

キャラクターモード使用 のフォームオーバーレイ の印刷について

PAGESプリンターではキャラクターモード（制御命令のテキスト形式の記述：詳細は下記）に変換されたプリンターの制御命令を使用することができます。

このキャラクターモードを使用することにより、ホスト・コンピュータ（ES/9000系あるいはAS/400系）からIBM日本語PC上のオンライン系ソフトウェア（3270系あるいは5250系）を介して、直接PC接続のページ・プリンターに制御命令やデータを送ることが可能になります。

ここでは、このキャラクターモードを中心に、拡張機能について説明します。

キャラクターモード（テキスト形式記述）を使用しての印刷は、5587以外の機種で可能となります。その場合、5584/5588/5586以外のプリンターでは次のエミュレーション・プログラム・カードが必須となりますのでご注意ください。

IBM5585-H01印刷装置の場合

5585エミュレーション・プログラム・カード III
(P/N: 57G9399 : フラッシュ・メモリ付)

あるいは

5585エミュレーション・プログラム・カード IV
(P/N: 57G9405 : フラッシュ・メモリなし)

IBM5589-H01印刷装置の場合

5589エミュレーション・カード III (P/N: 57G9346)

これらのIBMプリンターでは、主にIBM日本語PC上のオンライン系ソフトウェア（3270PC、5250PC等）を介してのホスト・コンピュータからのプリンター直接制御（オーバーレイや各種の制御命令の直接使用）を可能にするため、以下の機能が追加されています。

- I) **キャラクターモードによるデータ受信機能**
- II) **制御命令無視のための特殊モードの設定／解除コマンド**
(PAGES拡張制御コマンド)
- III) **オーバーレイ制御の機能追加**
(PAGES拡張制御コマンド)

1. キャラクターモード

1) キャラクターモードとは

これらの IBM ページプリンターには、様々な印刷を可能とするための数多くの制御命令が準備されています。これらの制御命令や、実際の印刷データは、16進表現で X'00' から X'FF' (10進数での0~255) までの1バイトで表示可能なすべての数値の中から自由に組み合わせられ、プリンターに送られます。

一方、オンライン系のシステムでは、ホストから端末側のプリンターに送ることのできるデータは、基本的に文字データに限られており、3270PC や 5250PC 等のオンライン系のプログラムの中でホスト系の文字コード (EBCDIC) から PC 系の文字コード (DOS 内部コード) に一意的に変換されてプリンターに送られています。このとき、特にプリンターの制御命令に頻繁に使用されている DOS 内部コード表現での X'20' 以下の数値は、文字コード用の範囲 (コード領域) から外れているため、ホストからは直接送ることができません。

また、プリンターの制御命令もホスト系のデータストリームである SCS (SNA データ形式) で制御できる範囲でしかサポートされていません。

つまり、本来ページプリンターが持っているいろいろな制御命令は PC からのみ直接送れるものの、ホスト (ES/9000 系または AS/400 系) からは直接送ることはできません。

そこでホストからオンライン系のプログラムを介してプリンターにデータを送る場合に、ホスト側ではプリンター用の制御命令をある法則に則って文字データに変換してから端末側へ送り出し、一方、端末側のオンライン系のプログラム上ではこれらはあたかも文字データとして処理を行わせ、次にプリンター側ではこれらを受信後、再度元の制御命令に戻すというような処理を行えば、ホストから直接、ワークステーション接続のページプリンターを制御することが可能になります。

これを実現したものがキャラクターモードです。

キャラクターモードとはテキスト形式で変換・記述されたデータをホストコンピュータから、PC のオンライン機能を介して、ページプリンターに送る一連の操作・手順を指します。

2) キャラクターモードの目的

上記の1)の説明のように、基本的にこのキャラクターモードは3270系、あるいは5250系のオンラインを経由しての、メインフレームホストからのPCプリンターの直接制御を可能とする目的を持っています。

この場合、オンライン系ソフトウェア(3270PC, 5250PC等)の変更は一切必要としておらず、ホスト側のアプリケーションソフトウェアとプリンター自体の変更(一部の機種ではエミュレーター・カードでの対応)により対処しようとするものです。

