

CITIZEN

Windows Layout SDK via Printer Driver

プログラミング マニュアル

Version 1.4.0 用

シチズン・システムズ株式会社

更新履歴

年月日	バージョン	履歴
2009.06.01	1.0.0.0	・新規（Layout Utilities ユーザーズ・ガイド）
2009.11.19	1.0.1.0	・印刷中ダイアログを非表示へ変更。
2013.11.22	1.1.0.0	<ul style="list-style-type: none"> ・対応 OS を追加： Windows 7、Windows 8、Windows 8.1、Windows Server 2008、 Windows Server 2008R2、Windows Server 2012。 ・64 ビット ユーザー アプリケーション 作成に対応。 ・マルチスレッド ユーザー アプリケーション による印刷に対応。 ・対応バーコードを追加： MaxiCode、Data Matrix、GS1 Composite、GS1 Databar (RSS)、iQR。 ・イメージのドット バイ ドット印刷に対応。 および、プロパティ名称を [FixedAspect] から [SizeMode] へ変更。 ・初回印刷速度を改善するメソッドを追加： PreparePrint()、HidePreview()。
2014.09.08	1.2.0.0	・Windows Layout SDK の更新はありません
2016.08.16	1.3.0.0	<ul style="list-style-type: none"> ・対応 OS を追加: Windows10 ・.Net Framework 4.0 ベースへ変更。 ・「Layout Print Engine」から「Layout SDK」へ呼称を変更。 ・Layout SDK プログラマー向けにユーザーズ ガイドから分離。（本書）
2017.09.29	1.4.0.0	・「Windows Layout SDK」を「Windows Layout SDK via Printer Driver」へ呼称を変更。

ご注意

1. 本書の内容の一部、または全部を無断で転載することは、固くお断りいたします。
2. 本書の内容については、事前の予告なしに変更することがあります。
3. 本書の内容については万全を期して作成いたしましたが、万一誤り・お気付きの点がございましたら、ご連絡くださいますようお願いいたします。
4. 運用した結果の影響につきましては、3項にかかわらず責任を負いかねますのでご了承ください。
5. 上記に同意いただけない場合は、本ライブラリをご使用いただけません。

著作権・商標について

- このプログラミング マニュアルの著作権は、シチズン・システムズ株式会社にあります。
 - CITIZEN は、シチズン時計株式会社の登録商標です。
 - Windows 及び Windows Server は、米国 Microsoft Corporation の米国およびその他の国における登録商標です。
 - Pentium は、米国およびその他の国における、Intel Corporation またはその子会社の商標または登録商標です。
 - QR コード、iQR コードは、(株)デンソーウェーブの登録商標です。
 - Android は、米国およびその他の国における Google Inc.の商標または登録商標です。
 - JavaScript は、米国およびその他の国における Oracle の商標または登録商標です。
- その他、記載されている会社名、製品名は、各社の商標または登録商標です。

目次

更新履歴	2
ご注意	3
著作権・商標について	3
目次	4
1. はじめに	5
1.1. ドキュメント対象範囲	5
1.2. システム概要	5
1.3. ライブラリ ファイル	8
1.4. ライブラリ 参照設定	7
1.5. 対応 PC	6
1.6. 対応プリンター	6
1.7. 機能一覧	9
2. ライブラリ インターフェース	10
2.1. コンストラクタ	10
2.2. Open メソッド	11
2.3. Close メソッド	12
2.4. BeginPrint メソッド	13
2.5. EndPrint メソッド	14
2.6. DoPrint メソッド	15
2.7. DoPreview メソッド	16
2.8. InitFrame メソッド	17
2.9. GetParts メソッド	18
2.10. SetPartsData メソッド	19
2.11. AddFrame メソッド	20
2.12. PreparePrint メソッド	21
2.13. HidePreview メソッド	22
3. コード サンプル (C#)	23
4. 注意事項	24

