



Legac-E Education

## **Loading an RRDS File**

---

Although there had been previous examples of loading a Relative Record Data Set (RRDS) this version was produced in 2017 in support of additional exercises developed for a CICS Command Level programming course.

The program uses an input file containing 80-byte images in which the fields are blank separated. It converts these records to 66 byte records by removing the blanks and the first 6-byte field of the input.

The first 6-bytes of the input do not form part of the output record as the first character indicates which of two files the output record is to be placed in, and the remaining 5 bytes are the output record number.

Included with the program listing which follows is the copy book member which describes the input record, and the JCL associated with defining the cluster and executing the load program.

The program was updated in July 2018 to exploit dynamic allocation for the two output RRDS Clusters. BPXWDYN is used to perform the dynamic allocation.



Legac-E Education

## Loading an RRDS File

---

```
CBL OFFSET
IDENTIFICATION DIVISION.
PROGRAM-ID.                RRDSLOAD.
AUTHOR.                    T.R.SAMBROOKS.
    INSTALLATION.
    DATE-WRITTEN.          22nd APR 2017.
ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
*-----*
*   Sample program to demonstrate an alternative approach to *
*   Exercise 15.1.                                           *
*-----*
INPUT-OUTPUT SECTION.
FILE-CONTROL.
    SELECT DSNS-IN          ASSIGN TO UT-S-RRDSDSNS.
    SELECT MODELS-IN       ASSIGN TO UT-S-MODELSIN.
    SELECT RRDS-DIESEL     ASSIGN TO NUMBERED-DIESEL
                            ORGANIZATION IS RELATIVE
                            ACCESS IS DYNAMIC
                            RELATIVE KEY IS WS-LOCO-NO
                            FILE STATUS      IS DIESEL-CODE
                                    VSAM-CODE.
    SELECT RRDS-STEAM      ASSIGN TO NUMBERED-STEAM
                            ORGANIZATION IS RELATIVE
                            ACCESS IS DYNAMIC
                            RELATIVE KEY IS WS-LOCO-NO
                            FILE STATUS      IS STEAM-CODE
                                    VSAM-CODE.
DATA DIVISION.
FILE SECTION.
FD  DSNS-IN                RECORDING MODE IS F
                            LABEL RECORDS ARE STANDARD
                            BLOCK CONTAINS 0 RECORDS
                            RECORD CONTAINS 80 CHARACTERS
                            DATA RECORD IS DSN-REC.
01  DSN-REC
FD  MODELS-IN              RECORDING MODE IS F
                            LABEL RECORDS ARE STANDARD
                            BLOCK CONTAINS 0 RECORDS
                            RECORD CONTAINS 80 CHARACTERS
                            DATA RECORD IS MI-RECORD.
01  MI-RECORD.
    COPY F100REC.
```



Legac-E Education

## Loading an RRDS File

---

```
FD  RRDS-DIESEL                RECORD CONTAINS 66 CHARACTERS
                                DATA RECORD IS FD-DIESEL.

01  FD-DIESEL.
    03  D-COY                    PIC X(10).
    03  D-CAT-NO                 PIC X(7).
    03  D-PRICE                  PIC 9(3)V99 COMP-3.
    03  D-LOCO-DES              PIC X(10).
    03  D-PWR-CLASS             PIC X(5).
    03  D-LOCO-NAME            PIC X(31).
FD  RRDS-STEAM                RECORD CONTAINS 66 CHARACTERS
                                DATA RECORD IS FD-STEAM.

01  FD-STEAM.
    03  S-COY                    PIC X(10).
    03  S-CAT-NO                 PIC X(7).
    03  S-PRICE                  PIC 9(3)V99 COMP-3.
    03  S-LOCO-DES              PIC X(10).
    03  S-PWR-CLASS             PIC X(5).
    03  S-LOCO-NAME            PIC X(31).
WORKING-STORAGE SECTION.
01  WS-ADHOC-CONSTANTS.
    03  WS-LOCO-NO              PIC 9(5) VALUE 1.
    03  LAST-LOCO-NO           PIC 9(5) VALUE 0.
    03  DIESEL-CODE            PIC 99.
    03  STEAM-CODE             PIC 99.
    03  VSAM-CODE.
        05  R15-RETURN          PIC 99 COMP.
        05  VSAM-FUNCTION        PIC 9 COMP.
        05  VSAM-FEEDBACK        PIC 999 COMP.
    03  WS-EOF-INDD            PIC X VALUE 'R'.
        88  ALL-DONE             VALUE 'D'.
        88  READ-AGAIN          VALUE 'R'.
    03  DYN-PGM                 PIC X(8) VALUE 'BPXWDYN '.
    03  DIESEL-DDN              PIC X(9) VALUE SPACES.
    03  STEAM-DDN               PIC X(9) VALUE SPACES.
01  ALLOC-DYNFILE.
    03                          PIC S9(4) COMP VALUE +80.
    03                          PIC X(9) VALUE
        'ALLOC FI('.
    03  DYN-DDNA                PIC X(9) VALUE SPACES.
    03                          PIC X(17) VALUE
        ' SHR MSG(WTP) DA('.
    03  DYN-DSN                 PIC X(45) VALUE SPACES.
```



