

```
2 ** XREF - Cross Reference Processor
3 *
4 * G. C., 80/03, for the Heath Co.
5 * W. Z., 80/06
6 *
7 * Copyright, 1980 by:
8 *
9 * Heath Co.
10 * Benton Harbor, MI
11 * 49022
12 *
13 *
14 * XREF process the XREF data file generated by
15 * ASM. The data is assumed to be in the format
16 * output by the *ESR* subroutine in ASM. It
17 * is as follows:
18 *
19 * DB '1234567' 7 byte label
20 * DW 0 statement number
21 * DB 0 reference type
22 *
```

## Assembly Constants

15:14:25 02-OCT-80

```

25 **      Assembly Constants
26 *
27
000.004    28 CN.TMP EQU    4
29
002.000    30 TMPBFL EQU    512      Temporary File Buffer Size
31
001.000    32 LSTBFL EQU    256      List File Buffer Size
33
000.005    34 CRTHL EQU    5      Cross-Reference Table History size
35
000.000    36      ORG    0
37
000.000    38 SYMBOL DS    7      Symbol Identifier
000.007    39 SYMBOLL EQU    *-SYMBOL
40
000.007    41 LINEM DS    2      Line Number
42
000.011    43 REFTYP DS    1      Reference Type
44
000.012    45 RECLEN EQU    *      Record Length

000.000    47      ORG    0
48
000.000    49 XH.LNK DS    2      pointer to next list element. (0=NIL)
50
000.002    51 XH.LIN DS    2      line number at which reference occurred
52
000.004    53 XH.FLG DS    1      reference type
54
000.005    55 XH.LEN EQU    *      XREF list entry length
000.005    57 XTEXT ASCII

59X **      ASCII CHARACTER EQUIVALENCES
60X
000.015    61X CR EQU    13      CARRIAGE RETURN
000.012    62X LF EQU    10      LINE FEED
000.200    63X NULL EQU    2000      PAD CHARACTER
000.000    64X NUL2 EQU    0
000.007    65X BELL EQU    7      BELL CHARACTER
000.177    66X RUBOUT EQU    177R
000.010    67X BKSP EQU    100      CTL-H
000.026    68X C.SYN EQU    260      SYNC
000.002    69X C.STX EQU    2      STX
000.047    70X QUOTE EQU    47Q
000.011    71X TAB EQU    11R
000.033    72X ESC EQU    33Q
000.012    73X NL EQU    12Q      NEW LINE (MDOS SYSTEMS)
000.212    74X ENL EQU    NL+200Q      NL + END-OF-LINE-FLAG
000.014    75X FF EQU    14Q      FORM FEED
000.001    76X CTLA EQU    01Q      CTL-A
000.002    77X CTLB EQU    02Q      CTL-B

```

## Definitions

ASCII

15:14:30 02-DCT-R0

000.003	78X	CTLG	EQU	030	CTL-C
000.004	79X	CTLD	EQU	040	CTL-D
000.017	80X	CTLO	EQU	170	CTL-O
000.020	81X	CTLP	EQU	200	CTL-P
000.021	82X	CTLQ	EQU	210	CTL-Q
000.023	83X	CTLS	EQU	230	CTL-S
000.032	84X	CTLZ	EQU	320	CTL-Z
000.005	85	XTEXT	HOSDEF		

87X \*\* HOSDEF - DEFINE HOS PARAMETER.

88X \*

89X

90X

000.040 91X VERS EQU 2\*16+0 VERSION 2.0

92X

000.377 93X SYSCALL EQU 3770 SYSCALL INSTRUCTION

94X

95X

000.000 96X ORG 0

97X

98X \* RESIDENT FUNCTIONS

99X

000.000 100X .EXIT DS 1 EXIT (MUST BE FIRST)

000.001 101X .SCIN DS 1 SCIN

000.002 102X .SCOUT DS 1 SCOUT

000.003 103X .PRINT DS 1 PRINT

000.004 104X .READ DS 1 READ

000.005 105X .WRITE DS 1 WRITE

000.006 106X .CONSL DS 1 SET/CLEAR CONSOLE OPTIONS

000.007 107X .CLRCD DS 1 CLEAR CONSOLE BUFFER

000.010 108X .LOADD DS 1 LOAD AN OVERLAY

000.011 109X .VERS DS 1 RETURN HDOS VERSION NUMBER

000.012 110X .SYSRES DS 1 PRECEDING FUNCTIONS ARE RESIDENT

111X

112X

113X \* \*HDOSQVLO.SYS\* FUNCTIONS

114X

000.040 115X ORG 40A

116X

000.040 117X .LINK DS 1 LINK (MUST BE FIRST)

000.041 118X .CTLG DS 1 CTL-C

000.042 119X .OPENR DS 1 OPENR

000.043 120X .OPENW DS 1 OPENW

000.044 121X .OPENU DS 1 OPENU

000.045 122X .OPENC DS 1 OPENC

000.046 123X .CLOSE DS 1 CLOSE

000.047 124X .POSIT DS 1 POSITION

000.050 125X .DELET DS 1 DELETE

000.051 126X .RENAM DS 1 RENAME

000.052 127X .SETTP DS 1 SETTOP

000.053 128X .DECODE DS 1 NAME DECODE

000.054 129X .NAME DS 1 GET FILE NAME FROM CHANNEL

000.055 130X .CLEAR DS 1 CLEAR CHAN

000.056 131X .CLEARA DS 1 CLEAR ALL CHANS



Definitions

ESVAL

15:14:33 02-DCI-80

```

179X **      S.VAL - SYSTEM VALUE DEFINITIONS.
180X *
181X *      THESE VALUES ARE SET AND MAINTAINED BY THE SYSTEM.
182X *
183X *      THE DECK HOSEQU MUST BE MODIFIED WHEN THIS IS MODIFIED.
184X
185X
040.277      186X      ORG      S.VAL
187X
040.277      188X S.DATE DS      9      SYSTEM DATE (IN ASCII)
040.310      189X S.DATC DS      2      CODED DATE
040.312      190X S.TIME DS      4      TIME FROM MIDNIGHT (IN TICS)
040.316      191X S.HMEM DS      2      HARDWARE HIGH MEMORY ADDRESS+1
192X
040.320      193X S.SYSM DS      2      FWA RESIDENT SYSTEM
194X
040.322      195X S.USRM DS      2      LWA USER MEMORY
196X
040.324      197X S.OMAX DS      2      MAX OVERLAY SIZE FOR SYSTEM
198X
199X
200X **      THE FOLLOWING FIVE CELLS SHOULD BE MODIFIED/READ ONLY VIA THE .CONSL SYSCALL
201X
000.200      202X CSL.ECH EQU      10000000B SUPPRESS ECHO
000.004      203X CSL.RAW EQU      00000100B Raw Mode I/O /80.09.ac/
000.002      204X CSL.WRF EQU      00000010B WRAP LINES AT WIDTH
000.001      205X CSL.CHR EQU      00000001B OPERATE IN CHARACTER MODE
206X
000.000      207X I.CSLMD EQU      0      S.CSLMD IS FIRST BYTE
040.326      208X S.CSLMD DS      1      CONSOLE MODE
209X
000.200      210X CTP.BKS EQU      10000000B TERMINAL PROCESSES BACKSPACES
000.100      211X CTP.FF EQU      01000000B Terminal Processes Form-Feed /80.09.ac/
000.040      212X CTP.MLI EQU      00100000B MAP LOWER CASE TO UPPER ON INPUT
000.020      213X CTP.MLO EQU      00010000B MAP LOWER CASE TO UPPER ON OUTPUT
000.010      214X CTP.2SB EQU      00001000B TERMINAL NEEDS TWO STOP BITS
000.002      215X CTP.BKM EQU      00000010B MAP BKSP (UPON INPUT) TO RUBOUT
000.001      216X CTP.TAB EQU      00000001B TERMINAL SUPPORTS TAB CHARACTERS
217X
000.001      218X I.CONTY EQU      1      S.CONTY IS 2ND BYTE
000.000      219X      ERRNZ *-S.CSLMD-I.CONTY
040.327      220X S.CONTY DS      1      CONSOLE TYPE FLAGS
000.002      221X I.CUSOR EQU      2      S.CUSOR IS 3RD BYTE
000.000      222X      ERRNZ *-S.CSLMD-I.CUSOR
040.330      223X S.CUSOR DS      1      CURRENT CURSOR POSITION
000.003      224X I.CONWI EQU      3      S.CONWI IS 4TH BYTE
000.000      225X      ERRNZ *-S.CSLMD-I.CONWI
040.331      226X S.CONWI DS      1      CONSOLE WIDTH
227X
000.001      228X CO.FLG EQU      00000001B CTL-D FLAG
000.200      229X CS.FLG EQU      10000000B CTL-S FLAG
230X
000.004      231X I.CONFL EQU      4      S.CONFL IS 5TH BYTE
000.000      232X      ERRNZ *-S.CSLMD-I.CONFL
040.332      233X S.CONFL DS      1      CONSOLE FLAGS
234X

```

Definitions

EVAL

15:14:34 02-OCT-80

040.333	235X	S.CAADR	DS	2	ADDRESS FOR ABORT PROCESSING (>256 IF VALID)
040.335	236X	S.CCTAB	DS	4	ADDR FOR CTL-A, CTL-B, CTL-C PROCESSING
040.343	237	XTEXT	FILDEF		

## 239X \*\* FILDEF - FILE TYPE DEFINITIONS.

	240X	*			
	241X	*	DB	377Q,FT,XXX	
	242X				
	243X				
000.000	244X	FT.ABS	EQU	0	ABSOLUTE BINARY
000.001	245X	FT.PIC	EQU	1	POSITION INDEPENDANT CODE
000.002	246X	FT.REL	EQU	2	RELOCATABLE CODE
000.003	247X	FT.BAC	EQU	3	COMPILED BASIC CODE
040.343	248	XTEXT	ABSDEF		

## 250X \*\* ABS FORMAT EQUIVALENCES.

	251X				
000.000	252X	ORG		0	
	253X				
000.000	254X	ABS.ID	DS	1	377Q = BINARY FILE FLAG
000.001	255X		DS	1	FILE TYPE (FT,ABS)
000.002	256X	ABS.LDA	DS	2	LOAD ADDRESS
000.004	257X	ABS.LEN	DS	2	LENGTH OF ENTIRE RECORD
000.006	258X	ABS.ENT	DS	2	ENTRY POINT
	259X				
000.010	260X	ABS.COD	DS	0	CODE STARTS HERE
000.010	261	XTEXT	FBDEF		

## 263X \*\* FILE BLOCK DEFINITIONS.

	264X				
000.000	265X	ORG		0	
000.000	266X	FB.CHA	DS	1	CHANNEL NUMBER
000.001	267X	FB.FLG	DS	1	FLAGS
000.002	268X	FB.FWA	DS	2	BUFFER FWA
000.004	269X	FB.PTR	DS	2	BUFFER POINTER
000.006	270X	FB.LIM	DS	2	LIMIT OF DATA IN BUFFER (READ OPERATIONS)
000.010	271X	FB.LWA	DS	2	LWA OF BUFFER
000.012	272X	FB.NAM	DS	4+8+4+1	NAME OF FILE
000.021	273X	FB.NAML	EQU	*-FB.NAM	
000.033	274X	FBENL	EQU	*	ENTRY LENGTH
000.033	275	XTEXT	ECDEF		

Definitions

ECDEF

15:14:39 02-OCT-80

## 277X \*\* ERROR CODE DEFINITIONS.

000.000	279X	ORG	0		
000.000	280X	DS	1	NO ERROR #0	
000.001	281X	EC:EOF	DS	1	END OF FILE
000.002	282X	EC:EDM	DS	1	END OF MEDIA
000.003	283X	EC:ILC	DS	1	ILLEGAL SYSCALL CODE
000.004	284X	EC:CNA	DS	1	CHANNEL NOT AVAILABLE
000.005	285X	EC:DNS	DS	1	DEVICE NOT SUITABLE
000.006	286X	EC:IDN	DS	1	ILLEGAL DEVICE NAME
000.007	287X	EC:IFN	DS	1	ILLEGAL FILE NAME
000.010	288X	EC:NRD	DS	1	NO ROOM FOR DEVICE DRIVER
000.011	289X	EC:FNO	DS	1	CHANNEL NOT OPEN
000.012	290X	EC:ILR	DS	1	ILLEGAL REQUEST
000.013	291X	EC:FUC	DS	1	FILE USAGE CONFLICT
000.014	292X	EC:FNF	DS	1	FILE NAME NOT FOUND
000.015	293X	EC:UND	DS	1	UNKNOWN DEVICE
000.016	294X	EC:ICN	DS	1	ILLEGAL CHANNEL NUMBER
000.017	295X	EC:DIF	DS	1	DIRECTORY FULL
000.020	296X	EC:IFC	DS	1	ILLEGAL FILE CONTENTS
000.021	297X	EC:NEM	DS	1	NOT ENOUGH MEMORY
000.022	298X	EC:RF	DS	1	READ FAILURE
000.023	299X	EC:WF	DS	1	WRITE FAILURE
000.024	300X	EC:WPU	DS	1	WRITE PROTECTION VIOLATION
000.025	301X	EC:WP	DS	1	DISK WRITE PROTECTED
000.026	302X	EC:FAP	DS	1	FILE ALREADY PRESENT
000.027	303X	EC:DDA	DS	1	DEVICE DRIVER ABORT
000.030	304X	EC:FL	DS	1	FILE LOCKED
000.031	305X	EC:FAD	DS	1	FILE ALREADY OPEN
000.032	306X	EC:IS	DS	1	ILLEGAL SWITCH
000.033	307X	EC:UUN	DS	1	UNKNOWN UNIT NUMBER
000.034	308X	EC:FNR	DS	1	FILE NAME REQUIRED
000.035	309X	EC:DIW	DS	1	DEVICE IS NOT WRITABLE (OR WRITE LOCKED)
000.036	310X	EC:UNA	DS	1	UNIT NOT AVAILABLE
000.037	311X	EC:ILV	DS	1	ILLEGAL VALUE
000.040	312X	EC:ILO	DS	1	ILLEGAL OPTION
000.041	313X	EC:VPM	DS	1	VOLUME PRESENTLY MOUNTED ON DEVICE
000.042	314X	EC:NVM	DS	1	NO VOLUME PRESENTLY MOUNTED
000.043	315X	EC:FUD	DS	1	FILE OPEN ON DEVICE
000.044	316X	EC:NPM	DS	1	NO PROVISIONS MADE FOR REMOUNTING MORE DISKS
000.045	317X	EC:DNI	DS	1	DISK NOT INITIALIZED
000.046	318X	EC:DNR	DS	1	DISK IS NOT READABLE
000.047	319X	EC:DSC	DS	1	DISK STRUCTURE IS CORRUPT
000.050	320X	EC:NCV	DS	1	NOT CORRECT VERSION OF HDOS
000.051	321X	EC:ND5	DS	1	NO OPERATING SYSTEM MOUNTED
000.052	322X	EC:IOI	DS	1	ILLEGAL OVERLAY INDEX
000.053	323X	EC:OTL	DS	1	OVERLAY TOO LARGE
000.054	324	XTEXT	DIRDEF		

## Definitions

DIR

19:14:43 02-OCT-80

```

326X **      DIRECTORY ENTRY FORMAT.
327X
000.000      328X      DRG      0
329X
330X
000.377      331X DF.EMP EQU      3770      FLAGS ENTRY EMPTY
000.378      332X DF.CLR EQU      3760      FLAGS ENTRY EMPTY; REST OF DIR ALSO CLEAR
333X
000.000      334X DIR.NAM DS      8      NAME
000.010      335X DIR.EXT DS      3      EXTENSION
000.013      336X DIR.PRO DS      1      PROJECT
000.014      337X DIR.VER DS      1      VERSION
000.015      338X DIRIDL EQU      *      FILE IDENTIFICATION LENGTH
339X
000.015      340X DIR.CLU DS      1      CLUSTER FACTOR
000.016      341X DIR.FLG DS      1      FLAGS
000.017      342X      DS      1      RESERVED
000.020      343X DIR.FGN DS      1      FIRST GROUP NUMBER
000.021      344X DIR.LGN DS      1      LAST GROUP NUMBER
000.022      345X DIR.LSI DS      1      LAST SECTOR INDEX (IN LAST GROUP)
000.023      346X DIR.CRD DS      2      CREATION DATE
000.025      347X DIR.ALD DS      2      LAST ALTERATION DATE
348X
000.027      349X DIRELEN EQU      *      DIRECTORY ENTRY LENGTH
000.027      350      XTEXT      IOCDEF

352X **      I/O CHANNEL DEFINITIONS.
353X
000.000      354X      DRG      0
355X
000.000      356X IOC.LNK DS      2      ADDRESS OF NEXT CHANNEL, =0 IF LAST
000.002      357X IOC.DDA DS      2      THREAD JUMP TO DEVICE DRIVER (VIA DEV TABLE)
358X
000.004      359X IOC.FLG DS      1      FILE TYPE FLAGS
000.001      360X FT.DD EQU      00000001B      =1 IF DIRECTORY DEVICE
000.002      361X FT.OR EQU      00000010B      =1 IF OPEN FOR READ
000.004      362X FT.OW EQU      00000100B      =1 IF OPEN FOR WRITE
000.010      363X FT.OU EQU      00001000B      =1 IF OPEN FOR UPDATE
000.020      364X FT.OC EQU      00010000B      =1 IF OPEN FOR CHARACTER MODE /80.02.BC/
000.003      365X IOC.SQL EQU      *-IOC.DDA      LENGTH OF INFO FOR SEQUENTIAL FILE (FROM IOC)
366X
000.005      367X IOC.GRT DS      2      ADDRESS OF GROUP RESERVATION TABLE
000.007      368X IOC.SPG DS      1      SECTORS PER GROUP, THIS DEVICE
000.010      369X IOC.CGN DS      1      CURRENT GROUP NUMBER
000.011      370X IOC.CSI DS      1      CURRENT SECTOR INDEX (IN CURRENT GROUP)
000.012      371X IOC.LGN DS      1      LAST GROUP NUMBER
000.013      372X IOC.LSI DS      1      LAST SECTOR INDEX (IN LAST GROUP)
000.010      373X IOC.DRL EQU      *-IOC.FLG      LENGTH OF INFO NORMALLY COPIED BACK TO
374X *      THE CHANNEL TABLE
000.014      375X IOC.DTA DS      2      DEVICE TABLE ADDRESS FOR THIS DEVICE
000.016      376X IOC.DES DS      2      SECTOR NUMBER OF DIRECTORY ENTRY
000.020      377X IOC.DEV DS      2      DEVICE CODE
000.022      378X IOC.UNI DS      1      UNIT NUMBER (0-9)
000.021      379X IOC.DIL EQU      *-IOC.DDA      LENGTH OF INFO FOR DIRECTORY FILE (FROM IOC)

```



## Definitions

IOC

15:14:45 02-OCT-80

	380X				
000.023	381X	IOC.DIR DS	DIRELEN		DIRECTORY ENTRY
	382X				
000.052	383X	IOCELEN EQU	*		IOC ENTRY LENGTH
	384X				
000.001	385X	IOCTD EQU	1		INDEX OF USER CHANNEL #0 IN CHANTAB (FIRST = 0)
000.052	386	XTEXT	STDEF		

