



# *FastPDFGen for .NET*

*Ondemand & Realtime*

*PDF Generator for Office Report*

---

## ユーザーズマニュアル

2009/03/03 版

株式会社 PM9

[ThePM9.com](http://ThePM9.com)

*Copyright 2009 PM9, Inc.  
All rights reserved.*

## 1. ご利用方法

---

FastPDFGen は、複数のページテンプレート(雛形)に画像やテキストデータを流し込み、ページを結合することにより PDF ファイルを生成するプログラミングライブラリです。定型形式の帳票を出力する為に利用されることを想定しています。

FastPDFGen for .NET は、テンプレート作成ツール(mkPDFtpl.exe)、PDF 生成ライブラリ(.NET アセンブリファイル形式:FastPDFGen.dll、FastPDFGenGDI.dll)から構成されます。通常の利用方法は、.NET 言語で帳票出力をコントロールするホストプログラムを作成し、そのホストプログラムから FastPDFGen をコントロールして PDF ファイルを生成します。

### 実行例

#### (1) テンプレートファイルの作成

MS-Word、MS-Excel、Illustrator 等の文書作成ツール・デザインツールにより、テンプレートの元となるファイルを作成し、Acrobat と mkPDFGen.exe によりテンプレートファイルに変換します。

詳しい作成手順は、3 章「帳票テンプレート作成方法」を御参照下さい。

#### (2) 帳票生成プログラムの作成

プログラムソースをエディタにて作成します。(C#による例、ファイル名:pdfsamle.cs)

```
using System;
using PM9;

class HelloClass {
    static void Main() {
        FastPDFGen pdfgen =
            new FastPDFGen("guest::2009/12/31::7F7A57A900FE12B2E08C897CA68B1B3E");
        pdfgen.compressField();
        pdfgen.setCryptMode("", "", true, false, false, false);
        pdfgen.start("/tmp/pdfgensample_out.pdf");
        pdfgen.startPage("/tmp/pdfgensample_index.pdf.tpl");
        pdfgen.setPageFieldData("f1", "漢字 ABC");
        pdfgen.setPageFieldData("f2", "abcdefghi");
        pdfgen.endPage();
        pdfgen.startPage("/tmp/pdfgensample.pdf.tpl");
        pdfgen.setPageFieldData("f1", "漢字 ABC");
        pdfgen.setPageFieldData("f2", "abcdefghi");
        pdfgen.setPageFieldData("f3", "12,345,678");
        pdfgen.endPage();
        pdfgen.finish();
    }
}
```

上記プログラムをコンパイルします。

```
csc /target:exe /o+ pdfsample.cs /r:FastPDFGen.dll /r:FastPDFGenGDI.dll
```

### (3) 実行プログラムによる帳票データの生成

生成された帳票出力プログラムを実行して下さい。

```
pdfsample.exe
```

上記プログラムの実行結果として¥tmp¥pdfgensample\_out.pdf が生成されます。

## 2. インストール

---

インストールの必要があるファイルは、mkPDFtpl.exe、FastPDFGen.dll、FastPDFGenGDI.dll の 3 つです。これらのファイルは、パッケージ CD-ROM に同梱されています。

mkPDFtpl.exe は、Windows PC 上のユーザデスクトップもしくは、適当なフォルダにコピーして下さい。

テンプレートを作成する際、テンプレートの元ファイルを mkPDFtpl.exe コマンドのアイコン上にドラッグ&ドロップすることにより使用しますので、アクセスし易い場所にコピーして下さい。

FastPDFGen.dll 及び FastPDFGenGDI.dll は、帳票生成プログラムから参照可能な場所にコピーして下さい。

.NET 環境は、dll(.NET アセンブリ)の検索を下記のいずれかの場所から行います。

- ・DEVPATH 環境変数に列挙されているディレクトリ
- ・グローバル・アセンブリ・キャッシュ
- ・アセンブリのコードベース

### 3. 帳票テンプレート作成方法

---

#### (1) 帳票デザインの用意

- MS-Word、Excel、Illustrator 等の文書作成ツール・デザインツールを利用して帳票テンプレートの元となる帳票デザインを作成して下さい。

PDF 化できるものであればどのような形式でもかまいません。

Adobe Acrobat は、ほとんどの形式のファイルを PDF 化することができます。

- 帳票デザインは、表紙、明細、裏表紙等、テンプレート結合処理の単位で作成して下さい。帳票用紙サイズに制約はありません。

#### (2) 帳票デザインの PDF ファイル化

- Acrobat を使用して帳票デザインファイルを PDF に変換して下さい。

帳票デザインファイルを Acrobat に直接読み込ませるか、デザインの作成に使用したツールの印刷メニューからプリンタ「Adobe PDF」を選んで印刷を行い PDF ファイルを生成して下さい。(Acrobat Distiller による PDF の生成)

