

000.001

```
1 PUBLIC EQU 1 NOT PUBLIC
3 *** SET - SET SYSTEM PARAMETERS.
4 *
5 * JGL, 1/16/78 FOR *HEATH* COMPANY
6 *
7 * COPYRIGHT 1978 BY HEATH COMPANY

9 *** SET - SET OPTIONS.
10 *
11 * SET DEV: OPTIONS
12 *
13 *
14 * SET HELP LIST GENERAL HELP OPTIONS
15 *
16 * SET TT: BKS TERMINAL PROCESSES BACKSPACES
17 * SET TT: MLI MAP LOWER CASE TO UPPER ON INPUT
18 * SET TT: HLO MASK LOWER CASE TO UPPER ON OUTPUT
19 * SET TT: BKM MAP BKSP TO RUBOUT
20 * SET TT: TAB SEND TAB CHARACTERS TO TERMINAL
21 * SET TT: FILL CC NN PAD CHARACTER CC WITH NN NULLS
22 * SET TT: HELP LIST SET OPTIONS FOR TT:
23 *
24 * SET SY: STEP NN SET TRACK STEP TIME TO NN
25 * SET SY: HELP LIST SET OPTIONS FOR SY:
26 *
27 * SET HDOS HELP LIST HDOS SETTABLE PARAMETERS
28 * SET HDOS STAND-ALONE ENABLE HDOS TO RUN STAND-ALONE
29 *
30 *
31 *** IN ADDITION TO THE ABOVE 'BUILT-IN' OPTIONS, SET HAS THE
32 * ABILITY TO SET OPTIONS IN DISK-RESIDENT DEVICE DRIVERS. THE
33 * FORMAT FOR THE COMMAND IS:
34 *
35 * SET dev: <OPTIONS>
36 *
37 * WHERE 'DEV:' IS A DEVICE AND (OPTIONALLY) A UNIT NUMBER.
38 *
39 * SET WILL LOAD THE DEVICE DRIVER INTO MEMORY, RELOCATE THE
40 * 'SET PREAMBLE' (THAT CODE BELOW DVD.ENT) AND CALL THE
41 * PREAMBLE CODE WITH
42 *
43 * (DE) = ADDRESS OF <OPTIONS> STRING
44 * (A) = UNIT NUMBER
45 *
46 * THE DEVICE DRIVER PREAMBLE IS RESPONSIBLE FOR ANALYZING THE
47 * OPTION STRING (WITH THE HELP OF SOME ROUTINES IN *SET*) AND
48 * SETTING ITS OWN OPTION FLAGS. WHEN IT RETURNS TO SET,
49 * *SET* WILL POST THE DRIVER BACK TO THE DISK.
```

SYMBOL

14:49:26 16-MAY-80

51 **** ASSEMBLY CONSTANTS

52

000.000

53 GN.DVD EQU 0 CHANNEL NUMBER FOR READING/WRITING DEVICE DRIVERS

54

55 ****

000.000	58	XTEXT	ASCII	
	60X **	ASCII CHARACTER EQUIVALENCES.		
	61X			
000.015	62X CR	ERU	13	CARRIAGE RETURN
000.012	63X LF	ERU	10	LINE FEED
000.200	64X NULL	ERU	2000	PAD CHARACTER
000.000	65X NUL2	ERU	0	
000.007	66X BELL	ERU	7	BELL CHARACTER
000.177	67X RUBOUT	ERU	1770	
000.010	68X BKSP	ERU	100	CTL-H
000.026	69X C.SYN	ERU	260	SYNC
000.002	70X C.STX	ERU	2	STX
000.047	71X QUOTE	ERU	470	
000.011	72X TAB	ERU	110	
000.033	73X ESC	ERU	330	
000.012	74X NL	ERU	120	NEW LINE (HDOS SYSTEMS)
000.212	75X ENL	ERU	NL+2000	NL + END-OF-LINE-FLAG
000.014	76X FF	ERU	140	FORM FEED
000.001	77X CTLA	ERU	010	CTL-A
000.002	78X CTLB	ERU	020	CTL-B
000.003	79X CTLC	ERU	030	CTL-C
000.004	80X CTLD	ERU	040	CTL-D
000.017	81X CTLO	ERU	170	CTL-O
000.020	82X CTLP	ERU	200	CTL-P
000.021	83X CTLQ	ERU	210	CTL-Q
000.023	84X CTLS	ERU	230	CTL-S
000.032	85X CTLZ	ERU	320	CTL-Z
000.000	86	XTEXT	HOSDEF	
	88X **	HOSDEF - DEFINE HOS PARAMETER.		
	89X *			
	90X			
	91X			
000.026	92X VERS	ERU	1*16+6	VERSION 1.6
	93X			
000.377	94X SYSCALL	ERU	3770	SYSCALL INSTRUCTION
	95X			
	96X			
000.000	97X	ORG	0	
	98X			
	99X *	RESIDENT FUNCTIONS		
	100X			
000.000	101X .EXIT	DS	1	EXIT (MUST BE FIRST)
000.001	102X .SCIN	DS	1	SCIN
000.002	103X .SCOUT	DS	1	SCOUT
000.003	104X .PRINT	DS	1	PRINT
000.004	105X .READ	DS	1	READ
000.005	106X .WRITE	DS	1	WRITE
000.006	107X .CONSL	DS	1	SET/CLEAR CONSOLE OPTIONS
000.007	108X .CLRCD	DS	1	CLEAR CONSOLE BUFFER
000.010	109X .LOADO	DS	1	LOAD AN OVERLAY
000.011	110X .VERS	DS	1	RETURN HDOS VERSION NUMBER

000.012 111X .SYSRES DS 1 PRECEDING FUNCTIONS ARE RESIDENT

112X
113X

114X * *HDOSOVLO.SYS* FUNCTIONS

115X

000.040 116X ORG 40A

117X

000.040 118X .LINK DS 1 LINK (MUST BE FIRST)

000.041 119X .CTLCL DS 1 CTL-C

000.042 120X .OPENR DS 1 OPENR

000.043 121X .OPENW DS 1 OPENW

000.044 122X .OPENU DS 1 OPENU

000.045 123X .OFENC DS 1 OFENC

000.046 124X .CLOSE DS 1 CLOSE

000.047 125X .POSIT DS 1 POSITION

000.050 126X .DELET DS 1 DELETE

000.051 127X .RENAM DS 1 RENAME

000.052 128X .SETTP DS 1 SETTOP

000.053 129X .DECODE DS 1 NAME DECODE

000.054 130X .NAME DS 1 GET FILE NAME FROM CHANNEL

000.055 131X .CLEAR DS 1 CLEAR CHAN

000.056 132X .CLEARA DS 1 CLEAR ALL CHANS

000.057 133X .ERROR DS 1 LOOKUP ERROR

000.060 134X .CHFLG DS 1 CHANGE FLAGS

000.061 135X .DISMT DS 1 FLAG SYSTEM DISK DISMOUNTED

000.062 136X .LOADD DS 1 LOAD DEVICE DRIVER

137X

138X

139X * *HDOSOVLI.SYS* FUNCTIONS

140X

000.200 141X ORG 2000

142X

000.200 143X .MOUNT DS 1 MOUNT (MUST BE FIRST)

000.201 144X .DMOUN DS 1 DISMOUNT

000.202 145X .MOMMS DS 1 MOUNT/NO MESSAGE

000.203 146X .DMNMS DS 1 DISMOUNT/NO MESSAGE

000.204 147X .RESET DS 1 RESET = DISMOUNT/MOUNT OF UNIT

000.205 148 XTEXT DIRDEF

149

150X

151X

152X ** DIRECTORY ENTRY FORMAT,

000.000 152X ORG 0

153X

154X

000.377 155X DF.EMP EQU 3770 FLAGS ENTRY EMPTY

000.376 156X DF.CLR EQU 3760 FLAGS ENTRY EMPTY, REST OF DIR ALSO CLEAR

157X

000.000 158X DIR.NAM DS 8 NAME

000.010 159X DIR.EXT DS 3 EXTENSION

000.013 160X DIR.PRO DS 1 PROJECT

000.014 161X DIR.VER DS 1 VERSION

000.015 162X DIR.IDL EQU * FILE IDENTIFICATION LENGTH

163X

000.015 164X DIR.CLU DS 1 CLUSTER FACTOR

DIR

000.016	165X	DIR.FLG	DS	1	FLAGS
000.017	166X		DS	1	RESERVED
000.020	167X	DIR.FGN	DS	1	FIRST GROUP NUMBER
000.021	168X	DIR.LGN	DS	1	LAST GROUP NUMBER
000.022	169X	DIR.LSI	DS	1	LAST SECTOR INDEX (IN LAST GROUP)
000.023	170X	DIR.CRD	DS	2	CREATION DATE
000.025	171X	DIR.ALD	DS	2	LAST ALTERATION DATE
	172X				
000.027	173X	DIRELEN	EQU	*	DIRECTORY ENTRY LENGTH
000.027	174	XTEXT	ECDEF		

176X ** ERROR CODE DEFINITIONS.

000.000	177X				
000.000	178X	ORG		0	
000.000	179X		DS	1	NO ERROR #0
000.001	180X	EC.EOF	DS	1	END OF FILE
000.002	181X	EC.EOM	DS	1	END OF MEDIA
000.003	182X	EC.ILC	DS	1	ILLEGAL SYSCALL CODE
000.004	183X	EC.CNA	DS	1	CHANNEL NOT AVAILABLE
000.005	184X	EC.DNS	DS	1	DEVICE NOT SUITABLE
000.006	185X	EC.IDN	DS	1	ILLEGAL DEVICE NAME
000.007	186X	EC.IFN	DS	1	ILLEGAL FILE NAME
000.010	187X	EC.NRD	DS	1	NO ROOM FOR DEVICE DRIVER
000.011	188X	EC.FNO	DS	1	CHANNEL NOT OPEN
000.012	189X	EC.ILR	DS	1	ILLEGAL REQUEST
000.013	190X	EC.FUC	DS	1	FILE USAGE CONFLICT
000.014	191X	EC.FNF	DS	1	FILE NAME NOT FOUND
000.015	192X	EC.UND	DS	1	UNKNOWN DEVICE
000.016	193X	EC.ICN	DS	1	ILLEGAL CHANNEL NUMBER
000.017	194X	EC.DIF	DS	1	DIRECTORY FULL
000.020	195X	EC.IFC	DS	1	ILLEGAL FILE CONTENTS
000.021	196X	EC.NEM	DS	1	NOT ENOUGH MEMORY
000.022	197X	EC.RF	DS	1	READ FAILURE
000.023	198X	EC.WF	DS	1	WRITE FAILURE
000.024	199X	EC.WPV	DS	1	WRITE PROTECTION VIOLATION
000.025	200X	EC.WP	DS	1	DISK WRITE PROTECTED
000.026	201X	EC.FAP	DS	1	FILE ALREADY PRESENT
000.027	202X	EC.DDA	DS	1	DEVICE DRIVER ABORT
000.030	203X	EC.FL	DS	1	FILE LOCKED
000.031	204X	EC.FAO	DS	1	FILE ALREADY OPEN
000.032	205X	EC.IS	DS	1	ILLEGAL SWITCH
000.033	206X	EC.UUN	DS	1	UNKNOWN UNIT NUMBER
000.034	207X	EC.FNR	DS	1	FILE NAME REQUIRED
000.035	208X	EC.DIW	DS	1	DEVICE IS NOT WRITABLE (OR WRITE LOCKED)
000.036	209X	EC.UNA	DS	1	UNIT NOT AVAILABLE
000.037	210X	EC.ILV	DS	1	ILLEGAL VALUE
000.040	211X	EC.ILO	DS	1	ILLEGAL OPTION
000.041	212X	EC.VPM	DS	1	VOLUME PRESENTLY MOUNTED ON DEVICE
000.042	213X	EC.NVM	DS	1	NO VOLUME PRESENTLY MOUNTED
000.043	214X	EC.FOP	DS	1	FILE OPEN ON DEVICE
000.044	215X	EC.NPM	DS	1	NO PROVISIONS MADE FOR REMOUNTING MORE DISKS
000.045	216X	EC.DNI	DS	1	DISK NOT INITIALIZED
000.046	217X	EC.DNR	DS	1	DISK IS NOT READABLE
000.047	218X	EC.DSC	DS	1	DISK STRUCTURE IS CORRUPT