## 388X \*\* SYMBOL DEFINITION TYPES.

/80.03.sc/

	389X				
000.000	390X	ST.UND EQU	0		UNDEFINED
000.001	391X	ST.LAB EQU	1		LABEL
000.002	392X	ST.EQU EQU	2		DEFINED VIA *EQU*
000.003	393X	ST.SET EQU	3		DEFINED VIA *SET*
000.007	394X	ST.MSK EQU	00000111B		Mask for definition classes
	395X				
000.010	396X	ST.NRF EQU	00001000B		No Cross References are to be taken
000.020	397X	ST.DNA EQU	00010000B		Definition of symbol not allowed
	398X				
000.100	399X	ST.REL EQU	01000000B		Relocatable
000.200	400X	ST.DBL EQU	10000000B		Doubly defined
000.052	401	XTEXT	XDEF		

## 403X \*\* XREF HISTORY REFERENCE-TYPE FLAGS

/80.03.gc/

	404X				
000.000	405X	XT.REF EQU	0		REFERENCED IN EXPRESSION
000.001	406X	XT.LAB EQU	1		DEFINED AS LABEL
000.002	407X	XT.EQU EQU	2		DEFINED VIA *EQU* PSEUDO
000.003	408X	XT.SET EQU	3		DEFINED VIA *SET* PSEUDO
000.004	409X	XT.NRF EQU	4		REFERENCED VIA *NOREF* PSEUDO

/WCZ062680/

Main Routine

15:14:47 02-OCT-80

```

412
042.170          413      ORG      USERFWA-ABS.COD
042.170 377 000      414      DB      377R,FT.ABS
042.172 200 042      415      DW      USERFWA
042.174 146 013      416      DW      MEML-USERFWA
042.176 200 042      417      DW      START
418
419
042.200          420      START  EQU      *
421
042.200 041 136 061 422      LXI      H,RMEML
042.203 377 052      423      SCALL   ,SETTP
042.205 332 376 044 424      JC       ERROR          NOT ENOUGH MEMORY AVAILABLE
425
426 *             CHECK THE HDOS VERSION.
427
042.210 247          428      ANA      A
042.211 377 011      429      DB      SYSCALL,VERS
042.213 332 270 043 430      JC       VERERR          PROBABLY NO VERSION SYSTEM CALL
042.216 376 040      431      CPI      VERS
042.220 302 270 043 432      JNZ      VERERR          NOT THE CORRECT VERSION OF HDOS
433
434 *             POP INFORMATION OFF THE STACK THAT WAS PASSED BY 'ASM'
435
436 *             FORMAT OF STACK:
437 *             POINTER..... DESCRIPTION
438 *
439 *             SP+20..... WIRE SWITCH
440 *             SP+18..... PAGE DEPTH
441 *             SP+16..... FORM DEPTH
442 *             SP+14..... CURRENT PAGE NUMBER
443 *             SP+12..... ADDRESS OF TITLE PORTION OF HEADING
444 *             SP+10..... ADDRESS OF LIST FILE BLOCK
445 *             SP+8..... ADDRESS OF TEMP FILE BLOCK
446 *             SP+6..... FWA OF SYMBOL TABLE
447 *             SP+4..... LIMIT OF SYMBOL TABLE
448 *             SP+2..... LOWEST ADDRESS OF MEMORY THAT
449 *             CAN'T BE OVERLAYED BY 'XREF'
450 *             SP      HANDSHAKE VALUE 'XA'
451
042.223 041 000 000 452      LXI      H,0          CHECK AND SEE IF ANYTHING WAS
042.226 071          453      DAD     SP          PASSED ON THE STACK
042.227 021 200 042 454      LXI      D,STACK
042.232 315 106 034 455      CALL   HLCFDE
042.235 312 252 042 456      JZ       START3      NOTHING PASSED IF (SP) = STACK
457
042.240 341          458      POP     H          PASSED ON STACK IS VALUE
042.241 021 101 130 459      LXI      D,'XA'      'XA' SO I KNOW I WAS INVOKED
042.244 315 106 054 460      CALL   HLCFDE      BY THE ASSEMBLER
042.247 312 323 042 461      JZ       START5      CONTINUE ONLY IF I WAS
462
463 *             'XREF' MUST BE LINKED TO BY 'ASM'... THE USER CANNOT TREAT IT
464 *             AS A COMMAND.
465
042.252          466      START3 EQU      *
042.252 315 136 031 467      CALL   $TYPX

```

Main Routine

15:14:48 02-OCT-80

042.255	007	130	122	468	DB	BELL, 'XREF Cannot be Used as a Command'
042.316	012	212		469	DB	NL,ENL
042.320	303	006	045	470	JMP	RESTART
				471		
042.323				472	START5 EQU	*
				473		
042.323	341			474	POP	H 'GET ADDR OF LOWEST AREA NEEDED IN HBASM'
042.324	021	136	061	475	LXI	D,RMEML
042.327	315	106	054	476	CALL	HLCPDE
042.332	077			477	CMC	
042.333	332	053	043	478	JC	MAIN0 DE < HL
				479		
				480	*	'TABLES IN MEMORY LEFT BY 'ASM' WOULD BE OVERLAYED'
				481	*	BY THIS VERSION OF XREF.
				482		
042.336	315	047	045	483	CALL	IERROR
042.341	315	136	031	484	CALL	\$TYPTX
042.344	124	150	151	485	DB	'This version of XREF'
042.371	157	166	145	486	DB	'overlays the needed tables from the assembler'
043.046	012	212		487	DB	NL,ENL
043.050	303	006	045	488	JMP	RESTART
				489		
043.053				490	MAIN0 EQU	*
043.053	341			491	POP	H
043.054	042	050	056	492	SHLD	SYMLWA RESTORE SYMBOL TABLE LIMIT
				493		
043.057	341			494	POP	H
043.060	042	046	056	495	SHLD	SYMFWA RESTORE FWA OF SYMTAB
				496		
043.063	001	033	000	497	LXI	B,FBENL MOVE TEMPORARY FILE BLOCK
043.066	321			498	POP	D
043.067	041	001	056	499	LXI	H,TEMPFB
043.072	315	252	030	500	CALL	\$MOVE
				501		
043.075	001	033	000	502	LXI	B,FBENL MOVE LIST FILE BLOCK
043.100	321			503	POP	D
043.101	041	348	055	504	LXI	H,LISTFB
043.104	315	252	030	505	CALL	\$MOVE
				506		
043.107	001	062	000	507	LXI	B,HEAD1L MOVE TITLE TO MY HEADING
043.112	321			508	POP	D
043.113	041	336	054	509	LXI	H,HEAD1
043.116	315	252	030	510	CALL	\$MOVE
				511		
043.121	361			512	POP	PSW GET VALUE OF CURRENT
043.122	062	231	055	513	STA	PAGNUM PAGE NUMBER
				514		
043.125	361			515	POP	PSW GET VALUE OF
043.126	062	230	055	516	STA	FORMDP FORM DEPTH
				517		
043.131	361			518	POP	PSW GET VALUE OF
043.132	062	227	055	519	STA	PAGEDP PAGE DEPTH
				520		
043.135	361			521	POP	PSW GET VALUE OF
043.136	062	226	055	522	STA	WIDE WIDE SWITCH
				523		

Main Routine

15:14:50 02-OCT-80

```

043.141 061 200 042 524 LXI SP,STACK
525
043.144 315 171 046 526 CALL PRS
527
528 * Type out the Symbol Table to the LIST file
529
043.147 315 022 050 530 CALL SRT
043.152 315 363 046 531 CALL PST
043.155 041 346 055 532 LXI H,LISTFB
043.160 315 312 053 533 CALL $FWBRK dump the symbol table buffer
534
535 * Build and output the Cross-Reference Table to the list output file
536
043.163 315 017 044 537 CALL BCRT
043.166 315 232 045 538 CALL PCRT
539
540 * PRINT XREF STATS
541
043.171 315 306 047 542 CALL PXS
043.174 315 273 051 543 CALL FNP6 FORCE FINAL NEW PAGE
043.177 041 346 055 544 LXI H,LISTFB
043.202 315 312 053 545 CALL $FWBRK EMPTY LISTING BUFFER
043.205 041 000 050 546 LXI H,PXSB
043.210 377 003 547 DB $SYSCALL,PRINT TYPE STATS
000.000 548 ERRNZ *-EXIT
549
550 * Exit
551
043.212 552 EXIT EQU *
553
043.212 041 136 061 554 LXI H,RMEHL
043.215 377 052 555 SCALL $SETTP
043.217 332 376 044 556 JC ERROR should never happen
557
043.222 041 346 055 558 LXI H,LISTFB
043.225 315 055 052 559 CALL $FCLO Close the LIST file
560
043.230 315 251 043 561 CALL DELTMP CLOSE AND DELETE TEMP FILE
562
043.233 257 563 XRA A
043.234 377 000 564 EXIT SCALL $EXIT
565
566 * HANDLE CTL-C HIT.
567
043.236 568 CCHIT EQU *
043.236 315 136 031 569 CALL $TYPTX
043.241 136 303 570 DB $C'+80H
043.243 315 251 043 571 CALL DELTMP
043.246 303 006 045 572 JMP RESTART
573
574 * DELETE TEMP FILE.
575
043.251 576 DELTMP EQU *
043.251 041 001 056 577 LXI H,TEMPFB Close the TEMPorary file
043.254 315 055 052 578 CALL $FCLO
579

```

Main Routine

15:14:54 02-OCT-80

```
043.257 021 034 056 580 LXI D,DEFLT
043.262 041 013 056 581 LXI H,TEMPFB+FB.NAM
043.265 377 050 582 SCALL .DELET Delete the TEMPorary file
583
043.267 311 584 RET
585
586 * NOT CORRECT VERSION OF HDOS.
587
043.270 588 VERERR EQU *
043.270 076 050 589 MVI A,EC.NCV
043.272 303 376 044 590 JMP ERROR
```

Subroutines

AHR

15:14:54 02-OCT-80

```

594 **      AHR      - Add Historic Reference
595 *
596 *      AHR adds a historic reference to the specified XREF table.
597 *
598 *      ENTRY: HL      = pointer to XREF table identifier
599 *
600 *      EXIT: entry in RECORD added to its XREF entry
601 *
602 *      USES: PSW,BC,DE,HL
603 *
604
043.275 315 355 043 605 AHR      CALL      ANS
606
043.300 124          607 AHR1     MOV       D,H
043.301 135          608          MOV       E,L
043.302 315 211 030 609          CALL     $HLIHL
000.000          610          ERRNZ   XH.LNK
043.305 174          611          MOV       A,H
043.306 265          612          ORA      L
043.307 302 300 043 613          JNZ      AHR1
614
043.312 315 013 045 615          CALL     GHR
616
043.315 175          617          MOV       A,L
043.316 022          618          STAX    D
043.317 023          619          INX     D
043.320 174          620          MOV       A,H
043.321 022          621          STAX    D
043.322 021 000 000 622          LXI     D,0
043.325 315 147 054 623          CALL     $INDS      set link pointer
043.330 000 000      624          DW      XH.LNK
625
043.332 353          626          XCHG
043.333 052 333 054 627          LHLD    RECORD+LINEM
043.336 353          628          XCHG
043.337 315 147 054 629          CALL     $INDS      set line number
043.342 002 000      630          DW      XH.LIN
631
043.344 072 335 054 632          LDA     RECORD+REFTYP
043.347 315 203 054 633          CALL     $INDSB     set flag byte
043.352 004 000      634          DW      XH.FLG
635
043.354 311          636          RET

```

```

638 **      ANS      - Accept Next Symbol
639 *
640 *      ANS accepts the next symbol.
641 *
642 *
043.355 257          643 ANS     XRA      A
644
043.356 266          645 ANS1    ORA      M
043.357 043          646          INX     H

```

Subroutines

ANS

15:14:55 02-OCT-80

```

043.360 362 356 043 647 JP ANS1 NOT TO THE END OF THE SYMBOL YET
648
043.363 043 649 INX H
043.364 043 650 INX H SKIP DATA
043.365 043 651 INX H
000.000 652 ERRNZ ANSA-3
653
043.366 311 654 RET
655
000.003 656 ANSA EQU 3 Number of data elements / entry

```

```

658 ** ANX - Advance to the Next XREF-history table entry
659 *
660 * ANX advances to the next XREF history table entry.
661 *
662 * ENTRY: HL = entry address
663 *
664 * EXIT: HL = HL advanced
665 *
666 * USES: PSW,HL
667 *
668
043.367 315 355 043 669 ANX CALL ANS
043.372 043 670 INX H
043.373 043 671 INX H
043.374 311 672 RET

```

```

674 ** BCGP1
675 *
676 * BCGP1 sets BC to the value of the group #1 size.
677 *
678 *
679 * ENTRY: NONE
680 *
681 * EXIT: BC = group #1 size
682 * PSW = 'Z' clear if group != 0
683 * 'Z' set if group == 0
684 *
685 * USES: PSW
686 *
687
043.375 001 000 000 688 BCGP1 LXI B,0
043.376 689 BCGP1 EQU *-2
690
044.000 170 691 MOV A,B
044.001 261 692 ORA C
044.002 311 693 RET

```

Subroutines

BCGF2

15:14:55 02-OCT-80

```

695 **      BCGP2
696 *
697 *      BCGP2 sets BC to the value of the group #2 size.
698 *
699 *
700 *      ENTRY:  NONE
701 *
702 *      EXIT:   BC      = group #2 size
703 *            PSW    = 'Z' clear if group != 0
704 *            'Z' set  if group == 0
705 *
706 *      USES:   PSW
707 *
708
044.003 001 000 000 709 BCGP2 LXI   B,0
044.004          710 BCGP2 EQU   *-2
          711
044.006 170          712      MOV   A,B
044.007 261          713      ORA   C
044.010 311          714      RET

```

```

716 **      BCGPSZ
717 *
718 *      BCGPSZ sets BC to the value of the Group size
719 *
720 *      ENTRY:  none
721 *
722 *      EXIT:   BC      = group size
723 *            PSW    = 'Z' clear if not zero
724 *            'Z' set  if      zero
725 *
726 *      USES:   BC
727 *
728
044.011 001 000 000 729 BCGPSZ LXI   B,0
044.012          730 BCGPSZ EQU   *-2
          731
044.014 170          732      MOV   A,B
044.015 261          733      ORA   C
044.016 311          734      RET

```

```

736 **      BCRT - Build Cross-Reference Table
737 *
738 *      BCRT builds the cross reference table from the temporary
739 *      XREF history file.
740 *
741 *
044.017 315 070 044 742 BCRT  CALL  BCRT1  move symbol table to high memory
          743
044.022 052 046 056 744      LHL  SYMFWA

```



## Subroutines

BCRT

15:15:02 02-DCI-80

```

044.025 021 136 061 745 LXI D,XREFTAB
044.030 315 144 044 746 CALL BCRT2 move to low memory inserting pointers
044.033 353 747 XCHG
044.034 001 010 000 748 LXI B,B
044.037 021 320 031 749 LXI D,$ZEROS
044.042 315 252 030 750 CALL $MOVE flag the end of the list
044.045 001 010 000 751 LXI B,B
044.050 021 320 031 752 LXI D,$ZEROS
044.053 315 252 030 753 CALL $MOVE
044.056 042 042 056 754 SHLD FREELST initialize free list pointer
755
044.061 315 306 044 756 CALL BHT build hash table
044.064 315 173 044 757 CALL BCRT3 build history records
044.067 311 758 RET

```

```

760 * Move the symbol table to high memory
761

```

```

044.070 052 050 056 762 BCRT1 LHLD SYMLWA
044.073 001 020 000 763 LXI B,16
044.076 011 764 DAD B
044.077 353 765 XCHG HL = end of symbol table + 16
044.100 052 046 058 766 LHLD SYMFWA
044.103 315 224 030 767 CALL $CHL
044.106 031 768 DAD D
044.107 104 769 MOV B,H
044.110 115 770 MOV C,L BC = move count
771
044.111 052 046 058 772 LHLD SYMFWA
044.114 353 773 XCHG DE = source
774
044.115 052 044 056 775 LHLD MEMLIM
044.120 175 776 MOV A,L
044.121 221 777 SUB C
044.122 157 778 MOV L,A
044.123 174 779 MOV A,H
044.124 230 780 SBB B
044.125 147 781 MOV H,A HL = HL - BC
044.126 042 046 058 782 SHLD SYMFWA update symbol table FWA
783
044.131 315 252 030 784 CALL $MOVE
785
044.134 021 360 377 786 LXI D,-16
044.137 031 787 DAD D
044.140 042 050 056 788 SHLD SYMLWA update symbol table LWA
789
044.143 311 790 RET

```

Subroutines

BCRT2

15:15:04 02-OCT-80

```

792 *      Move the symbol table to low memory and insert pointers
793
044.144 176 794 BCRT2 MOV  A,M
044.145 247 795 ANA  A
044.146 310 796 RZ          to the end of the list
797
044.147 315 337 050 798 CALL YSE
799
044.152 257 800 XRA  A
044.153 022 801 STAX D
044.154 023 802 INX  D          add 2 data bytes to the entry
044.155 022 803 STAX D
044.156 023 804 INX  D
805
044.157 315 106 054 806 CALL HLCPE
044.162 322 144 044 807 JNC  BCRT2
808
044.165 076 021 809 MVI  A,EC,NEM
044.167 303 376 044 810 JMP  ERROR
044.172 311 811 RET

```

```

813 *      Read the XREF history file and fill in references
814
044.173 815 BCRT3 EQU  *
816
044.173 001 012 000 817 LXI  B,RECLN
044.174 021 324 054 818 LXI  D,RECORD
044.201 041 001 056 819 LXI  H,TEMPFB
044.204 315 202 052 820 CALL $FREAR
044.207 330 821 RC          End-of-File
822
044.210 072 324 054 823 LDA  RECORD
044.213 247 824 ANA  A
044.214 310 825 RZ          At the end of the temporary file
826
044.215 021 324 054 827 LXI  D,RECORD
044.220 315 226 044 828 CALL BCRT4
044.223 303 173 044 829 JMP  BCRT3
830
044.226 315 066 045 831 BCRT4 CALL LXE
044.231 076 021 832 MVI  A,EC,NEM
044.233 312 302 044 833 JZ   BCRT5          was found in XREF table
834
835 *      Illegal data in temp file
836
044.236 315 047 045 837 CALL IERROR
044.241 315 136 031 838 CALL $IYPTX
044.244 111 154 154 839 DB  'Illegal Temp File Contents',ENL
044.277 303 212 043 840 JMP  EXIT
841
842 *      Identifier was found in the XREF table
843
044.302 315 275 043 844 BCRT5 CALL AHR

```

Subroutines

BCRT3

15:15:05 02-OCT-80

044.305 311

845

RET

847 \*\* BHT - Build Hash Table

848 \*

849 \*

850 \*

851 \*

852 \*

853 \*

854 \*

855 \*

856 \*

857 \*

858 \*

859 \*

860

044.306 021 052 061

861

BHT

LXI D,HASHTAB

044.311 041 136 061

862

LXI H,XREFTAB

044.314 006 101

863

MVI B,BHTA

864

044.316 176

865

BHT1

MOV A,M

044.317 247

866

ANA

A

044.320 312 340 044

867

JZ BHT2

to the end of the XREF history list

868

044.323 176

869

MOV A,M

044.324 346 177

870

ANI 1770

044.326 270

871

CMP

B

044.327 322 340 044

872

JNC BHT2

entry &gt;= b

873

874 \*

entry &lt; key

875

044.332 315 367 043

876

CALL ANX

044.335 303 316 044

877

JMP BHT1

878

879 \*

entry &gt;= key

880

044.340 353

881

BHT2

XCHG

044.341 163

882

MOV M,E

044.342 043

883

INX

H

044.343 162

884

MOV M,D

044.344 043

885

INX

H

044.345 353

886

XCHG

887

044.346 004

888

INR

B

044.347 170

889

MOV A,B

044.350 376 133

890

CFI BHTB+1

044.352 332 316 044

891

JC BHT1

not to the end of keys

892

044.355 311

893

RET

894

000.101

895

BHTA

EQU 'A'

first key

000.132

896

BHTB

EQU 'Z'

last key

000.002

897

BHTC

EQU 2

table entry sizes

Subroutines

CSE

15:15:06 02-OCT-80

```

899 ** CSE - Count Symbol Table Entries
900 *
901 * CSE counts the number of entries in the symbol table.
902 *
903
044.356 052 046 056 904 CSE LHL D SYMFWA
044.361 001 000 000 905 LXI B,0
906
044.364 176 907 CSE1 MOV A,M
044.365 247 908 ANA A
044.366 310 909 RZ at the end of the list
910
044.367 315 355 043 911 CALL ANS pass over the symbol table entry
044.372 003 912 INX B
044.373 303 364 044 913 JMP CSE1

```

```

915 ** ERROR - Error Processor
916 *
917 *
918
044.376 919 ERROR EQU *
044.376 046 012 920 MVI H,NL
045.000 377 057 921 SCALL .ERROR
045.002 315 136 031 922 CALL $TYPTX
045.005 207 923 DB BELL+80H
924
045.006 925 RESTART EQU *
045.006 076 001 926 MVI A,1 EXIT AND ABORT
045.010 303 234 043 927 JMP EXIT.

