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

Rev.03

[対象製品]
SLP720RTシリーズ
SLP721RTシリーズ

セイコーインスツル株式会社

Rev.01	2022年	3月
Rev.02	2022年	10月
Rev.03	2023年	4月

©セイコーインスツル株式会社　2022-2023
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。
その他の製品名および会社名は、各社の商標または登録商標です。
本書の内容は、断りなく変更することがあります。
本書及び本書に記載された製品の利用によって発生した
損害及びその回復に要する費用に対し、当社は一切の責任を負いかねます。
本書を第三者に無断で頒布することを禁じます。

はじめに

本書は、セイコーインスツル株式会社（以降：SII）が提供する「SII SDK for Windows」（以降：SDK）について説明します。
SDKは「対象製品」に記載のプリンタードライバー上で動作します。

対象製品

本書の対象製品について下記に記載します。

プリンター	インターフェイス	プリンタードライバー
SLP720RTシリーズ SLP721RTシリーズ	USB	SLP720RTシリーズ用 「SII Printer Driver for Windows」
	TCP/IP	

本書の表記

本書の表記について説明します。

用語一覧

本書で使用する用語について下記の通りに定義します。

用語	内容
技術説明書	下記の技術説明書。 ・SLP720RTシリーズ サーマルプリンタ 技術説明書 ・SLP721RTシリーズ サーマルプリンタ 技術説明書
取扱説明書	下記の取扱説明書。 ・SLP720RTシリーズ サーマルプリンタ 取扱説明書 ・SLP721RTシリーズ サーマルプリンタ 取扱説明書
プリンターコマンド	「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」。 プリンターコマンドの詳細は、「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。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。

記号

本書で使用されている記号について下記に記載します。

注意

◆ 注意事項や制限事項を記載しています。

参考

• 補足説明や関連事項を記載しています。

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 3-4
OpenMonPrinter 3-4
CloseMonPrinter 3-5
LockPrinter 3-5
UnlockPrinter 3-6
DirectIO 3-7
DirectIOEx 3-8
Reset 3-10
GetStatus 3-10
SetStatusBackFunction 3-11
SetStatusBackWnd 3-12
CancelStatusBack 3-13
PowerOff 3-14
GetCounter 3-14
ResetCounter 3-15
GetType 3-16
GetPrnCapability 3-17
OpenDrawer 3-18
ControlFeature 3-19
SendDataFile 3-20
DirectSendRead 3-20
GetPrintSmartLabelMode 3-22
SetPrintSmartLabelMode 3-22
SelectSmartLabelFile 3-23
PrintSmartLabelImageData 3-24
SetSmartLabelCharacterSet 3-25
ReplaceSmartLabelTextData 3-26
ReplaceSmartLabelImageData 3-27
ReplaceSmartLabelBarcodeData 3-29
GetProperty 3-30
SetProperty 3-31

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 プリンター 4-2
4.3.1 API一覧 4-2
共通API 4-2
専用API 4-2
4.3.2 共通プロパティ 4-4
LastError 4-4
IsValid 4-4
4.3.3 共通メソッド 4-5
OpenMonPrinter 4-5
CloseMonPrinter 4-5
4.3.4 専用プロパティ 4-6
Status 4-6
PrintSmartLabelMode 4-6
4.3.5 専用メソッド 4-7
LockPrinter 4-7
UnlockPrinter 4-7
DirectIOEx 4-8
ResetPrinter 4-9
SetStatusBack 4-10
CancelStatusBack 4-10
PowerOff 4-11
GetCounter 4-11
ResetCounter 4-12
GetType 4-13
GetPrnCapability 4-13
OpenDrawer 4-14
ControlFeature 4-15
SendDataFile 4-16
DirectSendRead 4-16
SelectSmartLabelFile 4-18
PrintSmartLabelImageData 4-18
ReplaceSmartLabelTextData 4-19
ReplaceSmartLabelImageData 4-20
ReplaceSmartLabelBarcodeData 4-21
GetProperty 4-23
SetProperty 4-24
4.3.6 イベント 4-25
StatusCallback 4-25

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 タイプID 6-3
6.5 対象ドロワ 6-4
6.6 ドロワ駆動時間 6-4

6.7 プリンターの機能 6-4
6.8 ラベルファイル印刷時用紙ID 6-5
6.9 プロパティID 6-5

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

8章 ラベル印字機能 8-1
8.1 ラベルファイルの構造 8-2
8.1.1 オブジェクトの種類とSDKでのサポート 8-2
8.1.2 ラベルファイルをSDKで印字する際の注意点 8-3
8.2 ラベルファイルの利用手順 8-5
8.2.1 ラベルファイルをそのままSDKから印字する場合 8-5
8.2.2 ラベルファイルのオブジェクトデータを置換して印字する場合 8-5

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、DEVMODEのプライベート領域の設定項目の一部の値を取得及び変更することが可能です。

SDKは下記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、各開発対象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、プリンターの設定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、SLP720RT/SLP721RT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、下記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 3.5、または4.0～4.8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、及びプリンタス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

本ソフトウェアは、プリンタードライバーと同時にインストールされます。

プリンタードライバーのインストール手順については、SLP720RT/SLP721RTシリーズ用「SII Printer Driver for Windows 取扱説明書」のインストール部分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

開発対象言語は下記の通りです。

- 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下記の通りです。

- SlpRtApi.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。

3.3 API一覧

Win32 APIに実装されているAPIは下記の通りです。

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	他のプロセスからのプリンターへのデータ送信及びハードウェアリセット要求を全て禁止(ロック)します。
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。
DirectIO	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	バイナリデータを送信及び受信します。
Reset	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	SetStatusBackFunctionまたはSetStatusBackWndで実行されたコールバック関数の登録を解除します。
PowerOff	プリンターを電源オフ状態にします。
GetCounter	メンテナンスカウンターを取得します。
ResetCounter	メンテナンスカウンターを初期化します。
GetType	プリンターの各種IDを取得します。
GetPrnCapability	プリンター情報を取得します。
OpenDrawer	指定したドロワを駆動します。本APIはSLP721RTのみサポートします。
ControlFeature	印刷以外のプリンターの機能を制御します。本APIはSLP721RTのみサポートします。
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 DirectSendReadW※1	SendDataFileで登録したコマンド定義ファイルについて、指定されたコマンド名に対応する送信データを送信します。
GetPrintSmartLabelMode	ラベルファイル印刷時の用紙を取得します。
SetPrintSmartLabelMode	ラベルファイル印刷時の用紙を指定します。
SelectSmartLabelFileA※1 SelectSmartLabelFileW※1	ラベル印字機能を利用するためのラベルファイル(*.sl)を指定します。
PrintSmartLabelImageData	SelectSmartLabelFileで選択したラベルを印字します。
SetSmartLabelCharacterSet	ReplaceSmartLabelTextDataAまたは ReplaceSmartLabelBarcodeDataAで置き換えるテキストデータのコードページを指定します。

API	機能概略
ReplaceSmartLabelTextDataA※1 **ReplaceSmartLabelTextDataW**※1	**SelectSmartLabelFile**で選択したラベルファイル(*.sl)のテキストオブジェクトの値を置き換えます。
ReplaceSmartLabelImageDataA※1 **ReplaceSmartLabelImageDataW**※1	**SelectSmartLabelFile**で選択したラベルファイル(*.sl)のイメージオブジェクトの値を置き換えます。
ReplaceSmartLabelBarcodeDataA※1 **ReplaceSmartLabelBarcodeDataW**※1	**SelectSmartLabelFile**で選択したラベルファイル(*.sl)のバーコードオブジェクトの値を置き換えます。
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を呼び出してください。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。

注意

◆ SLP720RT/SLP721RTは、ディスプレイ及びバーコードスキャナーのAPIをサポートしていません。

3.4 API詳細

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、「5.1 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126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ーの接続先は、USBまたはTCP/IPのどちらかに設定してください。
- 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、「5.1 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。
- 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、本APIで破棄されます。

LockPrinter

他のプロセスからのプリンターへのデータ送信及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 **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、ミリ秒(ms)で指定します。
有効範囲は3000～90000です。
3000未満の値を指定した場合は3000msに設定されます。
90000よりも大きい値を指定した場合は90000msに設定されます。

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、「5.1 エラーコード一覧」を参照してください。

備考

- 本APIを呼び出してから、**UnlockPrinter**が呼び出されるまでの間に、他のプロセスからプリンターに直接アクセスするようなAPIを呼び出した場合、それらのAPIは失敗します。
- 本APIによりロックされるのは他のプロセスのみ有効です。自身のプロセスからの呼び出しはAPIハンドルやスレッドに関わらずロックの対象とはなりません。
- 本APIにより、同一のAPIハンドルで最大99回まで多重のロックが可能です。ロックを解除するためには本APIを呼び出した回数と同じ回数の**UnlockPrinter**を呼び出してください。
- 他のプロセスで**LockPrinter**が呼び出されている場合、本APIを呼び出すと失敗します。
- TCP/IP接続の場合、送信をしない時間がTCP/IPの受信タイムアウト時間を超える前にロックを解除してください。ロックを解除しない場合は、データ抜けの発生や他のホストからの送信データがプリンターに割り込む可能性があります。
  受信タイムアウト時間については、SLP720RT/SLP721RTシリーズ用「SII Communication Setting Utility for Windows 取扱説明書」に記載の「3.2.3 各接続方式の設定画面」の「LAN通信の設定」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。

