

# COBOLとZONE形式

2024-10-30

PIC X(4) → PIC S9(4) を行ったらどうなる？を確認する

## 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。

```
$ ccbc -v
ccbc (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'
ccbc: error: no input files
$
```

今回はFreeBSD上にインストールされたGnuCOBOLを使ってみます。

## ZONE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ZONE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-ZONE-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). 03 BIN-REC00 PIC S9(4). 03 BIN-REC01 PIC S9(4). 03 BIN-REC02 PIC S9(4). 03 BIN-REC03 PIC S9(4). 03 BIN-REC04 PIC S9(4). 03 BIN-REC05 PIC S9(4). 03 BIN-REC06 PIC S9(4). 03 BIN-REC07 PIC S9(4). 03 BIN-REC08 PIC S9(4). 03 BIN-REC09 PIC S9(4).  WORKING-STORAGE SECTION. 01 VAL0P PIC S9(4) VALUE 1234. 01 VAL00 PIC S9(4) VALUE -1230. 01 VAL01 PIC S9(4) VALUE -1231. 01 VAL02 PIC S9(4) VALUE -1232. 01 VAL03 PIC S9(4) VALUE -1233. 01 VAL04 PIC S9(4) VALUE -1234. 01 VAL05 PIC S9(4) VALUE -1235. 01 VAL06 PIC S9(4) VALUE -1236. 01 VAL07 PIC S9(4) VALUE -1237. 01 VAL08 PIC S9(4) VALUE -1238. 01 VAL09 PIC S9(4) 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/cob79343_0.c:321:35: warning: illegal character encoding in string literal [-Winvalid-source-encoding]     321 module-&gt;module_formatted_date =       COB_MODULE_FORMATTED_DATE;       ^~~~~~ /tmp/cob79343_0.c:15:39: note: expanded from macro 'COB_MODULE_FORMATTED_DATE'     15 #define COB_MODULE_FORMATTED_DATE "10&lt;B7&gt;&lt;EE&gt; 30 2024 14:09:46"       ^~~~~~ 1 warning generated. \$ ./gendata \$ hd sample.bin 00000000 31 32 33 34 31 32 33 70 31 32 33 71 31 32 33 72  1234123p123q123r  00000010 31 32 33 73 31 32 33 74 31 32 33 75 31 32 33 76  123s123t123u123v  00000020 31 32 33 77 31 32 33 78 31 32 33 79  123w123x123y  0000002c \$ </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ZONE形式の最後桁の文字は 0x700x79になっている  
 事を確認できました。  
 数値 2024 は 0x32, 0x30, 0x32, 0x34 となり、数値 -2024 は 0x32, 0x30, 0x32, 0x74 となります。 **こ  
 な**形式です。

## PIC X(4)を PIC S9(4) にMOVEできるか

PIC X(4)の値を PIC S9(4) にMOVEしたらどうなるかを見てください。

※ZONE形式表現 "202t" が -2024 として扱えるのかを見てください

| ソースコード  | 実行結果   |
|---|--|
| <pre> sample.cbl ***** sample ***** IDENTIFICATION DIVISION. PROGRAM-ID. SIMPLE-MOVE-01. * DATA DIVISION. WORKING-STORAGE SECTION. 01 VAL01 PIC X(4) VALUE "2024". 01 VAL02 PIC X(4) VALUE "202t". 01 VAL03 PIC S9(4). 01 VAL04 PIC S9(4). 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> どうも型変換的な処理が行われていますね。素直に内容を転記する 事はできないように見えます。 \$ cobc -x sample.cbl /tmp/cob79487_0.c:212:35: warning: illegal character encoding in string literal [-Winvalid- source-encoding] 212 module-&gt;module_formatted_date = COB_MODULE_FORMATTED_DATE; ^~~~~~ /tmp/cob79487_0.c:15:39: note: expanded from macro 'COB_MODULE_FORMATTED_DATE' 15 #define COB_MODULE_FORMATTED_DATE "10&lt;B7&gt;&lt;EE&gt; 30 2024 14:23:17" ^~~~~~ 1 warning generated. \$ ./sample 01:[2024] 02:[202t] ← -2024 のZONE形式表現の文字列 03: +2024 04: +0202 -2024を期待してたけど... \$ </pre> |

REDEFINE句を使って PIC X(4)とPIC S9(4)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- SAMPLE-02.       *       DATA DIVISION.       WORKING-STORAGE SECTION.       01 VAL01 PIC X(4) VALUE "2024".       01 VAL02 PIC X(4) VALUE "202t".       01 VAL03 PIC S9(4).       01 VAL04 PIC S9(4).       01 FILLER REDEFINES VAL04.       03 VAL05 PIC X(4).        PROCEDURE DIVISION.       MAIN SECTION.        MOVE VAL01 TO VAL03.       MOVE VAL02 TO VAL05.        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 sample2.cbl /tmp/cob79918_0.c:213:35: warning: illegal character encoding in string literal [-Winvalid- source-encoding]     213     module-&gt;module_formatted_date = COB_MODULE_FORMATTED_DATE; ^~~~~~ /tmp/cob79918_0.c:15:39: note: expanded from macro 'COB_MODULE_FORMATTED_DATE'     15 #define COB_MODULE_FORMATTED_DATE "10&lt;B7&gt;&lt;EE&gt; 30 2024 15:06:08" ^~~~~~ 1 warning generated. \$ ./sample2 01:[2024] 02:[202t] 03: +2024 04: -2024  -2024になりました \$ </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-001>

Last update: **2024/11/09 20:55**