1. はじめに

本ドキュメントは、Windows Layout SDK via Printer Driver のプログラミング マニュアルです。

1.1. ドキュメント対象範囲

本ドキュメントは、CITIZEN レイアウト ファイルを利用する Windows アプリケーションの開発者が参照することを目的としています。

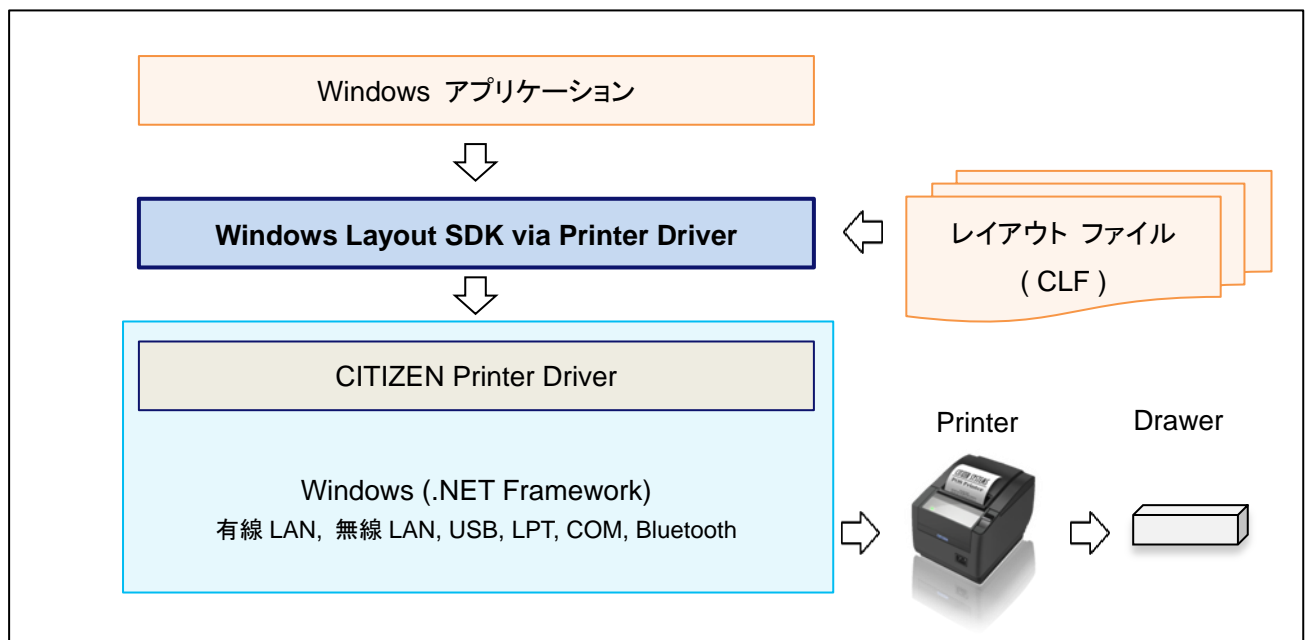
1.2. システム概要

本ライブラリは、CITIZEN レイアウト ファイルを利用する Windows アプリケーションから参照されることを想定しています。

本ライブラリは、CITIZEN Printer Driver を介して印刷します。

本ライブラリは、CLF 形式の CITIZEN レイアウト ファイルに対応します。

CLF 形式は、PC 搭載フォントを利用して印刷します。



ライブラリ システム 構成図

CITIZEN レイアウト ファイルは、Layout Editor で作成します。

Layout Editor の詳細については、「Layout SDK ユーザーズ・ガイド」をご参照してください。

1.3. 対応 PC

OS

- Windows XP
- Windows 7 (32bit, 64bit)
- Windows 8 (32bit, 64bit), Windows 8.1 (32bit, 64bit)
- Windows 10 (32bit, 64bit)
- Windows Server 2008, Windows Server 2008R2
- Windows Server 2012

動作に必要なソフトウェア

- .NET Framework 4.0
- CITIZEN Printer Driver^{*1}

^{*1} : "CITIZEN Text Only Printer" ドライバーは非対応です。

1.4. 対応プリンター

本ライブラリの対象プリンターは、シチズン製プリンタードライバーが対応するプリンターとなります。

各プリンターの機能詳細については、プリンターの取り扱い説明書をご参照ください。

1.5. 定義方法

<ライブラリの追加>

1. Layout SDK([**LayoutSDK_Setup_ja**])をインストールします。
詳細は、"Layout SDK ユーザーズ・ガイド" を参照してください。

<参照の追加>

Visual C# で参照を追加するには、

1. [ソリューション エクスプローラー]で、プロジェクト ノードを右クリックし、[参照の追加]をクリックします。
2. [参照の追加]ダイアログ ボックスで、[参照]タブをクリックします。
3. 以下のファイルを選択し、[OK]をクリックします。

```
C:\Program Files\CITIZEN\Layout Utilities\Citizen.LayoutUtilities.Printing.dll
```