3) キャラクターモードの仕様の概略

以下にキャラクターモードの仕様の概略を説明します。

(以下の例示中のデータにおいて、C'...'は文字表現、X'...'は16進表現を示します。

3.1 ホスト側の処理(ホストアプリケーションプログラム上の処理)

まず、キャラクターモードで送るべきデータを準備します。

3.2 キャラクターモードの開始宣言文字列

特殊に割り当てられた半角4文字(C'&\$\$%' = X'26 24 25 24'あるいはC'\$?!#' = X'24 3F 21 23' [DOS内部コード表現]を選択して使用)をキャラクターモードの開始宣言文字列として扱います。

この開始宣言文字列は通常のデータ中にはありえないような文字の並び方で規定されています。

この開始宣言文字列はプリンター内部では2種類が準備されており、選択が可能です。キャラクターモードを使用するか、しないか、また使用の場合はどちらの開始宣言文字列を選択するかは、あらかじめプリンターの操作パネル上で設定しておく必要があります。

3.3 キャラクターモードでのデータ長(カウントフィールド)の指定

開始宣言文字列の次に4バイトからなるデータ長を示すフィールドが続きます。

このフィールド以降、キャラクターモードの終端部まではすべてテキスト形式でデータを記述する必要があります。

カウントフィールドで指定するデータ長は、プリンター内部でバイナリー形式に変換後のデータ長を指定します。すなわち、実際にテキスト形式で送られる実データ長の1/2の長さを指定します。

また、カウント自体は16進表現で表すものとします。これはひとつの開始宣言文字列で送られるバイナリー形式の実データの長さを最大64Kバイトまでとするためです。

例：

400バイトのテキスト形式のデータを送る場合、（バイナリー形式変換後は半分の200バイトとなるため、）

200 = X'C8' なので、

c' & \$ % \$ 0 0 C 8 ... ' = X' 26 24 25 24 30 30 43 38 ... '

と送ります。

3.4 キャラクターモードでのデータの表現形式

データの表現形式は、文字についてはDOS内部コード（16進表現）を用いて、それを1文字ずつの構成文字単位に分解し、その各々をさらにEBCDIC表現（ホスト上の場合）に変換したものを使用します。

また、制御命令やバイナリーデータに関しては16進表現されたものを同様に1文字ずつの構成文字単位に分解し、その各々をさらにEBCDIC表現に変換します。

このEBCDIC表現に使用される文字は C'0' ~ C'9' (X'F0' ~ X'F9') と C'A' ~ C'F' (X'C1' ~ X'C6') もしくは C'a' ~ C'f' (X'81' ~ X'86') に限られます。

このEBCDIC表現の文字コードは PC上のオンライン系ソフトウェアによりDOS内部コードに変換されます。

最終的にプリンター上ではこのDOS内部コード表現に変換された文字である

C'0' ~ C'9' (X'F0' ~ X'F9') と C'A' ~ C'F' (X'C1' ~ X'C6')

もしくは C'a' ~ C'f' (X'81' ~ X'86') にPCなって受信されます。

したがって、ホストを使用せずに、ワークステーションからキャラクター形式でデータを送る場合には、直接、DOS内部コード表現に変換して使用してください。

4) キャラクターモードの使用手順の例

以下の例ではホストから次の 11 バイトからなるプリンターの制御命令とデータを送る場合を想定しています。

```
X' 1B 7E 02 00 01 4B 41 42 43 0D 0A'
```

文字ピッチ設定 文字'ABC' 復帰 改行
(制御命令はバイナリー表現, 文字はDOS内部コード表現での記述)

4.1 ホスト側の処理 (ホストのアプリケーションプログラム上の処理)

- まず、ホストはキャラクターモードで送るべきデータを準備します。この場合は上記の11バイトですが、もしも送るべきデータ量が64Kバイトを超えるようなときは、64Kバイト以下に分割します。
- 次にホストはキャラクターモードの開始宣言文字列を最初に端末側に送ります。開始宣言文字列は2種類準備されており、どちらを使用するかはあらかじめ決めておきます。
- 次にホストはキャラクターモードを使用して、送り出すデータカウントをテキスト形式に変換して送ります。

