



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

Rev.02

[対象製品]
MP-B20 シリーズ

セイコーインスツル株式会社

Rev.01	2017 年 2 月
Rev.02	2020 年 10 月

©セイコーインスツル株式会社 2017-2020

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MP-B20シリーズ用「SII Printer Driver for Windows」(以降: プリンタードライバー)上で動作する、MP-B20シリーズ用「SII SDK for Windows」(以降: SDK)について説明します。

記号

本書で使用されてい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。

対象プリンタードライバー

SDKでサポートするプリンタードライバーを以下に記載します。

- MP-B20シリーズ用「SII Printer Driver for Windows」

用語

本書で使用されている用語について以下に記載します。

用語	内容
プリンターコマンド	「MP-B20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。
ASB設定コマンド (ASB: Automatic Status Back)	プリンターコマンド「自動ステータス送信の有効・無効」。 詳細は、「MP-B20シリーズ サーマルプリンタ 技術説明書」の「6章 コマンドの機能」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に、拡張ステータスを付加した内容です。 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。

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る改ざんが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 動作条件	1-1
2章 インストール	2-1
3章 Win32 API	3-1
3.1 開発対象言語	3-1
3.2 ライブラリファイル	3-1
3.3 API一覧	3-2
3.4 API詳細	3-3
OpenMonPrinter	3-3
CloseMonPrinter	3-4
LockPrinter	3-4
UnlockPrinter	3-5
DirectIO	3-6
DirectIOEx	3-7
Reset	3-9
GetStatus	3-9
SetStatusBackFunction	3-10
SetStatusBackWnd	3-11
CancelStatusBack	3-12
PowerOff	3-13
GetCounter	3-13
ResetCounter	3-14
GetType	3-15
GetPrnCapability	3-16
SendDataFile	3-17
DirectSendRead	3-17
GetProperty	3-19
SetProperty	3-20
4章 .NET API	4-1
4.1 開発対象言語	4-1
4.2 ライブラリファイル	4-1
4.3 API一覧	4-2
4.4 API詳細	4-3
4.4.1 プロパティ	4-3
Status	4-3
LastError	4-3
IsValid	4-4
4.4.2 メソッド	4-5
OpenMonPrinter	4-5
CloseMonPrinter	4-5
LockPrinter	4-6
UnlockPrinter	4-6

DirectIOEx	4-7
ResetPrinter	4-8
SetStatusBack	4-9
CancelStatusBack	4-9
PowerOff	4-10
GetCounter	4-10
ResetCounter	4-11
GetType	4-11
GetPrnCapability	4-12
SendDataFile	4-13
DirectSendRead	4-13
GetProperty	4-15
SetProperty	4-15
4.4.3 イベント	4-17
StatusCallback	4-17

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 カウンターID	6-2
6.3 プリンターID	6-3
6.4 プロパティID	6-4

7章 コマンド定義ファイル 7-1

7.1 概要	7-1
7.2 フォーマット	7-1
7.3 利用方法	7-3

1章 概要

本章ではSDKの概要について説明します。

SDKには、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ライブラリファイルが含まれます。また、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。また、一部のプライベートなDEVMODE設定項目の値を取得及び変更することが可能です。

SDKは以下のライブラリファイルを内包しています。

- Win32開発環境用SDK(以降: Win32 API)
- .NET開発環境用SDK(以降: .NET API)

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。

1.1 動作条件

SDKの動作条件は、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。プリンタードライバーの動作環境などの詳細は、MP-B20シリーズ用「SII Printer Driver for Windows 取扱説明書」を参照してください。

その他に、以下の動作条件を満たす必要があります。

- .NET APIを利用する場合は、.NET Framework Version 2.0以上が必要です。コンピューターから.NET Frameworkをアンインストールすると.NET APIを利用できなくなります。
- SDKのすべての機能は双方向サポートの機能が有効、及びプリンタープールの機能が無効である必要があります。

2章 インストール

SDKのインストール方法は、MP-B20シリーズ用「SII Printer Driver for Windows 取扱説明書」のインストール方法を参照してください。

3章 Win32 API

本章ではWin32 APIについて説明します。

3.1 開発対象言語

開発対象言語は以下の通りです。

- Visual C++

3.2 ライブラリファイル

Win32 APIのライブラリファイルはダイナミックリンクライブラリ形式です。

ライブラリファイルのファイル名は以下の通りです。

- SiiMpb2Api.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。

3.3 API一覧

Win32 APIに実装されているAPIは以下の通りです。

API	機能概略
OpenMonPrinterA ^{※1} OpenMonPrinterW ^{※1}	Win32 APIの使用を開始し、APIハンドルを返します。
CloseMonPrinter	使用中のWin32 APIを終了します。
LockPrinter ^{※2}	プリンターへのデータ送信及びリセット制御をロックします。
UnlockPrinter	プリンターへのデータ送信及びリセット制御のロックを解除します。
DirectIO ^{※2、※3}	バイナリデータを送信及び受信します。 (受信データにはASB設定コマンドの応答は含まれません)
DirectIOEx ^{※2、※3}	バイナリデータを送信及び受信します。
Reset ^{※2、※3}	プリンターをハードウェアリセットします。
GetStatus	最新のプリンターステータスを取得します。
SetStatusBackFunction	プリンターステータスの変化を検出すると呼び出されるコールバック関数を登録します。
SetStatusBackWnd	プリンターステータスの変化を検出した際、クリックイベントが呼び出されるボタンのウィンドウハンドルと、プリンターステータスをセットする変数を登録します。
CancelStatusBack	SetStatusBackWnd 及び SetStatusBackFunction で実行されたコールバック関数の登録を解除します。
PowerOff ^{※2、※3}	プリンターを電源オフ状態にします。
GetCounter ^{※2、※3}	メンテナンスカウンターを取得します。
ResetCounter ^{※2、※3}	メンテナンスカウンターを初期化します。
GetType ^{※2、※3}	プリンターの各種IDを取得します。
GetPrnCapability ^{※2、※3}	プリンター情報を取得します。
SendDataFileA ^{※1} SendDataFileW ^{※1}	コマンド定義ファイルの内容を、SDK内部のメモリに登録します。
DirectSendReadA ^{※1、※2、※3} DirectSendReadW ^{※1、※2、※3}	SendDataFile で登録したコマンド定義ファイルの内容を実行します。
GetProperty	印刷設定内容(一部)を取得します。
SetProperty	印刷設定内容(一部)を変更します。

※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: 他のプロセスで **LockPrinter** を呼び出した場合、本 API は失敗します。

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

3.4 API詳細

注意

◆ Bluetooth接続では、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。

OpenMonPrinter

Win32 APIの使用を開始し、APIハンドルを返します。

```
INT OpenMonPrinter(  
    INT i_type,  
    LPCTSTR i_prt)
```

