

# DP-DS820 システム サンプルプログラム (VB.Net 版 Ver.0.1.0)

2016 年 03 月 18 日  
大日本印刷株式会社



本書の著作権は権利者にあります。一部または全部の無断転載・無断複製を禁止します。



本書の内容は予告なく変更することがあります。



Microsoft, Windows, .Net Framework は、米国 Microsoft Corporation の米国およびその他の国における商標または登録商標です。



仕向けを表す(A)の付かない DP-DS820 を米国へ転売したり、米国で使用したりすることはできません。

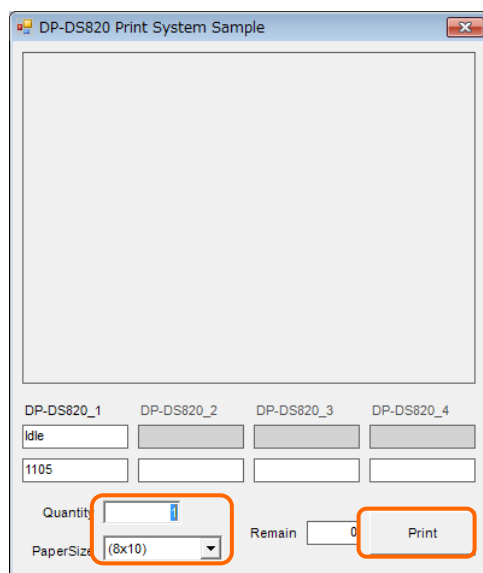
## 1. 概要

サンプルプログラムは、ホスト1台に対して4台の DP-DS820 を接続して印画することが可能です(1台～3台でも動作可能です)。印画要求がある場合、プリンターの空きバッファ数を確認して印画可能なプリンターを探し出し、印画データを送信します。

サンプルプログラムの対象は DP-DS820 です。

## 2. 操作方法

印画枚数を「Quantity」、用紙サイズを「Paper Size」に設定し、「Print」ボタンをクリックしてください。イメージが表示されデータが送信されます。プリンター毎にライフカウンタの値とステータスが表示されます。(\*1)



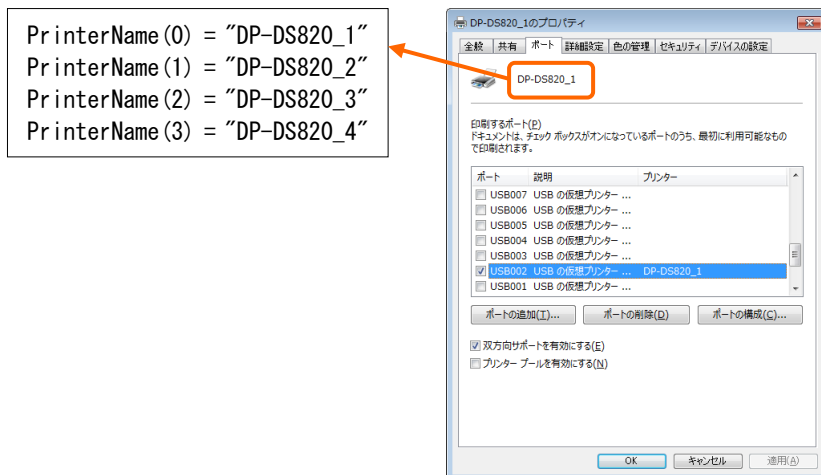
\* 1. 本サンプルプログラムには、エラー発生時のデータの再送信処理は入っていません。

### 3. ホストに合わせて変更する部分

サンプルプログラムは VB.Net で作成しており、Form1.vb の Form1\_Load() において、プリンターのデバイス名の設定を行っていますが、この部分はホストの設定に合わせてデバイス名を変更する必要があります。

#### ■ プリンターのデバイス名

インストールされているプリンターの名前を指定してください。



#### ■ 32/64bit OS の選択

OS が 64bit の場合、DS820Stat.vb の先頭で「#Const x64 = True」としてください。  
初期設定は 32bit 用で「#Const x64 = False」となっています。

```
#Const x64 = False
```

