

This is the help file from the diag-games-kermit.rk05 image file slightly modified by me.

[Getting online help](#)

[General operating infomation](#)

[Other operating system help and links to main help](#)

[OS/8 Summary](#)

System control and program exection

[Command interpreter](#)

[Boot from a device](#)

[Set system date](#)

[Configure system](#)

[Modify system operation](#)

[Set highest memory field](#)

[Load binary file](#)

[Run batch file](#)

[Run file from SYS](#)

[Run file from any device](#)

[Create logical name for device](#)

[Remove logical name for device](#)

[Load core image](#)

[Save memory to file](#)

[Start program in memory running](#)

[Save Command](#)

File operations

[List directory contents](#)

[Copy file](#)

[Delete file](#)

[Rename file](#)

[Copy file to printer](#)

[Display file on console](#)

[Defragment Drive](#)

[Write empty directory](#)

[Line editor](#)

[TECO, more complex editor](#)

[Process file with TECO macro](#)

[Paper tape editor](#)

[File compare](#)

[Compare files using a mask](#)

[Ascii file sorter](#)

[List memory used by binary files](#)

[File Transfer](#)

[Do file operation on peripherals](#)

[DEC 10 DECtape utility](#)

[File/device modification utility](#)

[Convert SV format file to bin](#)

[Dump SV file in octal](#)

[Copy RX disks](#)

Languages

[Focal 71](#)

[Basic](#)

[Fotran II/IV](#)

[PAL8 assembler](#)

[PAL8 errors](#)

[Cross reference PAL source](#)

[Sabr assembler](#)

[ODT Debugger](#)

[Link and load](#)

Other programs

[Kermit file transfer program](#)

[Scroll scope editor](#)

Help for program I don't have

[Disassemble file](#)

[Modify disk data](#)

[PIP with verify](#)

Games

[Various Games](#)

## **OS/8 MONITOR COMMANDS**

CMD	PROG	EXPL
<a href="#">ABSLDR</a>	LOAD	LOADS AND EXECUTES
<a href="#">ASSIGN</a>	KBM	ASSIGNS LOGICAL NAME
BACKSP	<a href="#">CAMP</a>	BACKSPACES DEV
<a href="#">BASIC</a>	BASIC	ENTERS BASIC SYSTEM
<a href="#">BOOT</a>	BOOT	BOOTSTRAPS TO DEV
CCL	CCL	DISABLES CCL
<a href="#">COMPARE</a>	SRCCOM	COMPARES FILES
COMPILE	PAL8	COMPILES PROG
	F4/FORT	
	BASIC	
	RALF	
	SABR	
<a href="#">COPY</a>	<a href="#">FOTP</a>	COPIES FILES
<a href="#">CREATE</a>	EDIT	OPENS FILE FOR EDITING
<a href="#">CREF</a>	PAL8	ASSEMBLES AND CHAINS TO CREF
	CREF	CREF'S LISTING
<a href="#">DATE</a>	KBM/CCL	SPECIFIES DATE
<a href="#">DEAS</a>	CCL	DEASSIGNS LOGICAL DEVICES
<a href="#">DELETE</a>	FOTP	DELETES FILES
<a href="#">DIRECT</a>	DIRECT	PRINTS DIRECTORIES
<a href="#">DUPLIC</a>	RXCOPY	COPIES RX DISKS
<a href="#">EDIT</a>	EDIT	EDITS FILE
EOF	CAMP	WRITES END-OF-FILE
EXECUTE	PAL8	COMPILES AND EXECUTES
	F4/FORT	
	BASIC	
	RALF	
	SABR	
<a href="#">GET</a>	KBM	GETS CORE-IMAGE
<a href="#">HELP</a>	HELP	LIST'S HELP FILE
<a href="#">LIST</a>	FOTP	LISTS FILES
<a href="#">LOAD</a>	ABSLDR	LOADS FILES
	LOAD(ER)	
<a href="#">MAKE</a>	TECO	MAKES NEW FILE FOR EDITING

<a href="#">MAP</a>	BITMAP	PRINTS BITMAP
<a href="#">MEMORY</a>	CCL	SPECIFIES MACHINE CORE SIZE
<a href="#">MUNG</a>	TECO	MUNGS FILE WITH TECO MACRO
<a href="#">ODT</a>	KBM	RUNS OCTAL DEBUGGER
<a href="#">PAL</a>	PAL8	RUNS PAL8
PRINT	LPTSPL	RUNS 'LPTSPL' IF PRESENT
PUNCH	FOTP	PUNCHES DATA
<a href="#">R</a>	KBM	RUNS PROGRAM FROM SYS:
<a href="#">RENAME</a>	FOTP	RENAMES FILES
RESORC	RESORC	PRINTS RESOURCES OF SYSTEMS
REWIND	CAMP	REWINDS DEV
<a href="#">RUN</a>	KBM	RUNS PROGRAM
<a href="#">SAVE</a>	KBM	SAVES CORE IMAGE
<a href="#">SET</a>	SET	ALTERS PARAMETERS
SKIP	CAMP	SKIPS RECORDS
<a href="#">SQUISH</a>	<a href="#">PIP</a>	SQUISHES DEV
<a href="#">START</a>	KBM	STARTS PROG
<a href="#">SUBMIT</a>	BATCH	STARTS BATCH JOB
<a href="#">TECO</a>	TECO	EDITS FILE
<a href="#">TYPE</a>	FOTP	TYPES FILES
<a href="#">UA</a>	CCL	REMEMBERS COMMAND
<a href="#">UB</a>	CCL	
<a href="#">UC</a>	CCL	
UNLOAD	CAMP	UNLOADS DEV
VERSION	CCL	TYPES VERSION #
<a href="#">ZERO</a>	PIP	ZEROES DEV

**@SWITCHES:**

```
-L      OUTPUT TO LPT:
-S      OUTPUT TO TV:
-T      OUTPUT TO TTY:
-P      OUTPUT TO PTP:
-D      OUTPUT TO DUMP:
-N      OUTPUT TO NULL:
-LS     PRODUCE LISTING
-NB     NO BINARY YET
-MP     PRODUCE MAP
-EXT    SET DEFAULT EXTENSION
```

**@FEATURES:**

```
/X      PASS SWITCH OPTION X TO PROGRAM
(XYZ)   PASS SWITCH OPTIONS TO PROGRAM
[N]     MAX OUTPUT SIZE
=NNN    PASS OCTAL NUMBER TO PROGRAM
#NNN    TAKE INTERNAL OCTAL FORM OF FILENAME
@FILE   REPLACE IN CMD LINE BY FILE'S CONTENTS
$       COMPLEMENT DEFAULT ALTMODE SWITCH
```

**ABSLDR.SV Absolute Binary Loader****@CALLING COMMANDS:**

```
.LOAD DEV:BINFILE.BN,...
.LOAD BINFILE.BN,... /FROM DSK
```

**@SWITCHES:**

```
/8      PROG DOESN'T USE BELOW 02000
/9      PROG DOESN'T USE BELOW 12000
/G      GO
/I      CORE IMAGE FILE
/P      PROG DOESN'T DESTROY EXTENDED BATCH RESIDENT
/R      RESET
```

```

/S      MULTIPLE BINARIES/FILE
/N      FORCE LOADING TO FIELD N (N IS AN OCTAL DIGIT)
=FNNNN SET STARTING ADDRESS