```

```

929 ** GHR - Get Historic Reference entry
930 *
931 * GHR allocates a historic reference entry block
932 *
933 * ENTRY: NONE
934 *
935 * EXIT: HL = pointer to entry block
936 *
937 * USES: PSW,HL
938 *
939
045.013 325 940 GHR PUSH D
045.014 021 005 000 941 LXI D,XH.LEN
045.017 052 042 056 942 LHL D FREELST
045.022 345 943 PUSH H
045.023 031 944 DAD D
045.024 353 945 XCHG
045.025 052 044 056 946 LHL MEMLIM
045.030 353 947 XCHG
045.031 315 106 054 948 CALL HLCFDE

```

Subroutines

BHR

15:15:08 02-OCT-80

```

045.034 076 021 949 MVI A,EC,NEM
045.036 322 376 044 950 JNC ERROR
951
045.041 042 042 056 952 SHLD FREELST
045.044 341 953 POP H
045.045 321 954 POP D
045.046 311 955 RET

957 ** IERROR - Internal Error
958 *
959 * IERROR process internal error.
960 *
961 * ENTRY: NONE
962 *
963 * EXIT: NONE
964 *
965 * USES: ALL
966 *
967
045.047 315 136 031 968 IERROR CALL $TYPTX
045.052 007 012 105 969 DB BELL,NL,'Error - ',200Q
045.065 311 970 RET

```

```

972 ** LXE - Look-Up Xref-History Table Entry
973 *
974 * LXE looks the XREF history table entry
975 * corresponding to the specified symbol table
976 * entry.
977 *
978 * ENTRY: DE = address of symbol identifier
979 *
980 * EXIT: PSW = 'Z' Clear if NOT found
981 * 'Z' Set if found
982 * HL = address of entry
983 *
984 * USES: PSW,DE,HL
985 *
986
045.066 041 136 061 987 LXE LXI H,XREFTAB
988
989 * Look-up any existing Hash Table entry
990
045.071 032 991 LDAX D
045.072 326 101 992 SUI BHTA
045.074 332 116 045 993 JC LXE1 not in table
045.077 376 032 994 CPI BHTB-BHTA+1
045.101 322 116 045 995 JNC LXE1 not in table
996
997 * Fetch entry for in range identifier
998

```

Subroutines

LXE

15:15:08 02-OCT-80

```

045.104 207          999          ADD    A          A = 2 * A
045.105 041 052 041 1000         LXI    H,HASHTAB
045.110 315 072 030 1001         CALL   $DADA
045.113 315 211 030 1002         CALL   $HLIHL      HL = hash table entry
1003
1004 *          Search until found, or beyond its spot
1005
045.116 174          1006 LXE1    MOV    A,M
045.117 247          1007         ANA    A
045.120 302 127 045 1008         JNZ    LXE3      not to the end of the list
1009
045.123 076 001     1010 LXE2    MVI    A,1
045.125 247          1011         ANA    A          flag not found
045.126 311          1012         RET
1013
045.127 315 251 050 1014 LXE3    CALL   SRT13
045.132 310          1015         RZ          is found
045.133 332 123 045 1016         JC    LXE2      *DE < *HL, is not in the table
1017
045.136 315 367 043 1018         CALL   ANX
045.141 303 116 045 1019         JMP    LXE1      try the next entry

1021 **         MG2    - Move Group 2
1022 *
1023 *         MG2 moves group #2 to a temporary holding buffer
1024 *
1025 *         ENTRY: DE = group #1
1026 *         HL   = group #2
1027 *
1028
045.144 345          1029 MG2    PUSH   H
045.145 325          1030         PUSH  D
045.146 021 213 045 1031         LXI    D,MG2A
1032
045.151 315 337 050 1033         CALL   TSE
1034
045.154 321          1035         POP   D
045.155 343          1036         XTHL      save new group #2 address
1037
1038 *         move the old elements up
1039
045.156 006 000     1040         MVI    B,0      BC = temp entry length
045.160 305          1041         PUSH  B
045.161 325          1042         PUSH  D          save fwa of group 1
1043
045.162 325          1044         PUSH  D          save source address
045.163 353          1045         XCHG
045.164 315 224 030 1046         CALL   $CHL
045.167 031          1047         DAD   D
045.170 315 313 054 1048         CALL   XCHGBC    BC = move count
045.173 321          1049         POP   D          DE = source
045.174 031          1050         DAD   D          HL = destination
045.175 315 252 030 1051         CALL   $MOVE

```

Subroutines

MG2

15:15:10 02-OCT-80

			1052				
045.200	341		1053	POP	H	restore destination	
045.201	301		1054	POP	B	BC = temp entry length	
045.202	021 213 045		1055	LXI	D, MG2A	DE = source ( save buffer address )	
045.205	315 252 030		1056	CALL	\$MOVE		
			1057				
045.210	353		1058	XCHG		DE = new group #1 address	
045.211	341		1059	POP	H	HL = new group #1 address	
			1060				
045.212	311		1061	RET			
			1062				
045.213			1063	MG2A DS	15	Temporary holding area for group #2	
			1065	** PCRT	-	Print Cross-Reference Table	
			1066	*			
			1067	*		PCRT prints the Cross-Reference table on the list device	
			1068	*			
			1069				
045.232			1070	PCRT	EQU	*	
045.232	076 040		1071	MVI	A, ' '	CHANGE HEADING TO INDICATE	
045.234	006 062		1072	MVI	B, HEAD2L	CROSS REFERENCE	
045.238	041 047 055		1073	LXI	H, HEAD2		
045.241			1074	PCRT0	EQU	*	
045.241	167		1075	MOV	M, A		
045.242	043		1076	INX	H		
045.243	005		1077	DCR	B		
045.244	302 241 045		1078	JNZ	PCRT0		
045.247	001 025 000		1079	LXI	B, SUBT2L		
045.252	021 161 055		1080	LXI	D, SUBT2		
045.255	041 047 055		1081	LXI	H, HEAD2		
045.260	315 252 030		1082	CALL	\$MOVE		
045.263	315 067 051		1083	CALL	FNP	FORCE NEW PAGE	
			1084				
045.266	041 138 061		1085	LXI	H, XREFTAB		
			1086				
045.271	176		1087	PCRT1	MOV	A, M	
045.272	247		1088		ANA	A	
045.273	310		1089		RZ		
			1090				
045.274	345		1091	PUSH	H		
045.275	315 321 045		1092	CALL	PCRT2	Build an output record	
045.300	332 312 045		1093	JC	PCRT1.5	Internal symbol ignored	
045.303	315 017 051		1094	CALL	WLD	Write the data to the list device	
045.306	001 000 144		1095	INW	PCRTDL, PCRTD		
045.312	341		1096	PCRT1.5	POP	H	
045.313	315 367 043		1097	CALL	ANX		
045.316	303 271 045		1098	JMP	PCRT1		

Subroutines

PCRT2

15:15:11 02-OCT-80

```

1100 * BUILD THE OUTPUT RECORD
1101
045.321 315 076 047 1102 PCRT2 CALL PST2 CHECK TYPE OF SYMROL
045.324 330 1103 RC Symbol was only of internal interest
045.325 315 037 051 1104 CALL COL COUNT OUTPUT LINE
045.330 315 117 047 1105 CALL PST2P OUTPUT THE SYMBOL AND ITS VALUE
1106
045.333 315 211 030 1107 CALL $HLIHL
000.000 1108 ERRNZ XH.LNK
1109
045.336 072 145 046 1110 LDA PCRT2
045.341 062 146 046 1111 STA PCRT2 Initialize the line-wrap count-down
1112
045.344 174 1113 PCRT3 MOV A,M
045.345 265 1114 ORA L
045.346 310 1115 RZ To the end of the references
1116
045.347 315 360 045 1117 CALL PCRT4
045.352 315 211 030 1118 CALL $HLIHL
000.000 1119 ERRNZ XH.LNK
045.355 303 344 045 1120 JMP PCRT3

045.360 345 1122 PCRT4 PUSH H
045.361 315 234 030 1123 CALL $INDL
045.364 002 000 1124 DW XH.LIN
045.366 076 005 1125 MVI A,PCRTBL GENERATE CORRECT NUMBER OF DIGITS
045.370 102 1126 MOV B,D
045.371 113 1127 MOV C,E BC = line number
045.372 041 136 046 1128 LXI H,PCRTB
045.375 315 157 031 1129 CALL $UDD UNPACK THE LINE NUMBER
1130
046.000 041 136 046 1131 LXI H,PCRTB
046.003 006 004 1132 MVI B,PCRTBL-1
046.005 176 1133 PCRT5 MOV A,M SPACE FILL LEADING ZEROES
046.006 376 060 1134 CPI '0'
046.010 302 022 046 1135 JNZ PCRT6
046.013 066 040 1136 MVI M,' '
046.015 005 1137 DCR B
046.016 043 1138 INX H
046.017 302 005 046 1139 JNZ PCRT5
1140
046.022 341 1141 PCRT6 POP H
046.023 345 1142 PUSH H
046.024 076 040 1143 MVI A,' '
046.026 062 143 046 1144 STA PCRTC initialize the flag to ' '
046.031 315 126 054 1145 CALL $INDLB A = reference type
046.034 004 000 1146 DW XH.FLG
046.036 041 232 055 1147 LXI H,TYPSRF
046.041 315 235 054 1148 CALL $TBL
046.044 176 1149 MOV A,M
046.045 302 053 046 1150 JNZ PCRT7 WAS NOT FOUND
046.050 062 143 046 1151 STA PCRTC STUFF REFERENCE TYPE FLAG
1152

```



Subroutines

PCRT4

15:15:13 02-OCT-80

046.053	072	146	046	1153	PCRT7	LDA	PCRTF		
046.056	247			1154		ANA	A		
046.057	314	102	046	1155		CZ	PCRTB	WRAP THE LINE	
				1156					
046.062	315	017	051	1157		CALL	WLD		
046.065	012	000	132	1158		DW	PCRTAL,PCRTA		
046.071	332	376	044	1159		JC	ERROR		
				1160					
046.074	041	146	046	1161		LXI	H,PCRTF		
046.077	065			1162		DCR	M	count the symbolic reference on this line	
				1163					
046.100	341			1164		POP	H	restore pointer to this element	
046.101	311			1165		RET			
046.102				1167	PCRTB	EQU	*		
046.102	315	017	051	1168		CALL	WLD		
046.105	001	000	144	1169		DW	PCRTDL,PCRTD		
046.111	315	037	051	1170		CALL	COL		
046.114	315	017	051	1171		CALL	WLD		
046.117	022	000	147	1172		DW	PCRTGL,PCRTG		
046.123	072	145	046	1173		LDA	PCRTE		
046.126	062	146	046	1174		STA	PCRTF	re-initialize the line-wrap counter	
046.131	311			1175		RET			
				1176					
046.132				1177	PCRTA	DS	0	Output record	
				1178					
046.132	040	040	040	1179		DB		Space between line numbers	
				1180					
046.136				1181	PCRTB	DS	5	Line Number	
000.005				1182	PCRYBL	EQU	*-PCRTB		
				1183					
046.143	040			1184	PCRTC	DB		Reference Flag	
				1185					
000.012				1186	PCRTAL	EQU	*-PCRTA		
				1187					
				1188					
046.144	012			1189	PCRTD	DB	NL	Go to the next line string	
000.001				1190	PCRTDL	EQU	*-PCRTD		
				1191					
046.145	006			1192	PCRTE	DB	6	Number of references per line	
				1193	*			NARROW 6, WIDE 10	
046.146	000			1194	PCRTF	DB	0	Count-Down until line wrap	
				1195					
046.147	040			1196	PCRTG	DB		error	
046.150	040			1197		DB			
046.151	040	040	040	1198		DB		symbol	
046.160	040			1199		DB			
046.161	040	040	040	1200		DB		value	
046.167	040			1201		DB		relocatable	
046.170	040			1202		DB		no-reference	
000.022				1203	PCRTGL	EQU	*-PCRTG		
000.000				1204	ERRNZ	PCRTGL-PSTL		must match header string	

Subroutines

PRS

15:15:15 02-OCT-80

```

1206 ** PRS - Preset
1207 *
1208 * PRS sets up the remaining XREF environmental parameters
1209 *
1210 * reinitialize file blocks
1211 * request all memory
1212 * set up control character processors
1213 * fill in headings
1214 * adjust values based on wide switch settings
1215 *
1216 *
046.171 1217 PRS EQU *
1218
1219 * Re-initialize the file blocks
1220
046.171 041 052 060 1221 LXI H,LSTBUF
046.174 042 350 055 1222 SHLD LISTFB+FB.FWA
046.177 041 052 061 1223 LXI H,LSTBUF+LSTBFL
046.202 042 356 055 1224 SHLD LISTFB+FB.LWA
046.205 041 346 055 1225 LXI H,LSTFB
046.210 315 162 052 1226 CALL %FCLEAR
1227
046.213 041 052 056 1228 LXI H,TEMPBUF
046.216 042 003 056 1229 SHLD TEMPFB+FB.FWA
046.221 041 052 060 1230 LXI H,TEMPBUF+TMPEFL
046.224 042 011 056 1231 SHLD TEMPFB+FB.LWA
046.227 041 001 056 1232 LXI H,TEMPFB
046.232 315 162 052 1233 CALL %FCLEAR
046.235 072 001 056 1234 LDA TEMPFB+FB.CHA
1235
046.240 072 001 056 1236 LDA TEMPFB+FB.CHA A = CHANNEL
046.243 021 034 056 1237 LXI D,DEFALTT
046.246 041 013 056 1238 LXI H,TEMPFB+FB.NAM
046.251 377 054 1239 SCALL %NAME SAVE THE FULLY SPECIFIED NAME
046.253 332 376 044 1240 JC ERROR
1241
046.256 041 001 056 1242 LXI H,TEMPFB
046.261 315 055 052 1243 CALL %FCLO
046.264 021 034 056 1244 LXI D,DEFALTT
046.267 041 001 056 1245 LXI H,TEMPFB
046.272 315 333 051 1246 CALL %FOPEU Re-Open the file for update
1247
1248 * Set-up control character processors
1249
046.275 076 003 1250 MVI A,CTLG SPECIAL HANDLING ONLY FOR CTL-C
046.277 041 236 043 1251 LXI H,CCHIT
046.302 377 041 1252 DB SYSCALL,%CTLG
1253
1254 * Request all available memory
1255
046.304 041 377 377 1256 LXI H,-1
046.307 377 052 1257 SCALL %SETIP
046.311 021 366 377 1258 LXI D,-10
046.314 031 1259 DAD D
046.315 042 044 056 1260 SHLD MEMLIM save memory limit
046.320 377 052 1261 SCALL %SETIP

```

Subroutines

FRS

15:15:17 02-OCT-80

```

046.322 332 376 044 1262 JC ERROR Error settings memory
1263
1264 * MOVE DATE TO HEADING LINE.
046.325 001 011 000 1265 LXI B,HEAD3L
046.330 021 277 040 1266 LXI D,S,DATE
046.333 041 132 055 1267 LXI H,HEAD3
046.336 315 252 030 1268 CALL $MOVE
1269
1270 * ADJUST VALUES BASED ON SETTING OF WIDE SWITCH.
046.341 072 226 055 1271 LDA WIDE
046.344 247 1272 ANA A
046.345 312 362 046 1273 JZ PRGX
046.350 078 012 1274 MOVI A,I0
046.352 062 145 046 1275 STA PCRITE NUMBER OF REFERENCES/LINE
046.355 078 006 1276 MOVI A,%
046.357 062 301 047 1277 STA PSTE NUMBER OF SYMBOLS/LINE
1278
046.362 1279 PRSX EQU *
046.362 311 1280 RET

1282 ** PST - Print Symbol Table
1283 *
1284 * PST prints the symbol table on the list device
1285 *
1286
046.363 1287 PST EQU *
046.363 052 046 056 1288 LHLD SYMFWA HL = fwa of the symbol table
046.366 072 301 047 1289 LDA PSTE INITIALIZE WRAP AROUND COUNTER
046.371 062 302 047 1290 STA PSTF
1291
046.374 176 1292 PST1 MOV A,M
046.375 247 1293 ANA A
046.376 312 247 047 1294 JZ PST8 at the end of the symbol table
1295
047.001 315 076 047 1296 CALL PST2 CHECK TYPE OF SYMBOL
047.004 332 374 046 1297 JC PST1 Symbol was of internal interest only
1298
047.007 345 1299 PUSH H IF BEGINNING OF NEW LINE,
047.010 072 302 047 1300 LDA PSTF THEN COUNT OUTPUT LINE
047.013 041 301 047 1301 LXI H,PSTE
047.016 276 1302 CMP M
047.017 314 037 051 1303 CZ EOL
047.022 341 1304 POP H
1305
047.023 315 117 047 1306 CALL PST2P PRINT SYMBOL AND VALUE
1307
047.026 072 302 047 1308 LDA PSTF
047.031 075 1309 INCR A
047.032 062 302 047 1310 STA PSTF
047.035 302 062 047 1311 JNZ PST1.6 still room for more symbols on this line
1312
1313 * wrap the output line
1314

