

OPTIONS NODECK,LIST,XREF,NOREL,OBJ(P)

THE LIST OF OPTIONS USED DURING THIS ASSEMBLY IS-- NODECK,LIST,XREF,NOREL,OBJ



##1TRK BIS - CYL 0, TRACK 1

ERR LOC    OBJECT CODE            ADDR STMT SOURCE STATEMENT            VER 15, MOD 00    25/09/15    PAGE    2

0000	2	PRINT ON,NODATA
	3 ##1TRK	START
	4 *	@LVL    EXP-Y
	6+	PRINT ON

@LVLEQ - SYTEM LEVEL NUMBER

ERR LOC    OBJECT CODE            ADDR STMT SOURCE STATEMENT                    VER 15, MOD 00    25/09/15    PAGE    3

0005	8+@SYLVL	EQU	5	SYSTEM LEVEL NUMBER
	9+	PRINT	ON	
	10 *	@SYS	EXP-Y	
	12+	PRINT	ON	

@SYSEQ - SYSTEM SOFTWARE EQUATES

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	25/09/15	PAGE	4
					14+	*****					
					15+	*	CPU EQUATES				*
					16+	*****					
					17+	*					
					18+	***	REGISTER EQUATES				
					19+	*					
				0002	20+	@REGL	EQU 2				HARDWARE REGISTER LENGTH
				0001	21+	@BR	EQU 1				BASE REGISTER
				0002	22+	@XR	EQU 2				USABLE INDEX REGISTER
				0004	23+	@PSR	EQU 4				PROGRAM STATUS REGISTER
				0008	24+	@ARR	EQU 8				ADDRESS RECALL REGISTER
				0010	25+	@IAR	EQU 16				INSTRUCTION ADDRESS REGISTER
				0020	26+	@P1IAR	EQU 32				PROGRAM LEVEL 1 IAR
				0040	27+	@P2IAR	EQU 64				PROGRAM LEVEL 2 IAR
				00C0	28+	@I1IAR	EQU X'C0'				INTERRUPT LEVEL 1 IAR Q-CODE
					29+	*					
					30+	***	EQUATES FOR BYTES OF AN INSTRUCTION				
					31+	*					
				0001	32+	@Q	EQU 1				Q-CODE BYTE
				0001	33+	@VQ	EQU 1				VARIABLE Q CODE FOR LENGTH
				0002	34+	@D1	EQU 2				1ST DISPLACEMENT
				0003	35+	@OP1	EQU 3				1ST ADDRESS
				0004	36+	@DOP2	EQU 4				2ND ADDR OF 5 BYTE INSTR.
				0004	37+	@OPD2	EQU 4				2ND DISP OF 5 BYTE INSTR.
				0003	38+	@DD2	EQU 3				2ND DISP OF 4 BYTE INSTR.
				0005	39+	@OP2	EQU 5				2ND ADDR OF 5 BYTE INSTR.
				0003	40+	@INST3	EQU 3				LENGTH OF 1 DISP INSTRUCTION
				0004	41+	@INST4	EQU 4				LENGTH OF 1 ADDR INSTRUCTION
				0005	42+	@INST5	EQU 5				LENGTH OF 1 DISP 1 ADDR INSTR.
				0006	43+	@INST6	EQU 6				LENGTH OF 2 ADDR INSTR.
					44+	*					
					45+	***	CONDITION CODES FOR BRANCHES				
					46+	*					
				0087	47+	@UCB	EQU X'87'				UNCONDITIONAL BRANCH
				0080	48+	@NOP	EQU X'80'				NO BRANCH
				0084	49+	@BH	EQU X'84'				BRANCH HIGH
				0082	50+	@BL	EQU X'82'				BRANCH LOW
				0081	51+	@BE	EQU X'81'				BRANCH EQUAL
				0004	52+	@BNH	EQU X'04'				BRANCH NOT HIGH
				0002	53+	@BNL	EQU X'02'				BRANCH NOT LOW
				0001	54+	@BNE	EQU X'01'				BRANCH NOT EQUAL
				0088	55+	@BOZ	EQU X'88'				BRANCH OVERFLOW ZONED
				00A0	56+	@BOL	EQU X'A0'				BRANCH OVERFLOW LOGICAL
				0008	57+	@BNOZ	EQU X'08'				BRANCH NO OVERFLOW ZONED
				0020	58+	@BNOL	EQU X'20'				BRANCH NO OVERFLOW LOGICAL
				0010	59+	@BT	EQU X'10'				BRANCH TRUE
				0090	60+	@BF	EQU X'90'				BRANCH FALSE
				0084	61+	@BP	EQU X'84'				BRANCH PLUS
				0082	62+	@BM	EQU X'82'				BRANCH MINUS
				0081	63+	@BZ	EQU X'81'				BRANCH ZERO
				0004	64+	@BNP	EQU X'04'				BRANCH NOT PLUS
				0002	65+	@BNM	EQU X'02'				BRANCH NOT MINUS
				0001	66+	@BNZ	EQU X'01'				BRANCH NOT ZERO
					67+	*					
					68+	***	MISCELLANEOUS CONSTANTS				
					69+	*					

@SYSEQ - SYSTEM SOFTWARE EQUATES

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	25/09/15	PAGE	5
		0000	70+	@ZERO	EQU 0				ZERO
		0001	71+	@B1	EQU 1				BINARY ONE
		00F0	72+	@DZERO	EQU X'F0'				DECIMAL ZERO
		0040	73+	@BLANK	EQU C' '				EBCDIC BLANK
		006B	74+	@COMMA	EQU C', '				EBCDIC COMMA
		0061	75+	@SLASH	EQU C'/'				EBCDIC FORWARD SLASH
		005B	76+	@DOLAR	EQU C'\$'				EBCDIC DOLLAR SIGN
		005C	77+	@ASTER	EQU C'*'				EBCDIC ASTERISK
		007B	78+	@NUMBR	EQU C'#'				EBCDIC NUMBER #
		007C	79+	@ASIGN	EQU C'@'				EBCDIC ASSIGN @
		00C1	80+	@CHARA	EQU C'A'				EBCDIC CHAR A
		00C6	81+	@CHARF	EQU C'F'				EBCDIC CHAR F
		00D9	82+	@CHARR	EQU C'R'				EBCDIC CHAR R
		00E9	83+	@CHARZ	EQU C'Z'				EBCDIC CHAR Z
		001E	84+	@EOS	EQU X'1E'				RETURN CARRIAGE
		001C	85+	@EOF	EQU X'1C'				END OF FILE CHARACTER
		005A	86+	@UPARW	EQU X'5A'				UPARROW FROM KEYBOARD INPUT
		004E	87+	@CPLUS	EQU C'+'				EBCDIC PLUS SIGN
		0060	88+	@MINUS	EQU C'-'				EBCDIC MINUS SIGN
		0001	89+	@DCALK	EQU X'01'				DCAL REQUESTED INDICATOR
		0020	90+	@PGCSZ	EQU 32				CORE SIZE IN PAGES
		2000	91+	@MINCR	EQU 256*@PGCSZ				CORE SIZE IN BYTES
		00F4	92+	@LINSZ	EQU 244				LENGTH OF INPUT LINE BUFFER
		0018	93+	@DTRSZ	EQU 24				NO. OF DISK SECTORS PER TRACK
		0030	94+	@SECCY	EQU 48				SECTORS PER CYLINDER
		0060	95+	@CARDL	EQU 96				LENGTH OF 3700 INPUT CARD
		0050	96+	@BCRDL	EQU 80				LENGTH OF 5081 INPUT CARD
		0005	97+	@MAPEN	EQU 5				DISP TO END OF FE CORE MAP
		0007	98+	@SDFLN	EQU 7				LENGTH OF SDF
		0006	99+	@VOLID	EQU 6				LENGTH OF DISK ID FIELD
		0007	100+	@HDLN	EQU 7				LENGTH OF PROGRAM HEADER
		0011	101+	@CLON	EQU X'11'				TURN ON COMMAND LITE Q-CODE
		0010	102+	@CLOFF	EQU X'10'				TURN off COMMAND LITE Q-CODE
			104+	*****					
			105+	DISK REGION EQUATES					*
			106+	*****					
			107+	*					
		0100	108+	@SCTS	EQU 256				LENGTH OF ONE SECTOR
		0500	109+	@WSFIT	EQU X'0500'				SECTOR ADDR OF WS FIT SCTR
		0503	110+	@WSTBL	EQU X'0503'				SECTOR ADDR OF WORKING STORAGE
		0005	111+	@DWBCY	EQU 5				BASE CYL SYSTEM WORK FILE
		0003	112+	@DWTB1	EQU 3				LOGICAL SCTR 1ST TEXT BLOCK
		00C0	113+	@DWSIZ	EQU 192				NO. OF WORK FILE DISK SECTORS
		0004	114+	@DSBCY	EQU 4				BASE CYL SYSTEM ROUTINES
		0000	115+	@DSCS1	EQU 0				COMPILER SUBROUTINE 1ST SCTR
		0007	116+	@DVBCY	EQU 7				BASE CYL VIRTUAL MEMORY
		0000	117+	@VMFD1	EQU 0				FILE DIRECTORY 1 PAGE
		0001	118+	@VMFD2	EQU 1				FILE DIRECTORY 2 PAGE
		0001	119+	@VMTRL	EQU 1				TRACE REFERENCE LIST PAGE
		0002	120+	@VMRS3	EQU 2				START OF VM RESIDENT SUBROUTINE
		0056	121+	@VENTA	EQU 86				FIRST PSEUDO CODE PAGE IN VM
		00FE	122+	@VMDDV	EQU 254				FUNC AND ARRAY TABLE - PAGE ONE
		0009	123+	@DCBCY	EQU 9				BASE CYL COMPILER VADDR TABLES
		0040	124+	@DCST1	EQU 64				STMT ADDRESS TABLE 1ST SECTOR
		0050	125+	@DCBT1	EQU 80				BRANCH ADDRESS TABLE 1ST SECTOR