```

### **BOOT.SV Boot From a Device**

@CALLING COMMANDS:

```
.BOOT/DV
```

@SWITCHES:

```

/CA     TA8E CASSETTE    CAPS-8
/DK     ANY DISK
/DL     LINCTAPE        DIAL
/DM     ANY DISK        DISK MONITOR
/DT     ANY TAPE
/LT     LINCTAPE
/PT     PT8E (LOADS BINLDR)
/RE     RK8E DISK
/RF     RF08,DF32 DISKS
/RK     RK8 DISK
/RX     RX8E FLOPPY DISK
/TD     TD8E DECTAPE
/TY     TYPESET (UNIT 4)
/VE     VERSION #
/TC     TC08 DECTAPE    ALL SYSTEMS
/ZE     ZEROES CORE (FIELD 0)

```

DEVICES ARE UNIT 0 IF NOT SPECIFIED

```
DV.     HALT AFTER LOADING BOOTSTRAP
```

### **BUILD.SV OS/8 System Builder**

@INTERNAL COMMANDS:

```

$ALTER GRP,LOC
$ALTER GRP,LOC=VALUE
$BOOT
$BUILD
$CTL ACTNAM
$CTL ACTNAM=VALUE
$CORE N
$DCB ACTNAM
$DCB ACTNAM=VALUE
$DELETE ACTNAM,...
$DSK ACTNAM
$DSK GRP:NAME
$EXAMINE GRP,LOC
$INSERT GRP
$INSERT GRP:NAME,...
$LOAD DEV:FILENM.BN
$LOAD ACTNAM
$NAME ACTNAM=NEWNAM
$PRINT
$QLIST
$REPLACE ACTNAM,...=GRP:NEWNAM,,,
$SIZE ACTNAM
$SIZE ACTNAM=VALUE
$SYS GRP
$SYS GRP:NAME,...
$UNLOAD GRP
$UNLOAD GRP:NAME,...

```

@ERRORS:

?BAD ARG	NO DEVICE NAME IN LOAD COMMAND
?BAD INPUT	INPUT NOT A VALID BINARY FILE
?BAD LOAD	BINARY HANDLER NOT IN CORRECT FORMAT
?BAD ORIGIN	ORIGIN IN BINARY FILE NOT IN RANGE 200-577
?CORE	NOT ENOUGH MEMORY AVAILABLE
?DSK	DSK IS NOT FILE STRUCTURED
?HANDLERS	MORE THAN 15 HANDLERS ARE ACTIVE
I/O ERROR	ERROR DURING LOAD
?NAME	MISSING NAME
NO ROOM	TOO MANY DEVICE HANDLERS LOADED
NAME NOT FOUND	DEVICE OR FILE NAME NOT FOUND
?PLAT	TOO MANY PLATTERS SPECIFIED FOR DEVICE
?SYNTAX	BAD SYNTAX
?SYS	HANDLER IS NOT A SYSTEM HANDLER OR TWO SYSTEM HANDLERS ARE ACTIVE OR HANDLER CORESIDENT WITH NON-ACTIVE SYS
SYS ERROR	I/O ERROR OCCURED WITH SYSTEM HANDLER. PRESS CONTINUE TO RETRY
SYS NOT FOUND	NO ACTIVE HANDLER BY NAME OF SYS DURING BOOTSTRAP

### **CREF.SV PAL Cross Reference**

#### @CALLING COMMANDS:

```
.CREF DEV:BINFILE.BN,DEV:LISTFILE.LS,DEV:TEMPFILE.TM<DEV:INFILE.PA
.CREF DEV:INFILE.PA /LISTING TO LINE PRINTER ONLY
.CREF INFILE.PA /LISTING TO LINE PRINTER, INPUT FILE ON DSK
```

#### @SWITCHES:

```
/E DON'T DELETE CREFLS.TM
/M MAMMOTH (TWICE AS MANY SYMBOLS, TWICE AS SLOW)
/P NO PASS 1 LISTING
/Q SABR
/R RALF
/U NO LISTING OR SYMBOL TABLE
/X NO LITERALS
```

### **EDIT.SV Editor**

#### @CALLING COMMANDS:

```
.EDIT DEV:OUTFILE.PA<DEV:INFILE.PA
.CREATE OUTFILE.PA
```

#### @SWITCHES:

```
/A RETURN TO EDITOR ON CLOSE
/B CONVERT 2 OR MORE SPACES TO TAB
/D PREDELETE
```

#### @ERRORS:

```
?0 INPUT ERROR
?1 OUTPUT ERROR
?2 CLOSE ERROR
?3 OPEN ERROR
?4 COULDN'T LOAD DEVICE HANDLER
```

#### @INTERNAL COMMANDS:

```
A APPEND TEXT
B LIST # OF CORE LOCATIONS LEFT
C CHANGE TEXT
D DELETE TEXT
E OUTPUT BUFFER, TRANSFER REST OF DATA, AND CLOSE
F AFTER J, SEARCH FOR NEXT OCCURRENCE OF SAME STRING
G GET AND LIST TAGGED LINE
I INSERT
```

```

J    INTER-BUFFER STRING SEARCH
K    KILL BUFFER
L    LIST TEXT
M    MOVE TEXT
N    WRITE BUFFER, KILL AND READ NEXT PAGE
P    WRITE TEXT BUFFER TO OUTPUT
Q    IMMEDIATE END OF FILE
R    READ TEXT FROM INPUT DEVICE
S    CHARACTER SEARCH
T    PUNCH TRAILER TAPE
V    PRINT ON LP08
Y    INPUT TEXT PAGE, NO OUTPUT
#    PRINT VERSION NO.

```

### **EPIC.SV Edit Punch and Compare**

#### @SWITCHES:

```

/0$    PAPER TAPE I/O TO/FROM OS/8 FILES
/E     DON'T PUNCH EOT
/H     SET HIGH BIT=N
/L     LOW SPEED
/P     PUNCH PATCH
/Z     REL BLOCK=0
=N     REL BLOCK TO PATCH
NO OUT FILE IS READ

```

```

FILE</1$    EDIT 'FILE'
C          CURRENT STATUS
E          EXIT TO CD
O,N       OPEN BLOCK N
R,N       READ BLOCK N
S,N,M     SEARCH FOR N WITH MASK M
W          WRITE

```

```

FILE1<FILE2/2$  COMPARE FILE1 AND FILE2
/A          ABORT
/B          BAD BLOCKS ONLY

```

### **LOAD.SV Fortran IV loader**

#### @CALLING COMMANDS:

```
.LOAD DEV:OUTFILE.LD<DEV:INFILE1.RL,...
```

#### @SWITCHES:

```

/C     MORE INPUT TO LOAD
/G     CHAIN TO RUN-TIME SYSTEM
/L     ACCEPT LIBRARY FILE
/O     MORE OVERLAYS
/S     SYMBOL MAP
/U     IGNORE RULES GOVERNING SUBROUTINE CALLS BETWEEN OVERLAYS