下記の Acrobat の各バージョンにて動作を確認しております。

- ・ Acrobat4
- ・ Acrobat5
- ・ Acrobat6 Professional
- ・ Acrobat7 Professional
- ・ Acrobat8 Professional
- ・ Acrobat9 Professional

Adobe Designer で作成されたフォームには対応しておりません。

## (3) アプリケーションデータ流し込みの為にフォームフィールドの作成

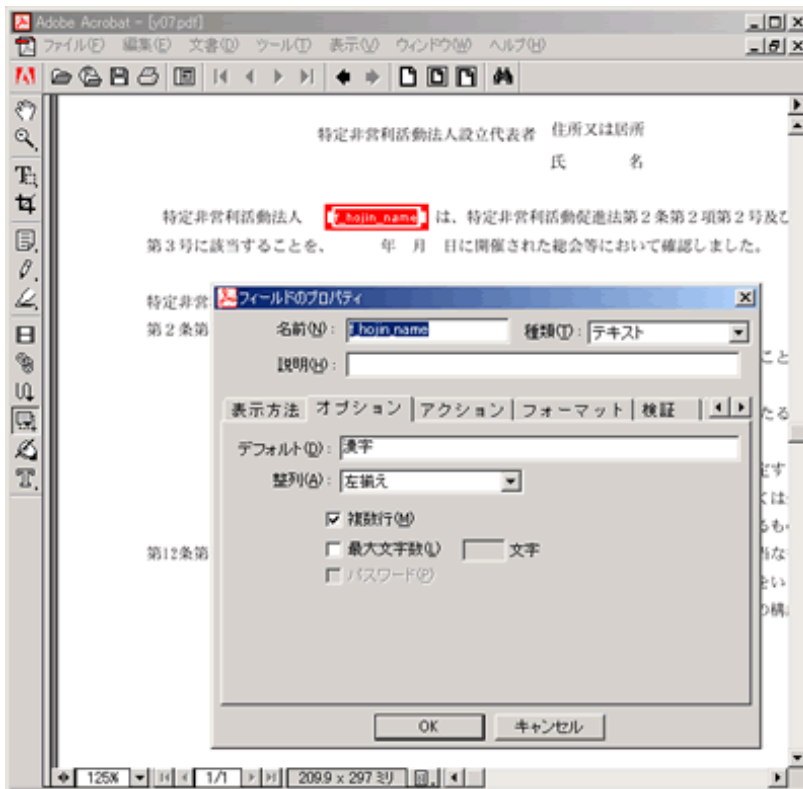
- 先程生成した PDF ファイルを Acrobat で開き、フォームフィールドツールを使ってフォームフィールドを作成して下さい。

テキストフィールドツールへのアクセスは、下記のようになります。

[プルダウンメニュー] [ツール] [高度な編集] [フォーム] [テキストフィールドツール]

または

[プルダウンメニュー] [ツール] [フォーム] [テキストフィールドツール]



テキストフィールドの各項目の設定を下記のように行って下さい。

- [一般] [名前]  
データを流し込む際に指定するフィールド名を設定。  
英数字および”\_”のみ使用可能。長さ 26 文字以内。
- [オプション] [デフォルト]  
必ず 1 文字以上の文字列を設定して下さい。文字列の内容は任意です。

- [オプション] [複数行]  
必ずチェックを ON にして下さい。
  
- [オプション] [整列]  
データを流し込んだときに指定の整列形式で表示されます。
  
- [表示方法]  
データを流し込んだときに指定の形式で表示されます。
  - ・フォント  
(例えば市販のバーコードフォントを指定することでバーコードの表示ができます)
  - ・サイズ  
「自動」は選択しないで下さい。かならず、いずれかのポイント数を選択して下さい。
  