パラメーター

i_type
オープンタイプ
2(固定)

i_prt
Win32 API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。

戻り値

成功 : プリンターを識別するAPIハンドルを返します。

失敗 : エラーコードを返します。
エラーコードの内容は、「5章 エラーコード一覧」を参照してください。

備考

- 1プロセスで同時に取得できるAPIハンドルの数は最大8個です。全プロセスでは最大32個になります。
- 本APIで取得したAPIハンドルを使用しなくなった場合、必ず**CloseMonPrinter**を使って無効にしてください。
- プリンタードライバの接続先がUSB、またはBluetooth以外の場合は、本APIは失敗します。
- プリンターが未接続または電源オフ状態でも本APIは成功します。

CloseMonPrinter

使用中のWin32 APIを終了します。

```
INT CloseMonPrinter(  
    INT i_hdl)
```

パラメーター

i_hdl

APIハンドル

OpenMonPrinterで取得したAPIハンドルを指定します。

戻り値

成功 : 0を返します。

失敗 : エラーコードを返します。

エラーコードの内容は、「5章 エラーコード一覧」を参照してください。

備考

- 本APIで呼び出したAPIハンドルが他のAPIで使用中の場合は、その処理が完了するまで本APIの制御は戻りません。
- 本APIで指定したAPIハンドルに関連付けられたすべての設定及びデータは破棄されます。

LockPrinter

プリンターへのデータ送信及びリセット制御をロックします。

```
INT LockPrinter(  
    INT i_hdl,  
    DWORD i_timeout)
```

パラメーター

i_hdl

APIハンドル

OpenMonPrinterで取得したAPIハンドルを指定します。

i_timeout

タイムアウト時間

本APIの成功を待つ時間を、msec(ミリ秒)単位で指定します。

設定範囲は3000ms～90000msです。

3000ms未満の値を指定した場合は3000msに設定されます。

90000msよりも大きい値を指定した場合は90000msに設定されます。

戻り値

成功 : 0を返します。

失敗 : エラーコードを返します。

エラーコードの内容は、「5章 エラーコード一覧」を参照してください。

備考

- 本APIは、プリンターへのデータ送信及びリセット制御をロックします。解除するためには、**UnlockPrinter**を使用してください。
- 本APIを呼び出してから、**UnlockPrinter**が実行されるまでの間に、他のプロセスからプリンターにデータ送信するAPI、またはリセット制御を行うAPIを使用すると失敗します。
- 本APIによるロックはプロセス内で有効です。よって、ロックしている間、同一プロセスの他のスレッドからはデバイスに直接アクセスするようなAPIも実行が可能です。
- すでにロックしているAPIハンドルでの本APIの繰り返し回数は最大99回です。ロック状態を解除するには本APIを実行した回数と同じ回数の**UnlockPrinter**を実行してください。

UnlockPrinter

プリンターへのデータ送信及びリセット制御のロックを解除します。

```
INT UnlockPrinter(  
    INT i_hdl)
```

パラメーター

i_hdl

APIハンドル

OpenMonPrinterで取得したAPIハンドルを指定します。

戻り値

成功 : 0を返します。

失敗 : エラーコードを返します。

エラーコードの内容は、「5章 エラーコード一覧」を参照してください。

備考

- **LockPrinter**で設定されたプリンターのロックを解除します。
- **LockPrinter**を複数回呼び出した場合、ロック解除には同じ回数分の本APIの呼び出しが必要です。

DirectIO

バイナリデータを送信及び受信します。

```
INT DirectIO(  
    INT i_hdl,  
    BYTE i_wlen,  
    LPBYTE i_wcmd,  
    LPBYTE io_rlen,  
    LPBYTE o_rbuf,  
    DWORD i_timeout,  
    BOOL i_flag)
```

パラメーター

i_hdl

APIハンドル

OpenMonPrinterで取得したAPIハンドルを指定します。

i_wlen

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。

i_wcmd

送信データバッファ

送信するデータが格納されたバッファを指定します。

io_rlen

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長を指定します。

データ取得が必要のない場合は、0バイトを指定します。

APIから制御が戻ると取得した受信データサイズが格納されます。

o_rbuf

受信データバッファ

取得するデータを格納するバッファを指定します。

i_timeout

タイムアウト時間

本APIの成功を待つ時間を、msec(ミリ秒)単位で指定します。

設定範囲は3000ms～90000msです。

3000ms未満の値を指定した場合は3000msに設定されます。

90000msよりも大きい値を指定した場合は90000msに設定されます。

i_flag

受信動作フラグ

受信動作を指定するフラグを以下から指定します。

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続けます。

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

戻り値

成功 : 0を返します。

失敗 : エラーコードを返します。

エラーコードの内容は、「5章 エラーコード一覧」を参照してください。

備考

- 本APIはResetで中断できます。
- 別データの割り込みが許されないコマンドやデータ、イメージデータの送信の場合は、**LockPrinter**を呼び出した後に本APIを使用してください。**LockPrinter**を呼び出さない場合、別プロセスからのデータが割り込む恐れがあります。
- 受信データにはASB設定コマンドの応答は含まれません。ASB設定コマンドの応答を含むデータの取得を行う場合は**DirectIOEx**を実行してください。
- 本APIでASB設定コマンドを無効に設定するデータを含まないでください。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。
- Bluetooth接続の場合は、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。プリンターをハードウェアリセットする場合は、**Reset**を使用してください。

DirectIOEx

バイナリデータを送信及び受信します。

```
INT DirectIOEx(  
    INT i_hdl,  
    DWORD i_wlen,  
    LPBYTE i_wcmd,  
    LPDWORD io_rlen,  
    LPBYTE o_rbuf,  
    DWORD i_timeout,  
    BOOL i_flag,  
    BYTE i_op)
```

パラメーター

i_hdl

APIハンドル

OpenMonPrinterで取得したAPIハンドルを指定します。

i_wlen

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。

i_wcmd

送信データバッファー

送信するデータが格納されたバッファーを指定します。

io_rlen

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長を指定します。

受信データサイズは最大4096バイトです。

4096バイトよりも大きい値を指定した場合は4096バイトに設定されます。

データ取得が必要のない場合は、0バイトを指定します。

APIから制御が戻ると受信されたデータサイズが格納されます。

o_rbuf

受信データバッファ

取得するデータを格納するバッファを指定します。

i_timeout

タイムアウト時間

本APIの成功を待つ時間を、msec(ミリ秒)単位で指定します。

設定範囲は3000ms～90000msです。

3000ms未満の値を指定した場合は3000msに設定されます。

90000msよりも大きい値を指定した場合は90000msに設定されます。

i_flag

受信動作フラグ

受信動作を指定するフラグを以下から指定します。

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続けます。

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

i_op

受信対象オプション

受信対象のデータを以下から指定します。

0 : ASB設定コマンドの応答を除いたデータを取得します。

1 : ASB設定コマンドの応答を含めたデータを取得します。

戻り値

成功 : 0を返します。

失敗 : エラーコードを返します。

エラーコードの内容は、「5章 エラーコード一覧」を参照してください。

備考

- 本APIは**Reset**で中断できます。
- 別データの割り込みが許されないコマンドやデータ、イメージデータの送信の場合は、**LockPrinter**を呼び出した後に本APIを使用してください。**LockPrinter**を呼び出さない場合、別プロセスからのデータが割り込む恐れがあります。
- 本APIでASB設定コマンドを無効に設定するデータを含まないでください。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。
- Bluetooth接続の場合は、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。プリンターをハードウェアリセットする場合は、**Reset**を使用してください。