```

### **LOADER.SV Fortran II Loader**

#### @CALLING COMMANDS:

```
.LOAD MAPFILE.MP<INFILE.RL,...
```

#### @SWITCHES:

```

/G     GO
/H     2 PAGE HANDLERS
/I     OS/8 FILE INPUT
/L     1ST INPUT FILE IS LIBRARY FILE

```

```

/M      PRODUCE MAP
/O      OS/8 FILE OUTPUT
/P      OUTPUT COUNT OF FREE PAGES
/R      RESTART
/U      OUTPUT UNDEFINED SYMBOLS
/N      LOAD IN FIELD N (0-7) OR HIGHER
=N      SET STARTING ADDRESS

```

### **BITMAP.SV Generate map of locations used by binary file**

#### @CALLING COMMANDS:

```
.MAP MAPFILE.MP<INFILE.BN,...
```

#### @SWITCHES:

```

/N      FORCES MAPPING OF ALL FILES TO FIELD N (0-7)
/R      RESET INTERNAL MAP
/S      ALLOW MULTIPLE BINARIES PER FILE
/T      INVERT TTY-STYLE OUTPUT SWITCH

```

### **ODT Octal Debugging Technique**

#### @CALLING COMMANDS:

```
.ODT
```

#### @INTERNAL COMMANDS:

```

NNNNN/  OPEN LOC
/        REOPEN LAST OPENED LOC
NN<CR>  DEPOSIT NN IN OPEN LOC, CLOSE LOC
NN<LF>  DEPOSIT NN IN OPEN LOC, CLOSE LOC, OPEN AND DISPLAY NEXT LOC
NN;...  DEPOSIT NN IN OPEN LOC, CLOSE AND OPEN NEXT LOC
<CR>    CLOSE PREVIOUSLY OPENED LOC
<LF>    CLOSE LOC, OPEN NEXT LOC
N+      OPEN CUR LOC+N
N-      OPEN CUR LOC-N
^       CLOSE LOC, OPEN LOC ADDRESSED BY CONTENTS
_       CLOSE LOC, OPEN POINTED TO BY CONTENTS
_NNG    GO
NNB     ESTABLISH BREAKPOINT
B       REMOVE BREAKPOINT
A       OPEN AC
L       OPEN LINK
C       CONTINUE FROM BREAKPOINT
NNC     CONTINUE, ITERATE NN TIMES
M       OPEN SEARCH MASK
<LF>    OPEN LOWER SEARCH LIMIT
<LF>    OPEN UPPER SEARCH LIMIT
NNW     SEARCH CORE FOR NN MASKED BETWEEN LIMITS
D       OPEN DATA FIELD (0010=FIELD 1)
F       OPEN FIELD FOR ^, _, W (0010=FIELD 1)
^0      SUPPRESS PRINTING

```

### **PAL8.SV PAL8 Assembler**

#### @CALLING COMMANDS:

```

.PAL DEV:BINFILE.BN,DEV:LISTFILE.LS,DEV:TEMPFILE.TM<DEV:INFILE.PA/C
.PAL DEV:BINFILE.BN,DEV:LISTFILE.LS<DEV:INFILE.PA
.PAL INFILE.PA /FROM DSK

```

```
.HELP PALERR PRINTS PAL8 ERROR MESSAGES
```

#### @SWITCHES:

```

/B      TREAT ! AS BYTE SHIFT
/C      CHAIN TO CREF
/D      DDT-COMPATIBLE SYMBOL TABLE
/E      ENABLE ERROR MSG ON LINK GENERATED
/F      DISABLE 0-FILL IN TEXT
/G      GO
/H      NON-PAGINATED OUTPUT
/J      DON'T LIST LINES CONDITIONALIZED OUT
/K      USE EXTRA CORE
/L      LOAD
/N      NO LISTING
/O      DISABLE ORIGIN 200 AFTER FIELD
/S      NO SYMBOL TABLE
/T      NO FORM FEEDS
/W      DON'T REMEMBER LITERALS

```

### **PAL8.SV (ERRORS)**

#### @ERRORS:

```

BE      TABLES OVERLAPPED
CF      CREF.SV NO ON SYS:
DE      DEVICE ERROR
DF      DEVICE FULL
IC      ILLEGAL CHARACTER
ID      ILLEGAL REDEFINITION
IE      ILLEGAL EQUALS
II      ILLEGAL INDIRECT
IP      ILLEGAL PSEUDO-OP
IZ      ILLEGAL PAGE ZERO REF
LD      SYS:ABSLDR.SV NOT FOUND
LG      LINK GENERATED
PE      PAGE EXCEEDED
PH      END OF SOURCE CONDITIONALIZED OUT
RD      REDEFINITION
SE      SYMBOL TABLE EXCEEDED
UO      UNDEFINED ORIGIN
US      UNDEFINED SYMBOL
ZE      PAGE 0 EXCEEDED

```

### **PIP.SV Peripheral Interchange Program**

#### @SWITCHES:

```

/A      ASCII MODE
/B      BINARY MODE
/C      ELIM TRAILING BLANKS
/D      DELETE OUTPUT FILE BEFORE TRANSFER
/G      IGNORE ERRORS
/I      IMAGE MODE
/O      OKAY TO COMPRESS OR ZERO
/S      SQUISH
/T      CONVERT TABS TO SPACES, ETC.
/V      VERSION #
/Y      COPY SYSTEM HEAD
/Z      ZERO OUTPUT DIRECTORY BEFORE TRANSFER
=N      # OF ADDITIONAL INFO WORDS (/Z OR /S)
=N      SIZE TO CLOSE OUTPUT FILE (/I)

```

### **PIP10.SV PDP-10 Peripheral Interchange Program**

#### @SWITCHES:

```

/B      BINARY MODE
/D      DELETE OLD OUTPUT FILE BEFORE TRANSFER

```



```

/F      FAST PDP-10 DIRECTORY
/I      IMAGE MODE
/L      LIST PDP-10 DIRECTORY
/P      PRESERVE LINE NUMBERS
/Z      ZERO PDP-10 DIRECTORY BEFORE TRANSFER

```

### **SABR.SV Symbolic Assembler for Binary Relocatable Programs**

#### @CALLING COMMANDS:

```

.COMPILE BINFILE.RL,LISTFILE.LS,MAPFILE.MP<INFILE.SB
.EXE DEV:BINFILE.RL,DEV:LISTFILE.LS,DEV:MAPFILE.MP<DEV:INFILE.SB

```

#### @SWITCHES:

```

/F      INPUT IS FROM FORT
/G      CHAIN TO LOADER AND GO
/L      CHAIN TO LOADER
/N      NO LISTING
/S      NO SYMBOL TABLE

```

#### @ERRORS:

```

A      WRONG NO. OFARG'S
C      BAD CHAR
D      I/O ERROR
E      NO END STMT
I      ILLEGAL SYNTAX
L      SYS:LOADER.SV NOT FOUND
M      MULTIPLY DEFINED SYMBOL
S      SYMBOL OVERFLOW
U      UNDEFINED SYMBOL

