

PrintStream[®] for Mobile

Ver2.1

利用ガイド・API リファレンス

1. はじめに	1
1.1 表記について	1
2. 動作環境	2
3. 概要	3
3.1 ファイル構成.....	3
3.2 PrintStream for Mobile ブロック図.....	4
4. Biz/Browser Mobile(CRS)からの呼び出し	5
4.1 PrintStream.dll リファレンス.....	5
4.1.1 PSInit	6
4.1.2 PSFinish	9
4.1.3 PSDrawFile	10
4.1.4 PSDrawMem.....	11
4.1.5 PSGetAPIVersion.....	13
4.1.6 PSSetMargin	14
4.1.7 PSSetScaling	15
4.1.8 PSGetDriverAPIVersion.....	16
4.2 PSPreview.dll リファレンス	17
4.2.1 PSPreviewFile	17
4.2.2 PSPreviewMem	19
4.2.3 PSGetAPIVersion.....	21
5. C/C++からの呼び出し	22
5.1 ヘッダファイル、インポートライブラリを利用する方法	22
5.2 動的ロードを利用する方法	22
5.3 PrintStream.dll リファレンス.....	23
5.3.1 PSInitC	23
5.3.2 PSFinishC.....	26
5.3.3 PSDrawFileC.....	27
5.3.4 PSDrawMemC.....	28
5.3.5 PSGetAPIVersionC	30
5.3.6 PSSetMarginC.....	31
5.3.7 PSSetScalingC.....	32
5.3.8 PSGetDriverAPIVersionC	33
5.3.9 PSGetErrorMessageC.....	34
5.4 PSPreview.dll リファレンス	35
5.4.1 PSPreviewFileC.....	35
5.4.2 PSPreviewMemC.....	37
5.4.3 PSPGetAPIVersionC.....	39

5.4.4 PSPGetErrorMessageC.....	40
6. .NET アプリケーションからの呼び出し.....	41
6.1 DllImport を利用する方法.....	41
6.2 PrintStream.dll リファレンス.....	42
6.2.1 PSInitC	42
6.2.2 PSFinishC.....	45
6.2.3 PSDrawFileC.....	46
6.2.4 PSDrawMemC.....	47
6.2.5 PSGetAPIVersionC	48
6.2.6 PSSetMarginC.....	49
6.2.7 PSSetScalingC.....	50
6.2.8 PSGetDriverAPIVersionC	51
6.2.9 PSPGetErrorMessageC.....	52
6.3 PSPreview.dll リファレンス	53
6.3.1 PSPreviewFileC.....	53
6.3.2 PSPreviewMemC.....	55
6.3.3 PSPGetAPIVersionC.....	57
6.3.4 PSPGetErrorMessageC.....	58
7. 対応プリンター一覧.....	59
8. PSPreview.dll プレビュー画面操作方法	60
8.1 ポップアップメニューを出す方法.....	60
8.2 スクロール.....	60
8.3 表示倍率変更.....	60
8.4 ページ移動.....	60
8.5 プレビューの終了.....	60
9. コーディング例.....	61
9.1 Biz/Browser Mobile からの呼び出し.....	61
9.1.1 SVG ファイルを COM4 ポートにマッピングした Bluetooth 経由で印刷する場合.....	61
9.1.2 メモリ上の PSS データを IrDA で印刷する場合	61
9.1.3 SVG ファイルをカラーでプレビューする場合.....	62
9.1.4 メモリ上の PSS データをモノクロでプレビューする場合.....	62
9.2 VC++からの呼び出し	63
9.2.1 SVG ファイルを COM4 ポートにマッピングした Bluetooth 経由で印刷する場合	63
9.2.2 メモリ上の PSS データを IrDA で印刷する場合	64
9.2.3 SVG ファイルをカラーでプレビューする場合.....	65
9.2.4 メモリ上の PSS データをモノクロでプレビューする場合.....	65

9.3 VC#からの呼び出し	66
9.3.1 SVG ファイルを COM4 ポートにマッピングした Bluetooth 経由で印刷する場合	66
9.3.2 メモリ上の PSS データを IrDA で印刷する場合	67
9.3.3 SVG ファイルをカラーでプレビューする場合	68
9.3.4 メモリ上の PSS データをモノクロでプレビューする場合	69

1. はじめに

PrintStream® for Mobile をご利用いただき、誠に有難うございます。

本ガイドは PrintStream® for Mobile の基本的な使用方法について理解していただくことを目的として作成されております。

本製品の導入に向けて本ガイドが、皆様のお役に立つことを心より願っております。

- 本ガイドは PrintStream® for Mobile を使った印刷・プレビュー方法を具体的な例を示しながら説明しています。
- 本ガイド および、付属のソフトウェアに関する所有権、知的財産権、その他全ての権利 および、権限は、アクシスソフト株式会社が所有しています。
- 本ガイド および、付属のソフトウェアは使用許諾契約書に記載されている内容に反して使用または複製することはできません。
- 本ガイドにより提供する情報やソフトウェアに間違いやバグが含まれていた場合にも、弊社はこれに起因して生じる直接的または間接的な損失、損害等について一切責任を負いません。
- 本ガイドに記載されている内容 および、付属のソフトウェアの仕様は、将来予告なく変更することがあります。
- Windows は米国マイクロソフト社の登録商標です。
- その他の製品名は各社の商標または登録商標です。

1.1 表記について

- ヒントについて
操作方法や、設定内容に関する参考情報は、ヒントマークで表示します。
- 注意について
特に注意が必要な情報については注意マークで表示します。



2. 動作環境

動作確認済みの OS は以下の通りです。

- Windows Mobile 5.0 ~ 6.1
- Windows CE 5.0、6.0

動作確認済みの Biz/Browser Mobile は以下の通りです。

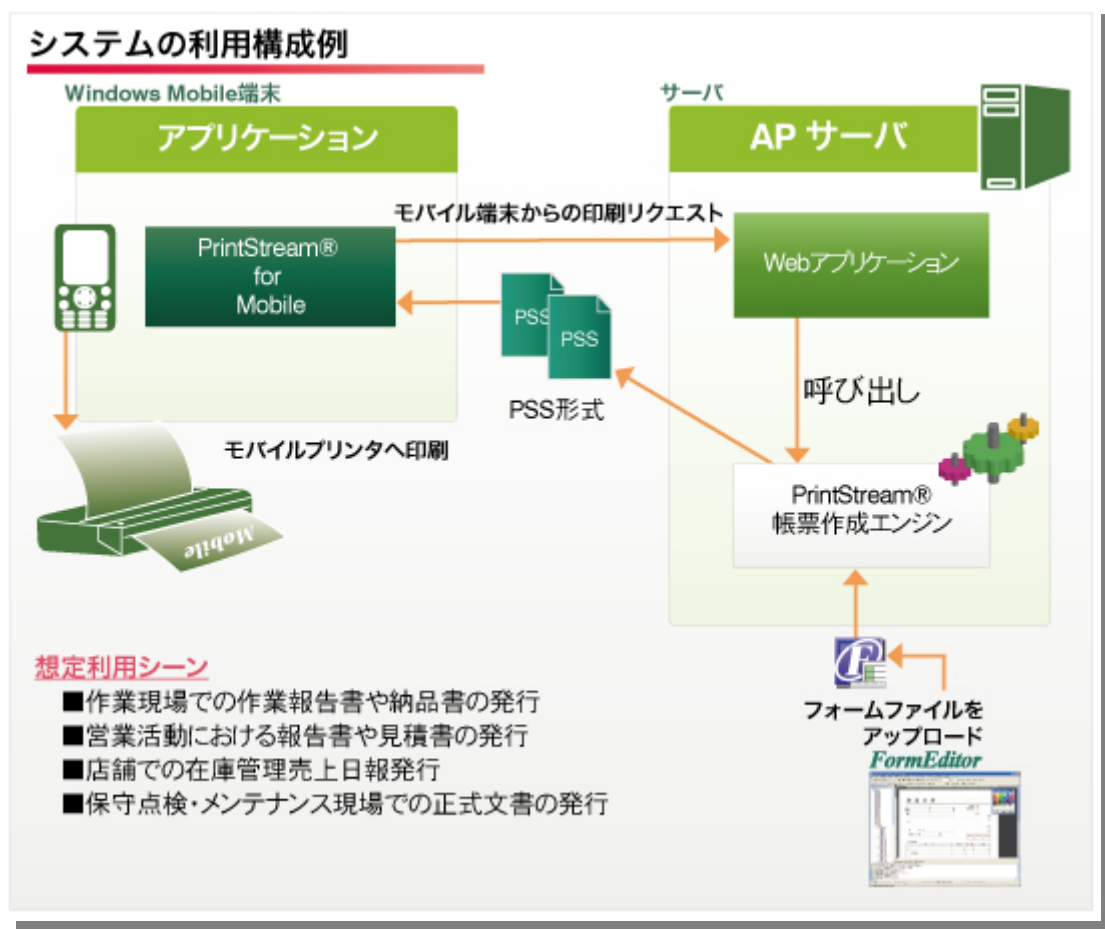
- V3.2.0.0 以降

3. 概要

PrintStream® for Mobile は Biz/PrintServer または PrintStream® Core SE により生成された帳票イメージ(PSS、SVG)をモバイル端末上で表示、または各種プリンタでの印刷を行うためのライブラリ(DLL)です。

DLL は Biz/Browser Mobile プラグイン形式の API、および C/C++/.NET から呼び出すための API を公開しています。

この API を利用し Biz/Browser Mobile および C、C++ネイティブアプリケーション、.NET アプリケーションから呼び出すことが可能です。



印刷時は PrintStream.dll を画面表示の際は PSPreview.dll を使用します。
PrintStream.dll は実行時に各プリンタ用の抽象化ドライバを介し印刷を行います。
各 DLL の API の詳しい解説は次章以降をご参照ください。

3.1 ファイル構成

PrintStream® for Mobile は標準で以下のファイルで構成されています。
各 DLL は必要に応じてドライバをリンクするため、各 DLL は同じフォルダか、または Windows フォルダ内に配置してください。