Legac-E Education

## Loading an RRDS File

```
01 UNALLOC-DYNFILE.
03 PIC S9(4) COMP VALUE +27.
03 PIC X(8) VALUE
    'FREE FI(' .
03 DYN-DDNU PIC X(8) VALUE SPACES.
03 PIC X(2) VALUE ' ' .
03 PIC X(9) VALUE 'MSG(WTP) ' .
PROCEDURE DIVISION.
A010-MAIN-PGM.
    PERFORM B010-INITIALIZATION.
    PERFORM
        READ MODELS-IN
        UNTIL ALL-DONE
        AT END
            MOVE 'D' TO WS-EOF-INDD
        NOT AT END
            MOVE LOCO-NO TO WS-LOCO-NO
            IF LOCO-TYPE = 'D'
                PERFORM B020-LOAD-DIESEL
            ELSE
                PERFORM B030-LOAD-STEAM
            END-IF
    END-READ
    END-PERFORM.
A010-MAIN-PGM-END.
*-----*
* This is the logical end of program RRDSLOAD *
*-----*
GOBACK.
B010-INITIALIZATION.
    OPEN INPUT DSNS-IN.
    PERFORM 2 TIMES
        READ DSNS-IN AT END CLOSE DSNS-IN
        END-READ
        UNSTRING DSN-REC
            DELIMITED BY '='
            INTO DYN-DDNA, DYN-DSN
        END-UNSTRING
        INSPECT DYN-DDNA
            REPLACING FIRST ' ' BY ')'
        IF DYN-DDNA = 'DIESEL) '
            MOVE DYN-DDNA TO DIESEL-DDN
        ELSE
            MOVE DYN-DDNA TO STEAM-DDN
        END-IF
        INSPECT DYN-DSN
            REPLACING FIRST ' ' BY ')'
        CALL DYN-PGM
        USING ALLOC-DYNFILE
    END-PERFORM.
    OPEN INPUT MODELS-IN
        OUTPUT RRDS-DIESEL
        RRDS-STEAM.
B010-INITIALIZATION-EXIT.
EXIT.
```



Legac-E Education

## Loading an RRDS File

---

```
B020-LOAD-DIESEL.
*-----*
*   Load a DIESEL record into the DIESEL cluster.   *
*-----*
      MOVE FB-COY           TO D-COY.
      MOVE FB-CAT-NO        TO D-CAT-NO.
      MOVE FB-PRICE         TO D-PRICE.
      MOVE FB-LOCO-DES      TO D-LOCO-DES.
      MOVE FB-LOCO-POWER    TO D-PWR-CLASS.
      MOVE FB-LOCO-NAME     TO D-LOCO-NAME.
      WRITE FD-DIESEL.
B020-LOAD-DIESEL-EXIT.
      EXIT.
B030-LOAD-STEAM.
*-----*
*   LOAD a STEAM record into the STEAM cluster.   *
*-----*
      IF LOCO-NO > 64999          DISPLAY 'Last steam record is '
                                  LAST-LOCO-NO UPON SYSOUT
                                  MOVE 'D' TO WS-EOF-INDD
      ELSE
                                  MOVE LOCO-NO TO LAST-LOCO-NO
                                  MOVE FB-COY   TO S-COY
                                  MOVE FB-CAT-NO TO S-CAT-NO
                                  MOVE FB-PRICE TO S-PRICE
                                  MOVE FB-LOCO-DES TO S-LOCO-DES
                                  MOVE FB-LOCO-POWER TO S-PWR-CLASS
                                  MOVE FB-LOCO-NAME TO S-LOCO-NAME
                                  WRITE FD-STEAM
      END-IF.
B030-LOAD-STEAM-EXIT.
      EXIT.
B040-TERMINATION.
      CLOSE
                                  MODELS-IN
                                  RRDS-DIESEL
                                  RRDS-STEAM.
      MOVE DIESEL-DDN           TO DYN-DDNU.
      CALL DYN-PGM              USING UNALLOC-DYNFILE.
      MOVE STEAM-DDN           TO DYN-DDNU.
      CALL DYN-PGM              USING UNALLOC-DYNFILE.
B040-TERMINATION-EXIT.
      EXIT.
```