- [フォーマット]、[検証]、[計算]  
FastPDFGen は、流し込んだテキストをそのまま表示します。  
ここで設定された属性は、無視されます。

FastPDFGen は PDF ファイル生成時に、フォームフィールドの ReadOnly 属性を ON にしフィールド名称を FastPDFGen が使用する内部名称に変更します。フォームフィールドの名称や ReadOnly 属性を変更したくない場合、例えばボタンや入力エリアとそれに連動する JavaScript での処理の為にフォームフィールドを定義する場合は、フォームフィールド名を、先頭が「\_」(アンダースコア)で始まる文字列に設定して下さい。

(4) 最適化オプション変更 (Acrobat9 Professional の場合のみ)

FastPDFGen は、オブジェクト生成オプションにて「ファイル全体を圧縮」を設定した PDF に対応していません。

次の要領で最適化オプションを変更して下さい。

- 1) 変換対象の PDF ファイルを Acrobat9 で開く。
- 2) アドバンスド PDF の最適化 最適化 オブジェクト生成オプションにて「文書構造を圧縮」に変更。
- 3) 画面右下の「OK」ボタンを押下げ。
- 4) 「最適化して名前を付けて保存」ダイアログボックスが表示されるので、「保存」ボタンを押下げ。

(5) 帳票デザインファイル(PDF 形式)の保存

編集作業を終えた後は、必ず「名前を付けて保存」によりファイルを保存して下さい。

「名前を付けて保存」を行うことにより、PDF の最適化が行われ、ファイルサイズを最小化することができます。結果的に FastPDFGen により生成される PDF ファイルも最適化されます。

(6) PDF 形式帳票デザインの帳票テンプレート(.tpl 形式)への変換

mkPDFtpl コマンドを使用して、帳票テンプレート(.tpl 形式)を作成します。

Windows デスクトップ上で mkPDFtpl.exe のアイコン上に PDF ファイルをドラッグして下さい。帳票デザインファイルと同一のディレクトリに .tpl 形式の帳票テンプレートファイルが作成されます。

コマンドラインから実行する場合は、下記のように指定して下さい。

mkPDFtpl 帳票デザインファイルの PATH(PDF 形式)

## 4. FastPDFGen 帳票生成クラス API リファレンス

使用宣言:ソースの先頭に記述

```
using PM9;
```

初期化(コンストラクタ)

```
public FastPDFGen FastPDFGen(string licenseKey)
```

licenseKey:                      ライセンスキー文字列

FastPDFGen 御購入の際、PM9 よりお知らせするライセンスキーを設定して下さい。

評価用ライセンスキーは、WEB 上(<http://www.pm9.com/newpm9/itbiz/pdf/index.php>)にて公開しております。

セキュリティ設定

```
public void setCryptMode(string user_password, string master_password,  
                          bool print_permission, bool mod_permission,  
                          bool copy_permission, bool annot_edit_permisson);
```

user\_password:                      ユーザパスワードの指定  
                                    文字列が長さ 0 の場合パスワード設定なし

master\_password:                    マスターパスワードの指定  
                                    文字列が長さ 0 の場合パスワード設定なし

print\_permission:                   印刷許可

mod\_permission:                    文書の変更許可

copy\_permission:                    内容のコピーまたは抽出許可

annot\_edit\_permisson:               注釈とフォームフィールドの作成許可

注意事項: 必ず「初期化」コマンドの直後で使用して下さい。

---

### PDF データ圧縮(フィールドデータの圧縮)

---

**public void compressField( );**

フォームフィールドオブジェクトのデータ圧縮を行い、生成されるPDFファイルのサイズを小さくします。

効率良く圧縮を行う為には、次のようにPDF生成コマンドを作成して下さい。

1. できるだけmakeArrayPageFieldコマンドを使用する
2. setPageImageData、drawLine、drawBoxコマンドは、makeArrayPageFieldコマンドの前に記述する

1つのテンプレートが2ページ以上で構成されている場合、そのテンプレートから生成されたページに対しては圧縮処理が行われません。

テンプレートは、1ページ単位で作成して下さい。

注意事項: 必ず「初期化」コマンドの直後で使用して下さい。

---

### PDF 生成処理開始

---

**public void start(string generate\_pdf\_path);**

generate\_pdf\_path: 生成する PDF ファイル PATH

---

### ページ生成・開始

---

**public void startPage(string template\_pdf\_path)**

template\_file\_path: 帳票テンプレートファイル PATH

---

フィールドの配列化・縦方向等間隔配置表示

---

**public void makeArrayPageField(string fieldName, int count, float interval)**

fieldName:       フィールド名  
count:            配列要素数  
interval:         縦方向配置間隔(配置間隔を POINT 数で指定)  
                  小数点付き数値で指定可能  
                  1 POINT = 1/72 inch

