



RP-E10 シリーズ
通信ライブラリ
アプリケーションプログラマーズガイド

U00127038310

本書の API は将来的に非サポートとなる API です。
新しく開発を行う場合は、POS シリーズ用「SII SDK for Windows
アプリケーションプログラマーズガイド」をご覧ください。

セイコーインスツル株式会社

U00127038300	2012 年	5 月
U00127038301	2012 年	7 月
U00127038302	2012 年	11 月
U00127038303	2013 年	2 月
U00127038304	2013 年	12 月
U00127038305	2014 年	6 月
U00127038306	2015 年	3 月
U00127038307	2016 年	6 月
U00127038308	2019 年	8 月
U00127038309	2021 年	5 月
U00127038310	2024 年	11 月

©セイコーインスツル株式会社 2012-2024

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RP-E10シリーズ向けプリンタードライバー(以降:プリンタードライバー)上で動作する通信ライブラリ(以降:通信ライブラリ)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

通信ライブラリ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。

- RP-E10シリーズ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以下に記載します。

用語	内容
技術説明書	RP-E10シリーズサーマルプリンタ技術説明書。
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」。 詳細は技術説明書を参照してください。
POSプリンターステータス	通信ライブラリ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。 詳細は『6.1 POS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	3-3
RpOpenMonPrinter	3-3
RpCloseMonPrinter	3-4
RpLockPrinter	3-4
RpUnlockPrinter	3-5
RpDirectIO	3-6
RpDirectIOEx	3-7
RpResetPrinter	3-9
RpGetStatus	3-9
RpSetStatusBackFunction	3-10
RpSetStatusBackWnd	3-11
RpCancelStatusBack	3-12
RpPowerOff	3-13
RpGetCounter	3-13
RpResetCounter	3-14
RpGetType	3-15
RpGetPrnCapability	3-16
RpOpenDrawer	3-17
RpSendDataFile	3-17
RpDirectSendRead	3-18
RpGetProperty	3-19
RpSetProperty	3-20
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
4.5 プロパティ	4-3
Status	4-3
LastError	4-3
IsValid	4-4

4.6	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.....	4-8
	CancelStatusBack.....	4-9
	PowerOff.....	4-9
	GetCounter.....	4-9
	ResetCounter.....	4-10
	GetType.....	4-11
	GetPrnCapability.....	4-12
	OpenDrawer.....	4-12
	SendDataFile.....	4-13
	DirectSendRead.....	4-14
	GetProperty.....	4-15
	SetProperty.....	4-16
4.7	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	POS プリンターステータス一覧	6-1
6.2	カウンターID	6-3
6.3	タイプ ID	6-3
6.4	フォントタイプ	6-4
6.5	プリンターID	6-4
6.6	対象ドロワ	6-4
6.7	ドロワキック時間	6-5
6.8	プロパティ ID	6-5

7 章 コマンド定義ファイル 7-1

7.1	フォーマット	7-1
7.2	制限事項	7-2

1章 概要

1.1 はじめに

本章では通信ライブラリの概要について説明します。

通信ライブラリとは、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ライブラリです。

通信ライブラリは、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。プリンタードライバーとは別にインストールが必要です。

通信ライブラリの利用により、アプリケーション開発においてプリンターを直接制御することと、ポートの種類に依存しない設計をすることができます。

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。

通信ライブラリの利用例として、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

原則、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取扱説明書を参照してください。

また、上記以外に、以下の動作条件を満たす必要があります。

- .NET Framework Version 2.0以上のインストール
- シリアル通信で利用する場合、[フロー制御]設定が「ハードウェア」であること
(詳細は、技術説明書を参照してください。)
- 双方向サポートの機能が有効であること
(設定方法は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。)
- プリンターコマンドの「周辺機器の選択」においてプリンターを無効に選択していないこと
(詳細は、技術説明書を参照してください。)
- 以下のすべてに該当する場合は、共有プリンターを利用すること
(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

2章 インストール

インストールについては、「SII ソフトウェアパッケージ 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開発環境用通信ライブラリ(Win32 API)について説明します。

3.2 開発対象言語

開発対象言語は以下の通りです。

- Visual C++

3.3 ライブラリファイル

通信ライブラリのファイル名は以下の通りです。

- SiiRpe1Api.dll

通信ライブラリのファイル保管場所は、Windowsのシステムフォルダーです。

特殊な場合を除き、通信ライブラリ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ので、フォルダーから移動せずにそのまま使用してください。

