDotLines

補正量

補正量をドット単位で指定します。
設定範囲は-48～255です。
-48未満の値を指定した場合は-48に、255を超える値を指定した場合は255に設定されます。補正を行わない場合は0を指定してください。

戻り値

成功時は0を返します。

失敗時はエラーコードを返します。詳細は『5章 エラーコード一覧』を参照してください。

メモ

- マーク紙の頭出し、及び頭出し位置の補正を行います。
- スタンダードモードでは、印字データを送信する前に、本APIを実行してください。

4章　.NET API

4.1 概要

本章では.NET開発環境用SDK(.NET API)について説明します。

4.2 開発対象言語

開発対象言語は下記の通りです。

- Visual C#
- Visual Basic

4.3 ライブラリファイル

ライブラリファイルのファイル名は下記の通りです。

- SiiMpa4ClassLib.dll

ライブラリファイルの保管場所は、グローバル・アセンブリ・キャッシュ(GAC)フォルダーです。

4.4 API一覧

SDKに実装されているAPIは下記の通りです。

- ネームスペース：SII.SPS.CommunicationLibrary.MobilePrinter

4.4.1 PrinterManager クラス

種別	API	機能概略
プロパティ	IsOpened	オープン状態を取得します。
メソッド	Open	指定したプリンターに対しSDKの使用を開始します。
メソッド	Close	SDKの使用を終了します。
メソッド	SetWriteTimeout	データ送信を行うAPIのタイムアウト値を設定します。
メソッド	GetWriteTimeout	データ送信を行うAPIのタイムアウト値を取得します。
メソッド	SetResponseTimeout	データ受信を行うAPIのタイムアウト値を設定します。
メソッド	GetResponseTimeout	データ受信を行うAPIのタイムアウト値を取得します。
メソッド	Write	プリンターへバイナリデータを送信します。
メソッド	Read	プリンターから受信したデータを取得します。
メソッド	GetReadSize	現在Readメソッドで取得できるデータサイズを取得します。
メソッド	WriteAndWaitResponse	バイナリデータの送信後、応答データを受信します。
メソッド	Reset	プリンターをリセットします。
メソッド	GetStatus	最新のプリンターステータスを取得します。
メソッド	StartCallbackFunction	プリンターステータスの変化を検知した際に呼び出されるコールバック関数を登録します。
メソッド	StopCallbackFunction	プリンターステータス変化を検出した際のコールバックを終了します
メソッド	RegisterStyleSheet	プリンターにスタイルシートを登録します。
メソッド	DeleteStyleSheet	プリンターからスタイルシートを削除します。
メソッド	RegisterLogo	プリンターにロゴを登録します。
メソッド	DeleteLogo	プリンターから、登録済みのロゴを削除します。
メソッド	GetPrinterInformation	指定したプリンター情報を取得します。
メソッド	ControlTransaction	対象のAPIを一括で処理します。
メソッド	SelectStandardMode	スタンダードモードを開始します。

種別	API	機能概略
メソッド	SelectPageMode	ページモードの印字領域と印字方向を指定し、ページモードを開始します。
メソッド	PrintPageModeData	ページモードデータを印字します。
メソッド	SetPageModeVerticalPosition	ページモードの縦方向の絶対位置を設定します。
メソッド	SetStandardModeArea	スタンダードモードの印字領域を設定します。
メソッド	SetStandardModeAlignment	スタンダードモードでの印字位置を設定します。
メソッド	SetHorizontalPosition	絶対位置を設定します。
メソッド	SetStandardModeBarcodeDirection	バーコードの印字方向を設定します。
メソッド	SetLineSpacing	改行量を設定します。
メソッド	SetCharacterRightSpace	文字の右スペース量を設定します。
メソッド	SelectCharacterSet	文字セットを設定します。
メソッド	SelectInternationalCharacterSet	国際文字セットを設定します。
メソッド	SetCharacterFormatting	文字修飾を設定します。
メソッド	PrintText	文字列を印字します。
メソッド	PrintLogo	プリンターに登録されたイメージを印字します。
メソッド	SendDataFile	指定したファイルのデータを送信します。
メソッド	PrintBarcode	バーコードを印字します。
メソッド	Print2Dcode	2次元バーコードを印字します。
メソッド	PrintPageModeRectangle	ページモード選択時に、矩形を印字します。
メソッド	FeedLine	行単位で紙送りをします。
メソッド	FeedDotLine	ドット単位で紙送りをします。
メソッド	FeedCutPosition	カット位置まで紙送りをします。
メソッド	FeedMarkPosition	マーク紙の頭出しによる紙送りをします。
イベント	statusChanged	応答されたプリンターステータスを通知するイベントです。

4.4.2 PrinterException クラス

種別	API	機能概略
メソッド	GetErrorCode	エラーコードを取得します。

4.5 API詳細(PrinterManagerクラス)

4.5.1 プロパティ

IsOpened

Openの呼び出し状態を取得します。

bool **IsOpened** { get; }

初期値

FALSE

備考

SDKのオープン状態を取得します。

- TRUE : Openが呼び出し済み状態です。
- FALSE : Openが呼び出されていない状態です。

4.5.2 メソッド

Open

指定したプリンターに対しSDKの使用を開始します。

void **Open**(
string *prnName*)

パラメーター

prnName
SDKを使用するプリンター名
出力するプリンター名（フレンドリネーム）を指定します。

戻り値

なし

備考

- 1プロセスで1インスタンスのみオープンできます。
- SDKを使用しなくなった場合は、必ず**Close**を呼び出してください。
- プリンタードライバーの接続先がUSB、Bluetooth、またはTCP/IP以外の場合は、本APIは失敗します。
- 本API呼び出し時にエラーが発生した場合、**PrinterException**をスローします。

Close

SDKの使用を終了します。

void **Close**()

パラメーター

なし

戻り値

なし

備考

- 本API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章 Win32 API』の**MpClose**を参照してください。

SetWriteTimeout

データ送信を行うAPIのタイムアウト値を設定します。

void **SetWriteTimeout**(
uint *timeout*)

パラメーター

timeout
データ送信タイムアウト値
データ送信を行うAPIの送信待ち時間(ミリ秒単位)を指定します。

戻り値

なし

備考

- 本API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章 Win32 API』の**MpGetWriteTimeout**を参照してください。

GetWriteTimeout

データ送信を行うAPIのタイムアウト値を取得します。

uint **GetWriteTimeout**()

パラメーター

なし

戻り値

データ送信タイムアウト値
データ送信を行うAPIの送信待ち時間(ミリ秒単位)を取得します。

備考

- 本API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章 Win32 API』の**MpGetWriteTimeout**を参照してください。

SetResponseTimeout

受信を行うAPIのタイムアウト値を設定します。

void **SetResponseTimeout**(
uint *timeout*)

パラメーター

timeout
データ受信タイムアウト値
データ受信を行うAPIの受信待ち時間（ミリ秒単位）を指定します。

戻り値

なし

備考

- 本API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章 Win32 API』の**MpSetResponseTimeout**を参照してください。

GetResponseTimeout

受信を行うAPIのタイムアウト値を取得します。

uint **GetResponseTimeout**()

パラメーター

なし

戻り値

データ受信タイムアウト値
データ受信を行うAPIの受信待ち時間（ミリ秒単位）を取得します。

備考

- 本API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章 Win32 API』の**MpGetResponseTimeout**を参照してください。

Write

プリンターへバイナリデータを送信します。

void **Write**(
byte[] *cmd*,
uint *offset*)

パラメーター

cmd
送信データ
プリンターへ送信するデータを指定します。

offset
送信データオフセット
送信するデータの開始位置を指定します。

戻り値

なし

備考

- 本API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章 Win32 API』の**MpWrite**を参照してください。

Read

プリンターから受信したデータを取得します。

byte[] **Read**()

パラメーター

なし

戻り値

受信データ
プリンターから受信したデータを取得します。

備考

- 本API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章 Win32 API』の**MpRead**を参照してください。

GetReadSize

Readメソッドで取得可能なデータサイズを取得します。

uint **GetReadSize**()