Legac-E Education

## Loading an RRDS File

---

```
C030-REPORT-STATUS.  
*-----*  
*   Report on VSAM File Status for de-bugging purposes   *  
*-----*  
*   DISPLAY 'RRDSLOAD -FILE STATUS -' FSTAT-CODE UPON SYSOUT.  
*   DISPLAY 'RRDSLOAD -RETURN CODE -' R15-RETURN UPON SYSOUT.  
*   DISPLAY 'RRDSLOAD -FUNCTION -' VSAM-FUNCTION UPON SYSOUT.  
*   DISPLAY 'RRDSLOAD -FEEDBACK -' VSAM-FEEDBACK UPON SYSOUT.  
C030-REPORT-STATUS-EXIT.  
EXIT.  
*-----*  
*   This is the physical end of program RRDSLOAD   *  
*-----*
```

## Input record Copy Book Member

```
03 F100-RECORD.  
05 FB-LOCO-KEY.  
    07 LOCO-TYPE      PIC X.  
    07 LOCO-NO       PIC 9(5).  
05                   PIC X.  
05 FB-COY           PIC X(10).  
05                   PIC X.  
05 FB-CAT-NO       PIC X(7).  
05                   PIC X.  
05 FB-PRICE        PIC 9(3)V99.  
05                   PIC X.  
05 FB-LOCO-DES     PIC X(10).  
05                   PIC X.  
05 FB-LOCO-POWER   PIC X(5).  
05                   PIC X.  
05 FB-LOCO-NAME    PIC X(31).
```



Legac-E Education

## Loading an RRDS File

---

### Invoking JCL

```
//          EXPORT SYMLIST=STU
//          SET   STU=&SYSUID
//S0010     EXEC PGM=IDCAMS
//SYSPRINT DD  SYSOUT=*
//INDD      DD  DSN=&&RRDS,DISP=(OLD,DELETE)
//SYSIN     DD  *,SYMBOLS=JCLONLY
DELETE &STU..RRDS.DIESEL
SET MAXCC = 0
DEFINE CLUSTER( NAME(&STU..RRDS.DIESEL) -
                VOLUMES(*) -
                RECORDS(15000) -
                RECORDSIZE(66 66) -
                NUMBERED ) -
        DATA( NAME(&STU..RRDS.DIESEL.DATA))
DELETE &STU..RRDS.STEAM
SET MAXCC = 0
DEFINE CLUSTER( NAME(&STU..RRDS.STEAM) -
                VOLUMES(*) -
                RECORDS(65000) -
                RECORDSIZE(66 66) -
                NUMBERED ) -
        DATA( NAME(&STU..RRDS.STEAM.DATA))
//*-----*
//*          WILL NOT LOAD MORE THAN 65000 RECORDS.          *
//*-----*
//          IF   RC = 0 THEN
//S0050     EXEC PGM=RRDSL0D
//STEPLIB  DD  DISP=SHR,DSN=&STU..LOAD.LIBRARY
//SYSOUT   DD  SYSOUT=*
//MODELSIN DD  DISP=SHR,DSN=&STU..CICS.COB(ENGINES)
//RRDSDSNS DD  *,SYMBOLS=JCLONLY
DIESEL=&STU..RRDS.DIESEL
STEAM=&STU..RRDS.STEAM
//          ENDIF
```