通信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通信ライブラリが正しく更新されません。

3.4 API一覧

通信ライブラリに実装されているAPIは以下の通りです。

API	機能概略
RpOpenMonPrinterA ^{※1} RpOpenMonPrinterW ^{※1}	指定したプリンターで通信ライブラリの使用を開始し、APIハンドルを返します。
RpCloseMonPrinter	指定したAPIハンドルでの通信ライブラリの使用を終了します。
RpLockPrinter ^{※2}	他のプロセスからのプリンターへのデータ送信およびハードウェアリセット要求を全て禁止(ロック)します。
RpUnlockPrinter	RpLockPrinterによる他プロセスからのアクセス禁止(ロック)を解除します。
RpDirectIO ^{※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)
RpDirectIOEx ^{※2、※3}	バイナリデータを送信及び受信します。
RpResetPrinter ^{※2、※3}	プリンターをリセットします。
RpGetStatus	最新のPOSプリンターステータスを取得します。
RpSetStatusBackFunction	POS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。
RpSetStatusBackWnd	POSプリンターステータスの変化を検出すると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、POSプリンターステータスをセットする変数を登録します。
RpCancelStatusBack	RpSetStatusBackWnd及びRpSetStatusBackFunctionにて実行されたコールバック関数の登録を解除します。
RpPowerOff ^{※2、※3}	プリンターを電源オフ状態にします。
RpGetCounter ^{※2、※3}	指定したメンテナンスカウンターを取得します。
RpResetCounter ^{※2、※3}	指定したメンテナンスカウンターを初期化します。
RpGetType ^{※2、※3}	プリンターのタイプIDとフォントタイプを取得します。
RpGetPrnCapability ^{※2、※3}	指定したプリンター情報を取得します。
RpOpenDrawer ^{※2、※3}	指定したドロワをオープンします。
RpSendDataFileA ^{※1} RpSendDataFileW ^{※1}	コマンド定義ファイルを指定し、コマンドを登録します。
RpDirectSendReadA ^{※1、※2、※3} RpDirectSendReadW ^{※1、※2、※3}	RpSendDataFileで登録したコマンドを実行します。
RpGetProperty	指定したプロパティIDを取得します。
RpSetProperty	指定したプロパティID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)を利用する場合は末尾に' A '、UNICODE(ユニコード)を利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RpLockPrinter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、本APIは失敗します。

3.5 API 詳細

注意

- ◆ Bluetooth接続の場合、一時的にでも接続が切れてしまった際には、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出来ません。

RpOpenMonPrinter

指定したプリンターで通信ライブラリの使用を開始し、APIハンドルを返します。

```
INT RpOpenMonPrinter(  
    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
通信ライブラリ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

成功時はプリンターを識別するAPIハンドルを返します。

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APIハンドルを利用しなくなった場合、必ずRpCloseMonPrinter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FILEの場合、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。

RpCloseMonPrinter

指定したAPIハンドルでの通信ライブラリの使用を終了します。

```
INT RpCloseMonPrinter(  
    INT i_hdl )
```

パラメーター

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

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

- RpSetStatusBackFunction及びRpSetStatusBackWndによるPOSプリンターステータスの監視を停止します。
- 本APIで呼び出したAPIハンドルが他のAPIで利用中の場合、その処理が完了するまで本APIは実行されません。
- RpSendDataFileによって登録したコマンドは破棄されます。

RpLockPrinter

他のプロセスからのプリンターへのデータ送信およびハードウェアリセット要求を全て禁止(ロック)します。

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

パラメーター

i_hdl
APIハンドル
RpOpenMonPrinter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
タイムアウト時間をmsec(ミリ秒)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

- 本APIを実行するとプリンターへのデータ送信及びリセット制御を占有します。これを解除するためにはRpUnlockPrinterを使用してください。
- 本APIを呼び出してから、RpUnlockPrinter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を利用すると失敗します。
- 本APIによる占有はプロセス内で有効です。よって、占有している間、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。
- すでに占有しているAPIハンドルでの本APIの繰り返し回数は最大99回です。占有状態を解除するには本APIを実行した回数と同じ回数のRpUnlockPrinterを実行してください。
- i_timeoutの設定範囲は3000 ms ~ 90000 msです。3000 ms未満の値を指定した場合は3000 msに、90000 msよりも大きい値を指定した場合は90000 msに設定されます。

RpUnlockPrinter

RpLockPrinterによる他プロセスからのアクセス禁止（ロック）を解除します。

```
INT RpUnlockPrinter(  
    INT i_hdl )
```

パラメーター

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

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

- RpLockPrinterで設定したプリンターのロックを解除します。
- RpLockPrinter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。

RpDirectIO

バイナリデータを送信及び受信します。

```
INT RpDirectIO(  
    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

RpOpenMonPrinter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

送信するデータを格納し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”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

タイムアウト時間をmsec(ミリ秒)単位で指定します。

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：リード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。

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RpResetPrinterにより中断されます。
- i_timeoutの設定範囲は3000 ms ~ 90000 msです。3000 ms未満の値を指定した場合は3000 msに、90000 msよりも大きい値を指定した場合は90000 msに設定されます。
- 複数のプロセスから通信ライブラリが利用され、かつ、送信するデータを分割し、本APIを複数回使用してデータ送信を行った場合、途中で別プロセスからの意図しない出力が割り込む可能性があります。特にイメージデータなど途中で他のデータの割り込みが許されないプリンターコマンドやデータを出力する場合、RpLockPrinterを呼び出した後、本APIを利用してください。
- 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RpDirectIOExを実行してください。
- 送信するバイナリデータにASB設定コマンドを無効に設定するデータを含めないでください。ASB設定コマンドを無効に設定するデータが含まれるとPOSプリンターステータスを取得するAPIは正常に動作しません。
- Bluetooth接続の場合、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。
ハードウェアリセットを行う場合はRpResetPrinterを利用してください。