```

Subroutines

PST

15:15:18 02-OCT-80

```

047.040 345      1315 PST1.3 PUSH  H
047.041 315 017 051 1316      CALL  WLD
047.044 001 000 303 1317      DW    PSTGL,PSTG
1318
047.050 072 301 047 1319      LDA   PSTE
047.053 062 302 047 1320      STA   PSTF
047.056 341      1321      POP   H
047.057 303 374 046 1322      JMP   PST1
1323
1324 *          output space to separate the symbols
1325
047.062 345      1326 PST1.6 PUSH  H
047.063 315 017 051 1327      CALL  WLD
047.066 002 000 304 1328      DW    PSTIL,PSTI
047.072 341      1329      POP   H
047.073 303 374 046 1330      JMP   PST1

1332 **       PST2
1333 *
1334 *          PST2 outputs a symbol and its value.  If 'C' is set at
1335 *          exit, then it is a dummy internal symbol and was not
1336 *          printed.  Hence it should not be counted.  Special
1337 *          symbols are flagged with a leading '0' in the symbol
1338 *          table, and are generated by *EPO* for pass evaluation
1339 *          errors.  (EPO is in ASM,HBA.)
1340 *
1341 *          EXIT!  PSW      = 'C' set if special symbol
1342
047.076 353      1343 PST2  XCHG          DE = fwa of symtab entry
1344
1345 *          Format the symbol name
1346

047.077 041 261 047 1347      LXI   H,PSTB
047.102 315 366 050 1348      CALL  USN          Unpack the symbol name
1349

047.105 072 261 047 1350      LDA   PSTB
047.110 376 060      1351      CPI   '0'
047.112 312 241 047 1352      JZ    PST7          Ignore symbols with initial '0'
047.115 247      1353      ANA   A
047.116 311      1354      RET          RETURN SAYING OK TO PRINT
1355

1356 *          Format the symbol type
047.117      1357 PST2P EQU    *
047.117 076 040      1358      MVI   A,
047.121 062 257 047 1359      STA   PSTA          Error flag
047.124 062 277 047 1360      STA   PSTD0         Relocatable flag
047.127 062 300 047 1361      STA   PSTD1         No references flag
1362

047.132 032      1363      LDAX  D          A = SYMBOL TYPE FLAG
047.133 346 007      1364      ANI   ST,MSK         Mask out any of the flag bits
000.000      1365      ER RNZ ST,UND
047.135 302 145 047 1366      JNZ  PST3
047.140 076 125      1367      MVI   A,'U'          FLAG UNDEFINED LIKE ERROR

```

Subroutines

PST2

15:15:19 02-OCT-80

047.142	062 257 047	1368		STA	PSTA	
		1369				
047.145	032	1370	PST3	LDAX	D	
047.146	346 010	1371		ANI	ST,NRF	
047.150	312 160 047	1372		JZ	PST4	No reference is not set
047.153	076 116	1373		MVI	A,'N'	
047.155	062 300 047	1374		STA	PSTD1	
		1375				
047.160	032	1376	PST4	LDAX	D	
047.161	346 200	1377		ANI	ST,DBL	
047.163	312 173 047	1378		JZ	PST5	NOT DOUBLY DEFINED
047.166	076 104	1379		MVI	A,'D'	FLAG DOUBLY DEFINED LIKE ERROR
047.170	062 257 047	1380		STA	PSTA	
		1381				
047.173	032	1382	PST5	LDAX	D	
047.174	346 100	1383		ANI	ST,REL	
047.176	312 206 047	1384		JZ	PST6	NOT RELOCATABLE
047.201	076 047	1385		MVI	A,'''''	
047.203	062 277 047	1386		STA	PSTD0	SET THE TYPE FIELD
		1387				
047.206	023	1388	PST6	INX	D	
		1389				
		1390	*			Format the symbol value
		1391				
047.207	032	1392		LDAX	D	
047.210	023	1393		INX	D	
047.211	365	1394		PUSH	PSW	
047.212	032	1395		LDAX	D	
047.213	023	1396		INX	D	
047.214	041 271 047	1397		LXI	H,PSTC	
047.217	315 266 054	1398		CALL	\$UOD	
047.222	361	1399		POP	PSW	
047.223	315 266 054	1400		CALL	\$UOD	FILL IN THE SYMBOL VALUE
000.000		1401		ERRNZ	PSTCL-6	
		1402				
		1403	*			Output the symbol table entry to the standard output file
		1404				
047.226	325	1405		PUSH	D	SAVE SYMBOL TABLE POINTER
047.227	315 017 051	1406		CALL	WLD	
047.232	022 000 257	1407		DW	PSTL,PSTA	
		1408				
047.236	341	1409		POP	H	restore the pointer to the next entry
047.237	247	1410		ANA	A	Clear 'C'
047.240	311	1411		RET		
		1412				
		1413	*			Ignore symbols with leading '0' as special internals
		1414				
047.241	353	1415	PST7	XCHG		HL = symbol pointer
047.242	043	1416		INX	H	
047.243	043	1417		INX	H	Advance over symbol data
047.244	043	1418		INX	H	
047.245	067	1419		STC		Flag error
047.246	311	1420		RET		
		1421				
		1422	*			At the end of the symbol table
		1423				

Subroutines

PST2

15:15:21 02-OCT-80

047.247		1424	PSTB	EQU	*	
047.247	315 017 051	1425		CALL	WLD	
047.252	001 000 303	1426		DW	PSTGL,PSTG	
		1427				
047.256	311	1428		RET		
		1429				
047.257	040	1430	PSTA	DB	' '	undefined flas
		1431				
047.260	040	1432		DB	' '	
		1433				
047.261	061 062 063	1434	PSTB	DB	'1234567'	symbol identifier
000.007		1435	PSTBL	EQU	*-PSTB	
000.000		1436		ERRNZ	PSTBL-SYMBOLL	
		1437				
047.270	040	1438		DB	' '	
		1439				
047.271	060 060 060	1440	PSTC	DB	'000000'	symbol value
000.006		1441	PSTCL	EQU	*-PSTC	
		1442				
047.277	040	1443	PSTDQ	DB	' '	Relocatable
047.300	040	1444	PSTD1	DB	' '	No references
		1445				
000.022		1446	PSTL	EQU	*-PSTA	
		1447				
		1448				
047.301	004	1449	PSTE	DB	4	number of symbols per line
		1450	*			NARROW 4, WIDE 6
047.302	000	1451	PSTF	DB	0	number of symbols output on the line
		1452				
047.303	012	1453	PSTG	DB	NL	string to wrap line
000.001		1454	PSTGL	EQU	*-PSTG	
		1455				
047.304	040 040	1456	PSTI	DB	' '	symbol separator
000.002		1457	PSTIL	EQU	*-PSTI	
		1459	**	PXS	-	PRINT XREF STATS
		1460	*			
		1461	*			PRINT THE NUMBER OF BYTES FREE.
		1462	*			
		1463	*	ENTRY	NONE	
		1464	*	EXIT	NONE	
		1465	*	USES	ALL	
		1466	*			
		1467				
047.306		1468	PXS	EQU	*	
		1469				
047.306	052 044 056	1470		LHLD	MEMLIM	FREE = MEMLIM-FREELST
047.311	353	1471		XCHG		
047.312	052 042 056	1472		LHLD	FREELST	
047.315	173	1473		MOV	A,E	
047.316	225	1474		SUB	L	
047.317	117	1475		MOV	C,A	
047.320	172	1476		MOV	A,D	

Subroutines

PXS

15:15:23 02-OCT-80

```

047.321 234 1477 SBB H
047.322 107 1478 MOV B,A
047.323 322 331 047 1479 JNC PXS4
047.326 001 000 000 1480 LXI B,0
1481
047.331 1482 PXS4 EQU *
047.331 041 000 030 1483 LXI H,PXSB FORMAT FOR OUTPUT
047.334 076 005 1484 MVI A,5
047.336 315 157 031 1485 CALL $UDD
1486
047.341 072 225 055 1487 LDA LINCNT SEE IF MESSAGE WILL FIT ON PAGE
047.344 376 003 1488 CPI 3 OR DO I NEED TO FNP
047.346 334 067 051 1489 CC FNP
1490
047.351 001 023 000 1491 LXI B,PXSAL LIST STAT
047.354 021 376 047 1492 LXI D,PXSA
047.357 041 346 055 1493 LXI H,LISTFB
047.362 315 130 053 1494 CALL $FWRIB
1495
047.365 072 225 055 1496 LDA LINCNT ACCOUNT FOR LINES PRINTED
047.370 306 003 1497 ADI 3
047.372 062 225 055 1498 STA LINCNT
1499
047.375 311 1500 RET
1501
047.376 012 012 1502 PXS4 DB NL,NL
050.000 060 060 060 1503 PXS4 DB '0000 Bytes Free',NL
000.023 1504 PXSAL EQU *-PXS4
050.021 212 1505 DB ENL

```

---

```

1507 ** SRT - SORT
1508 *
1509 * SRT sorts the symbol table
1510 *
1511
050.022 1512 SRT EQU *
1513
050.022 315 358 044 1514 CALL CSE
050.025 315 313 054 1515 CALL XCHGBC
050.030 042 335 050 1516 SHLD SRTA SAVE THE NUMBER OF ENTRIES
1517
050.033 021 001 000 1518 LXI D,1 INITIALIZE THE GROUP SIZE
1519
050.036 052 335 050 1520 SRT1 HLHI SRTA
050.041 315 106 054 1521 CALL HLCFDE
050.044 330 1522 RC
050.045 310 1523 RZ GROUP SIZE = NUMBER OF ELEMENTS
1524
050.046 325 1525 PUSH D
050.047 325 1526 PUSH D
050.050 315 062 050 1527 CALL SRT2
050.053 341 1528 POP H
050.054 321 1529 POP D

```

## Subroutines

SRT

15:15:24 02-OCT-80

```

050.055 031 1530 DAD D DOUBLE THE BRGOF SIZE
050.056 353 1531 XCHG DE = NEW GROUP SIZE
050.057 303 036 050 1532 JMP SRT1
1533

```

```

050.062 353 1535 SRT2 XCHG
050.063 042 012 044 1536 SHLD BCBPSZ. SAVE GROUP SIZE
050.066 052 046 056 1537 LHLD SYMFWA
1538
050.071 124 1539 SRT3 MOV D,H
050.072 135 1540 MOV E,L DE = FWA OF LIST
1541

```

```

050.073 315 011 044 1542 CALL BCBPSZ
1543

```

```

050.076 176 1544 SRT4 MOV A,M
050.077 247 1545 ANA A
050.100 310 1546 RZ
1547

```

```

050.101 315 355 043 1548 CALL ANS
050.104 013 1549 DCX B
050.105 170 1550 MOV A,B
050.106 261 1551 ORA C
050.107 302 076 050 1552 JNZ SRT4
1553

```

```

050.112 176 1554 MOV A,M
050.113 247 1555 ANA A
050.114 310 1556 RZ
1557

```

```

050.115 315 131 050 1558 CALL SRT5
1559

```

```

1560 * ADVANCE TO THE END OF GROUP #2
1561

```

```

050.120 315 003 044 1562 CALL BCBP2 BC = group #2 size
050.123 315 234 050 1563 CALL SRT9
1564
050.126 303 071 050 1565 JMP SRT3

```

```

050.131 345 1567 SRT5 PUSH H
050.132 315 011 044 1568 CALL BCBPSZ
050.135 315 313 054 1569 CALL XCHGBC
050.140 042 376 043 1570 SHLD BCBP1.
050.143 042 004 044 1571 SHLD BCBP2.
050.146 341 1572 POP H
1573

```

```

050.147 176 1574 SRT6 MOV A,M
050.150 247 1575 ANA A
050.151 310 1576 RZ
1577 GROUP 2 IS AT THE END OF THE LIST

```

```

050.152 315 375 043 1578 CALL BCBP1 GROUP 1 EXHAUSTED
050.155 310 1579 RZ
1580

```



```

050.156 315 003 044 1581 CALL BCGP2 GROUP 2 EXHAUSTED
050.161 310 1582 RZ
1583
050.162 315 170 050 1584 CALL SRT7
050.165 303 147 050 1585 JMP SRT6
1586
050.170 353 1587 SRT7 XCHG
050.171 315 251 050 1588 CALL SRT13
050.174 353 1589 XCHG
050.175 332 217 050 1590 JC SRT8 #2 < #1
1591
1592 * #1 <= #2
1593
050.200 353 1594 XCHG
050.201 315 355 043 1595 CALL ANS
050.204 353 1596 XCHG
050.205 345 1597 PUSH H
050.206 052 376 043 1598 LHLD BCGF1.
050.211 053 1599 DCX H
050.212 042 376 043 1600 SHLD BCGF1.
050.215 341 1601 POP H
050.216 311 1602 RET
1603
1604 * #1 > #2
1605
050.217 1606 SRT8 EQU *
050.217 315 144 045 1607 CALL MG2 MOVE GROUP 2
050.222 345 1608 PUSH H
050.223 052 004 044 1609 LHLD BCGF2.
050.226 053 1610 DCX H
050.227 042 004 044 1611 SHLD BCGF2.
050.232 341 1612 POP H
050.233 311 1613 RET

1615 ** SRT9
1616 *
1617 * SRT9 advances to the end of the group described by
1618 * HL and BC.
1619 *
1620 * ENTRY: BC = number of entries remaining
1621 * HL = pointer to the entries
1622 *
1623 * EXIT: HL = pointer to END_OF_GROUP + 1
1624 *
1625 * USES: PSW,BC,HL
1626 *
1627
050.234 176 1628 SRT9 MOV A,M
050.235 247 1629 ANA A
050.236 310 1630 RZ at the end of the list of entries
1631
050.237 170 1632 MOV A,B
050.240 261 1633 ORA C

```

Subroutines

SRT9

15:15:27 02-OCT-80

```

050.241 310      1634      RZ          the entry count has expired
              1635
050.242 315 355 043 1636      CALL     ANS
050.245 013      1637      DCX      B
050.246 303 234 050 1638      JMP      SRT9

```

```

              1640 **      SRT13
              1641 *
              1642 *      This compare assumes that #1 != #2.
              1643 *
              1644 *
              1645 *      ENTRY: DE      = ENTRY #1
              1646 *      HL      = ENTRY #2
              1647 *
              1648 *      EXIT:  PSW      = 'C' SET IF *DE < *HL
              1649 *      'C' CLEAR IF *DE >= *HL
              1650 *      'Z' SET IF *DE == *HL (can not happen)
              1651 *
              1652 *      DE,HL UPDATED
              1653 *
              1654 *
              1655 *      USES:  PSW
              1656 *
              1657
050.251      1658 SRT13 EQU *
050.251 305      1659      PUSH     B
050.252 325      1660      PUSH     D
050.253 345      1661      PUSH     H
              1662
050.254 353      1663 SRT14 XCHG
050.255 106      1664      MOV      B,M      B = *(DE)
050.256 353      1665      XCHG
050.257 116      1666      MOV      C,M      C = *(HL)
050.260 023      1667      INX     D
050.261 043      1668      INX     H
              1669
050.262 170      1670      MOV     A,B
050.263 251      1671      XRA     C
050.264 372 305 050 1672      JM      SRT17      SIGNS ARE NOT EQUAL
              1673
              1674 *      SIGNS ARE EQUAL
              1675
050.267 170      1676      MOV     A,B
050.270 247      1677      ANA     A
050.271 372 300 050 1678      JM      SRT15      To the end of both strings
050.274 271      1679      CMP     C
050.275 312 254 050 1680      JZ      SRT14
              1681
050.300 271      1682 SRT15 CMP     C
              1683
050.301 341      1684 SRT16 POP     H
050.302 321      1685      POP     D
050.303 301      1686      POP     B

```

Subroutines

SRT13

15:19:28 02-OCT-80

```

050.304 311      1687      RET
                1688
                1689 *      SIGNS ARE NOT EQUAL
                1690
050.305 171      1691 SRT17  MOV    A,C      STRIP OUT THE END BITS
050.306 346 177  1692      ANI    1770
050.310 117      1693      MOV    C,A
050.311 170      1694      MOV    A,B
050.312 346 177  1695      ANI    1770
050.314 271      1696      CMP    C
050.315 302 301 050 1697      JNZ   SRT16      They were not equal
                1698
050.320 170      1699      MOV    A,B
050.321 346 200  1700      ANI    2000
050.323 067      1701      STC
                1702      assume end of DE
050.324 302 301 050 1703      JNZ   SRT16
                1704
050.327 076 001  1704      MVI    A,1
050.331 247      1705      ANA    A      clear "Z" and clear "C"
050.332 303 301 050 1706      JMP   SRT16
                1707
050.335 000 000  1708 SRTA  DW    0      Number of Elements in the list
                .....
                1710 **     TSE    - Transfer Symbol Table Entry
                1711 *
                1712 *     TSE transfers the specified symbol table entry
                1713 *
                1714 *     ENTRY: DE      = destination
                1715 *           HL      = source
                1716 *
                1717 *     EXIT: DE,HL advanced to next entry
                1718 *           C      = number of bytes transferred
                1719 *
                1720 *     USES: PSW,BC,DE,HL
                1721 *
                1722
050.337 016 000  1723 TSE  MVI    C,0
                1724
                1725 *     Move symbol identifier
                1726
050.341 176      1727 TSE1  MOV    A,M
050.342 022      1728      STAX  D
050.343 023      1729      INX  D
050.344 043      1730      INX  H
050.345 014      1731      INR  C
050.346 247      1732      ANA  A
050.347 362 341 050 1733      JP   TSE1      NOT TO THE END OF THE SYMBOL YET
                1734
                1735 *     Move symbol data
                1736
050.352 006 003  1737      MVI    B,3
050.354 176      1738 TSE2  MOV    A,M
050.355 022      1739      STAX  D

```

Subroutines

TSE

15:15:29 02-OCT-80

```

050.356 023      1740      INX      D
050.357 043      1741      INX      H
050.360 014      1742      INR      C
050.361 005      1743      DCR      B
050.362 302 354 050 1744      JNZ      TSE2
                                1745
050.365 311      1746      RET

                                1748 **      USN      - Unpack Symbol Name
                                1749 *
                                1750 *      USN unpacks the symbol name pointed to by DE into
                                1751 *      the buffer pointed to by HL. The symbol name is
                                1752 *      space filled.
                                1753 *
                                1754 *
                                1755 *      ENTRY: DE      = address of symbol name
                                1756 *      HL      = address of destination buffer
                                1757 *
                                1758 *      EXIT:  DE, HL advanced
                                1759 *
                                1760 *      USES:  ALL
                                1761 *
                                1762
050.366      1763 USN      EQU      *
050.366 006 007 1764      MVI      B,SYMBOLL      B = maximum symbol identifier length
                                1765
050.370 032      1766 USN1     LDAX      D
050.371 365      1767      PUSH     PSW
050.372 346 177      1768      ANI      1770      MAP OUT HIGH ORDER BIT
050.374 167      1769      MOV      M,A
050.375 361      1770      POP      PSW
050.376 005      1771      DCR      B      COUNT THE BYTE
050.377 023      1772      INX      D
051.000 043      1773      INX      H
051.001 247      1774      ANA      A
051.002 362 370 050 1775      JF      USN1      NOT TO THE END OF THE SYMBOL NAME
                                1776
051.005 170      1777      MOV      A,B
051.006 247      1778      ANA      A
051.007 310      1779 USN2     RZ      ALL FINISHED PADDING
051.010 066 040 1780      MVI      M,' '
051.012 005      1781      DCR      B
051.013 043      1782      INX      H
051.014 303 007 051 1783      JMP      USN2      TRY TO PAD SOME MORE