000.050	219X	EC.NCV	DS	1	NOT CORRECT VERSION OF HDOS
000.051	220X	EC.NOS	DS	1	NO OPERATING SYSTEM MOUNTED
000.052	221X	EC.IOT	DS	1	ILLEGAL OVERLAY INDEX
000.053	222X	EC.OTL	DS	1	OVERLAY TOO LARGE
000.054	223		XTEXT	IOCDEF	
	225X	**			I/O CHANNEL DEFINITIONS.
	226X				
000.000	227X		ORG	0	
	228X				
000.000	229X	IOC.LNK	DS	2	ADDRESS OF NEXT CHANNEL, #0 IF LAST
000.002	230X	IOC.DDA	DS	2	THREAD JUMP TO DEVICE DRIVER (VIA DEV TABLE)
	231X				
000.004	232X	IOC.FLG	DS	1	FILE TYPE FLAGS
000.001	233X	FT.DD	EQU	00000001B	=1 IF DIRECTORY DEVICE
000.002	234X	FT.OR	EQU	00000010B	=1 IF OPEN FOR READ
000.004	235X	FT.DW	EQU	00000100B	=1 IF OPEN FOR WRITE
000.010	236X	FT.OU	EQU	00001000B	=1 IF OPEN FOR UPDATE
000.003	237X	IOC.SQL	EQU	*-IOC.DDA	LENGTH OF INFO FOR SEQUENTIAL FILE (FROM IOC)
	238X				
000.005	239X	IOC.GRT	DS	2	ADDRESS OF GROUP RESERVATION TABLE
000.007	240X	IOC.SPG	DS	1	SECTORS PER GROUP, THIS DEVICE
000.010	241X	IOC.CGN	DS	1	CURRENT GROUP NUMBER
000.011	242X	IOC.CSI	DS	1	CURRENT SECTOR INDEX (IN CURRENT GROUP)
000.012	243X	IOC.LGN	DS	1	LAST GROUP NUMBER
000.013	244X	IOC.LSI	DS	1	LAST SECTOR INDEX (IN LAST GROUP)
000.010	245X	IOC.DRL	EQU	*-IOC.FLG	LENGTH OF INFO NORMALLY COPIED BACK TO THE CHANNEL TABLE
	246X	*			DEVICE TABLE ADDRESS FOR THIS DEVICE
000.014	247X	IOC.DTA	DS	2	SECTOR NUMBER OF DIRECTORY ENTRY
000.016	248X	IOC.DES	DS	2	DEVICE CODE
000.020	249X	IOC.DEV	DS	2	UNIT NUMBER (0-9)
000.022	250X	IOC.UNI	DS	1	LENGTH OF INFO FOR DIRECTORY FILE (FROM IOC)
000.021	251X	IOC.DIL	EQU	*-IOC.DDA	
	252X				
000.023	253X	IOC.DIR	DS	DIRLEN	DIRECTORY ENTRY
	254X				
000.052	255X	IOCELEN	EQU	*	IOC ENTRY LENGTH
	256X				
000.001	257X	IOCCID	EQU	1	INDEX OF USER CHANNEL #0 IN CHANTAB (FIRST = 0)
000.052	258		XTEXT	HOSEQU	
	260X	**			HDOS SYSTEM EQUIVALENCES.
	261X	*			
	262X				
024.000	263X	S.GRT0	EQU	24000A	SYSTEM AREA FOR GRT0
025.000	264X	S.GRT1	EQU	25000A	SYSTEM AREA FOR GRT1
026.000	265X	S.GRT2	EQU	26000A	SYSTEM AREA FOR GRT2
	266X				
030.000	267X	ROMBOOT	EQU	30000A	ROM BOOT ENTRY
	268X				
040.100	269X		ORG	40100A	FREE SPACE FROM PAM-8

	270X				
040.100	271X	DS	8		JUMP TO SYSTEM EXIT
040.110	272X D.CON	DS	16		DISK CONSTANTS
040.130	273X SYDD	EQU	*		SYSTEM DISK ENTRY POINT
040.130	274X D.VEC	DS	24*3		SYSTEM ROM ENTRY VECTORS
040.240	275X D.RAM	DS	31		SYSTEM ROM WORK AREA
040.277	276X S.VAL	DS	36		SYSTEM VALUES
040.343	277X S.INT	DS	115		SYSTEM INTERNAL WORK AREAS
041.126	278X	DS	16		
041.146	279X S.SOVR	DS	2		STACK OVERFLOW WARNING
041.150	280X	DS	42200A-*		SYSTEM STACK
001.032	281X STACKL	EQU	*-S.SOVR		STACK SIZE
	282X				
042.200	283X STACK	EQU	*		LWA+1 SYSTEM STACK
042.200	284X USERFWA	EQU	*		USER FWA
042.200	285	XTEXT	EDCON		
	287X **				D.CON DETAILED EQUIVALENCES.
	288X *				
	289X *				HDOSEQU MUST BE MODIFIED WHEN THIS TABLE IS MODIFIED.
	290X				
040.110	291X	ORG	D.CON		
	292X				
040.110	293X D.XITA	DS	2		SEE SYSTEM ROM FOR DESCRIPTION
040.112	294X D.WRITA	DS	1		
040.113	295X D.WRITE	DS	1		
040.114	296X D.WRITC	DS	1		
040.115	297X D.MAIA	DS	1		
040.116	298X D.LPSA	DS	1		
040.117	299X D.SDPA	DS	1		
040.120	300X D.SDPB	DS	1		
040.121	301X D.STSA	DS	1		
040.122	302X D.STSB	DS	1		
040.123	303X D.WHDA	DS	1		
040.124	304X D.WNHA	DS	1		
040.125	305X D.WSCA	DS	1		
	306X				
040.126	307X D.ERTS	DS	2		TRACK AND SECTOR OF LAST DISK ERRORS
040.130	308	XTEXT	ESVAL		
	310X **				S.VAL - SYSTEM VALUE DEFINITIONS.
	311X *				
	312X *				THESE VALUES ARE SET AND MAINTAINED BY THE SYSTEM.
	313X *				
	314X *				THE DECK HDOSEQU MUST BE MODIFIED WHEN THIS IS MODIFIED.
	315X				
	316X				
040.277	317X	ORG	S.VAL		
	318X				
040.277	319X S.DATE	DS	9		SYSTEM DATE (IN ASCII)

SET - SET SYSTEM PARAMETERS
COMMON DECK DEFINITIONS

ESVAL

HEATH HBASH V1.4 01/20/78
14:49:48 16-MAY-80

PAGE 8

040.310	320X	S.DATC	DS	2	CODED DATE
040.312	321X	S.TIME	DS	4	TIME FROM MIDNIGHT (IN TICS)
040.316	322X	S.HIMEM	DS	2	HARDWARE HIGH MEMORY ADDRESS#1
	323X				
040.320	324X	S.SYSM	DS	2	FWA RESIDENT SYSTEM
	325X				
040.322	326X	S.USRM	DS	2	LWA USER MEMORY
	327X				
040.324	328X	S.OMAX	DS	2	MAX OVERLAY SIZE FOR SYSTEM
	329X				
	330X				
	331X	**			THE FOLLOWING FIVE CELLS SHOULD BE MODIFIED/READ ONLY VIA THE .CONSL SYSCALL
	332X				
000.200	333X	CSL.ECH	EQU	10000000B	SUPPRESS ECHO
000.002	334X	CSL.WRF	EQU	00000010B	WRAP LINES AT WIDTH
000.001	335X	CSL.CHR	EQU	00000001B	OPERATE IN CHARACTER MODE
	336X				
000.000	337X	I.CSLMD	EQU	0	S.CSLMD IS FIRST BYTE
040.326	338X	S.CSLMD	DS	1	CONSOLE MODE
	339X				
000.200	340X	CTP.BKS	EQU	10000000B	TERMINAL PROCESSES BACKSPACES
000.040	341X	CTP.MLI	EQU	00100000B	MAP LOWER CASE TO UPPER ON INPUT
000.020	342X	CTP.MLO	EQU	00010000B	MAP LOWER CASE TO UPPER ON OUTPUT
000.010	343X	CTP.2SB	EQU	00001000B	TERMINAL NEEDS TWO STOP BITS
000.002	344X	CTP.BKM	EQU	00000010B	MAP BKSP (UPON INPUT) TO RUBOUT
000.001	345X	CTP.TAB	EQU	00000001B	TERMINAL SUPPORTS TAB CHARACTERS
	346X				
000.001	347X	I.CONTY	EQU	1	S.CONTY IS 2ND BYTE
000.000	348X	ERRNZ	*	*-S.CSLMD-I.CONTY	
040.327	349X	S.CONTY	DS	1	CONSOLE TYPE FLAGS
000.002	350X	I.CUSOR	EQU	2	S.CUSOR IS 3RD BYTE
000.000	351X	ERRNZ	*	*-S.CSLMD-I.CUSOR	
040.330	352X	S.CUSOR	DS	1	CURRENT CURSOR POSITION
000.003	353X	I.CONWI	EQU	3	S.CONWI IS 4TH BYTE
000.000	354X	ERRNZ	*	*-S.CSLMD-I.CONWI	
040.331	355X	S.CONWI	DS	1	CONSOLE WIDTH
	356X				
000.001	357X	CO.FLG	EQU	00000001B	CTL-O FLAG
000.200	358X	CS.FLG	EQU	10000000B	CTL-S FLAG
	359X				
000.004	360X	I.CONFL	EQU	4	S.CONFL IS 5TH BYTE
000.000	361X	ERRNZ	*	*-S.CSLMD-I.CONFL	
040.332	362X	S.CONFL	DS	1	CONSOLE FLAGS
	363X				
040.333	364X	S.CAADR	DS	2	ADDRESS FOR ABORT PROCESSING (>256 IF VALID)
040.335	365X	S.CCTAB	DS	6	ADDR FOR CTL-A, CTL-B, CTL-C PROCESSING
040.343	366	XTEXT	ESINT		


```

368X **      S.INT - SYSTEM INTERNAL WORKAREA DEFINITIONS.
369X *
370X *
371X *      THESE CELLS ARE REFERENCED BY OVERLAYS AND MAIN CODE, AND
372X *      MUST THEREFORE RESIDE IN FIXED LOW MEMORY.
373X
040.343      374X      ORG      S.INT
375X
376X **      CONSOLE STATUS FLAGS
377X
040.343      378X S.CDB   DS      1      CONSOLE DESCRIPTOR BYTE
000.000      379X CDB.HB5 EQU      0000000B
000.001      380X CDB.HB4 EQU      00000001B      =0 IF HB-5, =1 IF HB-4
040.344      381X S.BAUD DS      2      [0-14] HB-4 BAUD RATE, =0 IF HB-5
382X *      [15] =1 IF BAUD RATE => 2 STOP BITS
383X
384X **      TABLE ADDRESS WORDS
385X
040.346      386X S.DLINK DS      2      ADDRESS OF DATA IN HDOS CODE
040.350      387X S.OFWA DS      2      FWA OVERLAY TABLE
040.352      388X S.CFWA DS      2      FWA CHANNEL TABLE
040.354      389X S.DFWA DS      2      FWA DEVICE TABLE
040.356      390X S.RFWA DS      2      FWA RESIDENT HDOS CODE
391X
392X **      DEVICE DRIVER DELAYED LOAD FLAGS
393X
040.360      394X S.DDLDA DS      2      DRIVER LOAD ADDRESS (HIGH BYTE=0 IF NO LOAD PENDING)
040.362      395X S.DDLEN DS      2      CODE LENGTH IN BYTES
040.364      396X S.DDGRP DS      1      GROUP NUMBER FOR DRIVER
040.365      397X      DS      1      HOLD PLACE
398X *S.DDSEC      DS      2      SECTOR NUMBER FOR DRIVER (* OBSOLETE *)
040.366      399X S.DDDTA DS      2      DEVICE'S ADDRESS IN DEVLST +DEV.RES
040.370      400X S.DDOPC DS      1      OPEN OPCODE PENDING
401X
402X **      OVERLAY MANAGEMENT FLAGS
403X
000.001      404X OVL.IN   EQU      00000001B      IN MEMORY
000.002      405X OVL.RES EQU      00000010B      PERMINANTLY RESIDENT
000.014      406X OVL.NUM EQU      00001100B      OVERLAY NUMBER MASK
000.200      407X OVL.UCS EQU      10000000B      USER CODE SWAPPED FOR OVERLAY
408X
040.371      409X S.OVLFL DS      1      OVERLAY FLAG
040.372      410X S.UCSF DS      2      FWA SWAPPED USER CODE
040.374      411X S.UCSL DS      2      LENGTH SWAPPED USER CODE
040.376      412X S.OVLS DS      2      SIZE OF OVERLAY CODE
041.000      413X S.OVLE DS      2      ENTRY POINT OF OVERLAY CODE
414X
041.002      415X S.SSN  DS      2      SWAP AREA SECTOR NUMBER
041.004      416X S.DSN  DS      2      OVERLAY SECTOR NUMBER
417X
418X *      SYSCALL PROCESSING WORK AREAS
419X
041.006      420X S.CACC DS      1      (ACC) UPON SYSCALL
041.007      421X S.CODE DS      1      SYSCALL INDEX IN PROGRESS
422X
423X *      JUMPS TO ROUTINES IN RESIDENT HDOS CODE

```


041.114 477X AIO.CHA DS 2 ADDRESS OF CHANNEL BLOCK (IOC.DDA)

041.120 479X S.SCR DS 2 SYSTEM SCRATCH AREA ADDRESS
041.122 480 XTEXT MTRDEF

482X ** HDOS MONITOR PRIVATE RAM AREA DEFINITIONS.