Reset

プリンターをハードウェアリセットします。

```
INT Reset(  
    INT i_hdl)
```

パラメーター

i_hdl
APIハンドル
OpenMonPrinterで取得したAPIハンドルを指定します。

戻り値

成功 : 0を返します。

失敗 : エラーコードを返します。
エラーコードの内容は、「5章 エラーコード一覧」を参照してください。

備考

- 通信プロトコルを使用した(プリンターコマンドを使用しない)ハードウェアリセットを行います。
- 本APIを呼び出した場合、以下のAPIは中断されます。
 - **DirectIO**
 - **DirectIOEx**
 - **DirectSendRead**
- 本APIを実行した後、数秒経過してからデータを送信してください。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実行中のプリンターステータスは、「応答なし」の応答になります。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。
- Bluetooth接続で、プリンターがデータを受け付けけない状態で本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。

GetStatus

最新のプリンターステータスを取得します。

```
INT GetStatus(  
    INT i_hdl,  
    LPDWORD o_status)
```

パラメーター

i_hdl
APIハンドル
OpenMonPrinterで取得したAPIハンドルを指定します。

o_status

プリンターステータス変数

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 : 0を返します。

失敗 : エラーコードを返します。

エラーコードの内容は、「5章 エラーコード一覧」を参照してください。

備考

- 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プリンターステータスになります。
- 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。

SetStatusBackFunction

プリンターステータスの変化を検出すると呼び出されるコールバック関数を登録します。

```
INT SetStatusBackFunction(  
    INT i_hdl,  
    INT ( CALLBACK EXPORT *pStatusCB ) ( DWORD o_st ) )
```

パラメーター

i_hdl

APIハンドル

OpenMonPrinterで取得したAPIハンドルを指定します。

lpStatusCB

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。

o_st

プリンターステータス変数

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 : 0を返します。

失敗 : エラーコードを返します。

エラーコードの内容は、「5章 エラーコード一覧」を参照してください。

備考

- 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。

- 本APIで登録したコールバック関数は、以下のAPIで解除されます。
 - **CancelStatusBack**
 - **CloseMonPrinter**
- 登録されたコールバック関数内から同一のAPIハンドルでWin32 API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プリンターステータスになります。
- 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と比べ変化がない場合、コールバック関数は呼び出されません。
- 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。
- すでに登録済みで有効なコールバック関数を再度指定して本APIを呼び出した場合も、直後のプリンターステータス応答は行われます。
- 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。

SetStatusBackWnd

プリンターステータスの変化を検出した際、クリックイベントが呼び出されるボタンのウィンドウハンドルと、プリンターステータスをセットする変数を登録します。

```
INT SetStatusBackWnd(
    INT i_hdl,
    HANDLE i_Wnd,
    LPDWORD o_status )
```

パラメーター

i_hdl

APIハンドル

OpenMonPrinterで取得したAPIハンドルを指定します。

i_Wnd

ウィンドウハンドル

クリックイベントを送るボタンのウィンドウハンドルを指定します。

NULLを指定した場合、本APIを解除します。

o_status

プリンターステータス変数

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 : 0を返します。

失敗 : エラーコードを返します。

エラーコードの内容は、「5章 エラーコード一覧」を参照してください。

備考

- 本APIは以下のAPIで解除されます。
 - **CancelStatusBack**
 - **CloseMonPrinter**
- 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プリンターステータスになります。
- 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と比べ変化がない場合、クリックイベントは呼び出されません。
- 本APIによりボタンのウィンドウハンドルを登録すると、現在のプリンターステータスでクリックイベントを呼び出します。
- ボタンのウィンドウハンドルを登録した状態で本APIを呼び出した場合、登録済みのボタンのウィンドウハンドルは無効となり、新しいボタンのウィンドウハンドルが登録されます。
- すでに登録済みで有効なボタンのウィンドウハンドルを再度指定して本APIを呼び出した場合も、直後のプリンターステータス応答は行われます。
- クリックイベント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クリックイベント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。

CancelStatusBack

SetStatusBackWnd及び**SetStatusBackFunction**で実行されたコールバック関数の登録を解除します。

```
INT CancelStatusBack(  
    INT i_hdl)
```

パラメーター

i_hdl

APIハンドル

OpenMonPrinterで取得したAPIハンドルを指定します。

戻り値

成功 : 0を返します。

失敗 : エラーコードを返します。

エラーコードの内容は、「5章 エラーコード一覧」を参照してください。

備考

- **SetStatusBackWnd**及び**SetStatusBackFunction**のいずれの登録も行われていない場合でも本APIは成功します。

PowerOff

プリンターを電源オフ状態にします。

```
INT PowerOff(  
    INT i_hdl)
```

パラメーター

i_hdl

APIハンドル

OpenMonPrinterで取得したAPIハンドルを指定します。

戻り値

成功 : 0を返します。

失敗 : エラーコードを返します。

エラーコードの内容は、「5章 エラーコード一覧」を参照してください。

GetCounter

メンテナンスカウンターを取得します。

```
INT GetCounter(  
    INT i_hdl,  
    WORD i_readno,  
    LPDWORD o_readcounter)
```

パラメーター

i_hdl

APIハンドル

OpenMonPrinterで取得したAPIハンドルを指定します。

i_readno

カウンターID

取得するカウンターIDを指定します。

指定可能な内容は、「6.2 カウンターID」を参照してください。

o_readcounter

カウンター変数

取得するカウンター値を格納する変数を指定します。

戻り値

成功 : 0を返します。

失敗 : エラーコードを返します。

エラーコードの内容は、「5章 エラーコード一覧」を参照してください。

ResetCounter

メンテナンスカウンターを初期化します。

```
INT ResetCounter(  
    INT i_hdl,  
    WORD i_readno)
```

パラメーター

i_hdl

APIハンドル

OpenMonPrinterで取得したAPIハンドルを指定します。

i_readno

カウンターID

初期化するカウンターIDを指定します。

指定可能な内容は、「6.2 カウンターID」を参照してください。

戻り値

成功 : 0を返します。

失敗 : エラーコードを返します。

エラーコードの内容は、「5章 エラーコード一覧」を参照してください。

備考

- 本APIで指定したカウンターIDの値が初期化されたことを**GetCounter**で確認してください。

GetType

プリンターの各種IDを取得します。

```
INT GetType(  
    INT i_hdl,  
    LPBYTE o_typeID  
    LPBYTE o_fontID  
    LPBYTE o_exrom  
    LPBYTE o_special)
```

パラメーター

<i>i_hdl</i>	APIハンドル
	OpenMonPrinterで取得したAPIハンドルを指定します。
<i>o_typeID</i>	ID1
	1(固定)が格納されます。
<i>o_fontID</i>	ID2
	2(固定)が格納されます。
<i>o_exrom</i>	予約
	NULLを指定してください。
<i>o_special</i>	予約
	NULLを指定してください。

