

NORTHERN ILLINOIS UNIVERSITY

Superwylbur Macro: D50TD80

A Report submitted to the
University Honors Program
in Partial Fulfillment of the
Requirements of the Baccalaureate Degree
with University Honors

Department of Computer Science

by

Karen M. Rowe

Dekalb, Illinois

May, 1989

JOURNAL OF THESIS ABSTRACTS

THESIS SUBMISSION FORM

AUTHOR: Karen Rowe

THESIS TITLE: Superwylbur Macro:D50TD80

THESIS ADVISOR: Dr. Lee Gaines

ADVISOR'S DEPT: Computer Science

DATE: 12/16/88

HONORS PROGRAM: University Honors Program

NAME OF COLLEGE: Northern Illinois University

PAGE LENGTH: 1

BIBLIOGRAPHY (YES OR NO): No

ILLUSTRATED (YES OR NO): No

COPIES AVAILABLE (HARD COPY, MICROFILM, DISKETTE): hard copy

IS YOUR THESIS OR ANY PART BEING SUBMITTED FOR PUBLICATION? YES ___ NO x
IF ANY PART HAS BEEN ACCEPTED FOR PUBLICATION, PLEASE INDICATE
WHERE _____

SUBJECT HEADINGS: (CHOOSE FIVE KEY WORDS) Macro, Superwylbur, programming,
Computer Science, computers

ABSTRACT (100-200 words):

see attachment

For Office Use:

THESIS NUMBER: _____

The purpose of this project was to write a Superwylbur macro, called D50TD80, for the Computing Information Center at Northern Illinois University. The project was completed using Superwylbur, an interactive computing environment which enables access to the University's mainframe computer. Superwylbur macro programming combines Superwylbur commands along with a set of special instructions for branching and decision making. The D50TD80 macro was designed to assist users in moving data sets from the current 3350 disk packs to the newly installed, more efficient, 3380 disk packs. This macro allows the user three options: 1) To create a partitioned data set from sequential data sets, and place on a 3380 disk pack, 2) To move an entire partitioned data set to a 3380 disk pack, and 3) To move a sequential data set to a 3380 disk pack. Options 2 and 3 are executed by calling modified versions of existing macros. Changes to these macros included additional error checking and more informative prompts. The first option begins by asking the user for the name and location of the partitioned data set that they wish to create. The user is next prompted for the names of the sequential data sets to be saved as members of the partitioned data set. The requested data sets are then placed into the specified PDS. The macro was coded to be user friendly by anticipating a variety of user errors, and making the prompts and error messages friendly yet informative. The method followed to complete the end product was basically a "hands on" approach. Using The Superwylbur Macro Programming Manual as a primary reference, a simple macro was written which can assist a user in uploading and downloading files from Superwylbur. After some practice with the language, the D50TD80 macro was coded. This macro will be available for public use in the Spring of 1989.

Approved: *Lee M. Gaines*

Department of: *Computer Science*

Date: *December 16, 1988*

The purpose of my project was to write a Superwylbur macro, called D50TD80, for the Computing Information Center here at Northern Illinois University. The project was completed using Superwylbur, an interactive computing environment which enables access to the University's mainframe computer. Superwylbur macro programming combines Superwylbur commands along with a set of special instructions for branching and decision making. The D50TD80 macro was designed to assist users in moving data sets from the current 3350 disk packs (ACA001 through ACA008) to the newly installed, more efficient, 3380 disk packs (ACA101 through ACA105). In order to better utilize this disk space, users are going to be encouraged to use partitioned data sets. A partitioned data set is a group of sequential data sets, each called a member, which uses the disk space more efficiently.

Creating a partitioned data set (PDS) from sequential data sets for a 3380 disk pack is just one of the options the D50TD80 macro allows. The macro also allows the user to move an entire partitioned data set to a 3380 disk pack, and, to move a sequential data set to a 3380 disk pack. These last two options are executed by calling modified versions of existing macros. Changes I made to these macros included additional error checking and more informative prompts. The first option begins by asking the user for the name and location of the partitioned data set that they wish to create. The user is next prompted for the names of the sequential data sets to be saved as members of the partitioned data set. The requested data sets are then placed

into the specified PDS. The macro was coded to be user friendly by anticipating a variety of user errors, and making the prompts and error messages friendly yet informative.

The method followed to complete the end product was basically a "hands on" approach. Using The Superwylbur Macro Programming Manual as a primary reference, I wrote a simple macro which can assist a user in uploading and downloading files from Superwylbur. After some practice with the language, the D50TD80 macro was coded. This macro will be available for public use in the Spring of 1989.

SUPERWYLBUR Macros

- I D50TD80 Macro
- II COPYSEQ Macro
- III CPDS Macro

I D50TD80 Macro


```

0. *****
10. *
20. * *****
30. * * D50TD80 MACRO *
40. * *****
50. *
60. * FUNCTION: THIS MACRO WILL MOVE DATA SETS FROM THE
70. * ***** 3350 DISK PACKS (I.E. ACA001-ACA008) TO
80. * THE NEW 3380 DISK PACKS (ACA101-ACA104).
90. *
100. * NAME: D50TD80 (WRITTEN BY KAREN M. ROWE)
110. * ****
120. *
130. * DEVICES: THIS MACRO CAN BE USED TO MOVE AN ENTIRE
140. * ***** PDS OR SEQUENTIAL DATASET. IT CAN ALSO BE
150. * USED TO CREATE A PDS OR TO ADD MEMBERS TO
160. * AN EXISTING PDS.
170. *
180. *
190. * NOTES:
200. * =====
210. * * D50TD80 CALLS TWO OTHER MACROS
220. * 1) MODIFIED VERSION OF CPDS
230. * 2) MODIFIED VERSION OF COPYSEQ
240. * * D50TD80 REQUIRES 3 TEMPORARY DATA SETS
250. * * D50TD80 EDITS ALL INPUT FROM USER
251. * * THE CHECK FOR VOLUMES ACA101 THRU ACA104
252. * WILL NEED TO BE MODIFIED IF OTHER 3380
253. * VOLUMES ARE ADDED
260. *
270. * *****
280.
290. SET LEN 150
291. LET TEMP A = .TEMPORARY
292. LET TEMP B = .TEMPORARY
293. LET TEMP C = .TEMPORARY
300. SET EXIT ERROR 1980
310. SET EXIT ATTN 2040
320. *
330. LET CONT = ' *** Press <RETURN> to continue *** '
340. IF (.T3270 = 1) THEN CLEAR SSA
350. DEMAND STRING INSTRUCTIONS 'Enter ''y'' for instructions or hit RETURN '
360. LET INSTRUCTIONS = .TRIM:(INSTRUCTIONS;" ";"B")
370. IF (.UPPER:(INSTRUCTIONS) = "Y") THEN GO TO 410
380. * ELSE
380.1 IF (.T3270 = 1) THEN CLEAR SSA
381. L 2120/2330 T %.MACRO UNN
390. T ''
400. DEMAND STR CONTIN %%CONT
410. *
420. IF (.T3270 = 1) THEN CLEAR SSA
460. LET TOTLNES = 0
470. LET MSG1 = 'Nothing entered. Re-enter.'
480. *
490. TYPE 'Options available by this macro'
500. TYPE ' 1) Create PDS from sequential members and place on 3380 disk.'
505. TYPE ' or add members to an existing PDS'
510. TYPE ' 2) Move entire PDS to 3380 disk'
520. TYPE ' 3) Move sequential data set to 3380 disk'
530.