例:

ホストはEBCDICコード体系を使用するため、開始宣言文字列とデータカウントは、次のようになります。データカウントは11であり、11は16進の4桁表現ではX'000B'となるため、

文字表現	:	C'	&	\$	%	\$	0	0	0	B'
EBCDIC 16進表現	:	X'	50	E0	6C	E0	F0	F0	F0	C2'
							開始宣言		カウント	

を端末側の送り出します。

- この後、ホストはキャラクターモードを使用して送り出すデータに対し、1バイトずつ順にテキスト形式に変換します。

例:

最初のデータであるX'1B'については、その16進表現の構成文字である、1とBに分解し、それぞれを'1'という文字データと'B'という文字データとして別々に取り扱います。

次に、'I'という文字データをEBCDIC文字コード表現であるX'F1'に変換し、次に'B'という文字データをEBCDIC文字コード表現であるX'C2'に変換します。この変換を次のように、すべてにデータに対して行います。

原データ: X' 1B 7E 02 00 01 4B 41 42 43 0D 0A'
 文字に分解: C' 1 B 7 E 0 2 0 0 0 1 4 B 4 1 4 2 4 3 0 D 0 A'
 コード変換: X' F1 C2 F7 C5 F0 F2 F0 F0 F0 F1 F4 C2 F4 F1 F4 F2 F4
 F3 F0 C4 F0 C1'

e) ホストはこれらのデータを端末のワークステーションに送り出します。

4.2 ワークステーションのオンライン系ソフトウェアの処理

f) ワークステーションでは上記c) 及びd) で生成されて送られてきたデータはすべて文字コードと認識するため、EBCDICからDOS内部コードへ一意的に変換します。(この処理はオンライン系ソフトウェアの内部で自動的に行われます。)

例:

EBCDIC表現:

開始宣言とカウント: X' 50 E0 6C E0 F0 F0 F0 C2' + X' F1 C2 F7 C5
 F0 F2 F0 F0 F0 F1 F4 C2 F4 F1 F4 F2 F4 F3 F0 C4 F0 C1'

DOS内部コードに変換された後:

開始宣言とカウント: X' 26 24 25 24 30 30 30 42' + X' 31 42 37 45 30
 32 30 30 30 31 34 42 34 31 34 32 34 33 30 44 30 41'

g) ワークステーションはDOS内部コードに変換された上記データをプリンターに送ります。

4.3 プリンターの内部処理

h) プリンターは「キャラクターモードの使用」が操作パネル上で設定されていると常に開始宣言文字列を検出するための処理を行います。

開始宣言文字列を検出すると、その直後の4バイトをテキスト形式のカウントとして認識し、後続のカウント分のデータに対し、いったんDOS内部コードに該当する文字に置き換えます。

DOS内部コードの文字に置き換えたもの : C' & \$ % \$ 0 0 0 B' +
 C' 1 B 7 E 0 2 0 0 0 1 4 B 4 1 4 2 4 3 0 D 0 A'

- i) 次のこのデータ部分に対して、連続する2バイトの文字毎に、文字の結合を行い、元のバイナリー形式のデータを再構築します。

例 :

C' 1 B 7 E 0 2 0 0 0 1 4 B 4 1 4 2 4 3 0 D 0 A'
 X' 1B 7E 02 00 01 4B 41 42 43 0D 0A'

- j) プリンターはこの再構築したバイナリーデータを正式な印刷用制御命令と認識し、通常の制御命令および文字コードとしての解釈を行ない、その結果を印刷します。

例 :

X' 1B 7E 02 00 01 4B 41 42 43 0D 0A'
 文字ピッチ設定 文字'ABC' 復帰 改行

- k) テキスト形式のデータがすべて無くなるまで上記処理を繰り返します。

プリンターはいったんキャラクターモードに入ると、テキスト形式のデータ中にエミュレータソフトウェアによって挿入されたバイナリー形式の制御命令は、特殊処理を必要とされるもの以外はすべて無視します。