### 4. 印画データ

#### ■ 画像フォーマット

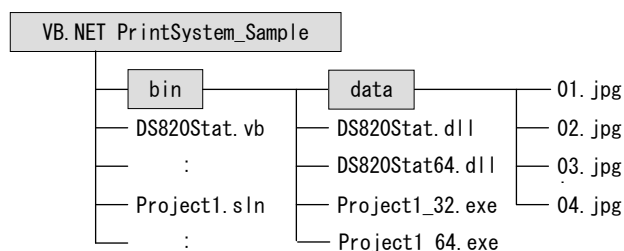
サンプルプログラムで対応している画像フォーマットは JPEG のみです。

#### ■ フォルダ

サンプルプログラムのフォルダにある "Data" フォルダに印画するデータを入れてください。

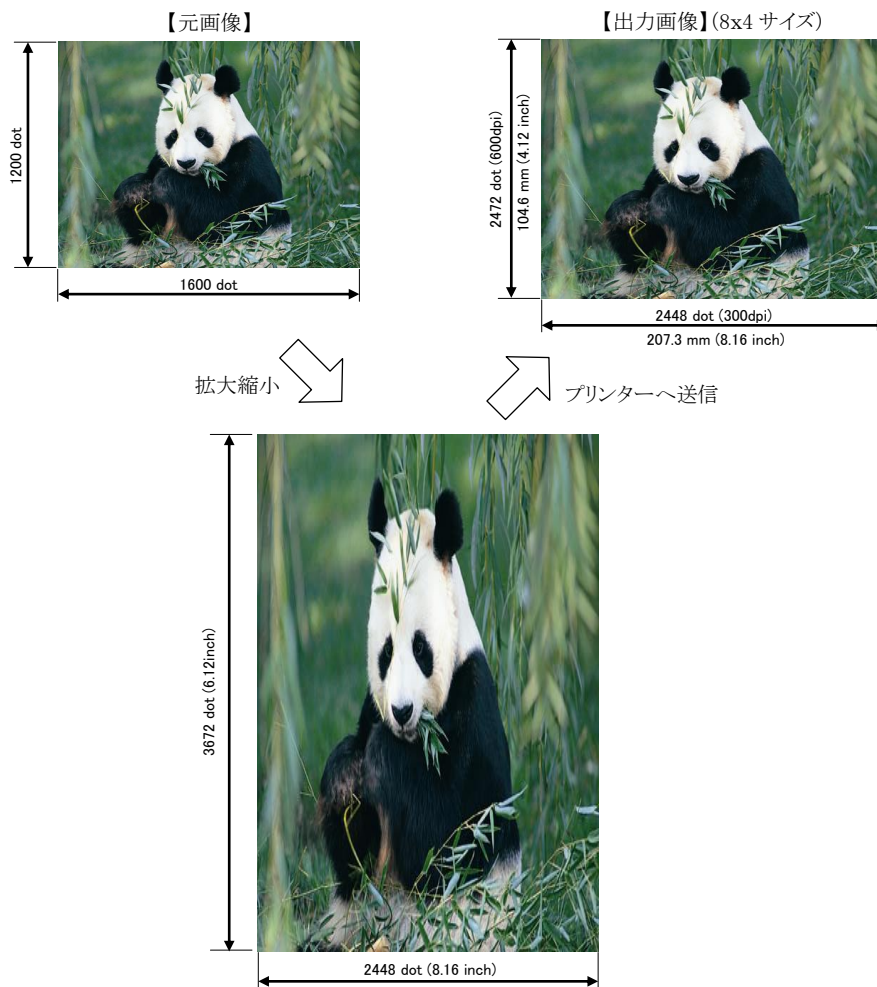
#### ■ ファイル名

サンプルプログラムでは、01.jpg ~ 04.jpg のファイルを繰り返しプリンターに送信しますので、印画する画像に 01~04 のファイル名をつけて、"Data" フォルダに入れてください。



## 5. 拡大縮小

Microsoft .Net を使用した印画は、.NetFramework の System.Drawing.Printing 名前空間を使用します。拡大縮小は、出力サイズを用紙サイズに合わせることににより行っています。プリンタドライバの設定から用紙の横サイズを読み込み、イメージの横サイズを用紙の横幅に合わせます。横サイズがどれだけ拡張されるのか比率を求め、その比率から縦サイズを求めています。



### ■ アプリケーション側での処理

解像度設定 300×600dpi で 8x4 サイズに出力しようとした場合、2448×2472dot のサイズに拡大縮小してプリンターに出力する必要があります。

System.Drawing.Printing 名前空間を使用してプリンターへ出力する場合、用紙サイズに合わせて出力サイズを指定すると、イメージ作成で拡大縮小が行われ、プリンターへ出力されます。このときサイズ指定は論理値が用いられますが、この単位はプリンターの解像度に関係なく用紙サイズにより決まります。8x4 サイズの場合、207.3×104.6mm の論理値(1/100inch 単位)は 816×412 となります。従って出力サイズの指定は、まず横サイズを 816 にして、横サイズを 816 にした場合の縦サイズを求め、Graphics.DrawImage でサイズ指定します。

実際のプログラムでは下記のように計算しています。

```

' 出力サイズを求める
Print_Width = e.PageBounds.Width           ← 816
Ratio_Width = Print_Width / img.Width       ← 1600 イメージの横サイズ (pixel 単位)
Print_Height = Img.Height * Ratio_Width     ← 1200 イメージの縦サイズ (pixel 単位)

' イメージを出力先へコピー
e.Graphics.DrawImage( Img, 0, 0, Print_Width, Print_Height )
                                816          612