```

### **SET.SV**

#### @CALLING COMMANDS:

```

.SET DEV PARAMETER(S)
.SET DEV NO PARAMETER(S)

```

#### @PARAMETERS:

```

HANDLER      CHANGE DEVICE HANDLERS WITHIN BUILD
READONLY     DECLARE DEVICE TO BE READ ONLY
FILES        DECLARE DEVICE TO BE FILE STRUCTURED
DVC          CHANGE DEVICE CODES
VERSION X    CHANGE VERSION
LOCATION N[=M] EXAMINE OR CHANGE LOCATIONS
LV8E        DECLARE LINE PRINTER TO BE AN LV8E
LA8A        DECLARE LINE PRINTER TO BE LA180 ON DKC8-AA
LA78        SAME AS .SET LPT NO LA8A
WIDTH N     SET WIDTH OF LINE PRINTER OR TTY
LC          DECLARE LINE PRINTER OR TTY TO HAVE LOWER CASE
ECHO        RESTORE TTY CHARACTER ECHOING
PAGE        RESTORE TTY ^S AND ^Q FACILITIES
TAB         IN TTY PRINT TABS (DON'T SIMULATE WITH SPACES)
FILL        IN TTY APPEND FILL CHARACTERS AFTER TABS
FLAG        IN TTY FLAG LOWER CASE CHARACTERS
SCOPE       ERASE CHARACTER ON TTY RUBOUTS
ESC         PRINT ESC(ASCII 033) WITHOUT CONVERTING IT TO $ SIGN
ARROW       PRINT CONTROL CHARACTERS WITH UP ARROW (E.G. ^C, ^S)
HEIGHT [M]  SET TTY SCREEN HEIGHT
PAUSE [N]   SET TTY PAUSE TIME
COL N       SET DIRECT TO USE N COLUMNS (.SET TTY COL 2)
CODE N      CHANGE TTY IOTS OR CARD READER CODES
PARITY EVEN/ODD SET MAGTAPE PARITY
OS8         DECLARE SYS TO BE OS/8
OS78        DECLARE SYS TO BE OS/8

```

INIT XXXXX CAUSE SYS TO EXECUTE XXXXX ON BOOTSTAPPING

### **SRCCOM.SV Source Compare Utility**

#### @CALLING COMMANDS:

.COMPAR DEV:OUTFILE.PA<DEV:INFILE1.PA,DEV:INFILE2.PA  
 .COMPAR OUTFILE.PA<INFILE1.PA,INFILE2.PA /FILES ON DSK

#### @SWITCHES:

/B COMPARE BLANK LINES  
 /C DON'T COMPARE (SLASHED) COMMENTS  
 /S DON'T COMPARE TABS AND SPACES  
 /T CONVERT TABS TO SPACES ON OUTPUT  
 /X DON'T COMPARE OR PRINT COMMENTS

#### @ERRORS:

?0 INSUFFICIENT CORE  
 ?1 INPUT ERROR FILE 1 (OR LESS THAN 2 INPUT FILES)  
 ?2 INPUT ERROR FILE 2  
 ?3 OUTPUT FILE TOO LARGE  
 ?4 OUTPUT ERROR  
 ?5 CAN'T OPEN OUTPUT FILE

### **BATCH.SV Run a Batch File**

#### @CALLING COMMANDS:

.SUBMIT SPOOLDEV:<INPUTDEV:FILE.BI

#### @SWITCHES:

/C CARDS  
 /E DON'T ABORT ON MONITOR, CD AND CCL ERRORS  
 /P PTR  
 /Q NO BATCH LOG  
 /H HUSH  
 /T OUTPUT TO TTY  
 /U UNATTENDED  
 /6 USE 026 CARD CODES

### **TECO.SV Editor**

#### @CALLING COMMANDS:

.TECO DEV:OUTFILE.PA<DEV:INFILE.PA  
 .TECO FILE.PA /ON DSK  
 .MAKE DEV:OUTFILE.PA  
 .MAKE OUTFILE.PA /ON DSK  
 .MUNG DEV:INFILE.PA,TECO MACRO ARGUMENT TEXT

#### @ERRORS:

?ILL ILLEGAL COMMAND  
 ?UTC UNTERMINATED COMMAND  
 ?IQN ILLEGAL Q-REGISTER NAME  
 ?PDO INTERNAL PUSH DOWN OVERFLOW (RECURSION)  
 ?MEM MEMORY OVERFLOW  
 ?STL SEARCH STRING TOO LONG  
 ?ARG ARGUMENT ERROR  
 ?IFN ILLEGAL FILE NAME  
 ?SNI SEMICOLON NOT IN ITERATION  
 ?BNI CLOSE BRACKET NOT IN ITERATION  
 ?POP POINTER OFF PAGE  
 ?QMO Q-REGISTER OVERFLOW  
 ?UTM UNTERMINATED MACRO

```
?OUT    OUTPUT ERROR
?INP    INPUT ERROR
?FER    FILE ERROR
?FUL    OUTPUT COMMAND WOULD HAVE OVERFLOWED
?NAY    NEGATIVE ARGUMENT TO Y
?IEC    ILLEGAL E CHARACTER
?IQC    ILLEGAL " CHARACTER
?NAE    NO ARGUMENT BEFORE =
?NAU    NO ARGUMENT BEFORE U
?NAQ    NO ARGUMENT BEFORE "
?SRH    FAILING SEARCH
?NAP    NEGATIVE OR 0 ARGUMENT TO P
?NAC    NEGATIVE ARGUMENT TO ,
?NIC    NEGATIVE OR 0 ITERATION COUNT
?NAS    NEGATIVE OR 0 COUNT TO SEARCH
?WLO    CAN'T WRITE OUT ERROR MESSAGE OVERLAY
?NFO    NO FILE FOR OUTPUT
```

[More TECO information](#)

### **FOTP.SV**

#### @CALLING COMMANDS:

```
.COPY DEV:OUTFILE.EX<DEV:INFILE.EX      /* IS WILD NAME OR EXTENSION
.REN DEV:NEWFILE.EX<DEV:OLDFILE.EX      /* IS WILD CHARACTER
.DEL DEV:FILE.EX
.LIST DEV:FILE.EX                        /= .COPY LPT:<DEV:FILE.EX
.TYPE DEV:FILE.EX                        /= .COPY TTY:<DEV:FILE.EX
```

#### @SWITCHES:

```
/C      MATCH ONLY FILES WITH CURRENT DATE
/D      DON'T TRANSFER (I.E. AT MOST ONLY DELETE)
/F      REQUEST NEW DEVICE IF OUT OF ROOM
/L      TYPE LOG OF INPUT FILENAME MATCHES (*)
/N      NO PRE-DELETE
/O      MATCH ONLY FILES WITH OTHER THAN TODAY'S DATE
/Q      QUERY USER ABOUT FILE BEFORE OPERATION (*)
/R      RENAME
/T      USE TODAY'S DATE
/U      TREAT EACH INPUT SPECIFICATION SEPARATELY
/V      MATCH FILES NOT OF FORM SPECIFIED
/W      PRINT VERSION #
```

#### NOTES:

(\*) /D CAUSES LOG OF OUTPUT FILES (IF /L ALSO)

IF INDEV: EQUALS OUTDEV:, THEN /N IS FORCED.