この特殊処理される命令は印字結果に影響を及ぼす種類の命令ではないので、結果的にはホストとワークステーションプリンターの間で、非常に高いデータの透過性が保証されます。

3270PCプログラムのバージョン 9.0 以降では SCS 制御コードのうち、TRN 制御コード (Transparent : X'35') が機能拡張されました。この機能拡張された透過モードを使用すれば、キャラクターモードを使用しなくても、ホストから直接プリンターの制御命令を送ることが可能になります。

詳しくは 3270PC プログラムのバージョン 9.0 (以降) の説明書を参照してください。

2. オーバーレイ制御の機能

オーバーレイ・コマンドの形式とパラメーター：

書式：X'1B 7E 62 LEN N1 N2'

パラメーター：

LEN： フラグとオーバーレイIDを含んだデータ長（2バイト）

N1： フラグ = 制御情報（1バイト）

10：連続オーバーレイ制御

1F：連続オーバーレイの終了

N2： オーバーレイID（登録番号）（連続オーバーレイ制御の場合のみ複数バイト可，その他はすべて1バイト）

[00～FE：最大255種類のオーバーレイIDが指定可能です]

ただし，フラグがX'03'の場合は，IDの指定は不要となります。

1. 連続オーバーレイ制御（フラグ X'10'）

連続オーバーレイとは，プリンターがデータを印刷する際に，指定されたオーバーレイと各ページのデータとを毎ページ連続して，自動的に重ね合わせながら印刷する機能です。

この連続オーバーレイと，フラグがX'00'またはX'02'で指定される「オーバーレイ呼出し」との違いは，「オーバーレイ呼出し」ではオーバーレイ呼出し指定命令が印刷データ中の各ページの先頭に必要であるのに対し，連続オーバーレイでは，いったん印刷データの前頭に一回だけ指定しておけば，終了が宣言されるまでの後続のすべてのページにプリンターが自動的にオーバーレイを重ね合わせながら印刷する点にあります。

連続オーバーレイには次のような2種類の指定方法があります。

- a) 単一のオーバーレイを終了が宣言されるまで各ページに自動的に繰り返して展開するための指定。 [使用例・その1参照]
- b) 複数のオーバーレイを指定された順序で各ページに自動的に順次展開し，終了が宣言されるまで（順次の展開を）繰り返し印刷を行うための指定。 [使用例・その2参照]

以上いずれの場合も，オーバーレイは指定された方法で繰り返して発生され，印刷データは受信した順にページ毎に展開されてゆきます。

フラグ: X'10'

書式 : X'1B 7E 62 LEN 10 N2(1) N2(2) N2(n)'

N2(n) : 1バイトのデータでオーバーレイの登録番号を指します。

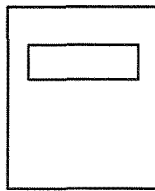
N2の有効範囲は、X'00'からX'FE'までです。(従来と同じ)

ただし、nの最大は16個までです。

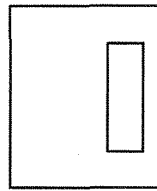
使用例 :

下記のいずれの場合も、既に単一あるいは複数のオーバーレイは登録が済んでいることを前提とします。

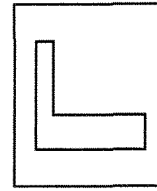
オーバーレイ
登録番号 #01



オーバーレイ
登録番号 #02



オーバーレイ
登録番号 #03



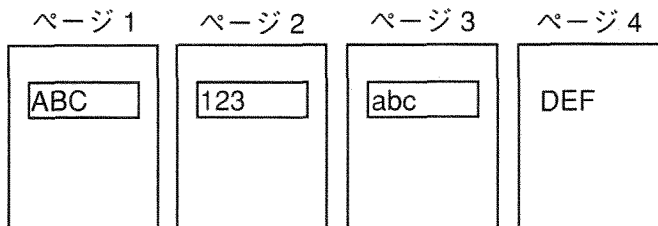
その1) 単一オーバーレイの連続印刷

コマンド列 :