戻り値

成功 : 0を返します。

失敗 : エラーコードを返します。

エラーコードの内容は、「5章 エラーコード一覧」を参照してください。

GetPrnCapability

プリンター情報を取得します。

```
INT GetPrnCapability(  
    INT i_hdl,  
    BYTE i_id,  
    LPBYTE io_datsize,  
    LPBYTE o_dat)
```

パラメーター

i_hdl

APIハンドル

OpenMonPrinterで取得したAPIハンドルを指定します。

i_id

プリンターID

取得するプリンターIDを指定します。

指定可能な内容は、「6.3 プリンターID」を参照してください。

io_datsize

受信データサイズ

取得するプリンターIDを格納するバッファサイズを指定します。

APIから制御が戻ると取得した受信データサイズが格納されます。

指定したバッファサイズが取得するプリンターIDの応答サイズよりも小さい場合、本APIは失敗し、指定したプリンターIDの応答サイズが格納されます。

o_dat

受信データバッファ

取得するプリンターIDの値を格納するバッファを指定します。

戻り値

成功 : 0を返します。

失敗 : エラーコードを返します。

エラーコードの内容は、「5章 エラーコード一覧」を参照してください。

SendDataFile

コマンド定義ファイルの内容を、SDK内部のメモリに登録します。

```
INT SendDataFile(  
    INT i_hdl,  
    LPCTSTR i_fname)
```

パラメーター

i_hdl
APIハンドル
OpenMonPrinterで取得したAPIハンドルを指定します。

i_fname
コマンド定義ファイル名
決められた書式で作成されたコマンド定義ファイル名を指定します。

戻り値

成功 : 0を返します。

失敗 : エラーコードを返します。
エラーコードの内容は、「5章 エラーコード一覧」を参照してください。

備考

- コマンド定義ファイルの詳細は、「7章 コマンド定義ファイル」を参照してください。
- 本APIによって登録されたコマンド定義は、CloseMonPrinterで破棄されます。

DirectSendRead

SendDataFileで登録したコマンド定義ファイルの内容を実行します。

```
INT DirectSendRead(  
    INT i_hdl,  
    LPCTSTR i_cname,  
    LPCTSTR i_rtype,  
    LPDWORD io_rlen,  
    LPBYTE o_rbuf,  
    DWORD i_timeout,  
    BOOL i_flag)
```

パラメーター

i_hdl
APIハンドル
OpenMonPrinterで取得したAPIハンドルを指定します。

i_cname
コマンド名
SendDataFileで登録したコマンド名を指定します。

i_rtype

受信データタイプ名

受信するデータのタイプを以下から指定します。

ASB : ASB設定コマンドの応答のみを受信データに保存します。

Other : 上記以外のプリンターからの応答を受信データに保存します。

io_rlen

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長を指定します。

最大受信データサイズは4096バイトです。

4096バイトよりも大きい値を指定した場合は4096バイトに設定されます。

データ取得が必要のない場合は、0バイトを指定します。

APIから制御が戻ると取得した受信データサイズが格納されます。

o_rbuf

受信データバッファ

取得するデータを格納するバッファを指定します。

i_timeout

タイムアウト時間

本APIの成功を待つ時間を、msec(ミリ秒)単位で指定します。

設定範囲は3000ms～90000msです。

3000ms未満の値を指定した場合は3000msに設定されます。

90000msよりも大きい値を指定した場合は90000msに設定されます。

i_flag

受信動作フラグ

受信動作を指定するフラグを以下から指定します。

TRUE : 何かしらのデータを受信するか、タイムアウトになるまで受信を続けます。

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

戻り値

成功 : 0を返します。

失敗 : エラーコードを返します。

エラーコードの内容は、「5章 エラーコード一覧」を参照してください。

備考

- 本APIでASB設定コマンドを無効に設定するデータを含まないでください。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。
- 本APIはResetにより中断できます。

GetProperty

印刷設定内容(一部)を取得します。

```
INT GetProperty(  
    INT i_hdl,  
    LPDEVMODE i_devmode,  
    BYTE i_pid,  
    LPBYTE o_dat,  
    LPDWORD io_size )
```

パラメーター

i_hdl

APIハンドル

OpenMonPrinterで取得したAPIハンドルを指定します。

i_devmode

Devmodeアドレス

Devmodeアドレスを指定します。

i_pid

プロパティID

取得するプロパティIDを指定します。

指定可能な内容は、「6.4 プロパティID」を参照してください。

o_dat

取得データバッファー

取得するプロパティIDの内容を格納するバッファーを指定します。

NULLを指定した場合、本APIは失敗し、指定したプロパティIDの応答サイズが*io_size*に格納されます。

io_size

取得データサイズ

取得するデータの最大長を指定します。

APIから制御が戻ると取得したデータサイズが格納されます。

指定したサイズがプロパティIDの応答サイズよりも小さい場合、本APIは失敗し、指定したプロパティIDの応答サイズが格納されます。

戻り値

成功 : 0を返します。

失敗 : エラーコードを返します。

エラーコードの内容は、「5章 エラーコード一覧」を参照してください。

SetProperty

印刷設定内容(一部)を変更します。

```
INT SetProperty(  
    INT i_hdl,  
    LPDEVMODE i_devmode,  
    BYTE i_pid,  
    LPBYTE i_dat,  
    LPDWORD i_size )
```

パラメーター

i_hdl

APIハンドル

OpenMonPrinterで取得したAPIハンドルを指定します。

i_devmode

Devmodeアドレス

Devmodeアドレスを指定します。

i_pid

プロパティID

変更するプロパティIDを指定します。

指定可能な内容は、「6.4 プロパティID」を参照してください。

i_dat

設定データバッファー

変更するプロパティIDの内容が格納されたバッファーを指定します。

i_size

設定データサイズ

変更するプロパティIDの内容が格納されたバッファーサイズを指定します。

戻り値

成功 : 0を返します。

失敗 : エラーコードを返します。

エラーコードの内容は、「5章 エラーコード一覧」を参照してください。

4章 .NET API

本章では.NET APIについて説明します。

4.1 開発対象言語

開発言語は以下の通りです。

- Visual Basic .NET
- Visual C#

4.2 ライブラリファイル

.NET APIのライブラリファイルはクラスライブラリ形式です。

ライブラリファイルのファイル名は以下の通りです。

- SiiMpb2ClassLib.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。

4.3 API一覧

.NET APIに実装されているAPIは以下の通りです。

- ネームスペース : Sii.SDK.MPosPrinter
- クラス名 : StatusAPI

種別	API	機能概略
プロパティ	Status	最新のプリンターステータスを取得します。
プロパティ	LastError	最後に実行したAPIのエラー値を取得します。
プロパティ	IsValid	OpenMonPrinterの呼び出し状態を取得します。
メソッド	OpenMonPrinter	.NET APIの使用を開始します。
メソッド	CloseMonPrinter	使用中の.NET APIを終了します。
メソッド	LockPrinter	プリンターへのデータ送信及びリセット制御をロックします。
メソッド	UnlockPrinter	プリンターへのデータ送信及びリセット制御のロックを解除します。
メソッド	DirectIOEx	バイナリデータを送信及び受信します。
メソッド	ResetPrinter	プリンターをハードウェアリセットします。
メソッド	SetStatusBack	StatusCallbackイベントによるプリンターステータスの通知を開始します。
メソッド	CancelStatusBack	SetStatusBackイベントによるプリンターステータスの通知を停止します。
メソッド	PowerOff	プリンターを電源オフ状態にします。
メソッド	GetCounter	メンテナンスカウンターを取得します。
メソッド	ResetCounter	メンテナンスカウンターを初期化します。
メソッド	GetType	プリンターの各種IDを取得します。
メソッド	GetPrnCapability	プリンター情報を取得します。
メソッド	SendDataFile	コマンド定義ファイルの内容を、SDK内部のメモリに登録します。
メソッド	DirectSendRead	SendDataFileで登録したコマンド定義ファイルの内容を実行します。
メソッド	GetProperty	印刷設定内容(一部)を取得します。
メソッド	SetProperty	印刷設定内容(一部)を変更します。
イベント	StatusCallback	応答されたプリンターステータスを通知します。