```

```

540. DEMAND STRING ANSWER 'Choice (1, 2, or 3) : '
550. LET ANSWER = .TRIM:(ANSWER;' '; 'B')
560. IF (ANSWER = '') THEN TYPE MSG1
570.     THEN TYPE ' (Press BREAK or ESCAPE to stop macro execution)'
580.     THEN GO TO 530
590. IF (ANSWER = 1 .AND ANSWER = 2 .AND ANSWER = 3)
600.     THEN TYPE 'Invalid option entered. Re-Enter'
610.     THEN GO TO 530
620. IF (ANSWER = '2') CALL FROM &A1OKMR1.WYLIB:CPDS ON ACA001
630. IF (ANSWER = '3') CALL FROM &A1OKMR1.WYLIB:COPYSEQ ON ACA001
650. THEN GO TO 2060
660. *
670. TYPE 'Enter the name of a new PDS or an existing PDS into which'
680. TYPE 'you would like your sequential data sets placed '
690.
700. DEMAND STRING PDSNAME '>'
710. LET PDSNAME = .TRIM:(.UPPER:(%PDSNAME)' '; 'B')
720. IF (PDSNAME = '') THEN TYPE MSG1
730.     THEN GO TO 690
740. *
750.
760. TYPE 'Enter the volume where ' %PDSNAME 'resides or will reside, '
770. TYPE ' i.e. ACA101, ACA102, ACA103, or ACA104'
780. DEMAND STRING PDSVOL '>'
790. LET PDSVOL = .TRIM:(.UPPER:(%PDSVOL);' '; 'B')
800. IF(PDSVOL='') THEN TYPE MSG1
810.     THEN GO TO 780
820. COL END T %TEMPA NEW CMD SHO VOL S LIKE %PDSVOL
830. IF (.SIZE:(%TEMPA) .EQ 1) THEN CLEAR T %TEMPA
840.     THEN GO TO 900
850. * ELSE
860.     CLEAR T %TEMPA
870.     TYPE 'Invalid volume name entered. Re-enter volume'
880.     GO TO 780
890. *END IF
900.
901. IF (PDSVOL = 'ACA101' .AND PDSVOL = 'ACA102' .AND PDSVOL = 'ACA103' .AND PDSVOL = 'ACA104') TYPE '**ERROR'
902.     THEN TYPE 'PDS volume must be one of the following: ACA101, ACA102, ACA103, OR ACA104'
902.5.     THEN TYPE 'Please re-enter volume'
903.     THEN GO TO 780
904.
910. TYPE 'Does this PDS already exist (Type Y or N) ? '
920.
930. DEMAND STRING EXIST '>'
940. LET EXIST = .SUBSTRING:(.TRIM:(.UPPER:(%EXIST)' '; 'B');1;1)
950. IF (EXIST = '') THEN TYPE MSG1
960.     THEN GO TO 920
970. IF (EXIST = 'Y' .AND EXIST = 'N') THEN T 'Invalid response. Please enter ''Y'' or ''N''.'
980.     THEN GO TO 920
990. *
1000. IF (EXIST = 'N') THEN GO TO 1100
1010. *ELSE
1020. COL END T %TEMPB NEW CMD SHO DS NAMES LIKE %PDSNAME ON %PDSVOL
1030. IF (.SIZE:(%TEMPB) .EQ 2) THEN CLEAR T %TEMPB
1040.     THEN GO TO 1170
1050. * ELSE
1060.     CLEAR T %TEMPB
1070.     TYPE %PDSNAME ' does not exist on ' %PDSVOL ' Re-enter PDS name.'
1080.     GO TO 690
1090.