X'1B 7E 62 00 02 10 01'	(連続印刷の開始) □内は登録番号
C'ABC' X'0D 0C'	(1ページ目のデータ)
C'123' X'0D 0C'	(2ページ目のデータ)
C'abc' X'0D 0C'	(3ページ目のデータ)
X'1B 7E 62 00 01 1F'	(連続印刷の終了)
C'EFG' X'0D 0C'	(4ページ目のデータ)

これは登録番号01のオーバーレイを毎ページ連続して印刷させることを意味します。

印刷結果 : 次の4枚の印刷が行われます。



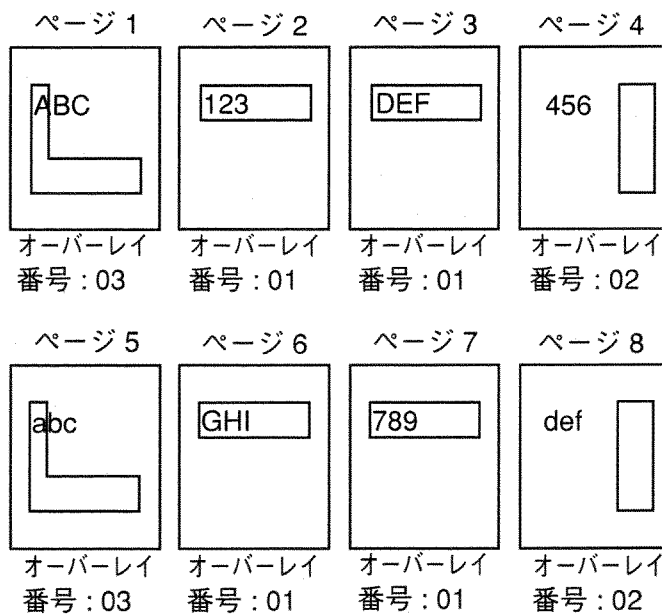
その2) 複数オーバーレイの連続印刷

コマンド列:

X' 1B 7E 62 00 05 10 03010102 ' (連続印刷の開始) □内は登録番号
 C' ABC ' X' 0D 0C ' (1ページ目のデータ)
 C' 123 ' X' 0D 0C ' (2ページ目のデータ)
 C' DEF ' X' 0D 0C ' (1ページ目のデータ)
 C' 456 ' X' 0D 0C ' (2ページ目のデータ)
 C' abc ' X' 0D 0C ' (3ページ目のデータ)
 C' GHI ' X' 0D 0C ' (1ページ目のデータ)
 C' 789 ' X' 0D 0C ' (2ページ目のデータ)
 C' def ' X' 0D 0C ' (3ページ目のデータ)
 X' 1B 7E 62 00 01 1F ' (連続印刷の終了)

これは登録番号 03 01 01 02 のオーバーレイをページ毎に順に繰り返しながら連続して印刷させることを意味します。

印刷結果: 次の 8 枚の印刷が行われます。



上記の印刷中、データの中に取消し命令 (CAN:X'18') が含まれていても、オーバーレイは終了しません。連続オーバーレイの終了の際は必ず次の「連続オーバーレイの終了」命令を使用してください。

なお、強制的に連続オーバーレイを終了させる場合は、プリンターをいったん印刷不可にしたのち、プリンターの電源を切ってください。

2. 連続オーバーレイの終了 (フラグ X'1F')

連続オーバーレイの印刷を終了します。

この制御命令は必ず、オーバーレイを印刷する最後のページの印字開始命令の直後 (次ページのTOF) に置いてください。

プリンターはこの制御命令を受信後は (その現在のページから) 直ちに連続オーバーレイの自動展開を止め、通常の印字に戻ります。

【キャラクターモードをご使用の際の注意事項】

●開始宣言文字列およびデータ長 (カウントフィールド) に誤りがないかを厳重にチェックのうえご使用ください。

また、キャラクターモードのご使用に先立って、元のバイナリー形式のデータが正しく印字することを事前に確認されることをおすすめします。

ソースデータとなる元のバイナリー形式のデータが正しく制御命令の形式を保持していないと、予想されない印字結果やプリンターの誤動作の原因となります。

●ひとつの制御命令を2つ以上の開始宣言文字列を含むキャラクターモードに分割して送ることはできません。

すなわち、ひとつのキャラクターモードのデータ列 (開始宣言文字列+カウントフィールド+カウント分のデータ) の中で使用する制御命令は次のキャラクターモードのデータ列にまたがらないようにしてください。

