

ファイル調査1□Format-Hexの使い方メモ

2025-09-10
自分用メモ

これは何？

PowerShellのFormat-Hexコマンドレットの使い方の簡単な例。なのでスクリプト組んでエラー処理、とかは考慮しない。

バイナリデータの確認で「データがでかすぎてテキストエディタで開けない」とあったのでPowerShellのFormat-Hexを使ってみては？と話す。

回答は「これでどう確認すればいいかわからない」と言われてしまったのでやり方をサンプルで残す。

項目NAMEを文字列にする方法は[ファイル調査2：バイト配列から文字列に変換](#)を参照。

EBCDICのコードと文字の対応表は[EBCDICコード表](#)を参照。

サンプルデータファイル

事前に確認用のサンプルデータファイル SAMPLEBIN を作る。
レコードフォーマットは固定長で、COBOLイメージだとこんな感じ。

```
DATA DIVISION.  
    WORKING-STORAGE SECTION.  
        01 DATARECORD.  
            03 ID          PIC 9(4).  
            03 NAME       PIC X(8).  
            03 TAG        PIC 9(2).
```

レコード長14バイトで3レコードを格納する。つまりファイルサイズは42バイトになる。
そしてこのサンプルデータファイルはH社のホストコンピュータで作られたファイルとしよう。

```
E:\WK>dir SAMPLEBIN  
ドライブ E のボリューム ラベルは Backup です  
ボリューム シリアル番号は 6000-4D89 です  
  
E:\WK のディレクトリ  
  
2025/09/10  20:20                42 SAMPLEBIN  
1個のファイル                42 バイト  
0個のディレクトリ 878,738,391,040 バイトの空き領域  
  
E:\WK>
```

項目TAGの値の調査をする

プログラムが異常終了したと連絡があり「項目TAGの処理で問題が起きたと説明された」とする。

項目TAGはレコード先頭から13バイト目に2バイトで格納される数値項目と分かるので、この内容が正しいか確認する事になる。

また、3レコード有るので、各レコードに問題がないか確認していくことになる。

つまり項目TAGは

- 1レコード目：0バイト目から13バイト目のなかで、12～13バイト目
- 2レコード目：14バイト目から27バイト目のなかで、26～27バイト目
- 3レコード目：28バイト目から41バイト目のなかで、40～41バイト目

となる。

レコード	バイト範囲	値	表現
1	12□13	F0 F0	00
2	26□27	C2 C1	BA
3	40□41	E9 E9	ZZ

Loc	Disp													
	+00	+01	+02	+03	+04	+05	+06	+07	+08	+09	+0A	+0B	+0C	+0D
1レコード目	0x0000		ID						NAME					TAG
2レコード目	0x000E		ID						NAME					TAG
3レコード目	0x001C		ID						NAME					TAG

項目TAGの位置は

- 1レコード目 0x0000+0C = 0x000C
- 2レコード目 0x000E+0C = 0x001A
- 3レコード目 0x001C+0C = 0x0028

PowerShell 7.5.xの場合

クライアントPCだとPowerShell 7.5.x になっていることが多いのではないだろうかと思う。この場合Format-Hexコマンドレットは必要なオプションを備えているので面倒が少ない。

Format-Hexの-OffsetオプションでHexダンプ(16進数ダンプ)の表示開始位置を指定し、-Countオプションで、表示するバイト数を指定する。

レコード長は14バイトなので、レコード毎の表示は -Offset オプション □ -count オプションで

```
-Offset ( レコード □ 1 ) × 14 -Count 14
```

の値を指定すればよい。

つまり

- 1レコード目：-Offset 0 -Count 14 → 14バイト表示、項目TAGは 12～13バイト目
- 2レコード目：-Offset 14 -Count 14 → 14バイト表示、項目TAGは 12～13バイト目
- 3レコード目：-Offset 28 -Count 14 → 14バイト表示、項目TAGは 12～13バイト目