4.4 API詳細

注意

◆ Bluetooth接続では、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。

4.4.1 プロパティ

Status

最新のプリンターステータスを取得します。

```
SII.SDK.MposPrinter.ASB Status { get; }
```

初期値

ASB.NO_RESPONSE

備考

- 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プリンターステータスになります。
- `isValid`がFALSEの時は、正しい値が取得できません。
- 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。

LastError

最後に実行したAPIのエラー値を取得します。

```
SII.SDK.MposPrinter.ErrorCode LastError { get; }
```

初期値

SUCCESS

備考

- エラーコードの内容は、「5章 エラーコード一覧」を参照してください。

IsValid

OpenMonPrinterの呼び出し状態を取得します。

```
bool IsValid { get; }
```

初期値

FALSE

備考

- TRUE : OpenMonPrinterに成功した状態です。
- FALSE : OpenMonPrinterに成功していない状態です。

4.4.2 メソッド

OpenMonPrinter

.NET APIの使用を開始します。

```
ErrorCode OpenMonPrinter(  
    OpenType type,  
    string name )
```

パラメーター

type
オープンタイプ
OpenType.TYPE_PRINTER（固定）

name
.NET API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。

戻り値

エラーコードを返します。エラーコードの内容は、「5章 エラーコード一覧」を参照してください。

備考

- 1プロセスで取得できるAPIハンドルの数は1個です。
- .NET APIを使用しなくなった場合、必ず**CloseMonPrinter**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、またはBluetooth以外の場合は、本APIは失敗します。
- プリンターが未接続または電源オフ状態でも本APIは成功します。

CloseMonPrinter

使用中の.NET APIを終了します。

```
ErrorCode CloseMonPrinter()
```

戻り値

エラーコードを返します。エラーコードの内容は、「5章 エラーコード一覧」を参照してください。

備考

- 本APIで指定したAPIに関連付けられたすべての設定及びデータは破棄されます。

LockPrinter

プリンターへのデータ送信及びリセット制御をロックします。

```
ErrorCode LockPrinter(  
    int timeout )
```

パラメーター

timeout

タイムアウト時間

本APIの成功を待つ時間を、msec(ミリ秒)単位で指定します。

設定範囲は3000ms～90000msです。

3000ms未満の値を指定した場合は3000msに設定されます。

90000msよりも大きい値を指定した場合は90000msに設定されます。

戻り値

エラーコードを返します。エラーコードの内容は、「5章 エラーコード一覧」を参照してください。

備考

- 本APIは、プリンターへのデータ送信及びリセット制御をロックします。解除するためには、**UnlockPrinter**を使用してください。
- 本APIの繰り返し回数は最大99回です。解除するには本APIを実行した回数と同じ回数の**UnlockPrinter**を実行してください。

UnlockPrinter

プリンターへのデータ送信及びリセット制御のロックを解除します。

```
ErrorCode UnlockPrinter()
```

戻り値

エラーコードを返します。エラーコードの内容は、「5章 エラーコード一覧」を参照してください。

備考

- **LockPrinter**を複数回呼び出した場合、ロック解除には同じ回数分の本APIの呼び出しが必要です。

DirectIOEx

- (a)のAPIは、バイナリデータを送信後、受信データをプリンターからバイナリデータとして取得します。
(b)のAPIは、バイナリデータを送信後、受信データをプリンターから文字列データとして取得します。
(c)のAPIは、バイナリデータを送信します。

(a) ErrorCode **DirectIOEx**(

byte[] *cmd*,
ref byte[] *data*,
int *timeout*,
bool *readFlag*,
byte *option*)

(b) ErrorCode **DirectIOEx**(

byte[] *cmd*,
out string *data*,
int *timeout*,
byte *option*)

(c) ErrorCode **DirectIOEx**(

byte[] *cmd*,
int *timeout*)

パラメーター

cmd

送信データバッファー

送信するデータが格納されたバッファーを指定します。

data

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。

受信データサイズは最大4096バイトです。

4096バイトよりも大きい値を指定した場合は4096バイトに設定されます。

データ取得が必要のない場合は、0バイトを指定します。

APIから制御が戻ると受信されたデータサイズが格納されます。

timeout

タイムアウト時間

本APIの成功を待つ時間を、msec(ミリ秒)単位で指定します。

設定範囲は3000ms～90000msです。

3000ms未満の値を指定した場合は3000msに設定されます。

90000msよりも大きい値を指定した場合は90000msに設定されます。

readFlag

受信動作フラグ

受信動作を指定するフラグを以下から指定します。

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続けます。

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

option

受信対象オプション

受信対象のデータを以下から指定します。

0 : ASB設定コマンドの応答を除いたデータを取得します。

1 : ASB設定コマンドの応答を含めたデータを取得します。

戻り値

エラーコードを返します。エラーコードの内容は、「5章 エラーコード一覧」を参照してください。

備考

- 本APIは**ResetPrinter**で中断されます。
- 本APIでASB設定コマンドを無効に設定するデータを含まないでください。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。
- Bluetooth接続の場合は、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。プリンターをハードウェアリセットする場合は、**ResetPrinter**を使用してください。
- ref byte[] *data*は本APIが成功した場合、呼び出し前の指定サイズを限度に、受信データのサイズにリサイズされます。
- 受信データに0x02が含まれる場合、0x02は0x5fに変換されます。

ResetPrinter

プリンターをハードウェアリセットします。

ErrorCode **ResetPrinter**()

戻り値

エラーコードを返します。エラーコードの内容は、「5章 エラーコード一覧」を参照してください。

備考

- 通信プロトコルを使用した(プリンターコマンドを使用しない)ハードウェアリセットを行います。
- 本APIを呼び出した場合、以下のAPIは中断されます。
 - **DirectIOEx**
 - **DirectSendRead**
- 本APIを実行した後、数秒経過してからデータを送信してください。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実行中のプリンターステータスは、「応答なし」の応答になります。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。

