



Legac-E Education

## Unload a PDS(E) using COBOL

This program, produced in November 2013 on a z/OS V1R13 system, illustrates how a COBOL programmer could produce a sequential copy of a Partitioned Data Set which contains information such as; Job Control Language, source code, or parameter data. This technique will not work for program load libraries.

Note that the program is Link Edited with AMODE(24).

```
CBL DATA(24)
IDENTIFICATION DIVISION.
PROGRAM-ID.                PDSULOAD
AUTHOR.                    T.R.SAMBROOKS.
    INSTALLATION.
    DATE-WRITTEN.          17TH NOV 2013.
ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
    SKIP1
*-----*
*  Routine to unload a PDS(E) into a suitable form for input  *
*  to IEBUPDTE. This is not appropriate for Load Libraries.  *
*-----*
    SKIP3
INPUT-OUTPUT SECTION.
FILE-CONTROL.
    SELECT IEB-SYSIN          ASSIGN UT-S-SYSIN.
    SELECT SEQ-MEM-I         ASSIGN UT-S-SYSUT2.
    SELECT SEQ-MEM-O         ASSIGN UT-S-SEQMEMSO.
DATA DIVISION.
FILE SECTION.
FD  IEB-SYSIN                RECORDING MODE IS F
                                BLOCK CONTAINS 0 RECORDS
                                RECORD CONTAINS 80 CHARACTERS
                                DATA RECORD IS CNTL-STMT.
01  CNTL-STMT                PIC X(80).
FD  SEQ-MEM-I                RECORDING MODE IS F
                                BLOCK CONTAINS 0 RECORDS
                                RECORD CONTAINS 81 CHARACTERS
                                DATA RECORD IS FROM-IEBPTPCH.
01  FROM-IEBPTPCH            PIC X(81).
FD  SEQ-MEM-O                RECORDING MODE IS F
                                BLOCK CONTAINS 0 RECORDS
                                RECORD CONTAINS 80 CHARACTERS
                                DATA RECORD IS IEB-PUNCH.
01  IEBUPDTE-REC            PIC X(80).
    EJECT
```



Legac-E Education

## Unload a PDS(E) using COBOL

```
WORKING-STORAGE SECTION.
01  ADHOC-FIELDS.
    03  WS-CNT                PIC S9(4) COMP VALUE +0.
    03  WS-DPOS               PIC S9(4) COMP VALUE +1.
    03  EOF-FLAG              PIC X VALUE 'R'.
01  IEBPTPCH-CONTROL.
    03  OPT-LIST              PIC S9(4) COMP VALUE +0.
    03  DDN-LIST              PIC S9(4) COMP VALUE +0.
    03  IEBPTPCH              PIC X(8) VALUE 'IEBPTPCH'.
    03  CNTL-1                PIC X(80) VALUE
        ' PUNCH TYPORG=PO,MAXFLDS=1 '.
    03  CNTL-2                PIC X(80) VALUE
        ' RECORD FIELD=(80) '.
01  IEBUPDTE-CONTROL.
    03                          PIC X(12) VALUE './ ADD NAME='.
    03  N-NAME                 PIC X(8)  VALUE SPACE.
    03                          PIC X(60) VALUE SPACE.
01  PUNCHED-RECORD.
    03  CNTL-CHAR              PIC X.
    03  REAL-DATA-1.
        05  SEP-REC            PIC X(13).
        05  M-NAME             PIC X(8).
        05                          PIC X(59).
    03  REAL-DATA-2            REDEFINES REAL-DATA-1.
        05                          PIC X(80).

PROCEDURE DIVISION.
    PERFORM UNLOAD-MEMBERS.
    OPEN INPUT SEQ-MEM-I, OUTPUT SEQ-MEM-O.
*-----*
*   Read each IEBPTPCH record in turn, checking to see if it   *
*   is a member separator, and if it is convert it to an      *
*   IEBUPDTE ./ ADD statement.                                *
*-----*
    PERFORM                      UNTIL EOF-FLAG = 'D'
        READ SEQ-MEM-I            INTO PUNCHED-RECORD
                                AT END MOVE 'D' TO EOF-FLAG
                                END-READ

    IF EOF-FLAG = 'R'
        IF SEP-REC = 'MEMBER NAME '
                                MOVE M-NAME TO N-NAME
                                WRITE IEBUPDTE-REC FROM
                                    IEBUPDTE-CONTROL
                                ELSE WRITE IEBUPDTE-REC FROM REAL-DATA-2
                                END-IF
    END-IF
END-PERFORM.
```



Legac-E Education

## Unload a PDS(E) using COBOL

```
*-----*
* The logical end of the source code program - PDSULOAD      *
*-----*
      CLOSE                      SEQ-MEM-I, SEQ-MEM-O.
      GOBACK.
UNLOAD-MEMBERS.
      OPEN  OUTPUT IEB-SYSIN.
      WRITE CNTL-STMT           FROM CNTL-1.
      WRITE CNTL-STMT           FROM CNTL-2.
      CLOSE IEB-SYSIN.
      CALL IEBPTPCH              USING OPT-LIST, DDN-LIST.
      IF RETURN-CODE > 0
          DISPLAY 'IEBPTPCH RETURN CODE =' RETURN-CODE.
ENLOAD-MEMBERS-EXIT.
      EXIT.
*-----*
* The physical end of the source code program - PDSULOAD      *
*-----*
```

## Invoking JCL

```
//JOBNAMEC JOB CLASS=A,MSGCLASS=X,NOTIFY=&SYSUID
//S0010 EXEC PGM=PDSULOAD,
//          PARM='ALL31(OFF),HEAP(, ,BELOW)/'
//SYSPRINT DD DUMMY          Unwanted IEBPTPCH messages
//SYSIN DD UNIT=3390,LRECL=80,RECFM=FB,BLKSIZE=0,
//          SPACE=(80,(10,10)),AVGREC=U
//SYSUT1 DD DISP=SHR,DSN=name_of_library_to_be_unloaded
//SYSUT2 DD UNIT=3390,LRECL=81,RECFM=FB,BLKSIZE=0,
//          SPACE=(81,(500,500)),AVGREC=U
//SEQMEMSO DD UNIT=3390,LRECL=80,RECFM=FB,BLKSIZE=0,
//          SPACE=(80,(500,500)),AVGREC=U,DISP=(,CATLG),
//          DSN=name_of_unloaded_version
```