となる。

```
PS E:\WK> $PSVersionTable
```

```
Name                Value
----                -
PSVersion           7.5.2   ←面倒じゃないバージョンだ！
PSEdition           Core
GitCommitId        7.5.2
OS                  Microsoft Windows 10.0.22631
Platform            Win32NT
PSCompatibleVersions {1.0, 2.0, 3.0, 4.0...}
PSRemotingProtocolVersion 2.3
SerializationVersion 1.1.0.1
WSManStackVersion  3.0
```

```
PS E:\WK> Format-Hex -Path SAMPLEBIN -Offset 0 -Count 14 ← 1レコード目の表示指定
```

```
Label: E:\WK\SAMPLEBIN
```

```
Offset Bytes                Ascii
      00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
-----
000000000000000000 F0 F0 F1 F0 8E C4 8C A2 81 40 81 40 F0 F0
øðñð0Ä0ç0@0@ðð ☐← 項目TAGの値は 0xF0,0xF0 EBCDICで"00"の意
```

```
PS E:\WK> Format-Hex -Path SAMPLEBIN -Offset 14 -Count 14 ← 2レコード目の表示指定
```

```
Label: E:\WK\SAMPLEBIN
```

```
Offset Bytes                Ascii
      00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
-----
00000000000000000E F0 F0 F2 F0 8E C4 8C A2 81 40 81 40 C2 C1
øððð0Ä0ç0@0@ÄÄ ☐← 項目TAGの値は 0xC2,0xC1 EBCDICで"BA"の意
```

```
PS E:\WK> Format-Hex -Path SAMPLEBIN -Offset 28 -Count 14 ← 3レコード目の表示指定
```

```
Label: E:\WK\SAMPLEBIN
```

```
Offset Bytes                Ascii
      00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
-----
00000000000000001C F0 F0 F3 F0 93 79 8D B2 8C A2 81 40 E9 E9
øððð0y0²0ç0@éé ☐← 項目TAGの値は 0xE9,0xE9 EBCDICで"ZZ"の意
```

```
PS E:\WK>
```

2レコード目、3レコード目の項目TAGは数字のみで構成されていないのでエラーの原因はこの2レコードでしょう。

PowerShell 5.1.xの場合

サーバだとこのバージョンになっている物も多いんじゃないかなと思う。この場合Format-Hexコマンドレットだけでは対応できないので工夫する必要がある。

ファイル内容をバイト配列として保存し、配列の範囲を Format-Hex へ与えてHexダンプ(16進数ダンプ)させる。配列の開始インデクス、終了インデクスを与える必要がある。

つまり項目TAGは

- 1レコード目：配列要素 0番～13番で14バイト表示、項目TAGは 12～13番
- 2レコード目：配列要素 14番～27番で14バイト表示、項目TAGは 26～27番
- 3レコード目：配列要素 28番～41番で14バイト表示、項目TAGは 40～41番

となる。

```
PS E:\WK> $PSVersionTable
```

Name	Value
-----	-----
PSVersion	5.1.22621.5697 □← アー面倒なバージョンだ...
PSEdition	Desktop
PSCompatibleVersions	{1.0, 2.0, 3.0, 4.0...}
BuildVersion	10.0.22621.5697
CLRVersion	4.0.30319.42000
WSManStackVersion	3.0
PSRemotingProtocolVersion	2.3
SerializationVersion	1.1.0.1

```
PS E:\WK> $bindata = Get-Content -Path SAMPLEBIN -Raw -Encoding Byte □← ファイルをバイト型配列としてメモリに確保する。
```

```
PS E:\WK> $bindata[0..13] | Format-Hex □← 1レコード目の表示指定
```

パス:

```
00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
```

```
00000000 F0 F0 F1 F0 8E C4 8C A2 81 40 81 40 F0 F0           ððððÄç@@ðð □← 項目TAGの値は 0xF0,0xF0 EBCDICで"00"の意
```

```
PS E:\WK> $bindata[14..27] | Format-Hex □← 2レコード目の表示指定
```

パス:

```
00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
```

```
00000000 F0 F0 F2 F0 8E C4 8C A2 81 40 81 40 C2 C1           ððððÄç@@ÁÁ □← 項目TAGの値は 0xC2,0xC1 EBCDICで"BA"の意
```

```
PS E:\WK> $bindata[28..41] | Format-Hex ← 3レコード目の表示指定
```

パス:

```
00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
```

```
00000000 F0 F0 F3 F0 93 79 8D B2 8C A2 81 40 E9 E9           ðððy²ç@ée ← 項  
目TAGの値は 0xE9,0xE9 EBCDICで"ZZ"の意
```

```
PS E:\WK>
```

2レコード目、3レコード目の項目TAGは数字のみで構成されていないのでエラーの原因はこの2レコードでしょう。

[技術資料](#), [Windows](#), [PowerShell](#), [Format-Hex](#)

From:

<https://wiki.hgotoh.jp/> - 努力したWiki

Permanent link:

<https://wiki.hgotoh.jp/documents/windows/powershell/powershell-004>

Last update: **2026/03/10 12:06**