483X
000.000 484X ORG 0
000.000 485X M.SYSM DS 1 SYSCALL ITERATION COUNT
000.001 486X M.SALO DS 1 STAND-ALONE FLAG
000.002 487X M.CSLC DS 1 LINES IN CONSOLE BUFFER
000.003 488X M.CPRE DS 1 CONSOLE PREVIOUS CHARACTER
000.004 489X M.CRUB DS 1 CONSOLE RUBOUT FLAG
000.005 490X M.CINT DS 1 CONSOLE INTERRUPT FLAG
000.006 491X M.CIN DS 2 CONSOLE CB IN POINTER
000.010 492X M.COUT DS 2 CONSOLE CB OUT POINTER
000.012 493X M.CFWA DS 2 CONSOLE CB FWA POINTER
000.014 494X M.CLWA DS 2 CONSOLE CB LWA POINTER
000.016 495X M.CDLY DS 1 CONSOLE PAD CHARACTER COUNT
000.017 496X M.CDCA DS 2 ADDRESS OF CHARACTER BEING PADDED
000.021 497 XTEXT BDEF

499X ** DEVICE DRIVER COMMUNICATION FLAGS.

500X *
501X
000.000 502X ORG 0
503X
000.000 504X DC.REA DS 1 READ
000.001 505X DC.WRI DS 1 WRITE
000.002 506X DC.RER DS 1 READ REGARDLESS
000.003 507X DC.OPR DS 1 OPEN FOR READ
000.004 508X DC.OPW DS 1 OPEN FOR WRITE
000.005 509X DC.OPU DS 1 OPEN FOR UPDATE
000.006 510X DC.CLO DS 1 CLOSE
000.007 511X DC.ABT DS 1 ABORT
000.010 512X DC.MOU DS 1 MOUNT DEVICE
000.011 513X DC.LOD DS 1 LOAD DEVICE DRIVER
000.012 514X DC.MAX DS 1 MAXIMUM ENTRY INDEX
000.013 515 XTEXT FLTDEF

FLTDEF

517X ** FLTDEF - DEFAULT SECTOR DEFINITIONS

518X				
519X	ORG	0		
000.000	520X FLT.CTY	DS	1	CONSOLE TYPE FLAGS (FOR S.CONTY)
000.001	521X FLT.CWI	DS	1	CONSOLE WIDTH (FOR S.CONWI)
000.002	522X FLT.CFC	DS	1	CONSOLE FILL CHARACTERS NEEDED
000.003	523X FLT.CRF	DS	1	CONSOLE CHARACTER REQUIRING FILL (3770 IF NONE)
000.004	524X FLT.MNC	DS	1	MAXIMUM NUMBER OF I/O CHANNELS
000.005	525X FLT.TDT	DS	1	TRACK SEEK DELAY TIME (MS/2)
000.006	526X FLT.CDB	DS	1	CONSOLE DEFINITION BYTE
000.007	527X FLT.CRD	DS	2	CONSOLE BAUD RATE
000.011	528X FLT.BOP	DS	1	BOOTUP FLAGS
000.012	529X FLT.SAL	DS	1	STAND-ALONE FLAG (1=0 => CAN GO STAND-ALONE)
000.013	530	XTEXT	FILDEF	

532X ** FILDEF - FILE TYPE DEFINITIONS.

533X	*			
534X	*	DB	3770,FT.XXX	
535X				
536X				
000.000	537X FT.ABS	EQU	0	ABSOLUTE BINARY
000.001	538X FT.PIC	EQU	1	POSITION INDEPENDANT CODE
000.002	539X FT.REL	EQU	2	RELOCATABLE CODE
000.003	540X FT.BAC	EQU	3	COMPILED BASIC CODE
000.013	541	XTEXT	ABSDEF	

543X ** ABS FORMAT EQUIVALENCES.

544X				
545X	ORG	0		
546X				
000.000	547X ABS.ID	DS	1	3770 = BINARY FILE FLAG
000.001	548X	DS	1	FILE TYPE (FT.ABS)
000.002	549X ABS.LDA	DS	2	LOAD ADDRESS
000.004	550X ABS.LEN	DS	2	LENGTH OF ENTIRE RECORD
000.006	551X ABS.ENT	DS	2	ENTRY POINT
552X				
000.010	553X ABS.COD	DS	0	CODE STARTS HERE
000.010	554	XTEXT	PICDEF	

556X ** PIC FORMAT EQUIVALENCES.

557X				
558X	ORG	0		
559X				
000.000	560X PIC.ID	DS	1	3770 = BINARY FILE FLAG
000.001	561X	DS	1	FILE TYPE (FT.PIC)
000.002	562X PIC.LEN	DS	2	LENGTH OF ENTIRE RECORD
000.004	563X PIC.PTR	DS	2	INDEX OF START OF PIC TABLE
564X				
000.006	565X PIC.COD	DS	0	CODE STARTS HERE

000.006 566 XTEXT DVDDEF

568X ** DEVICE DRIVER EQUIVALENCES.

569X
000.307 570X DVDFLV EQU 3070 DEVICE DRIVER FLAG VALUE
571X
000.006 572X ORG PIC.COD STARTS AT PIC CODE AREA
573X
000.006 574X DVD.DVD DS 1 MUST BE DVDFLV, FLAGS TO HDOS AS DRIVER
000.007 575X DVD.CAP DS 1 DEVICE CAPABILITY FLAG
000.010 576X DVD.MUM DS 1 MOUNTED UNIT MASK
000.011 577X DVD.MNU DS 1 MAXIMUM NUMBER OF UNITS
000.012 578X DVD.UFL DS 8 UNIT SUB-CAPABILITY FLAGS FOR UNITS 0-7
000.022 579X DVD.SET DS 1 = DVDFLV IFF DRIVER WILL TAKE SET OPTIONS
000.023 580X DS 24 RESERVED, MUST BE 0
000.053 581X DVD.STE EQU * ENTRY FOR 'SET' INVOCATION
582X
002.000 583X DVD.ENT EQU 2000A DRIVER ENTRY POINT (MUST BE MULT OF 256)
584
585
042.170 586 ORG USERFWA-ABS.COD
042.170 377 000 587 DF 377Q,FT.ABS
042.172 200 042 588 DW USERFWA LOAD ADDR
042.174 232 011 589 DW MEML-USERFWA LOAD LENGTH
042.176 333 042 590 DW ENTRY ENTRY ADDRESS

593 *** FIXED ADDRESS UTILITY ROUTINES.
 594 *
 595 * THESE VECTORS RESIDE AT A FIXED ADDRESS RELATIVE TO USERPWA
 596 * AND ARE CALLED BY DEVICE DRIVER SET CODE.
 597 *
 598 *
 000.000 599 ERRNZ STACK-*
 042.200 000 600 DB 0 STACK+1 MUST BE 0
 601 *
 602 *

604 *** SNA. - SCAN TO NEXT ARGUMENT.
 605 *
 606 * SNA IS CALLED TO SKIP BLANKS UNTIL THE NEXT ARGUMENT.
 607 *
 608 * ENTRY (BC) = LINE POINTER
 609 * EXIT (BC) UPDATED
 610 * 'Z' SET IF AT END OF LINE
 611 * USES A,F,B,C
 612 *
 613 *
 042.201 303 153 052 614 SNA. JMP SKB SCAN NEXT ARGUMENT

616 *** DCS. - DELIMIT CHARACTER STRING.
 617 *
 618 * DCS SCANS A CHARACTER STRING (ANYTHING BUT BLANKS) IN
 619 * THE LINE.
 620 *
 621 * ENTRY (BC) = LINE POINTER
 622 * EXIT (BC) UPDATED PAST STRING
 623 * (DE) = ADDR FIRST STRING CHARACTER
 624 * (HL) = ADDR LAST STRING CHARACTER
 625 * (A) = STRING LENGTH
 626 * 'Z' SET IF STRING EMPTY
 627 * USES ALL
 628 *
 629 *
 042.204 303 327 051 630 DCS. JMP DCS

632 *** CNA. - CONVERT NUMERIC ARGUMENT.
 633 *
 634 * CNA. CONVERTS A NUMERIC ARGUMENT IN THE COMMAND LINE
 635 * TO A BINARY VALUE.
 636 *
 637 * THE NUMBER MAY USE A 'B', 'O', 'Q', OR 'D' POSTRADIX.
 638 *
 639 * ENTRY (BC) = LINE POINTER

640 * (A) = DEFAULT RADIX
641 * EXIT (BC) UPDATED
642 * (HL) = VALUE
643 * 'C' CLEAR IF OK
644 * 'C' SET IF ERROR
645 * (A) = ERROR CODE
646 * USES ALL
647
648
042,207 303 357 051 649 CNA. JMP DNF DECODE NUMERIC FIELD

651 *** FST. - FIND IN SERIAL TABLE
652 *
653
042,212 303 163 052 654 FST. JMP \$FST.

656 *** TBLS. - TABLE SEARCH
657 *
658
042,215 303 126 053 659 TBLS. JMP \$TBLS.

661 *** WTBLs. - WORD TABLE SEARCH
662 *
663
042,220 303 157 053 664 WTBLs. JMP \$WTBLs.

666 *** LBD. - LOOK UP BAUD RATE
667 *
668
042,223 303 266 052 669 LBD. JMP \$LBD.

671 *** SOP. - SET OPTIONS
672 *
673
042,226 303 350 052 674 SOP. JMP SOP.

676 *** PBF. - PROCESS BYTE FLAG
677 *
678
042.231 303 017 053 679 PBF. JMP PBF

681 *** PBU. - PROCESS BYTE VALUE
682 *
683
042.234 303 036 053 684 PBU. JMP PBU
042.237 685 DS 60 RESERVED


```

688 *** MAIN ROUTINE.
689 *
690
691
042.333 ENTRY EQU *
042.333 041 032 055 693 LXI H,RMEML
042.336 377 052 694 DB SYSCALL,SETTP SET TOP OF USED MEMORY
042.340 041 000 000 695 LXI H,0
042.343 071 696 DAD SP
042.344 104 697 MOV B,H
042.345 115 698 MOV C,L (BC) = COMMAND LINE
042.346 315 354 042 699 CALL SET1 PERFORM PROCESSING
042.351 303 240 043 700 JMP EXIT
701
702 * GET DEVICE CODE
703 *
704 * THIS SECTION IS CALLED FROM ABOVE. A 'RETURN' CAUSES SET TO EXIT
705
042.354 315 327 051 706 SET1 CALL DCS
042.357 312 061 043 707 JZ ERR.IDS ILLEGAL DEVICE SPEC
042.362 176 708 MOV A,M
042.363 366 200 709 ORI 2000
042.365 167 710 MOV M,A
042.366 041 015 043 711 LXI H,SETA
042.371 315 163 052 712 CALL $FST
042.374 302 332 050 713 JNZ POF NOT BUILT-IN; PATCH DISK FILE AND RETURN
042.377 176 714 MOV A,M
043.000 315 061 031 715 CALL $TJMP
043.003 244 043 716 DW SETSY SY:
043.005 065 044 717 DW SETTT TT:
043.007 131 047 718 DW SETHLP HELP
043.011 056 050 719 DW SETVER VERSION SWITCH
043.013 105 050 720 DW SETHOS HDOS PARAMETERS
721
043.015 060 043 722 SETA DW SETAE END ADDRESS
043.017 001 723 DB 1 1 BYTE DATA VALUES
724
043.020 123 131 272 725 DB 'SY','+2000,0
043.024 123 131 040 726 DB 'SY0','+2000,0
043.031 124 124 272 727 DB 'TT','+2000,1
043.035 124 124 060 728 DB 'TT0','+2000,1
043.042 110 105 114 729 DB 'HEL','+2000,2
043.047 126 105 322 730 DB 'VE','+2000,3
043.053 110 104 117 731 DB 'HDO','+2000,4
732
043.060 000 733 SETAE DB 0 END OF TABLE
  
```

ERROR ABORTS

ERROR

14:50:12 16-MAY-80

```

737 *** WHEN VARIOUS ERROR CONDITIONS ARE DETECTED, CODE MAY DO A
738 * 'BAIL OUT' JUMP TO ONE OF THE ERROR ABORT ADDRESSES.
739 *
740 * THE STACK MAY BE UNCLEAN, FILES BE OPEN, ETC.
741 * THE ABORT ROUTINES WILL CLEAN THE STACK, ISSUE ANY APPROPRIATE MESSAGES,
742 * AND EXIT, LEAVING THE O/S TO CLEAN UP ANY OPEN FILES.

744 ** ERR.IDS - ILLEGAL DEVICE SPECIFICATION
745
043.061 315 136 031 746 ERR.IDS CALL $TYPTX
043.064 007 012 111 747 DB BELL,NL,'Illegal Device Specification',ENL
043.123 303 240 043 748 JMP EXIT

750 ** ERR.ILO - ILLEGAL OPTION
751
043.126 315 136 031 753 ERR.ILO CALL $TYPTX
043.131 007 012 111 754 DB BELL,NL,'Illegal Option',ENL
043.152 303 240 043 755 JMP EXIT

757 ** ERR.IOV - ILLEGAL OPTION VALUE
758
043.155 315 136 031 760 ERR.IOV CALL $TYPTX
043.160 007 012 111 761 DB BELL,NL,'Illegal Option Value',ENL
043.207 303 240 043 762 JMP EXIT

764 ** ERROR - HDOS RETURNED ERROR
765
043.212 365 767 ERROR PUSH PSW
043.213 315 136 031 768 CALL $TYPTX
043.216 007 012 105 769 DB BELL,NL,'Error - ',+2000
043.230 361 770 POP PSW
043.231 046 212 771 MVI H,ENL
043.233 377 057 772 DB SYSCALL, .ERROR
043.235 303 240 043 773 JMP EXIT

775 ** EXIT - EXIT TO HDOS.
776 *
777 * EXIT EXITS TO HDOS WITH THE 'ABORT' FLAG SET.
778 * THIS FORCES THE CONSOLE TO BE RECONFIGURED, IN CASE THE SET
779 * COMMAND CAUSED CONSOLE HANDLING CHANGES.
780
043.240 076 001 781 EXIT MVI A,1 FORCE RESET
043.242 377 000 782 SCALL ,EXIT