```

## Subroutines

WLD

15:15:30 02-OCT-80

```

1785 **      WLD      - Write to List Device
1786 *
1787 *      WLD writes the specified number of bytes to
1788 *      the device controlled by the LISTFB file
1789 *      file block.
1790 *
1791 *
1792 *      ENTRY:  RET      = number of bytes
1793 *             RET+2    = address of bytes
1794 *
1795 *      EXIT:   RET+4
1796 *
1797 *      USES:   ALL
1798 *
1799 *
051.017 343 1800 WLD  XTHL
051.020 116 1801      MOV   C,M
051.021 043 1802      INX   H
051.022 106 1803      MOV   B,M
051.023 043 1804      INX   H          BC = COUNT
1805
051.024 136 1806      MOV   E,h
051.025 043 1807      INX   H
051.026 126 1808      MOV   D,M
051.027 043 1809      INX   H          DE = buffer pointer
051.030 343 1810      XTHL
1811
051.031 041 346 055 1812      LXI   H,LISTFB
051.034 303 130 053 1813      JMP   $FWRIB

```

```

1815 **      COL - COUNT OUTPUT LINES.
1816 *
1817 *      COL IS CALLED TO COUNT AN OUTPUT LINE BEFORE IT IS WRITTEN.
1818 *
1819 *      ENTRY  NONE
1820 *      EXIT   NONE
1821 *      USES   A,F
1822 *
051.037 1823 COL  EQU   *
051.037 345 1824      PUSH  H          SAVE REGS
051.040 325 1825      PUSH  D
051.041 305 1826      PUSH  B
1827
051.042 072 225 055 1828      LDA   LINCNT      Q. TIME FOR
051.045 247 1829      ANA   A          NEW PAGE
051.046 302 054 051 1830      JNZ   COL3        BR IF NOT
1831
051.051 315 067 051 1832      CALL  FNP         PREPARE NEW PAGE
1833
051.054 1834 COL3 EQU   *
051.054 072 225 055 1835      LDA   LINCNT      COUNT LINE
051.057 075 1836      DCR   A
051.060 062 225 055 1837      STA   LINCNT

```

Subroutines

COL

15:15:30 02-OCT-80

```

1838
051.063 301 1839 POP B RESTORE REGS
051.064 321 1840 POP D
051.065 341 1841 POP H
1842
051.066 311 1843 RET

1845 ** FNP - FORCE NEW PAGE
1846 *
1847 * FNP CAUSES A PAGE EJECT, BY FORMFEED OR BY LINEFEED,
1848 * WHICHEVER IS REQUIRED.
1849 * THEN IT PRINTS THE HEADING, IF REQUESTED.
1850 *
1851 * 1ST TIME ENTERED, THERE IS NO NEED TO DO SKIP TO NEW
1852 * PAGE, ASSUME AT TOP OF NEW PAGE WHEN 'XREF' STARTED.
1853 *
1854 * ENTRY NONE
1855 * EXIT NONE
1856 * USES ALL
1857
051.067 1858 FNP EQU *
051.067 076 001 1859 MVI A,1 INDICATE PRINT
051.071 062 31A 051 1860 STA FNPD THE HEADING
1861
051.074 1862 FNP EQU *
051.074 072 313 051 1863 LDA FNPC Q. 1ST TIME HERE
051.077 247 1864 ANA A
051.100 312 112 051 1865 JZ FNPO BR IF NOT
051.103 257 1866 XRA A RESET FLAG
051.104 062 313 051 1867 STA FNPC
051.107 303 175 051 1868 JMP FNP3
1869
051.112 1870 FNP0 EQU *
051.112 072 230 055 1871 LDA FORMDP Q. USE FORMFEEDS
051.115 247 1872 ANA A
051.116 302 140 051 1873 JNZ FNP1 BR IF NOT -- MUST USE LINEFEEDS
1874
1875 * DEVICE WILL TAKE A FORMFEED.
051.121 001 001 000 1876 LXI B,1
051.124 021 311 051 1877 LXI D,FNPA
051.127 041 346 055 1878 LXI H,LISTFB
051.132 315 130 053 1879 CALL $FWRIB
051.135 303 175 051 1880 JMP FNP3
1881
1882 * MUST USE LINEFEEDS.
051.140 1883 FNP1 EQU *
051.140 041 227 055 1884 LXI H,PAGEDP
051.143 226 1885 SUB M (A) = GAP SPACE
051.144 041 225 055 1886 LXI H,LINCNT
051.147 206 1887 ADD M (A) = AMOUNT NEEDED
051.150 312 175 051 1888 JZ FNP3 BR IF NO LINES NEEDED
051.153 1889 FNP2 EQU *
051.153 001 001 000 1890 LXI B,1

```

Subroutines

FNP

15:15:33 02-OCT-80

```

051.156 021 312 051 1891 LXI D,FNFB
051.161 041 346 055 1892 LXI H,LISTFB
051.164 365 1893 PUSB FSW SAVE COUNT
051.165 315 130 053 1894 CALL $FWRIB WRITE LF
051.170 361 1895 POP FSW
051.171 075 1896 DCR A
051.172 302 153 051 1897 JNZ FNP2 GO SOME MORE
1898
1899 * PRINT HEADING.
051.175 1900 FNP3 EQU *
051.175 072 314 051 1901 LDA FNPD
051.200 247 1902 ANA A
051.201 310 1903 RZ REQUESTED NOT TO PRINT HEADING
051.202 072 231 055 1904 LDA PAGNUM INCREMENT PAGE NUMBER
051.205 074 1905 INR A
051.206 062 231 055 1906 STA PAGNUM
1907
051.211 117 1908 MOV C,A UNPACK IN TO HEADING LINE
051.212 006 000 1909 MVI B,0
051.214 041 154 055 1910 LXI H,HEAD4
051.217 076 003 1911 MVI A,HEAD4L
051.221 315 157 031 1912 CALL $UDD
051.224 041 154 055 1913 LXI H,HEAD4 POINT TO PAGE NUMBER /80.09.BB/
1914
051.227 016 002 1915 MVI CY,HEAD4L-1 DO AMOUNT-Y NUMBERS /80.09.BB/
051.231 176 1916 FNP4 MOV A,M GET CHARACTER /80.09.BB/
051.232 376 060 1917 CPI '0' IS HE A ZERO? /80.09.BB/
051.234 302 246 051 1918 JNZ FNPS NO, NO MORE SUPPRESS /80.09.BB/
051.237 066 040 1919 MVI M,' ' YES, CHANGE TO SPACES /80.09.BB/
051.241 043 1920 INX H POINT TO NEXT /80.09.BB/
051.242 015 1921 DCR C SEE IF MORE TO GO /80.09.BB/
051.243 302 231 051 1922 JNZ FNP4 IF MORE, GO BACK /80.09.BB/
1923
051.246 1924 FNPS EQU * /80.09.BB/
051.246 001 223 000 1925 LXI B,HEADL WRITE HEADING
051.251 021 336 054 1926 LXI D,HEADING
051.254 041 346 055 1927 LXI H,LISTFB
051.257 315 130 053 1928 CALL $FWRIB
1929
1930 * ADJUST PAGE LINE COUNTER.
051.262 072 227 055 1931 LDA PAGEDP
051.265 326 003 1932 SUI 3 ACCOUNT FOR HEADING LINES
051.267 062 225 055 1933 STA LINCNT SET LINES REMAINING
1934
051.272 311 1935 RET
1936
1937 * FORCE NEW PAGE WITHOUT PAGE HEADING.
1938
051.273 1939 FNPS EQU *
051.273 257 1940 XRA A INDICATE DON'T
051.274 062 314 051 1941 STA FNPD PRINT HEADING
051.277 315 074 051 1942 CALL FNP
051.302 072 227 055 1943 LDA PAGEDP
051.305 062 225 055 1944 STA LINCNT
051.310 311 1945 RET
1946

```

Subroutines

FNP

15:15:35 02-OCT-80

051.311	014	1947	FNPA	DB	FF	FORMFEED
051.312	012	1948	FNPB	DB	NL	LINEFEED
051.313	001	1949	FNPC	DB	1	FLAG := <>0 INDICATES 1ST TIME HERE
051.314		1950	FNPD	DS	1	FLAG := <>0 PRINT PAGE HEADING



051.315 1953 XTEXT FOPE

1955X \*\* \$FOPEX - OPEN FILE BLOCK FOR I/O  
 1956X \*  
 1957X \* \$FOPEX IS CALLED BEFORE ANY I/O IS DONE VIA A  
 1958X \* FILE BLOCK. \$FOPEX SETS UP THE FILE BLOCK, AND OPENS  
 1959X \* THE FILE VIA \*HDOS\*.  
 1960X \*  
 1961X \* ENTRY (DE) = ADDRESS OF DEFAULT BLOCK  
 1962X \* (HL) = ADDRESS OF FILE BLOCK  
 1963X \* EXIT TO \$FERROR IF ERROR  
 1964X \* TO CALLER IF OK  
 1965X \* USES A,F,B,C,D,E  
 1966X \*  
 1967X \*

051.315 315 342 051 1968X \$FOPER CALL \$FOPER.

051.320 320 1969X RNC

051.321 303 016 054 1970X JMP \$FERROR IN ERROR

1971X

051.324 315 345 051 1972X \$FOPEW CALL \$FOPEW.

051.327 320 1973X RNC

051.330 303 016 054 1974X JMP \$FERROR IN ERROR

1975X

051.333 315 350 051 1976X \$FOPEU CALL \$FOPEU.

051.336 320 1977X RNC

051.337 303 016 054 1978X JMP \$FERROR IN ERROR

1979X

1980X

051.342 076 002 1981X \$FOPER, MVI A,FT,OR FILE TYPE OF OPEN FOR READ

051.344 001 1982X DB 001Q LXI,B TO SKIP NEXT MVI

051.345 076 004 1983X \$FOPEW, MVI A,FT,OW OPEN FOR WRITE

051.347 001 1984X DB 001Q LXI,B TO SKIP NEXT MIV

051.350 076 006 1985X \$FOPEU, MVI A,FT,OR+FT,OW

1986X

1987X \* (A) = FILE FLAGS

1988X

051.352 345 1989X PUSH H SAVE FILE BLOCK ADDRESS

051.353 365 1990X PUSH PSW SAVE NEW FLAGS

000.000 1991X ERRNZ FB,CHA

051.354 106 1992X MOV B,M (B) = CHANNEL NUMBER

051.355 305 1993X PUSH B SAVE HANNEL NUMBER

000.000 1994X ERRNZ FB,FLG-FB,CHA-1

051.356 043 1995X INX H

051.357 117 1996X MOV C,A (C) = NEW FILE FLAGS

051.360 176 1997X MOV A,M (A) = CURRENT TYPE

051.361 247 1998X ANA A

051.362 171 1999X MOV A,C (A) = NEW FLAGS TO BE SET

051.363 312 375 051 2000X JZ \$FOPE1 NOT ALREADY OPEN

2001X

2002X \* ALREADY OPEN. SQUACK

2003X

051.366 301 2004X POP B RESTORE (BC)

051.367 361 2005X POP PSW DISCARD NEW FLAGS

Common Decks

#FOPE

15:15:39 02-OCT-80

051.370	341	2006X	POP	H	(HL) = FB ADDRESS
051.371	076 031	2007X	MVI	A,EC,FA0	FILE ALREADY OPEN
051.373	067	2008X	STC		
051.374	311	2009X	RET		
		2010X			
000.000		2011X	ERRNZ	FB,FWA-FB,FLG-1	
051.375	043	2012X \$FOPE1	INX	H	(HL) = #FB,FWA
051.376	116	2013X	MOV	C,M	
051.377	043	2014X	INX	H	
052.000	106	2015X	MOV	B,M	(BC) = FB,FWA
052.001	043	2016X	INX	H	
000.000		2017X	ERRNZ	FB,PTR-FB,FWA-2	
052.002	161	2018X	MOV	M,C	SET FB,PTR = FB,FWA
052.003	043	2019X	INX	H	
052.004	160	2020X	MOV	M,B	
052.005	043	2021X	INX	H	
000.000		2022X	ERRNZ	FB,LIM-FB,PTR-2	
052.006	161	2023X	MOV	M,C	SET FB,LIM = FB,FWA
052.007	043	2024X	INX	H	
052.010	160	2025X	MOV	M,B	
052.011	043	2026X	INX	H	
000.000		2027X	ERRNZ	FB,NAM-FB,LIM-4	
052.012	043	2028X	INX	H	
052.013	043	2029X	INX	H	(HL) = #FB,NAM
		2030X			
		2031X *		FILE BLOCK POINTERS SETUP, OPEN FILE	
		2032X			
052.014	345	2033X	PUSH	H	SAVE NEW ADDRESS FOR NAME
052.015	041 046 052	2034X	LXI	H,\$FOPEB	
052.020	247	2035X	ANA	A	
052.021	312 030 052	2036X	JZ	\$FOPE2	/78,10,BC/
000.000		2037X	ERRNZ	,EXIT	
052.024	315 235 054	2038X	CALL	\$TBLS	FIND CODE
052.027	176	2039X	MOV	A,M	
052.030	062 036 052	2040X \$FOPE2	STA	\$FOPEA	SET SYSCALL CODE
052.033	341	2041X	POP	H	(HL) = #FB,NAM
052.034	361	2042X	POP	PSW	(A) = CHANNEL NUMBER
052.035	377 000	2043X	DB	SYSCALL,EXIT	
052.036		2044X \$FOPEA	ERU	*-1	SYSCALL CODE
052.037	321	2045X	POP	D	(D) = NEW FLAG
052.040	341	2046X	POP	H	(HL) = FILE BLOCK ADDRESS
052.041	330	2047X	RC		EXIT IF ERROR
052.042	043	2048X	INX	H	
000.000		2049X	ERRNZ	FB,FLG-1	
052.043	162	2050X	MOV	M,D	SET NEW FLAGS
052.044	053	2051X	DCX	H	RESTORE (HL)
052.045	311	2052X	RET		
		2053X			
052.046	002 042	2054X \$FOPEB	DB	FT,OR,.OPENR	TABLE OF SYSCALL CODES
052.050	004 043	2055X	DB	FT,OW,.OPENW	
052.052	006 044	2056X	DB	FT,OR+FT,OW,.OPENU	
052.054	000	2057X	DB	0	SHOULD NOT OCCUR
052.055		2058X	XTEXT	FCLO	

Common Decks

\$FCLO

15:15:43 02-OCT-80

```

2060X ** $FCLO - CLOSE FILE BLOCK.
2061X *
2062X * $FCLO IS CALLED TO TERMINATE PROCESSING THROUGH A FILE
2063X * BLOCK.
2064X *
2065X * ENTRY (HL) = FILE BLOCK ADDRESS
2066X * EXIT TO $FERROR IF ERROR
2067X * TO CALLER IF OK
2068X * USES A,F,B,C,D,E
2069X
2070X
052.055 315 064 052 2071X $FCLO CALL $FCLO,
052.060 320 2072X RNC NO ERROR
052.061 303 016 054 2073X JMP $FERROR
2074X
052.064 345 2075X $FCLO, PUSH H SAVE FILE BLOCK ADDRESS
000.000 2076X ERNZ FB,FLG-1
052.065 043 2077X INX H (HL) = $FB,FLG
052.066 176 2078X MOV A,M
052.067 066 000 2079X MVI M,0 CLEAR FLAG
052.071 247 2080X ANA A
052.072 312 160 052 2081X JZ $FCLO4 FILE NOT OPEN
052.075 346 004 2082X ANI FY,00
052.077 312 152 052 2083X JZ $FCLO3 NO WRITING, NO FLUSHING NEEDED
2084X
2085X * WAS OPEN FOR WRITE. SEE IF NEED FLUSH THE LAST SECTOR
2086X
052.102 315 234 030 2087X CALL $INDL
052.105 003 000 2088X DW FB,PTR-FB,FLG
052.107 325 2089X PUSH D SAVE (FB,PTR)
052.110 315 234 030 2090X CALL $INDL (DE) = (FB,FWA)
052.113 001 000 2091X DW FB,FWA-FB,FLG
052.115 341 2092X POP H (HL) = (FB,PTR)
052.116 175 2093X MOV A,L
052.117 223 2094X SUB E
052.120 117 2095X MOV C,A
052.121 174 2096X MOV B,H
052.122 232 2097X SBB D
052.123 107 2098X MOV B,A (BC) = AMOUNT IN BLOCK
052.124 261 2099X ORA C
052.125 312 152 052 2100X JZ $FCLO3 NONE TO FLUSH
2101X
2102X * NEED TO FLUSH BUFFER
2103X *
2104X * (BC) = DATA AMOUNT
2105X * (DE) = FWA
2106X * (HL) = LWA+1
2107X
052.130 171 2108X MOV A,C
052.131 247 2109X ANA A
052.132 312 145 052 2110X JZ $FCLO2 DONT HAVE PARTIAL SECTOR
2111X
2112X * ZERO FILL PARTIAL SECTOR
2113X
052.135 066 000 2114X $FCLO1 MOV M,0
052.137 043 2115X INX H

```

Common Decks

\$FCLO

15:15:44 02-OCT-80

```

052.140 014      2116X      INR      C
052.141 302 135 052 2117X      JNZ      $FCLO1
052.144 004      2118X      INR      B          COUNT ANOTHER FULL SECTOR
052.145 341      2119X $FCLO2  POP      H          (HL) = FB.FWA
052.146 176      2120X      MOV      A,M        (A) = CHANNEL NUMBER
000.000          2121X      ERRNZ   FB.CHA
052.147 345      2122X      PUSH    H
052.150 377 005  2123X      DB      SYSCALL,WRITE      FLUSH
                2124X
                2125X *      READY TO CLOSE FILE.
                2126X *
                2127X *      'C' SET IF ERROR
                2128X *      (A) = ERROR CODE
                2129X
052.152 341      2130X $FCLO3  POP      H          (HL) = FILE BLOCK ADDRESS
052.153 330      2131X      RC
000.000          2132X      ERRNZ   FB.CHA      ERROR
052.154 176      2133X      MOV      A,M        (A) = CHANNEL NUMBER
052.155 345      2134X      PUSH    H
052.156 377 046  2135X      DB      SYSCALL,CLOSE  CLOSE CHANNEL
052.160 341      2136X $FCLO4  POP      H          (HL) = FILE BLOCK ADDRESS
052.161 311      2137X      RET
052.162          2138X      XTEXT  FCLEAR