RpDirectIOEx

バイナリデータを送信及び受信します。

```
INT RpDirectIOEx(  
    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
RpOpenMonPrinter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
送信するデータを格納し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。
データ取得が必要のない場合は、“0”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

タイムアウト時間をmsec(ミリ秒)単位で指定します。

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: リード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。

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

1: ASB設定コマンドの応答を含めたデータを取得

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RpResetPrinterにより中断されます。
- i_timeoutの設定範囲は3000 ms ~ 90000 msです。3000 ms未満の値を指定した場合は3000 msに、90000 msよりも大きい値を指定した場合は90000 msに設定されます。
- 複数のプロセスから通信ライブラリが利用され、かつ、送信するデータを分割し、本APIを複数回使用してデータ送信を行った場合、途中で別プロセスからの意図しない出力が割り込む可能性があります。特にイメージデータなど途中で他のデータの割り込みが許されないプリンターコマンドやデータを出力する場合、RpLockPrinterを呼び出した後、本APIを利用してください。
- 受信データサイズは最大4096バイトです。これを超える値を指定した場合、4096バイトに設定されます。
- 送信するバイナリデータにASB設定コマンドを無効に設定するデータを含めないでください。ASB設定コマンドを無効に設定するデータが含まれるとPOSプリンターステータスを取得するAPIは正常に動作しません。
- Bluetooth接続の場合、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。ハードウェアリセットを行う場合はRpResetPrinterを利用してください。

RpResetPrinter

プリンターをリセットします。

```
INT RpResetPrinter(  
    INT i_hdl )
```

パラメーター

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

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

- 通信プロトコルを利用した(プリンターコマンドを利用しない)ハードウェアリセットを行います。
- RpDirectIO、RpDirectIOExまたはRpDirectSendReadを実行中に本APIを呼び出した場合、それらのAPIは中断されます。
- 本APIでハードウェアリセットを行った後、数秒経過してからデータ送信をしてください。ハードウェアリセット中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、POSプリンターステータスは未接続の応答になります。
- 本関数実行時にプリンターがデータを受け付けられない状態になっていた場合、プリンターのリセット処理が正常に実施されず、印字化け等が発生する可能性があります。

RpGetStatus

最新のPOSプリンターステータスを取得します。

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

パラメーター

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

o_status

POSプリンターステータス変数

POS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、その時点で最後に受信しているPOSプリンターステータスを応答します。
- プリンターが接続されていても、POSプリンターステータスが取得できていない場合、本APIは失敗します。
- POSプリンターステータスの内容については『6.1 POSプリンターステータス一覧』を参照してください。

RpSetStatusBackFunction

POS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 RpSetStatusBackFunction(  
    INT i_hdl,  
    INT ( CALLBACK EXPORT *lpStatusCB ) ( DWORD o_st ) )
```

パラメーター

i_hdl

APIハンドル

RpOpenMonPrinter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

POS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。

o_st

POSプリンターステータス変数

POS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

- 本APIで登録されたコールバック関数はRpCancelStatusBack、RpCloseMonPrinterで解除されます。また、lpStatusCBにNULLを指定しても解除されます。

- 登録されたコールバック関数内から同一のAPIハンドルで通信ライブラリのAPIを呼び出すことはできません。
- プリンターが接続されていてもPOSプリンターステータスが取得できていない場合、本APIは失敗します。
- 再接続を検出した場合、その時点で最後に受信しているPOSプリンターステータスを応答します。
- POS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POS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。
- 本APIによりコールバック関数を登録すると、現在のPOS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。
- すでにコールバック関数を登録した状態で再度本APIを呼び出した場合、登録されているAPIは無効になり、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POS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。
- POS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については保証されません。
- POSプリンターステータスの内容については『6.1 POSプリンターステータス一覧』を参照してください。

RpSetStatusBackWnd

POSプリンターステータスの変化を検出すると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、POS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 RpSetStatusBackWnd(
    INT i_hdl,
    HANDLE i_Wnd,
    LPDWORD o_status )
```

パラメーター

i_hdl

APIハンドル

RpOpenMonPrinter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。

o_status

POSプリンターステータス変数

POS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

- 本APIはRpCancelStatusBack、RpCloseMonPrinterで解除されます。また、ボタンのウィンドウハンドルにNULLを指定することでも解除されます。
- プリンターが接続されていてもPOSプリンターステータスが取得できていない場合、本APIは失敗します。
- 再接続を検出した場合、その時点で最後に受信しているPOSプリンターステータスを応答します。
- プリンターからPOS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POSプリンターステータスと比べ変化がない場合には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POS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。
- すでにボタンのウィンドウハンドルを登録した状態で再度本APIを呼び出した場合、登録されているボタンのウィンドウハンドルは無効になり、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POS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。
- POS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については保証されません。
- POSプリンターステータスの内容については『6.1 POSプリンターステータス一覧』を参照してください。

RpCancelStatusBack

RpSetStatusBackWnd及びRpSetStatusBackFunctionにて実行されたコールバック関数の登録を解除します。

```
INT RpCancelStatusBack(  
    INT i_hdl )
```

パラメーター

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

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

- RpSetStatusBackWnd及びRpSetStatusBackFunction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。

RpPowerOff

プリンターを電源オフ状態にします。

```
INT RpPowerOff(  
    INT i_hdl )
```

パラメーター

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

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。

RpGetCounter

指定したメンテナンスカウンターを取得します。

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

パラメーター

i_hdl
APIハンドル
RpOpenMonPrinter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。

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

- カウンターIDの内容については『6.2 カウンターID』を参照してください。

RpResetCounter

指定したメンテナンスカウンターを初期化します。

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

パラメーター

i_hdl

APIハンドル

RpOpenMonPrinter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。

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

- カウンターIDの内容については『6.2 カウンターID』を参照してください。
- 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RpGetCounterで確認してください。

RpGetType

プリンターのタイプIDとフォントタイプを取得します。