```


SYHLP

```
833 ** SYHLP - SY HELP OPTION
834 *
835
043.342 315 136 031 836 SYHLP CALL $TYPTX
043.345 012 012 123 837 DB NL,NL,Set Options for SY:,NL,NL
043.374 123 124 105 838 DB 'STEP nnn Set Track Step Time',NL
044.033 110 105 114 839 DB 'HELP' Type this text',NL
044.061 012 212 840 DB NL,ENL
044.063 257 841 XRA A CLEAR CARRY
044.064 311 842 RET
```

SETTT - SET TT: OPTIONS

14:50:14 16-MAY-80

```

845 **      SETTT - SET TT: OPTIONS
846 *
847
848
044.065 021 256 044 849 SETTT LXI D,TTOPRC
044.070 041 102 044 850 LXI H,TTOTAB
044.073 315 350 052 851 CALL SOP
044.076 332 212 043 852 JC ERROR
044.101 311 853 RET
    
```

```

044.102 255 044 855 TTOTAB DW TTOTAB LWA
044.104 003 856 DB 3 VALUE BYTES
857
044.105 102 113 323 858 DB 'BK',S'+200Q,CTYI,377Q-CTP,BKS,CTP,BKS
044.113 115 114 311 859 DB 'ML',I'+200Q,CTYI,377Q-CTP,MLI,CTP,MLI
044.121 115 114 317 860 DB 'ML',O'+200Q,CTYI,377Q-CTP,MLO,CTP,MLO
044.127 102 113 315 861 DB 'BK',M'+200Q,CTYI,377Q-CTP,BKM,CTP,BKM
044.135 124 101 302 862 DB 'TA',B'+200Q,CTYI,377Q-CTP,YAB,CTP,YAB
044.143 061 123 302 863 DB '1S',B'+200Q,CTYI,377Q-CTP,2SB,0
044.151 062 123 302 864 DB '2S',B'+200Q,CTYI,377Q-CTP,2SB,CTP,2SB
044.157 116 117 102 865 DB 'NOBK',S'+200Q,CTYI,377Q-CTP,BKS,0
044.167 116 117 115 866 DB 'NOML',I'+200Q,CTYI,377Q-CTP,MLI,0
044.177 116 117 115 867 DB 'NOML',O'+200Q,CTYI,377Q-CTP,MLO,0
044.207 116 117 102 868 DB 'NOBK',M'+200Q,CTYI,377Q-CTP,BKM,0
044.217 116 117 124 869 DB 'NOTA',B'+200Q,CTYI,377Q-CTP,TAB,0
044.227 127 111 104 870 DB 'WIDT',H'+200Q,WIDI,0,0
044.237 106 111 114 871 DB 'FIL',L'+200Q,FILI,0,0
044.246 110 105 114 872 DB 'HEL',F'+200Q,TTHLPI,0,0
873
044.255 000 874 TTOTAB DB 0 END OF TABLE
    
```

```

044.256 876 TTOPRC DS 0
877
000.000 878 CTYI EQU *-TTOPRC/2
044.256 266 044 879 DW STICTY
880
000.001 881 WIDI EQU *-TTOPRC/2
044.260 306 044 882 DW STTWID
883
000.002 884 FILI EQU *-TTOPRC/2
044.262 344 044 885 DW STTFIL
886
000.003 887 TTHLPI EQU *-TTOPRC/2
044.264 044 045 888 DW TTHLP
    
```

SETT - SET TT: OPTIONS

STCTY

14:50:16 16-MAY-80

```

890 ** STCTY = CONSOLE TYPE FLAGS.
891 *
892
893
044.266      894 STCTY EQU *
044.266 126 895 MOV B,M
044.267 043 896 INX H
044.270 136 897 MOV E,M (E) = VALUE
044.271 072 327 040 898 LDA S,CONTY
044.274 242 899 ANA D
044.275 263 900 ORA E
044.276 062 327 040 901 STA S,CONTY
044.301 076 000 902 MVI A,FLT,CTY
044.303 303 000 052 903 JMP SHO SET HOS OPTIONS, AND EXIT

```

905 ** STTWID - SET WIDTH OPTION

906 *

907 * SET TT: WIDTH NN

908

909

```

044.306 076 012 910 STTWID MVI A,10 (A) = DEFAULT BASE
044.310 315 357 051 911 CALL DNF DECODE NUMERIC FIELD
044.313 332 155 043 912 JC ERR,IOV
044.316 174 913 MOV A,H
044.317 247 914 ANA A
044.320 302 155 043 915 JNZ ERR,IOV ILLEGAL OPTION VALUE
044.323 175 916 MOV A,L
044.324 376 024 917 CPI 20
044.326 332 155 043 918 JC ERR,IOV TOO NARROW
044.331 062 331 040 919 STA S,CONWI
044.334 137 920 MOV E,A
044.335 026 000 921 MVI B,0
044.337 076 001 922 MVI A,FLT,CWI
044.341 303 000 052 923 JMP SHO SET HOS OPTION AND EXIT

```

925 ** STTFIL - FILL OPTION.

926 *

927 * SET TT: FILL CC NN

928

929

```

044.344 076 012 930 STTFIL MVI A,10 (A) = DEFAULT BASE
044.346 315 357 051 931 CALL DNF DECODE NUMERIC FIELD
044.351 332 155 043 932 JC ERR,IOV
044.354 174 933 MOV A,H
044.355 264 934 ORA H
044.356 302 155 043 935 JNZ ERR,IOV
044.361 265 936 ORA L
044.362 312 155 043 937 JZ ERR,IOV ILLEGAL OPTION VALUE
044.365 365 938 PUSH PSW SAVE CHARACTER CODE
044.366 076 012 939 MVI A,10 (A) = DEFAULT BASE
044.370 315 357 051 940 CALL DNF DECODE NUMERIC FIELD
044.373 332 155 043 941 JC ERR,IOV

```

```

044.376 174          942      MOV      A,H
044.377 267          943      ORA      A
045.000 302 155 043  944      JNZ      ERR.IOV      ILLEGAL VALUE
045.003 105          945      MOV      B,L      (B) = DELAY COUNT
                                946
                                947 *      SET IN MEMORY
                                948
045.004 052 346 040  949      LHLD    S,DLINK
045.007 021 016 000  950      LXI    D,M.CDLY
045.012 031          951      DAD    D
045.013 160          952      MOV    M,B      SET DELAY COUNT
000.000          953      ERRNZ  M,CDCA-M.CDLY-1
045.014 043          954      INX    H
045.015 315 211 030  955      CALL  $HLIHL      (HL) = ADDRESS FOR CHAR
045.020 130          956      MOV    E,B      (E) = DELAY
045.021 361          957      POP   PSW      (A) = DELAY CHARACTER
045.022 167          958      MOV    M,A
045.023 365          959      PUSH  PSW
045.024 026 000      960      MVI    D,0
045.026 076 002      961      MVI    A,FLT.CFC
045.030 315 000 052  962      CALL  SHD      SET FILL COUNT
045.033 361          963      POP   PSW
045.034 137          964      MOV    E,A
045.035 026 000      965      MVI    D,0
045.037 076 003      966      MVI    A,FLT.CRF
045.041 303 000 052  967      JMP    SHD      SET CHAR REQUIRING FILL AND EXIT
  
```

970 ** TTHLP - HELP OPTION FOR DEVICE TT:

971 *
 972

```

045.044 315 136 031  973 TTHLP CALL $TYPTX
045.047 012 012 123  974 DB     NL,NL,'SET Options for TT:',NL,NL
045.076 102 113 123  975 DB     'BKS      CRT terminal allows backspace characters',NL
045.154 115 114 111  976 DB     'MLI      Map lower case input to upper case',NL
045.224 115 114 117  977 DB     'MLO      Map lower case output to upper case',NL
045.275 102 113 115  978 DB     'BKM      Treat 'BKSP' codes (on input) as 'DELETE' ('RUBOUT')',NL
045.367 124 101 102  979 DB     'TAB      Terminal can process tab codes',NL
046.033 012          980 DB     NL
046.034 011 124 150  981 DB     /      The above options can be preceded by 'NO' to negate their',NL
046.127 011 105 146  982 DB     /      Effect. (I.E. SET TT: NOTAB )',NL
046.167 012          983 DB     NL
046.170 061 123 102  984 DB     '1SB     Use One Stop Bit for Console Terminal',NL
046.243 062 123 102  985 DB     '2SB     Use Two Stop Bits for Console Terminal',NL
046.317 127 111 104  986 DB     'WIDTH NN   Set console width to NN characters',NL
046.373 106 111 114  987 DB     'FILL CC NN  Pad occurrences of character CC with NN null',NL
047.063 011 011 143  988 DB     /      characters',NL
047.100 110 105 114  989 DB     'HELP      Type this text',NL
047.125 012 212      990 DB     NL,ENL
047.127 257          991 XRA    A      CLEAR CARRY
047.130 311          992 RET
  
```

SETHLP

```
995 *** SETHLP - PRINT HELP TEXT.  
996 *  
997  
998  
047.131 315 136 031 999 SETHLP CALL $TYPYX  
047.134 012 012 107 1000 DB NL,NL,'General Command Format:',NL  
047.168 012 1001 DB NL  
047.167 040 040 040 1002 DB / SET xx: opt',NL  
047.206 040 040 040 1003 DB / xx: -- Device Name',NL  
047.237 040 040 040 1004 DB / opt -- Desired Option',NL  
047.273 012 012 1005 DB NL,NL  
047.275 106 157 162 1006 DB /For HELP with a specific device, type:',NL  
047.344 012 1007 DB NL  
047.345 040 040 040 1008 DB / SET xx: HELP',NL  
047.365 012 012 1009 DB NL,NL  
047.367 124 157 040 1010 DB /To determine the version of SET, type:',NL  
050.036 012 1011 DB NL  
050.037 040 040 040 1012 DB / SET Ver',NL  
050.052 012 1013 DB NL  
1014  
000.001 1015 IF PUBLIC  
1016 DB NL  
1017 DB /To list settable *HDOS* options, type:',NL  
1018 DB NL  
1019 DB / SET HDOS HELP',NL  
1020 DB NL  
1021 ENDF  
1022  
050.053 212 1023 DB ENL  
050.054 257 1024 XRA A CLEAR CARRY  
050.055 311 1025 RET
```


SETVER - SET VERSION

SETVER

14:50:18 16-MAY-80

```
1029 *** SETVER - SET VERSION
1030 *
1031 * SETVER PRINTS THE VERSION OF THIS SET PROGRAM
1032 *
1033
050.056 315 136 031 1034 SETVER CALL $TYPTX
050.061 123 105 124 1035 DB 'SET',TAB,'Version!'
050.077 061 056 066 1036 DB VERS/16+'0',',',VERS&00001111B+'0'
050.102 212 1037 DB ENL
050.103 257 1038 XRA A CLEAR CARRY
050.104 311 1039 RET
```

SETHOS - SET HDOS PARAMETERS

14:50:18 16-MAY-80

```

1042 **      SETHOS - SET HDOS PARAMETERS
1043 *
1044 *      SETHOS PERMITS THE SETTING OF *HDOS* PARAMETERS.
1045 *
1046
050.105 021 162 050 1047 SETHOS LXI    D,HOSFRC
050.110 041 122 050 1048          LXI    H,HOSTAB
050.113 315 350 052 1049          CALL   SOP
050.116 332 212 043 1050          JC     ERROR
050.121 311          1051          RET

050.122 161 050      1053 HOSTAB DW    HOSTAB      END ADDRESS
050.124 002          1054          DB      2          ONE DATA BYTE
1055
050.125 123 124 101 1056          DB      'STAND-ALON',E'+2000,HOSSALI,1
050.142 116 117 123 1057          DB      'HOSTAND-ALON',E'+2000,HOSSALI,0
000.001          1058          IF      PUBLIC
1059          DB      'HEL',F'+2000,HELPI,0
1060          ENDIF
1061
050.161 000          1062 HOSTAB DB      0          END OF TABLE

050.162          1064 HOSPRC DS      0
050.162          1065          SET      *
1066
000.000          1067 HOSSALI EQU   *-,/2
050.162 164 050      1068          DW      HOSSAL
1069
000.001          1070          IF      PUBLIC
1071          HELPI EQU   *-,/2
1072          DW      HOSSHLP
1073          ENDIF

1075 **      HOSSAL - SET *HDOS* STAND ALONE FLAG
1076 *
1077 *      HOSSAL SETS THE *HDOS* STAND-ALONE FLAG ENABLING *HDOS*
1078 *      TO GO STAND ALONE UPON EXITS TO *SYSCMD.SYS* WHEN THE SYSTEM
1079 *      DISK HAS BEEN DISMOUNTED.
1080 *
1081 *
1082
050.164 176          1083 HOSSAL MOV    A,M
050.165 021 001 000 1084          LXI    D,M,SALO
050.170 052 346 040 1085          LHL   S,DLINK
050.173 031          1086          DAD    D
050.174 167          1087          MOV    M,A      SET *SALONE* TO 1, ( ?= 0 => SET ).
050.175 137          1088          MOV    E,A      (E) = NEW VALUE