パラメーター

なし

戻り値

取得可能データサイズ
Readメソッドで取得可能な受信データサイズを取得します。

備考

- 本API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章 Win32 API』の**MpRead**を参照してください。

WriteAndWaitResponse

バイナリデータの送信後、応答データを受信します。

byte[] **WriteAndWaitResponse**(
byte[] *cmd*,
uint *respSize*,
bool *respFlag*,
bool *incASBData*)

パラメーター

cmd
送信データ
プリンターへ送信するデータを指定します。

respSize

最大受信データサイズ

受信するデータサイズの最大値を指定します。

respFlag

受信動作フラグ

受信動作を指定します。

TRUE : 何らかのデータを受信するか、またはタイムアウトになるまで受信を続けます。

FALSE : 最大受信データサイズ分を受信するか、またはタイムアウトになるまで受信を続けます。

incASBData

受信データにASB設定コマンドの応答を含めるかを指定します。

受信対象のデータを指定します。

TRUE : ASB設定コマンドの応答を含めたデータを取得します。

FALSE : ASB設定コマンドの応答を除いたデータを取得します。

戻り値

受信データ

プリンターから受信したデータを取得します。

備考

- 本API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章 Win32 API』の**MpWriteAndWaitResponse**を参照してください。

Reset

プリンターをリセットします。

void **Reset**()

パラメーター

なし

戻り値

なし

備考

- 本API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章 Win32 API』の**MpReset**を参照してください。

GetStatus

最新のプリンターステータスを取得します。

PrinterStatus **GetStatus**()

パラメーター

なし

戻り値

プリンターステータス
最新のプリンターステータスを取得します。

備考

- 「PrinterStatus」の詳細は『6.1 プリンターステータス一覧』を参照してください。
- 本API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章 Win32 API』の**MpGetStatus**を参照してください。

StartCallbackFunction

プリンターステータスの変化を検知した際に呼び出されるコールバック関数を登録します。

void **StartCallbackFunction**()

パラメーター

なし

戻り値

なし

備考

- 本API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章 Win32 API』の**MpSetCallbackFunction**を参照してください。

StopCallbackFunction

プリンターステータス変化を検出した際のコールバックを終了します

void **StopCallbackFunction**()

パラメーター

なし

戻り値

なし

備考

- 本API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章 Win32 API』の**MpSetCallbackFunction**を参照してください。

RegisterStyleSheet

プリンターにスタイルシートを登録します。

void **RegisterStyleSheet**(
string *filePath*,
byte *regNum*)

パラメーター

filePath
スタイルシートのファイルパス
プリンターに登録するスタイルシートのファイルパスを指定します。

regNum
スタイルシート番号
プリンターに登録するスタイルシートの番号を指定します。

戻り値

なし

備考

- 本API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章 Win32 API』の**MpRegisterStyleSheet**を参照してください。

DeleteStyleSheet

プリンターから、登録済みのスタイルシートを削除します。

void **DeleteStyleSheet**(
byte *regNum*)

パラメーター

regnum
スタイルシート番号
プリンターから削除するスタイルシートの番号を指定します。

戻り値

なし

備考

- 本API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章 Win32 API』の**MpDeleteStyleSheet**を参照してください。

RegisterLogo

プリンターにロゴを登録します。

void **RegisterLogo**(
string *filePath*,
byte *regNum*)

パラメーター

filePath
ロゴイメージデータのファイルパス
プリンターに登録するロゴのファイルパスを指定します。
本APIでロゴイメージデータを登録する場合は、ディザリング有効に固定されます。

regNum
ロゴ番号
プリンターに登録するロゴ番号を指定します。

戻り値

なし

備考

- 本API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章 Win32 API』の**MpRegisterLogo**を参照してください。

RegisterLogoEx

プリンターにロゴを登録します。

void **RegisterLogoEx**(
string *filePath*,
byte *regNum*,
byte *option*)

パラメーター

filePath
ロゴイメージデータのファイルパス
プリンターに登録するロゴのファイルパスを指定します。

regNum
ロゴ番号
プリンターに登録するロゴ番号を指定します。

option
オプション
プリンターに登録する際のオプションを指定します。
下記の値を指定してください。

option	内容
0	ディザリング有効
1	ディザリング無効

戻り値

なし

備考

- 本API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章 Win32 API』の**MpRegisterLogoEx**を参照してください。

DeleteLogo

プリンターから、登録済みのロゴを削除します。

void **DeleteLogo**(
byte *regNum*)

パラメーター

regNum
ロゴ番号
プリンターから削除するロゴ番号を指定します。

戻り値

なし

備考

- 本API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章 Win32 API』の**MpDeleteLogo**を参照してください。

GetPrinterInformation

指定したプリンター情報を取得します。

byte[] **GetPrinterInformation**(
PrinterInformation *prnInfo*,
ref uint *data*)

パラメーター

prnInfo
プリンター情報
取得するプリンター情報の種類を指定します。
「PrinterInformation」に定義された値を指定します。

data
受信データ(数値)
*prnInfo*の応答形式が数値の場合、バイト配列のデータとは別に、uintで*data*にも値が格納されます。

戻り値

受信データ(バイナリ)
受信したフォント情報を取得します。

備考

- 「PrinterInformation」の詳細は『6.2 プリンター情報』を参照してください。
- 本API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章 Win32 API』の**MpGetPrinterInformation**を参照してください。

ControlTransaction

対象のAPIを一括で処理します。

void **ControlTransaction**(
TransactionFunction *control*)

パラメーター

control
制御方法
一括処理の制御方法を指定します。
「TransactionFunction」に定義された値を指定します。

戻り値

なし

備考

- 「TransactionFunction」の詳細は『6.3 一括処理制御方法』を参照してください。
- 本API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章 Win32 API』の**MpControlTransaction**を参照してください。

SelectStandardMode

スタンダードモードを開始します。

void **SelectStandardMode**()

パラメーター

なし

戻り値

なし

備考

- 本API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章 Win32 API』の**MpSelectStandardMode**を参照してください。

SelectPageMode

ページモードの印字領域と印字方向を指定し、ページモードを開始します。

void **SelectPageMode**(
ushort *startX*,
ushort *startY*,
ushort *width*,
ushort *height*,
Direction *direction*)

パラメーター

startX
横方向始点
ページモードの印字領域の横方向の始点をドット単位で指定します。

startY
縦方向始点
ページモードの印字領域の縦方向の始点をドット単位で指定します。

width
横方向長さ
ページモードの印字領域の横方向の長さをドット単位で指定します。

height
縦方向長さ
ページモードの印字領域の縦方向の長さをドット単位で指定します。

direction
印字方向
ページモードの印字方向を指定します。
「Direction」に定義された値を指定します。

戻り値

なし

備考

- 「Direction」の詳細は『6.4 印字方向』を参照してください。
- 本API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章 Win32 API』の**MpSelectPageMode**を参照してください。

PrintPageModeData

ページモードデータを印字します。

void **PrintPageModeData**()

パラメーター

なし

戻り値

なし

備考

- 本API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章 Win32 API』の**MpPrintPageModeData**を参照してください。

SetPageModeVerticalPosition

ページモードの縦方向の絶対位置を設定します。

void **SetPageModeVerticalPosition**(
ushort *verticalPosition*)

パラメーター

verticalPosition
縦方向のデータ展開開始の絶対位置
ページモードの縦方向のデータ展開開始の絶対位置をドット単位で指定します。

戻り値

なし

備考

- 本API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章 Win32 API』の**MpSetPageModeVerticalPosition**を参照してください。

SetStandardModeArea

スタンダードモードの印字領域を設定します。

void **SetStandardModeArea**(
ushort *leftMargin*,
ushort *printAreaWidth*)

パラメーター

leftMargin
左マージン位置
左マージンの位置をドット単位で指定します。

printAreaWidth
印字領域幅
印字領域幅をドット単位で指定します。

戻り値

なし

備考

- 本API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章 Win32 API』の**MpSetStandardModeArea**を参照してください。

SetStandardModeAlignment

スタンダードモードでの印字位置を設定します。

void **SetStandardModeAlignment**(
Alignment *align*)