```
INT RpGetType(  
    INT i_hdl,  
    LPBYTE o_typeID  
    LPBYTE o_font  
    LPBYTE o_exrom  
    LPBYTE o_special )
```

パラメーター

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

o_typeID
タイプID変数
タイプIDを格納する変数を指定します。

o_font
フォントタイプ変数
フォントタイプを格納する変数を指定します。

o_exrom
予約
NULLを指定してください。

o_special
予約
NULL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

- タイプIDの内容については『6.3 タイプID』を参照してください。
- フォントタイプの内容については『6.4 フォントタイプ』を参照してください。

RpGetPrnCapability

指定したプリンター情報を取得します。

```
INT RpGetPrnCapability(  
    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

RpOpenMonPrinter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。

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。

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IDの内容については『6.5 プリンターID』を参照してください。
- io_datsizeで指定したサイズが指定したプリンターIDの応答サイズよりも小さい場合、またはo_datにNULLが指定された場合APIは失敗となり、指定したプリンターIDの応答サイズがio_datsizeに格納されます。

RpOpenDrawer

指定したドロワをオープンします。

```
INT RpOpenDrawer(  
    INT i_hdl,  
    BYTE i_drawer,  
    BYTE i_pulse )
```

パラメーター

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

i_drawer
対象ドロワ
使用する対象ドロワを指定します。

i_pulse
ドロワキック時間
対象ドロワのドロワキック時間を指定します。

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

- 対象ドロワの内容については『6.6 対象ドロワ』を参照してください。
- ドロワキック時間の内容については『6.7 ドロワキック時間』を参照してください。

RpSendDataFile

コマンド定義ファイルを指定し、コマンドを登録します。

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

パラメーター

i_hdl

APIハンドル

RpOpenMonPrinter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したコマンド定義ファイル名を指定します。

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

- コマンド定義ファイルについては『7章 コマンド定義ファイル』を参照してください。
- 本APIで登録したコマンドはRpCloseMonPrinterを呼び出すことにより破棄されます。

RpDirectSendRead

RpSendDataFileで登録したコマンドを実行します。

```
INT RpDirectSendRead(  
    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

RpOpenMonPrinterで取得したAPIハンドルを指定します。

i_cname

コマンド名

RpSendDataFileで定義したコマンド名を指定します。

i_rtype

受信データタイプ名

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。
データ取得が必要のない場合は、“0”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

タイムアウト時間をmsec(ミリ秒)単位で指定します。

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：リード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。

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

- ・送信するバイナリデータにASB設定コマンドを無効に設定するデータを含めないでください。ASB設定コマンドを無効に設定するデータが含まれるとPOSプリンターステータスを取得するAPIは正常に動作しません。
- ・本APIはRpResetPrinterにより中断されます。
- ・i_timeoutの設定範囲は3000 ms ～ 90000 msです。3000 ms未満の値を指定した場合は3000 msに、90000 msよりも大きい値を指定した場合は90000 msに設定されます。
- ・受信データサイズは最大4096バイトです。これを超える値を指定した場合、4096バイトに設定されます。
- ・Bluetooth接続の場合、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。
ハードウェアリセットを行う場合はRpResetPrinterを利用してください。

RpGetProperty

指定したプロパティIDの内容を取得します。

```
INT RpGetProperty(  
    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

RpOpenMonPrinter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。

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。

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。

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

- ・プロパティIDの内容については『6.8 プロパティID』を参照してください。
- ・io_sizeで指定したサイズが指定したプロパティIDの応答サイズよりも小さい場合、またはo_datにNULLを指定した場合には、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。

Rp SetProperty

指定したプロパティIDの内容を変更します。

```
INT RpSetProperty(  
    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

RpOpenMonPrinter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。

i_dat

設定データバッファ

変更するプロパティIDの内容を格納したバッファを指定します。

i_size

設定データサイズ

変更するプロパティIDの内容を格納したバッファサイズを指定します。

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

- プロパティIDの内容については『06.8 プロパティID』を参照してください。

4章 .NET API

4.1 概要

本章では.NET開発環境用通信ライブラリ(.NET API)について説明します。

4.2 開発対象言語

開発言語は以下の通りです。

- Visual Basic .NET
- Visual C#

4.3 ライブラリファイル

通信ライブラリのファイル名は以下の通りです。

- SiiRpe1ClassLib.dll

通信ライブラリのファイル保管場所は、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

通信ライブラリに実装されているAPIは以下の通りです。

- ネームスペース : SiiPrinterSdk
- クラス名 : StatusAPI

種別	API	機能概略
プロパティ	Status	最新のPOS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	オープン状態を取得します。
メソッド	OpenMonPrinter	指定したプリンターで通信ライブラリの使用を開始します。
メソッド	CloseMonPrinter	通信ライブラリの使用を終了します。
メソッド	LockPrinter	他のプロセスからのプリンターへのデータ送信およびハードウェアリセット要求を全て禁止(ロック)します。
メソッド	UnlockPrinter	LockPrinterによる他プロセスからのアクセス禁止(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	プリンターをリセットします。
メソッド	SetStatusBack	POS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メソッドを登録します。
メソッド	CancelStatusBack	SetStatusBackにて実行されたコールバックメソッドの登録を解除します。
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	指定したメンテナンスカウンタを取得します。
メソッド	ResetCounter	指定したメンテナンスカウンタを初期化します。
メソッド	GetType	プリンターのタイプIDとフォントタイプを取得します。
メソッド	GetPrnCapability	指定したプリンター情報を取得します。
メソッド	OpenDrawer	指定したドロワをオープンします。
メソッド	SendDataFile	コマンド定義ファイルを指定し、送信データを登録します。
メソッド	DirectSendRead	SendDataFileで定義したコマンドを実行します。
メソッド	GetProperty	指定したプロパティIDを取得します。
メソッド	SetProperty	指定したプロパティIDを変更します。
イベント	StatusCallback	応答されたPOSプリンターステータスを処理するイベントです。

4.5 プロパティ

Status

最新のPOSプリンターステータスを取得します。