IF NO INPUT FILE, \*.\* IS FORCED EXCEPT FOR /D  
IF OUTPUT DEVICE SPECIFIED, BUT NO FILE, \*.\* IS ASSUMED.

```
^P      ABORT OPERATION, FIX OUTPUT DIRECTORY
^C      FIX OUTPT DIRECTORY, RETURN TO OS/8
^O      SUPPRESS TYPEOUT
```

### **KEYBOARD MONITOR AN OTHER COMMANDS**

#### @CALLING COMMANDS:

```
.ASSIGN DEV NAME          /ASSIGN NAME TO DEVICE
.DAY DD-MON-YY           /ENTER DATE INTO SYSTEM
.DEASSIGN                 /DEASSIGN LOGICAL DEVICE NAMES
.GET DEV FILE.EX         /LOAD CORE IMAGE
```

```
.MEMORY N           /SPECIFY HIGHEST MEMORY FIELD AVAILABLE
.R FILE             /EXECUTE FILE.SV FROM SYS
.RUN DEV FILE.EX    /EXECUTE FILE.EX FROM THE DEVICE
.SAVE DEV FILE.EX   /SAVE CORE IMAGE
.SQUISH DEV:        /COMPRESS FILE STORAGE ON DEVICE
.START FNNNN        /START EXECUTION
.UA COMMAND         /SAVE COMMAND(.UA<CR> EXECUTES IT)
.ZERO DEV:          /ZERO DEVICE'S DIRECTORY
```

### **CONVRT.SV**

SAVE FORMAT - BINARY FILE PUNCHER

THIS PROGRAM PRODUCES AN ABSOLUTE BINARY FILE FROM AN OS/8 SAVE FILE. ONLY THOSE AREAS OF MEMORY SPECIFIED IN THE CORE CONTROL BLOCK ARE ACTUALLY PUNCHED.

COMMAND DECODER RESPONSE IS:

OUTPUT < INPUT, INPUT .....

DEFAULT INPUT EXTENSION IS ".SV", DEFAULT OUTPUT IS THE HIGH SPEED PAPER TAPE PUNCH.

### **FUTIL.SV**

FUTIL IS AN OS-8 UTILITY PROGRAM THAT IS EXTREMELY USEFUL

I F Y O U K N O W W H A T Y O U ' R E D O I N G  
 --       -----       -----       -----       -----

IT IS EQUALLY AS DANGEROUS IF YOU DON'T !!!!!

FUTIL IS TOO COMPLICATED TO EXPLAIN IN A SIMPLE HELP FILE, SO PLEASE READ ABOUT IT IN THE OS-8 HANDBOOK UPDATE (DEC-S8-OSHBA-A-DN4) APPENDEX K.

ONE ESPECIALLY HANDY FEATURE IS THE ABILITY TO "SCAN" A GIVEN DEVICE TO ESTABLISH IF THERE ARE ANY "BAD BLOCKS". THIS IS SIMILIAR TO RT-11 PIP OPTION K.

[Futil documentation](http://www.pdp8.net/os/os8/os8_cmd.shtml)

### **SORT.SV - ASCII FILE SORTER**

**SORT:** SORTS INFIMATE NUMBER OF LINES IN ASSENDING OR DESENDING SEQUENCE ACCORDING TO ASCII CHARACTER SET, FAST! (TIME IN SECONDS TO SORT FILE OF N BLOCKS APROX= 3 + N/3 + N/10\*(LOG10(N/10)) ).

**TO RUN:** .R SORT  
 \* OUTPUT < ORDER, INFIL [,INFIL ...]

**WHERE:** OUTPUT WILL CONTAIN SORTED FILE  
 INFIL IS THE UNSORTED INPUT FILE  
 ORDER SPECIFIES THE COLUMNS (IN DECIMAL, 1 IS LEFTMOST) TO BE SORTED;  
 POSITIVE FOR ASSENDING, NEGATIVE FOR DECENDING; MAY USE FROM-TO:  
 EG: "1-10, -12-19, 11, -20, 25-21<CR>"

**ERRORS:**

USER ERROR 2 SYS: I/O (HANDLER, FETCH, ETC)  
 USER ERROR 3 INPUT DEVICE I/O  
 USER ERROR 4 OUTPUT DEVICE I/O  
 USER ERROR 5 BAD NUMBER IN ORDER FILE  
 USER ERROR 6 NO INFIL  
 USER ERROR 7 TO MANY NUMBERS IN ORDER FILE  
 USER ERROR 8 NUMBER IN ORDER FILE TOO LARGE  
 USER ERROR 9 LINE TOO LONG IN INFIL (128. CHAR MAX)  
 OR A "^Y" IN INFIL

**DIRECT.SV****@CALLING COMMANDS**

.DIR DEV:LISTFILE.DI<DEV:FILETYPE /\* IS WILD NAME OR EXTENSION  
 .DIR FILETYPE /\* IS WILD CHARACTER

**@SWITCHES:**

/A ALPHABETIZE DIRECTORY BEFORE PROCESSING  
 /B INCLUDE STARTING BLOCK NUMBERS (OCTAL)  
 /C LIST ONLY FILES WITH THE CURRENT DATE  
 /D SORT FILES BY DATE, THEN ALPHABETICALLY  
 /E INCLUDE EMPTIES IN THE LISTING  
 /F FAST MODE (NO LENGTHS OR DATES)  
 /H LIST CONTENTS OF THE HEADER BLOCK  
 /I PRINT ADDITIONAL INFORMATION WORDS  
 /M LIST ONLY THE EMPTIES  
 /N PRINT DATES IN NUMERIC FORM (MM/DD/YY)  
 /O LIST ONLY FILES WITH OTHER THAN TODAY'S DATE  
 /P ELIMINATE PAGING (NO FORMFEEDS IN THE OUTPUT)  
 /R LIST REMAINDER OF FILES AFTER THE FIRST ONE  
 /T PRINT THE MOST TIMELY (RECENT) FILES FIRST  
 /U TREAT EACH INPUT SPECIFICATION SEPARATELY  
 /V LIST FILES NOT OF THE FORM SPECIFIED  
 /W PRINT THE VERSION NUMBER  
 /X ALPHABETIZE BY EXTENSION, THEN BY NAME  
 =N USE N COLUMNS - OUTPUT IS ORDERED BY COLUMN

**OCOMP.SV - COMPARE & DUMP FILES UNDER A MASK**

DEFAULT OUTPUT IS TO TTY:

DEFAULT OUTPUT EXTENSION IS .LS.

THE MASK IS SPECIFIED AS THE LEAST SIGNIFICANT 12 BITS OF =.

**1) COMPARE:**

TWO INPUT FILES ARE COMPARED UNDER A MASK.  
 IF MASK=0, USE 7777.

**2) DUMP:**

ONE INPUT FILE.  
 DUMP ALL WORDS WITH NO MASK BITS OFF.  
 IF MASK=0, USE 7777.

**3) SEARCH:**

ONE INPUT FILE WITH /S.  
 DUMP ALL WORDS EQUAL TO THE MASK.

AFTER SPECIFIED OPERATIONS ARE COMPLETED, THE COMMAND DECODER IS RECALLED, UNLESS INPUT WAS TERMINATED BY AN ALT MODE CHARACTER.