パラメーター

align
印字位置
スタンダードモードで印字する場合の印字位置を指定します
「Alignment」に定義された値を指定します。

戻り値

なし

備考

- 「Alignment」の詳細は『6.5 印字位置』を参照してください。
- 本API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章 Win32 API』の**MpSetStandardModeAlignment**tを参照してください。

SetHorizontalPosition

絶対位置を設定します。

void **SetHorizontalPosition**(
ushort *horizontalPosition*)

パラメーター

horizontalPosition
横方向の印字開始位置
横方向の印字開始位置をドット単位で指定します。

戻り値

なし

備考

- 本API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章 Win32 API』の**MpSetHorizontalPosition**を参照してください。

SetStandardModeBarcodeDirection

バーコードの印字方向を設定します。

void **SetStandardModeBarcodeDirection**(
Rotate *rotate*)

パラメーター

rotate
印字方向
バーコードの印字方向を指定します。
「Rotate」に定義された値を指定します。

戻り値

なし

備考

- 「Rotate」の詳細は『6.13 回転印字』を参照してください。
- 本API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章 Win32 API』の**MpSetStandardModeBarcodeDirection**を参照してください。

SetLineSpacing

改行量を設定します。

void **SetLineSpacing**(
byte *lineSpacing*)

パラメーター

lineSpacing
改行量
改行量をドット単位で指定します。

戻り値

なし

備考

- 本API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章 Win32 API』の**MpSetLineSpacing**を参照してください。

SetCharacterRightSpace

文字の右スペース量を設定します。

void **SetCharacterRightSpace**(
byte *space*)

パラメーター

space
文字の右スペース量
文字の右スペース量をドット単位で指定します。

戻り値

なし

備考

- 本API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章 Win32 API』の**MpSetCharacterRightSpace**を参照してください。

SelectCharacterSet

文字セットを設定します。

void **SelectCharacterSet**(
CharacterSet *charSet*)

パラメーター

charSet
文字セット
文字セットを指定します。
「CharacterSet」に定義された値を指定します。

戻り値

なし

備考

- 「CharacterSet」の詳細は『6.6 文字セット』を参照してください。
- 本API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章 Win32 API』の**MpSelectCharacterSet**を参照してください。

SelectInternationalCharacterSet

国際文字セットを設定します。

void **SelectInternationalCharacterSet**(
InternationalCharacterSet *intCharSet*)

パラメーター

intCharSet
国際文字セット
使用する国際文字セットを指定します。
「InternationalCharacterSet」に定義された値を指定します。

戻り値

なし

備考

- 「InternationalCharacterSet」の詳細は『6.7 国際文字セット』を参照してください。
- 本API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章 Win32 API』の**MpSelectInternationalCharacterSet**を参照してください。

SetCharacterFormatting

文字修飾を設定します。

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

パラメーター

type
文字フォント
文字フォントを指定します。
「CharacterType」に定義された値を指定します。

verticalScale
文字サイズ（縦方向）
文字サイズ（縦方向）を指定します。
「CharacterScale」に定義された値を指定します。

horizontalScale
文字サイズ（横方向）
文字サイズ（横方向）を指定します。
「CharacterScale」に定義された値を指定します。

underline
アンダーライン
アンダーラインを指定します。
「Underline」に定義された値を指定します。

bold
強調印字
強調印字を指定します。
「Bold」に定義された値を指定します。

reverse
白黒反転印字
白黒反転印字を指定します。
「Reverse」に定義された値を指定します。

rotate
回転印字
回転印字を指定します。
「Rotate」に定義された値を指定します。

戻り値

なし

備考

- 「CharacterType」の詳細は『6.8 文字フォント』を参照してください。
- 「CharacterScale」の詳細は『6.9 文字サイズ』を参照してください。

- 「Underline」の詳細は『6.10 アンダーライン』を参照してください。
- 「Bold」の詳細は『6.11 強調印字』を参照してください。
- 「Reverse」の詳細は『6.12 白黒反転印字』を参照してください。
- 「Rotate」の詳細は『6.13 回転印字』を参照してください。
- 本API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章 Win32 API』の**MpSetCharacterFormatting**を参照してください。

PrintText

文字列を印字します。

void **PrintText**(
string *text*)

パラメーター

text
送信するテキストデータ
プリンターに送信するテキストデータを指定します。

戻り値

なし

備考

- 本API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章 Win32 API』の**MpPrintText**を参照してください。

PrintLogo

プリンターに登録されたイメージを印字します。

void **PrintLogo**(
byte *regNum*)

パラメーター

regNum
ロゴ番号
印字するプリンターに登録されたロゴ番号を指定します。

戻り値

なし

備考

- 本API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章 Win32 API』の**MpPrintLogo**を参照してください。

SendDataFile

指定したファイルのデータを送信します。

void **SendDataFile**(
string *filePath*)

パラメーター

filePath
ファイルパス
送信するデータを含んだファイルを指定します。
本APIでイメージファイルを登録する場合は、ディザリング有効に固定されます。

戻り値

なし

備考

- 本API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章 Win32 API』の**MpSendDataFile**を参照してください。

SendDataFileEx

指定したファイルのデータを送信します。

void **SendDataFileEx**(
string *filePath,*
byte *option*)

パラメーター

filePath
ファイルパス
送信するデータを含んだファイルを指定します。

option

オプション

プリンターに送信する際のオプションを指定します。
下記の値を指定してください。

option	内容
0	ディザリング有効
1	ディザリング無効

戻り値

なし

備考

- 本API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章 Win32 API』の**MpSendDataFileEx**を参照してください。

PrintBarcode

バーコードを印字します。

void **PrintBarcode**(
TypeBarcode *type*,
byte[] *data*,
ModuleWidthBarcode *moduleWidth*,
byte *moduleHeight*,
HriPositionBarcode *hri*,
NwRatioBarcode *nwRatio*)

パラメーター

type

バーコード種類

印字するバーコードの種類を指定します。
「TypeBarcode」に定義された値を指定します。
詳細は『6.14 バーコード種類』を参照してください。

data

バーコードデータ

印字するバーコードデータを指定します。

moduleWidth

モジュール幅または細エレメント

バーコードのモジュール幅を指定します。
「ModuleWidthBarcode」に定義された値を指定します。
詳細は『6.15 バーコードモジュール幅』を参照してください。

moduleHeight

バーコード高さ

バーコードのモジュール高さをドット単位で指定します。
詳細は『6.16 バーコード高さ』を参照してください。

hri

HRI文字

バーコードHRI文字の文字フォントと位置を指定します。
「HriPositionBarcode」に定義された値を指定します。
詳細は『6.17 バーコードHRI文字』を参照してください。

nwRatio

N:W比

バーコードのN:W比を指定します。
「NwRatioBarcode」に定義された値を指定します。
詳細は『6.18 バーコードN:W比』を参照してください。

戻り値

なし

備考

- 本API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章 Win32 API』の**MpPrintBarcode**を参照してください。

Print2Dcode

2次元バーコードを印字します。

void **Print2Dcode**(
Type2Dcode *type*,
byte[] *data*,
Mode2Dcode *mode*,
ModuleSize2Dcode *moduleSize*,
byte *moduleHeight*,
byte *column*,
byte *row*,
ErrorCorrect2Dcode *errCorrect*)

パラメーター

type

2次元バーコード種類

印字する2次元バーコードの種類を指定します。
「Type2Dcode」に定義された値を指定します。
詳細は『6.19 2次元バーコード種類』を参照してください。

data

2次元バーコードデータ

印字する2次元バーコードデータを指定します。

mode

2次元バーコードのモード

2次元バーコードのモードを指定します。
「Mode2Dcode」に定義された値を指定します。
詳細は『6.20 2次元バーコードモード』を参照してください。

moduleSize

2次元バーコードのモジュールサイズ

2次元バーコードのモジュールサイズを指定します。
「ModuleSize2Dcode」に定義された値を指定します。
詳細は『6.21 2次元バーコードモジュールサイズ』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