```

```

050.176 026 000 1089 MVI D,0 (D) = CHANGE MASK
050.200 076 012 1090 MVI A,FLT,SAL INDEX OF BYTE TO CHANGE
000.001 1091 IF PUBLIC
1092 ELSE
050.202 365 1093 PUSH PSW
050.203 173 1094 MOV A,E
050.204 247 1095 ANA A
050.205 312 326 050 1096 JZ HOS1
050.210 315 136 031 1097 CALL $TYPTX
050.213 012 111 164 1098 DB NL,'It is Now Pitch Dark. If You Proceed, You Will Likely '
050.303 106 141 154 1099 DB 'Fall Into a Pit.',NL,BELL,ENL
050.326 361 1100 HOS1 POP PSW
1101 ENDIF
050.327 303 000 052 1102 JMP SHO SET *HDOS* OPTIONS, AND EXIT
000.001 1103 IF PUBLIC
1104 HOSHLP SPACE 4,10
1105 ** HOSHLP - PROCESS HELP OPTION FOR PSEUDO-DEVICE *HDOS*
1106 *
1107 *
1108
1109 HOSHLP CALL $TYPTX
1110 DB NL,NL,'Set Options for HDOS',NL,NL
1111 DB 'STAND-ALONE Flag Stand-Alone Operation Legal',NL
1112 DB 'HELP Print this Text',NL
1113 DB NL,ENL
1114 XRA A CLEAR CARRY
1115 RET
1116 ENDIF
    
```

```

1120 ** PDF - PATCH DISK FILE.
1121 *
1122 * PDF IS CALLED TO SET (PATCH) A DEVICE DRIVER FILE ON THE DISK.
1123 *
1124 * 1) THE DRIVER IS FOUND, AND READ IN
1125 * 2) ITS FORMAT IS CHECKED
1126 * 3) THE 'SET PREAMBLE' IS RELOCATED
1127 * 4) ITS 'SET' CODE AREA IS ENTERED
1128 * 5) THE DRIVER, EXCEPT FOR THE SET PREAMBLE (BECAUSE IT WAS RELOCATED)
1129 * IS WRITTEN BACK OUT
1130 *
1131 * ENTRY (DE) = ADDRESS OF 'dev:' SPECIFICATION
1132 * EXIT TO CALLER OF OK
1133 * TO 'ERR.???' IF ERROR
1134 * USES ALL
1135 *
1136 *
050.332 1137 PDF EQU *
050.332 032 1138 LDAX D (A) = FIRST CHAR OF DEVNAME
050.333 247 1139 ANA A
050.334 372 061 043 1140 JM ERR.IDS ILLEGAL DEV SPECIFIED
050.337 062 163 051 1141 STA PDFB
050.342 023 1142 INX D
050.343 032 1143 LDAX D (A) = 2ND CHAR OF DEVNAME
050.344 247 1144 ANA A
050.345 372 061 043 1145 JM ERR.IDS
050.350 062 164 051 1146 STA PDFB+1
050.353 023 1147 INX D
050.354 032 1148 LDAX D (A) = UNIT NUMBER OR '?'
050.355 356 272 1149 XRI ':+2000 ASSUME ':'
050.357 312 376 050 1150 JZ PDF1 IS ' (A) = UNIT NUMBER = 0
050.362 032 1151 LDAX D
050.363 326 060 1152 SUI '0' DECODE UNIT NUMBER
050.365 332 061 043 1153 JC ERR.IDS NOT A UNIT NUMBER
050.370 376 010 1154 CPI 7+1
050.372 322 061 043 1155 JNC ERR.IDS NOT A UNIT NUMBER
050.375 023 1156 INX D POINT TO ':'
1157 *
1158 * (A) = UNIT NUMBER (IN BINARY)
1159 * (DE) = ADDRESS OF ':'
1160 *
050.376 365 1161 PDF1 PUSH PSW SAVE UNIT NUMBER
050.377 032 1162 LDAX D
051.000 023 1163 INX D
051.001 376 272 1164 CPI ':+2000
051.003 302 061 043 1165 JNE ERR.IDS ILLEGAL DEVICE SPECIFICATION
1166 *
1167 * HAVE DEVICE NAME CRACKED OUT, FIND IN DIRECTORY
1168 *
051.004 325 1169 PUSH D SAVE POINTER TO COMMAND LINE
051.007 041 157 051 1170 LXI H,PDFA
000.000 1171 ERRNZ CN,DVD USING CHANNEL 0
051.012 257 1172 XRA A
051.013 377 044 1173 SCALL .OPENU OPEN DEVICE DRIVER FILE
051.015 332 212 043 1174 JC ERROR REPORT ERROR
1175 *

```

```

1176 * READ IN FIRST BLOCK, SEE IF DEVICE DRIVER, AND HOW LONG
1177
051.020 001 000 001 1178 LXI B,256
051.023 021 032 054 1179 LXI D,BUFF
000.000 1180 ERRNZ CN,DVD USING CHANNEL 0
051.026 257 1181 XRA A
051.027 377 004 1182 SCALL .READ
051.031 332 212 043 1183 JC ERROR
1184
051.034 072 040 054 1185 LDA BUFF+DVD,DVD
051.037 376 307 1186 CPI DVDFLV
051.041 312 122 051 1187 JE PDF3 OK
051.044 315 136 031 1188 CALL $TYPTX
051.047 007 012 104 1189 DB BELL,NL,'Disk File DOES Not Have Proper Format',ENL
051.117 303 240 043 1190 JMP EXIT
1191
1192 * IS DEVICE DRIVER, SEE IF SETUP FOR 'SET' USE
1193
051.122 072 054 054 1194 PDF3 LDA BUFF+DVD,SET
051.125 376 307 1195 CPI DVDFLV
051.127 302 126 043 1196 JNE ERR,ILO ANY OPTION IS ILLEGAL, SINCE NOT SETUP
1197
1198 * LOAD AND RELOCATE DRIVER
1199
051.132 315 172 051 1200 CALL LDD LOAD DEVICE DRIVER
1201
1202 * CALL DRIVER PREAMBLE CODE
1203
051.135 321 1204 POP D (DE) = PARAMETER LIST ADDRESS
051.136 361 1205 POP PSM (A) = UNIT NUMBER
051.137 315 105 054 1206 CALL BUFF+DVD,STE ENTER AT SET POINT
051.142 332 212 043 1207 JC ERRDR ILLEGAL OPTION
1208
1209 * HAVE SUCCESSFULLY SET, WRITE DRIVER BACK
1210
051.145 315 272 051 1211 CALL WDD WRITE DEVICE DRIVER
000.000 1212 ERRNZ CN,DVD
051.150 257 1213 XRA A (A) = CN,DVD
051.151 377 046 1214 SCALL .CLOSE CLOSE FILE
051.153 332 212 043 1215 JC ERROR
051.156 311 1216 RET ALL DONE
1217
1218
051.157 123 131 060 1219 PDFA DB 'SYO:/' DEVICE DRIVER FILE NAME
051.163 1220 PDFB DS 2 CODE STORES NAME HERE
051.165 056 104 126 1221 DB ',DVD',0
  
```

```

1223 **      LDD - LOAD DEVICE DRIVER.
1224 *
1225 *      LDD LOADS THE REST OF A DEVICE DRIVER INTO MEMORY, AND DOES A
1226 *      PARTIAL RELOCATION.
1227 *
1228 *      LDD IS ENTERED WITH THE FIRST 256 BYTES OF THE DRIVER IN
1229 *      'BUFF'. THE DEVICE DRIVER IS OPEN ON CHANNEL 'CN:DVD', AND
1230 *      POSITIONED JUST AFTER THE FIRST SECTOR.
1231 *
1232 *      LDD COMPUTES THE LENGTH OF THE DRIVER, READS IT ALL IN (INCLUDING
1233 *      RELOCATION TABLES), AND THEN RELOCATES ANY CODE IN THE SET PREAMBLE.
1234 *
1235 *      ENTRY   NONE
1236 *      EXIT    TO CALLER IF OK
1237 *            TO AN ERROR ABORT ADDRESS IF PROBLEMS
1238 *      USES    ALL
1239 *
1240
051.172 052 034 054 1241 LDD   LHL D, BUFF+PIC.LEN
051.175 001 377 000 1242     LXI   B, 255
051.200 011                1243     DAD   B                (HL) = ROUND UP
051.201 104                1244     MOV   R, H
051.202 016 000          1245     MVI   C, 0          (BC) = SECTOR COUNT
1246
051.204 041 032 054 1247     LXI   H, BUFF
051.207 011                1248     DAD   B                (HL) = NEW LWA FOR SET AND BUFFERS
051.210 305                1249     PUSH  B                SAVE (BC)
051.211 377 052          1250     SCALL ,SETTP
051.213 332 212 043 1251     JC    ERROR          NO ROOM
051.216 301                1252     POP   B
1253
051.217 021 032 055 1254     LXI   D, BUFF+256
000.000                1255     ERRNZ CN, DVD
051.222 005                1256     DCR   B                COUNT - 1 FOR SECTOR ALREADY READ
051.223 257                1257     XRA   A                (A) = CHANNEL NUMBER
051.224 377 004          1258     SCALL ,READ          READ IN REST OF DRIVER
051.226 332 212 043 1259     JC    ERROR          PROBLEMS
1260
1261 *      RELOCATE ALL REFERENCES WITHIN THE FIRST 512 BYTES
1262
051.231 052 036 054 1263     LHL D, BUFF+PIC.PTR (HL) = ADDRESS OF RELOCATION TABLE
051.234 001 032 054 1264     LXI   B, BUFF        (BC) = RELOCATION CONSTANT
051.237 011                1265     DAD   B                (HL) = ABS. ADDRESS OF REL. TABLE
051.240 136                1266     LDD1  MOV   E, M
051.241 043                1267     INX   H
051.242 126                1268     MOV   D, M          (DE) = REL ADDRESS OF WORD TO RELOCATE
051.243 043                1269     INX   H
051.244 172                1270     MOV   A, D
051.245 263                1271     ORA   E
051.246 310                1272     RZ                    DONE
1273
1274 *      (DE) = REL ADDRESS OF WORD TO RELOCATE. MAKE SURE IS ELIGIBLE
1275
000.000                1276     ERRNZ DVD, ENT-512
051.247 172                1277     MOV   A, D
051.250 376 002          1278     CPI   2

```

```

051.252 322 240 051 1279 JNC LDD1 IS IN DRIVER CODE, DONT RELOCATE
051.255 353 1280 XCHG
051.256 011 1281 DAD B
051.257 353 1282 XCHG (DE) = ABS ADDRESS OF WORD TO RELOCATE
051.260 032 1283 LDAX D
051.261 201 1284 ADD C
051.262 022 1285 STAX D
051.263 023 1286 INX D
051.264 032 1287 LDAX D
051.265 210 1288 ADC B
051.266 022 1289 STAX D
051.267 303 240 051 1290 JMP LDD1 RELOCATE NEXT REFERENCE

```

```

1292 ** WDD - WRITE DEVICE DRIVER.
1293 *
1294 * WDD IS CALLED TO WRITE THE MODIFIED PORTION OF THE DEVICE DRIVER
1295 * BACK.
1296 *
1297 * THE DEVICE DRIVER IN 'BUFF' IS WRITTEN BACK, EXCEPT FOR THE PREAMBLE,
1298 * WHICH WAS RELOCATED. THE DRIVER ITSELF WAS NOT RELOCATED, NOR WAS THE
1299 * RELOCATION TABLE ITSELF MODIFIED.
1300 *

```

```

1301 * ENTRY NONE
1302 * EXIT NONE
1303 * USES ALL

```

```

1304
1305
051.272 001 002 000 1306 WDD LXI B,DVD.ENT/256 (BC) = SECTOR NUMBER OF START OF CODE
000.000 1307 ERRNZ #DVD.ENT MUST BE MULT OF 256
000.000 1308 ERRNZ CN,DVD
051.275 257 1309 XRA A (A) = CN,DVD
051.276 377 047 1310 SCALL .POSIT POSITION FILE
051.300 332 212 043 1311 JC ERROR