```

```

2140X **      $FCLEAR - CLEAR FILE BLOCK.
2141X *
2142X *      $FCLEAR CLEARS OUT A FILE BLOCK BY SETTING THE POINTERS TO
2143X *      EMPTY, AND CLEARING ANY ERROR OR EOF FLAGS.
2144X *
2145X *      THE DISK (OR WHATEVER) FILE IS NOT POSITIONED, READ, WRITEN
2146X *      OPENED OR CLOSED.
2147X *
2148X *      ENTRY  (HL) = FB ADDRESS
2149X *      EXIT   NONE
2150X *      USES   A,F,B,C
2151X
2152X
052.162          2153X $FCLEAR  EQU      *
052.162 345      2154X      PUSH    H          SAVE FILE BLOCK ADDRESS
000.000          2155X      ERRNZ   FB.FLG-FB.CHA-1
052.163 043      2156X      INX      H
000.000          2157X      ERRNZ   FB.FWA-FB.FLG-1
052.164 043      2158X      INX      H          (HL) = #FB.FWA
052.165 116      2159X      MOV      C,M
052.166 043      2160X      INX      H
052.167 106      2161X      MOV      B,M        (BC) = FB.FWA
052.170 043      2162X      INX      H
000.000          2163X      ERRNZ   FB.PTR-FB.FWA-2
052.171 161      2164X      MOV      M,C        SET FB.PTR = FB.FWA
052.172 043      2165X      INX      H
052.173 160      2166X      MOV      M,B
052.174 043      2167X      INX      H
000.000          2168X      ERRNZ   FB.LIM-FB.PTR-2

```

Common Decks

\*FCLEAR

15:15:46 02-OCT-80

```

052.175 161      2169X      MOV      M,C      SET FB.LIM = FB.FWA
052.176 043      2170X      INX      H
052.177 160      2171X      MOV      M,B
052.200 341      2172X      POP      H      (HL) = FB.FWA
052.201 311      2173X      RET
052.202          2174      XTEXT    FREAB

2176X **      *FREAB - READ BYTES FROM FILE BUFFER.
2177X *
2178X *      *FREAB IS CALLED TO READ A NUMBER OF BYTES FROM A FILE BUFFER.
2179X *
2180X *      ENTRY   (BC) = BYTE COUNT
2181X *      (DE) = FWA FOR BYTES
2182X *      (HL) = ADDRESS OF FILE BUFFER
2183X *      EXIT    TO *FERROR* IF ERROR
2184X *      TO CALLER IF OK
2185X *      (BC) = UNREAD BYTE COUNT (ONLY IF EOF)
2186X *      (DE) = ADDRESS OF FIRST UNUSED BYTE
2187X *      'C' SET IF EOF DURING READ
2188X *      USES   A,F,B,C,D,E
2189X
2190X
052.202 315 215 052 2191X *FREAB CALL *FREAB.
052.205 320      2192X      RNC      RETURN IF OK
052.206 376 001 2193X      CPI      EC.EOF
052.210 302 016 054 2194X      JNE      *FERROR      ERROR IS NOT EOF
052.213 067      2195X      STC
052.214 311      2196X      RET      ERROR IS SIMPLY EOF
2197X
2198X
052.215          2199X *FREAB. EQU *
052.215 257      2200X      XRA      A
052.216 062 127 053 2201X      STA      EOFLG      CLEAR EOF FLAG
052.221 345      2202X      PUSH    H
052.222 315 353 052 2203X      CALL    CBT      COPY BUFFER POINTERS TO TEMP CELLS
2204X
2205X *      COPY DATA FROM BUFFER TO TARGET
2206X
052.225 325      2207X *REAB2 PUSH D      SAVE TARGET ADDRESS
052.226 072 116 053 2208X      LDA      T.FLG
052.231 346 002 2209X      ANI      FT.OR
052.233 076 011 2210X      MVI      A,EC.FNO      ASSUME FILE NOT OPEN FOR READ
052.235 067      2211X      STC
052.236 312 346 052 2212X      JZ      *REAB8      NOT OPEN FOR READ
052.241 170      2213X      MOV      A,B
052.242 261      2214X      ORA      C
052.243 312 346 052 2215X      JZ      *REAB8      ALL DONE
2216X
2217X *      COMPUTE MIN( DATA IN BUFFER, DATA REQUESTED)
2218X
052.246 052 121 053 2219X *REAB3 LHLD T.PTR
052.251 353      2220X      XCHG      (DE) = (FB.PTR) = ADDRESS OF DATA
052.252 052 123 053 2221X      LHLD T.LIM      (HL) = LIMIT ADDRESS

```

Common Decks

\$FREAB

15:15:48 02-OCT-80

```

052.255 175      2222X      MOV      A,L
052.256 223      2223X      SUB      E
052.257 157      2224X      MOV      L,A
052.260 174      2225X      MOV      A,H
052.261 232      2226X      SBB      D
052.262 147      2227X      MOV      H,A      (HL) = NUMBER OF BYTES IN BUFFER
052.263 171      2228X      MOV      A,C
052.264 225      2229X      SUB      L      COMPARE REQUESTED TO AVAILABLE
052.265 170      2230X      MOV      A,B
052.266 234      2231X      SBB      H
052.267 322 274 052 2232X      JNC      $REAB4      MORE REQUESTED THEN AVAILABLE
052.272 140      2233X      MOV      H,B
052.273 151      2234X      MOV      L,C      LIMIT TRANSFER TO REQUEST COUNT
052.274 174      2235X $REAB4 MOV      A,H
052.275 265      2236X      ORA      L
052.276 302 312 052 2237X      JNZ      $REAB6      SOME IN BUFFER
2238X
2239X *      BUFFER IS EMPTY, RE-FILL IT
2240X
052.301 315 033 053 2241X      CALL     $FFB      FILL FILE BUFFER
052.304 332 346 052 2242X      JC       $REAB8      ERROR CONDITION
052.307 303 246 052 2243X      JMP      $REAB3      COUNT NEW DATA
2244X
2245X *      GET THE DATA, MOVE IT FROM BUFFER TO TARGET
2246X *
2247X *      (BC) = REQUESTED COUNT
2248X *      (DE) = FROM
2249X *      (HL) = COUNT
2250X *      ((SP)) = TO
2251X
052.312 171      2252X $REAB6 MOV      A,C
052.313 225      2253X      SUB      L
052.314 117      2254X      MOV      C,A
052.315 170      2255X      MOV      A,B
052.316 234      2256X      SBB      H
052.317 107      2257X      MOV      B,A      REMOVE BYTES ABOUT TO BE MOVED FROM REQUEST COUNT
052.320 305      2258X      PUSH     B
052.321 343      2259X      XTHL     (HL) = REMAINING REQUEST COUNT
052.322 301      2260X      POP      B      (BC) = COUNT FOR THIS COPY
052.323 343      2261X      XTHL     (HL) = TARGET ADDR, ((SP)) = REMAINING REQ. COUNT
052.324 032      2262X $REAB7 LDAX     D
052.325 167      2263X      MOV      M,A
052.326 023      2264X      INX      D
052.327 043      2265X      INX      H
052.330 013      2266X      DCX      B
052.331 170      2267X      MOV      A,B
052.332 261      2268X      ORA      C
052.333 302 324 052 2269X      JNZ      $REAB7      MORE TO GO
052.336 353      2270X      XCHG
052.337 042 121 053 2271X      SHLD    T,PTR      UPDATE POINTER
052.342 301      2272X      POP      B      (BC) = REMAINING COUNT
052.343 303 225 052 2273X      JMP      $REAB2      SEE IF MORE IN BUFFER
2274X
2275X *      READ COMPLETE
2276X *
2277X *      (PSW) = COMPLETION FLAGS

```

Comm. Decks

.FREAB

.15:15:42 .02-DCI-80

```

2278X
052.346 321 2279X $REABB POP D RESTORE TARGET ADDRESS
052.347 341 2280X POP H
052.350 303 001 053 2281X JMP CTB COPY TEMP POINTERS BACK TO BLOCK, EXIT
052.353 2282 XTEXT FUTIL

```

```

2284X ** $FUTIL - UTILITY ROUTINES FOR FILE BLOCK ROUTINES.
2285X
2286X ** CBT - COPY BLOCK POINTERS TO TEMP CELLS.
2287X *
2288X * ENTRY (HL) = FILE BLOC FWA
2289X * EXIT NONE
2290X * USES A,F,H,L
2291X
052.353 325 2292X CBT PUSH D
052.354 305 2293X PUSH B SAVE REGISTERS
000.000 2294X ERRNZ TLEN-10 ASSUME 10 BYTES TO MOVE
052.355 021 115 053 2295X LXI D,T,CHA (DE) = TARGET FOR MOVE
052.360 006 005 2296X MVI B,10/2
052.362 176 2297X CBT1 MOV A,M COPY FILE BUFFER INTO WORK AREA
052.363 022 2298X STAX D
052.364 043 2299X INX H
052.365 023 2300X INX D
052.366 176 2301X MOV A,M
052.367 022 2302X STAX D
052.370 043 2303X INX H
052.371 023 2304X INX D
052.372 005 2305X DCR B
052.373 302 362 052 2306X JNZ CBT1 MORE TO GO
052.376 301 2307X POP B
052.377 321 2308X POP D (DE) = DATA TARGET ADDRESS
053.000 311 2309X RET
2310X
2311X

```

```

2312X ** CTB - COPY TEMP CELLS BACK TO FILE BLOCK.
2313X *
2314X * ENTRY (HL) = FILE BLOCK ADDRESS
2315X * EXIT NONE
2316X * USES NONE
2317X
053.001 365 2318X CBT PUSH PSW
053.002 325 2319X PUSH D
053.003 305 2320X PUSH B
053.004 345 2321X PUSH H SAVE REGISTERS
053.005 006 004 2322X MVI B,8/2
053.007 021 115 053 2323X LXI D,T,CHA
053.012 032 2324X CBT1 LDAX D
053.013 167 2325X MOV M,A
053.014 023 2326X INX D
053.015 043 2327X INX H
053.016 032 2328X LDAX D
053.017 167 2329X MOV M,A
053.020 023 2330X INX D

```

Common Decks

\*FUTIL

15:15:53 02-OCT-80

```

053.021 043      2331X      INX      H
053.022 005      2332X      BCR      R
053.023 302 012 053 2333X      JNZ      CTB1      RESTORE FILE BUFFER VALUES
053.024 341      2334X      POF      H
053.027 301      2335X      POF      B
053.030 321      2336X      POF      D
053.031 361      2337X      POF      PSW
053.032 311      2338X      RET

```

```

2340X **      *FFB - FILE FILE BUFFER.
2341X *
2342X *      *FFB FILLS THE FILE BUFFER BY READING FROM THE FILE.
2343X *
2344X *      ENTRY      NONE
2345X *      EXIT      'C' SET IF READ INCOMPLETE
2346X *      (A) = ERROR CODE
2347X *      'C' CLEAR IF READ COMPLETE
2348X *      DATA IN BUFFER
2349X *      USES      A:F:D:E:H:L
2350X
2351X

```

```

053.033 072 127 053 2352X *FFB LDA      EOFFLG
053.034 037      2353X      RAR
053.037 330      2354X      RC      EOF
2355X
2356X *      CAN READ MORE, DO SO
2357X
053.040 305      2358X      PUSH     B      SAVE COUNT
053.041 052 117 053 2359X      LHLI    T,FWA
053.044 042 121 053 2360X      SHLD   T,PTR      CLEAR REMOVAL POINTER
053.047 353      2361X      XCHB
053.050 052 125 053 2362X      LHLI    T,LWA
053.053 042 123 053 2363X      SHLD   T,LIM      SET DATA LIMIT
053.056 175      2364X      MOV     A,L
053.057 223      2365X      SUB     E
053.060 117      2366X      MOV     C,A
053.061 174      2367X      MOV     A,H
053.062 232      2368X      SBB     D
053.063 107      2369X      MOV     B,A      (RC) = ROOM IN BUFFER
053.064 072 115 053 2370X      LDA     T,CHA
053.067 377 094      2371X      DB      SYSCALL,READ      READ BUFFER
053.071 120      2372X      MOV     D,B      (D) = SECTORS UNREAD
053.072 301      2373X      POF     B      (RC) = DESIRED COUNT
053.073 320      2374X      RNC      GOT THE DATA
2375X
2376X *      ERROR ON READ. SEE IF EOF
2377X
053.074 027      2378X      RAL
053.075 062 127 053 2379X      STA     EOFFLG      SET EOF, WE HOPE
053.100 376 003      2380X      CPI     EC.EOF*2+1
053.102 037      2381X      RAR
053.103 300      2382X      RNE      IS NOT EOF, RETURN NOW!
053.104 072 124 053 2383X      LDA     T,LIM+1

```



Common Decks

\*FFR

15:15:53 02-OCT-80

053.107	222	2384X	SUB	D	
053.110	062 124 053	2385X	STA	T.LIM+1	SET AMOUNT OF DATA WE DID GET
053.113	247	2386X	ANA	A	
053.114	311	2387X	RET		EXIT WITH DATA
		2388X			
		2389X			
		2390X	**		TEMP CELLS TO HOLD FILE BLOCK POINTERS DURING I/O
		2391X			
000.000		2392X	ERRNZ	FB.CHA	
053.115	000	2393X	T.CHA	DB	0 CHANNEL NUMBER
000.000		2394X	ERRNZ	*-T.CHA-FB.FLG	
053.116	000	2395X	T.FLG	DB	0 FLAG BYTE
000.000		2396X	ERRNZ	*-T.CHA-FB.FWA	
053.117	000 000	2397X	T.FWA	DW	0
000.000		2398X	ERRNZ	*-T.CHA-FB.PTR	
053.121	000 000	2399X	T.PTR	DW	0
000.000		2400X	ERRNZ	*-T.CHA-FB.LIM	
053.123	000 000	2401X	T.LIM	DW	0
000.000		2402X	ERRNZ	*-T.CHA-FB.LWA	
053.125	000 000	2403X	T.LWA	DW	0
000.012		2404X	TLEN	EQU	*-T.CHA LENGTH OF TEMP CELLS
		2405X			
053.127	000	2406X	EODFLG	DB	0
053.130		2407	XTEXT	FWRIB	
		2409X	**		*FWRIB - WRITE BYTES FROM FILE BUFFER.
		2410X	*		
		2411X	*		*FWRIB IS CALLED TO WRITE A NUMBER OF BYTES FROM A FILE BUFFER.
		2412X	*		
		2413X	*	ENTRY	(BC) = BYTE COUNT
		2414X	*		(DE) = FWA FOR BYTES
		2415X	*		(HL) = ADDRESS OF FILE BUFFER
		2416X	*	EXIT	TO *FERROR* IF ERROR
		2417X	*		TO CALLER IF OK
		2418X	*		(DE) = ADDRESS OF FIRST UNWRITTEN BYTE
		2419X	*	USES	A,F,B,C,D,E
		2420X			
		2421X			
053.130	315 137 053	2422X	*FWRIB	CALL	*FWRIB.
053.133	320	2423X	RNC		RETURN IF OK
053.134	303 016 054	2424X	JMP	*FERROR	ERROR
		2425X			
		2426X			
053.137		2427X	*FWRIB.	EQU	*
053.137	345	2428X	PUSH	H	
053.140	315 353 052	2429X	CALL	CBT	COPY BUFFER POINTERS TO TEMP CELLS
		2430X			
		2431X	*		COPY DATA FROM USER AREA TO BUFFER
		2432X			
053.143	325	2433X	*WRIB2	PUSH	D
053.144	072 116 053	2434X	LDA	T.FLG	SAVE AREA ADDRESS
053.147	346 004	2435X	ANI	FT.DW	SEE IF OPEN FOR WRITE
053.151	312 305 053	2436X	JZ	*WRIB8	FILE NOT OPEN FOR WRITE

Common Decks

\$FWRIB

15:15:57 02-OCT-80

```

053.154 170      2437X      MOV      A,B
053.155 261      2438X      ORA      C
053.156 312 305 053 2439X      JZ       $WRIB8      ALL DONE
                2440X
                2441X *      COMPUTE MIN( ROOM IN BUFFER, WRITE COUNT REQUESTED)
                2442X
053.161 052 121 053 2443X $WRIB3 LHL D   T, PTR
053.164 353      2444X      XCHG                    (DE) = (FB, PTR) = ADDRESS OF ROOM
053.165 052 125 053 2445X      LHL D   T, LWA          (HL) = LIMIT ADDRESS
053.170 175      2446X      MOV      A, L
053.171 223      2447X      SUB      E
053.172 157      2448X      MOV      L, A
053.173 174      2449X      MOV      A, H
053.174 232      2450X      SBB      D
053.175 147      2451X      MOV      H, A          (HL) = BYTES OF ROOM IN BUFFER
053.176 171      2452X      MOV      A, C          COMPARE REQUESTED COUNT TO BUFFER ROOM
053.177 225      2453X      SUB      L
053.200 170      2454X      MOV      A, B
053.201 234      2455X      SBB      H
053.202 322 207 053 2456X      JNC      $WRIB4          MORE REQUESTED THEN ROOM
053.205 140      2457X      MOV      H, B
053.206 151      2458X      MOV      L, C          USE REQUESTED COUNT
053.207 174      2459X $WRIB4 MOV      A, H
053.210 245      2460X      ORA      L
053.211 302 251 053 2461X      JNZ      $WRIB6          SOME ROOM IN BUFFER
                2462X
                2463X *      BUFFER IS FULL, EMPTY IT
                2464X
053.214 305      2465X      PUSH     B              SAVE COUNT
053.215 052 117 053 2466X      LHL D   T, FWA
053.220 042 121 053 2467X      SHLD    T, PTR          CLEAR REMOVAL POINTER
053.223 353      2468X      XCHG
053.224 052 125 053 2469X      LHL D   T, LWA
053.227 175      2470X      MOV      A, L
053.230 223      2471X      SUB      E
053.231 117      2472X      MOV      C, A
053.232 174      2473X      MOV      A, H
053.233 232      2474X      SBB      D
053.234 107      2475X      MOV      B, A          (BC) = DATA IN BUFFER
053.235 072 115 053 2476X      LDA      T, CHA
053.240 377 005      2477X      DB      SYSCALL, WRITE WRITE BUFFER
053.242 301      2478X      POP      R              (BC) = DESIRED COUNT
053.243 322 161 053 2479X      JNC      $WRIB3          GOT THE DATA
                2480X
                2481X *      ERROR ON WRITE.
                2482X
053.246 303 305 053 2483X      JMP      $WRIB8          HAVE ERROR
                2484X
                2485X *      GOT THE DATA, MOVE IT FROM BUFFER TO TARGET
                2486X *
                2487X *      (BC) = REQUEST COUNT
                2488X *      (DE) = TO
                2489X *      (HL) = COUNT
                2490X *      ((SF)) = FROM
                2491X
053.251 171      2492X $WRIB6 MOV      A, C