2次元バーコードのモジュール高さを指定します。
詳細は『6.22 2次元バーコードモジュール高さ』を参照してください。

column

2次元バーコードのカラム数または1行のエレメント数

2次元バーコードのカラム数または1行のエレメント数を指定します。
詳細は『6.23 2次元バーコードカラム数』を参照してください。

row

2次元バーコードの段数

2次元バーコードの段数を指定します。
詳細は『6.24 2次元バーコード段数』を参照してください。

errCorrect

エラー訂正レベル

2次元バーコードのエラー訂正レベルを指定します。
「ErrorCorrect2Dcode」に定義された値を指定します。
詳細は『6.25 2次元バーコードエラー訂正レベル』を参照してください。

戻り値

なし

備考

- 本API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章 Win32 API』の**MpPrint2Dcode**を参照してください。

PrintPageModeRectangle

ページモード選択時に、矩形を印字します。

void **PrintPageModeRectangle**(
ushort *startX*,
ushort *startY*,
ushort *width*,
ushort *height*,
byte *thickness*)

パラメーター

startX
横方向始点
矩形の横方向の始点をドット単位で指定します。

startY
縦方向始点
矩形の縦方向の始点をドット単位で指定します。

width
横方向長さ
矩形の横方向長さをドット単位で指定します。

height
縦方向長さ
矩形の縦方向長さをドット単位で指定します。

thickness
線幅
矩形の線幅をドット単位で指定します。

戻り値

なし

備考

- 本API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章 Win32 API』の**MpPrintPageModeRectangle**を参照してください。

FeedLine

行単位で、指定量の紙送りを行います。

void **FeedLine**(
byte *lines*)

パラメーター

lines
行単位の紙送り量
紙送量を行単位で指定します。

戻り値

なし

備考

- 本API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章 Win32 API』の**MpFeedLine**を参照してください。

FeedDotLine

ドット単位で、指定量の紙送りを行います。

void **FeedDotLine**(
int *dotLines*)

パラメーター

dotLines
ドット単位の紙送り量
紙送量をドット単位で指定します。

戻り値

なし

備考

- 本API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章 Win32 API』の**MpFeedDotLine**を参照してください。

FeedCutPosition

カット位置まで紙送りをします。

void **FeedCutPosition**()

パラメーター

なし

戻り値

なし

備考

- 本API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章 Win32 API』の**MpFeedCutPosition**を参照してください。

FeedMarkPosition

マークによる頭出しをします。

void **FeedMarkPosition**(
int *dotLines*)

パラメーター

dotLines
補正量
補正量をドット単位で指定します。

戻り値

なし

備考

- 本API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章 Win32 API』の**MpFeedMarkPosition**を参照してください。

4.5.3 イベント

statusChanged

応答されたプリンターステータスを通知します。

event StatusCallbackHandler **statusChanged**

delegate void **StatusCallbackHandler**(PrinterStatus *status*)

パラメーター

status
プリンターステータス
プリンターステータスを格納する変数を指定します。

備考

- 「PrinterStatus」の詳細は『6.1 プリンターステータス一覧』を参照してください。
- 詳細は『3章 Win32 API』の**MpSetCallbackFunction**を参照してください。

4.6 API詳細(PrinterExceptionクラス)

4.6.1 メソッド

GetErrorCode

エラーコードを取得します。

ErrorCode **GetErrorCode**()

パラメーター

なし

戻り値

エラーコード
スローされた例外のエラーコードを取得します。

備考

- エラーコードの詳細は『5章 エラーコード一覧』を参照してください。

5章　エラーコード一覧

5.1 エラーコード一覧

主なエラーコードは下記の通りです。

マクロ定義(定数)	値	内容
SUCCESS	0	正常
ERR_OPENED	−20	指定したプリンターはすでにオープン済みです。
ERR_NO_PRN	−30	指定したプリンタードライバーがありません。
ERR_HANDLE	−60	APIハンドルの値が不正です。
ERR_TIMEOUT	−70	タイムアウト、またはビジー状態です。
ERR_ACCESS	−80	プリンター、もしくはファイルにアクセスできません。
ERR_PARAM	−90	パラメーターが不正です。
ERR_NOT_SUPPORT	−100	サポートしていない機能です。
ERR_OFFLINE	−110	切断状態またはプリンターがオフラインです。
ERR_NOT_SII	−120	対象外のプリンタードライバーです。
ERR_DISK_FULL	−170	プリンターがビジー状態です。
ERR_ENTRY_OVER	−190	処理の最大使用数を超えています。
ERR_EXIST	−210	すでに存在するモジュールを呼び出しています。
ERR_NOT_FOUND	−220	ファイルが見つかりません。または登録されていません。
ERR_WORKAREA_NO_MEMORY	−260	メモリの領域指定が不足しています。
ERR_WORKAREA_FAILED	−280	メモリの確保ができません。
ERR_EXEC_FUNC	−310	別のスレッドまたはプロセスで使用中のため、その機能は使用できません。
ERR_SPL_NOT_EXIST	−350	スプーラーサービス、またはSII Printer Software サービスが起動していません。
ERR_INVALID_DATA	−1020	不正なデータが指定されています。
ERR_READ_FAULT	−1030	プリンターからデータを受信できません。
ERR_WRITE_FAULT	−1040	プリンターにデータを送信できません。
ERR_CANCELLED	−1050	機能が取り消されました。

ERR_PRN_HAS_JOBS_QUEUED	−1060	プリンターに印刷待ちジョブがあります。
マクロ定義(定数)	値	内容
ERR_UNKNOWN_PORT	−1070	対象外のポートです。
ERR_INVALID_PRN_STATE	−1080	プリンターステータスが正常ではありません。
ERR_BAD_ENVIRONMENT	−1090	プリンタードライバーのインストールが正常に行われていない可能性があります。
ERR_PORT_NOT_OPENED	−1120	ポートがオープンされていません。
ERR_DATA_SIZE_TOOBIG	−1130	データサイズが最大値を超過しています。
ERR_ENCODE_FAILED	−1140	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_FILE_USED	−1150	指定されたファイルは別プロセスで使用中です。
ERR_FILE_INVALID	−1160	指定されたファイルは無効です。 (ファイル拡張子またはファイル構造が不正)
ERR_OVER_STYLE_NUM	−1170	指定されたファイルに登録されているスタイル数が規定値(64)を超過しています。
ERR_TRANSACTION_NOT_STARTED	−1180	一括処理が開始されていません。
ERR_IMAGE_DATA_CONVERT_FAILED	−1190	イメージのデータ変換に失敗しました
ERR_FUNCTION_NOT_CALLED	−1200	APIを実行できませんでした。

6章　引数情報

6.1 プリンターステータス一覧

プリンターステータスの応答内容の対応ビットは下記の通りです。

プリンターステータス		対応ビット (Win32 API)	PrinterStatus 構造体メンバ (.NET API)
電圧エラー	有り	*pdwStatus & 0x00000001≠0	Err_Voltage = TRUE
	無し	*pdwStatus & 0x00000001 = 0	Err_Voltage = FALSE
ハードウェアエラー / 初期化電圧エラー	有り	*pdwStatus & 0x00000002≠0	Err_Hardware = TRUE
	無し	*pdwStatus & 0x00000002 = 0	Err_Hardware = FALSE
ヘッド温度エラー	有り	*pdwStatus & 0x00000004≠0	Err_HeadTemperature = TRUE
	無し	*pdwStatus & 0x00000004 = 0	Err_HeadTemperature = FALSE
紙無しエラー	有り	*pdwStatus & 0x00000010≠0	Err_OutOfPaper = TRUE
	無し	*pdwStatus & 0x00000010 = 0	Err_OutOfPaper = FALSE
マーク検出時紙ジャムエラー	有り	*pdwStatus & 0x00000040≠0	Err_MarkPaperJam = TRUE
	無し	*pdwStatus & 0x00000040 = 0	Err_MarkPaperJam = FALSE
ペーパーカバーオープンエラー	有り	*pdwStatus & 0x00000080≠0	Err_CoverOpen = TRUE
	無し	*pdwStatus & 0x00000080 = 0	Err_CoverOpen = FALSE
フィードスイッチの状態	ON	*pdwStatus & 0x00000100≠0	State_FeedSwitch = TRUE
	OFF	*pdwStatus & 0x00000100 = 0	State_FeedSwitch = FALSE
紙送り状態	動作中	*pdwStatus & 0x00000400≠0	State_PaperFeed = TRUE
	停止	*pdwStatus & 0x00000400 = 0	State_PaperFeed = FALSE
復帰待ち状態	有り	*pdwStatus & 0x00000800≠0	State_ReturnWaiting = TRUE
	無し	*pdwStatus & 0x00000800 = 0	State_ReturnWaiting = FALSE
フラッシュメモリ書き換え中	有り	*pdwStatus & 0x00010000≠0	State_FlashMemoryRewriting = TRUE
	無し	*pdwStatus & 0x00010000 = 0	State_FlashMemoryRewriting = FALSE