```

```

1100.
1120.     IF (EXIST= 'N') THEN TYPE 'Enter the primary space allocation (Optional)'
1130.     THEN DEMAND STRING PRIME '(An integer from 1 to 5) >'
1150.     THEN LET PRIME = .TRIM:(.UPPER:(%PRIME) ' '; 'B')
1160.     THEN IF (PRIME = '') THEN LET PRIME = '1'
1170.
1175.     LET FLAG = 'Y'
1180.     TYPE 'Enter first data set name to be added to PDS.'
1190.
1200.     DEMAND STRING NAME '>'
1210.     LET NAME = .TRIM:(.UPPER:(%NAME) ' '; 'B')
1220.     IF (FLAG = 'Y' .AND NAME = '')
1230.         THEN TYPE MSG1
1240.         THEN TYPE ' '
1250.         THEN GO TO 1190
1260.     *
1270.     IF (NAME .EQ '') THEN GO TO 2050
1280.     * ELSE
1290.     TYPE 'Enter the volume where ' %%name 'resides '
1300.
1310.     DEMAND STRING VOL '>'
1320.     LET VOL = .TRIM:(.UPPER:(%VOL); ' '; 'B')
1330.     IF (VOL='') THEN TYPE MSG1
1340.     THEN GO TO 1300
1350.     COL END T %TEMPA CLR NEW CMD SHO VOLS LIKE %VOL
1360.     IF (.SIZE:(%TEMPA) .EQ 1) THEN CLEAR T %TEMPA
1370.     THEN GO TO 1430
1380.     * ELSE
1390.     CLEAR T %TEMPA
1400.     TYPE 'Invalid volume name entered. Re-enter.'
1410.     GO TO 1300
1420.     *
1430.
1440.     COL END T %TEMPB NEW CMD SHO DSNAME LIKE %NAME ON %VOL
1450.     IF (.SIZE:(%TEMPB) .EQ 2) THEN CLEAR T %TEMPB
1460.     THEN GO TO 1520
1470.     * ELSE
1480.     CLEAR T %TEMPB
1490.     TYPE %%NAME ' does not exist on ' %%vol ' Re-enter data set.'
1500.     GO TO 1190
1510.     *
1520.
1530.     COL END T %TEMPA CLR NEW CMD SHOW DSNAME %NAME ON %VOL
1540.     LET LFIVE = .LINE:(5;%TEMPA;0)
1550.     LET COL1 = .INDEX:(%LFIVE; 'LRECL=')
1560.     LET RECLN = .SUBSTRING:(%LFIVE; COL1+6; 4)
1570.     LET RECLN = .TRIM:(%RECLN; ' '; 'B')
1580.     IF (FLAG = 'Y') THEN LET HOLDLREC = RECLN
1590.     THEN GO TO 1660
1600.     * ELSE
1610.     IF (RECLN ^= HOLDLREC) THEN TYPE '*** ERROR ***'
1620.     THEN TYPE 'LRECL of ' %%name '(' %%RECLN ')' does not match '
1630.     THEN TYPE 'Lrecl of previously entered data sets (' %%HOLDLREC ')'
1640.     THEN TYPE %%name 'CANNOT be a member in ' %%PDSNAME
1650.     THEN GO TO 1930
1660.
1670.     LET LTHREE = .LINE:(3;%TEMPA;0)
1680.     LET LENGTH = .INDEX:(%LTHREE)
1690.     LET COL1 = .INDEX:(%LTHREE; "(")
1720.     LET NEW = .SUBSTRING:(%LTHREE; COL1+1; LENGTH-COL1-1)

```

```

1730. LET LENGTH = .INDEX:(%%NEW)
1740. LET COL1 = .INDEX:(%%NEW;" ")
1750. LET NEW = .SUBSTRING:(%%NEW;1;COL1)
1760. IF (NEW .GT 3) THEN TYPE '*** Error ***'
1770. THEN TYPE %%NAME 'Is too large to be a member in your PDS'
1780. THEN GOTO 1930
1790. *
1801. IF (FLAG = 'Y') GO TO 1809.5
1802. COL END T %TEMPA NEW CMD SHO MEMS LIKE %NAME FROM %PDSNAME ON %PDSVOL
1803. IF (.SIZE:(%TEMPA) ^= 3) GO TO 1809
1803.5 CLEAR T %TEMPA
1804. TYPE %%NAME 'already exists as a member in ' %%PDSNAME
1805. TYPE 'Would you like to resave ' %%NAME 'into ' %%PDSNAME '?' (enter Y or N)'
1806. DEMAND STRING QUEST '>'
1807. LET QUEST = .TRIM:(.UPPER:(QUEST);' ','B')
1807.5 IF (QUEST ^= 'Y' .AND QUEST ^= 'N') THEN TYPE 'Please answer Y or N'
1807.6 THEN GO TO 1806
1808. IF (QUEST = 'N') THEN GO TO 1940
1808.01 LET FLAG2 = 'Y'
1808.1 GO TO 1809.5
1808.5
1809. CLEAR T %TEMPA
1809.5
1810. USE T %TEMPC FROM %NAME ON %VOL
1820. TYPE %%NAME 'is now being saved into ' %%PDSNAME
1830. IF (FLAG ^= 'Y' .OR EXIST ^= 'N') GO TO 1901
1831. LET LINE1 = 'SPACE=(TRK,(
1850. LET LINE1 = LINE1 %%PRIME ',0,1),RLSE)'
1851. IF (RECLN = '6233') LET LINE2 = ''
1852. THEN GO TO 1870
1853. * ELSE
1860. LET LINE2 = 'LRECL=' %%HOLDLREC
1861. * END IF
1870. LET FULLNAME = .TRIM:(PDSNAME;" ";"B") ':' %%NAME
1871. *
1872. *
1880. QT SAVE T %TEMPC INTO %FULLNAME ON %PDSVOL %LINE2 %LINE1
1890. GO TO 1930
1900. *ELSE
1901. IF (FLAG2 = 'Y') QT RESAVE T %TEMPC INTO %FULLNAME ON %PDSVOL
1903. THEN GO TO 1930
1905. * ELSE
1910. QT SAVE T %TEMPC INTO %PDSNAME:%NAME ON %PDSVOL
1920. *
1930. * ENDIF
1931. *ENDIF
1939. LET FLAG = 'N'
1939.5 LET FLAG2 = 'N'
1940. TYPE 'Enter name of additional data set, or hit RETURN'
1950. GO TO 1190
1960. *
1970.
1980. IF (.SIZE:(%TEMPA) .NE -1) THEN CLEAR T %TEMPA
1990. IF (.SIZE:(%TEMPB) .NE -1) THEN CLEAR T %TEMPB
1991. IF (.SIZE:(%TEMPC) .NE -1) THEN CLEAR T %TEMPC
2000. TYPE 'An error occured in this macro. Try calling the macro again. No action taken.'
2020. GO TO 2090
2030.
2040. TYPE 'BREAK, ESCAPE, or ATTN was pressed. No action taken'
2050.