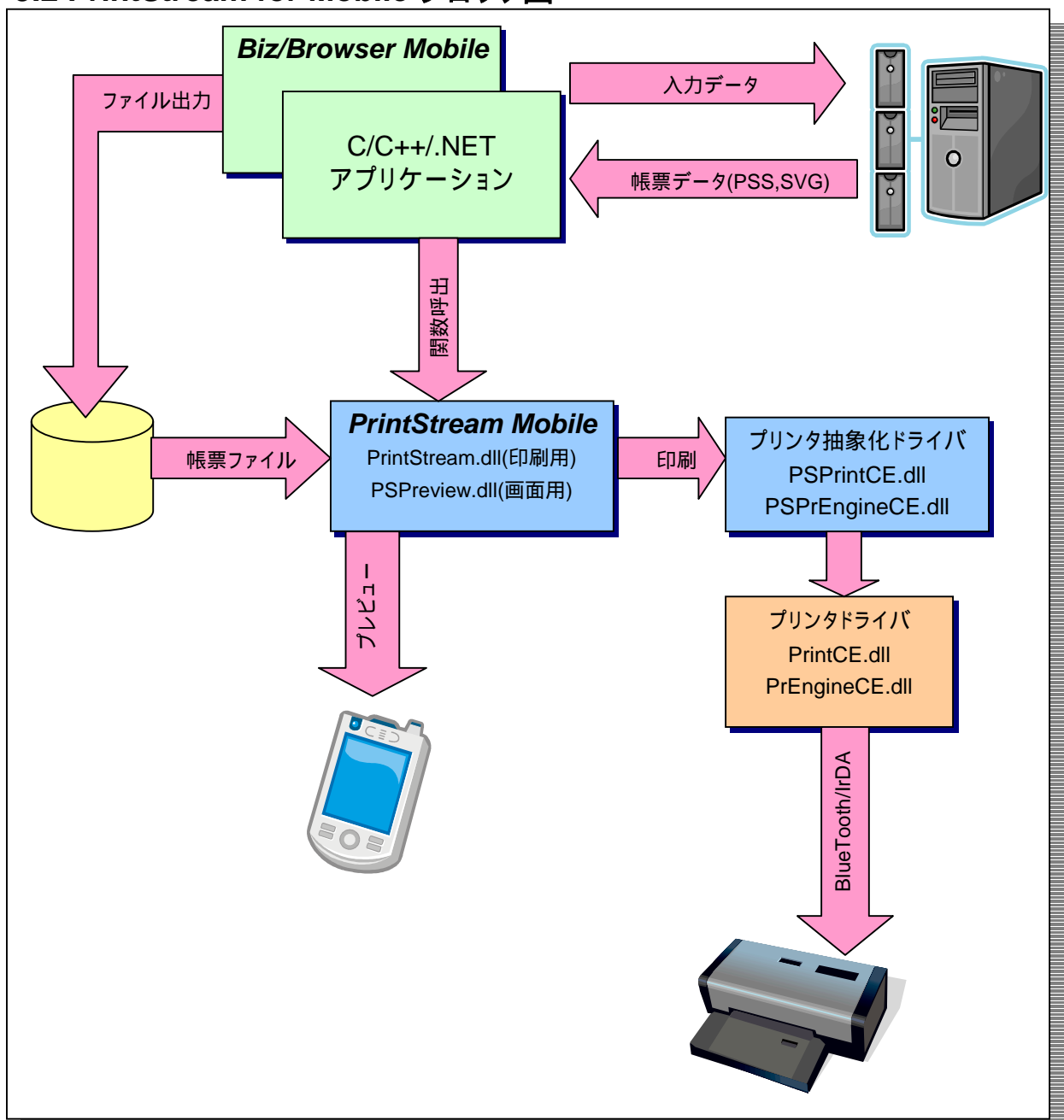
PrintStream.dll	印刷用 PrintStream® for Mobile メインモジュール
PSPreview.dll	画面表示用 PrintStream® for Mobile メインモジュール

PSPrintCE.dll, PSPrEngineCE.dll	プリンタ抽象化ドライバ
PrintCE.dll, PrEngineCE.dll	プリンタドライバ

インポートライブラリ(lib ファイル)を用いて開発する場合、すべてのDLL はWindows フォルダ内に配置する必要があります。

またランタイムライブラリとして Visual C++ 2005 のライブラリ msvc80.dll、MFC80U.dll が必要になります。

3.2 PrintStream for Mobile ブロック図



4. Biz/Browser Mobile(CRS)からの呼び出し

PrintStream.dll および PSPreview.dll は Biz/Browser Mobile プラグイン形式のインタフェースを持っているため Runtime オブジェクトを用い簡便に使用することができます。

Biz/Browser Mobile のプラグインとして使用する場合 Biz/Browser Mobile のインストールフォルダに DLL というフォルダを作成しその中に DLL を配置します。

Biz/Browser Mobile は Runtime オブジェクト作成時に自動的に DLL フォルダ内の DLL をリンクします。

4.1 PrintStream.dll リファレンス

以下の解説は

```
var ps = new Runtime("PrintStream.dll");
```

により変数 ps に Runtime オブジェクトを設定した状態を想定しています。

4.1.1 PSInit

プリンタドライバを初期化し、印刷情報を取得します。以降、取得した印刷情報を用い印刷を実行します。

本関数を実行した後は必ず PSFinish によりプリンタドライバの終了処理を行ってください。

呼出形式 `var printInfo = ps.CallDll("PSInit", driverDll, portType, portName, printer [,paperSize [, orientation [, color [, silentMode [, draftMode [, contrast [, brightness [, density]]]]]]]]);`

戻り値 印刷情報(今後の印刷 API に必要)

引数	"PSInit"	呼び出し関数名(固定値)
	String driverDll	プリンタ抽象化ドライバを指定します。 通常標準添付のドライバのうち、使用するプリンタに合わせて"PSPrintCE.dll"か"PSPrEngineCE.dll"を指定します。
	Number portType	出力ポートタイプを指定します。 1 : COM ポート 3 : IrDA
	String portName	portType が 1 の時のみ有効です。ポート名+":"で指定します。 例)"COM4:"
	Number printer	印刷対象プリンタを指定します。指定可能な値はドライバにより異なります。 詳細は 対応プリンター一覧 に記載されています。
	Number paperSize	用紙サイズを指定します。 プリンタでサポートされないサイズを指定した場合、動作はプリンタに依存します。 ただし、 一部機種 では縦方向サイズ[mm]を指定します。 0 : A4(デフォルト) 1 : A5 2 : B5 3 : レター 4 : リーガル
	Number orientation	印刷の向きを指定します。 一部機種 では縦方向に固定されます。 0 : 横 1 : 縦(デフォルト)
	Number color	印刷時の色モードを指定します。

- 0 : モノクロ
- 1 : 4色カラー(デフォルト)
- 2 : 3色カラー

0=モノクロを指定した場合、PrintStream® for Mobileはカラーをディザパターンに置き換え描画します。

Number **silentMode** 印刷進捗ダイアログを表示するかどうかを指定します。
 0 : 表示する(デフォルト)
 1 : 表示しない

Number **draftMode** ドラフト印刷モードを指定します。テスト印字などインク消費を抑えたい場合に使用します。
 0 : オフ(デフォルト)
 1 : オン

Number **contrast** 画像のコントラストを調整します。線や塗りつぶしには影響しません。
 0(弱) ~ 45(デフォルト) ~ 90(強)

Number **brightness** 画像の明るさを調整します。線や塗りつぶしには影響しません。
 -127(暗) ~ 0(デフォルト) ~ 127(明)

Number **density** "PSPrEngineCE.dll"使用時のみ
 印刷の濃度を調整します。
 -2(薄) ~ 0(標準) ~ +2(濃)

例外

メッセージ	内容
引数に誤りがあります。	引数の数や型が間違っています。
LoadLibrary でエラーが発生しました。エラーコード[126]	プリンタ抽象化ドライバが見つかりません。
[抽象化ドライバ DLL名]でエラー発生。<LoadLibrary に失敗しました。エラーコード[126]>	プリンタドライバが見つかりません。

[抽象化ドライバ DLL 名]でエラー発生。 < に失敗しました。 []> プリンタドライバで 公開している関数名 エラー内容	プリンタドライバ内でエラーが発生しました。
---	-----------------------

4.1.2 PSFinish

プリンタドライバをクローズし、印刷の終了処理を行います。

呼出形式 ps.CallDll("PSFinish", printInfo);

戻り値 なし

引数 "PSFinish" 呼び出し関数名(固定値)

Number **printInfo** PSInit で取得したプリンタ情報値を指定します。

例外	メッセージ	内容
	引数に誤りがあります。	引数の数や型が間違っています。

4.1.3 PSDrawFile

帳票イメージファイルを解析し印刷を行います。
PSDrawMem よりも少ない消費メモリで実行可能です。

本関数は、PSInit から PSFinish の間で何回でも呼び出せます。
また PSS を印刷する場合は、ファイルが連続する順番にすべてのファイルで、本関数を呼び出す必要があります。本制限以外の使用はサポート外となりますので、動作は不定となります。

呼出形式 var ret = ps.CallDll("PSDrawFile", printInfo, fileType, filePath);

戻り値 0 : 最後の PSS ファイルを処理もしくは PSS 以外のファイルを処理
1 : 次の PSS ファイルが必要

引数 "PSDrawFile" 呼び出し関数名(固定値)

Number **printInfo** PSInit で取得したプリンタ情報値を指定します。

Number **fileType** ファイル形式を指定します。
1 : SVG
2 : PSS

String **filePath** 帳票イメージファイルの絶対パスを指定します。

例外	メッセージ	内容
	引数に誤りがあります。	引数の数や型が間違っています。
	PSS が不正です。	PSS ファイルが壊れています。
	サポート外の PSS です。	PSS のバージョンがサポートしているバージョンより古いです。
	SVG が不正です。	SVG または PSS 内の SVG 部が壊れています。
	イメージファイルが見つかりません。	SVG 内に記述されているイメージファイルが見つかりません。
	メモリが不足しています。	帳票描画するためのメモリが足りません。
	[抽象化ドライバ DLL 名]でエラー発生。 < に失敗しました。 []> プリンタドライバで公開している関数名 エラー内容	プリンタドライバ内でエラーが発生しました。