特に64Kバイトを超えるような (例: イメージデータを含むような) 長いデータ列をキャラクターモードで送る場合は、あらかじめそれらを複数の64Kバイト以下のデータに分割してください。そして、分割されたデータにはおののちに正しい制御命令を付加し、ひとつが64Kバイト以下の完結された命令として生成されるようにしてください。

●キャラクターモードをご使用にならなくなった場合は、プリンターの操作パネル上で「使用しない」を設定してください。

印刷データ中にキャラクターモードの開始宣言文字列と同じデータ（半角4文字）が含まれている場合の誤動作の原因となります。

また、プリンターの操作パネルでキャラクターモードを使用することを設定した後は、ホストやPS/55上からバイナリー形式で開始宣言文字列と同じデータを開始宣言文字列以外には使用しないでください。

●3270系のオンラインでホストからの直接印刷を行う場合は、ホスト上の適用業務プログラムとのセッションはLUタイプ1（SCSデータ・ストリーム）を使用し、LUタイプ3（3270データ・ストリーム）は使用しないでください。正常な印字結果が得られない場合があります。

●ホストからフォームオーバーレイの直接印刷を行う場合、以下の手順に従っていただくことをおすすめします。

1) PCからホストへのフォームのアップロード

フォームオーバーレイ自体のデータはフォームオーバーレイジェネレータにて生成したテキスト形式のファイルをそのまま使用してください。

PCからホストへは、基本的にJISCIIからEBCDICへのコード変換を行ってからアップロードしてください。PC上でのバイナリー形式のままではホストへはアップロードしないでください。

アップロードの際のファイル転送はIBMから正式にサポートされている転送ユーティリティ・プログラムをご使用ください。

ホスト上ではこのフォームデータはテキストデータとして扱われることを前提としています。また、フォームのアップロードの際に、フォームデータをプリンター上から消去するためのフォーム消去ファイルも併せてアップロードしてください。

フォーム消去ファイルはFOGCLR.***というファイル名でPCのオーバーレイ用サブディレクトリーに入っています。

2) ホストからPCへのフォームのダウンロード

ホストからフォームデータをPCへ送り出す場合、SCSデータ・ストリームに含まれるテキスト・データとして扱ってください。

テキストデータとして送られると、オンライン系プログラムの中でEBCDICからJISCIIへの変換が行われ、PC上のデータに復元されます。

この復元されたデータは文字データとしてプリンターに送られます。

3) ホストから PC へのデータ送付

ホスト上でのデータ生成は何らかのアプリケーション・プログラムにて行っているだけでありますが、このアプリケーション・プログラムの作成の際は次の2点に関して十分な考慮をはらってください。

a) データのフォーム形態に合わせた再配置

データ・ベース等のデータに対し、フォームオーバーレイ印刷を行うためには、あらかじめデータを実際使用するフォームオーバーレイ帳表にしたがったカラム位置および行位置に合わせた位置調整を行っておく必要があります。このようなデータの位置調整を行うことを再配置化と呼びます。

この再配置化はカラム位置に関してはスペース文字による調整、行位置に関しては、復帰改行命令による調整を原則とします。

データの印刷が複数ページにわたる場合、ページの区切りには改ページ命令を入れてください。

これらの制御命令の使用方法に関しては b) を参照してください。

b) データ中の制御命令について

オンライン系ソフトウェアはホストからのSCS制御命令や、ソフトウェア自体が管理している印刷制御情報に基づき、独自で生成・変換したプリンター制御命令を発行します。

キャラクターモードを使用したフォームオーバーレイ印刷ではオンライン系のソフトウェアはオーバーレイを印刷していることを全く意識していないので、独自で発行される制御命令はオーバーレイ印刷の際に位置制御等で悪影響を与えます。