```


III CPDS Macro

```

1.      * CPDS - MACRO TO COPY ONE LIBRARY TO ANOTHER
2.
3.      SET EXIT ATTENTION 58
4.
4.01   LET TEMPY = .TEMPORARY
4.1    IF (.T3270 = 1) THEN CLR SSA
9.
10.    TYPE "Enter the name of the input PDS (required):"
11.    DEMAND STRING IDSN "> "
12.    LET IDSN = .UPPER:(.TRIM:(%IDSN;' '; 'B'))
12.1   IF (IDSN='')
12.2     THEN TYPE 'Nothing entered. Try again.'
12.4   THEN GOTO 11
13.
14.    TYPE "Enter the volume of the input PDS (required):"
15.    DEMAND STRING IVOL "> "
16.    IF (IVOL = '') GO TO 19
16.1   COL END T %TEMPY NEW CMD SHO VOLS LIKE %IVOL
17.    LET IVOL = ' ON ' .UPPER:(%IVOL)
18.    IF (.SIZE:(%TEMPY).EQ 1) THEN CLEAR T %TEMPY
18.1   THEN GOTO 19
18.11  * ELSE
18.12  CLEAR T %TEMPY
18.13  TYPE 'Invalid volume name entered. Re-enter.'
18.14  GOTO 15
18.15  COL END T %TEMPY NEW CMD SHO DSNAME LIKE %IDSN ON %IVOL
18.16  IF (.SIZE:(%TEMPY) .EQ 2) THEN CLEAR T %TEMPY
18.17  THEN GO TO 19
18.18  * ELSE
18.19  CLEAR T %TEMPY
18.2   TYPE 'Data set does not exist on specified volume. Re-Enter'
18.21  GO TO 10
19.    TYPE "Enter the name of the output data set (optional): "
20.    DEMAND STRING ODSN "> "
21.    LET ODSN = .UPPER:(.TRIM:(%ODSN;' '; 'B'))
22.    IF (ODSN = '') LET ODSN = %IDSN
23.
24.    TYPE "Enter the volume of the output data set (required): "
25.    DEMAND STRING OVOL "> "
25.001 IF (OVOL = '') THEN GO TO 29
25.011 LET OVOL = .TRIM:(.UPPER:(%OVOL);' '; 'B')
25.021 COL END T %TEMPY NEW CMD SHO VOLS LIKE %OVOL
25.031 IF (.SIZE:(%TEMPY).EQ 1) THEN CLEAR T %TEMPY
25.041 THEN GOTO 27
25.051 * ELSE
25.061 CLEAR T %TEMPY
25.071 TYPE 'Invalid volume entered. Re-enter'
25.081 GOTO 25
25.2
27.    LET OVOL = ' ON ' .UPPER:(%OVOL)
28.
29.    TYPE "Enter the format of the output data set "
29.1   TYPE "as Card, Edit, etc. (optional)"
30.    DEMAND STRING OFMT "> "
31.    IF (OFMT = '') GO TO 34
32.    LET OFMT = .UPPER:(%OFMT)
33.
34.    TYPE "Enter the space of the output data set (required): "
34.1   TYPE '      Format is (TRK,(primary,secondary,directory blocks)) '
35.    DEMAND STRING OSPA "> "

```

36. LET OSPA = .UPPER:(%OSPA)
37.
38. LET M = .TEMPORARY
39. LET N = .TEMPORARY
40. SET EXIT ATTENTION 57
41. C T %M NEW CMD SHO MEMS FR %IDSN %IVOL
42. DEL F L T %M
43. QT NUM T %M
44. QT COPY F/L T %M TO 5001 T %M
45. QT CH 1/2 TO 'USE FROM ' %%IDSN ':' IN F/5000 T %M NOL
46. QT CH EOL TO %%IVOL ' CLEAR T ' %%N IN F/5000 T %M NOL
47. QT CH 1/2 TO 'SAVE T ' %%N ' INTO ' %%ODSN ':' IN 5001/L T %M NOL
48. QT CH EOL TO %%OVOL IN 5001/L T %M
49. QT MOV 1001/L T %M TO 1.5 T %M BY 1 MERGE
50. QT CH EOL TO ' ' %%OFMT ' SPACE ' %%OSPA IN 1.5 T %M
51. QT CH '\$N\$' TO %%N IN 60 T %M.MACRO
52. QT COPY 60 T %M.MACRO TO END T %M
53. QT NUM T %M
54.
55. CALL T %M
56.
57. CLEAR T %M
58. RETURN
59.
60. CLEAR T \$N\$

II COPYSEQ Macro

```

1. *****
2. *
3. *
4. * -->> I E B G E N E R M A C R O <<--*
5. *
6. *
7. * FUNCTION: THIS MACRO WILL PERFORM BACKUP COPY,
8. * ***** COPY, AND PRINT OPERATIONS ON A
9. * SEQUENTIAL DATA SET OR PDS MEMBER.
10. *
11. * NAME: COPYSEQ (WRITTEN BY KEVIN A. BERG)
12. * ****
13. *
14. * DEVICES: THIS MACRO COPIES A SEQUENTIAL DATA SET
15. * ***** OR PDS MEMBER TO A NEW SEQUENTIAL DATA
16. * SET OR INTO A PARTITIONED DATA SET. THE
17. * MACRO ALSO CAN PRINT DATA SETS.
18. *
19. * SUMMARY: THIS MACRO WILL QUERY THE USER TO OBTAIN THE
20. * ***** INFORMATION REQUIRED TO BUILD THE PROPER JOB
21. * STREAM OF AN IEBCOPY UTILITY.
22. *
23. * NOTES:
24. * =====
25. * * COPYSEQ CALLS NO OTHER MACROS
26. * * COPYSEQ REQUIRES 2 TEMPORARY DATA SETS
27. * * COPYSEQ EDITS ALL INPUT FROM USER
28. * * COPYSEQ ALLOWS FOR MULTIPLE JOB STEPS
29. * * COPYSEQ USES NO EXTERNAL PARAMETES
30. *
31. * ++ COPYSEQ CREATES A TEMPORARY DATA SET (JOB.STREAM) TO
32. * GENERATE THE USER PROGRAM FOR EXECUTION.
33. *
33.1 *****
33.2 *
33.3 * FALL 1988
33.4 * REVISED BY KAREN ROWE
33.5 * CALLED BY THE D50TD80 MACRO
33.6 *
33.7 *****
35.
36.
36.1 MAIL 36.2 T %.MACRO TO A10PUB1
36.2 **D50TD80 - COPYSEQ USED**
37.
38.
39. * ALLOCATE TEMPORARY WORK DATA SET JOB.STREAM
40. QUIET COPY 418/424 T %.MACRO TO END NEW T JOB.STREAM
41. * CODE TO DIRECT RECOVERY
42. SET EXIT ERROR 409
43. SET EXIT ATTENTION 378
44.
45. LET GENERNUM = 1
46. LET (INVOL,VOL.TYPE,VOL.TYPE2,SEQNUM,PASSWORD,PASSWORD1,PASSWORD2,INDCB,OUTDCB) = " "
47. LET (DCB.DENSITY,TAPEVOL,OUTUNIT,INCATL,OUTCATL) = " "
48. LET PRCOMMENT1 = "/* THIS JOB STEP DIRECT COPIES A SEQUENTIAL DATA SET"
49. LET PRCOMMENT2 = "/* THIS JOB STEP EXPANDS A PARTITIONED DATA SET"
50. LET PRCOMMENT3 = "/* THIS JOB STEP PRINTS A SEQUENTIAL DATA SET"
51.
52. DEM STR HELP 'Enter "y" for instructions or hit RETURN >'