プリンターステータス		対応ビット (Win32 API)	PrinterStatus 構造体メンバ (.NET API)
バッテリ残量	バッテリなし	*pdwStatus & 0x00700000 = 0x00000000	State_Battery = 0
	レベル4 または5	*pdwStatus & 0x00700000 = 0x00100000	State_Battery = 1
	レベル3	*pdwStatus & 0x00700000 = 0x00200000	State_Battery = 2
	レベル2	*pdwStatus & 0x00700000 = 0x00300000	State_Battery = 3
	レベル1	*pdwStatus & 0x00700000 = 0x00400000	State_Battery = 4
バッテリエラー	有り	*pdwStatus & 0x00800000≠0	Err_Battery = TRUE
	無し	*pdwStatus & 0x00800000 = 0	Err_Battery = FALSE
未接続・応答なし※1, ※2	未接続・ 応答なし	*pdwStatus = 0	State_Err_Offline = TRUE
	いいえ	*pdwStatus≠0	State_Err_Offline = FALSE

※1: ASB設定コマンドの応答に対し、拡張されたステータスです。
「未接続・応答なし」の場合、Win32 APIでは全ビットが0、.NET APIではState_Battery以外はFALSEまたは0での応答となります。

※2: 無線通信での接続の場合、「未接続・応答なし」検出に時間がかかる場合があります。

6.2 プリンター情報

指定可能なプリンター情報とその応答形式は下記の通りです。

プリンター情報（PrinterInformation型）		内容	応答形式
メンバ	値		
GET_NV_MEM_CAP	0	NVグラフィックスメモリ容量	数値
GET_NV_MEM_REM_CAP	1	NVグラフィックスメモリの残容量	数値
GET_NV_MEM_KEYCODE_LIST	2	定義されているNVグラフィックスのキーコード一覧	バイナリ
GET_REM_USER_MEM_CAP_DEFRAG	3	ユーザー領域の整理後の残り容量	数値
GET_REM_USER_MEM_CAP	4	ユーザー領域の残り容量応答	バイナリ
GET_FUNC_SET_RESP	10	機能設定応答	バイナリ
GET_PRN_ID_MODEL	50	プリンターID （機種ID）	バイナリ
GET_PRN_ID_TYPE	51	プリンターID （タイプID）	バイナリ
GET_PRN_ID_ROM_VER	52	プリンターID （ROMバージョンID）	バイナリ
GET_PRN_ID_FIRM_VER_MAIN	53	プリンターID （ファームウェアバージョン(main)）	バイナリ
GET_PRN_ID_MFR	54	プリンターID （メーカー名）	バイナリ
GET_PRN_ID_MODEL_NAME	55	プリンターID （モデル名）	バイナリ
GET_PRN_ID_FIRM_VER_BOOT	56	プリンターID （ファームウェアバージョン(boot)）	バイナリ
GET_PRN_ID_FIRM_CHECKSUM_BOOT	57	プリンターID （ファームウェアチェックサム(boot)）	数値
GET_PRN_ID_FIRM_CHECKSUM_MAIN	58	プリンターID （ファームウェアチェックサム(main)）	数値
GET_PRN_ID_FIRM_CHECKSUM	59	プリンターID （ファームウェアチェックサム(main+boot)）	数値
GET_MAINT_NUM_FEED_LINE	100	メンテナンスカウンター （紙送り行数(100ドットライン単位)）	数値
GET_MAINT_NUM_HEAD_ACTIVE	101	メンテナンスカウンター （サーマルヘッド通電回数(100ドットライン単位)）	数値
GET_MAINT_DRIVE_TIME	102	メンテナンスカウンター （製品の駆動時間(分)）	数値
GET_MAINT_NUM_FEED_LINE_INTEGRATION	103	メンテナンスカウンター （紙送り行数(100ドットラインごと)(積算)）	数値

プリンター情報（PrinterInformation型）		内容	応答形式
メンバ	値		
GET_MAINT_NUM_HEAD_ACTIVE_INTEGRATION	104	メンテナンスカウンター (サーマルヘッド通電回数(100ドットラインごと)(積算))	数値
GET_MAINT_DRIVE_TIME_INTEGRATION	105	メンテナンスカウンター (製品の駆動時間(分単位)(積算))	数値
GET_HFONT_24_CHECKSUM	301	半角フォントIDの送信 (24ドットフォント、チェックサム)	数値
GET_HFONT_24_ID	302	半角フォントIDの送信 (24ドットフォント、ID)	バイナリ
GET_HFONT_24_INT_CHAR	303	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)	バイナリ
GET_HFONT_16_CHECKSUM	304	半角フォントIDの送信 (16ドットフォント、チェックサム)	数値
GET_HFONT_16_ID	305	半角フォントIDの送信 (16ドットフォント、ID)	バイナリ
GET_HFONT_16_INT_CHAR	306	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)	バイナリ
GET_FFONT_LANG	351	全角フォントIDの送信 (言語名)	バイナリ
GET_FFONT_STANDARD	352	全角フォントIDの送信 (規格名)	バイナリ
GET_FFONT_COMPANY	353	全角フォントIDの送信 (社名)	バイナリ
GET_FFONT_CHECKSUM	354	全角フォントIDの送信 (チェックサム)	数値
GET_FFONT_DATA_SIZE	355	全角フォントIDの送信 (データサイズ)	数値
GET_FFONT_ROM_ID	356	全角フォントIDの送信 (ROM_ID)	バイナリ

6.3 一括処理制御方法

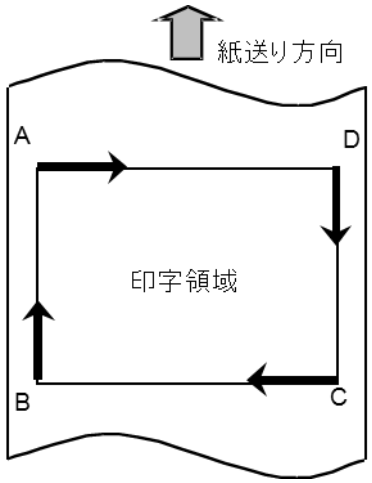
指定可能な一括処理制御方法は下記の通りです。

一括処理制御方法（TransactionFunction型）		内容
メンバ	値	
TRANSACTION_CLEAR	0	一括処理の終了
TRANSACTION_START	1	一括処理の開始
TRANSACTION_PRINT	2	一括印字と一括処理の終了

6.4 印字方向

指定可能な印字方向は下記の通りです。

印字方向(Direction型)		内容	
メンバ	値	印字方向	始点
DIRECTION_LEFT_TO_RIGHT	0	左→右	左上(下図A)
DIRECTION_BOTTOM_TO_TOP	1	下→上	左下(下図B)
DIRECTION_RIGHT_TO_LEFT	2	右→左	右下(下図C)
DIRECTION_TOP_TO_BOTTOM	3	上→下	右上(下図D)



6.5 印字位置

指定可能な印字位置は下記の通りです。

印字位置(Alignment型)		内容
メンバ	値	
ALIGN_LEFT	0	左揃え
ALIGN_CENTER	1	中央揃え
ALIGN_RIGHT	2	右揃え