```

まず横サイズを 816 にします。次に横サイズを 816 にした時の比率を求めます。その比率から縦サイズを求めると 612 となります。この値を Graphics.DrawImage に設定することにより、用紙サイズに合わせて拡大縮小が行われプリンターに出力されます。このときに、縦サイズ 612 は用紙の縦サイズ 412 をオーバーしていますが、オーバーした分は切り捨てられます。

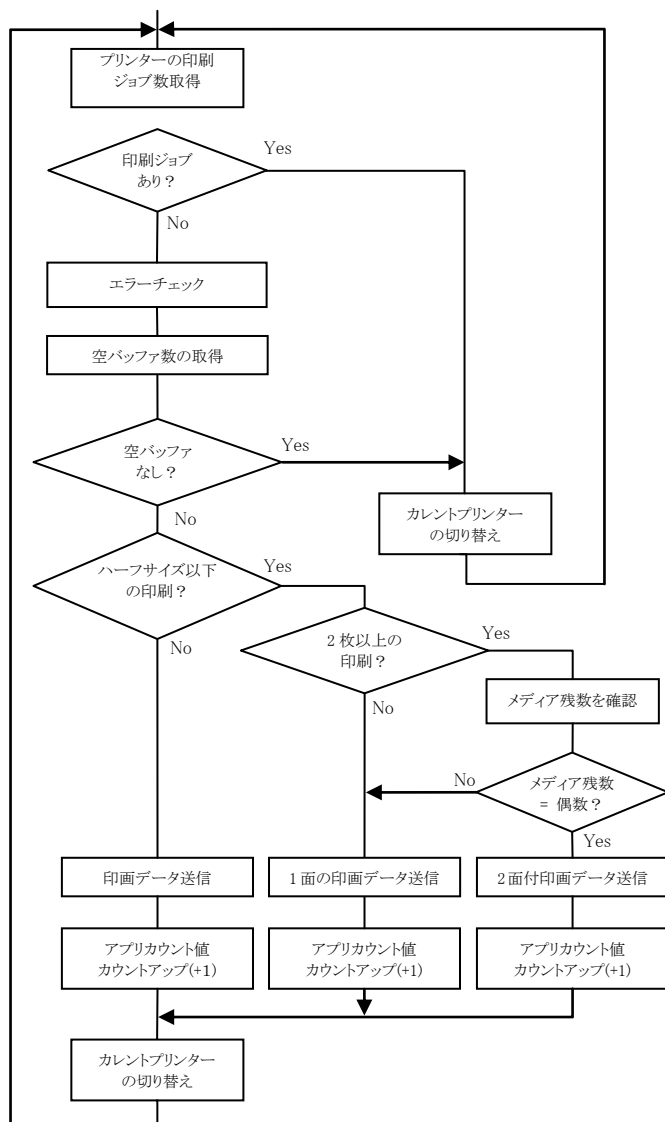
## 6. 印画データ送信手順

本サンプルプログラムは、以下の手順で印画データを送信しています。

- (1) 印画要求がある場合、印刷キューに印刷ジョブが在るか確認します。  
印刷ジョブが印刷キューに1分以上残っている場合は、エラーとして検出します。
- (2) 印刷ジョブが無い場合には、プリンターのステータスを取得し、エラーが発生していないか確認します。
- (3) プリンターにエラーが発生していない場合には、プリンターの空きバッファ数の値を取得します。
- (4) プリンターに空きバッファがある場合には、再度プリンターのステータスを取得し、エラーが発生していないか確認します。
- (5) プリンターにエラーが発生していなければ、印画データを出力します。

### ■ エラー処理について

- プリンタステータスを取得して、プリンターにエラーが発生している場合には、エラー処理(リカバリー)を行う必要があります。エラー処理は、エラー発生時のプリンターのライフカウンタ値と、アプリケーション側でカウントしているカウンタ値とを比較して、どの画像まで印画されたかを判断して、エラーの発生していない他のプリンターへデータを送信します。
- 正常に印画が行われているときには、印刷キューにある印刷ジョブは数秒間でプリンターへ送信され消去されます。そのため長い間印刷キューに残っている事はありません。しかし何らかの理由(ケーブル抜けなど)で通信ができなくなると、印刷キューにずっと残ってしまう可能性があります。そのため、印刷キューに印刷ジョブが在る場合には、時間を確認し、ある一定時間以上印刷キューに残っていたらエラーと判断し、エラー処理を行う必要があります。



### 【注意】

- 印画データは1枚または2枚のデータを1つの印刷ジョブとして出力してください。  
(それより多くのデータを1つのジョブとして送信すると、送信前に複数の画像を処理するため、スループットが遅くなります)。
- 同じ画像を複数枚印画する場合は、1枚毎に、印画枚数分データを送信してください。

※本サンプルプログラムには、データ再送信などのエラーリカバリー処理は入っていません。