```

```

54. LET HELP = .TRIM:(%%HELP;" ";"B")
55. IF (.UPPER:(HELP)~="Y") GO TO 108
56. MAT (SOME "LH") (.PSOURCE) CLEAR SSA
57. *
58. * INSTRUCTIONS FOR MACRO IMPLEMENTATION
59. T "This macro has been designed to assist users with copying"
60. T "sequential data sets from disk to tape, tape to disk, disk"
61. T "to disk, and tape to tape. The macro also enables a user to"
62. T "create a printed listing of a sequential data set or PDS member."
63. T "*** You CANNOT use this macro to create a PDS ***"
99. T
100. T "*** NOTE: Only two tapes may be used for each call to this"
101. T " macro, and these tapes must be specified in the "
102. T " first job step (must be standard label tape)."
103. T
104. DEM STR ANSR "When you have read the instructions hit RETURN. "
105.
108. IF (.BIN~="" .AND .BIN~=0000) LET BIN = .BIN
109. THEN GO TO 120
110. T 'What bin do you wish to send your output to?'
111. DEM STR BIN '>'
112. LET BIN = .TRIM:(%%BIN;" ";"B")
113. IF (.INDEX:(BIN)~=4) T "***BIN NUMBER MUST BE FOUR DIGITS LONG"
114. THEN GO TO 110
115. MAT(DIGIT*4)(BIN) GO TO 120
116. T "***BIN NUMBER MUST BE NUMERIC, PLEASE TRY AGAIN"
117. GO TO 110
118.
119.
120. QUIET CH "$NAME$" TO %USER IN T JOB.STREAM
121. QUIET CH "$BIN$" TO %BIN IN T JOB.STREAM
122.
123. IF (GENERUM = 1) GO TO 129
124. DEM STR ANSR "Do you need to see the menu of operations again? (Y or N): "
126. IF (.UPPER:(ANSR)="N") GO TO 141
127. IF (.UPPER:(ANSR)~="Y") T ANSR " INVALID RESPONSE, Please enter Y or N"
128. THEN GO TO 123
129. MAT (SOME "LH") (.PSOURCE) CLEAR SSA
130. T "Choose from one of the following operations:"
131. T
132. T "1) Create a backup copy of a sequential data set or"
133. T " partitioned data set member."
134. T
135. T "2) Expand a partitioned data set by copying a new"
136. T " sequential data set into the partitioned data set."
137. T
138. T "3) Create a printed listing of a sequential data set"
139. T " or partitioned data set member."
140. T
141. DEM STR GENER.FUNCTION "Type in the number of the operation you wish to use. (1,2, or 3) :"
143. MAT (NOT DIGIT) (GENER.FUNCTION) T GENER.FUNCTION " INVALID RESPONSE, Please enter a number between 1 and 3"
144. THEN GO TO 141
145. IF (GENER.FUNCTION < 1 .OR GENER.FUNCTION > 3) T GENER.FUNCTION " INVALID RESPONSE, Please enter a number between 1 and
3"
146. THEN GO TO 141
147. QUIET COPY 425/427 T %MACRO TO END T JOB.STREAM
148. IF (GENER.FUNCTION=1) QUIET COPY 430/432 T %MACRO TO END T JOB.STREAM
149. THEN GO TO 168
150. IF (GENER.FUNCTION=2) QUIET COPY 430/432 T %MACRO TO END T JOB.STREAM
151. THEN GO TO 168
152. QUIET COPY 428/429 T %MACRO TO END T JOB.STREAM

```

```

153. LET (OUTGENER,OUTUNIT,OUTVOL,OUTSEQ,OUTPASS,OUTCATL) = " "
154.
155. DEM STR LRECL "What record length do you wish to set for the printout? :"
157. MAT (NOT DIGIT) (LRECL) T LRECL " <----*** INVALID RECORD LENGTH, Please enter a number less than 122"
158. THEN GO TO 155
159. IF (LRECL < 0 .OR LRECL > 121) T LRECL " <----*** INVALID RESPONSE, Record length MUST be less than 122"
160. THEN GO TO 155
161. QUIET CH "$RECLNS$" TO %LRECL IN T JOB.STREAM
165. *
166. * INITIALIZE REPLACEMENT VARIABLES
167. *
168. LET (UNIT.TYPE,VOLUME.TYPE,SEQNUM,CATL) = " "
169. LET (CATL,OUT.TYPE) = " "
170. * FIRST RETRIEVE INPUT DATA SET THEN OUTPUT DATA SET INFORMATION
171. *
175. T;T "The following questions concern the input data set"
176. LET ACTIVE.GENER = "IN"
177. GO TO 194
181. T;T "The following questions concern the output data set"
182. LET ACTIVE.GENER = "OUT"
186. T;DEM STR OUT.TYPE "Is the output data set new or old? (N or O) :"
188. LET OUT.TYPE = .TRIM:(%OUT.TYPE;" ";"B")
189. IF (.UPPER:(OUT.TYPE)="N") GO TO 194
190. IF (.UPPER:(OUT.TYPE)~="O") T OUT.TYPE " <----*** INVALID RESPONSE, Please enter 'N' OR 'O'"
191. THEN GO TO 186
193.
194. LET GENERNAME = " "
195. LET UNIT.TYPE = " "
196. LET SEQNUM = " "
197. LET VOLUME.TYPE = " "
198. LET CATL = "N"
199. LET DCB.DENSITY = " "
200. T 'Enter the name of your data set or PDS member'
200.1 T ' pdsname(membername).'
201. DEM STR GENERNAME '>'
202. LET GENERNAME = .UPPER:(.TRIM:(%GENERNAME;" ";"B"))
202.1 LET GENERNAME = .LOGONID ' ' .TRIM:(%GENERNAME;" ";"B")
203. ***T ;MAT (BOS ((LETTER (LETTER OR DIGIT)*0/7) ((". " LETTER (LETTER OR DIGIT)*0/7)*0/22) ((((" LETTER (LETTER OR DIGIT)*
0/7)"))*0/1)) AND ANY*1/44 EOS) GENERNAME T "This is your data set name --> " GENERNAME
204. **EN GO TO 208
205. ***ENERNAME " <----*** INVALID DATA SET NAME, PLEASE RETYPE"
206. ***TO 194
207. ***
208. T
209. IF (.UPPER:(ACTIVE.GENER)="OUT" .AND .UPPER:(OUT.TYPE)="N") GO TO 218
210. T "Is this a catalogued data set? (Y or N)";DEM STR CATL '>'
211. LET CATL = .TRIM:(%CATL;" ";"B")
212. IF (.UPPER:(CATL)="Y") GO TO 218
213. IF (.UPPER:(CATL)~="N") T CATL ' <----*** INVALID RESPONSE, PLEASE ANSWER Y OR N'
214. THEN GO TO 208
217.
218. T 'Does (or will) this data set reside on disk or tape? (D or T)'
219. DEM STR ANSR '>'
220. LET ANSR = .TRIM:(%ANSR;" ";"B")
221. IF (.UPPER:(ANSR)~="D") GO TO 236
222. LET UNIT.TYPE = "DISK"
223. IF (.UPPER:(CATL)="Y") GO TO 282
224. T 'Enter the disk volume number on which this data set does (or will) reside'
225. DEM STR VOLUME.TYPE '>'
226. LET VOLUME.TYPE = .TRIM:(%VOLUME.TYPE;" ";"B")
227. T