6.6 文字セット

指定可能な文字セットは下記の通りです。

文字セット(CharacterSet型)		内容
メンバ	値	
CODEPAGE_437	0	USA, Standard Europe (Code Page 437)
CODEPAGE_KATAKANA	1	Katakana
CODEPAGE_850	2	Multilingual (Code Page 850)
CODEPAGE_860	3	Portuguese (Code Page 860)
CODEPAGE_863	4	Canadian-French (Code Page 863)
CODEPAGE_865	5	Nordic (Code Page 865)

文字セット（CharacterSet型）		内容
メンバ	値	
CODEPAGE_857	13	Turkish (Code Page 857)
CODEPAGE_737	14	Greek (Code Page 737)
CODEPAGE_1252	16	Latin (Code Page 1252)
CODEPAGE_866	17	Russian (Code Page 866)
CODEPAGE_852	18	Eastern Europe (Code Page 852)
CODEPAGE_858	19	Euro (Code Page 858)
CODEPAGE_855	34	Cyrillic (Code Page 855)
CODEPAGE_864	37	Arabic (Code Page 864)
CODEPAGE_1250	45	Central European (Code Page 1250)
CODEPAGE_1251	46	Cyrillic (Code Page 1251)
CODEPAGE_1253	47	Greek (Code Page 1253)
CODEPAGE_1254	48	Turkish (Code Page 1254)
CODEPAGE_USERPAGE	255	ユーザーページ

6.7 国際文字セット

指定可能な国際文字セットは下記の通りです。

国際文字セット（InternationalCharacterSet型）		内容
メンバ	値	
INT_CHAR_SET_USA	0	アメリカ（USA）
INT_CHAR_SET_FRANCE	1	フランス（France）
INT_CHAR_SET_GERMANY	2	ドイツ（Germany）
INT_CHAR_SET_UNITED_KINGDOM	3	イギリス（United Kingdom）
INT_CHAR_SET_DENMARK_1	4	デンマークⅠ（Denmark I）
INT_CHAR_SET_SWEDEN	5	スウェーデン（Sweden）
INT_CHAR_SET_ITALY	6	イタリア（Italy）
INT_CHAR_SET_SPAIN_1	7	スペインⅠ（Spain I）
INT_CHAR_SET_JAPAN	8	日本（Japan）
INT_CHAR_SET_NORWAY	9	ノルウェー（Norway）
INT_CHAR_SET_DENMARK_2	10	デンマークⅡ（Denmark II）
INT_CHAR_SET_SPAIN_2	11	スペインⅡ（Spain II）
INT_CHAR_SET_LATIN_AMERICA	12	ラテンアメリカ（Latin America）
INT_CHAR_SET_ARABIA	17	アラビア（Arabia）

6.8 文字フォント

指定可能な文字フォントは下記の通りです。

文字フォント（CharacterType型）		内容
メンバ	値	
CHAR_TYPE_FONT_CURRENT ※1	0	現在の設定
CHAR_TYPE_FONT_A	1	フォントA（24×12）・漢字フォントA（24×24）
CHAR_TYPE_FONT_B	2	フォントB（16×8）・漢字フォントB（16×16）

※1：本値を設定した場合、コマンド送信は行わず、前回行った設定もしくはコマンドの初期値で動作します。

6.9 文字サイズ

指定可能な文字サイズは下記の通りです。

文字サイズ（CharacterScale型）		内容
メンバ	値	
CHAR_SCALE_CURRENT ※1	0	現在の設定
CHAR_SCALE_x1	1	1倍（標準）
CHAR_SCALE_x2	2	2倍
CHAR_SCALE_x3	3	3倍
CHAR_SCALE_x4	4	4倍
CHAR_SCALE_x5	5	5倍
CHAR_SCALE_x6	6	6倍
CHAR_SCALE_x7	7	7倍
CHAR_SCALE_x8	8	8倍

※1：本値を設定した場合、コマンド送信は行わず、前回行った設定もしくはコマンドの初期値で動作します。

6.10 アンダーライン

指定可能なアンダーラインは下記の通りです。

アンダーライン（Underline型）		内容
メンバ	値	
CHAR_UNDERLINE_CURRENT ※1	0	現在の設定
CHAR_UNDERLINE_NONE	1	アンダーライン解除
CHAR_UNDERLINE_1DOT	2	1ドット幅アンダーライン指定
CHAR_UNDERLINE_2DOT	3	2ドット幅アンダーライン指定

※1：本値を設定した場合、コマンド送信は行わず、前回行った設定もしくはコマンドの初期値で動作します。

6.11 強調印字

指定可能な強調印字は下記の通りです。

強調印字(Bold型)		内容
メンバ	値	
CHAR_BOLD_CURRENT ※1	0	現在の設定
CHAR_BOLD_OFF	1	強調印字なし
CHAR_BOLD_ON	2	強調印字あり

※1: 本値を設定した場合、コマンド送信は行わず、前回行った設定もしくはコマンドの初期値で動作します。

6.12 白黒反転印字

指定可能な白黒反転印字は下記の通りです。

白黒反転印字(Reverse型)		内容
メンバ	値	
CHAR_REVERSE_CURRENT ※1	0	現在の設定
CHAR_REVERSE_OFF	1	白黒反転印字なし
CHAR_REVERSE_ON	2	白黒反転印字あり

※1: 本値を設定した場合、コマンド送信は行わず、前回行った設定もしくはコマンドの初期値で動作します。

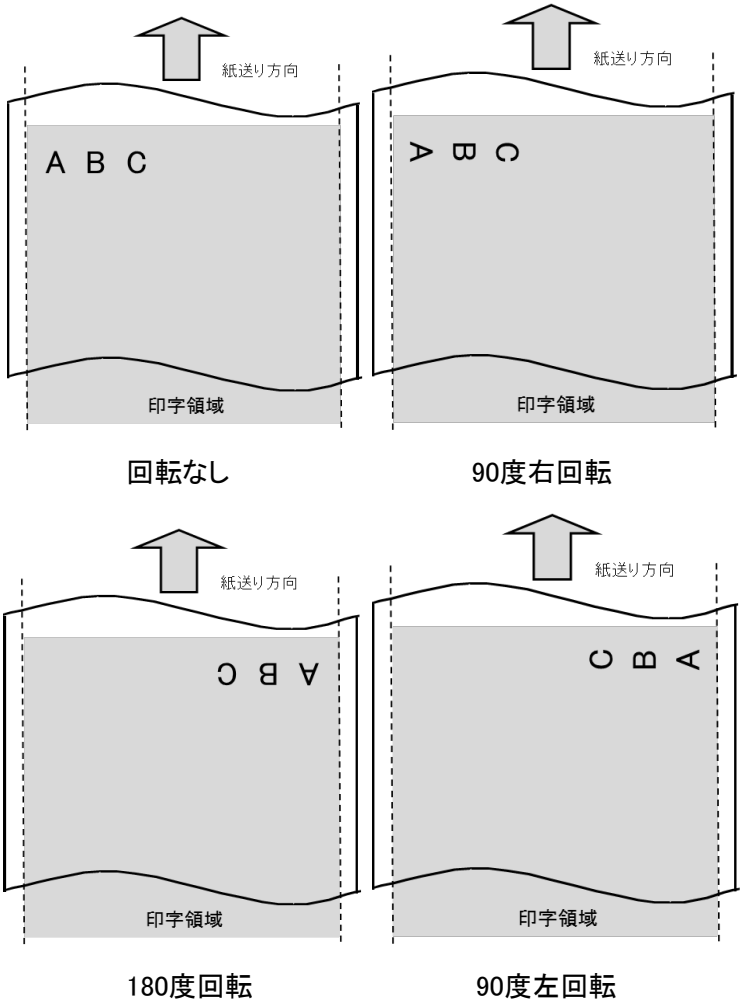
6.13　回転印字

指定可能な回転印字は下記の通りです。

回転印字(Rotate型)		内容
メンバ	値	
ROTATE_CURRENT ※1	0	現在の設定
ROTATE_NONE	1	回転なし
ROTATE_90_TO_RIGHT	2	90度右回転
ROTATE_180	3	180度回転
ROTATE_90_TO_LEFT	4	90度左回転