4.1.4 PSDrawMem

メモリ上の帳票イメージデータを解析し印刷を行います。
帳票ファイルを作成しないで済むため PSDrawFile よりセキュリティレベルは高くなりますが、メモリをより消費します。

Biz/Browser Mobile Ver3.2.1 以前のバージョンでは CallDll 関数では DLL に対しバイナリデータを渡すことができないため、PSS データを印字する場合は一度 Base64Encoder にて Base64 形式に変換後、dataType に 4 を指定し、data には Base64 変換した PSS データを指定してください。

本関数は、PSInit から PSFinish の間で何回でも呼び出せます。
また PSS を印刷する場合は、ファイルが連続する順番にすべてのファイルで、本関数を呼び出す必要があります。本制限以外の使用はサポート外となりますので、動作は不定となります。

呼出形式 var ret = ps.CallDll("PSDrawMem", printInfo, dataType, data);

戻り値 0 : 最後の PSS ファイルを処理もしくは PSS 以外のファイルを処理
1 : 次の PSS ファイルが必要

引数

"PSDrawMem"	呼び出し関数名(固定値)
Number printInfo	PSInit で取得したプリンタ情報値を指定します。
Number dataType	データ形式を指定します。 1 : SVG 2 : PSS 4 : Base64 変換した PSS
String data または ByteArray data	帳票イメージデータを指定します。

例外	メッセージ	内容
	引数に誤りがあります。	引数の数や型が間違っています。
	PSS が不正です。	PSS ファイルが壊れています。
	サポート外の PSS です。	PSS のバージョンがサポートしているバージョンより古いです。
	SVG が不正です。	SVG または PSS 内の SVG 部が壊れています。
	イメージファイルが見つかりません。	SVG 内に記述されているイメージファイルが見つかりません。
	メモリが不足しています。	帳票描画するためのメモリが足りません。

[抽象化ドライバ DLL 名]でエラー発生。< に失敗しました。 []> プリンタドライバで 公開している関数名 エラー内容	プリンタドライバ内でエラーが発生しました。
--	-----------------------

4.1.5 PSGetAPIVersion

API の仕様バージョンを取得します。

呼出形式 `var ret = ps.CallDll("PSGetAPIVersion");`

戻り値 API バージョン番号

引数 "PSGetAPIVersion" 呼び出し関数名(固定値)

例外 なし

4.1.6 PSetMargin

プリンタのマージンを指定します。

PrintStream® for Mobile は通常原点座標を用紙の左上端として座標計算を行います。プリンタのハードウェアにより印刷の原点座標が用紙の左上端ではない場合、マージンを指定することによりこの誤差を補正することができます。

呼出形式 ps.CallDll("PSetMargin", printInfo, width, height);

戻り値 なし

引数 "PSetMargin" 呼び出し関数名(固定値)

Number **printInfo** PSInit で取得したプリンタ情報値を指定します。

Number **width** 横マージンを mm 単位で指定します。

Number **height** 縦マージンを mm 単位で指定します。

例外

メッセージ	内容
引数に誤りがあります。	引数の数や型が間違っています。
[抽象化ドライバ DLL 名]でエラー発生。 < に失敗しました。 []> プリンタドライバで公開している関数名 エラー内容	プリンタドライバ内でエラーが発生しました。

4.1.7 PSSetScaling

印刷の拡大縮小率を指定します。

拡大の場合、端末のメモリ残量により印刷に失敗する場合があります。

呼出形式 ps.CallDll("PSSetScaling", printInfo, scale);

戻り値 なし

引数 "PSSetScaling" 呼び出し関数名(固定値)

Number **printInfo** PSInit で取得したプリンタ情報値を指定します。

Number **scale** 拡大縮小率を[%]で指定します。
50(最小) ~ 100(デフォルト) ~ 200(最大)

例外

メッセージ	内容
引数に誤りがあります。	引数の数や型が間違っています。
[抽象化ドライバ DLL 名]でエラー発生。 < に失敗しました。 []> プリンタドライバで公開している関数名 エラー内容	プリンタドライバ内でエラーが発生しました。

4.1.8 PSGetDriverAPIVersion

プリンタ抽象化ドライバの API の仕様バージョンを取得します。

PSInit で指定するパラメータはドライバ固有になるため、このバージョンを確認することで抽象化ドライバのパラメータに変更があった場合でも処理を切り分けることが可能です。

呼出形式 ps.CallDll("PSGetDriverAPIVersion", driverDll);

戻り値 API バージョン(抽象化ドライバがこの関数に対応していない場合 0)

引数 "PSGetDriverAPIVersion" 呼び出し関数名(固定値)

String driverDll プリンタ抽象化ドライバを指定します。
通常標準添付のドライバのうち、使用するプリンタに合わせて "PSPrintCE.dll"か "PSPrEngineCE.dll"を指定します。

例外

メッセージ	内容
引数に誤りがあります。	引数の数や型が間違っています。
[抽象化ドライバDLL名]でエラー発生。< に失敗しました。 []> プリンタドライバで公開している関数名 エラー内容	プリンタドライバ内でエラーが発生しました。

4.2 PSPreview.dll リファレンス

PSPreview.dll は Biz/Browser Mobile プラグイン形式のインタフェースを持っているため Runtime オブジェクトを用い簡便に使用することができます。

以下の解説は

```
var ps = new Runtime("PSPreview.dll");

```

により変数 ps に Runtime オブジェクトを設定した状態を想定しています。

4.2.1 PSPreviewFile

帳票イメージファイルを解析しプレビューを行います。PSPreviewMem よりも少ない消費メモリで実行可能です。

プレビュー終了後まで関数から処理は戻りません。

本関数は、filePath1 以降に複数のファイルパスを指定できます。PSS をプレビューする場合は、ファイルが連続する順番にすべてのファイルを filePath1 以降に指定する必要があります。本制限以外の使用は、サポート外となりますので、動作は不定となります。

モノクロモードでは、プレビュー画面がモノクロ表示となります。メモリの消費を抑える事ができる為、メモリが不足している環境でのプレビューに有効です。

呼出形式 ps.CallDll("PSPreviewFile", fileType, monoMode, filePath1...n);

戻り値 なし

引数	"PSPreviewFile"	呼び出し関数名(固定値)
	Number fileType	ファイル形式を指定します。 1 : SVG 2 : PSS
	Number monoMode	モノクロモードにするかどうかを指定します。 0 : カラー 1 : モノクロ
	String filePath1...n	帳票イメージファイルの絶対パスを指定します。 PSS が複数に分割されている場合は、すべてを指定します。

例外	メッセージ	内容
	引数に誤りがあります。	引数の数や型が間違っています。
	LoadLibrary でエラーが発生しました。エラーコード[126]	PrintStream.dll が見つかりません。

PrintStream.dll でエラー発生。<PSS が不正です。>	PSS ファイルが壊れています。
PrintStream.dll でエラー発生。<サポート外の PSS です。>	PSS のバージョンがサポートしているバージョンより古いです。
PrintStream.dll でエラー発生。<SVG が不正です。>	SVG または PSS 内の SVG 部が壊れています。
PrintStream.dll でエラー発生。<イメージファイルが見つかりません。>	SVG 内に記述されているイメージファイルが見つかりません。
PrintStream.dll でエラー発生。<メモリが不足しています。>	帳票描画するためのメモリが足りません。

4.2.2 PSPreviewMem

メモリ上の帳票イメージデータを解析しプレビューを行います。
帳票ファイルを作成しないで済むため PSPreviewFile よりセキュリティレベルは高くなりますが、メモリをより消費します。

プレビュー終了後まで関数から処理は戻りません。

Biz/Browser Mobile Ver3.2.1 以前のバージョンでは CallDll 関数では DLL に対しバイナリデータを渡すことができないため、PSS データを印字する場合は一度 Base64Encoder にて Base64 形式に変換後、dataType に 4 を指定し、data1 ~ n には Base64 変換した PSS データを指定してください。

本関数は、data1 以降に複数のデータを指定できます。PSS をプレビューする場合は、データが連続する順番にすべてのデータを data1 以降に指定する必要があります。本制限以外の使用は、サポート外となりますので、動作は不定となります。

モノクロモードでは、プレビュー画面がモノクロ表示となります。メモリの消費を抑える事ができる為、メモリが不足している環境でのプレビューに有効です。

呼出形式 ps.CallDll("PSPreviewMem", dataType, monoMode, data1...n);

戻り値 なし

引数	"PSPreviewMem"	呼び出し関数名(固定値)
	Number dataType	データ形式を指定します。 1 : SVG 2 : PSS 4 : Base64 変換した PSS
	Number monoMode	モノクロモードにするかどうかを指定します。 0 : カラー 1 : モノクロ
	String data1...n または ByteArray data1...n	帳票イメージデータを指定します。

例外	メッセージ	内容
	引数に誤りがあります。	引数の数や型が間違っています。
	LoadLibrary でエラーが発生しました。エラーコード[126]	PrintStream.dll が見つかりません。
	PrintStream.dll でエラー発生。<PSS が不正です。>	PSS ファイルが壊れています。

PrintStream.dll でエラー発生。<サポート外の PSS です。>	PSS のバージョンがサポートしているバージョンより古いです。
PrintStream.dll でエラー発生。<SVG が不正です。>	SVG または PSS 内の SVG 部が壊れています。
PrintStream.dll でエラー発生。<イメージファイルが見つかりません。>	SVG 内に記述されているイメージファイルが見つかりません。
PrintStream.dll でエラー発生。<メモリが不足しています。>	帳票描画するためのメモリが足りません。