SetStatusBack

StatusCallbackイベントによるプリンターステータスの通知を開始します。

ErrorCode **SetStatusBack()**

戻り値

エラーコードを返します。エラーコードの内容は、「5章 エラーコード一覧」を参照してください。

備考

- 本APIで開始したStatusCallbackイベントは、以下のAPIで停止されます。
 - **CancelStatusBack**
 - **CloseMonPrinter**
- StatusCallbackイベントにイベントハンドラーを登録していない場合、本APIは失敗します。
- 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プリンターステータスになります。
- 本APIによりプリンターステータスの通知を開始すると、現在のプリンターステータスでStatusCallbackイベントが発生します。
- 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、登録済みのメソッドは無効となり、新しいコールバック関数が登録されます。
- すでにStatusCallbackイベントを開始済みで再度本APIを呼び出した場合も、直後のプリンターステータス応答は行われます。
- プリンターステータスの受信からイベント発生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。

CancelStatusBack

StatusCallbackイベントによるプリンターステータスの通知を停止します。

ErrorCode **CancelStatusBack()**

戻り値

エラーコードを返します。エラーコードの内容は、「5章 エラーコード一覧」を参照してください。

備考

- **SetStatusBack**によりプリンターステータスの通知が開始されていない場合でも本APIは成功します。

PowerOff

プリンターを電源オフ状態にします。

ErrorCode **PowerOff**()

戻り値

エラーコードを返します。エラーコードの内容は、「5章 エラーコード一覧」を参照してください。

備考

- 本APIを実行すると、プリンターで電源オフに伴う処理が実行されます。

GetCounter

メンテナンスカウンターを取得します。

(a)のAPIは、カウンターをCounterIndexに定義されたカウンターIDで取得します。

(b)のAPIは、カウンターをID指定で取得します。

(a) ErrorCode **GetCounter**(
CounterIndex *index*,
bool *type*,
out int *data*)

(b) ErrorCode **GetCounter**(
byte *index*,
out int *data*)

パラメーター

index

カウンターID

取得するカウンターIDを指定します。

SII.SDK.MposPrinter.CounterIndexに定義されたカウンターIDを使用する場合は、(a)を指定してください。

カウンターIDを指定する場合は、(b)を指定してください。

指定可能な内容は、「6.2 カウンターID」を参照してください。

type

メンテナンスカウンターのタイプ

取得するメンテナンスカウンターのタイプを以下から指定します。

TRUE : 積算カウンター

FALSE : 初期化可能カウンター

data

カウンター変数

取得するカウンター値を格納する変数を指定します。

戻り値

エラーコードを返します。エラーコードの内容は、「5章 エラーコード一覧」を参照してください。

ResetCounter

メンテナンスカウンターを初期化します。

(a)のAPIは、カウンターをCounterIndexに定義されたカウンターIDで初期化します。

(b)のAPIは、カウンターをID指定で初期化します。

(a) ErrorCode **ResetCounter**(
CounterIndex *index*)

(b) ErrorCode **ResetCounter**(
byte *index*)

パラメーター

index

カウンターID

初期化するカウンターIDを指定します。

SII.SDK.MposPrinter.CounterIndexに定義されたカウンターIDを使用する場合は、(a)を指定してください。

カウンターIDを指定する場合、(b)を指定してください。

指定可能な内容は、「6.2 カウンターID」を参照してください。

戻り値

エラーコードを返します。エラーコードの内容は、「5章 エラーコード一覧」を参照してください。

備考

- 本APIで指定したカウンターIDの値が初期化されたことを**GetCounter**で確認してください。

GetType

プリンターの各種IDを取得します。

ErrorCode **GetType**(
out byte *typeId*,
out byte *fontID*,
out byte *exrom*,
out byte *special*)

パラメーター

typeId

ID1

1(固定)が格納されます。

fontID

ID2

2(固定)が格納されます。

exrom

予約

NULLを指定してください。

special

予約

NULLを指定してください。

戻り値

エラーコードを返します。エラーコードの内容は、「5章 エラーコード一覧」を参照してください。

GetPrnCapability

プリンター情報を取得します。

(a)のAPIは、応答データをバイナリデータとして取得します。

(b)のAPIは、応答データを文字列データとして取得します。

(a) ErrorCode **GetPrnCapability**(

byte *id*,

out byte[] *data*)

(b) ErrorCode **GetPrnCapability**(

byte *id*,

out string *data*)

パラメーター

id

プリンターID

取得するプリンターIDを指定します。

取得するプリンターIDの応答形式がHEXコードの場合は、(a)を指定してください。

取得するプリンターIDの応答形式がASCII文字列の場合は、(b)を指定してください。

指定可能な内容及び応答形式は、「6.3 プリンターID」を参照してください。

data

受信データバッファー

取得するプリンターIDの値を格納するバッファーを指定します。

戻り値

エラーコードを返します。エラーコードの内容は、「5章 エラーコード一覧」を参照してください。

SendDataFile

コマンド定義ファイルの内容を、SDK内部のメモリに登録します。

```
ErrorCode SendDataFile(  
    string fileName )
```

パラメーター

fileName

コマンド定義ファイル名

決められた書式で作成されたコマンド定義ファイル名を指定します。

戻り値

エラーコードを返します。エラーコードの内容は、「5章 エラーコード一覧」を参照してください。

備考

- コマンド定義ファイルの詳細は、「7章 コマンド定義ファイル」を参照してください。
- 本APIによって登録されたコマンド定義は、**CloseMonPrinter**で破棄されます。

DirectSendRead

(a)のAPIは、**SendDataFile**実行後、受信データをプリンターからバイナリデータとして取得します。

(b)のAPIは、**SendDataFile**実行後、受信データをプリンターから文字列として取得します。

(c)のAPIは、**SendDataFile**で登録したコマンド定義ファイルの内容を実行します。

```
(a) ErrorCode DirectSendRead(  
    string cmdName,  
    string readType,  
    ref byte[] data,  
    int timeout,  
    bool readFlag )
```

```
(b) ErrorCode DirectSendRead(  
    string cmdName,  
    string readType,  
    out string data,  
    int timeout )
```

```
(c) ErrorCode DirectSendRead(  
    string cmdName,  
    string readType,  
    int timeout )
```

パラメーター

cmdName

コマンド名

SendDataFileで定義したコマンド名を指定します。

readType

受信データタイプ名

受信データのタイプを以下から指定します。

ASB: ASB設定コマンドの応答のみを受信データに保存します。

Other: 上記以外のプリンターからの応答を受信データに保存します。

data

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。

timeout

タイムアウト時間

本APIの成功を待つ時間を、msec(ミリ秒)単位で指定します。

設定範囲は3000ms～90000msです。

3000ms未満の値を指定した場合は3000msに設定されます。

90000msよりも大きい値を指定した場合は90000msに設定されます。

readFlag

受信動作フラグ

受信動作を指定するフラグを以下から指定します。

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続けます。

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

戻り値

エラーコードを返します。エラーコードの内容は、「5章 エラーコード一覧」を参照してください。

備考

- 本APIでASB設定コマンドを無効に設定するデータを含まないでください。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。
- 本APIは**ResetPrinter**により中断されます。
- Bluetooth接続の場合は、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。プリンターをハードウェアリセットする場合は、**ResetPrinter**を使用してください。
- 受信データに0x02が含まれる場合、0x02は0x5fに変換されます。