配列化されたフィールドを `setPageFieldData` コマンド等で指定する際は、配列化前のフィールド名称の後ろに「\_(1から始まる要素番号)」(アンダースコアを2文字分と数字)を付加した文字列となります。

---

フィールドへのテキストデータ流し込み

---

**public void setPageFieldData(string fieldName, string fieldData)**

fieldName:       フィールド名  
fieldData:       流し込みデータ  
                  流し込みデータ中に改行文字(¥n)を入れることにより、  
                  フィールド中で改行することができます。

## フィールドへのテキストデータ流し込み(修飾指定)

---

```
public void setPageFieldDataWithAttribute(string fieldName, string fieldData,  
float linefeed, float font_size, int color)
```

fieldName: フィールド名

fieldData: 流し込みデータ

linefeed: 改行幅(POINT 数指定)を小数点付き数値で指定  
1 POINT = 1/72 inch  
-1 を指定するとフォームフィールド定義時のデフォルト値を使用

font\_size: フォントサイズ(POINT)を小数点付き数値で指定  
-1 を指定するとフィールド定義時のフォントサイズを使用

color: フォントカラー  
各 RGB 値を 16 進値で指定  
例) 0xFF0000 赤、 0x00FF00 緑、 0x0000FF 青  
-1 を指定するとフィールド定義時のフォントカラーを使用

## フィールドへのテキストデータ流し込み(修飾指定・フィールドフォントの指定)

---

```
public void setPageFieldDataWithAttribute(string fieldName, string fieldData,  
float linefeed, float font_size, int color, string font_name)
```

font\_name: フォント名称を指定  
指定する Font 名称は、テンプレート作成コマンド mkPDFGen の実行時に生成された.log ファイルの末尾に表示される Font 一覧のうちいずれか(font short\_name の部分)を指定する  
フォームフィールドに指定したい Font が無い場合は、  
ダミーのフォームフィールドを作成し、使用したい Font を設定する

---

フィールドへのテキストデータ流し込み(修飾指定・フィールドフォント・改行方式の指定)

---

```
public void setPageFieldDataWithAttribute(string fieldName, string fieldData,  
float linefeed, float font_size, int color, string font_name, int linefeed_mode)
```

font\_name:        フォント名称を指定  
                  ""(長さ 0 の文字列)を指定すると、フィールドのデフォルトフォントで表示される

linefeed\_mode:    複数行表示時の改行方式を指定  
                  「0」を指定した場合、日本語ワープロ風に自動改行を行う  
                  「1」を指定した場合、欧文ワープロ風に自動改行を行う  
                  「2」を指定した場合、自動改行を行わない

---

フィールドへのテキストデータ流し込み(修飾指定・フィールド中の文字列表示開始位置指定)

---

```
public void setPageFieldDataWithAttribute(string fieldName, string fieldData,  
float linefeed, float font_size, int color, string font_name, int linefeed_mode,  
start_ypos)
```

start\_ypos:        フィールド中の文字列表示開始位置を指定  
                  フィールド枠上段からの相対位置(POINT)を指定

---

フィールドへのテキストデータ流し込み(表示開始位置、文字間スペース、太字の指定)

---

```
public void setPageFieldDataWithAttribute(string fieldName, string fieldData,  
float linefeed, float font_size, int color, string font_name, int linefeed_mode,  
float start_ypos, float start_xpos, float char_spacing, float char_thickness)
```

start\_xpos:        フィールド中の文字列表示開始位置を指定  
                  フィールド枠左端からの相対位置(POINT)を指定

char\_spacing:     文字と文字の間のスペースを指定  
                  小数点付き数値(POINT 数指定)にて指定  
                  マイナス値を設定すると文字間が狭くなる

char\_thickness:   文字を指定した数量分太く表示する  
                  小数点付き数値(POINT 数指定)にて指定

---

### 円記号表示選択

---

#### **public void setYenSign(bool yenSignMode)**

yenSignMode: フィールドへのテキストデータ流し込み処理時の円記号の表示方式を選択する  
false のとき、流し込みデータ中の"¥"文字を"\"と表示  
true のとき、流し込みデータ中の"¥"文字を"¥"と表示