4.2.3 PSGetAPIVersion

API の仕様バージョンを取得します。

呼出形式 `var ret = ps.CallDll("PSGetAPIVersion");`

戻り値 API バージョン番号

引数 "PSGetAPIVersion" 呼び出し関数名(固定値)

例外 なし

5. C/C++からの呼び出し

この章では C/C++ネイティブアプリケーションから PrintStream.dll および PSPreview.dll を利用する方法について解説します。

5.1 ヘッダファイル、インポートライブラリを利用する方法

付属の PrintStream.h、PrintStream.lib、PSPreview.h、PSPreview.lib を開発環境に追加します。

簡単に利用できますが、DLL 配置場所は Windows フォルダ内に制限されます。

5.2 動的ロードを利用する方法

アプリケーションは必要に応じて PrintStream® for Mobile を Win32API の LoadLibrary で動的ロードし、関数ポインタを GetProcAddress で取得し実行します。実行後 DLL を FreeLibrary にて解放します。

手続きは増えますが DLL を任意の場所に配置できるメリットがあります。

呼び出しプロセス例としては以下ようになります

```
/* Biz/Browser Mobileプラグイン関数型 */
typedef (*PLUGINFUNC)(int, CRS_VARIANT**, CRS_VARIANT*);

/* DLLのローディングと関数ポインタの取得 */
HINSTANCE hDLL = LoadLibrary(_T("PrintStream.dll"));
PLUGINFUNC PSDrawFile = (PLUGINFUNC)GetProcAddress(hDLL, "PSDrawFile");

/* DLL解放 */
FreeLibrary(hDLL);
```

5.3 PrintStream.dll リファレンス

5.3.1 PSInitC

プリンタドライバを初期化し、印刷情報を取得します。以降、取得した印刷情報を用い印刷を実行します。

本関数を実行した後は必ず PSFinishC によりプリンタドライバの終了処理を行ってください。

呼出形式 int WINAPI PSInitC(
LPCTSTR pszDriverDLL,
int nPortType,
LPCTSTR pszPortName,
int nPrinter,
int nPaperSize,
int nOrientation,
int nColor,
int nSilentMode,
int nDraftMode,
int nContrast,
int nBrightness,
int nDensity,
int nReserved1
)

戻り値 0 以外 : 印刷情報(今後の印刷 API に必要)
0 : エラー

引数	LPCTSTR pszDriverDLL	プリンタ抽象化ドライバを指定します。 通常標準添付のドライバのうち、使用するプリンタに合わせて” PSPrintCE.dll”か”PSPrEngineCE.dll”を指定します。
	int nPortType	出力ポートタイプを指定します。 1 : COM ポート 3 : IrDA
	LPCTSTR pszPortName	ポート種別が 1 の時のみ有効です。ポート名+”:”で指定します。 例) ”COM4:”
	int nPrinter	印刷対象プリンタを指定します。指定可能な値はドライバにより異なります。 詳細は 対応プリンター一覧 に記載されています。
	int nPaperSize	用紙サイズを指定します。 プリンタでサポートされないサイズを指定した場合、動作はプリンタに依存します。 ただし、 一部機種 では縦方向サイズ[mm]を指定しま

	す。 0：A4(デフォルト) 1：A5 2：B5 3：レター 4：リーガル
int nOrientation	印刷の向きを指定します。 <u>一部機種</u> では縦方向に固定されます。 0：横 1：縦(デフォルト)
int nColor	印刷時の色モードを指定します。 0：モノクロ 1：4色カラー(デフォルト) 2：3色カラー 0=モノクロを指定した場合、PrintStream® for Mobileはカラーをディザパターンに置き換え描画します。
int nSilentMode	印刷進捗ダイアログを表示するかどうかを指定します。 0：表示する(デフォルト) 1：表示しない
int nDraftMode	ドラフト印刷モードを指定します。テスト印字などインク消費を抑えたい場合に使用します。 0：オフ(デフォルト) 1：オン
int nContrast	画像のコントラストを調整します。線や塗りつぶしには影響しません。 0(弱)～45(デフォルト)～90(強)
int nBrightness	画像の明るさを調整します。線や塗りつぶしには影響しません。 -127(暗)～0(デフォルト)～127(明)
int nDensity	印刷の濃度を調整します。 -2(薄)～0(標準)～+2(濃) "PSPrEngineCE.dll"使用時のみ有効で、他のドライバでは無視されます。
int nReserved1	将来の拡張のために予約された引数です。 現在は使用されていないため、0を指定します。

エラー

メッセージ	内容
引数に誤りがあります。	引数の数や型が間違っています。
LoadLibrary でエラーが発生しました。エラーコード[126]	プリンタ抽象化ドライバが見つかりません。
[抽象化ドライバ DLL 名]でエラー発生。 <LoadLibrary に失敗しました。エラーコード[126]>	プリンタドライバが見つかりません。
[抽象化ドライバ DLL 名]でエラー発生。<に失敗しました。 []> プリンタドライバで公開している関数名 エラー内容	プリンタドライバ内でエラーが発生しました。

5.3.2 PSFinishC

プリンタドライバをクローズし、印刷の終了処理を行います。

呼出形式 BOOL WINAPI PSFinishC(
 int nPrintInfo
)

戻り値 TRUE : 成功
 FALSE : 失敗

引数 **int nPrintInfo** PSInitC で取得したプリンタ情報値を指定します。

エラー	メッセージ	内容
	引数に誤りがあります。	引数の数や型が間違っています。

5.3.3 PSDrawFileC

帳票イメージファイルを解析し印刷を行います。
PSDrawMemC よりも少ない消費メモリで実行可能です。

本関数は、PSInitC から PSFinishC の間で何回でも呼び出せます。
また PSS を印刷する場合は、ファイルが連続する順番にすべてのファイルで、本関数を呼び出す必要があります。本制限以外の使用はサポート外となりますので、動作は不定となります。

呼出形式 BOOL WINAPI PSDrawFileC(
 int nPrintInfo,
 int nFileType,
 LPCTSTR pszFilePath
)

戻り値 TRUE : 成功
 FALSE : 失敗

引数 int nPrintInfo PSInitC で取得したプリンタ情報値を指定します。

 int nFileType ファイル形式を指定します。
 1 : SVG
 2 : PSS

 LPCTSTR
 pszFilePath 帳票イメージファイルの絶対パスを指定します。

エラー	メッセージ	内容
	引数に誤りがあります。	引数の数や型が間違っています。
	PSS が不正です。	PSS ファイルが壊れています。
	サポート外の PSS です。	PSS のバージョンがサポートしているバージョンより古いです。
	SVG が不正です。	SVG または PSS 内の SVG 部が壊れています。
	イメージファイルが見つかりません。	SVG 内に記述されているイメージファイルが見つかりません。
	メモリが不足しています。	帳票描画するためのメモリが足りません。
	[抽象化ドライバ DLL 名]でエラー発生。 < に失敗しました。 []> プリンタドライバで公開している関数名 エラー内容	プリンタドライバ内でエラーが発生しました。

5.3.4 PSDrawMemC

メモリ上の帳票イメージデータを解析し印刷を行います。
 帳票ファイルを作成しないで済むため PSDrawFileC よりセキュリティレベルは高くなりますが、メモリをより消費します。

本関数は、PSInitC から PSFinishC の間で何回でも呼び出せます。
 また PSS を印刷する場合は、ファイルが連続する順番にすべてのファイルで、本関数を呼び出す必要があります。本制限以外の使用はサポート外となりますので、動作は不定となります。

呼出形式 BOOL WINAPI PSDrawMemC(
 int nPrintInfo,
 int nDataType,
 char* pData,
 int nSize
)

戻り値 TRUE : 成功
 FALSE : 失敗

引数 int nPrintInfo, PSInit で取得したプリンタ情報値を指定します。

 int nDataType データ形式を指定します。
 1 : SVG
 2 : PSS

 char* pData 帳票イメージデータへのポインタを指定します。

 int nSize 帳票イメージデータのサイズをバイトで単位で指定します。

エラー	メッセージ	内容
	引数に誤りがあります。	引数の数や型が間違っています。
	PSS が不正です。	PSS ファイルが壊れています。
	サポート外の PSS です。	PSS のバージョンがサポートしているバージョンより古いです。
	SVG が不正です。	SVG または PSS 内の SVG 部が壊れています。
	イメージファイルが見つかりません。	SVG 内に記述されているイメージファイルが見つかりません。
	メモリが不足しています。	帳票描画するためのメモリが足りません。

[抽象化ドライバ DLL 名]でエラー発生。 < に失敗しました。 []> プリンタドライバで 公開している関数名 エラー内容	プリンタドライバ内でエラーが発生しました。
---	-----------------------

5.3.5 PSGetAPIVersionC

API の仕様バージョンを取得します。

呼出形式 int WINAPI PSGetAPIVersionC()

戻り値 API バージョン番号

引数 なし

エラー なし

5.3.6 PSetMarginC

プリンタのマージンを指定します。

PrintStream® for Mobile は通常原点座標を用紙の左上端として座標計算を行います。プリンタのハードウェアにより印刷の原点座標が用紙の左上端ではない場合、マージンを指定することによりこの誤差を補正することができます。

呼出形式 BOOL WINAPI PSetMarginC(
 int nPrintInfo,
 int nWidth,
 int nHeight
)

戻り値 TRUE : 成功
 FALSE : 失敗

引数 int nPrintInfo PSInitC で取得したプリンタ情報値を指定します。

 int nWidth 横マージンを mm 単位で指定します。

 int nHeight 縦マージンを mm 単位で指定します。

エラー	メッセージ	内容
	引数に誤りがあります。	引数の数や型が間違っています。
	[抽象化ドライバ DLL 名]でエラー発生。 < に失敗しました。 []> プリンタドライバで公開している関数名 エラー内容	プリンタドライバ内でエラーが発生しました。

5.3.8 PSGetDriverAPIVersionC

プリンタ抽象化ドライバの API の仕様バージョンを取得します

PSInit で指定するパラメータはドライバ固有になるため、このバージョンを確認することで抽象化ドライバのパラメータに変更があった場合でも処理を切り分けることが可能です。