GetProperty

印刷設定内容(一部)を取得します。

```
ErrorCode GetProperty(  
    IntPtr devmode,  
    byte id,  
    byte[] data,  
    ref uint size )
```

パラメーター

devmode

DevModeアドレス

Devmodeアドレスを指定します。

id

プロパティID

取得するプロパティIDを指定します。

指定可能な内容は、「6.4 プロパティID」を参照してください。

data

取得データバッファ

取得するプロパティIDの内容を格納するバッファを指定します。

size

取得データサイズ

取得するデータの最大長を指定します。

APIから制御が戻ると取得したデータサイズが格納されます。

戻り値

エラーコードを返します。エラーコードの内容は、「5章 エラーコード一覧」を参照してください。

SetProperty

印刷設定内容(一部)を変更します。

```
ErrorCode SetProperty(  
    IntPtr devmode,  
    byte id,  
    byte[] data,  
    uint size )
```

パラメーター

devmode

DevModeアドレス

Devmodeアドレスを指定します。

id

プロパティID

変更するプロパティIDを指定します。
指定可能な内容は、「6.4 プロパティID」を参照してください。

data

設定データバッファ

変更するプロパティIDの内容が格納されたバッファを指定します。

size

設定データサイズ

変更するプロパティIDの内容が格納されたバッファサイズを指定します。

戻り値

エラーコードを返します。エラーコードの内容は、「5章 エラーコード一覧」を参照してください。

4.4.3 イベント

StatusCallback

応答されたプリンターステータスを通知します。

event StatusCallbackHandler **StatusCallback**

delegate void **StatusCallbackHandler**(
ASB *status*)

パラメーター

status

プリンターステータス変数

プリンターステータスを格納する変数を指定します。

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。

備考

- プリンターステータスの通知を開始する場合は**SetStatusBack**を呼び出してください。
- **SetStatusBack**によりプリンターステータスの通知を開始すると、現在のプリンターステータスでイベントが発生します。
- プリンターステータスの通知を開始する前に、応答されたプリンターステータスを処理するイベントハンドラーを登録してください。
- イベントハンドラーを登録していない場合、**SetStatusBack**は失敗します。
- 本イベントを停止する場合は以下のAPIを呼び出してください。
 - **CancelStatusBack**
 - **CloseMonPrinter**
- 登録されたイベントハンドラー内から同一インスタンスで.NET API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プリンターステータスになります。
- 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と比べ変化がない場合、イベントは発生しません。
- プリンターステータスの受信からイベント発生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。

5章 エラーコード一覧

本章ではエラーコードについて記載します。

5.1 エラーコード一覧

主なエラーコードは以下の通りです。

マクロ定義(定数)	値	内容
SUCCESS	0	正常
ERR_TYPE	-10	オープンタイプのパラメーターエラーです。
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_PRINTER	-30	指定したプリンタードライバがありません。
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_FUNCTION	-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が起動していません。

マクロ定義(定数)	値	内容
ERR_LOCKED	-1000	プリンターがロックされています。
ERR_UNLOCKED	-1010	プリンターがロックされていません。
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_PRINTER_HAS_JOBS_QUEUED	-1060	プリンターに印刷待ちジョブがあります。
ERR_UNKNOWN_PORT	-1070	対象外のポートです。
ERR_INVALID_PRINTER_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。

6章 引数情報

本章では引数について記載します。

6.1 プリンタステータス一覧

プリンタステータスの応答内容の対応ビットは以下の通りです。サンプルプログラムのプリンタステータス処理も参考にしてください。

プリンタステータス	対応ビット		内容
電圧エラー	ASB.VP_ERR	0x00000000	電圧エラーなし
		0x00000001	電圧エラーあり
ハードウェアエラー	ASB.HARDWARE_ERR	0x00000000	ハードウェアエラーなし
		0x00000002	ハードウェアエラーあり
ヘッド温度エラー	ASB.HEAD_TEMPERATUR_ERR	0x00000000	ヘッド温度エラーなし
		0x00000004	ヘッド温度エラーあり
紙無しエラー	ASB.RECEIPT_END	0x00000000	紙無しエラーなし
		0x00000010	紙無しエラーあり
フィードスイッチの状態	ASB.PAPER_FEED	0x00000000	フィードスイッチの状態 = "Off"
		0x00000100	フィードスイッチの状態 = "On"
紙送り状態	ASB.NOW_PRINTING	0x00000000	停止
		0x00000400	動作中
復帰待ち状態	ASB.RETURN_WAITING	0x00000000	–
		0x00000800	復帰待ち状態
フラッシュメモリ書き換え中	ASB.FLASH_MEMORY_REWRITING	0x00000000	–
		0x00010000	フラッシュメモリ書き換え中

プリンターステータス	対応ビット		内容
バッテリー残量レベル※1	ASB.BATTERY_LOW	0x00100000	Low : 0x00100000
	ASB.BATTERY_MIDDLE	0x00200000	Middle : 0x00300000
	ASB.BATTERY_FULL	0x00400000	Full : 0x00700000
バッテリーエラー	ASB.BATTERY_ERR	0x00000000	バッテリーエラーなし
		0x00800000	バッテリーエラーあり
自動復帰エラー	ASB.AUTORECOVER_ERR※2	0x00000000	自動復帰エラーなし
		0x20000000	自動復帰エラーあり
復帰不可能エラー	ASB.UNRECOVER_ERR※2	0x00000000	復帰不可能エラーなし
		0x40000000	復帰不可能エラーあり
応答なし	ASB.NO_RESPONSE※2※3	0x00000000	プリンター応答
		0x80000000	未接続、または通信エラー

※1: Full→Middle→Low の順にビットを確認してください(例えば Full の場合、Middle 及び Low のビットもセットされます)。いずれのビットもクリアされている場合は、バッテリー残量は無しとなります。

※2: 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。

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

6.2 カウンターID

カウンターIDの応答内容は以下の通りです。

カウンターID		内容	ResetCounter/ ResetCounterによる初期化
ROLL_FEED_LINES	20	紙送りドットライン数 (100ドットライン単位)	可
ROLL_HEAD_CHARGE	21	サーマルヘッド通電回数 (100ドットライン単位)	可
OPERATION_TIME	70	製品の駆動時間(分単位)	可
ROLL_FEED_LINES	148	紙送りドットライン数 (100ドットライン単位)	不可(積算)
ROLL_HEAD_CHARGE	149	サーマルヘッド通電回数 (100ドットライン単位)	不可(積算)
OPERATION_TIME	198	製品の駆動時間(分単位)	不可(積算)