---

### フィールド非表示

---

#### **public void setPageFieldInvisible(string fieldName)**

fieldName: フィールド名

---

### JavaScript からアクセス可能なフィールド値の設定

---

#### **public void setPageFieldScriptValue(string fieldname, string fieldData)**

fieldName: フィールド名  
fieldData: 流し込みデータ

---

### 画像データ挿入(Path 指定/座標指定)

---

#### **public void setPageImageData(string imageFile, float xpos, float ypos, float zoom)**

imageFile: 元画像ファイルのパス  
xpos: 表示位置-X 軸(ページ左上を基点として POINT 数で指定)  
1 POINT = 1/72 inch  
ypos: 表示位置-Y 軸(ページ左上を基点として POINT 数で指定)  
zoom: 倍率

画像ファイルとして、BMP、GIF、JPEG、PNG 形式のファイルを指定することができます。  
IIS 環境で実行する場合、元画像ファイルは絶対パスで指定して下さい。

## 画像データ挿入(Bitmap オブジェクト指定/座標指定)

```
public void setPageImageData(Bitmap bitmap, float xpos, float ypos, float zoom)
```

bitmap:	Bitmap オブジェクト
xpos:	表示位置-X 軸(ページ左上を基点として POINT 数で指定) 1 POINT = 1/72 inch
ypos:	表示位置-Y 軸(ページ左上を基点として POINT 数で指定)
zoom:	倍率

## 画像データ挿入(Path 指定/フォームフィールド指定)

```
public void setPageFieldImageData(string fieldname, string imageFile, float zoom, int align,  
int valign)
```

fieldName:	画像の表示位置を指定するためのテキストフォームフィールドの名称
imageFile:	元画像ファイルのパス
zoom:	倍率を小数点付き数値で指定 0を設定すると、自動フィット
align:	左右寄せルール(-1: 左寄せ, 0: 中央, 1: 右寄せ)
valign:	上下寄せルール(-1: 上寄せ, 0: 中央, 1: 下寄せ)

画像ファイルとして、BMP、GIF、JPEG、PNG 形式のファイルを指定することができます。  
IIS 環境で実行する場合、元画像ファイルは絶対パスで指定して下さい。

## 画像データ挿入(Path 指定/フォームフィールド指定/画像背景の透明化)

```
public void setPageFieldImageData(string fieldname, string imageFile, float zoom, int align,  
int valign, int mask_mode)
```

mask_mode:	画像背景の透明化 (0: 透明化無し, 1: 白地を透明化, 2: 黒地を透明化)
------------	--

画像を他のフォームフィールドの上に上被せする場合は、画像を流し込むフィールドを最後に作成・定義して下さい。

画像ファイルとして、BMP、GIF、JPEG、PNG 形式のファイルを指定することができます。  
IIS 環境で実行する場合、元画像ファイルは絶対パスで指定して下さい。

画像データ挿入(Bitmap オブジェクト指定/フォームフィールド指定)

---

**public void setPageFieldImageData(string fieldname, Bitmap bitmap, float zoom, int align, int valign)**

fieldName:	画像の表示位置を指定するためのテキストフォームフィールドの名称
bitmap:	画像 Bitmap オブジェクト
zoom:	倍率を小数点付き数値で指定 0を設定すると、自動フィット
align:	左右寄せルール(-1: 左寄せ, 0: 中央, 1: 右寄せ)
valign:	上下寄せルール(-1: 上寄せ, 0: 中央, 1: 下寄せ)

画像データ挿入(Bitmap オブジェクト指定/フォームフィールド指定/画像背景の透明化)