※1: 本値を設定した場合、コマンド送信は行わず、前回行った設定もしくはコマンドの初期値で動作します。

・ 下記は**MpSetCharacterFormatting**または**SetCharacterFormatting**の動作を示した図です。



6.14　バーコード種類

指定可能なバーコード種類は下記の通りです。

バーコード種類(TypeBarcode型)		内容	分類
メンバ	値		
TYPE_BARCODE_UPC_A	0	UPC-A	マルチレベルバーコード
TYPE_BARCODE_UPC_E	1	UPC-E	マルチレベルバーコード
TYPE_BARCODE_JAN13	2	JAN13 (EAN13)	マルチレベルバーコード
TYPE_BARCODE_JAN8	3	JAN8 (EAN8)	マルチレベルバーコード
TYPE_BARCODE_CODE39	4	CODE39	2値レベルバーコード
TYPE_BARCODE_ITF	5	ITF	2値レベルバーコード
TYPE_BARCODE_CODABAR	6	CODABAR	2値レベルバーコード
TYPE_BARCODE_CODE128	7	CODE128	マルチレベルバーコード
TYPE_BARCODE_CODE93	8	CODE93	マルチレベルバーコード
TYPE_BARCODE_JAN13_ADDON2	9	JAN13 (EAN13) add-on 2	マルチレベルバーコード
TYPE_BARCODE_JAN13_ADDON5	10	JAN13 (EAN13) add-on 5	マルチレベルバーコード
TYPE_BARCODE_GS1_OMNI_DIRCTIONAL	11	GS1 Databar Omni-directional	2値レベルバーコード
TYPE_BARCODE_GS1_TRUNCATED	12	GS1 Databar Truncated	2値レベルバーコード
TYPE_BARCODE_GS1_LIMITED	13	GS1 Databar Limited	2値レベルバーコード
TYPE_BARCODE_GS1_EXPANDED	14	GS1 Databar Expanded	2値レベルバーコード

6.15　バーコードモジュール幅

指定可能なバーコードモジュール幅は下記の通りです。

バーコードモジュール幅 (ModuleWidthBarcode型)		内容	
メンバ	値	マルチレベルバーコードの モジュール幅(mm)	2値レベルバーコードの 細エレメント(mm)
MODULE_WIDTH_BARCODE_2	2	0.250	0.250 (2ドット)
MODULE_WIDTH_BARCODE_3	3	0.375	0.375 (3ドット)
MODULE_WIDTH_BARCODE_4	4	0.500	0.500 (4ドット)
MODULE_WIDTH_BARCODE_5	5	0.625	0.625 (5ドット)
MODULE_WIDTH_BARCODE_6	6	0.750	0.750 (6ドット)

6.16　バーコード高さ

バーコード種類が下記の場合は、指定可能なバーコード高さは下記の通りです。

バーコード種類	初期値	バーコード高さ範囲(ドット)
UPC-A	0※1	1 ～ 255
UPC-E		
JAN13 (EAN13)		
JAN8 (EAN8)		
CODE39		
ITF		
CODABAR		
CODE128		
CODE93		
JAN13 (EAN13) add-on 2		
JAN13 (EAN13) add-on 5		

※1: 0を指定した場合、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。
[高さ]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 / バーコードモジュール幅	初期値	バーコード高さ範囲(ドット)
GS1 Databar Omni-directional		
MODULE_WIDTH_BARCODE_2	0※1	66 ～ 255
MODULE_WIDTH_BARCODE_3		99 ～ 255
MODULE_WIDTH_BARCODE_4		132 ～ 255
MODULE_WIDTH_BARCODE_5		165 ～ 255
MODULE_WIDTH_BARCODE_6		198 ～ 255
GS1 Databar Truncated		
MODULE_WIDTH_BARCODE_2	0※1	26 ～ 255
MODULE_WIDTH_BARCODE_3		39 ～ 255
MODULE_WIDTH_BARCODE_4		52 ～ 255
MODULE_WIDTH_BARCODE_5		65 ～ 255
MODULE_WIDTH_BARCODE_6		78 ～ 255
GS1 Databar Limited		
MODULE_WIDTH_BARCODE_2	0※1	20 ～ 255
MODULE_WIDTH_BARCODE_3		30 ～ 255
MODULE_WIDTH_BARCODE_4		40 ～ 255
MODULE_WIDTH_BARCODE_5		50 ～ 255
MODULE_WIDTH_BARCODE_6		60 ～ 255

バーコード種類 バーコードモジュール幅	初期値	バーコード高さ範囲(ドット)
GS1 Databar Expanded		
MODULE_WIDTH_BARCODE_2	0※1	68 ～ 255
MODULE_WIDTH_BARCODE_3		102 ～ 255
MODULE_WIDTH_BARCODE_4		136 ～ 255
MODULE_WIDTH_BARCODE_5		170 ～ 255
MODULE_WIDTH_BARCODE_6		204 ～ 255

※1: 0を指定した場合、162ドットとなります。

6.17 バーコードHRI文字

指定可能なバーコードHRI文字は下記の通りです。

HRI文字(HriPositionBarcode型)		内容
メンバ	値	
HRI_BARCODE_NONE	0	HRI文字なし
HRI_BARCODE_TOP_FONT_A	1	バーコードの上(フォントA)
HRI_BARCODE_BOTTOM_FONT_A	2	バーコードの下(フォントA)
HRI_BARCODE_FONT_A	3	バーコードの上下(フォントA)
HRI_BARCODE_TOP_FONT_B	4	バーコードの上(フォントB)
HRI_BARCODE_BOTTOM_FONT_B	5	バーコードの下(フォントB)
HRI_BARCODE_FONT_B	6	バーコードの上下(フォントB)

6.18 バーコードN:W比

指定可能なバーコードN:W比は下記の通りです。

N:W比(NwRatioBarcode型)		内容
メンバ	値	
NWRATIO_BARCODE_NONE※2	0	–
NWRATIO_BARCODE_1TO2※1	0	1:2
NWRATIO_BARCODE_1TO2_5※1	1	1:2.5
NWRATIO_BARCODE_1TO3※1	2	1:3

※1: 下記のバーコード種類を使用する場合に指定してください。

- CODE39
- ITF
- CODABAR

※2: ※1で記載したバーコード以外を使用する場合に指定してください。

6.19 2次元バーコード種類

指定可能な2次元バーコード種類は下記の通りです。

2次元バーコード種類(Type2Dcode型)		内容
メンバ	値	
TYPE_2DCODE_QR_CODE	0	QR Code
TYPE_2DCODE_PDF417	1	PDF417
TYPE_2DCODE_DATA_MATRIX	2	Data Matrix
TYPE_2DCODE_MAXI_CODE	3	Maxi Code
TYPE_2DCODE_GS1_STACKED	4	GS1 Databar Stacked
TYPE_2DCODE_GS1_OMNI_DIRECTIONAL	5	GS1 Databar Stacked Omni-directional
TYPE_2DCODE_GS1_EXPANDED_STACKED	6	GS1 Databar Expanded Stacked

6.20 2次元バーコードモード

指定可能な2次元バーコードモードは下記の通りです。

2次元バーコードモード(Mode2Dcode型)※1		2次元バーコード種類による内容			
		QR Code	PDF417	Data Matrix	Maxi Code
メンバ	値	モデル	モデル	縦モジュール数 × 横モジュール数	モード
MODE_2DCODE_NONE	0	–	–	–	–
MODE_2DCODE_PDF417_STANDARD		–	通常モード	–	–
MODE_2DCODE_DATA_MATRIX_AUTO		–	–	自動	–
MODE_2DCODE_QR_CODE_MODEL1	1	Model 1	–	–	–
MODE_2DCODE_PDF417_COMPACT		–	簡易モード	–	–
MODE_2DCODE_DATA_MATRIX_10_10		–	–	10×10	–

