

COBOLとPackedDecimal形式

2024-11-04

ZONEの説明を書いたのでPackedDecimalも作成

GnuCOBOL

「そういやCOBOL使えたはずだな」とローカルのサーバで確認してみる。

```
$ cbc -v
cbc (GnuCOBOL) 3.2.0
Built      Oct 10 2024 06:43:09  Packaged  Jul 28 2023 17:02:56 UTC
C version  "FreeBSD Clang 18.1.5 (https://github.com/llvm/llvm-project.git
llvmorg-18.1.5-0-g617a15a9eac9)"
loading standard configuration file 'default.conf'
cbc: error: no input files
$
```

FreeBSD上にインストールされたGnuCOBOLを使ってみます。

PackedDecimal形式をファイルに書き出す

ググりながらコードを書く。そらで一から書くのもう無理。
行番号は入れてません。

実行するとファイル sample.bin にPackedDecimal形式の値を書き出します。

ソースコード	実行結果
<pre> gendata.cbl ***** sample **** IDENTIFICATION DIVISION. PROGRAM-ID. SIMPLE-GENERATE-PACK-01. * ENVIRONMENT DIVISION. INPUT-OUTPUT SECTION. FILE-CONTROL. SELECT BIN-FILE ASSIGN TO "sample.bin". DATA DIVISION. FILE SECTION. FD BIN-FILE. 01 BIN-REC. 03 BIN-REC-P PIC S9(4) COMP-3. 03 BIN-REC00 PIC S9(4) COMP-3. 03 BIN-REC01 PIC S9(4) COMP-3. 03 BIN-REC02 PIC S9(4) COMP-3. 03 BIN-REC03 PIC S9(4) COMP-3. 03 BIN-REC04 PIC S9(4) COMP-3. 03 BIN-REC05 PIC S9(4) COMP-3. 03 BIN-REC06 PIC S9(4) COMP-3. 03 BIN-REC07 PIC S9(4) COMP-3. 03 BIN-REC08 PIC S9(4) COMP-3. 03 BIN-REC09 PIC S9(4) COMP-3. WORKING-STORAGE SECTION. 01 VAL0P PIC S9(4) COMP-3 VALUE 1234. 01 VAL00 PIC S9(4) COMP-3 VALUE -1230. 01 VAL01 PIC S9(4) COMP-3 VALUE -1231. 01 VAL02 PIC S9(4) COMP-3 VALUE -1232. 01 VAL03 PIC S9(4) COMP-3 VALUE -1233. 01 VAL04 PIC S9(4) COMP-3 VALUE -1234. 01 VAL05 PIC S9(4) COMP-3 VALUE -1235. 01 VAL06 PIC S9(4) COMP-3 VALUE -1236. 01 VAL07 PIC S9(4) COMP-3 VALUE -1237. 01 VAL08 PIC S9(4) COMP-3 VALUE -1238. 01 VAL09 PIC S9(4) COMP-3 VALUE -1239. PROCEDURE DIVISION. MAIN SECTION. MOVE VAL0P TO BIN-REC-P. MOVE VAL00 TO BIN-REC00. MOVE VAL01 TO BIN-REC01. MOVE VAL02 TO BIN-REC02. MOVE VAL03 TO BIN-REC03. MOVE VAL04 TO BIN-REC04. MOVE VAL05 TO BIN-REC05. MOVE VAL06 TO BIN-REC06. MOVE VAL07 TO BIN-REC07. MOVE VAL08 TO BIN-REC08. MOVE VAL09 TO BIN-REC09. OPEN OUTPUT BIN-FILE. WRITE BIN-REC. CLOSE BIN-FILE. STOP RUN. </pre>	<pre> GnuCOBOLでコンパイルして実行してみます。警告が出ているけどとりあえず後回し。 \$ cobc -x gendata.cbl /tmp/cob62814_0.c:321:35: warning: illegal character encoding in string literal [-Winvalid-source-encoding] 321 module->module_formatted_date = COB_MODULE_FORMATTED_DATE; ^~~~~~ /tmp/cob62814_0.c:15:39: note: expanded from macro 'COB_MODULE_FORMATTED_DATE' 15 #define COB_MODULE_FORMATTED_DATE "11<B7><EE> 04 2024 09:51:49" ^~~~~~ 1 warning generated. \$./gendata \$ hd sample.bin 00000000 01 23 4c 01 23 0d 01 23 1d 01 23 2d 01 23 3d 01 .#L.#.##.##-.#=#. 00000010 23 4d 01 23 5d 01 23 6d 01 23 7d 01 23 8d 01 23 #M.#].#m.#}.##.## 00000020 9d 00000021 \$ </pre>