OTHER COMMAND DECODER OPTIONS:

/I IMAGE MODE

INPUT FILE IS CORE IMAGE (SAVE) FILE.  
 ADDRESSING IS BY MEMORY ADDRESS, RATHER THAN FILE ADDRESS.  
 DEFAULT INPUT EXTENSION IS .SV.

/C CCB OUTPUT  
 WHEN COMBINED WITH /I, CCB IS PRINTED AS PART OF THE RUN TITLE.

/D DIFFERENCE MODE  
 ONLY OUTPUT IS A MESSAGE TELLING THAT COMPARED FILES ARE  
 SAME OR DIFFERENT.

/F FORM FEED  
 ALL BLOCKS WHICH CAUSE OUTPUT HAVE THEIR OUTPUT PRECEDED  
 BY A FORM FEED CHARACTER.

/T TELETYPE  
 OUTPUT FORMAT IS NARROWED FOR OUTPUT TO 72 COLUMN TELETYPES.  
 THIS IS ALSO USEFUL FOR SCOPES.

/H HEADER  
 OUTPUT HEADER LINE EVEN IF OUTPUT DEVICE IS TTY:.

ERRORS ARE SIGNALLED BY "USER ERROR ..." MESSAGES

ERROR	CONDITION
1	MORE THAN TWO INPUT FILES.
2	OUTPUT FILE TOO SMALL.
3	OUTPUT HANDLER FETCH OR ENTER FAILED.
4	INPUT HANDLER FETCH OR LOOKUP FAILED.
5	INPUT FILE #1 ERROR.
6	INPUT FILE #2 ERROR.
7	OUTPUT FILE ERROR.
8	OUTPUT FILE CLOSE FAILED.
9	CCB DECODE ERROR.

### HELP.SV

@CALLING COMMANDS:

.HELP	List CCL Commands
.HELP COMMAND	Help on Command

### RXCOPY.SV RX Floppy Copy

@CALLING COMMANDS:

.DUPLIC OUTDEV:<INDEV:

@SWITCHES:

/C	Default, copy and match
/D	Format output diskette to double-density (RX02)
/M	Match with no implied copy
/N	Copy with no implied match
/P	Pause before and after accessing disk
/R	Read output device with no implied copy or match
/S	Format output diskette to single-density (RX02)
/V	PRINT VERSION NUMBER

Not all switches will work with all versions of RXCOPY

**DECX8 SYSTEM EXERCISER**

REFERENCE THE FILE [DECX8.TX](#) FOR ADDITIONAL INFORMATION.

ALSO, THERE ARE DOCUMENTS ON THE MICROFICHE LIBRARY FOR BUILDING AND INITIALIZING DECX8.

**ALIGNX - RK05 ALIGNMENT UTILITY**

To run this utility, type:

```
.R ALIGNX.DG
```

for a full explanation of the utility and a step-by-step procedure for aligning an RK05, type:

```
.LIST RKB0:ALIGN.TX
```

**RXREAD - RX01 Floppy read check utility**

This program is used to read a RX01 floppy disk and check it for any errors. The error report consists of the following:

- The track and sector number (in octal) of the bad sector
- The RX01 error and status (RXES) register contents
- The explanation of the RXES value (for example, if the error is a CRC error, it will say "CRC Error")
- The RX Definitive error code, and its description.

After an error, the program will ask if it is to re-write the bad sector. This question should be answered with "N" (no), unless the error was a CRC error. In this case, the sector can be re-written, which may (but may not) correct the error. In any case, the data is re-written with a good CRC. This may make at least part of a bad block usable.

**QUIT.SV - Directory Backup Program**

By typing:

```
".QUIT" , you can backup your Directory to a file called  
"DIRECT.TS" on RKB0:.
```

If the disk crashes, and the directory area is destroyed, it may be restored by booting to the "B" area of the disk, and using FUTIL to write this file back onto the "A" area as the new directory.

PLEASE NOTE that this is for extreme cases only ! Read the section on Advanced Techniques on the file "DIAG.TX" on RKB0: before attempting this procedure.

Periodic use of this program is an essential part of the housekeeping discipline that allows use of the Disk Recovery Techniques described on the file "DIAG.TX"

### RKCOPY      RK05 DISC COPY PROGRAM

This utility, written by Dick Murphy, copies and verifys RK05 packs between any drives. Since this is a sector by sector copy, it will copy any operating system . When called, the dialog is self explanatory.

To run type:

```
.R RKCOPY
```

### RKUTIL      RK05 UTILITY PROGRAM

This utility has several handy options including a

- Disk Copy Program that will copy & verify any 16 sector disk
- Disk Read Program that will verify that a disk has no CRC errors
- Disk Formatter Program to format new disks
- Disk Confidence Test that runs data patterns on a suspect disk

### HEADER.SV

```
.R HEADER
```

```
DEVICE NAME? SYS or SYS: or DSK or RKB1 or RXA1: etc.
DEVICE LENGTH? 737 or 3248 or 494 or (incl. directory blocks)
```

```
if the device contains a system area the following applies:
SYSTEM ID? OS/8, OS/12, OS/78, DS/8, or PS/8
VERSION NO.? 3 or whatever
RELEASE CODE? C or D or .....
if there is no system area the questions above are skipped
```

```
LOGICAL DEVICE NAME? CHEM or BOB: or UWF or ?? (1-4 letters)
VOLUME NO.? 1 or any number up to 4095; 0 = none
DESCRIPTIVE LABEL?
```

```
Here you may enter any identifying information (UPPER/lower
case, as you wish) - up to 128 characters, including CR/LF.
Terminate input with a CTRL/Z. Rubout and linefeed work
in their normal OS/8 fashion.(^Z)
```



**LOADING THE INUSE MACRO**

```
.R TECO
*ERINUSE.TE$YHXIHK$MI$$
```

**"WILD.TE" - Wildcard Macro**

The Wildcard macro, with the new version of CCL, allows the use of "\*" and "?" in otherwise illegal commands.

```
ex: .MAC RKB0:<RKB0:CCL???.MA
```

compiles all the files of the form "CCL.MA" on RKB0:

```
also .MUNG WILD,PRINT *.LS prints all the .LS files, etc.
```

"WILD.TE" is automatically called by:

```
COMPILE          PAL          MAC          LINK
EXECU
```

**VTEDIT.TE TECO VT52 Editor Macro**

KEYPAD LAYOUT for VTEDIT for VT52

"BLUE" SAVE TEXT*	"RED" TECO COMMAND	"GREY" UNSAVE TEXT	"^" UP IN COLUMN*
"7" OPEN LINE*	"8" PAGE*	"9" QUOTE NEXT*	"V" DOWN IN COLUMN*
"4" UP LINE*	"5" DELETE* CHARACTER	"6" DELETE LAST	">" CURSOR RIGHT*
"1" TOP OF PAGE	"2" BOTTOM OF PAGE	"3" START OF LINE	"<" CURSOR LEFT*
"0" DOWN LINE*	"." SEARCH*	"ENTER" SEARCH ARGUMENT*	

OTHER KEYS

```
CTRL/C  Exit from Macro
CTRL/D  Kill rest of line *
CTRL/K  Kill line *
CTRL/N  File search
CTRL/U  Kill start of line
[ESC] CTRL/P  Exit & execute
[ESC] CTRL/Z  Exit & close file
```

```
BK SP   Go to end of line *
DELETE  Delete previous *
```

**ARGUMENTS**

All starred (\*) commands  
Optionally take an argument  
entered as [ESC] [digits]

**WPFLOP User's Guide**

This document describes the operation of the word processing to OS/8

conversion program, WPFLOP.