呼出形式 int WINAPI PSGetDriverAPIVersionC(
 LPCTSTR **pszDriverDLL**,
)

戻り値 API バージョン(抽象化ドライバがこの関数に対応していない場合 0)

引数 LPCTSTR **pszDriverDLL** プリンタ抽象化ドライバを指定します。
 通常標準添付のドライバのうち、使用するプリン
 タに合わせて"PSPrintCE.dll"か
 "PSPrEngineCE.dll"を指定します。

エラー	メッセージ	内容
	引数に誤りがあります。	引数の数や型が間違っています。
	[抽象化ドライバDLL名]で エラー発生。< に失敗しま した。 []> プリンタドライバで公開 している関数名 エラー内容	プリンタドライバ内でエラーが発生しました。

5.3.9 PSGetErrorMessageC

エラー発生時の例外メッセージを取得します。

pszMessage に NULL を指定して呼び出すとメッセージを受け取るためのバッファの大きさを返します。

呼出形式 int WINAPI PSGetErrorMessageC (
 LPTSTR pszMessage,
 int nSize
)

戻り値 pszMessage が NULL の場合、確保すべきバッファのサイズ
0 : 失敗
0 以外 : 成功

引数 LPTSTR pszMessage メッセージを受け取るバッファのポインタを指定します。

NULL を指定して呼び出すとメッセージを受け取るためのバッファのサイズを返します。

int nSize バッファのサイズを指定します。

エラー なし

5.4 PSPreview.dll リファレンス

5.4.1 PSPreviewFileC

帳票イメージファイルを解析しプレビューを行います。PSPreviewMemC よりも少ない消費メモリで実行可能です。

プレビュー終了後まで関数から処理は戻りません。

本関数は、4つまでのファイルパスを指定できます。PSSをプレビューする場合は、ファイルが連続する順番にすべてのファイルを指定する必要があります。

ファイルが4分割より多くならないようサーバ側で調整する必要があります。

本制限以外の使用は、サポート外となりますので、動作は不定となります。

モノクロモードでは、プレビュー画面がモノクロ表示となります。メモリの消費を抑える事ができる為、メモリが不足している環境でのプレビューに有効です。

呼出形式 BOOL WINAPI PSPreviewFileC(
 int nFileType,
 int nMonoMode,
 LPCTSTR pszFilePath1,
 LPCTSTR pszFilePath2,
 LPCTSTR pszFilePath3,
 LPCTSTR pszFilePath4
)

戻り値 TRUE : 成功
 FALSE : 失敗

引数

int nFileType	ファイル形式を指定します。 1 : SVG 2 : PSS
int nMonoMode	モノクロモードにするかどうかを指定します。 0 : カラー 1 : モノクロ
LPCTSTR pszFilePath1 ~ 4	帳票イメージファイルの絶対パスを指定します。 PSS が複数に分割されている場合は、すべてを指定します。 使用しない引数は NULL を指定してください。

エラー	メッセージ	内容
	引数に誤りがあります。	引数の数や型が間違っています。
	LoadLibrary でエラーが発生しました。エラーコード[126]	PrintStream.dll が見つかりません。

PrintStream.dll でエラー発生。<PSS が不正です。>	PSS ファイルが壊れています。
PrintStream.dll でエラー発生。<サポート外の PSS です。>	PSS のバージョンがサポートしているバージョンより古いです。
PrintStream.dll でエラー発生。<SVG が不正です。>	SVG または PSS 内の SVG 部が壊れています。
PrintStream.dll でエラー発生。<イメージファイルが見つかりません。>	SVG 内に記述されているイメージファイルが見つかりません。
PrintStream.dll でエラー発生。<メモリが不足しています。>	帳票描画するためのメモリが足りません。

5.4.2 PSPreviewMemC

メモリ上の帳票イメージデータを解析しプレビューを行います。
帳票ファイルを作成しないで済むため PSPreviewFileC よりセキュリティレベルは高くなりますが、メモリをより消費します。

プレビュー終了後まで関数から処理は戻りません。

本関数は、4 つまでのデータを指定できます。PSS をプレビューする場合は、データが連続する順番にすべてのデータを指定する必要があります。

データが4分割より多くなる場合、バッファを用意し結合処理を行う必要があります。

本制限以外の使用は、サポート外となりますので、動作は不定となります。

モノクロモードでは、プレビュー画面がモノクロ表示となります。メモリの消費を抑える事ができる為、メモリが不足している環境でのプレビューに有効です。

呼出形式 BOOL WINAPI PSPreviewMemC(
 int nDataType,
 int nMonoMode,
 char* pData1,
 int nSize1,
 char* pData2,
 int nSize2,
 char* pData3,
 int nSize3,
 char* pData4,
 int nSize4
)

戻り値 TRUE : 成功
 FALSE : 失敗

引数	int nDataType	データ形式を指定します。 1 : SVG 2 : PSS
	int nMonoMode	モノクロモードにするかどうかを指定します。 0 : カラー 1 : モノクロ
	char* pData1 ~ 4	帳票イメージデータへのポインタを指定します。 使用しない引数は NULL を指定してください。
	int nSize1 ~ 4	帳票イメージデータのサイズをバイト単位で指定します。 使用しない引数は 0 を指定してください。

エラー

メッセージ	内容
引数に誤りがあります。	引数の数や型が間違っています。
LoadLibrary でエラーが発生しました。エラーコード[126]	PrintStream.dll が見つかりません。
PrintStream.dll でエラー発生。<PSS が不正です。>	PSS ファイルが壊れています。
PrintStream.dll でエラー発生。<サポート外の PSS です。>	PSS のバージョンがサポートしているバージョンより古いです。
PrintStream.dll でエラー発生。<SVG が不正です。>	SVG または PSS 内の SVG 部が壊れています。
PrintStream.dll でエラー発生。<イメージファイルが見つかりません。>	SVG 内に記述されているイメージファイルが見つかりません。
PrintStream.dll でエラー発生。<メモリが不足しています。>	帳票描画するためのメモリが足りません。

5.4.3 PSPGetAPIVersionC

API の仕様バージョンを取得します。

呼出形式 int WINAPI PSPGetAPIVersionC()

戻り値 API バージョン番号

引数 なし

エラー なし

5.4.4 PSPGetErrorMessageC

エラー発生時の例外メッセージを取得します。

pszMessage に NULL を指定して呼び出すとメッセージを受け取るためのバッファの大きさを返します。

呼出形式 int WINAPI PSPGetErrorMessageC (
 LPTSTR pszMessage,
 int nSize
)

戻り値 pszMessage が NULL の場合、確保すべきバッファのサイズ
0 : 失敗
0 以外 : 成功

引数 LPTSTR pszMessage メッセージを受け取るバッファのポインタを指定します。

NULL を指定して呼び出すとメッセージを受け取るためのバッファのサイズを返します。

int nSize バッファのサイズを指定します。

エラー なし

6. .NET アプリケーションからの呼び出し

この章では.NETアプリケーションからPrintStream.dllおよびPSPreview.dllを利用する方法について解説します。

6.1 DllImport を利用する方法

.NETからPrintStream.dll、PSPreview.dllを使用する場合、DllImport (System.Runtime.InteropServices名前空間)を使用し、C/C++用の関数を呼び出します。

DLLのハンドルを取得することはできませんが、通常の間数のように呼び出すことが可能です。また、DLLの配置場所を相対アドレスまたは絶対アドレスで指定することが可能です。呼び出し例としては以下のようになります。

```
//DllImportした関数を管理するクラス
static class PrintStreamDLL
{
    //PSGetDriverAPIVersionCのDllImport
    [DllImport("PrintStream.dll")]
    extern public static int PSGetDriverAPIVersionC(
        string pszDriverDLL
    );
}

//---呼び出し例---
int version = PrintStreamDLL.PSGetDriverAPIVersionC("PSPrintCE.dll");
```

6.2 PrintStream.dll リファレンス

6.2.1 PSInitC

プリンタドライバを初期化し、印刷情報を取得します。以降、取得した印刷情報を用い印刷を実行します。

本関数を実行した後は必ずPSFinishCによりプリンタドライバの終了処理を行ってください。

呼出形式 [DllImport("PrintStream.dll")]
extern public static int PSInitC(
 string strDriverDLL,
 int nPortType,
 string strPortName,
 int nPrinter,
 int nPaperSize,
 int nOrientation,
 int nColor,
 int nSilentMode,
 int nDraftMode,
 int nContrast,
 int nBrightness,
 int nDensity,
 int nReserved1
);

戻り値 0以外：印刷情報(今後の印刷APIに必要)
0：エラー

引数 string strDriverDLL プリンタ抽象化ドライバを指定します。
通常標準添付のドライバのうち、使用するプリンタに合わせて” PSPrintCE.dll”か”PSPrintCE.dll”を指定します。

int nPortType 出力ポートタイプを指定します。
1：COMポート
3：IrDA

string strPortName ポート種別が1の時のみ有効です。ポート名+”:”で指定します。
例)”COM4:”

int nPrinter 印刷対象プリンタを指定します。指定可能な値はドライバにより異なります。
詳細は[対応プリンター一覧](#)に記載されています。

int nPaperSize 用紙サイズを指定します。
プリンタでサポートされないサイズを指定した場合、動作はプリンタに依存します。
ただし、[一部機種](#)では縦方向サイズ[mm]を指定します。