```

```

228. IF (VOLUME.TYPE = "") GO TO 224
229. LET VOLUME.TYPE = .UPPER:(VOLUME.TYPE)
230. MAT (BOS LET*3 DIG*3 EOS) (VOLUME.TYPE) GO TO 282
231. T VOLUME.TYPE " <---*** INVALID VOLUME, PLEASE REENTER VOLUME"
232. GO TO 224
233.
236. IF (.UPPER:(ANSR)≠"T") T ANSR "<---*** INVALID DEVICE TYPE, Please enter 'D' or 'T'"
237. THEN GO TO 218
238. LET UNIT.TYPE = "TAPE"
239. IF (.UPPER:(CATL)="Y") GO TO 261
240.
241. T
242. T 'Enter the volume number for this data set (6 digits).'
243. DEM STR TAPEVOL '>'
244. LET (TAPEVOL,VOLUME.TYPE) = .TRIM:(%TAPEVOL;" ";"B")
245. T
246. IF (VOLUME.TYPE = "") GO TO 241
247. IF (.INDEX:(VOLUME.TYPE)≠6) T"Volume serial numbers MUST have 6 digits"
248. THEN GO TO 241
249. MAT (DIGIT*6) (VOLUME.TYPE) GO TO 253
250. T "Tape volume serial numbers MUST be numeric, Please retype volume number"
251. GO TO 241
252.
253. T "Enter the sequence number on the tape for this data set"
254. DEM STR SEQNUM ">"
255. LET SEQNUM = .TRIM:(%SEQNUM;" ";"B")
256. IF (SEQNUM = "" .OR SEQNUM= 0) T SEQNUM " INVALID SEQUENCE NUMBER, Please retype response"
257. THEN GO TO 253
258. MAT (NOT DIGIT) (SEQNUM) T SEQNUM " INVALID SEQUENCE NUMBER, Please retypr response"
259. THEN GO TO 253
260.
261. IF (GENERNUM>1) GO TO 273
262. T "Does this tape require a password? (Y or N)"
263. DEM STR PASSKEY '>'
264. LET PASSKEY = .TRIM:(%PASSKEY;" ";"B")
265. T
266. IF (.UPPER:(PASSKEY)="N") GO TO 273
267. IF (.UPPER:(PASSKEY)≠"Y") T PASSKEY " INVALID RESPONSE, Please enter Y or N"
268. THEN GO TO 261
269. T "Enter the password for this tape in the next line."
270. DEM STR PASSWORD ">"
271. LET PASSWORD = .TRIM:(%PASSWORD;" ";"B")
272.
273. T "Enter the density of this tape. (1600 or 6250)"
274. DEM STR DCB.DENSITY ">"
275. LET DCB.DENSITY = .TRIM:(%DCB.DENSITY;" ";"B")
276. IF (DCB.DENSITY≠1600 .AND DCB.DENSITY≠6250) T DCB.DENSITY " INVALID RESPONSE, PLEASE ENTER 1600 OR 6250 "
277. THEN GO TO 273
278. T
279. *
280. * SAVE INPUT DATA SET INFORMATION
281. *
282. IF (.UPPER:(ACTIVE.GENER) = "OUT") GO TO 301
283. LET INGENER = %GENERNAME
284. LET INUNIT = %UNIT.TYPE
285. LET INSEQ = %SEQNUM
286. LET INPASS = %PASSWORD
287. LET INVOL = %VOLUME.TYPE
288. LET INCATL = %CATL
289. IF (.UPPER:(INUNIT)≠"TAPE") GO TO 296