## USAGE

WPFLOP is used to transfer documents from word processing floppy disks to OS/8 media or from OS/8 media to word processing diskettes. The WPS floppy is accessed using the COS compatible floppy handlers included in this package. There are various modes of conversion which will be explained later.

## OPERATION

### WPS->OS/8

Before using WPFLOP to access a WPS floppy, one must know the number of the document which is to be read from the WPS floppy. For example, in the sample index below, the column labeled "Document number" is the number that WPFLOP uses to reference a document.

-- DOCUMENT INDEX --

Drive: 3, Name: DOC3 , # of Docs: 3, Blocks left: 621 (of 627)

```
-----
-
Document                               Elapsed
Time                                     Last
Number  Name                Created   Modified      Size  Version  Time
Total
-----
-
 3     Floppy Index
      12/7/79  NO/DA/TE         3    1      0:00  0:00
 2     WPFLOP User's Guide
      12/7/79  12/7/79 13:09    2    1      0:07  0:07
 1
      12/3/79  12/7/79 13:10    1    4      0:00  0:00
```

The document "WPFLOP User's Guide" (this file) is document number 3. This is the number that WPFLOP uses to access the file.

Before running WPFLOP, the COS mode floppy handlers must be installed in your OS/8 system. See the BUILD chapter in the OS/8 System Reference Manual for details on using BUILD. You can type ".HELP COSHAN" for further information on the handlers. This document assumes that the handlers have been installed with names of "RXB0:" for drive zero, and "RXB1:" for drive one.

Another method for installing the BYTE mode handlers is via the SET HANDLER command. To enable BYTE mode, type ".SET HANDLER FLOP = BYTE" and to return to normal mode, ".SET HANDLER BYTE = FLOP".

To copy the document "WPFLOP User's Guide" to a file on the OS/8 system device called WPFLOP.DC, use the following commands:

```
.R WPFLOP
```

```
*WPFLOP.DC<RXB0:2$      (Assumes WPS floppy in drive 0. $=ESC,
                          ALT or SEL, depending on the terminal.)
```

If no OS/8 output file is specified, the program assumes that the user wants the output on the console terminal (TTY:). Therefore, to print the same document on the console:

```
.R WPFLOP
```

```
*RXB0:2$                The "2" here is the number of the WPS
                          document which is to be transferred.
```

The general WPS->OS/8 command syntax is as follows:

```
*[dev:file.ex]<wdev:docn[/S][/S...][$]
  |         |         |         |         |
  ^         ^         ^         ^         ^
  |         |         |         |         |
  |         |         |         |         |_ ALT or ESC to exit when done.
  |         |         |         |         |
  |         |         |         |         |_ Optional switches
  |         |         |         |         |
  |         |         |         |         |_ WPS Document number
  |         |         |         |         |
  |         |         |         |         |_ Name for the OS/8 handler for the WPS floppy
  |         |         |         |         |
  |         |         |         |         |_ OS/8 Output file specification
  |         |         |         |         |
  |         |         |         |         |_ OS/8 output device
```

The asterisk (\*) is printed by the OS/8 command decoder. The items enclosed in braces [] are optional. For example, if no OS/8 output file is specified, TTY: is assumed. If the command line ends with an altmode, the program returns to OS/8 when it is finished. If it does not, the program responds with another "\*" and inputs another command string.

There are two options accepted when transferring from WPS to OS/8. These are the version switch (/V) and the compatibility switch (/C). If the version switch is in the command line, WPFLOP prints its version number before executing the command. If the compatibility mode switch is present, WPFLOP transfers the WPS document in a form compatible with OS/8 system programs. In this mode, underlines, dead keys, and bolding are ignored, but tabs are converted to tabs (ASCII 11). Also, the file is not justified. In normal mode, the output file contains overprinting as necessary to duplicate the WPS file.

Compatibility mode is used to transfer a WPS file to OS/8 in a form that can be used by, for example, OS/8 BASIC. This allows WPS to be used to enter and edit a BASIC program. To do this, create the file under WPS and transfer it to OS/8 using compatibility mode. If there are any errors, transfer it back to WPS, edit it there, and then back to OS/8, etc.

## OS/8->WPS

To transfer an OS/8 File to WPS, use the following syntax:

```
*wdev:<dev:file.ex/0[/S][$]
```

Note that this is similar to the above example. The most important differences are the output switch (/O) and the fact that no WPS document number is given in the command line. WPFL0P will use the next available document for the new file. The number that WPFL0P uses will be reported as a message of the form:

```
"Creating document number:n"
```

Where "n" is the number of the document created.

The /O switch is required to specify to WPFL0P that the output file is on the WPS floppy. Without this switch the program will assume that the input file is the WPS file.

The other switch allowed is the /W (word-wrap) switch. If this switch is specified, the returns in the OS/8 file will be converted to word-wrap returns. If there are two returns in a row, hard returns are used. Normally, all the returns in the OS/8 file are converted to hard returns.

The document created on the WPS device is not totally compatible with WPS as it is initially created. The most obvious problem is that tabs are output as only one space. To correct this, the WPS editor must be allowed to scroll thru the document. The initial ruler MUST be changed in order for the editor to correct the document. To facilitate this, the ruler that WPFL0P creates has tabs at locations 8 and 9. The first tab, at location 8, is incorrect. It is included to be deleted in order for the WPS editor to re-tabulate the document. Therefore, the correction sequence is:

- 1.Create the WPS document from OS/8
- 2.Call the WPS editor to correct the document
- 3.At the top of the document, type GOLD Ruler.
- 4.Space to column 8 with ADVANCE WORD.
- 5.Type the space bar. The first of the adjacent T's will disappear.
- 6.Type RETURN to embed the new ruler.
- 7.Type GOLD ADVANCE to scroll thru the document and correct it.
- 8.Type GOLD File to re-file the corrected document.

## RESTRICTIONS

The ruler created for the WPS document has a ragged ruler with a width of 80 columns and tabs every eight columns. This is the only ruler that can be created. However, you can change the ruler under WPS after the conversion.

The WPS to OS/8 code does not support bolded dead keys. The dead key sequence prints properly, but the sequence is not bolded.

The document created on the WPS device does not have an index entry. It will show up in the index as just the document number. To give the file a name, edit the index document,(document 1) to include an entry for the document. For example, if you created document 4, and want it to be called "Foobar", include the following line in the index:

```
<n>Foobar <#>4<>
```

The form used is:

```
<n>Name space <#>doc#<>
```

The n in the <> must be lower case, and the name must end with a space.

### **RT-11 Floppy Interchange Program**

This program is used to manipulate RT-11 floppy disk file structures under OS/8. (RX01 ONLY, RX02 Not supported). The operations available allow most of the functions of OS/8 PIP to be simulated on the RT-11 disk.