```

Common Decks

\$FWRIB

15:15:58 02-OCT-80

```

053.252 225      2493X      SUB      L
053.253 117      2494X      MOV      C,A
053.254 170      2495X      MOV      A,B
053.255 234      2496X      SBB      H
053.256 107      2497X      MOV      B,A      REMOVE BYTES ABOUT TO BE MOVED FROM REQUEST COUNT
053.257 305      2498X      PUSH    B
053.260 343      2499X      XTHL    (HL) = REMAINING REQUEST COUNT
053.261 301      2500X      POP      (BC) = COUNT FOR THIS COPY
053.262 343      2501X      XTHL    (HL) = TARGET ADDR. ((SP)) = REMAINING REQ. COUNT
053.263 176      2502X $WRIB7  MOV      A,M
053.264 022      2503X      STAX    D
053.265 023      2504X      INX     D
053.266 043      2505X      INX     H
053.267 013      2506X      DCX     B
053.270 170      2507X      MOV     A,B
053.271 261      2508X      ORA     C
053.272 302 263 053 2509X      JNZ     $WRIB7      MORE TO GO
053.275 353      2510X      XCHG
053.276 042 121 053 2511X      SALD   T,PTR      UPDATE POINTER
053.301 301      2512X      POP     B          (BC) = REMAINING COUNT
053.302 303 143 053 2513X      JMP     $WRIB2     SEE IF MORE IN BUFFER
2514X
2515X *          WRITE COMPLETE.
2516X *
2517X *          (PSW) = COMPLETION FLAGS
2518X
053.305 321      2519X $WRIB8  POP     D          RESTORE TARGET ADDRESS
053.306 341      2520X      POP     H
053.307 303 001 053 2521X      JMP     CTR        COPY TEMP POINTERS BACK TO BLOCK, EXIT
    
```

```

2523X **          $FWBRK - BREAKOUTPUT /80.02.80/
2524X *
2525X *          $FWBRK empties the specified buffer by filling it with NULLs
2526X *          and then writing it. Note this is used to insure that block
2527X *          mode I/O is output if it is not really a serial device (eg.
2528X *          writing to AT from *EDIT*).
2529X *
2530X *
2531X *          ENTRY: HL = FILE BLOCK POINTER
2532X *
2533X *          EXIT: HL = FILE BLOCK POINTER
2534X *          TO $FERROR IF ERROR
2535X *
2536X *          USES: PSW,BC,DE
2537X *
2538X
    
```

```

053.312 315 321 053 2539X $FWBRK CALL $FWBRK.
053.315 320      2540X      RNC          NO ERROR
2541X
053.316 303 016 054 2542X      JMP     $FERROR
2543X
053.321 345      2544X $FWBRK. PUSH H
053.322 315 353 052 2545X      CALL   CBT      COPY BUFFER TO TEMPORARY
    
```

Common Decks

#FWBRK

15:13:59 02-OCT-80

```

053.325 315 335 053 2546X CALL #FWBRK1
053.330 341 2547X POP H
053.331 315 001 053 2548X CALL CTR COPY TEMPORARY TO BUFFER
053.334 311 2549X RET
2550X
053.335 052 125 053 2551X #FWBRK1 LHL D T,LWA
053.340 353 2552X XCHG DE = BUFFER LWA
2553X LHL D T, PTR HL = BUFFER PTR
053.341 052 121 053 2554X MOV A,E
053.344 173 2554X MOV A,E
053.345 225 2555X SUB L
053.346 117 2556X MOV C,A
053.347 172 2557X MOV A,D
053.350 234 2558X SBB H
053.351 107 2559X MOV B,A BC = DE - HL
053.352 261 2560X ORA C
053.353 310 2561X RZ THE BUFFER IS ALREADY FLUSHED
2562X
2563X * FILL THE BUFFER WITH NULLS
2564X
053.354 170 2565X #FWBRK2 MOV A,B
053.355 261 2566X ORA C
053.356 312 370 053 2567X JZ #FWBRK3 NO MORE LEFT TO FILL
2568X
053.361 066 000 2569X MVI M,0
053.363 043 2570X INX H
053.364 013 2571X DCX B
053.365 303 354 053 2572X JMP #FWBRK2
2573X
053.370 052 117 053 2574X #FWBRK3 LHL D T,FWA
053.373 042 121 053 2575X SHLD T, PTR DE = BUFFER FWA
053.376 353 2576X XCHG HL = BUFFER LWA
053.377 052 125 053 2577X LHL D T,LWA
054.002 175 2578X MOV A,E
054.003 223 2579X SUB E
054.004 117 2580X MOV C,A
054.005 174 2581X MOV A,H
054.006 232 2582X SBB D
054.007 107 2583X MOV B,A BC = HL - DE ( BC = COUNT )
054.010 072 115 053 2584X LDA T,CHA
054.013 377 005 2585X DB SYSCALL, WRITE
054.015 311 2586X RET
054.016 2587 XTEXT FERROR

```

```

2589X ** $FERROR -- PROCESS FILE ERRORS.
2590X *
2591X * $FERROR IS CALLED TO COMPLAIN ABOUT AN ERROR ENCONTERED
2592X * WHEN PROCESSING FILES.
2593X *
2594X * ENTRY (A) = ERROR CODE
2595X * (HL) = ADDRESS OF FILE NAME - FB.NAM
2596X * EXIT TO RESTART
2597X * USES ALL
2598X

```

Canon Decks

\*FERROR

15:16:02 02-OCT-80

```

2599X
054.016 365 2600X $FERROR PUSH PSW SAVE CODE
054.017 315 136 031 2601X CALL $TYPTX
054.022 012 007 105 2602X DB NL,BELL,'ERROR ON FILE',' '+2000
054.042 021 012 000 2603X LXI D,FB.NAM
054.045 031 2604X DAD D
2605X
2606X * PRINT FILE NAME
2607X
054.046 176 2608X $FERR1 MOV A,M
054.047 043 2609X INX H ADVANCE MESSAGE
054.050 247 2610X ANA A
054.051 312 062 054 2611X JZ $FERR2
054.054 315 232 054 2612X CALL $WCHAR
054.057 303 046 054 2613X JMP $FERR1
2614X
2615X * TYPE ERROR MESSAGE
2616X
054.062 315 136 031 2617X $FERR2 CALL $TYPTX
054.065 040 055 240 2618X DB '-',' '+2000
054.070 046 012 2619X MVI H,NL
054.072 361 2620X POP PSW (A) = CODE
054.073 377 057 2621X DB SYSCALL,'ERROR
054.075 303 006 045 2622X JMP RESTART EXIT
2623
054.100 2624 XTEXT CHL

2626X ** $CHL - COMPLEMENT (HL).
2627X *
2628X * (HL) = -(HL) TWO'S COMPLEMENT
2629X *
2630X * ENTRY NONE
2631X * EXIT NONE
2632X * USES A,F,H,L
2633X
2634X
030.224 2635X $CHL EQU 30224 IN H17 ROM
054.100 2636 XTEXT CRLF

2638X ** $CRLF - TYPE CARRIAGE RETURN/ LINE FEED
2639X *
2640X * $CRLF IS USED TO GENERATE PADDED CRLF'S.
2641X *
2642X * ENTRY NONE
2643X * EXIT (A) = 0
2644X * USES A,F
2645X
2646X
054.100 076 012 2647X $CRLF MVI A,NL
054.102 377 002 2648X DB SYSCALL,'SCOUT

```

Common Decks

\$CRLF

15:16:05 02-OCT-80

```

054.104 257      2649X      XRA      A
054.105 311      2650X      RET
054.106          2651      XTEXT    DADA
.....
2653X **      $DADA - PERFORM (HrL) = (HrL) + (OrA)
2654X *
2655X *      ENTRY (HrL) = BEFORE VALUE
2656X *      (A) = BEFORE VALUE
2657X *      EXIT (HrL) = (HrL) + (OrA)
2658X *      'C' SET IF OVERFLOW
2659X *      USES   FrHrL
2660X
2661X
030.072      2662X $DADA EQU      30072A      IN H17 ROM
054.106      2663      XTEXT    DADA2
.....
2665X **      $DADA - ADD (OrA) TO (HrL)
2666X *
2667X *      ENTRY NONE
2668X *      EXIT (HL) = (HL) + (OrA)
2669X *      USES   A,F,HrL
2670X
2671X
030.101      2672X $DADA EQU      30101A      IN H17 ROM
054.106      2673      XTEXT    HLCPE
2674X **      HLCPE - (HL) COMPARED TO (DE)
2675X *
2676X *      THIS ROUTINE IS DOUBLE WORD COMPARE OF REGISTER PAIRS (DE) AND (HL),
2677X *
2678X *      ENTRY: (HL)&(DE) SET UP
2679X *
2680X *      EXIT: (PSW) =
2681X *      'Z' SET IF (HL) = (DE)
2682X *      'C' SET IF (HL) < (DE)
2683X *      'C' CLEAR IF (HL) >= (DE)
2684X *
2685X *
2686X *      USES: (PSW)
2687X *
2688X
054.106 174      2689X HLCPE MOV      A,H
054.107 272      2690X CMP      D      'C' SET => (A) < (D)
054.110 300      2691X RNZ
054.111 175      2692X MOV      A,L
054.112 273      2693X CMP      E      'C' SET => (L) < (E)
054.113 311      2694X RET
054.114          2695      XTEXT    HLIHL

```

Common Decks

\$HLIHL

15:14:08 02-OCT-80

```

2697X ** $HLIHL - LOAD HL INDIRECT THROUGH HL.
2698X *
2699X * (HL) = ((HL))
2700X *
2701X * ENTRY NONE
2702X * EXIT NONE
2703X * USES A;H;L
2704X *
030:211 2705X $HLIHL EQU 30211A IN R17 ROM
054.114 2706 XTEXT ILDEHL

2708X ** ILDEHL - INDEXED LOAD OF DE FROM HL
2709X *
2710X * DE GET THE FULL WORD VALUE POINTED TO BY HL, AND HL IS
2711X * INCREMENTED BY TWO.
2712X *
2713X * ENTRY: HL = ADDRESS OF FULL WORD VALUE
2714X *
2715X * EXIT: DE = (HL)
2716X * HL = HL + 2
2717X *
2718X * USES: DE
2719X *
2720X *
054.114 136 2721X ILDEHL MOV E,M
054:115 043 2722X INX H
054.116 126 2723X MOV D,M
054:117 043 2724X INX H
054.120 311 2725X RET
054:121 2726 XTEXT ISDEHL

2728X ** ISDEHL - INDEXED STORE OF DE AT HL
2729X *
2730X * STORE DE AT THE ADDRESS POINTED TO BY HL, AND INCREMENT HL
2731X * BY 2.
2732X *
2733X * ENTRY: DE = VALUE
2734X * HL = ADDRESS OF VALUE
2735X *
2736X * EXIT: (HL) = DE
2737X * HL = HL + 2
2738X *
2739X * USES: HL
2740X *
2741X *
054.121 163 2742X ISDEHL MOV M,E
054:122 043 2743X INX H
054.123 162 2744X MOV M,D
054:124 043 2745X INX H
054.125 311 2746X RET

```

Common Decks

ISDEHL

15:16:10 02-OCT-80

054.126

2747

XTEXT INDL

2749X \*\* \$INDL - INDEXED LOAD.  
 2750X \*  
 2751X \* \$INDL LOADS DE WITH THE TWO BYTES AT (HL)+DISPLACEMENT  
 2752X \*  
 2753X \* THIS ACTS AS AN INDEXED FULL WORD LOAD.  
 2754X \*  
 2755X \* (DE) = ( (HL) + DISPLACEMENT )  
 2756X \*  
 2757X \* ENTRY ((RET)) = DISPLACEMENT (FULL WORD)  
 2758X \* (HL) = TABLE ADDRESS  
 2759X \* EXIT TO (RET+2)  
 2760X \* USES A,F,D,E  
 2761X  
 2762X

030.234

2763X \$INDL

EQU 30234A

IN H17 ROM

054.126

2764

XTEXT INDXX

2766X \*\* \$INDLB - INDEXED LOAD BYTE  
 2767X \*  
 2768X \* BYTE INDEXED LOAD PRIMITIVE  
 2769X \*  
 2770X \* ENTRY: HL = BASE ADDRESS  
 2771X \* (RET) = FULL WORD RELOCATION  
 2772X \*  
 2773X \* EXIT: A = ( HL + (RET) )  
 2774X \*  
 2775X \* USES: A  
 2776X \*  
 2777X \*

054.126 353

2778X \$INDLB

XCHG

DE = BASE

054.127 343

2779X

XTHL

SAVE .DE.

054.130 325

2780X

PUSH

D

SAVE .BASE

054.131 305

2781X

PUSH

B

SAVE .BC.

2782X

054.132 116

2783X

MOV C,M

054.133 043

2784X

INX H

054.134 106

2785X

MOV B,M

BC = OFFSET

054.135 043

2786X

INX H

HL = .RET.

2787X

054.136 353

2788X

XCHG

HL = BASE

054.137 011

2789X

DAD B

HL = BASE + OFFSET

054.140 176

2790X

MOV A,M

A = (.BASE + .OFFSET.)

054.141 353

2791X

XCHG

HL = .RET.

2792X

054.142 301

2793X

POP B

RESTORE .BC.

054.143 321

2794X

POP D

RESTORE .BASE

054.144 343

2795X

XTHL

HL = .DE, ; (SP) = .RET.

054.145 353

2796X

XCHG

DE = .DE, ; HL = .BASE



Common Decks

\$INDBL

15:16:11 02-OCT-80

054.146 311

2797X

RET

2799X \*\* \$INDB - INDEXED STORE

2800X \*

2801X \* INDEXED STORE PRIMITIVE.

2802X \*

2803X \* ENTRY: HL = BASE ADDRESS

2804X \* DE = VALUE TO STORE

2805X \*

2806X \* EXIT: ( HL + (RET) ) = DE

2807X \*

2808X \* USES: NONE

2809X \*

2810X

054.147 315 313 054

2811X \$INDB CALL XCHGBC

054.152 343

2812X XTHL SAVE .BC.

054.153 325

2813X PUSH D

054.154 315 114 054

2814X CALL ILDEHL DE = OFFSET

054.157 315 313 054

2815X CALL XCHGBC BC = .RET.

054.162 353

2816X XCHG DE = BASE + HL = OFFSET

054.163 031

2817X DAD D HL = BASE + OFFSET

054.164 353

2818X XCHG

054.165 343

2819X XTHL SAVE BASE

054.166 353

2820X XCHG DE = VALUE

054.167 315 121 054

2821X CALL ISDEHL

054.172 341

2822X POP H HL = BASE

054.173 315 313 054

2823X CALL XCHGBC

054.176 343

2824X XTHL RESTORE .BC.

054.177 315 313 054

2825X CALL XCHGBC

054.202 311

2826X RET

2828X \*\* \$INDB - INDEXED BYTE STORE

2829X \*

2830X \* INDEXED BYTE STORE.

2831X \*

2832X \* ENTRY: A = VALUE TO STORE

2833X \* HL = BASE ADDRESS

2834X \* (RET) = OFFSET

2835X \*

2836X \* EXIT: NONE

2837X \*

2838X \* USES: PSW

2839X \*

2840X

054.203 353

2841X \$INDB XCHG DE = BASE

054.204 343

2842X XTHL SAVE .DE.

054.205 325

2843X PUSH D SAVE BASE

054.206 305

2844X PUSH B SAVE .BC.

2845X

054.207 116

2846X MOV C,M

Common Decks

\$INDSB

15:16:11 02-OCT-80

054,210	043	2847X	INX	H	
054,211	106	2848X	MOV	B,M	BC = OFFSET
054,212	043	2849X	INX	H	HL = ,RET,
		2850X			
054,213	353	2851X	XCHG		HL = BASE
054,214	011	2852X	DAD	B	HL = BASE + OFFSET
054,215	167	2853X	MOV	M,A	( BASE + OFFSET ) = A
054,216	353	2854X	XCHG		
		2855X			
054,217	301	2856X	POP	B	RESTORE ,BC,
054,220	321	2857X	POP	D	RESTORE BASE
054,221	343	2858X	XTHL		HL = ,DE, ; (SP) = ,RET,
054,222	353	2859X	XCHG		DE = ,DE, ; HL = BASE
054,223	311	2860X	RET		
054,224		2861	XTEXT	MOVE	

2863X \*\* \$MOVE - MOVE DATA

2864X \*

2865X \* \$MOVE MOVES A BLOCK OF BYTES TO A NEW MEMORY ADDRESS.

2866X \* IF THE MOVE IS TO A LOWER ADDRESS, THE BYTES ARE MOVED FROM FIRST TO LAST,

2867X \*

2868X \*

2869X \* IF THE MOVE IS TO A HIGHER ADDRESS, THE BYTES ARE MOVED FROM

2870X \*

2871X \*

2872X \*

2873X \*

2874X \*

2875X \*

2876X \*

2877X \*

2878X \*

2879X \*

2880X \*

2881X \*

2882X

2883X

030,252

2884X

2885

054,224

2887X \*\*

2888X \*

2889X \*

2890X \*

2891X \*

2892X

2893X

054,224

2894X

054,226

2895X

054,231

2896X

\$RCHAR - READ SINGLE CHARACTER FROM CONSOLE.