```

```

1312
1313 * WRITE BACK
1314

```

```

051.303 052 034 054 1315 LHLD BUFF+PIC,LEN
051.306 001 377 376 1316 LXI B,255-DVD.ENT
051.311 011 1317 DAD B
051.312 104 1318 MOV B,H
051.313 016 000 1319 MVI C,0 (BC) = LENGTH TO WRITE, ROUNDED TO SECTOR
051.315 021 032 056 1320 LXI D,BUFF+DVD.ENT
000.000 1321 ERRNZ CN,DVD
051.320 257 1322 XRA A
051.321 377 005 1323 SCALL .WRITE WRITE DRIVER BACK
051.323 332 212 043 1324 JC ERROR
051.326 311 1325 RET RETURN

```

```

1329 **   DCS - DELIMIT CHARACTER STRING.
1330 *
1331 *   DCS ADVANCES PAST THE NEXT CHARACTER STRING, AND LOCATES
1332 *   ITS STARTING AND ENDING ADDRESSES
1333 *
1334 *   ENTRY (BC) = LINE POINTER
1335 *   EXIT (BC) ADVANCED
1336 *   (DE) = STRING FWA
1337 *   (HL) = STRING LWA
1338 *   (A) = STRING SIZE
1339 *   'Z' SET IF EMPTY
1340 *   USES ALL
1341
1342
051.327 315 153 052 1343 DCS CALL SKB SKIP BLANKS
051.332 120 1344 MOV D,B
051.333 131 1345 MOV E,C
051.334 012 1346 DCS1 LDAX B
051.335 247 1347 ANA A
051.336 312 350 051 1348 JZ DCS2 END OF LINE
051.341 003 1349 INX B
051.342 376 040 1350 CPI ' '
051.344 302 334 051 1351 JNE DCS1 NOT END OF STRING
051.347 013 1352 DCX B POINT TO BLANK
1353
051.350 140 1354 DCS2 MOV H,B
051.351 151 1355 MOV L,C SET LWA
051.352 053 1356 DCX H
051.353 175 1357 MOV A,L
051.354 223 1358 SUB E
051.355 074 1359 INR A
051.356 311 1360 RET

1362 **   DNF - DECODE NUMERIC FIELD.
1363 *
1364 *   DNF CRACKS THE NEXT FIELD AS A NUMBER.
1365 *
1366 *   ENTRY (BC) = LINE POINTER
1367 *   (A) = DEFAULT BASE
1368 *   EXIT (HL) = VALUE
1369 *   (BC) UPDATED
1370 *   'C' CLEAR IF OK
1371 *   'E' SET IF ERROR
1372 *   (A) = ERROR CODE
1373 *   USES ALL
1374
1375
051.357 365 1376 DNF PUSH PSW SAVE POSTRADIX
051.360 315 153 052 1377 CALL SKB SKIP BLANKS
051.363 361 1378 POP PSW RESTORE POSTRADIX
051.364 140 1379 MOV H,B
051.365 151 1380 MOV L,C
051.366 315 203 053 1381 CALL $DNV DECODE NUMERIC VALUE

```


SUBROUTINES

DNF

14:50:23 16-MAY-80

```

051.371 104      1382      MOV      B,H
051.372 115      1383      MOV      C,L      RESET POINTER
051.373 353      1384      XCHG
051.374 320      1385      RNC
051.375 076 037  1386      MVI      A,EC,ILV  ALL OR
051.377 311      1387      RET      ILLEGAL VALUE

1389 **      SHD - SET HDOS OPTIONS.
1390 *
1391 *      SHD IS CALLED TO SET AN OPTION FIELD IN THE HDOS.SYS FILE
1392 *      ON THE DISK.
1393 *
1394 *      THIS FILE IS FLAGGED READ ONLY, SO SHD MUST GO THROUGH THE
1395 *      SYSTEM DEVICE DRIVER TO DO ITS DIRTY WORK.
1396 *
1397 *      ENTRY (A) = INDEX OF BYTE TO CHANGE
1398 *            (D) = MASK FOR CHANGE
1399 *            (E) = NEW VALUE
1400 *      EXIT  DONE
1401 *      USES  ALL
1402
1403
052.000 306 011  1404      SHD      ADI      PIC,COD+3
052.002 325      1405      PUSH     D          SAVE VALUES
052.003 345      1406      PUSH     PSW       SAVE INDEX INTO BINARY
1407
052.004 041 127 052 1408      LXI      H,SHDA
052.007 021 144 052 1409      LXI      D,SHDB
052.012 076 000      1410      MVI      A,CN,DVD
052.014 377 042      1411      DB      SYSCALL,OPENR  OPEN FILE TO GET INFO
052.016 332 212 043 1412      JC      ERROR
1413
1414 *      GET FIRST BLOCK
1415

052.021 052 352 040 1416      LHL      S,CFWA
000.000 1417      ERNZ    IOCCTD-1    MUST SKIP CHANNEL FOR #0
052.024 315 211 030 1418      CALL    $HLIHL      (HL) = ADDRESS OF CHANNEL #0
000.000 1419      ERNZ    IOC.LNK
000.000 1420      ERNZ    CN,DVD
052.027 021 004 000 1421      LXI      D,IOC,FLG
052.032 031      1422      DAD     D          (HL) = FLAG BYTE FOR THIS CHANNEL
052.033 076 014      1423      MVI      A,FT.OW+FT.OU
052.035 266      1424      ORA     M
052.036 167      1425      MOV     M,A          KLUDGE IT TO OPEN FOR UPDATE
1426

052.037 076 000      1427      MVI      A,CN,DVD
052.041 001 000 001 1428      LXI      B,256
052.044 021 032 054 1429      LXI      D,BUFF
052.047 377 004      1430      DR      SYSCALL,READ  READ IN THE FIRST SECTOR
052.051 332 212 043 1431      JC      ERROR
1432
1433 *      MODIFY THE SPECIFIED BYTE
1434

```

```

052.054 361      1435      POP      PSW      (A) = INDEX
052.055 321      1436      POP      D        (DE) = VALUES
052.056 041 032 054 1437      LXI      H,BUFF
052.061 315 101 030 1438      CALL    $DATA, (HL) = ADDRESS OF BYTE TO CHANGE
052.064 176      1439      MOV      A,M
052.065 242      1440      ANA      D
052.066 263      1441      ORA      E
052.067 167      1442      MOV      M,A      UPDATE
1443

```

1444 * RE-WRITE THE SECTOR

```

1445
052.070 076 000      1446      MVI      A,CN,DVD
052.072 001 000 000 1447      LXI      B,0
052.075 377 047      1448      DB      SYSCALL,POSIT
052.077 332 212 043 1449      JC      ERROR
1450
052.102 076 000      1451      MVI      A,CN,DVD
052.104 001 000 001 1452      LXI      B,256
052.107 021 032 054 1453      LXI      D,BUFF
052.112 377 005      1454      DB      SYSCALL,WRITE
052.114 332 212 043 1455      JC      ERROR
1456
052.117 076 000      1457      MVI      A,CN,DVD
052.121 377 046      1458      DB      SYSCALL,CLOSE CLOSE FILE
052.123 332 212 043 1459      JC      ERROR
052.126 311      1460      RET
1461

```

```

052.127 123 131 060 1462 SHDA  DB  'SY0:HDOS.SYS',0
052.144 000 000 000 1463 SHOB  DB  0,0,0,0,0,0

```

```

1465 ** SKB - SKIP BLANKS.
1466 *
1467 * SKB SKIPS BLANKS IN THE LINE.
1468 *
1469 * ENTRY (BC) = LINE POINTER
1470 * EXIT (BC) UPDATE
1471 * 'Z' SET IFF END OF LINE
1472 * USES A,F,B,C
1473

```

```

1474
052.152 003      1475 SKB1  INX      B
052.153 012      1476 SKB   LDAX     B
052.154 376 040      1477      CPI      ' '
052.156 312 152 052 1478      JE      SKB1      IF BLANK
052.161 247      1479      ANA      A
052.162 311      1480      RET

```

```

1483
1484
052.163 1485 XTEXT FST

1487X ** *FST - FIND IN SERIAL TABLE
1488X *
1489X * *FST SEARCHES A SERIAL TABLE FOR
1490X * A SPECIFIC KEY
1491X *
1492X * ENTRY (HL) = ADDR. OF TABLE
1493X * (DE) = ADDR. OF SEARCH KEY
1494X * EXIT (DE) = UNCHANGED
1495X * 'Z' CLEARED IF NO MATCH FOUND
1496X * (HL) = ADDR. OF NEXT AVAILABLE BYTE
1497X * 'Z' SET IF MATCH FOUND
1498X * (HL) = ADDR. OF FIRST DATA BYTE
1499X * USES A,F,H,L
1500X
1501X
1502X
052.163 305 1503X *FST PUSH B SAVE REGISTERS
052.164 325 1504X PUSH D
1505X
1506X * SAVE TABLE LIMIT AND DATA BYTE COUNT
1507X
052.165 136 1508X MOV E,M GET AND SAVE TABLE LIMIT
052.166 043 1509X INX H (HL) = 2KB BYTE OF SIZE
052.167 126 1510X MOV D,M
052.170 353 1511X XCHG
052.171 042 263 052 1512X SHLD *FST,L SAVE MAX. TABLE SIZE
1513X
052.174 353 1514X XCHG
052.175 043 1515X INX H (HL) = # OF BYTES OF DATA/ENTRY
052.176 176 1516X MOV A,M
052.177 062 265 052 1517X STA *FST.C
052.202 043 1518X INX H (HL) = BEGINNING OF DATA
052.203 321 1519X FST1 POP D RESTORE ADDR. TO SEARCH KEY
052.204 325 1520X PUSH D
1521X
1522X * CHECK FOR END OF DATA
1523X
052.205 176 1524X MOV A,M
052.206 267 1525X ORA A AT END OF DATA? ((A) = 0)
052.207 302 216 052 1526X JNZ FST2 NO, START MATCHING
052.212 074 1527X INR A CLEAR 'Z'
052.213 321 1528X POP D
052.214 301 1529X POP B RESTORE REGISTERS
052.215 311 1530X RET
1531X
052.216 032 1532X FST2 LDAX D (A) = KEY CHAR.
052.217 276 1533X CMP H COMPARE TO TABLE
052.220 302 234 052 1534X JNE FST3 NO MATCH, FIND NEXT KEY
052.223 247 1535X ANA A END OF KEY?

```

```

052.224 372 256 052 1536X JM FST4 YES, SET UP FOR EXIT
052.227 043 1537X INX H
052.230 023 1538X INX D
052.231 303 216 052 1539X JMP FST2
1540X
052.234 176 1541X FST3 MOV A,M SEARCH FOR END OF KEY
052.235 247 1542X ANA A TEST CHAR.
052.236 043 1543X INX H
052.237 362 234 052 1544X JF FST3 CONTINUE SEARCH
052.242 072 265 052 1545X LDA $FST,C (A) = # OF BYTES OF DATA/ENTRY
052.245 205 1546X ADD L
052.246 157 1547X MOV L,A
052.247 076 000 1548X MVI A,0
052.251 214 1549X ADC H
052.252 147 1550X MOV H,A (HL) = HEAD OF NEXT KEY
052.253 303 203 052 1551X JMP FST1 COMPARE NEXT KEY
1552X
052.256 257 1553X FST4 XRA A SET 'Z' FOR EXIT
052.257 043 1554X INX H (HL) = FIRST BYTE OF DATA
052.260 321 1555X POP D RESTORE REGISTERS
052.261 301 1556X POP B
052.262 311 1557X RET EXIT
1558X
1559X
052.263 1560X $FST,L DS 2
052.265 1561X $FST,C DS 1
052.266 1562 XTEXT LBD

```

```

1564X *** $LBD - LOOKUP BAUDRATE DIVISOR.
1565X *
1566X * $LBD TRANSLATES A BAUD RATE INTO THE PROPER DIVISOR FOR THE
1567X * 8250 CHIPS ON THE H8-4 SERIAL CARD.
1568X *
1569X * NOTE THAT $LBD DOES NOT ACTUALLY COMPUTE THE TRANSFORMATION, BUT
1570X * SIMPLY LOOKS UP THE VALUE IN A TABLE. THIS IS DONE TO DETECT TYPOS
1571X * IN THE USER SUPPLIED BAUD RATE.
1572X *
1573X * ENTRY (DE) = BAUD RATE (AS A BINARY NUMBER)
1574X * EXIT 'Z' SET IF VALID BAUD RATE
1575X * (HL) = DIVISOR
1576X * 'Z' CLEAR IF NOT VALID BAUD RATE
1577X * USES A,F,D,E,H,L
1578X *
1579X *

```

```

052.266 172 1580X $LBD MOV A,D
052.267 263 1581X ORA E (A) = CODE VALUE
052.270 041 303 052 1582X LXI H,LBDA (HL) = LOOKUP TABLE
052.273 315 157 053 1583X CALL $WTBLS WORD TABLE LOOKUP
052.276 176 1584X MOV A,M
052.277 043 1585X INX H
052.300 146 1586X MOV H,M
052.301 157 1587X MOV L,A
052.302 311 1588X RET RETURN WITH CONDITION CODE FROM $WTBLS