---

**public void setPageFieldImageData(string fieldname, Bitmap bitmap, float zoom, int align, int valign, int mask\_mode)**

mask_mode:	画像背景の透明化 (0: 透明化無し, 1: 白地を透明化, 2: 黒地を透明化)
------------	--

## 線分の描画

```
public void drawLine(float xpos, float ypos, float width, float height, float line_width, int dash1, int dash2, int dash3, int dash4)
```

xpos: 線分始点位置-X 軸(ページ左上を基点として POINT 数で指定)を  
小数点付き数値で指定

1 POINT = 1/72 inch

ypos: 線分始点位置-Y 軸(ページ左上を基点として POINT 数で指定)を  
小数点付き数値で指定

width: 線分始点位置から終点位置までの幅(POINT 数で指定)を  
小数点付き数値で指定

height: 線分始点位置から終点位置までの高さ(POINT 数で指定)を  
小数点付き数値で指定

line\_width: 線分の太さ(POINT 数で指定)を小数点付き数値で指定

dash1、dash2、dash3、dash4: 破線のパターン指定

dash1: 破線の最初の黒線部分の長さを POINT 数で指定

dash2: dash1 に続く白線部分の長さを POINT 数で指定

dash3: dash2 に続く黒線部分の長さを POINT 数で指定

dash4: dash3 に続く白線部分の長さを POINT 数で指定

実線の例: dash1=1, dash2=0, dash3=0, dash4=0

破線の例: dash1=5, dash2=2, dash3=0, dash4=0

一点鎖線の例: dash1=10, dash2=2, dash3=4, dash4=2

## 矩形の描画

---

```
public void drawBox(float xpos, float ypos, float width, float height, float line_width, int dash1,
int dash2, int dash3, int dash4)
```

xpos: 矩形表示位置-X 軸(ページ左上を基点として POINT 数で指定)を  
小数点付き数値で指定

1 POINT = 1/72 inch

ypos: 矩形表示位置-Y 軸(ページ左上を基点として POINT 数で指定)を  
小数点付き数値で指定

width: 矩形の幅(POINT 数で指定)を小数点付き数値で指定

height: 矩形の高さ(POINT 数で指定)を小数点付き数値で指定

line\_width: 線分の太さ(POINT 数で指定)を小数点付き数値で指定

dash1, dash2, dash3, dash4: 破線のパターン指定

dash1: 破線の最初の黒線部分の長さを POINT 数で指定

dash2: dash1 に続く白線部分の長さを POINT 数で指定

dash3: dash2 に続く黒線部分の長さを POINT 数で指定

dash4: dash3 に続く白線部分の長さを POINT 数で指定

実線の例: dash1=1, dash2=0, dash3=0, dash4=0

破線の例: dash1=5, dash2=2, dash3=0, dash4=0

一点鎖線の例: dash1=10, dash2=2, dash3=4, dash4=2

## ページ生成・終了

---

```
public void endPage()
```

## PDF 生成処理終了

---

```
public void finish();
```

---

## 5. エラーコード

---

FastPDFGen は、パラメータ不正や環境上の異常を検出した場合、次のような例外を発生します。

ライセンスキー不正

---

System.MemberAccessException    License Error::Illegal Licensekey

ライセンスキー期限切れ

---

System.MemberAccessException    License Error::License Time Expired

コマンド呼び出し順不正

---

System.MemberAccessException    Illegal method calling sequence.

帳票テンプレート読み出しエラー

---

System.ArgumentException    File Read Error(帳票テンプレートファイル名)

帳票テンプレート上に指定のフォームフィールドが存在しない

---

System.ArgumentException    Undefined Field Name(フォームフィールド名)

指定のフォームフィールドにデフォルト値が設定されていない

---

System.ArgumentException    Undefined Field Appearance(フォームフィールド名)

フィールドへのテキストデータ流し込みにてテンプレート上に存在しないフォントを指定した

---

System.ArgumentException    Undefined Font(font short\_name:フォント名)

イメージファイル読み出しエラー

---

**System.ArgumentException**    Imagefile Read Error(イメージファイル名)

生成された PDF ファイルの書き込みエラー

---

**System.ArgumentException**    File Write Error(生成 PDF ファイル名)

ファイルオープンエラー

---

**System.ArgumentException**    fopen::error(ファイル名)

ページ内流し込み文字列処理用ワークスペースオーバーフロー(制限:4 万文字/頁)

---

**System.MemberAccessException**    Field string workspace overflow. (Limit: ???KB/page)

**FastPDFGen for .NET ユーザーズマニュアル**

---

株式会社 PM9

〒158-0095

HomePage

e-mail

東京都世田谷区瀬田 3-8-14

<http://www.pm9.com/>

[info@pm9.com](mailto:info@pm9.com)

*Copyright 2009 PM9, Inc. All rights reserved.*

---