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、「5.1 エラーコード一覧」を参照してください。

備考

- **LockPrinter**が多重に呼び出されている場合、ロック解除には同じ回数の本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。
データ取得が必要ない場合は、0バイトを指定してください。
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ー
取得するデータを格納するバッファーを指定します。

i_timeout
タイムアウト時間
本APIの成功を待つ時間を、ミリ秒（ms）で指定します。
有効範囲は3000～90000です。
3000未満の値を指定した場合は3000msに設定されます。
90000よりも大きい値を指定した場合は90000msに設定されます。

i_flag
受信動作フラグ
受信動作を指定するフラグを下記から指定します。
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、「5.1 エラーコード一覧」を参照してください。

備考

- 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を呼び出してください。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。
- 送信するデータに、応答を無効にするASB設定コマンド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。
- 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を呼び出してください。
- 本APIはResetで中断できます。
- 他のプロセスでLockPrinterが呼び出されている場合、本APIを呼び出すと失敗します。
- 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。

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。
データ取得が必要ない場合は、0を指定してください。
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ー
取得するデータを格納するバッファーを指定します。

i_timeout

タイムアウト時間

本APIの成功を待つ時間を、ミリ秒（ms）で指定します。
有効範囲は3000～90000です。
3000未満の値を指定した場合は3000msに設定されます。
90000よりも大きい値を指定した場合は90000msに設定されます。

i_flag

受信動作フラグ

受信動作を指定するフラグを下記から指定します。
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下記から指定します。
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、「5.1 エラーコード一覧」を参照してください。

備考

- 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。
- 送信するデータに、応答を無効にするASB設定コマンド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**で中断できます。
- 他のプロセスで**LockPrinter**が呼び出されている場合、本APIを呼び出すと失敗します。
- 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。

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、「5.1 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を呼び出した後、数秒経過してからデータを送信してください。本API呼び出し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を呼び出した場合、下記のAPIは中断されます。
  - **DirectIO**
  - **DirectIOEx**
  - **DirectSendRead**
- 本API呼び出し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。
- 他のプロセスで**LockPrinter**が呼び出されている場合、本APIを呼び出すと失敗します。
- 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。

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、「5.1 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 **lpStatusCB*) (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、「5.1 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。

- 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。
- 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。
- 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。
- 本APIで登録したコールバック関数は、下記のAPIで解除されます。
  - **CloseMonPrinter**
  - **CancelStatusBack**
- 登録されたコールバック関数内から同一のAPIハンドルで下記のAPIを呼び出すことはできません。
  - **CloseMonPrinter**
  - **SetStatusBackFunction**
  - **SetStatusBackWnd**
  - **CancelStatusBack**
- 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、「5.1 エラーコード一覧」を参照してください。

備考

- 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。
- 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。
- 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。
- 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。
- 本APIは下記のAPIで解除されます。
  - CloseMonPrinter
  - CancelStatusBack
- 呼び出されたクリックイベントの処理ルーチン内から、同一のAPIハンドルで下記のAPIを呼び出すことはできません。
  - CloseMonPrinter
  - SetStatusBackFunction
  - SetStatusBackWnd
  - CancelStatusBack
- 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

SetStatusBackFunctionまたはSetStatusBackWndで呼び出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、「5.1 エラーコード一覧」を参照してください。

備考

- SetStatusBackFunctionまたはSetStatusBackWnd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、「5.1 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
取得するメンテナンスカウンターのカウンター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
取得した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、「5.1 エラーコード一覧」を参照してください。

備考

- 他のプロセスでLockPrinterが呼び出されている場合、本APIを呼び出すと失敗します。
- 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。

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
カウンター値を初期化するメンテナンスカウンターのカウンター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、「5.1 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のカウンター値が初期化されたことをGetCounterで確認してください。
- 他のプロセスでLockPrinterが呼び出されている場合、本APIを呼び出すと失敗します。
- 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。

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

o_typeID
ID1
3が格納されます。

o_fontID
ID2
2が格納されます。

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 : 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、「5.1 エラーコード一覧」を参照してください。

備考

- 他のプロセスで**LockPrinter**が呼び出されている場合、本APIを呼び出すと失敗します。
- 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。

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情報のプリンター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情報を格納するバッファーサイズを指定します。
APIから制御が戻ると取得したプリンター情報のデータサイズが格納されます。
指定したバッファーサイズが取得するプリンター情報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情報の応答サイズが格納されます。

o_dat

受信データバッファー

取得したプリンター情報を格納するバッファーを指定します。

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、「5.1 エラーコード一覧」を参照してください。

備考

- 他のプロセスで**LockPrinter**が呼び出されている場合、本APIを呼び出すと失敗します。
- 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。

OpenDrawer

指定したドロワを駆動します。
SLP721RTのみサポートします。

INT **OpenDrawer**(
INT *i_hdl*,
BYTE *i_drawer*,
BYTE *i_pulse*)

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_drawer

対象ドロワ

駆動する対象ドロワを指定します。
指定可能な内容は、「6.5 対象ドロワ」を参照してください。

i_pulse

ドロワ駆動時間

対象ドロワの駆動パルスのON/OFF時間を指定します。ON時間とOFF時間は同じ時間が指定されます。
指定可能な内容は、「6.6 ドロワ駆動時間」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、「5.1 エラーコード一覧」を参照してください。

備考

- 他のプロセスで**LockPrinter**が呼び出されている場合、本APIを呼び出すと失敗します。
- 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。
- ドロワの制御時間については、お客様がご利用されているドロワの仕様に従ってください。

ControlFeature

印刷以外のプリンターの機能を制御します。
SLP721RTのみサポートします。

INT **ControlFeature**(
INT *i_hdl*,
BYTE *i_feature*,
INT *i_arg1*,
INT *i_arg2*)

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_feature
プリンターの機能
制御するプリンターの機能を指定します。
指定可能な内容は、「6.7 プリンターの機能」を参照してください。

i_arg1
引数1
指定したプリンターの機能に対する引数を指定します。
指定可能な内容は、「6.7 プリンターの機能」を参照してください。

i_arg2
引数2
指定したプリンターの機能に対する引数を指定します。
指定可能な内容は、「6.7 プリンターの機能」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、「5.1 エラーコード一覧」を参照してください。

備考

- 他のプロセスで**LockPrinter**が呼び出されている場合、本APIを呼び出すと失敗します。
- 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。

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、「5.1 エラーコード一覧」を参照してください。

備考

- 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。
- 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。

DirectSendRead

SendDataFileで登録したコマンド定義ファイルについて、指定されたコマンド名に対応する送信データを送信します。

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下記から指定します。
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。
データ取得が必要ない場合は、0を指定してください。
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ー

取得するデータを格納するバッファーを指定します。

i_timeout

タイムアウト時間

本APIの成功を待つ時間を、ミリ秒（ms）で指定します。
有効範囲は3000～90000です。
3000未満の値を指定した場合は3000msに設定されます。
90000よりも大きい値を指定した場合は90000msに設定されます。

i_flag

受信動作フラグ

受信動作を指定するフラグを下記から指定します。
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、「5.1 エラーコード一覧」を参照してください。

備考

- 送信するデータに、応答を無効にするASB設定コマンド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**で中断できます。
- 他のプロセスで**LockPrinter**が呼び出されている場合、本APIを呼び出すと失敗します。
- 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。

GetPrintSmartLabelMode

ラベルファイル印刷時の用紙を取得します。

INT **GetPrintSmartLabelMode**(
INT *i_hdl*,
DWORD *o_id*)

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_id
用紙IDデータバッファー
ラベルファイル印刷時用紙IDを格納するバッファー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、「5.1 エラーコード一覧」を参照してください。

SetPrintSmartLabelMode

ラベルファイル印刷時の用紙を指定します。

INT **SetPrintSmartLabelMode**(
INT *i_hdl*,
DWORD *i_id*)

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
ラベルファイル印刷時用紙ID
ラベルファイル印刷時の用紙を指定してください。
指定可能な内容は、「6.8 ラベルファイル印刷時用紙ID」を参照してください。
無効な値を指定した場合は無視されます。

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、「5.1 エラーコード一覧」を参照してください。

備考

- 本APIの設定はPrintSmartLabelImageDataでのラベルファイル印刷時のみ有効です。
- 本APIによる用紙IDの指定は、CloseMonPrinterによって破棄されます。

SelectSmartLabelFile

ラベル印字機能を利用するためのラベルファイル(*.sl)を指定します。

INT SelectSmartLabelFile(
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
使用するラベルファイル(*.sl)名
アプリで作成したラベルファイルをパスで指定してください。

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、「5.1 エラーコード一覧」を参照してください。

備考

- 本APIで選択できるファイルは、1APIハンドルに対して1つです。
- 本APIで指定するファイルは、APIを呼び出すすべてのプロセスから読み取り可能にしてください。
- 本APIによるファイル選択は、CloseMonPrinterによって破棄されます。
- 使用できるラベルファイルには制限があります。制限については、「8章 ラベル印字機能」を参照してください。

PrintSmartLabelImageData

SelectSmartLabelFileで選択したラベルを印字します。

INT PrintSmartLabelImageData(
INT *i_hdl*,
DWORD *i_timeout*,
BYTE *i_option*)