@SYSEQ - SYSTEM SOFTWARE EQUATES

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	25/09/15	PAGE	6
					127+	*****					
					128+	*	DISK IOCR EQUATES				*
					129+	*****					
					130+	*					
					131+	***	DISK PARAMETER LIST (DPL) EQUATES				
					132+	*					
			0000	133+	@DCTRL	EQU	0				CONTROL PARAMETER
			0001	134+	@DCYL	EQU	1				LOGICAL CYLINDER NUMBER
			0002	135+	@DSAD	EQU	2				HEAD/SECTOR ADDRESS
			0003	136+	@DCNT	EQU	3				SECTOR COUNT
			0004	137+	@DBFR1	EQU	4				1ST BYTE OF DATA AREA
			0005	138+	@DBFR2	EQU	5				DATA AREA ADDRESS
			0002	139+	@DSPIN	EQU	X'02'				SPINDLE BIT IN DISK ADDRESS
			0006	140+	@DPLNG	EQU	6				LENGTH OF DSL
			0000	141+	@DPOS	EQU	X'00'				DPL - SEEK FUNCTION CODE
			0001	142+	@DGET	EQU	X'01'				DPL - READ FUNCTION CODE
			0002	143+	@DPUT	EQU	X'02'				DPL - WRITE FUNCTION CODE
			0031	144+	@DVERFY	EQU	X'31'				DPL - VERIFY FUNCTION CODE
			00FF	145+	@DWAIT	EQU	X'FF'				DPL - WAIT I/O COMPLETE FUNC COD
			0003	146+	@DSIVF	EQU	X'03'				SIO CTRL CODE FOR VERIFY
				147+	*						
			0002	148+	@DADDR	EQU	2				LENGTH OF DISK ADDRESS
			0002	149+	@VADDR	EQU	2				LENGTH OF VIRTUAL ADDRESS
			0002	150+	@CADDR	EQU	2				LENGTH OF CORE ADDRESS
				152+	*****						
				153+	*		PRINT PARAMETER LIST (PPL) EQUATES				*
				154+	*****						
				155+	*						
			0004	156+	@PPLNG	EQU	4				LENGTH OF PPL
			0000	157+	@PCTRL	EQU	0				CONTROL BYTE DISPLACEMENT
			0001	158+	@PRCNT	EQU	1				COUNT BYTE DISPLACEMENT
			0003	159+	@PDATA	EQU	3				DATA ADDR DISPLACEMENT
			0040	160+	@PRINT	EQU	X'40'				PRINT CONTROL
			0080	161+	@RETRN	EQU	X'80'				RETURN CARRIER CONTROL
			00C0	162+	@PRETR	EQU	@PRINT+@RETRN				PRINT AND RETURN CARRIER
			0010	163+	@TBLEF	EQU	X'10'				TAB LEFT CONTROL
			0001	164+	@INDEX	EQU	X'01'				INDEX FORMS CONTROL
			0011	165+	@TBLIX	EQU	@TBLEF+@INDEX				TAB LEFT AND INDEX CONTROL
			00FF	166+	@PWAIT	EQU	X'FF'				WITH AND CHECK ERROR CONTROL
			004F	167+	@RLDWN	EQU	X'4F'				ROLL DOWN CONTROL (CRT ONLY)
			0000	168+	@TBCNT	EQU	0				TAB LEFT COUNT
			0080	169+	@RTRNC	EQU	X'80'				CARRIER RETURN COUNT
			0075	170+	@EOFTC	EQU	X'75'				EOF RECORD TYPE CODE
				171+	*						
				172+	***		STATEMENT/SEGMENT HEADER EQUATES				
				173+	*						
			0000	174+	@SDF0	EQU	0				DISP TO NULL SEG INDICATOR
			0001	175+	@SDF1	EQU	1				DISP TO LENGTH OF SEGMENT
			0002	176+	@SDF2	EQU	2				DISP TO SEGMENTATION CODE
			0003	177+	@SDF3	EQU	3				DISP TO END OF SDF
			0005	178+	@SBLN	EQU	5				DISP TO STMT BINARY LINE NO.
			0006	179+	@STYPE	EQU	6				DISP TO STMT TYPE CODE
			0007	180+	@STEXT	EQU	7				DISP TO 1ST TEXT BYTE OF STMT
			0080	181+	@SNULL	EQU	X'80'				MASK FOR NULL SEG INDICATOR
				182+	*						* 1 = SEGMENT IS NULL

@SYSEQ - SYSTEM SOFTWARE EQUATES

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00 25/09/15 PAGE 7
				183+*			* 0 = SEGMENT IS NOT NULL	
				184+*				
				185+*			FOLLOWING ARE THE MASKS FOR THE SEGMENTATION	
				186+*			CODE. THE SEGMENTATION IS INDICATED BY VALUE	
				187+*			IN @SDF2 AS FOLLOWS:	
		0000		188+@SONLY	EQU	0	ONLY SEG. IN RECORD	
		0001		189+@SIST	EQU	1	1ST SEG. OF A MULTI-SEG RCD	
		0003		190+@SMIDL	EQU	3	MIDDLE SEG. OF A MULTI-SEG RCD	
		0002		191+@SLAST	EQU	2	LAST SEG. OF MULTI-SEG RCD	
		0002		192+@SBLNL	EQU	2	LENGTH OF STMT BINARY LINE NO.	
				193+*				
				194+****			FILE INDEX TABLE EQUATES SECTION	
				195+*				
				196+*			ALL DISPLACEMENT ARE CALCULATED FROM THE	
				197+*			* FIRST BYTE OF THE FIT TO THE RIGHTMOST BYTE	
				198+*			* OF THE SPECIFIED FIELD UNLESS OTHERWISE	
				199+*			* NOTED.	
				200+*				
		0002		201+@FDLNC	EQU	2	DISP TO FILE LINE COUNT	
		0002		202+@FLLNC	EQU	2	LNG OF FILE LINE COUNT FIELD	
		0000		203+@FDDBC	EQU	0	DISP TO FILE DATA BLOCK COUNT	
		0001		204+@FLDBC	EQU	1	LNG OF FILE DATA BLOCK COUNT	
		0009		205+@FLACE	EQU	9	DISP O ADDR OF CURR ENTRY	
		000B		206+@FDFNA	EQU	11	DISP TO ADDR OF 1ST NULL ENTRY	
		0002		207+@FLFNA	EQU	2	LNG OF ADDR OF 1ST NULL ENTRY	
		000C		208+@FDE1	EQU	12	DISP TO 1ST BYTE OF 1ST ENTRY	
		0004		209+@FLENT	EQU	4	LNG OF A FIT ENTRY	
				210+*				
				211+*			ENTRY FIELD DISPLACEMENTS ARE CALCULATED FROM	
				212+*			* THE 1ST BYTE OF THE ENTRY.	
				213+*				
		0000		214+@FDSD	EQU	0	DISP TO DB SECTOR DISP	
		0001		215+@FLSD	EQU	1	LNG OF DB SECTOR DISP FIELD	
		0002		216+@FDHLN	EQU	2	DISP TO HIGH LINE NO. FIELD	
		0002		217+@FLHLN	EQU	2	LNG OF HIGH LINE NO. FIELD	
		0003		218+@FDNSC	EQU	3	DISP TO DB NULL SPACE CNT FIELD	
		0001		219+@FLNSC	EQU	1	LNG OF DB NULL SPACE CNT FIELD	
				220+*				
				221+*			END OF SYSTEM SOFTWARE EQUATES	
				222+			PRINT ON	
				223 *			@FXD EXP-Y	
				225+			PRINT ON	



@FXDEQ - FIXED ADDRESSES FOR SYSTEM NUCLEUS

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00 25/09/15 PAGE 8
			227+		*****	
			228+		GLOBAL INDICATORS STORED IN THE SYSTEM NUCLEUS, ENTRY POINTS *	
			229+		FOR SYSNUC INTERFACE ROUINES. *	
			230+		*****	
0000			231+	ORG	X'0000'	*
	0000		232+	\$\$ZERO EQU	*	ENTRY POINT TO LOAD DUMP PGM
	0004		233+	\$FEARR EQU	\$\$ZERO+4	VALUE OF ADDR IN ARR ON FE AID
			234+			
	0025		235+	\$DISKN EQU	\$\$ZERO+37	ADDR OF ENTRY TO DISK IOCS
	00DE		236+	\$KE090 EQU	\$\$ZERO+X'00DE'	ADDR OF DKDISK ERR-PEND EXIT
	01D5		237+	\$KE130 EQU	\$\$ZERO+X'01D5'	ADDR OF DKDISK HARD ERROR EXIT
0345			239+	ORG	X'0345'	*
	0345		240+	\$ERLOG EQU	*	ADDR OF ENTRY TO LOG I/O ERRORS
	0363		241+	\$ER050 EQU	\$\$ZERO+X'0363'	START OF DISK OPS IN NERLOG
			243+		*****	
			244+		COMMUNICATION AREA REFERENCING NUCLEUS *	
			245+		*****	
			246+			
03C0			247+	ORG	X'03C0'	*
	03C0		248+	\$NUCBS EQU	*	START OF COMMUNICATION AREA
	03C0		249+	\$RMRGN EQU	\$NUCBS	ADDR OF BYTE CONTAINING THE
			250+			* SOFTWARE RIGHT MARGIN VALUE
	03C1		251+	\$LMRGN EQU	\$RMRGN+1	ADDR OF BYTE CONTAINING THE
			252+			* SOFTWARE LEFT MARGIN VALUE
	03C2		253+	\$PRPOS EQU	\$LMRGN+1	ADDR OF BYTE CONTAINING CURRENT
			254+			* POSITION OF MATRIX PRINTER
			255+			* HEAD
	03C3		256+	\$KEYCD EQU	\$PRPOS+1	ADDR OF BYTE CONTAINING KEYBOARD
			257+			* INDICATORS. A LIST OF THE
			258+			* INDICATORS AND MASKS FOLLOW
	0001		259+	\$CARDI EQU	X'01'	INPUT SOURCE INDR MASK
			260+			* 0 - KEYBOARD INPUT
			261+			* 1 - CARD OR PROC INPUT
	0002		262+	\$IOYES EQU	X'02'	I/O ROUTINES IN CORE INDR MASK
			263+			* 0 - I/O ROUTINES NOT IN CORE
			264+			* 1 - I/O ROUTINES IN CORE
	0004		265+	\$NOLST EQU	X'04'	NO LIST INDR MASK
			266+			* 0 - LISTING REQUIRED
			267+			* 1 - NO LISTING RESIRED
	0008		268+	\$GUFIR EQU	X'08'	GUFUDI ABORT INDR
			269+			* 1 - GUFUDI INTERRUPT, NOT ABOR
			270+			* 0 - GUFUDI ABORTED
			271+			* FOR THE ABOVE INDICATOR TO BE
			272+			* VALID, \$INTRP MUST BE PRESENT
	0010		273+	\$KYBSY EQU	X'10'	KEYBOARD BUSY INDR
			274+			* 0 - LINE FINISHED
			275+			* 1 - LINE NOT YET COMPLETE
	0020		276+	\$INRPT EQU	X'20'	INTERRUPT INDR
			277+			* 0 - PROGRAM NOT ABORTED
			278+			* 1 - PROGRAM ABOPRTED
	0040		279+	\$DTNMB EQU	X'40'	* 1 - AUTOMATIC LINE NUMBERS
			280+			* GENERATED FOR CARD INPUT
	0080		281+	\$TRUNK EQU	X'80'	TRUNCATED LINE INDR
			282+			* 1 - LAST LINE TRUNCATED