そのため、プリンターはオンライン系ソフトウェアが独自に発行する制御命令は無視して、文字データだけを印刷する機能を持っています。その機能を有効に利用することにより、ホスト側から PC 接続プリンターの制御がほぼ完全に行えます。

そのための手順は以下の通りです。

- i) 印刷データの先頭に上記 II の「制御命令無視モードの設定」命令をキャラクターモードで送ってください。

例：

&\$%\$00061B7E14000100 (文字データとして送る)

- ii) 実際の文字データはそのまま送ってください。

- iii) 上記 a) のデータの再配置制御のための復帰改行や改ページ命令はすべてキャラクターモードで送ってください。

例（復帰改行）：&\$\$%\$00020D0A

例（改ページ）：&\$\$%\$00010C

- iv) 復帰改行や改ページ以外の制御命令もキャラクターモードで送ることが可能です。位置の微調整，文字（フォント）の変更・拡大・縮小・修飾，文字ピッチ・行ピッチの変更，グラフィック描画，イメージデータの転送等も行えます。
- v) 全印刷データを送出後，「制御命令無視モードの解除」命令をキャラクターモードで送ってください。

例：

&\$\$%\$00061B7E14000101（文字データとして送る）

- 4) ホストから PC への使用したフォームの消去命令

上記 1) でフォームデータと一緒にアップロードしたフォーム消去ファイルをそのまま使用してください。

このフォーム消去ファイルもキャラクターモードに既に変換されています。

上記 2) のフォームのダウンロードと同じ方法で送り出してください。

以上でフォームオーバーレイの一連の印刷が終了します。

このほかフォームは PC 側で管理し，データのみをホストから印刷する方法がありません。

MMR方式のイメージ圧縮 について

大量のイメージデータをプリンターに送信する時はイメージの圧縮を利用すると効率的です。

イメージの圧縮アルゴリズムについては、数々の方式がありますが、PAGESのプリンターでは、現在イメージ描画コマンド (ESX 89) の Image Encoding Parameterの COMPRID (182ページ参照) の指定により、次の3つの選択が可能です。

COMPRID = X'01' IBM MMR方式
X'03' 圧縮なし
X'82' G4 MMR方式
X'86' RunLength 4

1. IBM MMR (Modified CCITT Modified READ) について

IBM MMRはCCITT (国際電信電話諮問委員会) のModified READ (G3 MR) アルゴリズムをもとにIBMが開発したものです。

G3 MRは1次元と2次元の白黒の変換点の相関をコード化し、圧縮するものです。

1次元モード (1D)では白黒のビットをそれぞれ水平方向のビット長のコード化によって圧縮するものです。

2次元モード (2D)では前ラインの白黒変化と現ラインの白黒変化の相関を主体にコード化することによって圧縮するものです。

IBM MMRとG3 MRとの相違点はつぎのとおりです。

CCITTで定義される「K値」が無限大となる。
つまり 1Dは最初のラインのみとなる。

EOL (End Of Line) は 1Dの前と 1Dから 2Dに変わる時、それから EOP (End Of Page) を構成する EOLだけとなる。

タイムフィルビットはない。

また IBM MMRとCCITTの Modified Modified READ (G4 MMR) との主な相違点は、IBM MMRが最初に 1Dデータで圧縮し、以降を EOLなしの 2Dデータで圧縮しているのに対して、G4 MMRが仮想の白ラインと与えられた水平ドット数で以降すべてを EOLなしの 2Dデータで圧縮している点です。

また IBM MMRでは EOPはImage Contentの終りに 1つだけ置かれます。

注意

■ IBM MMRではイメージ・データから水平方向、垂直方向のドット数が最終的に判るので Image Size Parameter の HSIZE, VSIZEはそれぞれ 0で構いません。

CCITT Modified READ (G3 MR) の詳細については次の資料または G3ファクシミリの一般的な解説書を参考にしてください。

"CCITT Recommendation T.4."CCITT T Series Volume VII - Fascicle VII.3

IBM MMRについては次節「IBM MMRとG3 MR, G4 MMRの関係」が参考になります。また同様の解説が次の文献に詳細に述べられています。

"Binary-image manipulation algorithms in the Image View Facility"
the IBM Journal of Research and Development,
Volume 31, Number 1, January 1987