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、ミリ秒（ms）で指定します。
有効範囲は3000～90000です。
3000未満の値を指定した場合は3000msに設定されます。
90000よりも大きい値を指定した場合は90000msに設定されます。

i_option

予約

0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、「5.1 エラーコード一覧」を参照してください。

備考

- 各オブジェクトのデータの置き換えに失敗した場合は、本APIは失敗します。
- オブジェクトを配置していないラベルファイルを指定すると、本APIは失敗します。
- 他のプロセスでLockPrinterが呼び出されている場合、本APIを呼び出すと失敗します。
- 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。
- コンピューター起動後の初回印刷は、印刷が開始されるまでに時間がかかる場合があります。
- ラベルファイルを利用した印字についての詳細は、「8章 ラベル印字機能」を参照してください。

SetSmartLabelCharacterSet

ReplaceSmartLabelTextDataAまたは**ReplaceSmartLabelBarcodeDataA**で置き換えるテキストデータのコードページを指定します。

INT **SetSmartLabelCharacterSet**(
INT *i_hdl*,
BYTE *i_charSet*)

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_charSet

使用するコードページ

指定する*i_charSet*を下記に示します。

文字セット	文字コードテーブル	*i_charSet*
システムの文字セット	(System Locale) システムに依存	0
タイ語	Thai (Code Page874)	1
日本語	Japanese (Code Page932/シフトJIS)	2
中国語(簡体字)	Simplified Chinese (Code Page936)	3
韓国語	Korean (Code Page949)	4
中国語(繁体字)	Traditional Chinese (Code Page950)	5
中央ヨーロッパ言語	Central European (Code Page1250)	6
キリル言語	Cyrillic (Code Page1251)	7
欧文	Latin (Code Page1252)	8
ギリシャ語	Greek (Code Page1253)	9
トルコ語	Turkish (Code Page1254)	10
ヘブライ語	Hebrew (Code Page1255)	11
アラビア語	Arabic (Code Page1256)	12
バルト語	Baltic (Code Page1257)	13
ベトナム語	Vietnamese (Code Page1258)	14

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、「5.1 エラーコード一覧」を参照してください。

備考

- 本APIを実行した後のコードページ適用対象APIを下記に示します。
  - ReplaceSmartLabelTextDataA
  - ReplaceSmartLabelBarcodeDataA

- 本APIでコードページの設定を行わない場合は、「システムの文字セット」でエンコードします。
- 本APIによるコードページ指定は、**CloseMonPrinter**によって解除されます。

ReplaceSmartLabelTextData

SelectSmartLabelFileで選択したラベルファイル(*.sl)のテキストオブジェクトの値を置き換えます。

INT **ReplaceSmartLabelTextData**(
INT *i_hdl*,
INT *i_mapId*,
LPCTSTR *i_text*)

パラメーター

i_hdl

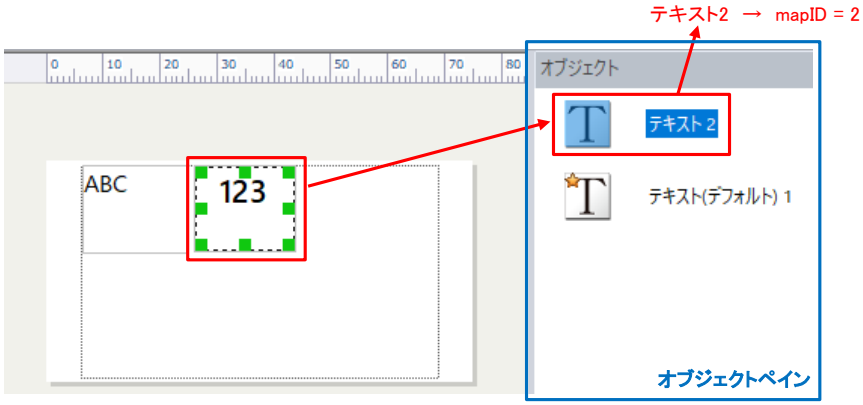
APIハンドル

OpenMonPrinterで取得したAPIハンドルを指定します。

i_mapId

テキストオブジェクトのID

アプリのラベルファイル(*.sl)に配置したテキストオブジェクトのIDを指定してください。テキストオブジェクトのIDはアプリのUI画面で確認できます。



Smart Label CreatorのUI画面



SII Layout EditorのUI画面

i_text

置き換えるテキストデータ

置き換えるテキストデータを入力してください。
指定する文字列の末尾は必ずNULLとしてください。
テキストデータの最大長さはNULLを含めて10000文字です。
テキストデータが空文字の場合は、*i_mapId*で指定したテキストオブジェクトでテキストデータの印字はされません。

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、「5.1 エラーコード一覧」を参照してください。

備考

- SelectSmartLabelFileを呼び出していない場合、本APIは失敗します。
- ReplaceSmartLabelTextDataAの場合、SetSmartLabelCharacterSetで設定された言語でエンコードします。
- ラベルファイルを利用した印字例については、「8章 ラベル印字機能」を参照してください。

ReplaceSmartLabelImageData

SelectSmartLabelFileで選択したラベルファイル（*.sl）のイメージオブジェクトを置き換えます。

```
INT ReplaceSmartLabelImageData(
    INT i_hdl,
    INT i_mapId,
    LPCTSTR i_fname )
```

パラメーター

i_hdl

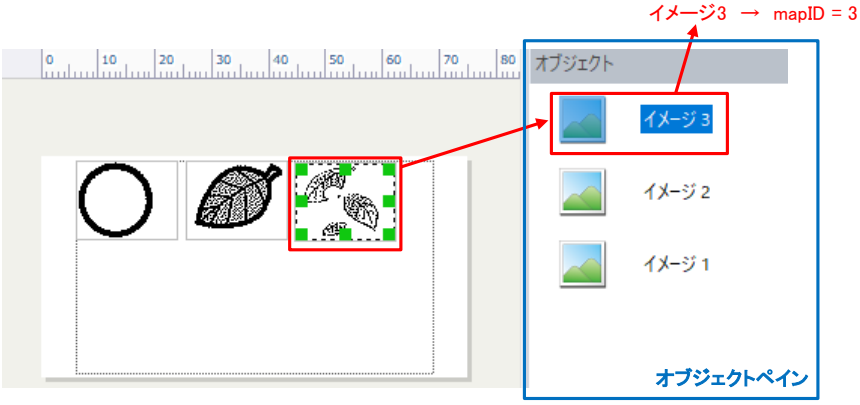
APIハンドル

OpenMonPrinterで取得したAPIハンドルを指定します。

i_mapId

イメージオブジェクトのID

アプリのラベルファイル（*.sl）に配置したイメージオブジェクトのIDを指定してください。イメージオブジェクトのIDはアプリのUI画面で確認できます。



Smart Label CreatorのUI画面



SII Layout EditorのUI画面

i_fname

置き換えるイメージデータ

置き換えるイメージデータのファイル名をパスで指定してください。
サポートするイメージファイルのファイル拡張子は.bmp、.gif、.jpg、.jpeg、.tiff、.tif、.pngです。
ただし、サポートする拡張子であっても、フォーマットによっては登録できない場合があります。

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、「5.1 エラーコード一覧」を参照してください。

備考

- SelectSmartLabelFileを呼び出していない場合、本APIは失敗します。
- 本APIで指定するファイルは、APIを呼び出すすべてのプロセスから読み取り可能にしてください。
- ラベルファイルを利用した印字例については、「8章 ラベル印字機能」を参照してください。

ReplaceSmartLabelBarcodeData

SelectSmartLabelFileで選択したラベルファイル(*.sl)のバーコードオブジェクトの値を置き換えます。

INT **ReplaceSmartLabelBarcodeData**(
INT *i_hdl*,
INT *i_mapId*,
LPCTSTR *i_barcodeData*)

パラメーター

i_hdl

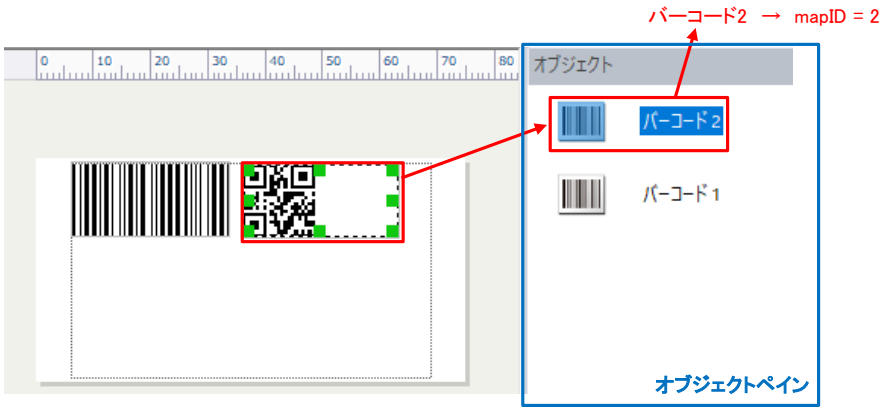
APIハンドル

OpenMonPrinterで取得したAPIハンドルを指定します。

i_mapId

バーコードオブジェクトのID

アプリのラベルファイル(*.sl)に配置したバーコードオブジェクトのIDを指定してください。
バーコードオブジェクトのIDはアプリのUI画面で確認できます。



Smart Label CreatorのUI画面



SII Layout EditorのUI画面

i_barcodeData

置き換えるバーコードデータ

置き換えるバーコードデータを入力してください。
指定する文字列の末尾は必ずNULLとしてください。

バーコードデータの最大長さはNULLを含めて5000文字です。5000文字を超えるバーコードデータが指定された場合、5001文字以降は破棄されます。
バーコードデータが空文字の場合は、*i_mapId*で指定したバーコードオブジェクトでバーコードデータの印字はされません。
置き換える文字列が無効なバーコードデータの場合でも、エラーにはなりません。有効なバーコードデータをご確認の上、指定してください。