	0 : A4(デフォルト) 1 : A5 2 : B5 3 : レター 4 : リーガル
int nOrientation	印刷の向きを指定します。 0 : 横 1 : 縦(デフォルト)
int nColor	印刷時の色モードを指定します。 0 : モノクロ 1 : 4色カラー(デフォルト) 2 : 3色カラー 0=モノクロを指定した場合、PrintStream® for Mobileはカラーをディザパターンに置き換え描画します。
int nSilentMode	印刷進捗ダイアログを表示するかどうかを指定します。 0 : 表示する(デフォルト) 1 : 表示しない
int nDraftMode	ドラフト印刷モードを指定します。テスト印字などインク消費を抑えたい場合に使用します。 0 : オフ(デフォルト) 1 : オン
int nContrast	画像のコントラストを調整します。線や塗りつぶしには影響しません。 0(弱) ~ 45(デフォルト) ~ 90(強)
int nBrightness	画像の明るさを調整します。線や塗りつぶしには影響しません。 -127(暗) ~ 0(デフォルト) ~ 127(明)
int nDensity	印刷の濃度を調整します。 -2(薄) ~ 0(標準) ~ +2(濃) "PSPrEngineCE.dll"使用時のみ有効で、他のドライバでは無視されます。
int nReserved1	将来の拡張のために予約された引数です。 現在は使用されていないため、0を指定します。

エラー

メッセージ	内容
引数に誤りがあります。	引数の数や型が間違っています。
LoadLibraryでエラーが発生しました。エラーコード[126]	プリンタ抽象化ドライバが見つかりません。
[抽象化ドライバDLL名]でエラー発生。 <LoadLibraryに失敗しました。エラーコード[126]>	プリンタドライバが見つかりません。
[抽象化ドライバDLL名]でエラー発生。<に失敗しました。 []> プリンタドライバで公開している関数名 エラー内容	プリンタドライバ内でエラーが発生しました。

6.2.2 PSFinishC

プリンタドライバをクローズし、印刷の終了処理を行います。

呼出形式 [DllImport("PrintStream.dll")]
extern public static bool PSFinishC(
 int nPrintInfo
);

戻り値 true : 成功
false : 失敗

引数 int nPrintInfo PSInitCで取得したプリンタ情報値を指定します。

エラー 引数に誤りがありま 引数の数や型が間違っています。
す。

6.2.3 PSDrawFileC

帳票イメージファイルを解析し印刷を行います。
PSDrawMemCよりも少ない消費メモリで実行可能です。

本関数は、PSInitCからPSFinishCの間で何回でも呼び出せます。
またPSSを印刷する場合は、ファイルが連続する順番にすべてのファイルで、本関数を呼び出す必要があります。本制限以外の使用はサポート外となりますので、動作は不定となります。

呼出形式 [DllImport("PrintStream.dll")]
extern public static bool PSDrawFileC(
 int nPrintInfo,
 int nFileType,
 string strFilePath
);

戻り値 true : 成功
false : 失敗

引数 int nPrintInfo PSInitCで取得したプリンタ情報値を指定します。

int nFileType ファイル形式を指定します。
1 : SVG
2 : PSS

string strFilePath 帳票イメージファイルの絶対パスを指定します。

エラー	メッセージ	内容
	引数に誤りがあります。	引数の数や型が間違っています。
	PSSが不正です。	PSSファイルが壊れています。
	サポート外のPSSです。	PSSのバージョンがサポートしているバージョンより古いです。
	SVGが不正です。	SVGまたはPSS内のSVG部が壊れています。
	イメージファイルが見つかりません。	SVG内に記述されているイメージファイルが見つかりません。
	メモリが不足しています。	帳票描画するためのメモリが足りません。
	[抽象化ドライバDLL名]でエラー発生。 < に失敗しました。 []> プリンタドライバで公開している関数名 エラー内容	プリンタドライバ内でエラーが発生しました。

6.2.4 PSDrawMemC

メモリ上の帳票イメージデータを解析し印刷を行います。
帳票ファイルを作成しないで済むためPSDrawFileCよりセキュリティレベルは高くなりますが、メモリをより消費します。

本関数は、PSInitCからPSFinishCの間で何回でも呼び出せます。
またPSSを印刷する場合は、ファイルが連続する順番にすべてのファイルで、本関数を呼び出す必要があります。本制限以外の使用はサポート外となりますので、動作は不定となります。

呼出形式 [DllImport("PrintStream.dll")]
extern public static bool PSDrawMemC(
 int nPrintInfo,
 int nFileType,
 byte[] pData,
 int nSize
);

戻り値 true : 成功
false : 失敗

引数 int nPrintInfo, PSInitで取得したプリンタ情報値を指定します。

int nDataType データ形式を指定します。
 1 : SVG
 2 : PSS

byte[] pData 帳票イメージデータへの配列を指定します。

int nSize 帳票イメージデータのサイズをバイトで単位で指定します。

エラー	メッセージ	内容
	引数に誤りがあります。	引数の数や型が間違っています。
	PSSが不正です。	PSSファイルが壊れています。
	サポート外のPSSです。	PSSのバージョンがサポートしているバージョンより古いです。
	SVGが不正です。	SVGまたはPSS内のSVG部が壊れています。
	イメージファイルが見つかりません。	SVG内に記述されているイメージファイルが見つかりません。
	メモリが不足しています。	帳票描画するためのメモリが足りません。
	[抽象化ドライバDLL名]でエラー発生。 < に失敗しました。 []> プリンタドライバで公開している関数名 エラー内容	プリンタドライバ内でエラーが発生しました。

6.2.5 PSGetAPIVersionC

APIの仕様バージョンを取得します。

呼出形式 [DllImport("PrintStream.dll")]
extern public static int PSGetAPIVersionC();

戻り値 APIバージョン番号

引数 なし

エラー なし

6.2.6 PSetMarginC

プリンタのマージンを指定します。

PrintStream® for Mobileは通常原点座標を用紙の左上端として座標計算を行います。プリンタのハードウェアにより印刷の原点座標が用紙の左上端ではない場合、マージンを指定することによりこの誤差を補正することができます。

呼出形式 [DllImport("PrintStream.dll")]
extern public static bool PSetMarginC(
 int nPrintInfo,
 int nWidth,
 int nHeight
);

戻り値 true : 成功
false : 失敗

引数 int nPrintInfo PSInitCで取得したプリンタ情報値を指定します。

int nWidth 横マージンをmm単位で指定します。

int nHeight 縦マージンをmm単位で指定します。

エラー	メッセージ	内容
	引数に誤りがあります。	引数の数や型が間違っています。
	[抽象化ドライバDLL名]でエラー発生。< に失敗しました。[]> プリンタドライバで公開している関数名 エラー内容	プリンタドライバ内でエラーが発生しました。

6.2.7 PSetScalingC

印刷の拡大縮小率を指定します。

拡大の場合、端末のメモリ残量により印刷に失敗する場合があります。

呼出形式 [DllImport("PrintStream.dll")]
extern public static bool PSetScalingC(
 int nPrintInfo,
 int nScale,
);

戻り値 true : 成功
false : 失敗

引数 int nPrintInfo PSInitCで取得したプリンタ情報値を指定します。

int nScale 拡大縮小率を[%]で指定します。
50(最小) ~ 100(デフォルト) ~ 200(最大)

エラー	メッセージ	内容
	引数に誤りがあります。	引数の数や型が間違っています。
	[抽象化ドライバDLL名]でエラー発生。<に失敗しました。[]> プリンタドライバで公開している関数名 エラー内容	プリンタドライバ内でエラーが発生しました。

6.2.8 PSGetDriverAPIVersionC

プリンタ抽象化ドライバのAPIの仕様バージョンを取得します

PSInitで指定するパラメータはドライバ固有になるため、このバージョンを確認することで抽象化ドライバのパラメータに変更があった場合でも処理を切り分けることが可能です。

呼出形式 [DllImport("PrintStream.dll")]
extern public static int PSGetDriverAPIVersionC(
 string strDriverDLL
);

戻り値 APIバージョン(抽象化ドライバがこの関数に対応していない場合0)

引数 string strDriverDLL プリンタ抽象化ドライバを指定します。
 通常標準添付のドライバ” PSPrintCE.dll”を指定
 します。

エラー	メッセージ	内容
	引数に誤りがあります。	引数の数や型が間違っています。
	[抽象化ドライバDLL名]で エラー発生。< に失敗しま した。 []> プリンタドライバで公開 している関数名 エラー内容	プリンタドライバ内でエラーが発生しました。

6.2.9 PSGetErrorMessageC

エラー発生時の例外メッセージを取得します。

pszMessageにNULLを指定して呼び出すとメッセージを受け取るためのバッファの大きさを返します。

呼出形式 [DllImport("PrintStream.dll")]
extern public static int PSGetErrorMessageC(
 StringBuilder **strMessage**,
 int **nSize**
);

戻り値 strMessageがnullの場合、確保すべきバッファのサイズ
0 : 失敗
0以外 : 成功

引数 StringBuilder メッセージを受け取るStringBuilderオブジェクトを
 strMessage 指定します。

 nullを指定して呼び出すとメッセージを受け取るための
 バッファのサイズを返します。

 int **nSize** バッファのサイズを指定します。

エラー なし

6.3 PSPreview.dll リファレンス

6.3.1 PSPreviewFileC

帳票イメージファイルを解析しプレビューを行います。PSPreviewMemCよりも少ない消費メモリで実行可能です。

プレビュー終了後まで関数から処理は戻りません。

本関数は、4つまでのファイルパスを指定できます。PSSをプレビューする場合は、ファイルが連続する順番にすべてのファイルを指定する必要があります。

ファイルが4分割より多くならないようサーバ側で調整する必要があります。

本制限以外の使用は、サポート外となりますので、動作は不定となります。

モノクロモードでは、プレビュー画面がモノクロ表示となります。メモリの消費を抑える事ができる為、メモリが不足している環境でのプレビューに有効です。

呼出形式 [DllImport("PSPreview.dll")]
extern public static bool PSPreviewFileC(
 int nFileType,
 int nMonoMode,
 string strFilePath1,
 string strFilePath2,
 string strFilePath3,
 string strFilePath4
);

戻り値 true : 成功
false : 失敗

引数

int nFileType	ファイル形式を指定します。 1 : SVG 2 : PSS
int nMonoMode	モノクロモードにするかどうかを指定します。 0 : カラー 1 : モノクロ
string strFilePath1 ~ 4	帳票イメージファイルの絶対パスを指定します。 PSSが複数に分割されている場合は、すべてを指定します。 使用しない引数はnullを指定してください。