@FXDEQ - FIXED ADDRESSES FOR SYSTEM NUCLEUS

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 25/09/15 PAGE 10
		285+	*****		
		286+*		REGISTER SAVE AREAS. THESE AREAS ARE AVAILABLE FOR	*
		287+*		TEMPORARELY USE BY ANY PROGRAM	*
		288+	*****		
	03C5	290+\$BRS	AV EQU	\$KEYCD+2	ADDR OF 2 BYTE BASE REG SAVE
	03C7	291+\$XRS	AV EQU	\$BRS+2	ADDR OF 2 BYTE XR SAVE AREA
	03CB	293+\$TAB	LN EQU	\$XRS+4	CURRENT AUTOMATIC LINE NUMBER
		294+*			* TO BE INSERTED IF TAB KEY
		295+*			* PRESSED. (ADDR OF LINE NO.)
	03CD	296+\$CA	ERR EQU	\$TABLN+2	ADDR OF ERROR CODE SAVED FOR
		297+*			* INTERFACE WITH ERRPGM
	03CF	298+\$IN	LNO EQU	\$CAERR+2	ADDR OF EXECUTION TIME LINE
		299+*			* NUMBER FOR INTERPRETER
	03CE	300+\$E	RRPG EQU	\$INLNO-1	ADDR OF INDICATOR BYTE IF
		301+*			* SPECIAL FUNCTION REQUESTED
		302+*			* OF ERROR PROGRAM
	0030	303+\$E	RSTK EQU	X'30'	TO BE MOVED TO \$ERRPG IF A STACK
		304+*			* OF ERROR CODES IS TO BE PROCES
	0035	305+\$E	RSFL EQU	X'35'	SYNTAX CHECKERS \$ERRPG SETTING
	0040	306+\$E	RFIL EQU	X'40'	TO BE MOVED TO \$ERRPG IF FILE
		307+*			* LINE ERROR OCCURS
	0050	308+\$E	R1N2 EQU	X'50'	TO BE MOVED TO \$ERRPG IF LEVEL
		309+*			* 1 AND 2 MESSAGES REQUIRED
	0080	310+\$E	RKEY EQU	X'80'	STANDARD ERROR SETTING USED BY
		311+*			* COMMAND ANALYZER ONLY
	03CF	312+\$E	RRCT EQU	\$INLNO	ADDR OF COUNT BYTE FOR STACK
		313+*			* OF ERROR MESSAGES

@FXDEQ - FIXED ADDRESSES FOR SYSTEM NUCLEUS

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00 25/09/15 PAGE 11
			315+	*****	*****	
			316+	*	SYSTEM STATUS EQUATES	*
			317+	*****	*****	
			318+	*		
		03D0	319+	\$XIND1 EQU	\$INLNO+1	ADDR OF PRIMARY EXEC MODE INDRS
			320+	*		* ENTRIES FOLLOW
		0001	321+	\$RUNIT EQU	X'01'	1 - EXECUTE IN RUN MODE
		0002	322+	\$STEPT EQU	X'02'	1 - EXECUTE IN STEP MODE
		0004	323+	\$TRACE EQU	X'04'	1 - EXECUTE IN TRACE MODE
			324+	*		THE THREE MODE INDICATORS ARE
			325+	*		MUTUALLY EXCLUSIVE. IF \$TRACE
			326+	*		IS ON, AT LEAST 1 OF THE TRACE
			327+	*		TYPE CODE MUST ALSO BE ON.
		0008	328+	\$TFLOW EQU	X'08'	1 - TRACE FLOW
		0010	329+	\$TRALL EQU	X'10'	1 - TRACE ALL
		0020	330+	\$TRVAR EQU	X'20'	1 - TRACE SELECTED VARIABLES
		0040	331+	\$XPREC EQU	X'40'	EXECUTION PRECISION INDR
			332+	*		* 0 - SHORT PRECISION
			333+	*		* 1 - LONG PRECISION
		0080	334+	\$VMDEF EQU	X'80'	VM USAGE INDR
			335+	*		* 1 - VIRTUAL MEMORY NOT EMPTY
			336+	*		* 0 - VIRTUAL MEMORY EMPTY
		03D1	338+	\$XIND2 EQU	\$XIND1+1	ADDR OF EXECUTION INDICATORS
			339+	*		* MASK AND INDRS FOLLOW
		0001	340+	\$EXCMD EQU	X'01'	EXECUTION INDR
			341+	*		* 1 - IN EXECUTION
		0002	342+	\$PAUSE EQU	X'02'	* 1 - PROGRAM IN PAUSE STATE
		0004	343+	\$PSTEP EQU	X'04'	* 1 - PAUSE CAUSED BY STEP MODE
		0008	344+	\$PSTMT EQU	X'08'	* 1 - PAUSE CAUSED BY PAUSE STMT
		0010	345+	\$ABORT EQU	X'10'	* 1 - ABORT EXECUTION
		03D2	347+	\$IOIND EQU	\$XIND2+1	I/O STATUS INDICATORS
			348+	*		* MASKS AND EXPLANATION FOLLOW
		0001	349+	\$MPDWN EQU	X'01'	MP STATE
			350+	*		* 0 - MATRIX PRINTER OPERATIONAL
			351+	*		* 1 - MATRIX PRINTER DOWN
		0002	352+	\$CRTAV EQU	X'02'	CRT AVAILABILITY
			353+	*		* 0 - NO CRT ON SYSTEM
			354+	*		* 1 - CRT ON THE SYSTEM
		0004	355+	\$CRTNO EQU	X'04'	SYSRNT ON CRT
			356+	*		* 0 - CRT NOT AVAIL FOR SYSRNT
			357+	*		* 1 - CRT MAY BE USED FOR SYSRNT
		0008	358+	\$CMDKY EQU	X'08'	KEYBOARD MODE
			359+	*		* 0 - NORMAL KEYBOARD INPUT
			360+	*		* 1 - COMMAND KEYS USE ONLY
		0010	361+	\$PGMST EQU	X'10'	PGM START KEY
			362+	*		* 0 - MAY BE USED FOR AUTO LINE
			363+	*		* 1 - NOT USED FOR AUTO LINE #
		0020	364+	\$HRDER EQU	X'20'	HARD ERROR INDICATOR
			365+	*		* 0 - SOFT ERROR
			366+	*		* 1 - HARD ERROR
		0040	367+	\$DTRDR EQU	X'40'	DATA RECORDER
			368+	*		* 0 - DATA RECORDER NOT ON SYSTE
			369+	*		* 1 - DATA RECORDER IS ON SYSTEM
		0080	370+	\$LNPTR EQU	X'80'	MP OPTION

@FXDEQ - FIXED ADDRESSES FOR SYSTEM NUCLEUS

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	25/09/15	PAGE 12
			371+*					* 1 - 50 LPM OPTION AVAILABLE
	03D3	373+	\$CRTIN EQU	\$IOIND+1				CRT COMMAND INDICATORS
		374+*						* MASKS AND EXPLANATION FOLLOW
	0001	375+	\$CRTUP EQU	X'01'				1 - CRT IN ROLL UP MODE
	0002	376+	\$CRTDN EQU	X'02'				1 - CRT IN ROLL DOWN MODE
	0004	377+	\$CRTPU EQU	X'04'				1 - POP UP CONDITION REQUESTED
	0008	378+	\$CRTSP EQU	X'08'				1 - ROLL STOP REQUESTED
	03D4	380+	\$INDR1 EQU	\$CRTIN+1				WORK FILE STATUS INDICATORS
		381+*						* MASKS AND EXPLANATION FOLLOW
	0001	382+	\$PROCI EQU	X'01'				PROCEDURE FILE INDR
		383+*						* 0 - NOT A PROCEDURE
		384+*						* 1 - A PROCEDURE
	0002	385+	\$PRESN EQU	X'02'				WORK FILE PRECISION INDR
		386+*						* 0 - SHORT PRECISION USED
		387+*						* 1 - LONG PRECISION BEING USED
	0004	388+	\$WSIND EQU	X'04'				WORKING STORAGE INDR MASK
		389+*						* 0 - WORKING STOR ON DISK IS EM
		390+*						* 1 - WORKING STORAGE IS NOT EMP
	0008	391+	\$WFLOK EQU	X'08'				WORK FILE LOCK INDR
		392+*						* 0 - FILE NOT PROTECTED
		393+*						* 1 - FILE PROTECTED
	0010	394+	\$FITIN EQU	X'10'				FIT SECTORS INDR MASK
		395+*						* 0 - FIT SECTORS NOT PRESENT
		396+*						* 1 - FIT SECTORS IN CORE
	0020	397+	\$PGMDT EQU	X'20'				PGM DATA FILE INDR
		398+*						* 1 - PROGRAM GENERATED
		399+*						* DATA FILE IN WORK FILE
	0040	400+	\$KEYDT EQU	X'40'				KEYBOARD OR CARD FILE INDR
		401+*						* 1 - KYBRD OR CARD GENERATED
		402+*						* DATA FILE IN WORK FILE
	0080	403+	\$BASIC EQU	X'80'				BASIC PROGRAM INDR
		404+*						* 1 - BASIC PGM IN WORK FILE
	03D5	406+	\$INDR2 EQU	\$INDR1+1				ADDR OF SYSTEM 1-BIT INDRS
		407+*						* MASKS AND EXPLANATION FOLLOW
	0002	408+	\$CMODE EQU	X'02'				CONVERSATIONAL MODE INDR MASK
		409+*						* 0 - UTILITY MODE
		410+*						* 1 - CONVERSATIONAL MODE
	0004	411+	\$ERPND EQU	X'04'				ERROR LOG PENDING INDR
		412+*						* 0 - NO LOGGING REQUIRED
		413+*						* 1 - ERROR LOGGING PENDING
	0008	414+	\$DKERR EQU	X'08'				DISK ERROR INDR
		415+*						* 0 - ERROR WAS NOT DISK
		416+*						* 1 - ERROR WAS DISK, 2 ENTRIES
		417+*						* REQUIRED IN HISTORY LOG
	0010	418+	\$FCIND EQU	X'10'				CRUSH INDR MASK
		419+*						* 1 - SINGLE LINE NO DELETION
		420+*						* THROUGH THE CMD ANALYZER REQUI
		421+*						* IF \$FUIND, \$FCIND AND \$FDIND A
		422+*						* ALL ZERO, CRUCHING OP REQUIRED
	0020	423+	\$FUIND EQU	X'20'				LINE PASSED INDR MASK
		424+*						* 1 - LINE PASSED
	0040	425+	\$FDIND EQU	X'40'				LINE NUMBER LIST
		426+*						* 1 - LINE NO LIST IS DELETED