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、「5.1 エラーコード一覧」を参照してください。

備考

- SelectSmartLabelFileを呼び出していない場合、本APIは失敗します。
- ReplaceSmartLabelBarcodeDataAの場合、SetSmartLabelCharacterSetで設定された言語でエンコードします。
- ラベルファイルを利用した印字例については、「8章 ラベル印字機能」を参照してください。

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
取得する印刷設定内容のプロパティIDを指定します。
指定可能な内容は、「6.9 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
取得した印刷設定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、指定した印刷設定内容のデータサイズが*io_size*に格納されます。

io_size

取得データサイズ

印刷設定内容を格納するバッファーサイズを指定します。
APIから制御が戻ると取得した印刷設定内容のデータサイズが格納されます。
指定したバッファーサイズが取得する印刷設定内容のデータサイズよりも小さい場合、本APIは失敗し、指定した印刷設定内容のデータ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、「5.1 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

変更する印刷設定内容のプロパティIDを指定します。
指定可能な内容は、「6.9 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

変更する印刷設定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

変更する印刷設定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、「5.1 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

開発対象言語は下記の通りです。

- 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下記の通りです。

- SlpRtCls.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 プリンター

4.3.1 API一覧

.NET API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.SDK.SlpRt
- クラス名　　：　StatusAPI

共通 API

種別	API	機能概略
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。

専用 API

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	PrintSmartLabelMode	ラベルファイル印刷時の用紙を指定または取得します。
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及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	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。
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	OpenDrawer	指定したドロワを駆動します。本APIはSLP721RTのみサポートします。
メソッド	ControlFeature	印刷以外のプリンターの機能を制御します。本APIはSLP721RTのみサポートします。

種別	API	機能概略
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を実行します。
メソッド	SelectSmartLabelFile	ラベル印字機能を利用するためのラベルファイル(*.sl)を指定します。
メソッド	PrintSmartLabelImageData	SelectSmartLabelFileで選択したラベルを印字します。
メソッド	ReplaceSmartLabelTextData	SelectSmartLabelFileで選択したラベルファイル(*.sl)のテキストオブジェクトの値を置き換えます。
メソッド	ReplaceSmartLabelImageData	SelectSmartLabelFileで選択したラベルファイル(*.sl)のイメージオブジェクトの値を置き換えます。
メソッド	ReplaceSmartLabelBarcodeData	SelectSmartLabelFileで選択したラベルファイル(*.sl)のバーコードオブジェクトの値を置き換えます。
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。

4.3.2 共通プロパティ

LastError

最後に呼び出したAPIのエラー値を取得します。

ErrorCode LastError { get; }

初期値

SUCCESS

備考

- エラーコードの内容は、「5.1 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。

4.3.3 共通メソッド

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、「5.1 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.NET APIの各クラス合わせて最大8インスタンスまで同時にオープンできます。
- .NET APIを使用しなくなった場合は、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は、USBまたはTCP/IPのどちらかに設定してください。
- 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、「5.1 エラーコード一覧」を参照してください。

備考

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

4.3.4 専用プロパティ

Status

最新のプリンターステータスを取得します。

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。

PrintSmartLabelMode

ラベルファイル印刷時の用紙を指定または取得します。

PrintSmartLabelMode { set; get; }

初期値

MODE_MARK

備考

- 指定可能な内容は、「6.8 ラベルファイル印刷時用紙ID」を参照してください。
- IsValidがFALSEの時は、正しい値が取得できません。

4.3.5 専用メソッド

LockPrinter

他のプロセスからのプリンターへのデータ送信及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
本APIの成功を待つ時間を、ミリ秒（ms）で指定します。
有効範囲は3000～90000です。
3000未満の値を指定した場合は3000msに設定されます。
90000よりも大きい値を指定した場合は90000msに設定されます。

戻り値

エラーコードを返します。エラーコードの内容は、「5.1 エラーコード一覧」を参照してください。

備考

- 本APIにより、最大99回まで多重のロックが可能です。ロックを解除するためには本APIを呼び出した回数と同じ回数の**UnlockPrinter**を呼び出してください。
- TCP/IP接続の場合、送信をしない時間がTCP/IPの受信タイムアウト時間を超える前にロックを解除してください。ロックを解除しない場合は、データ抜けの発生や他のホストからの送信データがプリンターに割り込む可能性があります。
  受信タイムアウト時間については、SLP720RT/SLP721RTシリーズ用「SII Communication Setting Utility for Windows 取扱説明書」に記載の「3.2.3 各接続方式の設定画面」の「LAN通信の設定」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

エラーコードを返します。エラーコードの内容は、「5.1 エラーコード一覧」を参照してください。

備考

- LockPrinterが多重に呼び出されている場合、ロック解除には同じ回数の本APIの呼び出しが必要です。

DirectIOEx

(a)の構文は、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)の構文は、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)の構文は、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。
データ取得が必要ない場合は、0を指定してください。
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、ミリ秒(ms)で指定します。
有効範囲は3000～90000です。
3000未満の値を指定した場合は3000msに設定されます。
90000よりも大きい値を指定した場合は90000msに設定されます。

readFlag
受信動作フラグ

受信動作を指定するフラグを下記から指定します。
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下記から指定します。
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、「5.1 エラーコード一覧」を参照してください。

備考

- 本APIが成功した場合、構文（a）の*data*は*data*で指定した受信データサイズを限度に、実際の受信データのサイズにリサイズされます。
- 送信するデータに、応答を無効にするASB設定コマンド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**で中断できます。
- 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、「5.1 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を呼び出した後、数秒経過してからデータを送信してください。本API呼び出し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を呼び出した場合、下記のAPIは中断されます。
  - **DirectIOEx**
  - **DirectSendRead**
- 本API呼び出し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、「5.1 エラーコード一覧」を参照してください。

備考

- 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。
- 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。
- 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。
- 本APIで開始したStatusCallbackイベントは、下記のAPIで停止されます。
  - CloseMonPrinter
  - CancelStatusBack
- プリンターステータスの受信からイベント発生まで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

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、「5.1 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、「5.1 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。
メンテナンスカウンターの取得には、CounterIndexに定義されたカウンターIDを用いる構文と、数値で指定する構文の2種類があります。
(a)の構文は、CounterIndexに定義されたカウンターIDを使用し、カウンター値を取得します。
(b)の構文は、カウンターIDを数値で指定し、カウンター値を取得します。

(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

取得するメンテナンスカウンターのカウンターIDを指定します。
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を数値で指定する場合は、(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下記から指定します。
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

取得した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、「5.1 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。
メンテナンスカウンターの初期化には、CounterIndexに定義されたカウンターIDを用いる構文と、数値で指定する構文の2種類があります。
(a)の構文は、CounterIndexに定義されたカウンターIDを使用し、カウンター値を初期化します。
(b)の構文は、カウンターIDを数値で指定し、カウンター値を初期化します。

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

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

パラメーター

index
カウンターID
カウンター値を初期化するメンテナンスカウンターのカウンターIDを指定します。
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を数値で指定する場合は、(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。

戻り値

エラーコードを返します。エラーコードの内容は、「5.1 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
3が格納されます。

fontID
ID2
2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、「5.1 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。
プリンター情報の取得には、PrinterIdに定義されたプリンターIDを用いる構文と、数値で指定する構文の2種類があります。
(a)の構文は、PrinterIdに定義されたプリンターIDを使用し、プリンター情報をバイナリデータで取得します。
(b)の構文は、PrinterIdに定義されたプリンターIDを使用し、プリンター情報を文字列データで取得します。
(c)の構文は、プリンターIDを数値で指定し、プリンター情報をバイナリデータで取得します。
(d)の構文は、プリンターIDを数値で指定し、プリンター情報を文字列データで取得します。

(a) ErrorCode **GetPrnCapability**(
PrinterId *id*,
out byte[] *data*)

(b) ErrorCode **GetPrnCapability**(
PrinterId *id*,
out string *data*)

(c) ErrorCode **GetPrnCapability**(
byte *id*,
out byte[] *data*)

(d) 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情報のプリンターIDを指定します。
PrinterIdに定義されたプリンターIDを使用する場合は、(a)、(b)を指定してください。(a)はプリンター情報の応答形式をバイナリコードで取得し、(b)はASCII文字列で取得します。
プリンターIDを数値で指定する場合は、(c)、(d)を指定してください。(c)はプリンター情報の応答形式をバイナリコードで取得し、(d)はASCII文字列で取得します。
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。

data

受信データバッファー

取得したプリンター情報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、「5.1 エラーコード一覧」を参照してください。