エラー	メッセージ	内容
	引数に誤りがあります。	引数の数や型が間違っています。
	LoadLibraryでエラーが発生しました。エラーコード[126]	PrintStream.dllが見つかりません。

PrintStream.dll で エラー発生。<PSSが不正です。>	PSSファイルが壊れています。
PrintStream.dll で エラー発生。<サポート外のPSSです。>	PSSのバージョンがサポートしているバージョンより古いです。
PrintStream.dll で エラー発生。<SVGが不正です。>	SVGまたはPSS内のSVG部が壊れています。
PrintStream.dll で エラー発生。<イメージファイルが見つかりません。>	SVG内に記述されているイメージファイルが見つかりません。
PrintStream.dll で エラー発生。<メモリが不足しています。>	帳票描画するためのメモリが足りません。

6.3.2 PSPreviewMemC

メモリ上の帳票イメージデータを解析しプレビューを行います。
帳票ファイルを作成しないで済むためPSPreviewFileCよりセキュリティレベルは高くなりますが、メモリをより消費します。

プレビュー終了後まで関数から処理は戻りません。

本関数は、4つまでのデータを指定できます。PSSをプレビューする場合は、データが連続する順番にすべてのデータを指定する必要があります。

データが4分割より多くなる場合、バッファを用意し結合処理を行う必要があります。

本制限以外の使用は、サポート外となりますので、動作は不定となります。

モノクロモードでは、プレビュー画面がモノクロ表示となります。メモリの消費を抑える事ができる為、メモリが不足している環境でのプレビューに有効です。

呼出形式 [DllImport("PSPreview.dll")]
extern public static bool PSPreviewMemC(int nDataType,
int nMonoMode,
byte[] pData1,
int nSize1,
byte[] pData2,
int nSize2,
byte[] pData3,
int nSize3,
byte[] pData4,
int nSize4
);

戻り値 true : 成功
false : 失敗

引数	int nDataType	データ形式を指定します。 1 : SVG 2 : PSS
	int nMonoMode	モノクロモードにするかどうかを指定します。 0 : カラー 1 : モノクロ
	byte[] pData1 ~ 4	帳票イメージデータへのポインタを指定します。 使用しない引数はnullを指定してください。
	int nSize1 ~ 4	帳票イメージデータのサイズをバイト単位で指定します。 使用しない引数は0を指定してください。

エラー

メッセージ	内容
引数に誤りがあります。	引数の数や型が間違っています。
LoadLibraryでエラーが発生しました。エラーコード[126]	PrintStream.dllが見つかりません。
PrintStream.dll でエラー発生。<PSSが不正です。>	PSSファイルが壊れています。
PrintStream.dll でエラー発生。<サポート外のPSSです。>	PSSのバージョンがサポートしているバージョンより古いです。
PrintStream.dll でエラー発生。<SVGが不正です。>	SVGまたはPSS内のSVG部が壊れています。
PrintStream.dll でエラー発生。<イメージファイルが見つかりません。>	SVG内に記述されているイメージファイルが見つかりません。
PrintStream.dll でエラー発生。<メモリが不足しています。>	帳票描画するためのメモリが足りません。

6.3.3 PSPGetAPIVersionC

APIの仕様バージョンを取得します。

呼出形式 [DllImport("PSPreview.dll")]
extern public static int PSPGetAPIVersionC();

戻り値 APIバージョン番号

引数 なし

エラー なし

6.3.4 PSPGetErrorMessageC

エラー発生時の例外メッセージを取得します。

pszMessageにnullを指定して呼び出すとメッセージを受け取るためのバッファの大きさを返します。

呼出形式 [DllImport("PSPreview.dll")]
extern public static int PSPGetErrorMessageC(
 StringBuilder **strMessage**,
 int **nSize**
);

戻り値 strMessageがnullの場合、確保すべきバッファのサイズ
0 : 失敗
0以外 : 成功

引数 StringBuilder メッセージを受け取るStringBuilderオブジェクトを
 strMessage 指定します。

 nullを指定して呼び出すとメッセージを受け取るため
 のバッファのサイズを返します。

 int **nSize** バッファのサイズを指定します。

エラー なし

7. 対応プリンター一覧

PSInit メソッド(C++/C#では PSInitC メソッド)に指定する抽象ドライバ DLL により、選択可能なプリンターが変化します。

以下に選択可能なプリンター一覧を示します。

抽象ドライバ名	プリンタ種別	プリンタ番号	用紙タイプの指定	用紙方向の指定
PSPrintCE.dll	HP インクジェット	0		
	エプソン インクジェット	2		
	Brother PJ-520/560	3		
	Canon インクジェット	5		
	Canon iP90	11		
	Canon iP100	14		
PSPrEngineCE.dll	Brother PJ-520/560	7		
	SII DPU-3445/S445	8	(1)	× (2)
	SII DPU-S245	1008	(1)	× (2)
	Brother MW1x0 シリーズ	21		
	Brother MW260	1021		
	SII MPU-L465	28	(1)	× (2)
	SANEI BLM-80	25	(1)	× (2)

- 1 用紙タイプではなく、縦サイズ[mm]の指定となります。
0 を指定した場合、帳票データに記載された縦サイズが設定されます。
- 2 引数の値に関係なく、常に縦方向固定となります。

8. PSPreview.dll プレビュー画面操作方法

8.1 ポップアップメニューを出す方法

- タッチパネル操作
- ・ タップアンドホールド
- キー操作
- ・ 決定キー長押し

8.2 スクロール

- タッチパネル操作
- ・ スクロールバー
- ・ ドラッグ
- キー操作
- ・ 方向キー

8.3 表示倍率変更

(倍率は 200%/150%/100%/75%/50%/25%が選択できる。最初は 100%)

- ポップアップメニュー
- ・ [倍率]メニューから [200%][150%][100%][75%][50%][25%]を選択
- キー操作
- ・ 決定キーを押し、画面左下に「 : 倍率 : ページ移動」とガイドが出たら、
ボタンで倍率を 1 段階アップ(200%表示時には無視される)
ボタンで倍率を 1 段階ダウン(25%表示時には無視される)

8.4 ページ移動

- ポップアップメニュー
- ・ [ページ]メニューから [次のページ][前のページ][最後のページ][最初のページ]を選択
- キー操作
- ・ 決定キーを押し、画面左下に「 : 倍率 : ページ移動」とガイドが出たら、
ボタンで次のページに移動(最終ページ表示時には無視される)
ボタンで前のページに移動(1 ページ目表示時には無視される)

8.5 プレビューの終了

- ポップアップメニュー
- ・ [閉じる]メニューを選択
- タッチパネル操作
- ・ ok ボタン押下

9. コーディング例

9.1 Biz/Browser Mobile からの呼び出し

9.1.1 SVG ファイルを COM4 ポートにマッピングした Bluetooth 経由で印刷する場合

```
Function Print( svgfilepath ) {
  try {
    var dll = new Runtime( "PrintStream.dll" );
    var ps = dll.CallDll( "PSInit", "PSPrintCE.dll", 1, "COM4:" );
    try {
      dll.CallDll( "PSDrawFile", ps, 1, svgfilepath );
    }
    catch ( e ) {
      dll.CallDll( "PSFinish", ps );
      throw e;
    }
    dll.CallDll( "PSFinish", ps );
  }
  catch ( e ) {
    //.MessageBox( e.message, "PrintStream for Mobile", $OK + $IconStop );
  }
}
```

9.1.2 メモリ上の PSS データを IrDA で印刷する場合

```
Function Print( pss1, pss2 ) {
  try {
    var dll = new Runtime( "PrintStream.dll" );
    var ps = dll.CallDll( "PSInit", "PSPrintCE.dll", 3 );
    try {
      dll.CallDll( "PSDrawMem", ps, 4, Base64Encoder.Convert( pss1 ) );
      dll.CallDll( "PSDrawMem", ps, 4, Base64Encoder.Convert( pss2 ) );
    }
    catch ( e ) {
      dll.CallDll( "PSFinish", ps );
      throw e;
    }
    dll.CallDll( "PSFinish", ps );
  }
  catch ( e ) {
    //.MessageBox( e.message, "PrintStream for Mobile", $OK + $IconStop );
  }
}
```

9.1.3 SVG ファイルをカラーでプレビューする場合

```
Function Preview( svgfilepath ) {
  try {
    var dll = new Runtime( "PSPreview.dll" );
    var ps = dll.CallDll( "PSPreviewFile", 1, 0, svgfilepath );
  }
  catch ( e ) {
    //.MessageBox( e.message, "PrintStream for Mobile", $OK + $IconStop );
  }
}
```

9.1.4 メモリ上の PSS データをモノクロでプレビューする場合

```
Function Preview( pss1, pss2 ) {
  try {
    var dll = new Runtime( "PSPreview.dll" );
    dll.CallDll( "PSPreviewMem", 4, 1,
      Base64Encoder.Convert( pss1 ),
      Base64Encoder.Convert( pss2 ) );
  }
  catch ( e ) {
    //.MessageBox( e.message, "PrintStream for Mobile", $OK + $IconStop );
  }
}
```

9.2 VC++からの呼び出し

9.2.1 SVG ファイルを COM4 ポートにマッピングした Bluetooth 経由で印刷する場合

```
#include "PrintStream.h"

void PutError()
{
    int nSize = PSGetErrorMessageC(NULL, 0);
    LPTSTR buf = new TCHAR[nSize];
    PSGetErrorMessageC(buf, nSize);
    MessageBox(NULL, buf, L"Error", MB_OK);
    delete [] buf;
}

void Print(LPCTSTR IpszSvgFilePath)
{
    // 印刷初期化
    int nPrintInfo = PSInitC(L"PSPrintCE.dll", 1, L"COM4:", 0, 0, 1, 2, 0, 0, 45, 0, 0, 0);
    if (nPrintInfo == 0) {
        PutError();
        return;
    }

    // 印刷実行
    if (!PSDrawFileC(nPrintInfo, 1, IpszSvgFilePath)) {
        PutError();
        PSFinish(nPrintInfo);
        return;
    }

    // 印刷終了
    if (!PSFinish(nPrintInfo)) {
        PutError();
    }
}
```