6.3 プリンターID

プリンターID及びその応答内容は以下の通りです。

プリンターID	内容		応答形式
1	機種ID	0x21	HEXコード
2	タイプID	0x30	HEXコード
3	ROMバージョンID	ROMバージョンによる	HEXコード
65	ファームウェアバージョン (main)	x.xx.xx	ASCII文字列
66	メーカー名	Seiko Instruments Inc.	ASCII文字列
67	モデル名	SII MP-B20 Series.	ASCII文字列
69	多国語フォントの種類	KANJI JAPANESE (日本語仕様)	ASCII文字列
97	ファームウェアバージョン (boot)	x.xx.xx	ASCII文字列
98	ファームウェアチェックサム (boot)	2バイトチェックサム	HEXコード
99	ファームウェアチェックサム (main)	2バイトチェックサム	HEXコード
100	ファームウェアチェックサム (main + boot)	2バイトチェックサム	HEXコード

6.4 プロパティID

注意

- ◆ SetPropertyを使用する場合、カスタムコマンド以外はデータサイズを1バイトで指定してください。
- ◆ []カッコ内は当該処理が行われるタイミングを示します。

プロパティID及びその内容は以下の通りです。

プロパティID	内容	
1	初期化	0 : する 1 : しない
3	余白	2 : 下余白最小 3 : 最大余白
4	濃度(パーセント)	70 ~ 130
5	方向	0 : 順 1 : 逆
6	縮小	0 : 縮小しない 20 ~ 100 : 指定倍率(パーセント)
9	カット位置までフィード	0 : する 1 : しない
10	[印刷開始] ロゴ	0 : なし 1 : 左寄せ 2 : 中央 3 : 右寄せ
11	[印刷開始] ロゴキーコード	0 ~ 99
15	[印刷開始] カスタムコマンド(128バイト)	コマンドデータ
17※1	[印刷開始] 紙送り(フィード)(dot)	0 ~ 255
20	[ページ開始] ロゴ	0 : なし 1 : 左寄せ 2 : 中央 3 : 右寄せ
21	[ページ開始] ロゴキーコード	0 ~ 99
25	[ページ開始] カスタムコマンド(128バイト)	コマンドデータ
27※1	[ページ開始] 紙送り(フィード)(dot)	0 ~ 255

プロパティID	内容	
30	[ページ終了] ロゴ	0 : なし 1 : 左寄せ 2 : 中央 3 : 右寄せ
31	[ページ終了] ロゴキーコード	0 ~ 99
35	[ページ終了] カスタムコマンド(128バイト)	コマンドデータ
37※1	[ページ終了] 紙送り(フィード)(dot)	0 ~ 255
40	[印刷終了] ロゴ	0 : なし 1 : 左寄せ 2 : 中央 3 : 右寄せ
41	[印刷終了] ロゴキーコード	0 ~ 99
45	[印刷終了] カスタムコマンド(128バイト)	コマンドデータ
47※1	[印刷終了] 紙送り(フィード)(dot)	0 ~ 255
50	用紙サイズ	0 : Letter 1 : A4 3 : 58 mm 4 ~ 255 : 上記以外の用紙※2
51	印刷の向き	0 : 縦 1 : 横
52	カラー印刷モード	0 : システム※3 1 : ドライバー※3
60※4	プリセット	1 : 58 mmレシート設定 5 : A4 → 58 mm縮小設定 6 : ユーザー設定

※1: 同じタイミングのフィードは、最後の設定が有効になります。

※2: 設定することはできません。取得のみ行えます。

※3: 詳細は、MP-B20 シリーズ用「SII Printer Driver for Windows 取扱説明書」の「3.2 用紙・品質」を参照してください。

※4: 本項目を指定した場合、他のプロパティIDの一部の内容が無視されます。詳細は、MP-B20 シリーズ用「SII Printer Driver for Windows 取扱説明書」の「3.4.4 プリセットの利用」を参照してください。

7章 コマンド定義ファイル

7.1 概要

コマンド定義ファイルとは、送信データを決められたフォーマットで記述し、ANSI形式及びUNICODE形式で保存されたテキストファイル(*.txt)です。

7.2 フォーマット

コマンド定義ファイルのフォーマットを説明します。

Command definition

```
[Command name 1] = [Transmission data 1]#[Comment 1]
[Command name 2] = [Transmission data 2]#[Comment 2]
[Command name 3] = [Transmission data 3]#[Comment 3]
:
:
```

名称	説明
Command definition コマンド定義	フォーマットに沿って記述したコマンド名単位の水データです。 必ず1コマンド定義に対して1行で記述してください。
Command name コマンド名	DirectSendRead で指定する任意のコマンド名を記述します。 <ul style="list-style-type: none">・ “=”、“#”以外のASCII文字を使用してください。ASCII文字以外を記述した場合、その文字だけ無視されます。・ 大文字、小文字は区別されます。・ 登録可能なコマンド名は最大33バイトです。34バイト以上記述した場合、33バイトで登録され、34バイト以降の文字は無視されます。・ 1個のコマンド定義ファイルの中で同じコマンド名を記述している場合、最初のコマンド名が登録されます。すでに登録されているコマンド名を指定した場合は無視されます。・ コマンド名と送信データの間には“=”を記述してください。

名称	説明
Transmission data 送信データ	<p>プリンターに送信する、プリンターコマンド及びデータを記述します。</p> <ul style="list-style-type: none"> 送信データにバイナリデータを指定する場合は、データは2桁の16進数で記述してください。各データの間には半角スペースを挿入してください。 バイナリデータには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。プリンターをハードウェアリセットする場合は、Reset または ResetPrinter を使用してください。 送信データに文字列を指定する場合は、ASCII文字で記述してください。 文字列を指定する場合は、""で囲います。 送信データのデータサイズは最大10240バイトです。ただし、送信データを文字列で指定した場合のサイズは、指定した文字列をバイナリデータに変換した後のサイズです。 後述のコマンド定義例の送信データサイズは以下の通りです。 <ul style="list-style-type: none"> CmdName_1 : 3バイト CmdName_2 : 4バイト CmdName_3 : 3バイト CmdName_4 : 6バイト
Comment コメント	<p>コマンド定義の説明などを記述します。</p> <ul style="list-style-type: none"> コメントの先頭に"#"を記述してください。 記述する文字の制限はありません。 コメントは省略可能です。

コマンド定義例:

CmdName_1="SII"#Comments1

CmdName_2=53 49 49 0A

CmdName_3=1D 56 00#Feed the paper to cut position.

CmdName_4=1B 40 "SII" 0A#Initialize the printer, the strings "SII" and line feed.

注意

- ◆ コマンド定義ファイルのファイルサイズは、最大4GBです。
- ◆ 登録可能なコマンド定義数は、システムの使用可能メモリによって異なります。

7.3 利用方法

コマンド定義ファイルの利用方法を説明します。

1. コマンド定義ファイルを作成してください。
2. コマンド定義ファイルを**SendDataFile**でSDK内部のメモリに登録してください。
3. コマンド名を**DirectSendRead**で指定すると、送信データの内容が実行されます。

注意

- ◆ 登録したコマンド定義は、**CloseMonPrinter**で破棄されます。
- ◆ コマンド定義ファイル登録後に、新たなコマンド定義ファイルを登録すると、コマンド定義ファイルの内容は上書きされます。



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>

大阪支店

大阪府豊中市新千里西町 1-1-4 千里中央ツインビル別館

〒560-0083

電話番号：06-6871-9483（直通） ファクシミリ：06-6871-9474

©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。