```
SII.Driver.PosPrinter.ASB Status { get; }
```

初期値

0

備考

- 本プロパティの失敗はLastErrorで取得してください。失敗の場合、本プロパティの内容は不定です。
- 取得できる値は『6.1 POSプリンターステータス一覧』を参照してください。
- 詳細については『3章 Win32 API』のRpGetStatus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.Driver.PosPrinter.ErrorCode LastError { get; }
```

初期値

SUCCESS

備考

取得できる値は『5章 エラーコード一覧』を参照してください。

IsValid

オープン状態を取得します。

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

初期値

FALSE

備考

以下の値を取得します。

- TRUE : すでにオープンしている状態です。
- FALSE : オープンしていない状態です。

4.6 メソッド

OpenMonPrinter

指定したプリンターで通信ライブラリ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
 TYPE_PRINTER: プリンタードライバ名を指定してオープンします。(固定)

name
 通信ライブラリ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

戻り値については『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1つのみオープンできます。
- 通信ライブラリを利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の接続先がFILEの場合、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

通信ライブラリの使用を終了します。

```
ErrorCode CloseMonPrinter()
```

戻り値

戻り値については『5章 エラーコード一覧』を参照してください。

備考

詳細については『3章 Win32 API』のRpCloseMonPrinterを参照してください。

LockPrinter

他のプロセスからのプリンターへのデータ送信およびハードウェアリセット要求を全て禁止(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
タイムアウト時間をmsec(ミリ秒)単位で指定します。

戻り値

戻り値については『5章 エラーコード一覧』を参照してください。

備考

詳細については『3章 Win32 API』のRpLockPrinterを参照してください。

UnlockPrinter

LockPrinterによる他プロセスからのアクセス禁止(ロック)を解除します。

```
ErrorCode UnlockPrinter()
```

戻り値

戻り値については『5章 エラーコード一覧』を参照してください。

備考

詳細については『3章 Win32 API』のRpUnlockPrinterを参照してください。

DirectIOEx

バイナリデータを送信及び受信します。
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。

```
ErrorCode DirectIOEx(  
    byte[] cmd,  
    ref byte[] data,  
    int timeout,  
    bool readFlag,  
    byte option )
```

バイナリデータを送信及び受信します。
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。

```
ErrorCode DirectIOEx(  
    byte[] cmd,  
    out string data,  
    int timeout,  
    byte option )
```

バイナリデータ送信します。

```
ErrorCode DirectIOEx(  
    byte[] cmd,  
    int timeout )
```

パラメーター

cmd

送信データバッファ

送信するデータを格納したバッファを指定します。

data

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。

timeout

タイムアウト時間

タイムアウト時間をmsec(ミリ秒)単位で指定します。

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 : 何か受信するかタイムアウトまで受信を続けます。

FALSE : リードサイズ分を受信するか、タイムアウト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

1 : ASB設定コマンドの応答を含めたデータを取得

戻り値

戻り値については『5章 エラーコード一覧』を参照してください。

備考

- 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。
- 詳細については『3章 Win32 API』のRpDirectIOExを参照してください。

ResetPrinter

プリンターをリセットします。

```
ErrorCode ResetPrinter()
```

戻り値

戻り値については『5章 エラーコード一覧』を参照してください。

備考

詳細については『3章 Win32 API』のRpResetPrinterを参照してください。

SetStatusBack

POS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メソッドを登録します。

```
ErrorCode SetStatusBack()
```

戻り値

戻り値については『5章 エラーコード一覧』を参照してください。

備考

詳細については『3章 Win32 API』のRpSetStatusBackFunctionを参照してください。

CancelStatusBack

SetStatusBackにて実行されたコールバックメソッドの登録を解除します。

```
ErrorCode CancelStatusBack()
```

戻り値

戻り値については『5章 エラーコード一覧』を参照してください。

備考

詳細については『3章 Win32 API』のRpCancelStatusBackを参照してください。

PowerOff

プリンターを電源オフ状態にします。

```
ErrorCode PowerOff()
```

戻り値

戻り値については『5章 エラーコード一覧』を参照してください。

備考

詳細については『3章 Win32 API』のRpPowerOffを参照してください。

GetCounter

指定したメンテナンスカウンターを取得します。

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

```
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、またはSII.Driver.PosPrinter.CounterIndexに定義されたカウンターIDの値を指定します。

type

メンテナンスカウンターのタイプ

以下の取得するメンテナンスカウンターのタイプ

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

戻り値については『5章 エラーコード一覧』を参照してください。

備考

カウンターIDの内容については『6.2 カウンターID』を参照してください。

詳細については『3章 Win32 API』のRpGetCounterを参照してください。

ResetCounter

指定したメンテナンスカウンターを初期化します。