OpenDrawer

指定したドロワを駆動します。
SLP721RTのみサポートします。

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

パラメーター

drawer

対象ドロワ

駆動する対象ドロワを指定します。
指定可能な内容は、「6.5 対象ドロワ」を参照してください。

pulse

ドロワ駆動時間

対象ドロワの駆動パルスのON/OFF時間を指定します。ON時間とOFF時間は同じ時間が指定されます。
指定可能な内容は、「6.6 ドロワ駆動時間」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、「5.1 エラーコード一覧」を参照してください。

備考

- ドロワの制御時間については、お客様がご利用されているドロワの仕様に従ってください。

ControlFeature

印刷以外のプリンターの機能を制御します。
SLP721RTのみサポートします。

ErrorCode **ControlFeature**(
Feature *feature*,
int *arg1*,
int *arg2*)

パラメーター

feature

プリンターの機能

制御するプリンターの機能を指定します。
指定可能な内容は、「6.7 プリンターの機能」を参照してください。

arg1

引数1

指定したプリンターの機能に対する引数を指定します。
指定可能な内容は、「6.7 プリンターの機能」を参照してください。

arg2

引数2

指定したプリンターの機能に対する引数を指定します。
指定可能な内容は、「6.7 プリンターの機能」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、「5.1 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、「5.1 エラーコード一覧」を参照してください。

備考

- 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。
- 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。

DirectSendRead

SendDataFileで登録したコマンド定義ファイルについて、指定されたコマンド名に対応する送信データを送信します。

(a)の構文は、**SendDataFile**で登録したコマンド定義ファイルについて、指定されたコマンド名に対応する送信データを送信後、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)の構文は、**SendDataFile**で登録したコマンド定義ファイルについて、指定されたコマンド名に対応する送信データを送信後、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)の構文は、**SendDataFile**で登録したコマンド定義ファイルについて、指定されたコマンド名に対応する送信データを送信します。

(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
受信するデータのタイプを下記から指定します。
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、ミリ秒（ms）で指定します。
有効範囲は3000～90000です。
3000未満の値を指定した場合は3000msに設定されます。
90000よりも大きい値を指定した場合は90000msに設定されます。

readFlag
受信動作フラグ
受信動作を指定するフラグを下記から指定します。
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、「5.1 エラーコード一覧」を参照してください。

備考

- 本APIは**ResetPrinter**で中断できます。
- 受信データに0x02が含まれる場合、0x02は0x5fに変換されます。

SelectSmartLabelFile

ラベル印字機能を利用するためのラベルファイル(*.sl)を指定します。

ErrorCode **SelectSmartLabelFile**(
string *fileName*)

パラメーター

fileName
使用するラベルファイル(*.sl)名
アプリで作成したラベルファイルのファイル名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、「5.1 エラーコード一覧」を参照してください。

備考

- 本APIで選択できるファイルは、1インスタンスに対して1つです。
- 本APIで指定するファイルは、APIを呼び出すすべてのプロセスから読み取り可能にしてください。
- 本APIによるファイル選択は、**CloseMonPrinter**によって破棄されます。
- 使用できるラベルファイルには制限があります。制限については、「8章 ラベル印字機能」を参照してください。

PrintSmartLabelImageData

SelectSmartLabelFileで選択したラベルを印字します。

ErrorCode **PrintSmartLabelImageData**(
int *timeout,*
byte *option*)

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、ミリ秒(ms)で指定します。
有効範囲は3000～90000です。
3000未満の値を指定した場合は3000msに設定されます。
90000よりも大きい値を指定した場合は90000msに設定されます。

option
予約
0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、「5.1 エラーコード一覧」を参照してください。

備考

- 各オブジェクトのデータの置き換えに失敗した場合は、本APIは失敗します。
- オブジェクトを配置していないラベルファイルを指定すると、本APIは失敗します。
- コンピューター起動後の初回印刷は、印刷が開始されるまでに時間がかかる場合があります。
- ラベルファイルを利用した印字についての詳細は、「8章 ラベル印字機能」を参照してください。

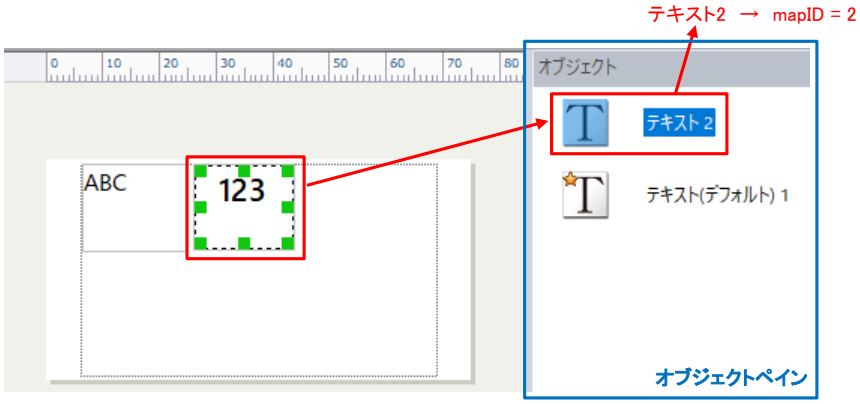
ReplaceSmartLabelTextData

SelectSmartLabelFileで選択したラベルファイル(*.sl)のテキストオブジェクトの値を置き換えます。

ErrorCode **ReplaceSmartLabelTextData**(
int *mapId,*
string *text*)

パラメーター

mapId
テキストオブジェクトのID
アプリのラベルファイル(*.sl)に配置したテキストオブジェクトのIDを指定してください。テキストオブジェクトのIDはアプリのUI画面で確認できます。



Smart Label CreatorのUI画面



SII Layout EditorのUI画面

text

置き換えるテキストデータ

置き換えるテキストデータを入力してください。
指定する文字列の末尾は必ずNULLとしてください。
テキストデータの最大長さはNULLを含めて10000文字です。
テキストデータが空文字の場合は、*mapId*で指定したテキストオブジェクトでテキストデータの印字はされません。

戻り値

エラーコードを返します。エラーコードの内容は、「5.1 エラーコード一覧」を参照してください。

備考

- SelectSmartLabelFileを呼び出していない場合、本APIは失敗します。
- ラベルファイルを利用した印字についての詳細は、「8章 ラベル印字機能」を参照してください。

ReplaceSmartLabelImageData

SelectSmartLabelFileで選択したラベルファイル(*.sl)のイメージオブジェクトの値を置き換えます。

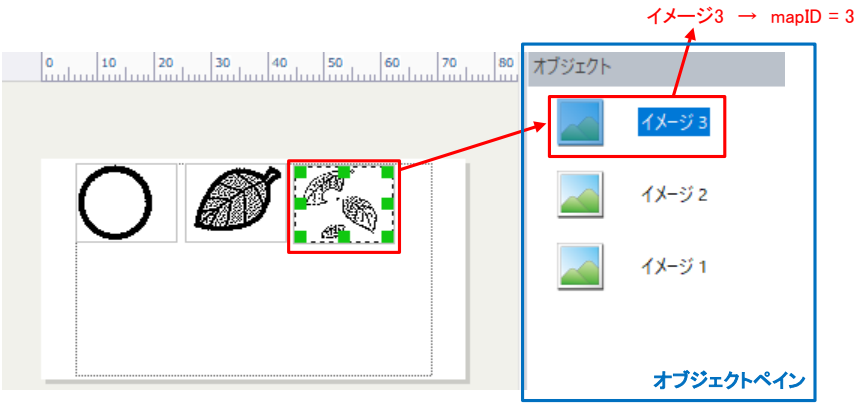
ErrorCode **ReplaceSmartLabelTextData**(
int *mapId,*
string *fileName*)

パラメーター

mapId

イメージオブジェクトのID

アプリのラベルファイル(*.sl)に配置したイメージオブジェクトのIDを指定してください。イメージオブジェクトのIDはアプリのUI画面で確認できます。



Smart Label CreatorのUI画面



SII Layout EditorのUI画面

fileName

置き換えるイメージデータ

置き換えるイメージデータのファイル名を指定してください。
サポートするイメージファイルのファイル拡張子は.bmp、.gif、.jpg、.jpeg、.tiff、.tif、.pngです。
ただし、サポートする拡張子であっても、フォーマットによっては登録できない場合があります。

戻り値

エラーコードを返します。エラーコードの内容は、「5.1 エラーコード一覧」を参照してください。

備考

- SelectSmartLabelFileを呼び出していない場合、本APIは失敗します。
- 本APIで指定するファイルは、APIを呼び出すすべてのプロセスから読み取り可能にしてください。
- ラベルファイルを利用した印字についての詳細は、「8章 ラベル印字機能」を参照してください。