@FXDEQ - FIXED ADDRESSES FOR SYSTEM NUCLEUS

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00 25/09/15 PAGE 13
		0080	427+\$READY EQU	X'80'	PRINT READY INDR	
			428+*		* 0 - READY WILL BE PRINTED	
			429+*		* 1 - READY WON'T BE PRINTED	
		03D6	431+\$INDR3 EQU	\$INDR2+1	ADDR OF SYSTEM 1-BIT INDRS	
			432+*		* MASKS AND EXPLANATION FOLLOW	
		0001	433+\$DBLOK EQU	X'01'	SAVE PROTECTED WORK FILE MASK	
			434+*		* 1 - FILE MAY BE SAVED TO \$\$LIB	
		0002	435+\$LIST EQU	X'02'	KLISTN INDR	
			436+*		* 0 - IGNORE ROLL DOWN KEY	
			437+*		* 1 - EXCEPT ROLL DOWN KEY	
		0004	438+\$ERHRD EQU	X'04'	ERRPGM HARD ERROR INDR	
			439+*		* 1 - ERRPGM WILL EXECUTE HARD	
			440+*		* HALT AFTER PRINTING MSG	
		0008	441+\$NOENB EQU	X'08'	KEYBOARD ENABLE INDR	
			442+*		* 0 - KEYBOARD NOT ENABLED -	
			443+*		* GUFUDI WILL ENABLE	
			444+*		* 1 - KEYBOARD HAS ALREADY	
			445+*		* BEEN ENABLED	
		0010	446+\$CLBFR EQU	X'10'	CLEAR INPUT LINE BUFFER INDR	
			447+*		* 0 - DON'T CLEAR LINE BUFFER	
			448+*		* 1 - CLEAR THE INPUT LINE BUFF	
		0020	449+\$MOUNT EQU	X'20'	MOUNT KEYBOARD INDR MASK	
			450+*		* 1 - ONLY MOUNT COMMAND VALID	
		0040	451+\$NWRKR EQU	X'40'	REMOVABLE DISK WORK AREA INDR	
			452+*		* 0 - CORRECT WORK AREA ON R1	
			453+*		* 1 - NO WORK AREA ON R1	
		0080	454+\$NWRKF EQU	X'80'	FIXED DISK WORK AREA INDR	
			455+*		* 0 - CORRECT WORK AREA ON F1	
			456+*		* 1 - NO WORK AREA ON F1	
		03D7	458+\$DKSIZ EQU	\$INDR3+1	ADDR OF DISK SIZE INDR	
			459+*		* MASKS AND EXPLANATION FOLLOW	
		0001	460+\$DK100 EQU	X'01'	1 - SYSTEM HAS 100 CYLS	
		0002	461+\$DK200 EQU	X'02'	1 - SYSTEM HAS 200 CYLS	
		0004	462+\$DK400 EQU	X'04'	1 - SYSTEM HAS 400 CYLS	
		0008	463+\$DK600 EQU	X'08'	1 - SYSTEM HAS 600 CYLS	
		0010	464+\$DK800 EQU	X'10'	1 - SYSTEM HAS 800 CYLS	
		03D8	466+\$XIND3 EQU	\$DKSIZ+1	PAST \$XIND1	
			467+*		* SEE \$XIND1 FOR INDR MASKS	
		03DA	469+\$FILIB EQU	\$XIND3+2	ADDR OF CURRENT FILE LIB DADDR	
		03DC	470+\$USRDR EQU	\$FILIB+2	ADDR OF REL DISP TO 1ST USER BK	
		03DD	471+\$CONFIG EQU	\$USRDR+1	CONFIGURATION INDRS	
		0001	472+\$22IMP EQU	X'01'	0 - 13 INCH MATRIX PRINTER	
			473+*		1 - 22 INCH MATRIX PRINTER	
		0002	474+\$16K EQU	X'02'	1 - CPU HAS 12 KBYTE	
		0004	475+\$12K EQU	X'04'	1 - CPU HAS 16 KBYTE	
			476+*		* IF BOTH OFF: CPU HAS 8 KBYTE	
		0008	477+\$16CKY EQU	X'08'	0 - KEYBOARD HAS 8 CMD KEYS	
			478+*		1 - KEYBOARD HAS 16 CMD KEYS	
		0080	479+\$BIGCD EQU	X'80'	1 - CPU HAS 129 DATA RECORDER	
		03DF	481+\$LEVEL EQU	\$CONFIG+2	ADDR OF SYSTEM LEVEL NUMBER	

@FXDEQ - FIXED ADDRESSES FOR SYSTEM NUCLEUS

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00 25/09/15 PAGE 14
		03E0	483+\$DBGUF	EQU	\$LEVEL+1	ADDR OF GUFUDI DEBUG INDR
		0080	484+\$CRUSH	EQU	X'80'	0 - CRUSH THE FILE
		0040	485+\$REORD	EQU	X'40'	0 - REORDER THE FILE
		0020	486+\$IRKEY	EQU	X'20'	1 - ENABLE KEYBOARD INPUT
		0010	487+\$IOPGS	EQU	X'10'	D1 PAGES INDR: 0 - ONE
		0008	488+\$CALLI	EQU	X'08'	PROCEDURE CALL INDR
			489+*			* 0 - NOT A CALL
			490+*			* 1 - A CALL
		03E1	492+\$KEYBD	EQU	\$DBGUF+1	KEYBOARD TYPE INDR
			493+*			* THIS VALUE WILL BE A BINARY
			494+*			* VALUE FROM 1 TO 12 INDICATING
			495+*			* WHICH DATA TABLE IS IN USE
		03E2	497+\$CRPOS	EQU	\$KEYBD+1	ADDR OF CURRENT CURSOR POSITION
		03E3	498+\$BUFPT	EQU	\$CRPOS+1	LINE PRINTER BUFFER POINTER 1-3
		03E4	499+\$LPRP3	EQU	\$BUFPT+1	LINE PRINTER FLAGS 1-3
		03E5	500+\$LPROS	EQU	\$LPRP3+1	TRUE LINE PRINTER PRINT POS. 1-3
		03E6	502+\$NEXTB	EQU	\$LPROS+1	REL DADDR PROCEDURE CALL 1-4
		03E7	503+\$NEXTL	EQU	\$NEXTB+1	DISPLACEMENT WITHIN DB 1-4
		03E8	504+\$DFDET	EQU	\$NEXTL+1	GRAPRO INTERNAL INDR 1-4
		03EA	505+\$LPRIO	EQU	\$DFDET+2	LINE PRINTER BUF INC. + PDAR 1-4
		03F5	507+\$PTCH1	EQU	\$DKSIZ+30	LAST BYTE OF NUCLUES AREA
			508+*****			*****
			509+*		TABLES AND SYSTEM WORK AREAS	*
			510+*****			*****
		03F6	511+\$VOLID	EQU	\$PTCH1+1	ADDR OF LEFT BYTE VOLID TABLE
		03F6	512+\$VOLR1	EQU	\$VOLID	ADDR LEFT BYTE VOLID FOR R1
		03FE	513+\$VOLF1	EQU	\$VOLR1+8	ADDR LEFT BYTE VOLID FOR F1
		0406	514+\$VOLR2	EQU	\$VOLF1+8	ADDR LEFT BYTE VOLID FOR R2
		040E	515+\$VOLF2	EQU	\$VOLR2+8	ADDR LEFT BYTE VOLID FOR F2
		0419	516+\$PKERT	EQU	\$VOLID+35	ADDR OF 1ST ENTRY IN PACK ERROR
			517+*			* RATE TABLE
		042D	518+\$PASWD	EQU	\$PKERT+20	ADDR OF CURRENT PASSWORD
		042E	519+\$HISTE	EQU	\$PASWD+1	LEFT BYTE OF HISTORY LOG ENTRY
		0435	520+\$HIST1	EQU	\$HISTE+7	ADDR OF 1ST ENTRY OF HIST LOG
		043A	521+\$DATE	EQU	\$HIST1+5	ADDR OF CURRENT DATE
		043B	522+\$EXFTR	EQU	\$DATE+1	ADDR OF CORE EXPANSION FACTOR
			523+*			* THIS VALUE WILL BE ADDED TO
			524+*			* BUFFER ADDRESS (SET FOR 8K)
			525+*			* TO RE-POSITION THEM FOR
			526+*			* LARGER MACHINES
		0443	527+\$WFNME	EQU	\$EXFTR+8	ADDR OF WORK FILE NAME
		0040	528+\$WFDEF	EQU	X'40'	WORK FILE DEFINED INDR
			529+*			* THIS MASK IS USED ON \$WFNME
			530+*			* 0 - WORK FILE UNDEFINED
			531+*			* 1 - WORK FILE DEFINED
		0449	532+\$DPLSV	EQU	\$WFNME+6	ADDR OF 6 BYTE DPL SAVE AREA
			533+*			* FOR KEYBOARD PROGRAMS
		044B	534+\$PRDEV	EQU	\$DPLSV+2	ADDR OF 2 BYTE FIELD POINTING
			535+*			* TO THE SYSTEM PRINTER IOCR
		044D	536+\$CRTAD	EQU	\$PRDEV+2	ADDR OF ENTRY TO RELOCATE CRT
		0454	537+\$PLST1	EQU	\$CRTAD+7	ADDR OF THREE 7-BYTES ENTRY I/O
		045B	538+\$PLST2	EQU	\$PLST1+7	* PARM LISTS MOST RECENTLY USED



[illegible]



@FXDEQ - FIXED ADDRESSES FOR SYSTEM NUCLEUS

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 25/09/15 PAGE 16
		542+	*****		
		543+	*	ENTRY POINTS TO INTERFACE ROUTINES AND THEIR WORK AREAS	*
		544+	*****		
		0465	546+	\$SPRNT EQU \$C0001+1	ADDR OF ENTRY TO THE SYSTEM
			547+	*	* PRINTER IOCR
		0469	548+	\$CAERK EQU \$SPRNT+4	ADDR OF ENTRY TO ERR ROUTINE
			549+	*	* INTERFACE. ERROR CODE MUST
			550+	*	* BE STORED PREVIOUS TO ENTRY
		046F	551+	\$ERDPL EQU \$CAERK+6	ADDR OF LEFT BYTE OF ERRPGM
			552+	*	* LOAD DPL
		0472	553+	\$ERMAD EQU \$ERDPL+3	ADDR OF DK ADDR, CNT OF ERRPGM
		0476	554+	\$CIMSK EQU \$ERMAD+4	ADDR OF THE INQUIRY REQUEST INDR
			555+	*	* X'87' IR NOT DISABLED
			556+	*	* X'80' IR MASKED
		0480	557+	\$CIEXT EQU \$CIMSK+10	ADDR OF IR EXIT INSTRUCTION
		0483	558+	\$CIENT EQU \$CIEXT+3	ADDR OF ENTRY FOR IR
		048D	559+	\$UNMSK EQU \$CIENT+10	ADDR OF ENTRY TO UNMASK IR
			560+	*	* IF NO SUSPENDED IR, CALLING
			561+	*	* PROGRAM RETURNED TO
		0496	562+	\$CISUS EQU \$UNMSK+9	ADDR OF INDR FOR SUSPENDED IR
			563+	*	* IF X'80' AN IR OCCURRED WHILE
			564+	*	* IR WAS MASKED
			565+	*	* IF X'87' NO IR TOOK PLACE
			566+	*	* WHILE IR WAS MASKED
		049D	567+	\$CAIPL EQU \$CISUS+7	ADDR OF ENTRY TO ABORT CURRENT
			568+	*	* OP AND RE-ENABLE KEYBOARD AND
		04A1	569+	\$CARPL EQU \$CAIPL+4	ADDR OF ENTRY TO ABORT CURRENT
			570+	*	* OP AND ENABLE IR
		04B4	571+	\$CABLD EQU \$CARPL+X'13'	ADDR OF ENTRY TO ABORT CURRENT O
		04BA	572+	\$PAUSD EQU \$CABLD+6	ADDR OF ENTRY OF ROUTINE TO
			573+	*	* SWAP CORE
		04D6	574+	\$RSTR EQU \$PAUSD+X'1C'	ADDR OF ENTRY TO ENTRY CORE
			575+	*	* FROM DISK
		04F2	576+	\$PSDXR EQU \$RSTR+X'1C'	ADDR OF SAVED XR IN NPAUSE
		04FA	577+	\$PSDBR EQU \$PSDXR+8	ADDR OF SAVED BR IN NPAUSE
		04FE	578+	\$SRTRN EQU \$RSTR+X'28'	ADDR OF RETURN ADDR FROM \$PAUSD
		050D	579+	\$SFAID EQU \$SRTRN+15	ADDR OF RETURN IF FE AID REQUEST
			580+	*	* IF THE ABOVE TWO ADDRESSES ARE
			581+	*	* EQUAL, RETURN TO \$RSTR WILL BE
			582+	*	* BE FROM THE FE AID PROGRAM
		050E	583+	\$CSDPL EQU \$RSTR+X'38'	ADDR OF LEFT BYTE OF SAVE/RSTR D
		0511	584+	\$SWPCR EQU \$CSDPL+3	ADDR OF DKADDR, COUNT FOR CORE
			585+	*	* SAVE AREA
		0517	586+	\$EXADR EQU \$SWPCR+6	ADDRR OF DK ADDR, COUNT OF EXEC
			587+	*	* TIME MESSAGE PROGRAM
		051A	588+	\$LOADR EQU \$EXADR+3	ADDR OF ENTRY TO BLAST LOAD
			589+	*	* PROGRAM NOT RESIDING ON CYL 4
			590+	*	* RETURN IS TO CALLING PROGRAM
		051E	591+	\$RLOAD EQU \$LOADR+4	ADDR OF ENTRY TO BLAST LOAD
			592+	*	* PROGRAM NOT RESIDING ON CYL 4
		0522	593+	\$BLOAD EQU \$RLOAD+4	ADDR OF ENTRY TO BLAST LOAD
			594+	*	* PROGRAM RESIDING ON CYL 4
		054A	595+	\$LOADB EQU \$BLOAD+X'28'	ADDR OF SPECIAL ENTRY TO
			596+	*	* NBLOAD FOR SFLOAD/SFFIND
			597+	*	* AND FZPINV