```

```
1589X  
1590X  
1591X **      BAUD RATE VS 8250 DIVISOR TABLE.  
1592X *  
1593X *      KEY IS BAUD RATE SQUEEZED INTO ONE BYTE  
1594X  
052.303      1595X LDA      DS      0  
052.303 151 1596X      DB      2400/256!*2400  
052.304 040 000 1597X      DW      000060A  
052.306 245 1598X      DB      9600/256!*9600  
052.307 014 000 1599X      DW      000014A  
052.311 132 1600X      DB      600/256!*600  
052.312 300 000 1601X      DW      000300A  
052.314 113 1602X      DB      19200/256!*19200  
052.315 008 000 1603X      DW      000006A  
052.317 322 1604X      DB      4800/256!*4800  
052.320 030 000 1605X      DW      000030A  
052.322 264 1606X      DB      1200/256!*1200  
052.323 140 000 1607X      DW      000140A  
052.325 055 1608X      DB      300/256!*300  
052.326 200 001 1609X      DW      001200A  
052.330 074 1610X      DB      7200/256!*7200  
052.331 020 000 1611X      DW      000020A  
052.333 036 1612X      DB      3600/256!*3600  
052.334 040 000 1613X      DW      000040A  
052.336 017 1614X      DB      1800/256!*1800  
052.337 100 000 1615X      DW      000100A  
052.341 156 1616X      DB      110/256!*110  
052.342 027 204 1617X      DW      204027A  
052.344 226 1618X      DB      150/256!*150  
052.345 000 003 1619X      DW      003000A  
052.347 000 1620X      DB      0  
052.350      1621      XTEXT      SDF      END OF TABLE
```

```

1623X **      SOP = SET OPTIONS
1624X *
1625X *      PROCESS OPTION SET VIA OPTION TABLE AND PROCESSOR TABLE.
1626X *
1627X *      OPTION TABLE FORMAT: P(1)=PROCESSOR INDEX; P(2);...;P(N)
1628X *
1629X *      DW      <END OF TABLE>
1630X *      DB      N
1631X *      DB      <<SEARCH STRIN";'6";'2000;P;P(1);...;P(N)
1632X *
1633X *
1634X *      <EOT> DB      0      END OF TABLE
1635X *
1636X *      PROCESSOR TABLE FORMAT:
1637X *
1638X *      DW      PROC.0
1639X *      DW      PROC.1
1640X *
1641X *
1642X *      DW      PROC.N
1643X *
1644X *
1645X *
1646X *      ENTRY: (BC) = LINE POINTER
1647X *      (DE) = JUMP TABLE ADDRESS
1648X *      (HL) = OPTION TABLE ADDRESS
1649X *
1650X *      EXIT: (RET) = TO PROCESSOR IF NO ERROR
1651X *      = 'C' SET IF ERROR
1652X *      (A) = ERROR CODE
1653X *      (BC) = LINE POINTER UPDATED
1654X *      (HL) = ADDRESS OF NEXT AVAILABLE DATA BYTE
1655X *
1656X *      USES: ALL
1657X
1658X
052.350 325 1659X SDF PUSH D
052.351 345 1660X PUSH H
052.352 315 327 051 1661X CALL DCS (DE) = FWA; (HL) = LWA
052.355 312 011 053 1662X JZ SOP1
052.360 176 1663X MOV A,M
052.361 366 200 1664X ORI 2000
052.363 167 1665X MOV M,A
052.364 341 1666X POP H (HL) = OPT. TABLE ADDR.
052.365 315 163 052 1667X CALL $FST
052.370 302 012 053 1668X JNZ SOP2
052.373 353 1669X XCHG (DE) = ADDR. OF FIRST DATA BYTE
052.374 341 1670X POP H (HL) = JUMP TABLE ADDR.
052.375 032 1671X LDAX D (A) = PROCESSOR INDEX
052.376 007 1672X RLC X 2
052.377 023 1673X INX D
053.000 315 101 030 1674X CALL $DADA;
053.003 315 211 030 1675X CALL $HLIHL (HL) = PROCESSOR ADDRESS
053.006 345 1676X PUSH H
053.007 353 1677X XCHG (HL) = NEXT DATA BYTE ADDRESS
053.010 311 1678X RET ENTER PROCESSOR

```

```

1679X
053.011 341      1680X SOP1  POP  H
053.012 321      1681X SOP2  POP  D
053.013 076 040  1682X      MVI  A,EC,ILD      ILLEGAL OPTION SPECIFICATION
053.015 067      1683X      STC
053.016 311      1684X      RET
053.017          1685X      XTEXT PBF
    
```

```

1687X **      PBF - PROCESS BYTE FLAG
1688X *
1689X *      PROCESS BYTE FLAG OPTIONS, THE FORMAT FOR TABLE ENTRIES IS:
1690X *
1691X *      <MASK>,<VALUE>,<LOW ADDR.>,<HIGH ADDR.>
1692X *
1693X *
1694X *      ENTRY: (HL)      = ADDRESS OF TABLE VECTOR
1695X *
1696X *      EXIT: (RET)      = 'C' CLEAR IF OK
1697X *                  = 'C' SET IF ERROR
1698X *                  (A) = ERROR CODE
1699X *
1700X *      USES: ALL
    
```

```

1701X *
1702X *
053.017 176      1703X PBF  MOV  A,M      (A) = MASK
053.020 043      1704X      INX  H
053.021 365      1705X      PUSH PSW
053.022 246      1706X      ANA  M      MASK UNUSED BITS OUT OF VALUE
053.023 127      1707X      MOV  D,A      (D) = VALUE
053.024 043      1708X      INX  H
053.025 315 211 030 1709X      CALL $HLIHL      (HL) = ADDRESS TO STORE BYTE
053.030 361      1710X      POP  PSW
053.031 057      1711X      CMA
053.032 246      1712X      ANA  M      MASK OUT PREVIOUS VALUE
053.033 262      1713X      ORA  D      SET NEW FLAGS
053.034 167      1714X      MOV  M,A      PATCH IT
053.035 311      1715X      RET
053.036          1716X      XTEXT PBF
    
```

```

1718X **      PBF - PROCESS BYTE VALUES
1719X *
1720X *      PROCESS BYTE VALUE OPTIONS. THE FORMAT FOR TABLE ENTRIES
1721X *      IS:
1722X *
1723X *      <DEFAULT RADIX>,<MIN.>,<MAX.>,<LOW ADDR.>,<HIGH ADDR.>
1724X *
1725X *
1726X *      ENTRY: (BC)      = NEXT TEXT CHARACTER ADDRESS
1727X *                  (HL) = TABLE VECTOR ADDRESS
1728X *
    
```

```

1729X *      EXIT:  (BC)  = UPDATED
1730X *      'C' CLEAR IF OK
1731X *      'C' SET IF ERROR
1732X *      (A) = ERROR CODE
1733X *
1734X *      USES:  ALL
1735X *
1736X
053.036 176 1737X PBV  MOV  A,M      (A) = DEFAULT RADIX
053.037 043 1738X      INX  H
053.040 345 1739X      PUSH H      SAVE VECTOR ADDRESS
053.041 315 357 051 1740X  CALL DNF      (HL) = VALUE
053.044 332 107 053 1741X  JC   PBV2
053.047 174 1742X  MOV  A,M
053.050 247 1743X  ANA  A
053.051 302 107 053 1744X  JNZ  PBV2
053.054 321 1745X  POP  D
053.055 353 1746X  XCHG      (HL) = NEXT TABLE ADDRESS, (E) = VALUE
053.056 305 1747X  PUSH  B      SAVE TEXT POINTER
053.057 106 1748X  MOV  B,M      (B) = MIN.
053.060 043 1749X  INX  H
053.061 116 1750X  MOV  C,M      (C) = MAX.
053.062 043 1751X  INX  H
053.063 315 211 030 1752X  CALL $HLIHL  (HL) = BYTE VALUE ADDRESS
053.066 173 1753X  MOV  A,E
053.067 270 1754X  CMP  B
053.070 332 111 053 1755X  JC   PBV3      (A) < MIN.
053.073 014 1756X  INR  C
053.074 312 103 053 1757X  JZ   PBV1      IGNORE COMPARE IF C=3770
053.077 271 1758X  CMP  C
053.100 322 111 053 1759X  JNC  PBV3      (A) >= MAX. + 1
053.103 301 1760X PBV1  POP  B      RESTORE TEXT ADDR.
053.104 167 1761X  MOV  M,A     PATCH IT
053.105 257 1762X  XRA  A      CLEAR CARRY
053.106 311 1763X  RET
1764X
053.107 341 1765X PBV2  POP  H
053.110 305 1766X  PUSH B
053.111 301 1767X PBV3  POP  B
053.112 076 037 1768X  MVI  A,EC.ILV  ILLEGAL VALUE SPECIFICATION
053.114 067 1769X  STC
053.115 311 1770X  RET
053.116 1771X  XTEXT TJMP

```

```

1773X **      $TJMP - TABLE JUMP.
1774X *
1775X *      USAGE
1776X *
1777X *      CALL  $TJMP      (A) = INDEX
1778X *      DW    ADDR1
1779X *
1780X *
1781X *

```



```

1782X *      DW      ADDRn
1783X *
1784X *      ENTRY   (A) = INDEX
1785X *      EXIT    TO PROCESSOR
1786X *      (A) = INDEX*2
1787X *      USES   NONE.
1788X
1789X
031.061     1790X $TJMP EQU    31061A      IN H17 ROM; (A) = INDEX*2
1791X
031.062     1792X $TJMP EQU    31062A      IN H17 ROM
053.116     1793      XTEXT  MUB6

```

```

1795X **     $MUB6 - MULTIPLY BX16 UNSIGNED.
1796X *
1797X *     $MUB6 MULTIPLIES A 16 BIT VALUE BY A 8
1798X *     BIT VALUE.
1799X *
1800X *     ENTRY   (A) = MULTIPLIER
1801X *           (DE) = MULTIPLICAND
1802X *     EXIT    (HL) = RESULT
1803X *           'Z' SET IF NOT OVERFLOW
1804X *     USES   A,F,H,L
1805X
031.007     1806X $MUB6 EQU    31007A      IN H17 ROM
053.116     1807X      XTEXT  DADA2
1808

```

```

1810X **     $DADA. - ADD (0,A) TO (H,L)
1811X *
1812X *     ENTRY   NONE
1813X *     EXIT    (HL) = (HL) + (0A)
1814X *     USES   A,F,H,L
1815X
030.101     1816X $DADA. EQU    30101A      IN H17 ROM
053.116     1817X      XTEXT  HLIHL
1818

```

```

1820X **     $HLIHL - LOAD HL INDIRECT THROUGH HL.
1821X *
1822X *     (HL) = ((HL))
1823X *
1824X *     ENTRY   NONE
1825X *     EXIT    NONE
1826X *     USES   A,H,L
1827X
030.211     1828X $HLIHL EQU    30211A      IN H17 ROM

```

*HLIHL

053.116

1829

XTEXT CVD

1831X ** *CVD - CHECK FOR VALID DIGIT.
1832X *
1833X * CVD EXAMINES A DIGIT TO SEE IF IT IS A VALID DECIMAL DIGIT.
1834X *
1835X * ENTRY (HL) = ADDRESS OF CHARACTER
1836X * EXIT 'C' SET IF ILLEGAL
1837X * (A) = VALUE
1838X * USES A,F

053.116 176 1841X *CVD MOV A,M (A) = CHARACTER
053.117 328 060 1842X *CVD. SUI '0'
053.121 330 1843X RC ILLEGAL
053.122 376 012 1844X CPI '0'
053.124 077 1845X CMC
053.125 311 1846X RET
053.126 1847 XTEXT TBL5

1849X ** *TBL5 - TABLE SEARCH
1850X *
1851X * TABLE FORMAT
1852X *
1853X * DB KEY1,VAL1;
1854X * :
1855X * :
1856X * DB KEYN,VALN
1857X * DB 0
1858X *
1859X * ENTRY (A) = PATTERN
1860X * (H,L) = TABLE FWA
1861X * EXIT (A) = PATTERN IF FOUND
1862X * 'Z' SET IF FOUND
1863X * 'Z' CLEAR IF NOT FOUND OR PATTERN=0 /78.10.6C/
1864X * USES A,F,H,L
1865X
1866X

053.126 305 1867X *TBL5 PUSH B
053.127 376 000 1868X CPI 0 /78.10.6C/
053.131 312 153 053 1869X JZ TBL2 /78.10.6C/
053.134 107 1870X MOV B,A
053.135 176 1871X TBL1 MOV A,M (A) = CHARACTER
053.136 043 1872X INX H
053.137 270 1873X CMP B
053.140 312 155 053 1874X JZ TBL3 IF MATCH
053.143 247 1875X ANA A
053.144 043 1876X INX H SKIP PAST
053.145 302 135 053 1877X JNZ TBL1 IF NOT END OF TABLE
053.150 053 1878X DCX H