ReplaceSmartLabelBarcodeData

SelectSmartLabelFileで選択したラベルファイル(*.sl)のバーコードオブジェクトの値を置き換えます。

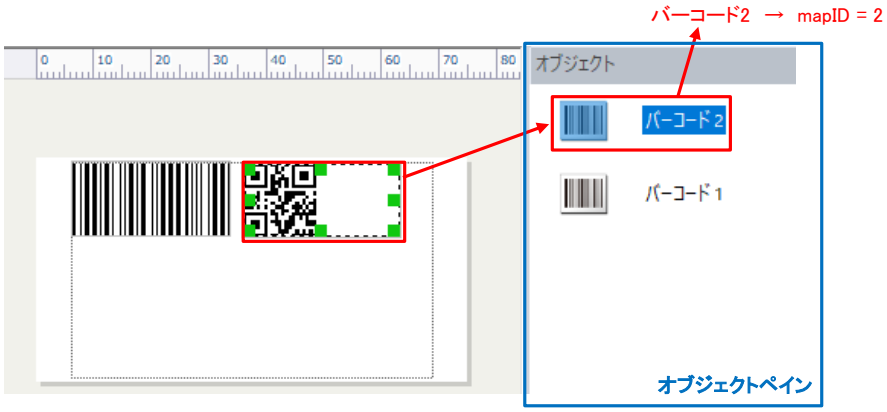
```
ErrorCode ReplaceSmartLabelBarcodeData(
    int mapId,
    string barcodeData )
```

パラメーター

mapId

バーコードオブジェクトのID

アプリのラベルファイル(*.sl)に配置したバーコードオブジェクトのIDを指定してください。
バーコードオブジェクトのIDはアプリのUI画面で確認できます。



Smart Label CreatorのUI画面



SII Layout EditorのUI画面

barcodeData

置き換えるバーコードデータ

置き換えるバーコードデータを入力してください。
指定する文字列の末尾は必ずNULLとしてください。
バーコードデータの最大長さはNULLを含めて5000文字です。5000文字を超えるバーコードデータが指定された場合、5001文字以降は破棄されます。
バーコードデータが空文字の場合は、*mapId*で指定したバーコードオブジェクトでバーコードデータの印字はされません。
置き換える文字列が無効なバーコードデータの場合でも、エラーにはなりません。有効なバーコードデータをご確認の上、指定してください。

戻り値

エラーコードを返します。エラーコードの内容は、「5.1 エラーコード一覧」を参照してください。

備考

- **SelectSmartLabelFile**を呼び出していない場合、本APIは失敗します。
- ラベルファイルを利用した印字についての詳細は、「8章 ラベル印字機能」を参照してください。

GetProperty

印刷設定内容(一部)を取得します。
印刷設定内容の取得には、PropertyIdに定義されたプロパティIDを用いる構文と、数値で指定する構文の2種類があります。
(a)の構文は、PropertyIdに定義されたプロパティIDを使用し、プロパティの印刷設定内容(一部)を取得します。
(b)の構文は、プロパティIDを数値で指定し、プロパティの印刷設定内容(一部)を取得します。

(a) ErrorCode **GetProperty**(
IntPtr *devmode*,
PropertyId *id*,
byte[] *data*,
ref uint *size*)

(b) 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
取得する印刷設定内容のプロパティIDを指定します。
PropertyIdに定義されたプロパティIDを使用する場合は、(a)を指定してください。
プロパティIDを数値で指定する場合は、(b)を指定してください。
指定可能な内容は、「6.9 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ー
取得した印刷設定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
印刷設定内容を格納するバッファーサイズを指定します。
APIから制御が戻ると取得した印刷設定内容の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、「5.1 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。
印刷設定内容の変更は、PropertyIdに定義されたプロパティIDを用いる構文と、数値で指定する構文の2種類があります。
(a)の構文は、PropertyIdに定義されたプロパティIDを使用し、印刷設定内容(一部)を変更します。
(b)の構文は、プロパティIDを数値で指定し、印刷設定内容(一部)を変更します。

(a) ErrorCode **SetProperty**(
IntPtr *devmode*,
PropertyId *id*,
byte[] *data*,
ref uint *size*)

(b) ErrorCode **SetProperty**(
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
変更する印刷設定内容のプロパティIDを指定します。
PropertyIdに定義されたプロパティIDを使用する場合は、(a)を指定してください。
プロパティIDを数値で指定する場合は、(b)を指定してください。
指定可能な内容は、「6.9 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ー
変更する印刷設定内容が格納されたバッファーを指定します。

size
設定データサイズ
変更する印刷設定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

エラーコードを返します。エラーコードの内容は、「5.1 エラーコード一覧」を参照してください。

4.3.6 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が格納されます。
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。
- 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。
- 本イベントを停止する場合は下記のAPIを呼び出してください。
  - **CloseMonPrinter**
  - **CancelStatusBack**
- 登録されたイベントハンドラー内から下記のAPIを呼び出すことはできません。
  - **CloseMonPrinter**
  - **SetStatusBack**
  - **CancelStatusBack**
- 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下記の通りです。

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	・ネットワーク上で到達できない、もしくはアクセス中にサービスが停止されるなどの理由でプリンターにアクセスできません。 ・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_NON_TARGETTED_DRIVER	−120	指定したプリンタードライバーは対象外です。
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_IMAGE_FAILED	−250	ラベルイメージの作成に失敗しました。

マクロ定義（定数）	値	内容
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、またはSmart Label Printer RT series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ないのに、UnlockPrinterが実行されました。
ERR_INVALID_DATA	−1020	不正なデータが指定されています。
ERR_READ_FAULT	−1030	プリンターからの応答がありません。
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_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。サンプルプログラムのプリンターステータス処理も参考にしてください。

プリンターステータス	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
カッタエラー	ASB_AUTOCUTTER_ERR	0x00000000	カッタエラーなし
		0x00000008	カッタエラーあり
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
マーク検出時紙ジャムエラー	ASB_MARK_PAPER_JAM_ERR	0x00000000	紙ジャムエラーなし
		0x00000040	紙ジャムエラーあり
カバーオープンエラー	ASB_COVER_OPEN	0x00000000	ペーパーカバーが閉じている
		0x00000080	ペーパー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	動作中

プリンターステータス	対応ビット		内容
復帰待ち状態	ASB_RETURN_WAITING	0x00000000	–
		0x00000800	復帰待ち状態
テイクンセンサの状態	ASB_TAKEN_SENSOR	0x00000000	紙取り出し済み
		0x00002000	紙取り出し待ち
ドロワスイッチ入力の状態	ASB_DRAWER_KICK	0x00000000	Low※1
		0x00008000	High
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
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	0x00000000	プリンター応答
		0x80000000	未接続、または通信エラー

※1: SLP720RTはLowに固定です。
※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。

6.2 カウンターID

カウンターIDと、その内容は下記の通りです。

カウンターID		内容	ResetCounterによる 初期化
.NET API	.NET API Win32 API		
ROLL_FEED_LINES	20	紙送り行数	可
ROLL_HEAD_CHARGE	21	サーマルヘッド通電回数	可
PAPER_CUT	50	オートカッタ駆動回数	可
OPERATION_TIME	70	製品の駆動時間	可
ROLL_FEED_LINES※1	148	紙送り行数(積算)	不可
ROLL_HEAD_CHARGE※1	149	サーマルヘッド通電回数(積算)	不可
PAPER_CUT※1	178	オートカッタ駆動回数(積算)	不可
OPERATION_TIME※1	198	製品の駆動時間(積算)	不可

※1: .NET APIの場合、積算カウンターのカウンター値の取得は**GetCounter**の*type*で指定します。APIの詳細は、.NET API**GetCounter**を参照してください。

6.3 プリンターID

プリンターIDと、その内容は下記の通りです。

プリンターID		内容	応答形式
.NET API	.NET API Win32 API		
DEVICE	1	機種ID	数値（1バイト） 0x28
TYPE	2	タイプID	数値（1バイト） 「6.4 タイプID」を参照
ROM_VER	3	ROMバージョンID	数値（1バイト）
FW_VER_MAIN	65	ファームウェアバージョン（main）	ASCII文字列 ″x.xx.xx″
MFG	66	メーカー名	ASCII文字列 ″Seiko Instruments Inc.″
MDL	67	モデル名	ASCII文字列 ″SII SLP720RT Series.″
FONT	69	多国語フォントの種類	ASCII文字列 ″KANJI JAPANESE″
FW_VER_BOOT	97	ファームウェアバージョン（boot）	ASCII文字列 ″x.xx.xx″
FW_SUM_BOOT	98	ファームウェアチェックサム（boot）	数値（2バイト）
FW_SUM_MAIN	99	ファームウェアチェックサム（main）	数値（2バイト）
FW_SUM	100	ファームウェアチェックサム（main + boot）	数値（2バイト）

6.4 タイプID

タイプIDの応答内容の対応ビットと、その内容は下記の通りです。
タイプIDとはプリンターコマンド「プリンタIDの送信」によって取得されるプリンター情報の一部です。

対応ビット	内容	
0x01	紙幅選択	0 ： 58 mm 1 ： 80 mm
0x02	予約済み	1に固定
0x04	SLP720RT（予約済み） SLP721RT（予約済み）	0に固定 1に固定
0x08	未定義	0に固定
0x10	予約済み	0 ： 無し 1 ： 有り
0x20	予約済み	0 ： 無し 1 ： 有り
0x40	予約済み	0に固定
0x80	未定義	0に固定