9.2.2 メモリ上の PSS データを IrDA で印刷する場合

```
#include "PrintStream.h"

void PutError()
{
    int nSize = PSGetErrorMessageC(NULL, 0);
    LPTSTR buf = new TCHAR[nSize];
    PSGetErrorMessageC(buf, nSize);
    MessageBox(NULL, buf, L"Error", MB_OK);
    delete [] buf;
}

void Print(char* lpPss1, int nPssSize1, char* lpPss2, int nPssSize2)
{
    // 印刷初期化
    int nPrintInfo = PSInitC(L"PSPrintCE.dll", 2, NULL, 0, 0, 1, 2, 0, 0, 45, 0, 0, 0);
    if (nPrintInfo == 0) {
        PutError();
        return;
    }

    /* 印刷実行(PSSデータ1) */
    if ( !PSDrawMemC(nPrintInfo, 2, lpPss1, nPssSize1) ) {
        PutError();
        PSFinishC(nPrintInfo);
        return;
    }
    /* 印刷実行(PSSデータ2) */
    if ( !PSDrawMemC(nPrintInfo, 2, lpPss2, nPssSize2) ) {
        PutError();
        PSFinishC(nPrintInfo);
        return;
    }
    /* 印刷終了 */
    if ( !PSFinishC(nPrintInfo) ) {
        PutError();
    }
    return;
}
```

9.2.3 SVG ファイルをカラーでプレビューする場合

```
#include "PSPreview.h"

void PutError()
{
    int nSize = PSPGetErrorMessageC(NULL, 0);
    LPTSTR buf = new TCHAR[nSize];
    PSPGetErrorMessageC(buf, nSize);
    MessageBox(NULL, buf, L"Error", MB_OK);
    delete [] buf;
}

void Preview( LPCTSTR IpszSvgFilePath )
{
    if( !PSPreviewFileC(1, 0, IpszSvgFilePath, NULL, NULL, NULL) ){
        PutError();
    }
}
```

9.2.4 メモリ上の PSS データをモノクロでプレビューする場合

```
#include "PSPreview.h"

void PutError()
{
    int nSize = PSPGetErrorMessageC(NULL, 0);
    LPTSTR buf = new TCHAR[nSize];
    PSPGetErrorMessageC(buf, nSize);
    MessageBox(NULL, buf, L"Error", MB_OK);
    delete [] buf;
}

void Preview(char* Ipss1, int nPssSize1, char* Ipss2, int nPssSize2)
{
    if ( !PSPreviewMemC(2, 1, Ipss1, nPssSize1, Ipss2, nPssSize1,
        NULL, 0, NULL, 0) ) {
        PutError();
    }
    return;
}
```

9.3 VC#からの呼び出し

9.3.1 SVG ファイルを COM4 ポートにマッピングした Bluetooth 経由で印刷する場合

```
using System.Text;
using System.Runtime.InteropServices;
using System.Windows.Forms;

static class SVGPrint
{
    [DllImport("PrintStream.dll", CharSet = CharSet.Unicode)]
    extern private static int PSInitC(
        string pszDriverDLL, int nPortType, string pszPortName,
        int nPrinter, int nPaperSize, int nOrientation,
        int nColor, int nSilentMode, int nDraftMode,
        int nContrast, int nBrightness);

    [DllImport("PrintStream.dll", CharSet = CharSet.Unicode)]
    extern private static bool PSFinishC(int nPrintInfo);

    [DllImport("PrintStream.dll", CharSet = CharSet.Unicode)]
    extern private static bool PSDrawFileC(
        int nPrintInfo, int nFileType, string pszFilePath);

    [DllImport("PrintStream.dll", CharSet = CharSet.Unicode)]
    extern private static int PSGetErrorMessageC(
        StringBuilder pszMessage, int nSize);

    private static void PutError()
    {
        int nSize = PSGetErrorMessageC(null, 0);
        StringBuilder buf = new StringBuilder(nSize + 1);
        PSGetErrorMessageC(buf, buf.Capacity);
        MessageBox.Show(buf.ToString(), "Error");
    }

    public static void Print(string lpszSvgFilePath)
    {
        // 印刷初期化
        int nPrintInfo = PSInitC("PSPrintCE.dll", 1, "COM4:", 0, 0, 1, 2, 0, 0, 45, 0);
        if (nPrintInfo == 0) {
            PutError();
            return;
        }

        // 印刷実行
        if (!PSDrawFileC(nPrintInfo, 1, lpszSvgFilePath)) {
            PutError();
            PSFinishC(nPrintInfo);
            return;
        }

        // 印刷終了
        if (!PSFinishC(nPrintInfo))
        {
            PutError();
        }
    }
}
```

9.3.2 メモリ上の PSS データを IrDA で印刷する場合

```
using System.Text;
using System.Runtime.InteropServices;
using System.Windows.Forms;

static class IrDAPrint
{
    [DllImport("PrintStream.dll", CharSet = CharSet.Unicode)]
    extern private static int PSInitC(
        string pszDriverDLL, int nPortType, string pszPortName,
        int nPrinter, int nPaperSize, int nOrientation,
        int nColor, int nSilentMode, int nDraftMode,
        int nContrast, int nBrightness);

    [DllImport("PrintStream.dll", CharSet = CharSet.Unicode)]
    extern private static bool PSFinishC(int nPrintInfo);

    [DllImport("PrintStream.dll", CharSet = CharSet.Unicode)]
    extern private static bool PSDrawMemC(
        int nPrintInfo, int nFileType,
        byte[] pData, int nSize);

    [DllImport("PrintStream.dll", CharSet = CharSet.Unicode)]
    extern private static int PSGetErrorMessageC(
        StringBuilder pszMessage, int nSize);

    private static void PutError()
    {
        int nSize = PSGetErrorMessageC(null, 0);
        StringBuilder buf = new StringBuilder(nSize + 1);
        PSGetErrorMessageC(buf, nSize);
        MessageBox.Show(buf.ToString(), "Error");
    }

    public static void Print(byte[] lpPss1, byte[] lpPss2) {
        // 印刷初期化
        int nPrintInfo = PSInitC("PSPrintCE.dll", 2, null, 0, 0, 1, 2, 0, 0, 45, 0);
        if (nPrintInfo == 0) {
            PutError();
            return;
        }
        /* 印刷データ(1)印刷実行 */
        if (!PSDrawMemC(nPrintInfo, 2, lpPss1, lpPss1.Length)) {
            PutError();
            return;
        }
        /* 印刷データ(2)印刷実行 */
        if (!PSDrawMemC(nPrintInfo, 2, lpPss2, lpPss2.Length)) {
            PutError();
            return;
        }
        /* 印刷終了 */
        if (!PSFinishC(nPrintInfo)) {
            PutError();
            return;
        }
    }
}
```

9.3.3 SVG ファイルをカラーでプレビューする場合

```
using System.Text;
using System.Runtime.InteropServices;
using System.Windows.Forms;

static class SvgPreview
{
    [DllImport("PSPreview.dll", CharSet = CharSet.Unicode)]
    extern private static int PSPGetErrorMessageC(
        StringBuilder pszMessage,
        int nSize
    );
    [DllImport("PSPreview.dll", CharSet = CharSet.Unicode)]
    extern private static bool PSPreviewFileC(
        int nFileType, int nMonoMode,
        string pszFilePath1, string pszFilePath2,
        string pszFilePath3, string pszFilePath4);

    private static void PutError()
    {
        int nSize = PSPGetErrorMessageC(null, 0);
        StringBuilder buf = new StringBuilder(nSize + 1);
        PSPGetErrorMessageC(buf, nSize);
        MessageBox.Show(buf.ToString(), "Error");
    }

    public static void Preview(string lpszSvgFilePath)
    {
        /* プレビュー実行 */
        if (!PSPreviewFileC(1, 0, lpszSvgFilePath, null, null, null))
        {
            PutError();
            return;
        }
    }
}
```


9.3.4 メモリ上の PSS データをモノクロでプレビューする場合

```
using System.Text;
using System.Runtime.InteropServices;
using System.Windows.Forms;

static class PssPreview
{
    [DllImport("PSPreview.dll", CharSet = CharSet.Unicode)]
    extern private static int PSPGetErrorMessageC(
        StringBuilder pszMessage,
        int nSize
    );
    [DllImport("PSPreview.dll", CharSet = CharSet.Unicode)]
    extern private static bool PSPreviewMemC(
        int nDataType, int nMonoMode,
        byte[] pData1, int nSize1,
        byte[] pData2, int nSize2,
        byte[] pData3, int nSize3,
        byte[] pData4, int nSize4);

    private static void PutError()
    {
        int nSize = PSPGetErrorMessageC(null, 0);
        StringBuilder buf = new StringBuilder(nSize + 1);
        PSPGetErrorMessageC(buf, nSize);
        MessageBox.Show(buf.ToString(), "Error");
    }

    public static void Preview(byte[] lpPss1, byte[] lpPss2)
    {
        /* データサイズ取得 */
        int nPss1Size = (lpPss1 != null ? lpPss1.Length : 0);
        int nPss2Size = (lpPss2 != null ? lpPss2.Length : 0);

        /* プレビュー実行 */
        if (!PSPreviewMemC(2, 1, lpPss1, nPss1Size, lpPss2, nPss2Size, null, 0, null, 0))
        {
            PutError();
            return;
        }
    }
}
```

PrintStream for Mobile

利用ガイド・API リファレンス

2009年2月3日 Ver.1.0
2009年9月1日 Ver.2.0
2010年3月8日 Ver.2.1

発行: アクシスソフト株式会社
〒170 - 0013 東京都豊島区東池袋 3 - 23 - 5 ダヴィンチ東池袋ビル