Visual Basic で参照を追加するには、

1. [ソリューション エクスプローラー]で、対象のプロジェクトの[My Project]ノードをダブルクリックします。
2. プロジェクト デザイナーで、[参照設定]タブをクリックします。
3. [追加]をクリックして [参照の追加]ダイアログ ボックスを開きます。
4. [参照の追加]ダイアログ ボックスで、[参照]タブをクリックします。
5. 以下の 2 つのファイルを選択し、[OK]をクリックします。

```
C:\Program Files\CITIZEN\Layout Utilities\Citizen.LayoutUtilities.Printing.dll
```

<名前空間の定義>

Visual C# の場合：

```
using Citizen.LayoutUtilities.Printing;
```

Visual Basic の場合：

```
Imports Citizen.LayoutUtilities.Printing;
```

1.6. ライブラリ ファイル構成

本ライブラリは、以下のファイルで構成されています。

- AxInterop.QRMAKERADLib.dll
- Citizen.LayoutUtilities.Common.dll
- **Citizen.LayoutUtilities.Printing.dll***
- GrapeSystems.Core.Common.dll
- GrapeSystems.Core.Drawing.Fx20.dll
- GrapeSystems.Core.Parts.dll
- GrapeSystems.Core.Parts.Frames.dll
- GrapeSystems.Library.BarcodeAd.dll
- GrapeSystems.Library.Controls.dll
- GrapeSystems.Library.Image.dll
- Interop.QRMAKERADLib.dll

* : 対象のプロジェクトに参照の追加が必要なファイルです。

1.7. 機能一覧

本ライブラリは以下の機能を提供します。

メソッド一覧

No.	名称	機能
1	Controller	コンストラクタです。インスタンスを生成します。
2	Open	指定されたパス名の CLF レイアウト ファイルを開きます。
3	Close	現在開いている CLF レイアウト ファイルを閉じます。
4	BeginPrint	印刷の準備を開始します。
5	EndPrint	印刷データを破棄します。
6	DoPrint	指定されたプリンターで印刷を実行します。
7	DoPreview	指定されたプリンターによる印刷プレビューを表示します。
8	InitFrame	印刷用データを設定するフレームを検索します。 指定されたフレーム名で検索し、内部で管理しているフレーム番号を返します。
9	GetParts	印刷用データを設定する部品を検索します。 InitFrame() で取得したフレーム番号および部品名で、設定対象とする部品を検索し、内部で管理している部品番号を返します。
10	SetPartsData	部品に印刷用データ(テキスト、バーコード、イメージ)を設定します。 InitFrame() で取得したフレーム番号および GetParts() で取得した部品番号で、設定対象とする部品を検索し、指定された文字列を設定します。
11	AddFrame	印刷用データが設定されたフレームを印刷対象として登録します。
12	PreparePrint	初回印刷を高速化するために事前準備します。
13	HidePreview	

2. ライブラリ インターフェース

2.1. コンストラクタ

<定義>

```
Citizen.LayoutUtilities.Printing.Controller ()
```

<機能>

コンストラクタです。インスタンスを生成します。

<引数>

なし

<戻り値>

なし

<使用例>

```
Citizen.LayoutUtilities.Printing.Controller clpe =  
    new Citizen.LayoutUtilities.Printing.Controller();
```

2.2. Open メソッド

<定義>

```
int Open ( string pathName )
```

<機能>

指定されたパス名の CLF レイアウト ファイルを開きます。

<引数>

pathName

テンプレートとなる CLF レイアウト ファイルのフルパス名を指定します。

<戻り値>

- 0 : 正常に CLF レイアウト ファイルを開くことができたことを意味します。
- 0 以外 : CLF レイアウト ファイルが見つからないなど、何らかのエラーが発生したことを意味します。