```
ErrorCode ResetCounter(  
    CounterIndex index )
```

```
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、またはSII.Driver.PosPrinter.CounterIndexに定義されたカウンターIDの値を指定します。

戻り値

戻り値については『5章 エラーコード一覧』を参照してください。

備考

詳細については『3章 Win32 API』のRpResetCounterを参照してください。

GetType

プリンターのタイプIDとフォントタイプを取得します。

```
ErrorCode GetType(  
    out byte typeId,  
    out byte font,  
    out byte exrom,  
    out byte special )
```

パラメーター

typeId
 タイプID変数
 タイプIDを格納する変数を指定します。

font
 フォントタイプ変数
 フォントタイプを格納する変数を指定します。

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

戻り値については『5章 エラーコード一覧』を参照してください。

備考

- タイプIDの内容については『6.3 タイプID』を参照してください。
- フォントタイプの内容については『6.4 フォントタイプ』を参照してください。
- 詳細については『3章 Win32 API』のRpGetTypeを参照してください。

GetPrnCapability

指定したプリンター情報を取得します。
応答データをバイナリデータとして取得します。

```
ErrorCode GetPrnCapability(  
    byte id,  
    out byte[] data )
```

指定したプリンター情報を取得します。
応答データを文字列データとして取得します。

```
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。

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

戻り値については『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ID内容については『6.5 プリンターID』を参照してください。
- 詳細については『3章 Win32 API』のRpGetPrnCapabilityを参照してください。

OpenDrawer

指定したドロワをオープンします。

```
ErrorCode OpenDrawer(  
    Drawer drawer,  
    Pulse pulse )
```

パラメーター

drawer

対象ドロワ

使用するSII.Driver.PosPrinter.Drawerに定義されたオープンする対象ドロワの値を指定します。

pulse

ドロワキック時間

SII.Driver.PosPrinter.Drawerに定義されたドロワキック時間の値を指定します。

戻り値

戻り値については『5章 エラーコード一覧』を参照してください。

備考

- 対象ドロワの内容については『6.6 対象ドロワ』を参照してください。
- ドロワキック時間の内容については『6.7 ドロワキック時間』を参照してください。
- 詳細については『3章 Win32 API』のRpOpenDrawerを参照してください。

SendDataFile

コマンド定義ファイルを指定し、送信データを登録します。

```
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したコマンド定義ファイル名を指定します。

戻り値

戻り値については『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については『7章 コマンド定義ファイル』を参照してください。
- 詳細については『3章 Win32 API』のRpSendDataFileを参照してください。

DirectSendRead

SendDataFileで定義したコマンドを実行します。
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。

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

SendDataFileで定義したコマンドを実行します。
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。

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

SendDataFileで定義したコマンドを実行します。

```
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

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

タイムアウト時間をmsec(ミリ秒)単位で指定します。

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：何か受信するかタイムアウトまで受信を続けます。

FALSE：リードサイズ分を受信するか、タイムアウトまで受信を続けます。

戻り値

戻り値については『5章 エラーコード一覧』を参照してください。

備考

- 受信データに0x02が含まれる場合、0x02は0x5fに変換されます。
- 詳細については『3章 Win32 API』のRpDirectSendReadを参照してください。

GetProperty

指定したプロパティIDの内容を取得します。

```
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

data

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。

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

戻り値については『5章 エラーコード一覧』を参照してください。

備考

- プロパティIDの内容については『6.8 プロパティID』を参照してください。
- 詳細については『3章 Win32 API』のRpGetPropertyを参照してください。

SetProperty

指定したプロパティIDの内容を変更します。

```
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

data

設定データバッファ

変更するプロパティIDの内容を格納し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の内容を格納したバッファのサイズを指定します。

戻り値

戻り値については『5章 エラーコード一覧』を参照してください。

備考

- プロパティIDの内容については『6.8 プロパティID』を参照してください。
- 詳細については『3章 Win32 API』のRpSetPropertyを参照してください。

4.7 イベント

StatusCallback

応答されたPOSプリンターステータスを処理するイベントです。