@FXDEQ - FIXED ADDRESSES FOR SYSTEM NUCLEUS

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00 25/09/15 PAGE 17
		054E	598+	\$TROVR EQU	\$BLOAD+X'2C'	ADDR OF FE TRACE INDR
			599+	*		* @NOP - NO TRACE PERFORMED
			600+	*		* @UCB - TRACE PERFORMED
		0550	601+	\$BLRTN EQU	\$TROVR+2	ADDR OF RETURN POINT FROM ZTRACE
		0569	602+	\$BLNOE EQU	\$BLRTN+X'19'	ADDR OF NO EXECUTE INDR-NBLOAD
			603+	*		* @NOP - CALLING PGM RETURNED TO
			604+	*		* @UCB - LOADED PROGRAM EXECUTED
			605+	*		* ENTRY TO \$LOADR SETS THE ABOVE
			606+	*		* INDR TO @NOP. IF THE CALLING
			607+	*		* SETS THE INDR TO @NOP BEFORE
			608+	*		* CALLING \$BLOAD, RETURN WILL BE
			609+	*		* MADE UPON COMPLETION OF THE
			610+	*		* ABSOLUE LOAD
		0571	611+	\$LDRTN EQU	\$BLOAD+X'4F'	ADDR OF THE RETURN ADDR IN NBLOA
		0579	612+	\$BLDPL EQU	\$BLOAD+X'57'	ADDR OF LEFT BYTE OF \$BLOAD'S
			613+	*		* DPL (DPL OF LAST PGM LOADED)
		057F	614+	\$WAITF EQU	\$BLDPL+6	ADDR OF LEFT BYTE OF DISK
			615+	*		* WAIT AND CHECK ERRORS DPL
		0583	616+	\$GUFIO EQU	\$WAITF+4	ADDR OF DK ADDR, COUNT OF GUFUDI
		0587	617+	\$BSADR EQU	\$GUFIO+4	ADDR OF DADDR RELOCATION FACTOR
			618+	*		* FOR PGMS NOT RESIDING ON CYL 6
		0588	619+	\$FEMAP EQU	\$BSADR+1	ADDR OF START OF CORE MAP
		05A2	620+	\$ZTRAD EQU	\$FEMAP+X'1A'	ADDR OF ZTRACE DADDR
05FF			622+	ORG	X'05FF'	
		05FF	623+	\$IPLDV EQU	*	ADDR OF IPL INDR
			624+	*		* X'00' - IPL WAS FROM R1
			625+	*		* X'01' - IPL WAS FROM F1
		0600	626+	\$ENDNU EQU	\$IPLDV+1	ADDR OF THE FIRST BYTE
			627+	*		* FOLLOWING SYSNUC
			628+	*		
			629+	*		
			630	*	@HDW EXP-Y	
			632+	*	PRINT ON	

@HDWEQ - SYSTEM HARDWARE I/O EQUATES

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00 25/09/15 PAGE 18
				634+	*****			
				635+	*	DISK HARDWARE EQUATES	*	
				636+	*****			
				637+	*			
				638+	***	DISK CONTROL FIELD EQUATES		
				639+	*			
	0000			640+	@PFLAG	EQU	0	F-BYTE
	0001			641+	@PCYL	EQU	1	C-BYTE
	0002			642+	@PSAD	EQU	2	S-BYTE
	0003			643+	@PCNT	EQU	3	N-BYTE
				644+	*			
	0004			645+	@DCFLN	EQU	4	LENGTH OF DISK CTRL FIELD
	0001			646+	@DCYMV	EQU	X'01'	DIRECTION BIT IN SEEK S-BYTE
				647+	*			
	0006			648+	@DFCR	EQU	6	DFCR Q-CODE FOR LIO
	0004			649+	@DFDR	EQU	4	DFDR Q-CODE FOR LIO
				650+	*			
	0000			651+	@DSEEK	EQU	X'00'	SIO Q-CODE SEEK FUNCTION
	0001			652+	@DREAD	EQU	X'01'	SIO Q-CODE READ FUNCTION
	0002			653+	@DWRTIT	EQU	X'02'	SIO Q-CODE WRITE FUNCTION
				654+	*			
	0001			655+	@DCWID	EQU	X'01'	CTRL BYTE FOR SIO WRITE ID
	0000			656+	@SKCTL	EQU	X'00'	CTRL BYTE FOR SIO SEEK
	0003			657+	@DVERY	EQU	X'03'	CTRL BYTE FOR SIO VERIFY
	0000			658+	@DCTRW	EQU	X'00'	SIO CTRL FOR READ/WRITE DATA
	0001			659+	@DCRID	EQU	X'01'	SIO CTRL FOR READ ID
				660+	*			
	0002			661+	@DBUSY	EQU	2	CONDITION CODE FOR DISK BUSY
	0000			662+	@DERR	EQU	0	CONDITION CODE FOR DISK ERROR
	0002			663+	@DVST1	EQU	X'02'	SNS I/O CODE FOR BYTES 0,1
	0003			664+	@DVST2	EQU	X'03'	SNS I/O CODE FOR BYTES 2,3
	00A0			665+	@SPINA	EQU	X'A0'	DEV CODE ADDR DISK SPINDLE A
	00B0			666+	@SPINB	EQU	X'B0'	DEV CODE ADDR DISK SPINDLE B
	0001			667+	@ALTFL	EQU	1	ALTERNATE TRACK FLAG BYTE
	0002			668+	@DEFLG	EQU	2	DEFECTIVE TRACK FLAG BYTE
	0000			669+	@NORFL	EQU	0	NORMAL TRACK FLAG BYTE
	0001			670+	@HSTQR	EQU	1	Q+R BYTE ENTRIES IN HISTORY LOG
	0005			671+	@HSTSN	EQU	5	SENSE BYTE ENTRY IN HISTORY LOG
	0006			672+	@HSTPE	EQU	6	ERROR TYPE ENTRY IN HISTORY LOG
	0007			673+	@HSTEN	EQU	7	END OF 1ST ENTRY IN HISTORY LOG
	0009			674+	@HSTAD	EQU	9	DISK ADDR ENTRY IN HISTORY LOG
	000F			675+	@HSTVI	EQU	15	VOL-ID ENTRY IN HISTORY LOG
	0000			676+	@DHARD	EQU	0	HARD ERR INDR MASK FOR @ HSTPE
				677+	*			
				678+	***	DISK ERROR STATUS BITS		
				679+	*			
	0000			680+	@SNSB0	EQU	0	SENSE BYTE 0 DISPLACEMENT
	0001			681+	@SNSB1	EQU	1	SENSE BYTE 1 DISPLACEMENT
	0002			682+	@SNSB2	EQU	2	SENSE BYTE 2 DISPLACEMENT
	0003			683+	@SNSB3	EQU	3	SENSE BYTE 3 DISPLACEMENT
				684+	*			
				685+	***	BYTE 0		
				686+	*			
	0040			687+	@DERIN	EQU	X'40'	INTERVENTION REQUIRED
	0020			688+	@DERMA	EQU	X'20'	MISSING ADDR MARK
	0010			689+	@DEREQ	EQU	X'10'	EQUIPMENT CHECK

@HDWEQ - SYSTEM HARDWARE I/O EQUATES

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00 25/09/15 PAGE 19
		0008	690+	@DERD2 EQU	X'08'	DATA CHECK
		0004	691+	@DERNR EQU	X'04'	NO RECORD FOUND
		0002	692+	@DERTC EQU	X'02'	TRACK CONDITION CHECK
		0001	693+	@DERSC EQU	X'01'	SEEK CHECK
		0080	694+	@DUNSF EQU	X'80'	UNSAFE CONDITION MASK - BYTE 2
			695+	*		
			696+	***	BYTE 1	
			697+	*		
		0020	698+	@DERCE EQU	X'20'	END OF CYLINDER
		0004	699+	@OVRUN EQU	X'04'	OVERRUN
			701+	*****		
			702+	*	MATRIX PRINTER I/O EQUATES	*
			703+	*****		
		0004	704+	@PLNGH EQU	4	LENGTH OF PCF
		0002	705+	@SYCNT EQU	2	DISP OF CNT IN SYNC CK PCF
		0003	706+	@RTCNT EQU	3	RETURN CNT BYTE IN PCF
		00E4	707+	@PDAR EQU	X'E4'	DATA LSR FOR MP
		00E6	708+	@PCAR EQU	X'E6'	CONTROL LSR FOR MP
		0000	709+	@PSIOR EQU	X'00'	SIO CTRL CODE FOR MP
		00E0	710+	@PSIOQ EQU	X'E0'	SIO Q-CODE FOR MP
		00E2	711+	@PBUSY EQU	X'E2'	TIO BUSY CODE
		00E1	712+	@PFORM EQU	X'E1'	TIO FORMS CHECK CODE
		00E2	713+	@PLITE EQU	X'E2'	LIO INDR LIGHT CODE
		00E0	714+	@PERR EQU	X'E0'	TIO ERROR CHECK CODE
		0020	715+	@PMGCK EQU	X'20'	MARGIN CHECK BIT
		00E2	716+	@PSNSQ EQU	X'E2'	MP SENSE I/O Q-CODE
			718+	*****		
			719+	*	KEYBOARD EQUATES FOR DEPRES	*
			720+	*****		
		001E	721+	@KENAB EQU	X'1E'	ENABLE, UNLOCK KEYBOARD CTRL
		001F	722+	@KEXIT EQU	X'1F'	RESTORE ENABLE KEYBOARD EXIT CTR
		001B	723+	@KELOK EQU	X'1B'	LOCK, EXIT, DISABLE CTRL
		0020	724+	@KCMDK EQU	X'20'	COMMAND KEY MASK
		0001	725+	@CKY01 EQU	1	COMMAND KEY 1
		0002	726+	@CKY02 EQU	2	COMMAND KEY 2
		0003	727+	@CKY03 EQU	3	COMMAND KEY 3
		0004	728+	@CKY04 EQU	4	COMMAND KEY 4
		0005	729+	@CKY05 EQU	5	COMMAND KEY 5
		0006	730+	@CKY06 EQU	6	COMMAND KEY 6
		0007	731+	@CKY07 EQU	7	COMMAND KEY 7
		0008	732+	@CKY08 EQU	8	COMMAND KEY 8
		0009	733+	@CKY09 EQU	9	COMMAND KEY 9
		000A	734+	@CKY10 EQU	10	COMMAND KEY 10
		000B	735+	@CKY11 EQU	11	COMMAND KEY 11
		000C	736+	@CKY12 EQU	12	COMMAND KEY 12
		000D	737+	@CKY13 EQU	13	COMMAND KEY 13
		000E	738+	@CKY14 EQU	14	COMMAND KEY 14
		000F	739+	@CKY15 EQU	15	COMMAND KEY 15
		0010	740+	@CKY16 EQU	16	COMMAND KEY 16
		0010	741+	@KEYBD EQU	X'10'	KEYBOARD Q-CODE
		0000	742+	@CMOFF EQU	X'00'	LIO M+N BYTE CMND INDRS OFF
		0001	743+	@CMLON EQU	X'01'	LIO M+N BYTE CMND INDRS ON
		0010	744+	@KFUNK EQU	X'10'	FUNCTION KEY MASK
		000D	745+	@KLEAR EQU	X'0D'	CLEAR COMMAND KEY MASK