<使用例>

```
clpe.Open( "my_layout_File.CLF" );
```

2.3. Close メソッド

<定義>

```
void Close ()
```

<機能>

現在開いている CLF レイアウト ファイルを閉じます。

<引数・戻り値>

なし

<使用例>

```
clpe.Close();
```

1.1. BeginPrint メソッド

<定義>

```
void BeginPrint ()
```

<機能>

印刷の準備を開始します。

<引数・戻り値>

なし

<使用例>

```
clpe.BeginPrint();
```

2.4. EndPrint メソッド

<定義>

```
void EndPrint ()
```

<機能>

印刷データを破棄します。

<引数・戻り値>

なし

<使用例>

```
clpe.EndPrint();
```

2.5. DoPrint メソッド

<定義>

```
int DoPrint ( string printerName )
```

<機能>

指定されたプリンターで印刷を実行します。

<引数>

printerName

印刷を行うプリンター名を指定します。

空文字列が指定された場合は、デフォルトプリンターに印刷します。

<戻り値>

0 : 正常終了したことを意味します。

0 以外 : 指定されたプリンターが見つからないなど、何らかのエラーが発生したことを意味します。

<使用例>

```
clpe.DoPrint( "Printer Name" );
```

2.6. DoPreview メソッド

<定義>

```
int DoPreview ( string printerName )
```

<機能>

指定されたプリンターによる印刷プレビューを表示します。

<引数>

printerName

印刷プレビューを行うプリンター名を指定します。

空文字列が指定された場合は、デフォルトプリンターによる印刷プレビューを表示します。

<戻り値>

0 : 正常終了したことを意味します。

0 以外 : 指定されたプリンターが見つからないなど、何らかのエラーが発生したことを意味します。

<使用例>

```
clpe.DoPreview( "Printer Name" );
```


2.7. InitFrame メソッド

<定義>

```
int InitFrame ( string frameName )
```

<機能>

印刷用データを設定するフレームを検索します。

指定されたフレーム名で検索し、内部で管理しているフレーム番号を返します。

<引数>

frameName

対象とするフレーム名を指定します。

<戻り値>

0 以上 : 内部で管理しているフレーム番号を意味します。

-1 : 指定されたフレームが見つからないなど、何らかのエラーが発生したことを意味します。

<使用例>

```
int frameIndex = clpe.InitFrame( "Frame1" );
```

2.8. GetParts メソッド

<定義>

```
int GetParts ( int frameIndex, string partsName )
```

<機能>

印刷用データを設定する部品を検索します。

`InitFrame()`で取得したフレーム番号および部品名で設定対象とする部品を検索し、内部で管理している部品番号を返します。

<引数>

frameIndex

対象とするフレーム番号 (`InitFrame()`で取得したもの)を指定します。

partsName

対象とする部品名を指定します。

<戻り値>

- 0 以上 : 内部で管理している部品番号を意味します。
- 1 : 指定された部品が見つからないなど、何らかのエラーが発生したことを意味します。

<使用例>

```
int partsIndex = clpe.GetParts( frameIndex, "Text1" );
```

2.9. SetPartsData メソッド

<定義>

```
int SetPartsData ( int frameIndex, int partsIndex, string setText )
```

<機能>

部品に印刷用データ(テキスト、バーコード、イメージ)を設定します。

InitFrame()で取得したフレーム番号と、GetParts で取得した部品番号で、
設定対象とする部品を検索し、指定された文字列を設定します。

<引数>

frameIndex

対象とするフレーム番号 (InitFrame()で取得したもの)を指定します。

partsIndex

対象とする部品番号 (GetParts で取得したもの)を指定します。

setText

設定する印刷用データ(テキスト、バーコード、イメージ)を指定します。

<戻り値>

- 0 : 正常終了したことを意味します。
- 0 以外 : 設定に失敗した場合など、何らかのエラーが発生したことを意味します。

<使用例>

```
clpe.SetPartsData( frameIndex, partsIndex, "New Text" );
```

```
clpe.SetPartsData( frameIndex, partsIndex, "New Image File Path" );
```