SET - SET SYSTEM PARAMETERS
COMMON DECKS

\$TBLS HEATH HBASM V1.4 01/20/78
14:50:59 16-MAY-80

PAGE 43

053.151	053	1879X	DCX	H		
053.152	257	1880X	XRA	A	SET TO ZERO FOR OLD USERS	/78.10.GC/
053.153	376 001	1881X	TBL2	CPI	1	CLEAR ZERO /78.10.GC/
		1882X				
		1883X	*	DONE		
		1884X				
053.155	301	1885X	TBL3	POP	B	
053.156	311	1886X		RET		
053.157		1887		XTEXT	WTBLS	

1889X ** \$WTBLS - TABLE SEARCH
 1890X *
 1891X * \$WTBLS LOOKS UP WORD VALUES IN A TABLE, USING A ONE-BYTE
 1892X * KEY.
 1893X *
 1894X * TABLE FORMAT
 1895X *
 1896X * DB KEY1
 1897X * DW VAL1
 1898X * .
 1899X * .
 1900X * DB KEYN
 1901X * DW VALN
 1902X * DB 0
 1903X *
 1904X * ENTRY (A) = PATTERN
 1905X * (H,L) = TABLE FWA
 1906X * EXIT (A) = PATTERN IF FOUND
 1907X * 'Z' SET IF FOUND
 1908X * USES A,F,H,L
 1909X
 1910X

053.157	305	1911X	\$WTBLS	PUSH	B	
053.160	107	1912X		MOV	B,A	
053.161	176	1913X	\$WTBLS	MOV	A,M	(A) = CHARACTER
053.162	043	1914X		INX	H	
053.163	270	1915X		CMF	B	
053.164	312 201 053	1916X		JZ	\$WTBLS	IF MATC
053.167	247	1917X		ANA	A	
053.170	043	1918X		INX	H	
053.171	043	1919X		INX	H	SKIP PAST
053.172	302 161 053	1920X		JNZ	\$WTBLS	IF NOT END OF TABLE
053.175	053	1921X		DCX	H	
053.176	053	1922X		DCX	H	
053.177	053	1923X		DCX	H	
053.200	264	1924X		GRA	H	CLEAR 'Z'
		1925X				
		1926X	*	DONE		
		1927X				
053.201	301	1928X	\$WTBLS	POP	B	
053.202	311	1929X		RET		
053.203		1930		XTEXT	DNV	

```

1932X ** $DNV = DECODE NUMERIC VALUE.
1933X *
1934X * $DNV DECODES A NUMERIC VALUE (IN THE FORM OF AN ASCII STRING)
1935X * INTO A BINARY NUMBER. THE MAXIMUM MAGNITUDE IS
1936X * 65535D.
1937X *
1938X * THE NUMBER MAY CONTAIN A POSTRADIX OF 'B' (BINARY)
1939X * 'O' OR 'D' (OCTAL) OR 'D' (DECIMAL)
1940X *
1941X * ENTRY (HL) = ADDRESS OF FIRST BYTE OF NUMBER
1942X * (A) = DEFAULT BASE (2 FOR BINARY, 10 FOR DECIMAL, ETC.)
1943X * EXIT 'C' CLEAR IF OK
1944X * (HL) ADVANCED PAST NUMBER (AND POSTRADIX)
1945X * (DE) = VALUE
1946X * 'C' SET IF ERROR
1947X * USES ALL
1948X *
1949X *
053.203 062 320 053 1950X $DNV STA $DNVA SET DEFAULT BASE
053.206 104 1951X MOV B,H
053.207 115 1952X MOV C,L (BC) = TEXT ADDRESS
1953X
1954X * SCAN FOR POSTRADIX
1955X
053.210 176 1956X $DNV1 MOV A,M
053.211 315 117 053 1957X CALL %CVD CHECK FOR VALID DECIMAL DIGIT
053.214 043 1958X INX H
053.215 322 210 053 1959X JNC $DNV1 MORE TO GO
053.220 053 1960X DCX H REMOVE EXTRA INCREMENT
053.221 171 1961X MOV A,C
053.222 275 1962X CNP L SEE IF THERE WERE ANY NUMBERS
053.223 067 1963X STC ASSUME NOT
053.224 310 1964X RE ERROR
1965X
1966X * OUT OF NUMBERS. SEE IF POSTRADIX FOLLOWS
1967X
053.225 176 1968X MOV A,M (A) = PROPOSED POSTRADIX
053.226 345 1969X PUSH H SAVE END ADDRESS
053.227 041 321 053 1970X LXI H,$DNVB
053.232 247 1971X ANA A
053.233 312 253 053 1972X JZ $DNV2 NO POSTRADIX
053.236 315 126 053 1973X CALL %TBLS
053.241 176 1974X MOV A,M
053.242 302 253 053 1975X JNE $DNV2 NOT POSTRADIX
053.245 341 1976X POP H
053.246 043 1977X INX H SKIP POSTRADIX
053.247 345 1978X PUSH H
053.250 062 320 053 1979X STA $DNVA SET NEW POSTRADIX
053.253 021 000 000 1980X $DNV2 LXI D,0 (DE) = ACCUMULATOR
1981X
1982X * BUILD NUMBER
1983X
053.256 072 320 053 1984X $DNV3 LDA $DNVA (A) = BASE
053.261 365 1985X PUSH PSW SAVE BASE
053.262 315 007 031 1986X CALL %MUS6 MULTIPLY
053.265 321 1987X POP D (D) = BASE

```

\$DNU

```

053.266 332 316 053 1988X JC $DNU4 OVERFLOW
053.271 012 1989X LDAX B (A) = DIGIT
053.272 326 060 1990X SUI '0'
053.274 003 1991X INX B
053.275 272 1992X CMP D COMPARE TO BASE
053.276 077 1993X CMC
053.277 332 316 053 1994X JC $DNU4 TOO LARGE A DIGIT
053.302 315 101 030 1995X CALL $DADA. ADD TO VALUE
053.305 353 1996X XCHG (DE) = VALUE
053.306 012 1997X LDAX B
053.307 315 117 053 1998X CALL $CUB.
053.312 322 256 053 1999X JNC $DNU3 MORE TO GO
053.315 247 2000X ANA A CLEAR CARRY
053.316 341 2001X $DNU4 POP H RESTORE POINTER
053.317 311 2002X RET EXIT
2003X
053.320 000 2004X $DNU4 DB 0 DEFAULT BASE
053.321 102 002 2005X $DNU4 DB 'B',2 POSTRADIX TABLE
053.323 117 010 2006X DB '0',8
053.325 121 010 2007X DB '0',8
053.327 104 012 2008X DB 'H',10
053.331 000 2009X DB 0
053.332 2010 XTEXT TYPTX

```

2012X ** \$TYPTX - TYPE TEXT.

2013X *

2014X * \$TYPTX IS CALLED TO TYPE A BLOCK OF TEXT ON THE SYSTEM CONSOLE.

2015X *

2016X * IMBEDDED ZERO BYTES INDICATE A CARRIAGE RETURN LINE FEED;

2017X * A BYTE WITH THE 2000 BIT SET IS THE LAST BYTE IN THE MESSAGE.

2018X *

2019X * ENTRY (RET) = TEXT

2020X * EXIT TO (RET+LENGTH)

2021X * USES A;F

2022X

2023X

031.136 2024X \$TYPTX EQU 31136A IN H17 ROM

2025X

031.144 2026X \$TYPTX EQU 31144A IN H17 ROM

SET - SET SYSTEM PARAMETERS
 CROSS REFERENCE TABLE

XREF V1.1
 PAGE 51

FLT.CDB	000006	526L		
FLT.CFC	000002	522L	961	
FLT.CRF	000003	523L	966	
FLT.CTY	000000	520L	902	
FLT.CWI	000001	521L	922	
FLT.NNC	000004	524L		
FLT.SAL	000012	529L	1090	
FLT.TDT	000005	525L	830	
FST.	042212	654L		
FST1	052203	1519L	1551	
FST2	052216	1526	1532L	1539
FST3	052234	1534	1541L	1544
FST4	052256	1536	1553L	
FT.ABS	000000	537E	587	
FT.BAC	000003	540E		
FT.DD	000001	233E		
FT.DR	000002	234E		
FT.OU	000010	236E	1423	
FT.DW	000004	235E	1423	
FT.PIC	000001	538E		
FT.REL	000002	539E		
HOS1	050326	1096	1100L	
HOSPRC	050162	1047	1064L	
HOSSAL	050164	1068	1083L	
HOSSALI	000000	1056	1057	1067E
HOSTAB	050122	1048	1053L	
HOSTAPE	050161	1053	1062L	
I.CONFL	000004	360E	361	
I.CONTY	000001	347E	348	
I.CONWI	000003	353E	354	
I.CSLMD	000000	337E		
I.CUSOR	000002	350E	351	
IDC.CGN	000010	241L		
IDC.CSI	000011	242L		
IDC.DDA	000002	230L	237	251
IDC.DES	000016	248L		
IDC.DEV	000020	249L		
IDC.DIL	000021	251E		
IDC.RIR	000023	253L		
IDC.DRL	000010	245E		
IDC.DTA	000014	247L		
IDC.FLG	000004	232L	245	1421
IDC.GRT	000005	239L		
IDC.LGN	000012	243L		
IDC.LNK	000000	229L	1419	
IDC.LSI	000013	244L		
IDC.SFG	000007	240L		
IDC.SGL	000003	237E		
IDC.UNI	000022	250L		
IDCCTD	000001	257E	1417	
IDCELEN	000052	255E		
LBD.	042223	669L		
LBD.	052303	1582	1595L	
LDD	051172	1200	1241L	
LDD1	051240	1266L	1279	1290
LF	000012	63E		
M.CDCA	000017	496L	953	
M.CDLY	000016	495L	950	953

SET - SET SYSTEM PARAMETERS
CROSS REFERENCE TABLE

XREF V1.1
PAGE 53

S.DATE	040277	319L								
S.DCS	041033	434L								
S.DDATA	040366	399L								
S.DDGRP	040364	396L								
S.DDLDA	040360	394L								
S.DDLEN	040362	395L								
S.DDOPC	040370	400L								
S.DFWA	040354	389L								
S.DIREA	041016	428L								
S.DLINK	040346	386L	949	1085						
S.FASER	041013	427L								
S.FCI	041021	429L								
S.GRTQ	024000	263E								
S.GRT1	025000	264E								
S.GRT2	026000	265E								
S.GUP	041027	431L								
S.HIMEM	040316	322L								
S.INT	040343	277L	374							
S.JUMPS	041010	425L								
S.MDUNT	041032	433L								
S.OFWA	040350	387L								
S.OHAX	040324	328L								
S.OSN	041004	416L								
S.OVLE	041000	413L								
S.OVFL	040371	409L								
S.OVLS	040376	412L								
S.OVSTK	041035	441L								
S.RFWA	040356	390L								
S.SCI	041024	430L								
S.SCR	041120	479L								
S.SDD	041010	426L								
S.SDVR	041146	279L	281							
S.SSN	041002	415L								
S.SYSM	040320	324L								
S.TIME	040312	321L								
S.UCSF	040372	410L								
S.UCSL	040374	411L								
S.USRM	040322	326L								
S.VAL	040277	276L	317							
SET1	042354	699	706L							
SETA	043015	711	722L							
SETAE	043060	722	733L							
SETHLP	047131	718	999L							
SETHOS	050105	720	1047L							
SETSY	043244	716	789L							
SETTT	044065	717	849L							
SETVER	050056	719	1034L							
SHD	052000	831	903	923	962	967	1102	1404L		
SHOA	052127	1408	1462L							
SHOB	052144	1409	1463L							
SKB	052153	614	1343	1377	1476L					
SKB1	052152	1475L	1478							
SNA	042201	614L								
SQP	052350	674	791	851	1049	1659L				
SQP1	042226	674L								
SQP1	053011	1662	1680L							
SQP2	053012	1668	1681L							
SSYSTP	043303	807	817L							

CROSS REFERENCE TABLE

STACK	042200	283E	599						
STACKL	001032	281E							
STPI	000000	799	806E						
STICTY	044266	879	894E						
STTFIL	044344	885	930L						
STYWIN	044308	882	910L						
SYDD	040130	273E							
SYHCP	043342	810	836L						
SYHLPI	000001	800	809E						
SYOPRC	043277	789	804L	806	809				
SYOTAB	043261	790	796L						
SYOTABE	043276	796	802L						
SYSCALL	000377	94E	694	772	1411	1430	1449	1454	1458
TAB	000011	72E	1035						
TBL1	053135	1871L	1877						
TBL2	053153	1869	1881L						
TBL3	053155	1874	1885L						
TBL5	042215	659L							
TTHLP	045044	888	973L						
TTHLPI	000003	872	887E						
TTOPRC	044256	849	876L	878	881	884	887		
TTOTAB	044102	850	855L						
TTOTABE	044255	855	874L						
USERFWA	042200	284E	586	588	589				
VERS	000026	92E	1036	1036					
WDR	051272	1211	1306L						
WIDI	000001	870	881E						
WTBLS	042220	664L							

24874 BYTES FREE