@HDWEQ - SYSTEM HARDWARE I/O EQUATES

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	25/09/15	PAGE	20
		001C	746+	@TYPO EQU	X'1C'				SIO CTRL FOR TYPAMATIC
		0002	747+	@TYPAM EQU	X'02'				TYPAMATIC FUNCTION BIT
		0080	748+	@PRITY EQU	X'80'				PARITY ERROR BIT
		0011	749+	@KHARD EQU	X'11'				SIO CTRL FOR HARD ERROR
		0012	750+	@FLDIN EQU	X'12'				LIGHT FIELD INDR Q-BYTE
			752+	*****					
			753+	*	CRT I/O EQUATES				*
			754+	*****					
			755+	*					
		0092	756+	@CRTDS EQU	X'92'				SIO Q-BYTE
		0092	757+	@DSBSY EQU	X'92'				CRT BUSY MASK
		0090	758+	@CRTQ EQU	X'90'				LIO Q-BYTE
		0090	759+	@CRERR EQU	X'90'				CRT ERROR MASK
		0040	760+	@CURSR EQU	X'40'				CURSOR BIT
		0040	761+	@DLNLG EQU	64				LENGTH OF CRT LINE
		000F	762+	@DLNCT EQU	15				NUMBER OF LINES IN BUFFER
		0004	763+	@CRPRY EQU	X'04'				PARITY ERROR BIT
		0010	764+	@BKSPC EQU	X'10'				BACKSPACE CTRL BYTE
		0010	765+	@4K EQU	16				NUMBER OF SCTRS = 4K
			767+	*****					
			768+	*	GENERAL EQUATES FOR 3.7B CARD READER/PUNCH				*
			769+	*****					
			770+	*					
			771+	***	SIO FUNCTION CODES				
			772+	*					
		0000	773+	@CC37B EQU	X'00'				SIO CONTROL CODE
			774+	*					
			775+	***	TIO FUNCTION CODES				
			776+	*					
		00F2	777+	@BZ37B EQU	X'F2'				DEVICE BUSY CODE
		00F0	778+	@ER37B EQU	X'F0'				I/O CHECK OR NOT READY
			779+	*					
			780+	***	LIO FUNCTION CODES				
			781+	*					
		00F0	782+	@LO37B EQU	X'F0'				LOAD READ ADDESS REGISTER
			783+	*					
			784+	***	SNS FUNCTION CODES				
			785+	*					
		00F2	786+	@SN37B EQU	X'F2'				STORE ERROR STATUS BYTES
			788+	*****					
			789+	*	3.7B CARD READER EQUATES				*
			790+	*****					
		00F0	791+	@CD37B EQU	X'F0'				DEVICE ADDRESS - READER
		00F1	792+	@RD37B EQU	X'F1'				SIO READ FUNCTION
			794+	*****					
			795+	*	3.7B CARD PUNCH EQUATES				*
			796+	*****					
		00F0	797+	@PN37B EQU	X'F0'				DEVICE ADDRESS - PUNCH
		00F2	798+	@PC37B EQU	X'F2'				SIO PUNCH FUNCTION
			800+	*****					
			801+	*	ERROR FUNCTION CODES				*

@HDWEQ - SYSTEM HARDWARE I/O EQUATES

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00 25/09/15 PAGE 21
			802+	*****		
		0040	803+	@TJ37B EQU	X'40'	TRANSPORT JAM
		0004	804+	@CP37B EQU	X'04'	COMPARE ERROR
		0005	805+	@RT37B EQU	X'05'	RETRY COUNT
		00A0	806+	@NTRDY EQU	X'A0'	CARD READER NOT READY TEST
			808+	*****		
			809+	*	PPL EQUATES	*
			810+	*****		
		00FF	811+	@WA37B EQU	X'FF'	WAIT AND CHECK FOR ERRORS
		0080	812+	@PD37B EQU	X'80'	PUNCH DATA
		00C0	813+	@IP37B EQU	X'C0'	INSERT AND PUNCH DATA
		0040	814+	@ID37B EQU	X'40'	INSERT DATA
			815+	*	END OF SYSTEM HARDWARE I/O EQUATES	
			816+		PRINT ON	
			817	*	@CY0 EXP-Y	
			819+		PRINT ON	



@CY0EQ - CYLINDER ZERO EQUATES

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00 25/09/15 PAGE 22
			821+	*****	*****	
			822+	*	DISK TABLE EQUATES	*
			823+	*****	*****	
	0006		824+	#VOLNG EQU 6	LENGTH OF VOL ID	
	0005		825+	#VOLOC EQU 5	DISPLACEMENT OF VOL ID ON SCTR	
	0008		826+	#VLTBE EQU #VOLNG+2	LENGTH OF VOLID TABLE ENTRY	
			828+	*****	*****	
			829+	*	SDS (ERROR LOG) EQUATES	*
			830+	*****	*****	
	0003		831+	#PKRTD EQU 3	DISP TO END OF PK ERR/RATE ENTRY	
	0003		832+	#PKRDD EQU 3	DISP TO RESPECTIVE READ COUNTER	
	0001		833+	#PKWTD EQU 1	DISP TO RESPECTIVE WRITE COUNTER	
	0002		834+	#PKCNT EQU 2	LENGTH OF IN-CORE COUNTERS	
	002B		835+	#PKMRW EQU 43	DISP TO MASTER RD/WT COUNTERS	
	000B		836+	#PKVRD EQU 11	DISP TO VOLUME RD COUNTERS IN SD	
	0007		837+	#PKVWD EQU 7	DISP TO VOLUME WT COUNTERS IN SD	
	0004		838+	#PKRTL EQU 4	LENGTH PACK ERROR RATE ENTRY	
	0004		839+	#RDWTL EQU 4	LENGTH RD/WT ERROR RATE COUNTER	
	0001		841+	#CNDIS EQU 1	SECTOR DISPLACEMENT OF	
			842+	*	* CONFIGURATION RECORD	
			844+	*****	*****	
			845+	*	ERROR HISTORY TABLE EQUATES	*
			846+	*****	*****	
	0008		847+	#HISLN EQU 8	LENGTH OF HISTORY TABLE ENTRY	
	0002		848+	#DKEXT EQU #HISLN-#VOLNG	HIST LOG EXTENSION FOR DISK ERRO	
	0001		849+	#HSENT EQU 1	DISP OF DISP TO NEXT OBR ENTRY	
	0003		850+	#HISDX EQU 3	DISP OF DISP PAST LAST ENTRY	
	0000		851+	#HISTQ EQU 0	DISP OF SIO Q BYTE	
	0001		852+	#HISTR EQU 1	DISP OF SIO CNTL BYTE	
	0003		853+	#HISN1 EQU 3	DISP OF PRIMARY SENSE REG	
	0005		854+	#HISN2 EQU 5	DISP OF SECONDARY SENSE REG	
	0006		855+	#HISCT EQU 6	DISP OF RETRY COUNT	
	0007		856+	#HSEND EQU 7	DISP OF END OF 1ST ENTRY	
	0007		857+	#HISTC EQU 7	DISP OF DCF F-BYTE	
	0008		858+	#HISTS EQU 8	DISP OF DCF S-BYTE	
	0009		859+	#HISTN EQU 9	DISP OF DCF N-BYTE	
	000F		860+	#HISTV EQU 15	DISP OF DISK VOL-ID	
			862+	*****	*****	
			863+	*	CYLINDER ZERO DISK ADDRESSES	*
			864+	*****	*****	
	0010		865+	#CORSV EQU X'0010'	DADDR OF TEMP CORE SAVE AREA	
	0005		866+	#@CORS EQU 5	SCTR COUNT TEMP CORE SAVE AREA	
	009C		867+	#NEROV EQU X'009C'	DADDR OF NERLOG OVERLAY	
	0003		868+	#@NERO EQU 3	SCTR COUNT NERLOG OVERLAY	
	001D		869+	#OBRAD EQU X'001D'	DADDR OF OBR TABLE	
	0002		870+	#@OBRA EQU 2	SCTR COUNT OF OBR	
	000C		871+	#VLSDR EQU X'000C'	DADDR OF VOL STATISTICS SCTR R1	
	0001		872+	#@VLSD EQU 1	SCTR COUNT OF VOL STATISTICS	
	000D		873+	#MVSDR EQU X'000D'	DADDR OF MASTER VOL STAT SCTR	
	0001		874+	#@MVSD EQU 1	SCTR COUNT OF MASTER VOL STAT	
	0011		875+	#SDRDK EQU X'0011'	DADDR OF DISK SDR SCTR	
	0019		876+	#IOSDR EQU X'0019'	DADDR OF NON-DISK SDR SCTR	

@CY0EQ - CYLINDER ZERO EQUATES

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	25/09/15	PAGE	23
		0005	877+	#CNFIG	EQU	X'0005'			DADDR OF CONFIG RECORD
		0001	878+	#FIGSC	EQU	1			SCTR COUNT OF CONFIG RECORD
		0009	879+	#VOLF1	EQU	X'0009'			DADDR OF VOLUME LABEL (F1)
		0008	880+	#VOLR1	EQU	X'0008'			DADDR OF VOLUME LABEL (R1)
		0001	881+	#@VLAB	EQU	1			SCTR COUNT OF VOLUME LABEL
		0024	882+	#VTCR1	EQU	X'0024'			DADDR OF R1 VTOC
		0025	883+	#VTCF1	EQU	X'0025'			DADDR OF F1 VTOC
		0026	884+	#VTCR2	EQU	X'0026'			DADDR OF R2 VTOC
		0027	885+	#VTCF2	EQU	X'0027'			DADDR OF F2 VTOC
		0002	886+	#@VCNT	EQU	2			SCTR COUNT OF VTOC
		00DC	887+	#PTFDA	EQU	X'00DC'			DADDR OF PTF LOG
		0001	888+	#@PTFS	EQU	1			SCTR COUNT FOR PTF LOG
		0006	889+	#@PTFL	EQU	6			LENGTH OF ENTRY IN PTF LOG
		890+	*			END OF CYLINDER ZERO EQUATES			
		891+				PRINT ON			
		892	*			@SPF EXP-Y			
		894+				PRINT ON			