2次元バーコードモード(Mode2Dcode型)※1		2次元バーコード種類による内容			
		QR Code	PDF417	Data Matrix	Maxi Code
メンバ	値	モデル	モデル	縦モジュール数 × 横モジュール数	モード
MODE_2DCODE_QR_CODE_MODEL2	2	Model 2	−	−	−
MODE_2DCODE_DATA_MATRIX_12_12		−	−	12×12	−
MODE_2DCODE_MAXI_CODE_2		−	−	−	Mode2
MODE_2DCODE_DATA_MATRIX_14_14	3	−	−	14×14	−
MODE_2DCODE_MAXI_CODE_3		−	−	−	Mode3
MODE_2DCODE_DATA_MATRIX_16_16	4	−	−	16×16	−
MODE_2DCODE_MAXI_CODE_4		−	−	−	Mode4
MODE_2DCODE_DATA_MATRIX_18_18	5	−	−	18×18	−
MODE_2DCODE_MAXI_CODE_5		−	−	−	Mode5
MODE_2DCODE_DATA_MATRIX_20_20	6	−	−	20×20	−
MODE_2DCODE_DATA_MATRIX_22_22	7	−	−	22×22	−
MODE_2DCODE_DATA_MATRIX_24_24	8	−	−	24×24	−
MODE_2DCODE_DATA_MATRIX_26_26	9	−	−	26×26	−
MODE_2DCODE_DATA_MATRIX_32_32	10	−	−	32×32	−
MODE_2DCODE_DATA_MATRIX_36_36	11	−	−	36×36	−
MODE_2DCODE_DATA_MATRIX_40_40	12	−	−	40×40	−
MODE_2DCODE_DATA_MATRIX_44_44	13	−	−	44×44	−
MODE_2DCODE_DATA_MATRIX_48_48	14	−	−	48×48	−
MODE_2DCODE_DATA_MATRIX_52_52	15	−	−	52×52	−
MODE_2DCODE_DATA_MATRIX_64_64	16	−	−	64×64	−
MODE_2DCODE_DATA_MATRIX_72_72	17	−	−	72×72	−
MODE_2DCODE_DATA_MATRIX_80_80	18	−	−	80×80	−
MODE_2DCODE_DATA_MATRIX_88_88	19	−	−	88×88	−
MODE_2DCODE_DATA_MATRIX_96_96	20	−	−	96×96	−
MODE_2DCODE_DATA_MATRIX_104_104	21	−	−	104×104	−
MODE_2DCODE_DATA_MATRIX_120_120	22	−	−	120×120	−
MODE_2DCODE_DATA_MATRIX_132_132	23	−	−	132×132	−
MODE_2DCODE_DATA_MATRIX_144_144	24	−	−	144×144	−
MODE_2DCODE_DATA_MATRIX_8_18	25	−	−	8×18	−
MODE_2DCODE_DATA_MATRIX_8_32	26	−	−	8×32	−
MODE_2DCODE_DATA_MATRIX_12_26	27	−	−	12×26	−
MODE_2DCODE_DATA_MATRIX_12_36	28	−	−	12×36	−
MODE_2DCODE_DATA_MATRIX_16_36	29	−	−	16×36	−
MODE_2DCODE_DATA_MATRIX_16_48	30	−	−	16×48	−

※1: 表に記載のない2次元バーコード種類を使用する場合は、0(MODE_2DCODE_NONE)を指定してください。

6.21　2次元バーコードモジュールサイズ

指定可能な2次元バーコードモジュールサイズは下記の通りです。

2次元バーコードモジュールサイズ（ModuleSize2Dcode型）※1		2次元バーコード種類による内容			
		QR Code	PDF417	Data Matrix	GS1 Databar すべて
メンバ	値	モジュールサイズ	公称細エレメント幅	モジュールサイズ	モジュールサイズ
MODULE_SIZE_2DCODE_NONE	0	－	－	－	－
MODULE_SIZE_2DCODE_DEFAULT		初期値（6ドット）	初期値（3ドット）	初期値（6ドット）	初期値（6ドット）
MODULE_SIZE_2DCODE_2	2	2ドット	2ドット	2ドット	2ドット
MODULE_SIZE_2DCODE_3	3	3ドット	3ドット	3ドット	3ドット
MODULE_SIZE_2DCODE_4	4	4ドット	4ドット	4ドット	4ドット
MODULE_SIZE_2DCODE_5	5	5ドット	5ドット	5ドット	5ドット
MODULE_SIZE_2DCODE_6	6	6ドット	6ドット	6ドット	6ドット
MODULE_SIZE_2DCODE_7	7	7ドット	7ドット	7ドット	7ドット
MODULE_SIZE_2DCODE_8	8	8ドット	8ドット	8ドット	8ドット
MODULE_SIZE_2DCODE_9	9	9ドット	－	9ドット	9ドット
MODULE_SIZE_2DCODE_10	10	10ドット	－	10ドット	10ドット
MODULE_SIZE_2DCODE_11	11	11ドット	－	11ドット	11ドット
MODULE_SIZE_2DCODE_12	12	12ドット	－	12ドット	12ドット
MODULE_SIZE_2DCODE_13	13	13ドット	－	13ドット	13ドット
MODULE_SIZE_2DCODE_14	14	14ドット	－	14ドット	14ドット
MODULE_SIZE_2DCODE_15	15	15ドット	－	15ドット	15ドット
MODULE_SIZE_2DCODE_16	16	16ドット	－	16ドット	16ドット

※1：表に記載のない2次元バーコード種類を使用する場合は、0(MODULE_SIZE_2DCODE_NONE)を指定してください。

6.22　2次元バーコードモジュール高さ

指定可能な2次元バーコードモジュール高さは下記の通りです。

2次元バーコード種類※1	2次元バーコード種類による内容		
	PDF段高さ		1段の高さ
	初期値	範囲(ドット)	範囲(ドット)
PDF417	0※2	2 ～ 127	-
GS1 Databar Stacked Omni-directional	-		33 ～ 255

※1: 表に記載のない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: 0を指定した場合、162ドットとなります。

6.23　2次元バーコードカラム数

指定可能な2次元バーコードカラム数は下記の通りです。

2次元バーコード種類※1	2次元バーコード種類による内容	
	データ領域のカラム数	1行のエレメント数※2
PDF417	0※3、1 ～ 30	-
GS1 Databar Expanded Stacked	-	2 ～ 20

※1: 表に記載のない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: 1行のエレメント数には偶数値を指定してください。
※3: 0を指定した場合、カラム数は自動となります。

6.24　2次元バーコード段数

指定可能な2次元バーコード段数は下記の通りです。

2次元バーコード種類※1	2次元バーコード種類による内容
	段数
PDF417	0※2、3 ～ 90

※1: 表に記載のない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: 0を指定した場合、段数は自動となります。

6.25　2次元バーコードエラー訂正レベル

指定可能な2次元バーコードエラー訂正レベルは下記の通りです。

2次元バーコードエラー訂正レベル (ErrorCorrect2Dcode型)※1		2次元バーコード種類による内容	
		QR Code	PDF417
メンバ	値	エラー訂正レベル	エラー訂正レベル
ERR_CORRECTION_2DCODE_NONE	0	−	−
ERR_CORRECTION_2DCODE_ QR_CODE_L		L	−
ERR_CORRECTION_2DCODE_ PDF417_0		−	0
ERR_CORRECTION_2DCODE_ QR_CODE_M	1	M	−
ERR_CORRECTION_2DCODE_ PDF417_1		−	1
ERR_CORRECTION_2DCODE_ QR_CODE_Q	2	Q	−
ERR_CORRECTION_2DCODE_ PDF417_2		−	2
ERR_CORRECTION_2DCODE_ QR_CODE_H	3	H	−
ERR_CORRECTION_2DCODE_ PDF417_3		−	3
ERR_CORRECTION_2DCODE_ PDF417_4	4	−	4
ERR_CORRECTION_2DCODE_ PDF417_5	5	−	5
ERR_CORRECTION_2DCODE_ PDF417_6	6	−	6
ERR_CORRECTION_2DCODE_ PDF417_7	7	−	7
ERR_CORRECTION_2DCODE_ PDF417_8	8	−	8

※1: 表に記載のない2次元バーコード種類を使用する場合は、0(ERR_CORRECTION_2DCODE_NONE)を指定してください。