```
event StatusCallbackHandler StatusCallback  
  
delegate void StatusCallbackHandler(  
    ASB status )
```

パラメーター

status
POSプリンターステータス変数
POS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

- POSプリンターステータスの内容については『6.1 POSプリンターステータス一覧』を参照してください。
- 詳細については『3章 Win32 API』のRpSetStatusBackFunction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	処理の最大利用数を超過しています。
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利用中のため、その機能は利用できません。
ERR_SPL_NOT_EXIST	-350	スプーラーサービスが起動していません。
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。

マクロ定義(定数)	値	内容
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

6.1 POSプリンターステータス一覧

注意

- ◆ POSプリンターステータスは、ASB設定コマンド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。
- ◆ 下表に記載のないビットは予約です。

POSプリンターステータスの応答内容の対応ビットは以下の通りです。

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
ヘッドエラー/初期化電圧エラー	ASB_HEAD_ERR	0x00000000	ヘッドエラー/初期化電圧エラーなし
		0x00000002	ヘッドエラー/初期化電圧エラーあり
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
オートカッターエラー	ASB_AUTOCUTTER_ERR	0x00000000	オートカッターエラーなし
		0x00000008	オートカッターエラーあり
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
ペーパーニアエンド	ASB_RECEIPT_NEAR_END	0x00000000	ペーパーニアエンドなし
		0x00000020	ペーパーニアエンドあり
マーク検出時紙ジャムエラー	ASB_MARK_PAPER_JAM_ERR	0x00000000	マーク検出時紙ジャムエラーなし
		0x00000040	マーク検出時紙ジャムエラーあり

ステータス	対応ビット		内容
カバーオープンエラー	ASB_COVER_OPEN	0x00000000	カバーが閉じている
		0x00000080	カバーが開いている
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	印字中
復帰待ち状態	ASB_RETURN_WAITING	0x00000000	－
		0x00000800	復帰待ち状態
ドロワセンサーの状態	ASB_DRAWER_KICK	0x00000000	ドロワセンサーの状態＝“L”
		0x00008000	ドロワセンサーの状態＝“H”
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
周辺機器選択	ASB_PERIPHERAL_EQUIPMENT	0x00000000	プリンター
		0x00020000	その他
自動復帰エラー	ASB_AUTORECOVER_ERR ※1	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 ※1	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 ※1	0x00000000	プリンター応答
		0x80000000	未接続、または通信エラー

※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。

取得できる応答値は上記の値を合計した値です。

ただし、応答なしの場合の値は0x80000000となります。

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	
ROLL_FEED_LINES	20	初期化可能	紙送りドットライン数 (100 ドットライン単位)
ROLL_HEAD_CHARGE	21	初期化可能	サーマルヘッド通電回数 (100 ドットライン単位)
PAPER_CUT	50	初期化可能	オートカッター駆動回数
OPERATION_TIME	70	初期化可能	製品の駆動時間(分単位)
ROLL_FEED_LINES	148	積算	紙送りドットライン数 (100 ドットライン単位)
ROLL_HEAD_CHARGE	149	積算	サーマルヘッド通電回数 (100 ドットライン単位)
PAPER_CUT	178	積算	オートカッター駆動回数
OPERATION_TIME	198	積算	製品の駆動時間(分単位)

6.3 タイプID

タイプIDの応答内容は以下の通りです。

応答値	内容
0x01	0 : マルチバイトコード対応なし 1 : マルチバイトコード対応あり
0x02	0 : オートカッターなし 1 : オートカッターあり
0x04	0に固定
0x08	0に固定
0x10	0に固定
0x20	0に固定
0x40	0に固定
0x80	0に固定

取得できる応答値は上記の値を合計した値です。

6.4 フォントタイプ

フォントタイプの応答内容は以下の通りです。

応答値	内容
2	Japanese kanji(JIS)

6.5 プリンターID

指定可能なプリンター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	0x1a	HEXコード
2	タイプID	『0 タイプID』を参照	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 RP-E10 Series."	ASCII文字列
69	多国語フォントの種類	日本語仕様: "KANJI JAPANESE"	ASCII文字列

6.6 対象ドロワ

指定可能な対象ドロワは以下の通りです。

対象ドロワ		内容
1	SII_RP_DRAWER_1	ドロワ1をオープン
2	SII_RP_DRAWER_2	ドロワ2をオープン

6.7 ドロワキック時間

指定可能なドロワキック時間は以下の通りです。

ドロワキック時間		内容
1	SII_RP_PULSE_100	100ミリ秒後ドロワを駆動
2	SII_RP_PULSE_200	200ミリ秒後ドロワを駆動
3	SII_RP_PULSE_300	300ミリ秒後ドロワを駆動
4	SII_RP_PULSE_400	400ミリ秒後ドロワを駆動
5	SII_RP_PULSE_500	500ミリ秒後ドロワを駆動
6	SII_RP_PULSE_600	600ミリ秒後ドロワを駆動
7	SII_RP_PULSE_700	700ミリ秒後ドロワを駆動
8	SII_RP_PULSE_800	800ミリ秒後ドロワを駆動

6.8 プロパティID

注意

- ◆ 記載のないプロパティIDの内容は、プリンタードライバーで設定した値が有効です。