```

```

290. LET UNIT.TYPE = %%INUNIT
291. LET INVOL = %%TAPEVOL
292. LET VOL.TYPE1 = %%INVOL
293. LET SEQNUM = %%INSEQ
294. LET PASSWORD1 = %%INPASS
295. LET INDCB = %%DCB.DENSITY
296. IF (GENER.FUNCTION=3) GO TO 319
297. GO TO 181
298. *
299. * SAVE OUTPUT DATA SET INFORMATION INTO THE OUTGENER DATA NAMES
300. *
301. LET OUTGENER = %%GENERNAME
302. LET OUTUNIT = %%UNIT.TYPE
303. LET OUTVOL = %%VOLUME.TYPE
304. LET OUTSEQ = %%SEQNUM
305. LET OUTPASS = %%PASSWORD
306. LET OUTCATL = %%CATL
307. IF (.UPPER:(OUTUNIT)≠"TAPE") GO TO 319
308. LET UNIT.TYPE = %%OUTUNIT
309. LET OUTVOL = %%TAPEVOL
310. LET VOL.TYPE2 = %%OUTVOL
311. LET SEQNUM = %%OUTSEQ
312. LET PASSWORD2 = %%OUTPASS
313. LET OUTDCB = %%DCB.DENSITY
314.
315. *
316. *
317. * CHANGE JOB STREAM DUMMY FIELDS TO USER SUPPLIED VALUES
318. *
M 319. IF (GENER.FUNCTION = 1) QUIET CH "//*MESSAGE THIS JOB STEP USES TAPE # $TAPES$ $PASSWORD$" TO %%PRCOMMENT1 IN T JOB.STREA
M 320. IF (GENER.FUNCTION = 2) QUIET CH "//*MESSAGE THIS JOB STEP USES TAPE # $TAPES$ $PASSWORD$" TO %%PRCOMMENT2 IN T JOB.STREA
M 321. IF (GENER.FUNCTION = 3) QUIET CH "//*MESSAGE THIS JOB STEP USES TAPE # $TAPES$ $PASSWORD$" TO %%PRCOMMENT3 IN T JOB.STREA
322. IF (.UPPER:(INUNIT)="TAPE") QUIET CH "//*MESSAGE" TO "/* " AND "$TAPE2$" TO %%VOL.TYPE1 AND "$PASSWORD2$" TO %%PAS
SWORD1 IN T JOB.STREAM
323. IF (.UPPER:(OUTUNIT)="TAPE") QUIET CH "//*MESSAGE" TO "/* " AND "$TAPE2$" TO %%VOL.TYPE2 AND "$PASSWORD2$" TO %%
PASSWORD2 IN T JOB.STREAM
324. IF (.UPPER:(INUNIT)≠"TAPE" .AND .UPPER:(OUTUNIT)≠"TAPE") QUIET DEL "//*MESSAGE"*1/10 T JOB.STREAM
325. LET STEPNAME = "STEP" GENERNUM
326. QUIET CH "$STEP#$" TO %%STEPNAME IN T JOB.STREAM
327.
328. IF (.UPPER:(OUT.TYPE)="N") QUIET CH "$OUTDISP$" TO "NEW,KEEP,DELETE" IN T JOB.STREAM
329. IF (.UPPER:(OUT.TYPE)="O") QUIET CH "$OUTDISP$" TO "OLD,KEEP" IN T JOB.STREAM
330.
331. IF ((.UPPER:(INCATL)="Y" .AND .UPPER:(INVOL)="TAPE") .OR (.UPPER:(OUTCATL)="Y" .AND .UPPER:(OUTVOL)="TAPE")) QUIET CH "T
APE #" TO "PASSWORD =" IN T JOB.STREAM
332. QUIET CH "$INDSNAMES$" TO %%INGENER AND "$OUTDSNAMES$" TO %%OUTGENER IN T JOB.STREAM
333. IF (.UPPER:(INCATL)="Y") QUIET CH "/* VOL=SER=$INVOL$,UNIT=$INUNIT$,DCBIN$" TO "+++++" IN T JOB.STREAM
334. IF (.UPPER:(OUTCATL)="Y") QUIET CH "/* VOL=SER=$OUTVOL$,UNIT=$OUTUNIT$,DCBOUT$" TO "+++++" AND "SPACE=$SPAC
EPARMS," TO "" IN T JOB.STREAM
335.
336. IF (INDCB=1600 .AND .UPPER:(INUNIT)="TAPE") LET INDCB = "DCB=DEN=3,"
337. IF (INDCB=6250 .AND .UPPER:(INUNIT)="TAPE") LET INDCB = "DCB=DEN=4,"
338. IF (OUTDCB=1600 .AND .UPPER:(OUTUNIT)="TAPE") LET OUTDCB = "DCB=DEN=3,"
339. IF (OUTDCB=6250 .AND .UPPER:(OUTUNIT)="TAPE") LET OUTDCB = "DCB=DEN=4,"
340. QUIET CH "$DCBIN$" TO %%INDCB IN T JOB.STREAM
341. QUIET CH "$DCBOUT$" TO %%OUTDCB IN T JOB.STREAM
342.
343. IF (.UPPER:(INUNIT)="DISK" .AND .UPPER:(OUTUNIT)="DISK") GO TO 362
344. IF (.UPPER:(INUNIT)="TAPE") GO TO 350
345. IF (.UPPER:(INUNIT)≠"TAPE") QUIET CH "//*MESSAGE THIS JOB STEP USES TAPE # $TAPES$ $PASSWORD$" TO %%PRCOMMENT1 IN T JOB.STREA