[Full instruction for RTFLOP, RTIN, RTOUT](#)

### **COS Compatible (BYTE MODE) RX01 floppy handlers for OS/8**

These files allow an OS/8 user more efficient usage of the space available on an RX01 floppy disk. The two files, RXBNS.PA and RXBSY.PA are the non-system and system handlers, respectively. They are fully compatible with the OS/8 operating system. The system handler is two pages long and follows the conventions for two page handlers that allows BASIC and FORTRAN to work properly. They allow the OS/8 diskette to hold 658 blocks as opposed to only 494 with the OS/8 handler.

The system handler can not be booted with the conventional RX01 bootstrap program. To correct this, a program called RXFIX is included. This program is run after BUILD to modify the system head (block 0) to allow bootstrap with the conventional program. It is very important that this program be run again to restore block 0 before BUILD is run from the system.

Examples on how to install the handlers:

```
.SET SYS NO INIT      !In case there is an initial command set up.
.RUN SYS BUILD
$LOAD RXBNS           !Load the handlers.
$INSERT RX8:RXB0-1    !Install the non-system handlers
$BOOT
```

```

SYS BUILT
.SAVE SYS BUILD                !Now the non-system handlers are on your OS/8
                                !system.

```

To build a system floppy, first boot your conventional OS/8 system in drive 1. Then:

```

.RUN SYS BUILD
$LOAD RXBSY                    !Load the system handler
$REPLACE SYS=BYTE,SYS         !Change the system handler
$INSERT BYTE,RXB0-1           !Install the co-resident drive 0 and 1 handlers.
$BOOT                          !Boot the new OS/8 system.
WRITE ZERO DIRECT?YES
SYS BUILT
.SAVE SYS BUILD                !Save the new build.
.RUN RXA1 FOTP                 !Copy files to the new drive
*SYS:<RXA1:BUILD.SV/V         !Except BUILD.
*^C
.R RXFIX                       !Fix the boot.

```

Before running BUILD on the new system device, run RXFIX to repair block zero:

```

.R RXFIX
.RUN SYS BUILD
.
.                               !BUILD Commands
.
$BOOT
SYS BUILT
.SAVE SYS BUILD
.R RXFIX                       !Make it bootable.

```

A patch must be installed to OS/8 PIP to allow the ZERO and SQUISH commands to work with the new handlers.

```

.GET SYS PIP
.ODT                          ! NOTE: If you are using the NED Diagnostic
13650/0000 6556                !      System, these patches are
already                          !
^C                               !      incorporated into PIP.SV
.SAVE SYS PIP

```

Since these handlers are COS compatible, they can be used to copy files to and from COS-310 diskettes. Also, COS310 FILEX can convert a COS file to an OS/8 file on these diskettes that is readable by OS/8.

### RXFIX.SV            BYTE Mode Floppy Bootstrap Fixer

The system handler can not be booted with the conventional RX01 bootstrap program. To correct this, a program called RXFIX is included. This program is run after BUILD to modify the system head (block 0) to allow bootstrap with the conventional program. It is very important that this program be run again to restore block 0 before BUILD is run from the system.

This is only done when using the BYTE Mode handler, and this action is incorporated into the Batch Streams that make the BYTE Mode Floppys.

This program acts like a flip flop; run it once and it makes the

target floppy bootable; run it again and it makes it "Buildable".

Before running BUILD on the new system device, run RXFIX to repair block zero:

```
.R RXFIX
.RUN SYS BUILD
      .
      .           !BUILD Commands
      .
$BOOT
SYS BUILT
.SAVE SYS BUILD
.R RXFIX           !Make it bootable.
```

**The macro to eliminate the imbedded "bolds" in a converted WPS to OS8 document is:**

```
<N [CR] [Rubout] ^N [LF] $R10I$>$$
```

### **SHELLY.DG or RFCRUNCH**

This program was written around 1974 by Bob Shelly of Product Support. It provides an excellent checkout of the RF08 Disk System. It is an especially good check of a TSS-8 System. It does, however need an MQ to operate. This means an 8E or later or an 8I with an EAE.

This program is best used as a GO / NO GO test since the error printouts are poor at best. The typeouts are interpreted as follows:

D

```
XXXXXXXXXXXXX = total number of errors
```

W

```
XXXXXXXXXXXXX = positive word count that should have been transferred
```

Unit	Track	Disk Mem Addr	
----	----	-----	
X	XXXXXXX	XXXXXXXXXXXXX	= address where xfer SHOULD have started
X	XXXXXXX	XXXXXXXXXXXXX	= calculated address where xfer DID start

```
XXXXXXXXXXXXX = DMA of last WRITE done
XXXXXXXXXXXXX = DMA of last READ done
-----X = bit 11 is 0 if writing; 1 if reading
```

S

```
XXXXXXXXXXXXX = status register minus photo-sync bit
```

Program Locations

```
-----
```

120 - momory where program was running  
 121 - Field disk was breaking to  
 122 - unit # and 6 bits of track address  
 123 - 7th bit of track address and DMA  
 124 -  
 125 - stall  
 126 to - next transfer

### **SDUMP.SV or SVDUMP.SV**

THIS PROGRAM DUMPS THE CONTENTS OF A SAVE FILE IN OCTAL AS A LOADED CORE IMAGE.

APPROPRIATE COMMAND DECODER RESPONSE IS

"OUTPUT < INPUT1,INPUT2,..."

DEFAULT OUTPUT IS LPT, DEFAULT INPUT EXTENSION IS .SV.

DUMP FORMAT IS NORMALLY 20 (OCTAL) WORDS WIDE, ABOUT 105 COLUMNS. SPECIFYING A "/N" OPTION CHANGES THE FORMAT TO 10 WORDS WIDE, ALLOWING IT TO FIT ON A TTY.

### **CAMP.SV Cassette & Magtape Positioner**

To call camp from the system device type:

R CAMP

Camp prints a # to indicate that it is ready to receive a command. The command line entered may be terminated with carriage return (CAMP retains control) or an ALTMODE (ESC, control returns to the keyboard monitor).

Each CAMP command begins with a keyword consisting of two or more letters. Letters that are required at in upper case.

BACKspace

EOF

HELp

REWInd

SKIP

UNload

VERsion

The following commands are supported. DEV: is the permanent name of a cassette or magnetic tape drive. The "N" is an unsigned decimal number representing the number of files or records to perform the operation over. The number must be in the range 0-4095. If no number is entered 1 is assumed.

BACKSPACE DEV: N FILES

BACKSPACE DEV: N RECORDS

EOF DEV:

SKIP DEV: N FILES

SKIP DEV: N RECORDS

SKIP DEV: EOD

REWIND DEV:

UNLOAD DEV:

VERSION

HELP

The EOF command writes a single file mark.



Feel free to contact me, David Gesswein [djg@pdp8online.com](mailto:djg@pdp8online.com) with any questions, comments on the web site, or if you have related equipment, documentation, software etc. you are willing to part with. I am interested in anything PDP-8 related, computers, peripherals used with them, DEC or third party, or documentation.

[PDP-8 Home Page](#) [PDP-8 Site Map](#) [PDP-8 Site Search](#)

Mirror site if this site is slow [highgate.comm.sfu.ca](http://highgate.comm.sfu.ca)