- ◆ RpSetProperty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バイトに指定してください。
- ◆ []カッコ内は当該処理が行われるタイミングを示します。

指定可能なプロパティ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
2	速度※1	0 : 中速（静音） 1 : 低速 2 : 中速（品質） 3 : 高速
3	余白	0 : 最小余白 1 : 上余白最小 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

プロパティID	内容	
6	縮小	0 : 縮小しない 25 ~ 100 : 指定倍率(パーセント)
7	用紙カット	0 : カットしない 1 : フルカット(ジョブ単位) 2 : パーシャルカット(ジョブ単位) 3 : フルカット(ページ単位) 4 : パーシャルカット(ページ単位) 5 : パーシャルカット(ページ間)
8	マーク紙の頭出し	0 : しない 1 : ページ単位 2 : ジョブ単位
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
12	[印刷開始] ドロワ ^{※2}	0 : 利用しない 1 : ドロワ1を利用 2 : ドロワ2を利用
13	[印刷開始] ON時間(x 2ms)	1 ~ 255
14	[印刷開始] OFF時間(x 2ms)	1 ~ 255
15	[印刷開始] カスタムコマンド(128バイト)	プリンターコマンド
16 ^{※3}	[印刷開始] 紙送り(バックフィード)(dot)	0 ~ 74
17 ^{※3}	[印刷開始] 紙送り(フィード)(dot)	0 ~ 255
18	[印刷開始] ブザー	0 : ON 1 : OFF
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)	プリンターコマンド
26 ^{※3}	[ページ開始] 紙送り(バックフィード)(dot)	0 ~ 74

プロパティID	内容	
27※3	[ページ開始] 紙送り(フィード)(dot)	0 ~ 255
28	[ページ開始] ブザー	0 : ON 1 : OFF
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)	プリンターコマンド
36※3	[ページ終了] 紙送り(バックフィード)(dot)	0 ~ 74
37※3	[ページ終了] 紙送り(フィード)(dot)	0 ~ 255
38	[ページ終了] ブザー	0 : ON 1 : OFF
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
42	[印刷終了] ドロワ※2	0 : 利用しない 1 : ドロワ1を利用 2 : ドロワ2を利用
43	[印刷終了] ON時間(x 2ms)	1 ~ 255
44	[印刷終了] OFF時間(x 2ms)	1 ~ 255
45	[印刷終了] カスタムコマンド(128バイト)	プリンターコマンド
46※3	[印刷終了] 紙送り(バックフィード)(dot)	0 ~ 74
47※3	[印刷終了] 紙送り(フィード)(dot)	0 ~ 255
48	[印刷終了] ブザー	0 : ON 1 : OFF
50	用紙サイズ	0 : Letter 1 : A4 2 : 80 mm 3 : 58 mm 4 ~ 255: 上記以外の用紙※4
51	印刷の向き	0 : 縦 1 : 横

プロパティID	内容	
52	カラー印刷モード	0 : システム ^{※5} 1 : ドライバー ^{※5}
53	ウォーターマーク	0 : なし 1 : 左上 2 : 中央上 3 : 右上 4 : 左 5 : 中央 6 : 右 7 : 左下 8 : 中央下 9 : 右下
60 ^{※6}	プリセット	0 : 80 mmレシート設定 1 : 58 mmレシート設定 2 : 80 mmマーク紙設定 3 : 58 mmマーク紙設定 4 : A4 → 80 mm縮小設定 5 : A4 → 58 mm縮小設定 6 : ユーザー設定 ^{※4}

※1: プリンターのF/Wバージョンが1.02以前の場合は、「0: 中速(静音)」以外を選択してください。

※2: 各タイミングのドロウを同時に利用することはできません。

※3: 同じタイミングのフィード、バックフィードは、最後の設定が有効になります。

※4: 設定することはできません。取得のみ行えます。

※5: 詳細については『プリンタードライバー取扱説明書』を参照してください。

※6: 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。詳細については『プリンタードライバー取扱説明書』を参照してください。

7章 コマンド定義ファイル

7.1 フォーマット

コマンド定義ファイルは以下のフォーマットで作成します。

コマンド定義

- 1つのコマンド定義は必ず1行で記述します。
- 複数のコマンドを定義する場合には、複数行で記述します。
- コメントは省略できます。

[コマンド名1]=[プリンターコマンド]#[コメント1]

[コマンド名2]=[プリンターコマンド]#[コメント2]

[コマンド名3]=[プリンターコマンド]#[コメント3]

・
・
・

実際のコマンド定義パターン例：

CmdName_1="SII"#Comments1

CmdName_2="SII" 0A#Comments2

CmdName_3=53 49 49 0A

[コマンド名]

- “=”の左側にプリンターに送信するデータのコマンド名を記述します。
- コマンド名には“=”、“#”以外のASCII文字を使用してください。
- コマンド名にASCII文字以外が見つかった場合、その文字は無視されます。
- コマンド名は大文字/小文字を区別します。
- 登録可能なコマンド名は最大33バイトです。34バイト以上コマンド名を指定した場合は、33バイトで登録され以降の文字は無視されます。
- 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。

[プリンターコマンド]

- “=”の右側にプリンターに送信するプリンターコマンドを記述します。
- プリンターコマンドに文字列を指定する場合、“”で囲って記述します。
- プリンターコマンドを文字列で指定する場合、ASCII文字を使用してください。
- プリンターコマンドにバイナリデータを指定する場合、データは2桁の16進数で記述します。
各データの間にはスペースを挿入します。
- プリンターコマンドのデータサイズは最大10240バイトです。10240バイトまでのプリンターコマンドを指定してください。
ただし、プリンターコマンド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。
コマンド定義例のCmdName_1の送信データサイズは3バイト、CmdName_2の送信データサイズは4バイト、CmdName_3の送信データサイズは4バイトです。

[コメント]

- コメントを記述する場合、コメントの先頭に“#”を指定してください。
- コメントに使用する文字に制限はありません。

7.2 制限事項

コマンド定義ファイルの制限事項について以下に記載します。

- 読み込み可能なファイルサイズは最大4ギガバイトです。
- 使用可能なファイルはANSI形式及びUNICODE形式です。
- 登録可能なコマンド数は、システムの使用可能メモリによって制限されます。