2.10. AddFrame メソッド

<定義>

```
int AddFrame ( int frameIndex )
```

<機能>

印刷用データが設定されたフレームを、印刷対象として登録します。

<引数>

frameIndex

対象とするフレーム番号 (initFrame() で取得したもの) を指定します。

<戻り値>

- 0 以上 : 内部で管理している印刷登録時のフレーム番号を意味します。
- 1 : 登録に失敗した場合など、何らかのエラーが発生したことを意味します。

<使用例>

```
clpe.AddFrame( frameIndex );
```

2.11. PreparePrint メソッド

<定義>

```
int PreparePrint ( string pathName )
```

<機能>

初回印刷を高速化するために事前準備します。

<引数>

pathName

利用する CLF レイアウト ファイルのフルパス名を指定します。

<戻り値>

- 0 : 正常終了したことを意味します。
- 0 以外 : 何らかのエラーが発生したことを意味します。

<使用例>

```
clpe.PreparePrint( "my_layout_File.CLF" );
```

—POINT—

本ソフトウェアは、.NET Framework を利用して印刷を行います。

.NET Framework ライブラリは、印刷の際にユーザーが意識することなく自動的にメモリへロードされます。初回印刷はその処理に数秒を要するために遅くなり、二回目以降の印刷は高速に処理されます。

PreparePrint(), HidePreview() は、この初回印刷の遅さを改善するために用意したメソッドです。初回印刷の遅さを改善したい場合、印刷前にいずれかのメソッドを 1 度 実行してください。

なお、改善の効果は動作環境に依存するため一律ではありません。

もし、二回目以降の印刷においても遅くなる現象が見られた場合、定期的に本メソッドを実行する事を検討して下さい。

2.12. HidePreview メソッド

<定義>

```
int HidePreview ( string pathName )
```

<機能>

初回印刷を高速化するために事前準備します。

HidePreview() は PreparePrint() より初回印刷が高速化されます。ただし、HidePreview() の処理時間は長くなり、また、プレビュー生成ダイアログが表示される点を考慮してください。

<引数>

pathName

利用する CLF レイアウト ファイルのフルパス名を指定します。

<戻り値>

- 0 : 正常終了したことを意味します。
- 0 以外 : 何らかのエラーが発生したことを意味します。

<使用例>

```
clpe.HidePreview( "my_layout_File.CLF" );
```

—POINT—

本ソフトウェアは、.NET Framework を利用して印刷を行います。

.NET Framework ライブラリは、印刷の際にユーザーが意識することなく自動的にメモリーへロードされます。初回印刷はその処理に数秒を要するために遅くなり、二回目以降の印刷は高速に処理されます。

PreparePrint(), HidePreview() は、この初回印刷の遅さを改善するために用意したメソッドです。初回印刷の遅さを改善したい場合、印刷前にいずれかのメソッドを 1 度 実行してください。

なお、改善の効果は動作環境に依存するため一律ではありません。

もし、二回目以降の印刷においても遅くなる現象が見られた場合、定期的に本メソッドを実行する事を検討して下さい。

3. コード サンプル (C#)

```
Citizen.LayoutUtilities.Printing.Controller clpe =  
    new Citizen.LayoutUtilities.Printing.Controller(); // Windows Layout SDK  
  
int result = clpe.Open( "CLF Layout File Name" );  
if (0 == result) {  
    clpe.BeginPrint();  
  
    int frameIndex = clpe.InitFrame( "Frame1" );  
    int partsIndex = clpe.GetParts( frameIndex, "Text1" );  
    clpe.SetPartsData( frameIndex, partsIndex, "New Text" );  
    clpe.AddFrame( frameIndex );  
  
    clpe.DoPrint( "Printer Name" );  
  
    clpe.EndPrint();  
    clpe.Close();  
}
```

—POINT—

詳しくはレイアウト SDK サンプル プログラムをご参照ください。

<http://www.citizen-systems.co.jp/support/download/printer/sdk/index.html>

4. 注意事項

本ライブラリの注意事項を以下に示します。

3.1. 印刷完了確認について

本ライブラリは、Windows が提供する Print Spooler サービスを利用して印刷します。このため印刷完了確認ができません。ご了承ください。

Windows Layout SDK via Printer Driver

プログラミング マニュアル

Version 1.4.0 用