取得できる応答内容は上記の値を合計した値です。
例: 応答値 = 0x13

6.5 対象ドロワ

対象ドロワと、その内容は下記の通りです。

対象ドロワ		内容
.NET API	.NET API Win32 API	
DRAWER_1	1	ドロワ1を駆動
DRAWER_2	2	ドロワ2を駆動

6.6 ドロワ駆動時間

ドロワ駆動時間と、その内容は下記の通りです。

対象ドロワ		内容
.NET API	.NET API Win32 API	
TIME_100	1	100 ms ドロワを駆動
TIME_200	2	200 ms ドロワを駆動
TIME_300	3	300 ms ドロワを駆動
TIME_400	4	400 ms ドロワを駆動
TIME_500	5	500 ms ドロワを駆動
TIME_600	6	600 ms ドロワを駆動
TIME_700	7	700 ms ドロワを駆動
TIME_800	8	800 ms ドロワを駆動

ドロワの制御時間については、お客様がご利用されているドロワの仕様に従ってください。

6.7 プリンターの機能

プリンターの機能と、その内容は下記の通りです。

カウンターID		内容	引数1	引数2
.NET API	.NET API Win32 API			
EXT_BUZZER	1	外部ブザーを鳴動	ブザーパターン※1 0 : パターン1 1 : パターン2 2 : パターン3 3 : パターン4	ブザー鳴動回数※2 1 ～ 255(回)

※1: ブザーパターンについては「SLP721RT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。
※2: 外部ブザーの鳴動は下記のいずれかの条件で停止します。
・設定したブザーの回数分の鳴動

・カバーオープン
・プリンターコマンド「外付けブザーの停止」の実行

6.8 ラベルファイル印刷時用紙ID

ラベルファイル印刷時用紙IDと、その内容は下記の通りです。

マークあり・なし用紙選択		内容
.NET API	.NET API Win32 API	
MODE_MARK	0	PrintSmartLabelImageData実行時に用紙の逆方向紙送りを伴う印字開始位置への紙送りを行います。
MODE_NONE_MARK	1	PrintSmartLabelImageData実行時に印字開始位置への紙送りは行いません。

6.9 プロパティID

注意

◆ SetPropertyを呼び出す場合、ロゴキーコード及び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。

プロパティIDと、その内容は下記の通りです。
[ ]カッコ内はオプション設定において当該処理が行われるタイミングを示します。オプション設定については、SLP720RT/SLP721RTシリーズ用「SII Printer Driver for Windows 取扱説明書」の「3.3.7 オプションの設定」を参照してください。

プロパティID		内容	
.NET API	.NET API Win32 API		
INIT	1	初期化	0 : する 1 : しない
SPEED	2	速度	0 : 高速※1 1 : 中速※1
MARGIN	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
DENSITY	4	濃度(パーセント)	70 ～ 130
DIRECTION	5	方向	0 : 順 1 : 逆
REDUCTION	6	縮小	0 : 縮小しない 15 ～ 100 : 指定倍率(パーセント)

プロパティID		内容	
.NET API	.NET API Win32 API		
CUT	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)
MARK_FEED	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
CUT_FEED	9	カット位置までフィード	0 : する 1 : しない
START_DOC_LOGO	10	[印刷開始] ロゴ	0 : なし 1 : 左寄せ 2 : 中央 3 : 右寄せ
START_DOC_LOGO_KEY	11	[印刷開始] ロゴキーコード (2バイト)※2	各バイト 0x20 ～ 0x39
START_DOC_DRAWER	12	[印刷開始] ドロワ※3	0 : 利用しない 1 : ドロワ1を利用 2 : ドロワ2を利用
START_DOC_DRAWER_ON	13	[印刷開始] ON時間(x 2 ms)※4	1 ～ 255
START_DOC_DRAWER_OFF	14	[印刷開始] OFF時間(x 2 ms)	1 ～ 255
START_DOC_CMD	15	[印刷開始] カスタムコマンド (128バイト)	コマンドデータ
START_DOC_BACK_FEED	16※5	[印刷開始] 紙送り(バックフィード)(dot)	0 ～ -68
START_DOC_FEED	17※5	[印刷開始] 紙送り(フィード) (dot)	0 ～ 255
START_DOC_EXT_BUZZER	110	[印刷開始] 外部ブザー	0 : パターン1 1 : パターン2 2 : パターン3 3 : パターン4 255: OFF
START_DOC_EXT_BUZZER_RPT	111	[印刷開始] 外部ブザー 鳴動回数※6	1 ～ 255

プロパティID		内容	
.NET API	.NET API Win32 API		
START_PAGE_LOGO	20	[ページ開始] ロゴ	0 : なし 1 : 左寄せ 2 : 中央 3 : 右寄せ
START_PAGE_LOGO_KEY	21	[ページ開始] ロゴキーコード (2バイト)※2	各バイト 0x20 ～ 0x39
START_PAGE_CMD	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
START_PAGE_BACK_FEED	26※5	[ページ開始] 紙送り(バックフィード)(dot)	0 ～ -68
START_PAGE_FEED	27※5	[ページ開始] 紙送り(フィード) (dot)	0 ～ 255
START_PAGE_EXT_BUZZER	120	[ページ開始] 外部ブザー	0 : パターン1 1 : パターン2 2 : パターン3 3 : パターン4 255: OFF
START_PAGE_EXT_BUZZER_RPT	121	[ページ開始] 外部ブザー 鳴動回数※6	1 ～ 255
END_PAGE_LOGO	30	[ページ終了] ロゴ	0 : なし 1 : 左寄せ 2 : 中央 3 : 右寄せ
END_PAGE_LOGO_KEY	31	[ページ終了] ロゴキーコード (2バイト)※2	各バイト 0x20 ～ 0x39
END_PAGE_CMD	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
END_PAGE_BACK_FEED	36※5	[ページ終了] 紙送り(バックフィード)(dot)	0 ～ -68
END_PAGE_FEED	37※5	[ページ終了] 紙送り(フィード) (dot)	0 ～ 255
END_PAGE_EXT_BUZZER	130	[ページ終了] 外部ブザー	0 : パターン1 1 : パターン2 2 : パターン3 3 : パターン4 255: OFF

プロパティID		内容	
.NET API	.NET API Win32 API		
END_PAGE_EXT_BUZZER_RPT	131	[ページ終了] 外部ブザー 鳴動回数※6	1 ～ 255
END_DOC_LOGO	40	[印刷終了] ロゴ	0 : なし 1 : 左寄せ 2 : 中央 3 : 右寄せ
END_DOC_LOGO_KEY	41	[印刷終了] ロゴキーコード (2バイト)※2	各バイト 0x20 ～ 0x39
END_DOC_DRAWER	42	[印刷終了] ドロワ※3	0 : 利用しない 1 : ドロワ1を利用 2 : ドロワ2を利用
END_DOC_DRAWER_ON	43	[印刷終了] ON時間(x 2 ms)※4	1 ～ 255
END_DOC_DRAWER_OFF	44	[印刷終了] OFF時間(x 2 ms)	1 ～ 255
END_DOC_CMD	45	[印刷終了] カスタムコマンド (128バイト)	コマンドデータ
END_DOC_BACK_FEED	46※5	[印刷終了] 紙送り(バックフィード)(dot)	0 ～ -68
END_DOC_FEED	47※5	[印刷終了] 紙送り(フィード) (dot)	0 ～ 255
END_DOC_EXT_BUZZER	140	[印刷終了] 外部ブザー	0 : パターン1 1 : パターン2 2 : パターン3 3 : パターン4 255: OFF
END_DOC_EXT_BUZZER_RPT	141	[印刷終了] 外部ブザー 鳴動回数※6	1 ～ 255
PAPER_SIZE	50	用紙サイズ	0 : Letter 1 : A4 3 : 58 mm (54×3276 mm) 7 : 58 mm (52×3276 mm) 8 : 40 mm (36×3276mm) 9 ～ 255 : 上記以外の用紙※7
ORIENTATION	51	印刷の向き	0 : 縦 1 : 横
COLOR_MODE	52	カラー印刷モード	0 : システム※8 1 : ドライバー※8

プロパティID		内容	
.NET API	.NET API Win32 API		
PRESET	60※9	プリセット	1 (116) : 58 mm レシート設定※10 118 : 58 mm マーク頭出し設定(ラベル紙) 5 (119) : A4 -> 58 mm 縮小設定※10 80 : 40 mm レシート設定 81 : 40 mm マーク紙設定 83 : A4 -> 40 mm 縮小設定 6 (255) : ユーザー設定

※1: 印字速度についての詳細は、「取扱説明書」を参照してください。
※2: 格納順は、上位桁、下位桁の順です。
※3: 印刷開始と印刷終了の両方でドロワを駆動することはできません。
※4: ドロワの制御時間については、お客様がご利用されているドロワの仕様に従ってください。
※5: 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。
※6: 外部ブザーの鳴動は下記のいずれかの条件で停止します。
・設定したブザーの回数分の鳴動
・カバーオープン
・プリンターコマンド「外付けブザーの停止」の実行
※7: 設定することはできません。取得のみ行えます。
※8: カラー印刷モードの詳細は、SLP720RT/SLP721RT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。
※9: 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。プリセットの詳細は、SLP720RT/SLP721RT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。
※10: 設定では、2種類あるパラメーターはどちらも使用できます。取得では、括弧の無い値を取得します。