hdコマンドで sample.bin をhexダンプした結果を見ます。

この環境下ではGnuCOBOLで扱っている負値のPackedDecimal形式の最後バイトの下位4ビットは0xDになっている事を確認できました。

数値 2024 は 0x02, 0x02, 0x4C となり、数値 -2024 は 0x02, 0x02, 0x4D となります。 [こんな形式](#)です。

PIC X(3)を PIC S9(4) COMP-3 にMOVEできるか

PIC X(3)の値を PIC S9(4) COMP-3にMOVEしたらどうなるかを見てみます。

*PackdDecimal形式表現 "0x01,0x23,0x4D" が -2024 として扱えるのかを見てみます

ソースコード	実行結果
<pre>sample.cbl \$ vi sample.cbl **** sample **** IDENTIFICATION DIVISION. PROGRAM-ID. SIMPLE-PACK-02. * DATA DIVISION. WORKING-STORAGE SECTION. 01 VAL01 PIC X(3) VALUE X"02024C". 01 VAL02 PIC X(3) VALUE X"02024D". 01 VAL03 PIC S9(4) COMP-3. 01 VAL04 PIC S9(4) COMP-3. PROCEDURE DIVISION. MAIN SECTION. MOVE VAL01 TO VAL03. MOVE VAL02 TO VAL04. DISPLAY "01: [" VAL01 "]" DISPLAY "02: [" VAL02 "]" DISPLAY "03: " VAL03. DISPLAY "04: " VAL04. STOP RUN.</pre>	<pre>これは予想通り。素直に内容を転記する事はできないようです。 \$ cobc -x sample.cbl /tmp/cob63029_0.c:212:35: warning: illegal character encoding in string literal [-Winvalid- source-encoding] 212 module->module_formatted_date = COB_MODULE_FORMATTED_DATE; ^~~~~~ /tmp/cob63029_0.c:15:39: note: expanded from macro 'COB_MODULE_FORMATTED_DATE' 15 #define COB_MODULE_FORMATTED_DATE "11<B7><EE> 04 2024 10:10:25" ^~~~~~ 1 warning generated. \$./sample 01: [L] ← 0x4cが'L' 02: [M] ← 0x4dが'M' 03: +0000 何か変換されてる 04: +0000 何か変換されてる \$</pre>

PackedDecimalもZONEと同様にREDEFINE句を使って PIC X(3)とPIC S9(4) COMP-3を重ねてあげればよいようです。

ソースコード	実行結果
<pre> sample2.cbl ***** sample **** IDENTIFICATION DIVISION. PROGRAM-ID. SIMPLE-PACK-02. * DATA DIVISION. WORKING-STORAGE SECTION. 01 VAL01 PIC X(3) VALUE X"02024C". 01 VAL02 PIC X(3) VALUE X"02024D". 01 VAL03 PIC S9(4) COMP-3. 01 FILLER REDEFINES VAL03. 03 VAL05 PIC X(3). 01 VAL04 PIC S9(4) COMP-3. 01 FILLER REDEFINES VAL04. 03 VAL06 PIC X(3). PROCEDURE DIVISION. MAIN SECTION. MOVE VAL01 TO VAL05. MOVE VAL02 TO VAL06. DISPLAY "01: [" VAL01 "]"". DISPLAY "02: [" VAL02 "]"". DISPLAY "03: " VAL03. DISPLAY "04: " VAL04. STOP RUN. </pre>	<pre> 望みの結果となりました。 \$ cobc -x sample.cbl /tmp/cob63116_0.c:217:35: warning: illegal character encoding in string literal [-Winvalid- source-encoding] 217 module->module_formatted_date = COB_MODULE_FORMATTED_DATE; ^~~~~~ /tmp/cob63116_0.c:15:39: note: expanded from macro 'COB_MODULE_FORMATTED_DATE' 15 #define COB_MODULE_FORMATTED_DATE "11<B7><EE> 04 2024 10:19:40" ^~~~~~ 1 warning generated. \$./sample 01: [L] 02: [M] 03: +2024 予想通り 04: -2024 予想通り \$ </pre>

技術資料, [COBOL](#), [ZONE形式](#), [REDEFINE](#)

From:

<https://wiki.hgotoh.jp/> - 努力したWiki

Permanent link:

<https://wiki.hgotoh.jp/documents/cobol/cobol-002>

Last update: 2024/11/09 20:53