2. 圧縮なしについて

この時は圧縮されません。

注意

- 水平方向のドット数は復元前に判らないので指定する必要があります。垂直方向のドット数は最終的に判るので指定する必要はありません。
従って、Image Size ParameterのHSIZEはかならず0以外の指定が必要で、VSIZEは0で構いません。

3. G4 MMR (CCITT T.6 Facsimile Coding Scheme) について

G4 MMRはCCITT (国際電信電話諮問委員会)が CCITT T.6 G4 Facsimile Coding Schemeで標準化した圧縮方法です。

注意

- G4 MMRでは2次元のみの圧縮で最初に仮想の白ラインを仮定しているため必ず水平方向のドット数の指定が必要です。
従って、Image Size ParameterのHSIZEはかならず0以外の指定が必要で、VSIZEは0で構いません。

CCITT T.6 G4 Facsimile Coding Scheme (G4 MMR)の詳細については次の資料、またはG4ファクシミリの一般的な解説書を参考にしてください。

"CCITT Recommendation T.6."CCITT T Series Volume VII - Fascicle VII.3

4. IBM MMRとG3 MR, G4 MMRの関係

付録

G3 MR	IBM MMR	G4 MMR (仮想白ライン+水平ドット数)
1D EOL	1D EOL	
1D データ + タイムフィルビット(任意)	1D データ	2D データ
2D EOL	2D EOL	
2D データ + タイムフィルビット(任意)	2D データ	2D データ
.	2D データ	2D データ
.	2D データ	2D データ
2D EOL	.	.
2D データ + タイムフィルビット(任意)	.	.
1D EOL	.	.
1D データ + タイムフィルビット(任意)	.	.
2D EOL	.	.
2D データ + タイムフィルビット(任意)	.	.
.	.	.
.	.	.
2D EOL	.	.
2D データ + タイムフィルビット(任意)	.	.
.	.	.
.	.	.
1D EOL	.	.
1D データ + タイムフィルビット(任意)	.	.
2D EOL	.	.
2D データ + タイムフィルビット(任意)	.	.
.	.	.
.	.	.
2D EOL	2D データ	2D データ
2D データ + タイムフィルビット(任意)	2D データ	2D データ
EOP(1D EOL x 6)	EOP(1D EOL x 6)	EOFB(EOL x 2) (注)
0ビットフィル	0ビットフィル	0ビットフィル

注意

- EOFB=End Of Facsimile Block
- 1Dデータ, 2Dデータのコード化方法はすべての方式で共通です。

5. イメージのフォーマットについて

イメージのフォーマットはImage Encoding Parameterの RECRDIDにより (182 ページ参照) 次の2つの指定が可能です。

RECRDID = X'01' RIDIC フォーマット
 = X'03' Bottom-to-Top フォーマット

RIDICについて

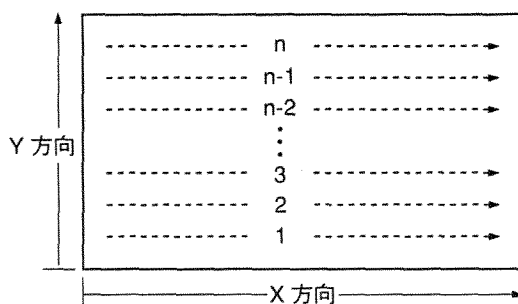
RIDICは (Recorded Image Data Inline Coding) の略で、イメージの作成方法は次のように左から右に (X方向) また上から下に (Y方向) です。



またそれぞれの行のビット数は8の整数倍である必要があります。
 8の倍数に足りないときは0のビットを追加して下さい。

Bottom-to-Topについて

Bottom-to-Topでのイメージの作成方法は次のように左から右に (X方向) また下から上に (Y方向) です。



またそれぞれの行のビット数は32の整数倍である必要があります。
 32の倍数に足りないときは0のビットを追加して下さい。