```

349.

```

350. IF (.UPPER:(INCATL)="Y") GO TO 354
351. QUIET CH "$INUNIT$" TO %%INUNIT IN T JOB STREAM
352. LET INSEQ = "(,RETAIN,,SER=" %%INVOL "),LABEL=(" %%INSEQ ",SL)"
353. QUIET CH "SER=$INVOL$" TO %%INSEQ IN T JOB STREAM
354. IF (.UPPER:(OUTCATL)="Y") GO TO 367
355. QUIET CH "$OUTUNIT$" TO %%OUTUNIT IN T JOB STREAM
356. IF (.UPPER:(OUTUNIT)="DISK") QUIET CH "$SPACEPARM$" TO "(TRK,(25,1),RLSE)" AND "$OUTVOL$" TO %%OUTVOL IN T JOB STREAM
357. THEN GO TO 367
358. LET OUTSEQ = "(,RETAIN,,SER=" %%OUTVOL "),LABEL=(" %%OUTSEQ ",SL)"
359. QUIET CH "SER=$OUTVOL$" TO %%OUTSEQ AND "SPACE=$SPACEPARM$," TO "" IN T JOB STREAM
360. GO TO 367
361.
362. IF (.UPPER:(INCATL)!="Y") QUIET CH "$INUNIT$" TO %%INUNIT IN T JOB STREAM
363. THEN QUIET CH "$INVOL$" TO %%INVOL IN T JOB STREAM
364. IF (.UPPER:(OUTCATL)!="Y") QUIET CH "$OUTUNIT$" TO %%OUTUNIT AND "$SPACEPARM$" TO "(TRK,(25,1),RLSE)" AND "$OUTVOL$" TO
%%OUTVOL IN T JOB STREAM
365.
366.
367. T "Do you wish to process another data set? (Type Y or N)"
368. DEM STR ANSR ">"
369. LET ANSR = .TRIM:(%%ANSR;" ";"B")
370. QUIET COPY 433 T %.MACRO TO END T JOB STREAM
371. IF (.UPPER:(ANSR)="N") GO TO 406
372. IF (.UPPER:(ANSR)!="Y") T ANSR " <----** INVALID RESPONSE, PLEASE ANSWER Y OR N"
373. THEN GO TO 367
374. LET GENERNUM = GENERNUM + 1
375. QUIET COPY 420/424 T %.MACRO TO END T JOB STREAM
376. GO TO 123
377.
378. LET RETURN = .LATTN - 1
379. SET EXIT ATTENTION 409
380.
381. T
382. T
383. T "You have entered an attention exit. Do you wish to continue (Y or N)?"
386. DEM STR QUEST ">"
387. LET QUEST = .TRIM:(%%QUEST;" ";"B")
388. IF (.UPPER:(QUEST)="Y") SET EXIT ATTENTION 378
389. THEN GO TO %RETURN
390. IF (.UPPER:(QUEST)="N") GO TO 415
391. T QUEST "<----** INVALID RESPONSE . . . TYPE Y OR N"
392. GO TO 385
393.
402.
403.
404. * EXECUTE IEBGENER JOB STREAM
405. *
406. QUIET DEL '+*10 1/10 T JOB STREAM;LIST T JOB STREAM UNN
407. RUN UNN T JOB STREAM;T "GENR" %.USER " IS YOUR JOB NAME"
408.
414. CLEAR T JOB STREAM
415. RETURN
416.
417.
418. //GENR$NAME$ JOB
419. /*JOBPARM L=1,R=$BIN$,A=2
420. //**MESSAGE THIS JOB STEP USES TAPE # $TAPE$ $PASSWORD$
421. //**MESSAGE THIS JOB STEP USES TAPE # $TAPE2$ $PASSWORD2$
422. // $STEP#$ EXEC PGM=IEBGENER,COND=(O,NE)
423. //SYSPRINT DD SYSOUT=A
424. //SYSIN DD DUMMY *** NO CONTROL CARDS FOR THIS JOB ***

```



```
425. //SYSUT1 DD DSN=$INDSNAME$,
426. // VOL=SER=$INVOL$,UNIT=$INUNIT$, $DCBIN$
427. // DISP=(OLD,KEEP)
428. //SYSUT2 DD SYSOUT=A, **** PRINT THE DATA SET ACC ****
429. // DCB=(RECFM=F,LRECL=$RECLEN$,BLKSIZE=$RECLEN$)
430. //SYSUT2 DD DSN=$OUTDSNAME$,
431. // VOL=SER=$OUTVOL$,UNIT=$OUTUNIT$, $DCBOUT$
432. // SPACE=$SPACEPARM$,DISP=($OUTDISP$)
433. /*
```

UPLOAD Macro

```

0.01 *****
0.02 ***
0.1 *** WRITTEN BY KAREN ROWE
0.2 *** 10/14/88
0.3 ***
1. *** THIS MACRO IS USED TO HELP IN UPLOADING AND DOWNLOADING
2. *** TO AND FROM SUPERWYLBER WHEN USER IS LOGGED ON TO A 300
3. *** BAUD LINE ( SOURCE='DIAL-WYL').
4. ***
4.1 *****
4.2 LET TEMPA = .TEMPORARY
4.3 LET TEMP = .TEMPORARY
5. IF (.SOURCE ^= 'DIAL-WYL') THEN GOTO 100
6. T 'Do you wish to upload or download (up,down,no)?'
6.09 DEMAND STRING ANSWER '>'
6.1 IF (ANSWER = '') TYPE 'Nothing entered. Please enter UP, DOWN, or N'
6.2 THEN GO TO 6.09
6.9 LET ANSWER = .UPPER:(.substring:(ANSWER;1;1))
7. IF (ANSWER ^= 'U') THEN GOTO 25
8. SET NO BREAK
9. SET UPLOW
10. CLR TEXT
10.5 TYPE 'Ready for uploading'
11. COL UNN
12. RETURN
14. *
25. IF (ANSWER ^= 'D') THEN GOTO 100
25.1 T 'ENTER the name of the data set to be down loaded'
25.11 DEMAND STRING NAME '>'
25.111 LET NAME = .TRIM:(.UPPER:(%%NAME));' '; 'B')
25.112 IF (NAME = '') THEN TYPE ' '
25.113 THEN TYPE 'NOTHING ENTERED. TRY AGAIN.'
25.114 THEN TYPE '(PRESS BREAK OR ESCAPE TO STOP.)'
25.115 THEN TYPE ' '
25.116 THEN GO TO 25.1
25.117 *
25.2 T 'ENTER the volume where dataset resides'
25.3 DEMAND STRING VOLUME '>'
25.4 LET VOLUME = .TRIM:(.UPPER:(%%VOLUME));' '; 'B')
25.5 IF (VOLUME = '') THEN TYPE ' '
25.6 THEN TYPE 'NOTHING ENTERED. TRY AGAIN.'
25.7 THEN TYPE '(PRESS BREAK OR ESCAPE TO STOP.)'
25.8 THEN TYPE ' '
25.9 THEN GO TO 25.2
25.91 COL END T %TEMPA NEW CMD SHOW VOLS LIKE %VOLUME
25.92 IF (.SIZE:(%%TEMPA) .EQ 1) THEN CLEAR T %TEMPA
25.93 THEN GO TO 26
25.94 * ELSE
25.95 CLEAR T TEMPA
25.96 TYPE ' '
25.97 TYPE 'INVALID VOLUME NAME ENTERED. RE-ENTER.'
25.98 GO TO 25.2
26. DEMAND STRING EMPTY 'Prepare PC for download - hit ENTER when ready'
27. USE FROM %NAME ON %VOLUME CLR
28. LIST UNN
100. RETURN

```