@SPFEQ - SYSTEM PROGRAM FILE EQUATES

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	25/09/15	PAGE 24
					896+	*****				
					897+	*	SYSTEM PROGRAM FILE (SPF) EQUATES			*
					898+	*****				
					899+	*				
				0000	900+	##\$#0TR EQU	X'0000'			DISK ADDR OF ##0TRK
				0700	901+	##\$#0T EQU	X'0700'			CORE LOAD ADDRESS OF ##0TRK
				0018	902+	##\$#@#0T EQU	24			SECTOR COUNT OF ##0TRK
					903+	*				
				0080	904+	##\$#1TR EQU	X'0080'			DISK ADDR OF ##1TRK
				0000	905+	##\$#1T EQU	X'0000'			CORE LOAD ADDRESS OF ##1TRK
				0018	906+	##\$#@#1T EQU	24			SECTOR COUNT OF ##1TRK
					907+	*				
				0000	908+	##\$#DRT EQU	X'0000'			DISK ADDR OF ##DRTY
				0000	909+	##\$#DR EQU	X'0000'			CORE LOAD ADDRESS OF ##DRTY
				0008	910+	##\$#@#DR EQU	08			SECTOR COUNT OF ##DRTY
					911+	*				
				0020	912+	##\$INST EQU	X'0020'			DISK ADDR OF #INSTD
				0600	913+	##\$INS EQU	X'0600'			CORE LOAD ADDRESS OF #INSTD
				0010	914+	##\$#@INS EQU	16			SECTOR COUNT OF #INSTD
					915+	*				
				0080	916+	##\$BCOM EQU	X'0080'			DISK ADDR OF #BCOMP
				0600	917+	##\$BCO EQU	X'0600'			CORE LOAD ADDRESS OF #BCOMP
				0018	918+	##\$#@BCO EQU	24			SECTOR COUNT OF #BCOMP
					919+	*				
				0100	920+	##\$LOAD EQU	X'0100'			DISK ADDR OF #LOADR
				0600	921+	##\$LOA EQU	X'0600'			CORE LOAD ADDRESS OF #LOADR
				0013	922+	##\$#@LOA EQU	19			SECTOR COUNT OF #LOADR
					923+	*				
				014C	924+	##\$DPRI EQU	X'014C'			DISK ADDR OF #DPRIN
				0700	925+	##\$DPR EQU	X'0700'			CORE LOAD ADDRESS OF #DPRIN
				0005	926+	##\$#@DPR EQU	05			SECTOR COUNT OF #DPRIN
					927+	*				
				0180	928+	##\$KGOS EQU	X'0180'			DISK ADDR OF #KGOSL
				0C00	929+	##\$KGO EQU	X'0C00'			CORE LOAD ADDRESS OF #KGOSL
				0002	930+	##\$#@KGO EQU	02			SECTOR COUNT OF #KGOSL
					931+	*				
				0188	932+	##\$KEDI EQU	X'0188'			DISK ADDR OF #KEDIT
				0C00	933+	##\$KED EQU	X'0C00'			CORE LOAD ADDRESS OF #KEDIT
				000E	934+	##\$#@KED EQU	14			SECTOR COUNT OF #KEDIT
					935+	*				
				01C4	936+	##\$KENA EQU	X'01C4'			DISK ADDR OF #KENAB
				0C00	937+	##\$KEN EQU	X'0C00'			CORE LOAD ADDRESS OF #KENAB
				0006	938+	##\$#@KEN EQU	06			SECTOR COUNT OF #KENAB
					939+	*				
				0200	940+	##\$DREA EQU	X'0200'			DISK ADDR OF #DREAD
				0889	941+	##\$DRE EQU	X'0889'			CORE LOAD ADDRESS OF #DREAD
				0001	942+	##\$#@DRE EQU	01			SECTOR COUNT OF #DREAD
					943+	*				
				0204	944+	##\$KMOU EQU	X'0204'			DISK ADDR OF #KMOUN
				0C00	945+	##\$KMO EQU	X'0C00'			CORE LOAD ADDRESS OF #KMOUN
				0004	946+	##\$#@KMO EQU	04			SECTOR COUNT OF #KMOUN
					947+	*				
				0214	948+	##\$KRMO EQU	X'0214'			DISK ADDR OF #KRMOV
				0C00	949+	##\$KRM EQU	X'0C00'			CORE LOAD ADDRESS OF #KRMOV
				0003	950+	##\$#@KRM EQU	03			SECTOR COUNT OF #KRMOV
					951+	*				

## @SPFEQ - SYSTEM PROGRAM FILE EQUATES

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	25/09/15	PAGE 25
		0220	952+##\$KPAS	EQU	X'0220'			
		0C00	953+##\$KPA	EQU	X'0C00'			
		0005	954+##\$@KPA	EQU	05			
			955+*					
		0234	956+##\$KEXT	EQU	X'0234'			
		0C00	957+##\$KEX	EQU	X'0C00'			
		0003	958+##\$@KEX	EQU	03			
			959+*					
		0240	960+##\$DSPL	EQU	X'0240'			
		2800	961+##\$DSP	EQU	X'2800'			
		0004	962+##\$@DSP	EQU	04			
			963+*					
		0250	964+##\$TSYK	EQU	X'0250'			
		1000	965+##\$TSY	EQU	X'1000'			
		0003	966+##\$@TSY	EQU	03			
			967+*					
		0280	968+##\$KRNU	EQU	X'0280'			
		0700	969+##\$KRN	EQU	X'0700'			
		0003	970+##\$@KRN	EQU	03			
			971+*					
		028C	972+##\$KROV	EQU	X'028C'			
		0D00	973+##\$KRO	EQU	X'0D00'			
		000A	974+##\$@KRO	EQU	10			
			975+*					
		0290	976+##\$KOV	EQU	X'0290'			
		0E00	977+##\$KOV	EQU	X'0E00'			
		0009	978+##\$@KOV	EQU	09			
			979+*					
		02B4	980+##\$KWRI	EQU	X'02B4'			
		0C00	981+##\$KWR	EQU	X'0C00'			
		0002	982+##\$@KWR	EQU	02			
			983+*					
		02BC	984+##\$KREA	EQU	X'02BC'			
		0C00	985+##\$KRE	EQU	X'0C00'			
		0002	986+##\$@KRE	EQU	02			
			987+*					
		02C4	988+##\$KWID	EQU	X'02C4'			
		0C00	989+##\$KWI	EQU	X'0C00'			
		0002	990+##\$@KWI	EQU	02			
			991+*					
		02CC	992+##\$KRUN	EQU	X'02CC'			
		0C00	993+##\$KRU	EQU	X'0C00'			
		0003	994+##\$@KRU	EQU	03			
			995+*					
		0300	996+##\$KDNT	EQU	X'0300'			
		0C00	997+##\$KDN	EQU	X'0C00'			
		0010	998+##\$@KDN	EQU	16			
			999+*					
		030C	1000+##\$KMER	EQU	X'030C'			
		0D00	1001+##\$KME	EQU	X'0D00'			
		0003	1002+##\$@KME	EQU	03			
			1003+*					
		0350	1004+##\$TDCK	EQU	X'0350'			
		1000	1005+##\$TDC	EQU	X'1000'			
		0003	1006+##\$@TDC	EQU	03			
			1007+*					

@SPFEQ - SYSTEM PROGRAM FILE EQUATES

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	25/09/15	PAGE	26
				035C	1008+	#\$KDEL	EQU X'035C'				DISK ADDR OF #KDELE
				0C00	1009+	#\$KDE	EQU X'0C00'				CORE LOAD ADDRESS OF #KDELE
				0010	1010+	#\$@KDE	EQU 16				SECTOR COUNT OF #KDELE
					1011+	*					
				03BC	1012+	#\$KCTL	EQU X'03BC'				DISK ADDR OF #KCTL0
				0C00	1013+	#\$KCT	EQU X'0C00'				CORE LOAD ADDRESS OF #KCTL0
				0009	1014+	#\$@KCT	EQU 09				SECTOR COUNT OF #KCTL0
					1015+	*					
				0400	1016+	#\$KLIS	EQU X'0400'				DISK ADDR OF #KLIST
				0C00	1017+	#\$KLI	EQU X'0C00'				CORE LOAD ADDRESS OF #KLIST
				0011	1018+	#\$@KLI	EQU 17				SECTOR COUNT OF #KLIST
					1019+	*					
				0444	1020+	#\$KLOG	EQU X'0444'				DISK ADDR OF #KLOGO
				0C00	1021+	#\$KLO	EQU X'0C00'				CORE LOAD ADDRESS OF #KLOGO
				0008	1022+	#\$@KLO	EQU 08				SECTOR COUNT OF #KLOGO
					1023+	*					
				0484	1024+	#\$SPSY	EQU X'0484'				DISK ADDR OF #SPSYN
				0C00	1025+	#\$SPS	EQU X'0C00'				CORE LOAD ADDRESS OF #SPSYN
				0001	1026+	#\$@SPS	EQU 01				SECTOR COUNT OF #SPSYN
					1027+	*					
				0488	1028+	#\$KSAV	EQU X'0488'				DISK ADDR OF #KSAVE
				0C00	1029+	#\$KSA	EQU X'0C00'				CORE LOAD ADDRESS OF #KSAVE
				0011	1030+	#\$@KSA	EQU 17				SECTOR COUNT OF #KSAVE
					1031+	*					
				04CC	1032+	#\$SPAC	EQU X'04CC'				DISK ADDR OF #SPACK
				0C00	1033+	#\$SPA	EQU X'0C00'				CORE LOAD ADDRESS OF #SPACK
				0004	1034+	#\$@SPA	EQU 04				SECTOR COUNT OF #SPACK
					1035+	*					
				04DC	1036+	#\$SPOV	EQU X'04DC'				DISK ADDR OF #SPOVL
				0806	1037+	#\$SPO	EQU X'0806'				CORE LOAD ADDRESS OF #SPOVL
				0003	1038+	#\$@SPO	EQU 03				SECTOR COUNT OF #SPOVL
					1039+	*					
				0508	1040+	#\$KPOO	EQU X'0508'				DISK ADDR OF #KPOOL
				0C00	1041+	#\$KPO	EQU X'0C00'				CORE LOAD ADDRESS OF #KPOOL
				000D	1042+	#\$@KPO	EQU 13				SECTOR COUNT OF #KPOOL
					1043+	*					
				053C	1044+	#\$KCHA	EQU X'053C'				DISK ADDR OF #KCHAN
				0C00	1045+	#\$KCH	EQU X'0C00'				CORE LOAD ADDRESS OF #KCHAN
				000C	1046+	#\$@KCH	EQU 12				SECTOR COUNT OF #KCHAN
					1047+	*					
				058C	1048+	#\$KSVL	EQU X'058C'				DISK ADDR OF #KSVLA
				0980	1049+	#\$KSV	EQU X'0980'				CORE LOAD ADDRESS OF #KSVLA
				0002	1050+	#\$@KSV	EQU 02				SECTOR COUNT OF #KSVLA
					1051+	*					
				0594	1052+	#\$KSSP	EQU X'0594'				DISK ADDR OF #KSSPN
				0C00	1053+	#\$KSS	EQU X'0C00'				CORE LOAD ADDRESS OF #KSSPN
				000B	1054+	#\$@KSS	EQU 11				SECTOR COUNT OF #KSSPN
					1055+	*					
				05C0	1056+	#\$KNAM	EQU X'05C0'				DISK ADDR OF #KNAME
				0C00	1057+	#\$KNA	EQU X'0C00'				CORE LOAD ADDRESS OF #KNAME
				0008	1058+	#\$@KNA	EQU 08				SECTOR COUNT OF #KNAME
					1059+	*					
				0600	1060+	#\$KSYM	EQU X'0600'				DISK ADDR OF #KSYMB
				0C00	1061+	#\$KSY	EQU X'0C00'				CORE LOAD ADDRESS OF #KSYMB
				000F	1062+	#\$@KSY	EQU 15				SECTOR COUNT OF #KSYMB
					1063+	*					