ENTRY NONE

EXIT (A) = CHARACTER

USES A,F

\$RCHAR DB SYSCALL, SCIN

JC \$RCHAR NOT READY

RET

Common Decks

\*RCHAR

15:16:14 02-OCT-80

			2897X						
054.232	377	002	2898X	\$WCHAR	DB	SYSCALL, .SCOUT			
054.234	311		2899X		RET				
054.235			2900		XTEXT	TBLS			
			2902X	**		\$TBLS - TABLE SEARCH			
			2903X	*					
			2904X	*		TABLE FORMAT			
			2905X	*					
			2906X	*	DB	KEY1, VAL1,			
			2907X	*	.	.			
			2908X	*	.	.			
			2909X	*	DB	KEYN, VALN			
			2910X	*	DB	0			
			2911X	*					
			2912X	*	ENTRY	(A) = PATTERN			
			2913X	*		(H,L) = TABLE FWA			
			2914X	*	EXIT	(A) = PATTERN IF FOUND			
			2915X	*		'Z' SET IF FOUND			
			2916X	*		'Z' CLEAR IF NOT FOUND OR PATTERN=0		/78.10.GC/	
			2917X	*	USES	A,F,H,L			
			2918X						
			2919X						
054.235	305		2920X	\$TBLS	PUSH	B			
054.236	376	000	2921X		CPI	0		/78.10.GC/	
054.240	312	262 054	2922X		JZ	TBL2		/78.10.GC/	
054.243	107		2923X		MOV	B,A			
054.244	176		2924X	TBL1	MOV	A,M	(A) = CHARACTER		
054.245	043		2925X		INX	H			
054.246	270		2926X		CMF	B			
054.247	312	264 054	2927X		JZ	TBL3	IF MATCH		
054.252	247		2928X		ANA	A			
054.253	043		2929X		INX	H	SKIP PAST		
054.254	302	244 054	2930X		JNZ	TBL1	IF NOT END OF TABLE		
054.257	053		2931X		DCX	H			
054.260	053		2932X		DCX	H			
054.261	257		2933X		XRA	A	SET TO ZERO FOR OLD USERS	/78.10.GC/	
054.262	376	001	2934X	TBL2	CPI	1	CLEAR ZERO	/78.10.GC/	
			2935X						
			2936X	*	DONE				
			2937X						
054.264	301		2938X	TBL3	POP	B			
054.265	311		2939X		RET				
054.266			2940		XTEXT	TYPTX			

Common Decks

#TYPTX

15:16:16 02-OCT-80

```

2942X ** #TYPTX = TYPE TEXT.
2943X *
2944X * #TYPTX IS CALLED TO TYPE A BLOCK OF TEXT ON THE SYSTEM CONSOLE.
2945X *
2946X * IMBEDDED ZERO BYTES INDICATE A CARRIAGE RETURN LINE FEED.
2947X * A BYTE WITH THE 2000 BIT SET IS THE LAST BYTE IN THE MESSAGE.
2948X *
2949X * ENTRY (RET) = TEXT
2950X * EXIT TO (RET+LENGTH)
2951X * USES A,F
2952X
2953X
031.136 2954X #TYPTX EQU 31136A IN HI7 ROM
2955X
031.144 2956X #TYPTX EQU 31144A IN HI7 ROM
054.266 2957 XTEXT UDD

2959X ** #UDD = UNPACK DECIMAL DIGITS.
2960X *
2961X * UDD CONVERTS A 16 BIT VALUE INTO A SPECIFIED NUMBER OF
2962X * DECIMAL DIGITS. THE RESULT IS ZERO FILLED.
2963X *
2964X * ENTRY (B,C) = ADDRESS VALUE
2965X * (A) = DIGIT COUNT
2966X * (H,L) = MEMORY ADDRESS
2967X * EXIT (HL) = (HL) + (A)
2968X * USES ALL
2969X
2970X
031.157 2971X #UDD EQU 31157A IN HI7 ROM
054.266 2972 XTEXT UDD

2974X ** #UDD = UNPACK OCTAL DIGITS.
2975X *
2976X * UDD CONVERTS A SINGLE BYTE INTO 3 OCTAL DIGITS., ZERO FILL
2977X *
2978X * ENTRY (A) = BYTE VALUE
2979X * (H,L) = ADDRESS OF 3 BYTE AREA FOR DIGITS
2980X * EXIT (H,L) = (H,L)+3
2981X * USES A,H,L
2982X
2983X
054.266 305 2984X #UDD PUSH B
054.267 006 003 2985X MVI B,3 (B) = LOOP COUNT
054.271 247 2986X ANA A CLEAR CARRY
2987X
054.272 027 2988X UDD1 RAL
054.273 027 2989X RAL
054.274 027 2990X RAL
054.275 365 2991X PUSH FSW SAVE VALUE

```

054.276	346	007	2992X	ANI	7	
054.300	306	060	2993X	ADI	'0'	
054.302	167		2994X	MOV	M,A	STORE DIGIT
054.303	043		2995X	INX	H	
054.304	361		2996X	POP	PSW	RESTORE VALUE
054.305	005		2997X	DCR	B	
054.306	302	272 054	2998X	JNZ	UDD1	IF MORE TO GO
054.311	301		2999X	POP	B	RESTORE (B,C)
054.312	311		3000X	RET		EXIT
054.313			3001	XTEXT	XCHGBC	

3003X **	XCHGBC	-	XCHG BC
3004X *			
3005X *	EXCHANGE THE 'BC' REGISTER PAIR WITH THE 'HL' REGISTER PAIR.		
3006X *			
3007X *	ENTRY: BC	=	ORIGINAL BC
3008X *	HL	=	ORIGINAL HL
3009X *			
3010X *	EXIT: BC	=	ORIGINAL HL
3011X *	HL	=	ORIGINAL BC
3012X *			
3013X *	USES: BC,HL		
3014X *			
3015X			

054.313	365		3016X	XCHGBC	PUSH	PSW
054.314	170		3017X		MOV	A,B
054.315	104		3018X		MOV	B,H
054.316	147		3019X		MOV	H,A
054.317	171		3020X		MOV	A,C
054.320	115		3021X		MOV	C,L
054.321	157		3022X		MOV	L,A
054.322	361		3023X		POP	PSW
054.323	311		3024X		RET	
054.324			3025	XTEXT	ZEROS	

3027X **	8	CONSTANT ZERO BYTES.
3028X		
031.320	3029X	\$ZEROS EQU 31320A IN H17 ROM

Data

15:16:20 02-OCT-80

```

3032 **      Date
3033 *
3034
054.324..... 3035 RECORD DS      RECLEN      TEMPORARY RECORD STORAGE

```

```

3037 **      HEADING LINE,
3038 *
3039
054.334..... 3040 HEADING EQU      *
054.334..... 3041 HEAD1 DS      50
000.062..... 3042 HEAD1L EQU      *-HEAD1
055.020 011..... 3043 DB      TAB      /80.09.BB/
055.021 110 105 101..... 3044 DB      'HEATH XREF #104.06.00'
055.046 012..... 3045 DB      NL
055.047 123 131 115..... 3046 HEAD2 DS      /SYMBOL TABLE
055.105 040 040 040..... 3047 DB
000.062..... 3048 HEAD2L EQU      *-HEAD2
000.000..... 3049 ERRNZ HEAD2L-50
055.131 011..... 3050 DB      TAB      /80.09.BB/
055.132 104 104 055..... 3051 HEAD3 DS      'DD-MMM-YY'
000.011..... 3052 HEAD3L EQU      *-HEAD3
055.143 040 040 040..... 3053 DB      PAGE      /80.09.BB/
055.154..... 3054 HEAD4 DS      3
000.003..... 3055 HEAD4L EQU      *-HEAD4
055.157 012 012..... 3056 DB      NL,NL
000.223..... 3057 HEADL EQU      *-HEADING
3058
055.161 103 122 117..... 3059 SUBT2 DS      'CROSS REFERENCE TABLE'
000.025..... 3060 SUBT2L EQU      *-SUBT2
000.035..... 3061 ERRMI HEAD2L-SUBT2L
3062
3063 **      Output Line
3064 *
3065
055.204..... 3066 LINE DS      0      TEMP. OUTPUT LINE
3067
055.204..... 3068 LINEA DS      SYMBOLL
055.215 011..... 3069 DB      TAB
3070
055.216..... 3071 LINER DS      5      LINE NUMBER
000.005..... 3072 LINERL EQU      *-LINER
3073
055.223 000..... 3074 LINEC DS      0      FLAG
3075
055.224 012..... 3076 DB      NL
3077
000.017..... 3078 LINEL EQU      *-LINE
3079
3080 **      OUTPUT LINE COUNTERS, SWITCHES, ETC.
3081 *
3082
055.225 000..... 3083 LINCNT DS      0      OUTPUT LINE COUNTER
055.226..... 3084 WIDE DS      1      WIDE SWITCH

```

Data

15:16:21 02-OCT-80

055.227	3085	PAGEDP	DS	1	PAGE DEPTH
055.230	3086	FORMDP	DS	1	FORM DEPTH
055.231	3087	PAGNUM	DS	1	PAGE NUMBER

## 3089 \*\* XREF HISTORY TYPES

3090 \*

3091

055.232	3092	TYPSTRF	DS	0	
055.232	001 114	3093	DB	XT.LAB,'L'	
055.234	002 105	3094	DB	XT.EQU,'E'	
055.236	003 123	3095	DB	XT.SET,'S'	
055.240	004 116	3096	DB	XT.NRF,'N'	/WC2062680/
055.242	000 040	3097	DB	XT.REF,''	
055.244	000 000	3098	DB	0,0	

055.246	3100	PATCH	DS	64	PATCH AREA	
	3101					
055.346	3102	MEML	EQU	*		
	3103					
055.346	3104	LISTFB	DS	0	List File Block	
055.346	3105		DS	FBENL		
	3106					
056.001	3107	TEMPFB	DS	0	Temporary File Block	
056.001	3108		DS	FBENL		
	3109					
056.034	3110	DEFALTT	DS	6	Saved default device and extension for TEMPFB	
	3111					
056.042	000 000	3112	FREELST	DW	0	Pointer to Free_List for XREF-history entries
	3113					
056.044	000 000	3114	MEMLIM	DW	0	Memory limit
	3115					
056.046	3116	SYMFWA	DS	2	SYMTAB first word address	
	3117					
056.050	3118	SYMLWA	DS	2	SYMTAB last word address	

## 3120 \*\* Buffers and Tables

3121 \*

3122

056.052	3123	TMPBUF	DS	TMPBFL	
	3124				
060.052	3125	LSTBUF	DS	LSTRFL	
	3126				
061.052	3127	HASHTAB	EQU	*	
061.052	3128		DS	BHTB-BHTA+1*BHTC	Hash Table
	3129				
061.136	3130	XREFTAB	EQU	*	Resinings of Cross-Reference Table
	3131				

Data

15:16:23 02-OCT-80

061.136

3132 RMEML EQU \*

RUNNING MEMORY LIMIT

3133

061.136

3134 END PRS

ASSEMBLY COMPLETE

3134 STATEMENTS

0 ERRORS DETECTED

11708 BYTES FREE





## CROSS REFERENCE TABLE

PAGE 66

.CHFLG	000060	133L							
.CLEAN	000205	148L							
.CLEAR	000055	130L							
.CLEARA	000056	131L							
.CLOSE	000046	123L	2135						
.CLRCD	000007	107L							
.CONSL	000006	106L							
.CTLC	000041	118L	1252						
.DAD	000206	149L							
.DECODE	000053	128L							
.DELET	000050	125L	582						
.DISMT	000061	134L							
.DMNMS	000203	146L							
.DMOUN	000201	144L							
.ERROR	000057	132L	921	2621					
.EXIT	000000	100L	564	2037	2043				
.LINK	000040	117L							
.LOADD	000062	135L							
.LOADQ	000010	108L							
.MONMS	000202	145L							
.MOUNT	000200	143L							
.NAME	000054	129L	1239						
.OPEN	000063	136L							
.OFENC	000045	122L							
.OFENR	000042	119L	2054						
.OFENU	000044	121L	2056						
.OFENW	000043	120L	2055						
.POSIT	000047	124L							
.PRINT	000003	103L	547						
.READ	000004	104L	2371						
.RENAM	000051	126L							
.RESET	000204	147L							
.SCIN	000001	101L	2894						
.SCOUT	000002	102L	2648	2898					
.SETTF	000052	127L	423	555	1257	1261			
.SYSRES	000012	110L							
.VERS	000011	109L	429						
.WRITE	000005	105L	2123	2477	2585				
.ABS.CDD	000010	260L	413						
.ABS.ENT	000006	258L							
.ABS.ID	000000	254L							
.ABS.LDA	000002	256L							
.ABS.LEN	000004	257L							
.AHR	043275	605L	844						
.AHR1	043300	607L	613						
.ANS	043355	605	643L	669	911	1548	1595	1636	
.ANS1	043356	645L	647						
.ANSA	000003	652	656E						
.ANX	043367	669L	876	1018	1097				
.BCGF1	043375	688L	1578						
.BCGF1.	043376	689E	1570	1598	1600				
.BCGF2	044003	709L	1562	1581					
.BCGF2.	044004	710E	1571	1609	1611				
.BCGPSZ	044011	729L	1542	1568					
.BCGPSZ.	044012	730E	1536						
.BCRT	044017	537	742L						
.BCRT1	044070	742	762L						
.BCRT2	044144	746	794L	807					

## CROSS REFERENCE TABLE

BCRT3	044173	757	815E	829		
BCRT4	044226	828	831L			
BCRT5	044302	833	844L			
BELL	000007	65E	468	923	969	2602
BHT	044306	756	861L			
BHT1	044316	865L	877	891		
BHT2	044340	867	872	881L		
BHTA	000101	863	895E	992	994	3128
BHTB	000132	890	896E	994	3128	
BHTC	000002	897E	3128			
BKSP	000010	67E				
C.STX	000002	69E				
C.SYN	000026	68E				
CBT	052353	2203	2292L	2429	2545	
CBT1	052362	2297L	2306			
CCHIT	043236	568E	1251			
CN.TMP	000004	28E				
CO.FLG	000001	228E				
COL	051037	1104	1170	1303	1823E	
COL3	051054	1830	1834E			
CR	000015	61E				
CRTHL	000005	34E				
CS.FLG	000200	229E				
CSE	044356	904L	1514			
CSE1	044364	907L	913			
CSL.CHR	000001	205E				
CSL.ECH	000200	202E				
CSL.RAW	000004	203E				
CSL.WRP	000002	204E				
CTB	053001	2281	2318L	2521	2548	
CTB1	053012	2324L	2333			
CTLA	000001	76E				
CTLB	000002	77E				
CTLC	000003	78E	1250			
CTLD	000004	79E				
CTLO	000017	80E				
CTLP	000020	81E				
CTLQ	000021	82E				
CTLS	000023	83E				
CTLZ	000032	84E				
CTP.2SB	000010	214E				
CTP.BKM	000002	215E				
CTP.BKS	000200	210E				
CTP.FF	000100	211E				
CTP.MLI	000040	212E				
CTP.MLO	000020	213E				
CTP.TAB	000001	216E				
D.CUN	040110	164L				
D.RAM	040240	167L				
D.VEC	040130	166L				
DEFALTT	056034	580	1237	1244	3110L	
DELTMP	043251	561	571	576E		
DF.CLR	000376	332E				
DF.EMP	000377	331E				
DIR.ALD	000025	347L				
DIR.CLU	000015	340L				
DIR.CRD	000023	346L				
DIR.EXT	000010	335L				

## CROSS REFERENCE TABLE

DIR.FGN	000020	343L								
DIR.FLG	000016	341L								
DIR.LGN	000021	344L								
DIR.LSI	000022	345L								
DIR.NAM	000000	334L								
DIR.PRO	000013	336L								
DIR.VER	000014	337L								
DIRELEN	000027	349E	381							
DIRIDL	000015	338E								
EC.CNA	000004	284L								
EC.DDA	000027	303L								
EC.DIF	000017	295L								
EC.DIW	000035	309L								
EC.DNI	000045	317L								
EC.DNR	000046	318L								
EC.DNS	000005	285L								
EC.DSC	000047	319L								
EC.EOF	000001	281L	2193	2380						
EC.EQM	000002	282L								
EC.FAO	000031	305L	2007							
EC.FAP	000026	302L								
EC.FL	000030	304L								
EC.FNF	000014	292L								
EC.FNO	000011	289L	2210							
EC.FNR	000034	308L								
EC.FOD	000043	315L								
EC.FUC	000013	291L								
EC.ICN	000016	294L								
EC.IDN	000006	286L								
EC.IFC	000020	296L								
EC.IFN	000007	287L								
EC.ILC	000003	283L								
EC.ILO	000040	312L								
EC.ILR	000012	290L								
EC.ILV	000037	311L								
EC.IOI	000052	322L								
EC.IS	000032	306L								
EC.NCV	000050	320L	589							
EC.NEM	000021	297L	809	832	949					
EC.NOS	000051	321L								
EC.NFM	000044	316L								
EC.NRD	000010	288L								
EC.NVM	000042	314L								
EC.OTL	000053	323L								
EC.RF	000022	298L								
EC.UNA	000036	310L								
EC.UND	000015	293L								
EC.UUN	000033	307L								
EC.VPM	000041	313L								
EC.WF	000023	299L								
EC.WF	000025	301L								
EC.WPV	000024	300L								
ENL	000212	74E	469	487	839	1505				
EOFFLG	053127	220I	2352	2379	2406L					
ERRDR	044376	424	556	590	810	919E	950	1159	1240	1262
ESC	000033	72E								
EXIT	043212	54B	552E	840						
EXIT.	043234	564L	927							





PCRTA	046132	1158	1177L	1186				
PCRTAL	000012	1158	1186E					
PCRTB	046136	1128	1131	1181L	1182			
PCRTBL	000005	1125	1132	1182E				
PCRTC	046143	1144	1151	1184L				
PCRTD	046144	1095	1169	1189L	1190			
PCRTDL	000001	1095	1169	1190E				
PCRTE	046145	1110	1173	1192L	1275			
PCRTF	046146	1111	1153	1161	1174	1194L		
PCRTG	046147	1172	1196L	1203				
PCRTGL	000022	1172	1203E	1204				
PRS	046171	528	1217E					
PRSX	046362	1273	1279E					
PST	046363	531	1287E					
PST1	046374	1292L	1297	1322	1330			
PST1.3	047040	1315L						
PST1.6	047062	1311	1326L					
PST2	047076	1102	1296	1343L				
PST2P	047117	1105	1306	1357E				
PST3	047145	1366	1370L					
PST4	047160	1372	1376L					
PST5	047173	1378	1382L					
PST6	047206	1384	1388L					
PST7	047241	1352	1415L					
PST8	047247	1294	1424E					
PSTA	047257	1359	1368	1380	1407	1430L	1446	
PSTB	047261	1347	1350	1434L	1435			
PSTBL	000007	1435E	1436					
PSTC	047271	1397	1440L	1441				
PSTCL	000008	1401	1441E					
PSTD0	047277	1360	1386	1443L				
PSTD1	047300	1361	1374	1444L				
PSTE	047301	1277	1289	1301	1319	1449L		
PSTF	047302	1290	1300	1308	1310	1320	1451L	
PSTG	047303	1317	1426	1453L	1454			
PSTGL	000001	1317	1426	1454E				
PSTI	047304	1328	1456L	1457				
PSTIL	000002	1328	1457E					
PSTL	000022	1204	1407	1446E				
PXS	047308	542	1468E					
PXS4	047331	1479	1482E					
PXSA	047376	1492	1502L	1504				
PXSAL	000023	1491	1504E					
PXSB	050000	548	1483	1503L				
QUOTE	000047	70E						
RECLEN	000012	45E	817	3035				
RECORD	054324	627	632	818	823	827	3035L	
REFTYP	000011	43L	632					
RESTART	045006	470	488	572	925E	2622		
RMEML	061136	422	475	554	3132E			
ROMBOOT	030000	159E						
RUBOOT	000177	66E						
S.CAADR	040333	235L						
S.CCTAB	040335	238L						
S.CONFL	040332	233L						
S.CONTY	040327	220L						
S.CONWI	040331	226L						
S.CSLMD	040326	208L	219	222	225	232		





## CROSS REFERENCE TABLE

T.PTR	053121	2219	2271	2360	2399L	2443	2467	2511	2553	2575				
TAB	000011	71E	3043	3050	3069									
TBL1	054244	2924L	2930											
TBL2	054262	2922	2934L											
TBL3	054264	2927	2938L											
TEMPFB	056001	499	577	581	819	1229	1231	1232	1234	1236	1238	1242	1245	
		3107L												
TLEN	000012	2294	2404E											
TMPBFL	002000	30E	1230	3123										
TMPBUF	056052	1228	1230	3123L										
TSE	050337	798	1033	1723L										
TSE1	050341	1727L	1733											
TSE2	050354	1738L	1744											
TYPSRF	055232	1147	3092L											
UOD1	054272	2988L	2998											
USERFWA	042200	176E	413	415	416									
USN	050366	1348	1763E											
USN1	050370	1766L	1775											
USN2	051007	1779L	1783											
VERERR	043270	430	432	588E										
VER8	000040	91E	431											
WIDE	055226	522	1271	3084L										
WLD	051017	1094	1157	1168	1171	1316	1327	1406	1425	1800L				
XCHGBC	054313	1048	1515	1569	2811	2815	2823	2825	3016L					
XH.FLG	000004	53L	634	1146										
XH.LEN	000005	55E	941											
XH.LIN	000002	51L	630	1124										
XH.LNK	000000	49L	610	624	1108	1119								
XREFTAB	061136	745	862	987	1085	3130E								
XT.EQU	000002	407E	3094											
XT.LAB	000001	406E	3093											
XT.NRF	000004	409E	3096											
XT.REF	000000	405E	3097											
XT.SET	000003	408E	3095											

21144 BYTES FREE