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。 ・ ″=″、″#″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	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。 ・ 送信データに応答を無効にするASB設定コマンド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。 ・ 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「プリンタの初期化」以外のプリンターを初期化するプリンターコマンドを含めないでください。プリンターの初期化については、「技術説明書」を参照してください。ハードウェアリセットする場合は、**Reset**、または**ResetPrinter**を実行してください。 ・ 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下記の通りです。 ・ 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	コマンド定義の説明などを記述します。 ・ 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.
```

参考

- 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、登録済みのコマンド定義は上書きされます。

8章　ラベル印字機能

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

Smart Label Creatorはラベル作成ができるソフトウェアです。
SII Layout EditorはiOSまたはAndroidでラベル作成ができるソフトウェアです。

本書では、Smart Label Creator及びSII Layout Editorの共通の説明をする場合は、以降「アプリ」と称します。
個別に説明する必要がある場合は製品名を記載します。

アプリは下記のWebページからダウンロードができます。

- SLP720RT/SLP721RTシリーズ ダウンロードページ
  https://www.sii-ps.com/slp720rt/

SII Layout Editorは、下記のQRコードをスマートフォンで読み取ってストアへ移動すると、アプリをインストールできます。

- iOS

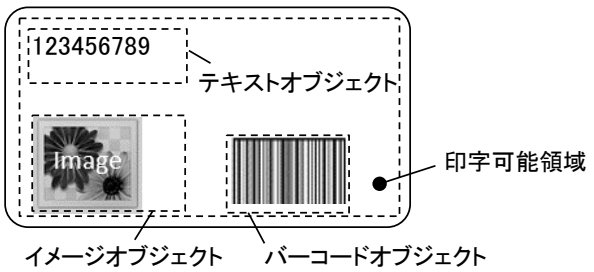
- Android

参考

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

8.1 ラベルファイルの構造

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



ラベルファイル （*.sl）の例

8.1.1 オブジェクトの種類とSDKでのサポート

SDKでサポートできるオブジェクトを下記に示します。

オブジェクト	説明	ラベルファイル（*.sl）作成元			SDKでのサポート
		SII Layout Editor	Smart Label Creator		
		iOS/Android	iOS/Android	Windows	
テキストオブジェクト	テキストデータを扱うオブジェクト	✓	✓	✓	✓
イメージオブジェクト	イメージデータを扱うオブジェクト	✓	✓	✓	✓
バーコードオブジェクト	バーコードデータを扱うオブジェクト	✓	✓	✓	✓
図（四角形）オブジェクト	図（四角形）の描画データを扱うオブジェクト	✓	―	✓	✓
図（円/楕円）オブジェクト	図（円/楕円）の描画データを扱うオブジェクト	✓	―	✓	✓
図（線）オブジェクト	図（線）の描画データを扱うオブジェクト	✓	―	✓	✓
飾り枠オブジェクト	飾り枠の描画データを扱うオブジェクト	―	―	✓	✓
連絡帳オブジェクト	デバイスの連絡帳からデータを取り込むオブジェクト	✓	✓	―	―
日付と時刻オブジェクト	日付と時刻のデータを扱うオブジェクト	✓	✓	✓	✓
差出人オブジェクト	差出人データを扱うオブジェクト	―	―	✓	✓
グループオブジェクト	複数のオブジェクトをグループ化したオブジェクト	―	―	✓	✓

8.1.2 ラベルファイルをSDKで印字する際の注意点

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

1) 全てのオブジェクト

- 印字可能領域外に配置されたオブジェクトの描画はサポートしていません。

2) テキストオブジェクト

アプリ

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

Windows Smart Label Creator

- 「Zip code」はサポートしていません。
- 「フィールドリンク」はサポートしていません。
- テキストの一部に対して装飾の解除や異なる装飾及びフォントサイズを付与する機能はサポートしていません。一文字目に付与されている装飾及びフォントサイズがテキスト全体の装飾となります。

3) イメージオブジェクト

Windows Smart Label Creator

- イメージソースが「ファイルのリンク」の場合、SDKではイメージオブジェクトの印字ができません。
- イメージソースが「クリップアート」の場合、SDKではイメージオブジェクトの印字ができません。

4) バーコードオブジェクト

アプリ

- アプリでサポートするバーコードのうち、SDKでは下記のバーコードをサポートしています。
  - CODE39
  - ITF
  - CODE128
  - UPC-A
  - EAN13
  - CODABAR
  - UPC-E
  - EAN8
  - EAN128
  - GS1 Databar Omni-directional
  - GS1 Databar Truncated
  - GS1 Databar Stacked
  - GS1 Databar Stacked Omni-directional
  - GS1 Databar Limited
  - GS1 Databar Expanded
  - GS1 Databar Expanded Stacked
  - Maxicode
  - PDF417
  - Data Matrix

- QRコード

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

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

Windows Smart Label Creator

- 「フィールドリンク」はサポートしていません。
- 下記のバーコード設定は反映しません。
  - PDF417のモード
  - Data Matrixのセキュリティ
  - QR Codeのモード

- DataMatrixの「モード」を1（長方形）に設定した場合、SDKではバーコードをオブジェクトの真ん中に描画します。

5) 図オブジェクト

SII Layout Editor 及び Windows Smart Label Creator

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

6) 日付と時刻オブジェクト

アプリ

- アプリで設定されているフォントがシステムにない場合、テキストはシステム標準のフォントで印字します。

7) 差出人オブジェクト

アプリ

- アプリで設定されているフォントがシステムにない場合、テキストはシステム標準のフォントで印字します。

8) グループオブジェクト

Windows Smart Label Cretor

- Smart Label CreatorとSDKではオブジェクトの描画位置が異なる場合があります。

8.2 ラベルファイルの利用手順

ラベルファイルを利用した印字手順を下記に示します。

8.2.1 ラベルファイルをそのままSDKから印字する場合

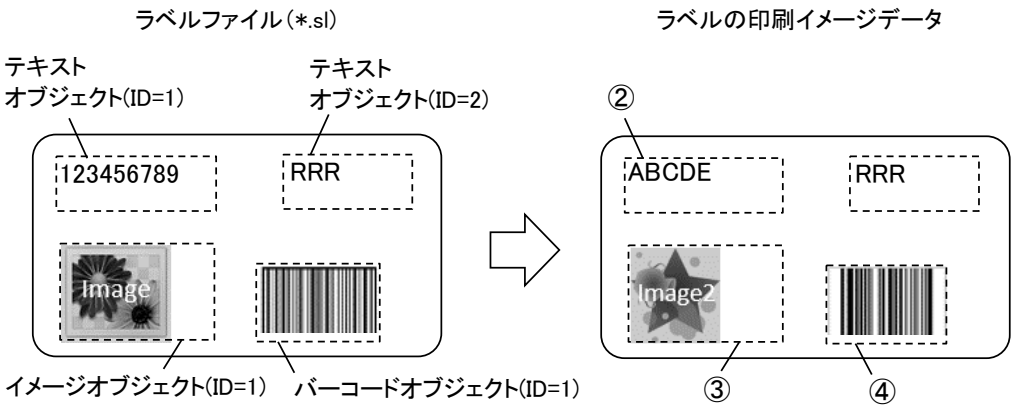
印字命令例（Win32）

1. ラベルファイルの指定
```
SelectSmartLabelFileW( i_hdl, filepath );
```
2. ラベルの印字
```
PrintSmartLabelImageData( i_hdl, timeout, 0 );
```

印字命令例（.NET）

1. ラベルファイルの指定
```
StatusAPI.SelectSmartLabelFile( filepath );
```
2. ラベルの印字
```
StatusAPI.PrintSmartLabelImageData( timeout, 0 );
```

8.2.2 ラベルファイルのオブジェクトデータを置換して印字する場合



印字命令例（Win32）

1. ラベルファイルの指定
```
SelectSmartLabelFileW( i_hdl, filepath );
```
2. ラベルのテキストデータの置換（テキストオブジェクト ID=1）
```
ReplaceSmartLabelTextData( i_hdl, 1, textData );
```
3. ラベルのイメージデータの置換（イメージオブジェクト ID=1）
```
ReplaceSmartLabelImageData( i_hdl, 1, filepath );
```
4. ラベルのバーコードデータの置換（バーコードオブジェクト ID=1）
```
ReplaceSmartLabelBarcodeData( i_hdl, 1, barcodeData );
```
5. ラベルの印字
```
PrintSmartLabelImageData( i_hdl, timeout, 0 );
```

印字命令例(.NET)

1. ラベルファイルの指定

```
StatusAPI.SelectSmartLabelFile( filepath );
```

2. ラベルのテキストデータの置換（テキストオブジェクト ID=1）

```
StatusAPI.ReplaceSmartLabelTextData( 1, textData );
```

3. ラベルのイメージデータの置換（イメージオブジェクト ID=1）

```
StatusAPI.ReplaceSmartLabelImageData( 1, filepath );
```

4. ラベルのバーコードデータの置換（バーコードオブジェクト ID=1）

```
StatusAPI.ReplaceSmartLabelBarcodeData( 1, barcodeData );
```

5. ラベルの印字

```
StatusAPI.PrintSmartLabelImageData( timeout, 0 );
```

SII

セイコーインスツル株式会社

プリントシステム事業部

千葉県千葉市美浜区中瀬1-8　〒261-8507

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

ホームページ : https://www.sii-ps.com

大阪営業所

大阪府大阪市北区豊崎3-2-1 淀川5番館　〒531-0072

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

◎本書の内容は、製品の改良に伴い、予告なしに変更することがあります。