@SPFEQ - SYSTEM PROGRAM FILE EQUATES

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	25/09/15	PAGE 27
				063C	1064+	#\$KPRT	EQU X'063C'			DISK ADDR OF #KPRTC
				0C00	1065+	\$\$\$KPR	EQU X'0C00'			CORE LOAD ADDRESS OF #KPRTC
				0009	1066+	#\$@KPR	EQU 09			SECTOR COUNT OF #KPRTC
					1067+	*				
				0680	1068+	#\$KSET	EQU X'0680'			DISK ADDR OF #KSETI
				0E00	1069+	\$\$\$KSE	EQU X'0E00'			CORE LOAD ADDRESS OF #KSETI
				0004	1070+	#\$@KSE	EQU 04			SECTOR COUNT OF #KSETI
					1071+	*				
				0690	1072+	#\$GRAP	EQU X'0690'			DISK ADDR OF #GRAPR
				0889	1073+	\$\$\$GRA	EQU X'0889'			CORE LOAD ADDRESS OF #GRAPR
				0003	1074+	#\$@GRA	EQU 03			SECTOR COUNT OF #GRAPR
					1075+	*				
				06A4	1076+	#\$KALL	EQU X'06A4'			DISK ADDR OF #KALLO
				0C00	1077+	\$\$\$KAL	EQU X'0C00'			CORE LOAD ADDRESS OF #KALLO
				000F	1078+	#\$@KAL	EQU 15			SECTOR COUNT OF #KALLO
					1079+	*				
				0700	1080+	#\$KRLA	EQU X'0700'			DISK ADDR OF #KRLAB
				0700	1081+	\$\$\$KRL	EQU X'0700'			CORE LOAD ADDRESS OF #KRLAB
				0004	1082+	#\$@KRL	EQU 04			SECTOR COUNT OF #KRLAB
					1083+	*				
				0710	1084+	#\$KRVL	EQU X'0710'			DISK ADDR OF #KRVLA
				0800	1085+	\$\$\$KRV	EQU X'0800'			CORE LOAD ADDRESS OF #KRVLA
				000D	1086+	#\$@KRV	EQU 13			SECTOR COUNT OF #KRVLA
					1087+	*				
				0744	1088+	#\$KDIS	EQU X'0744'			DISK ADDR OF #KDISP
				0D00	1089+	\$\$\$KDI	EQU X'0D00'			CORE LOAD ADDRESS OF #KDISP
				0005	1090+	#\$@KDI	EQU 05			SECTOR COUNT OF #KDISP
					1091+	*				
				0780	1092+	#\$KDOV	EQU X'0780'			DISK ADDR OF #KDOVR
				0E00	1093+	\$\$\$KDO	EQU X'0E00'			CORE LOAD ADDRESS OF #KDOVR
				000C	1094+	#\$@KDO	EQU 12			SECTOR COUNT OF #KDOVR
					1095+	*				
				07B4	1096+	#\$VCRT	EQU X'07B4'			DISK ADDR OF #VCRTI
				2000	1097+	\$\$\$VCR	EQU X'2000'			CORE LOAD ADDRESS OF #VCRTI
				0008	1098+	#\$@VCR	EQU 08			SECTOR COUNT OF #VCRTI
					1099+	*				
				07D4	1100+	#\$EXMS	EQU X'07D4'			DISK ADDR OF #EXMSG
				0C00	1101+	\$\$\$EXM	EQU X'0C00'			CORE LOAD ADDRESS OF #EXMSG
				0003	1102+	#\$@EXM	EQU 03			SECTOR COUNT OF #EXMSG
					1103+	*				
				0800	1104+	\$\$\$COR	EQU X'0800'			DISK ADDR OF ##CORE
				0000	1105+	\$\$\$CO	EQU X'0000'			CORE LOAD ADDRESS OF ##CORE
				003A	1106+	#\$@CO	EQU 58			SECTOR COUNT OF ##CORE
					1107+	*				
				0928	1108+	\$\$\$ERM	EQU X'0928'			DISK ADDR OF ##ERMS
				0000	1109+	\$\$\$ER	EQU X'0000'			CORE LOAD ADDRESS OF ##ERMS
				0032	1110+	#\$@ER	EQU 50			SECTOR COUNT OF ##ERMS
					1111+	*				
				0A30	1112+	#\$KHEL	EQU X'0A30'			DISK ADDR OF #KHELP
				0C00	1113+	\$\$\$KHE	EQU X'0C00'			CORE LOAD ADDRESS OF #KHELP
				000C	1114+	#\$@KHE	EQU 12			SECTOR COUNT OF #KHELP
					1115+	*				
				0A80	1116+	#\$MIPP	EQU X'0A80'			DISK ADDR OF #MIPPE
				0C00	1117+	\$\$\$MIP	EQU X'0C00'			CORE LOAD ADDRESS OF #MIPPE
				000D	1118+	#\$@MIP	EQU 13			SECTOR COUNT OF #MIPPE
					1119+	*				

@SPFEQ - SYSTEM PROGRAM FILE EQUATES

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	25/09/15	PAGE 28
				0AC8	1120+	#\$KSOV	EQU X'0AC8'			DISK ADDR OF #KSOVR
				0C20	1121+	\$\$\$KSO	EQU X'0C20'			CORE LOAD ADDRESS OF #KSOVR
				000D	1122+	#\$@KSO	EQU 13			SECTOR COUNT OF #KSOVR
					1123+	*				
				0B00	1124+	#\$VXIT	EQU X'0B00'			DISK ADDR OF #VXITI
				0600	1125+	\$\$\$VXI	EQU X'0600'			CORE LOAD ADDRESS OF #VXITI
				0002	1126+	#\$@VXI	EQU 02			SECTOR COUNT OF #VXITI
					1127+	*				
				0B08	1128+	\$\$\$VUF	EQU X'0B08'			DISK ADDR OF ##VUFA
				0600	1129+	\$\$\$#VU	EQU X'0600'			CORE LOAD ADDRESS OF ##VUFA
				0002	1130+	#\$@#VU	EQU 02			SECTOR COUNT OF ##VUFA
					1131+	*				
				0B80	1132+	#\$VLOA	EQU X'0B80'			DISK ADDR OF #VLOAD
				0600	1133+	\$\$\$VLO	EQU X'0600'			CORE LOAD ADDRESS OF #VLOAD
				0002	1134+	#\$@VLO	EQU 02			SECTOR COUNT OF #VLOAD
					1135+	*				
				0B88	1136+	#\$VODK	EQU X'0B88'			DISK ADDR OF #VODKA
				0600	1137+	\$\$\$VOD	EQU X'0600'			CORE LOAD ADDRESS OF #VODKA
				0016	1138+	#\$@VOD	EQU 22			SECTOR COUNT OF #VODKA
					1139+	*				
				0BAC	1140+	#\$TVKB	EQU X'0BAC'			DISK ADDR OF #TVKBT
				0FC0	1141+	\$\$\$TVK	EQU X'0FC0'			CORE LOAD ADDRESS OF #TVKBT
				0001	1142+	#\$@TVK	EQU 01			SECTOR COUNT OF #TVKBT
					1143+	*				
				0C00	1144+	#\$VVMR	EQU X'0C00'			DISK ADDR OF #VVMRS
				0000	1145+	\$\$\$VVM	EQU X'0000'			CORE LOAD ADDRESS OF #VVMRS
				0030	1146+	#\$@VVM	EQU 48			SECTOR COUNT OF #VVMRS
					1147+	*				
				0D00	1148+	#\$FMST	EQU X'0D00'			DISK ADDR OF #FMSTD
				0200	1149+	\$\$\$FMS	EQU X'0200'			CORE LOAD ADDRESS OF #FMSTD
				0052	1150+	#\$@FMS	EQU 82			SECTOR COUNT OF #FMSTD
					1151+	*				
				0EA8	1152+	#\$UEXL	EQU X'0EA8'			DISK ADDR OF #UEXLI
				0C00	1153+	\$\$\$UEX	EQU X'0C00'			CORE LOAD ADDRESS OF #UEXLI
				000E	1154+	#\$@UEX	EQU 14			SECTOR COUNT OF #UEXLI
					1155+	*				
				0F00	1156+	#\$UALL	EQU X'0F00'			DISK ADDR OF #UALLO
				0C00	1157+	\$\$\$UAL	EQU X'0C00'			CORE LOAD ADDRESS OF #UALLO
				0011	1158+	#\$@UAL	EQU 17			SECTOR COUNT OF #UALLO
					1159+	*				
				0F80	1160+	#\$KCND	EQU X'0F80'			DISK ADDR OF #KCNDI
				0C00	1161+	\$\$\$KCN	EQU X'0C00'			CORE LOAD ADDRESS OF #KCNDI
				0010	1162+	#\$@KCN	EQU 16			SECTOR COUNT OF #KCNDI
					1163+	*				
				1000	1164+	\$\$\$CSA	EQU X'1000'			DISK ADDR OF #CSAV
				0000	1165+	\$\$\$#CS	EQU X'0000'			CORE LOAD ADDRESS OF #CSAV
				003A	1166+	#\$@#CS	EQU 58			SECTOR COUNT OF #CSAV
					1167+	*				
				1128	1168+	\$\$\$SSA	EQU X'1128'			DISK ADDR OF #SSAV
				0000	1169+	\$\$\$#SS	EQU X'0000'			CORE LOAD ADDRESS OF #SSAV
				0001	1170+	#\$@#SS	EQU 01			SECTOR COUNT OF #SSAV
					1171+	*				
				1180	1172+	\$\$\$SAV	EQU X'1180'			DISK ADDR OF ##SAVM
				0000	1173+	\$\$\$#SA	EQU X'0000'			CORE LOAD ADDRESS OF ##SAVM
				0108	1174+	#\$@#SA	EQU 264			SECTOR COUNT OF ##SAVM
					1175+	*				



@SPFEQ - SYSTEM PROGRAM FILE EQUATES

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	25/09/15	PAGE	29
		1700	1176+	#\$FIST	EQU	X'1700'			
		0E00	1177+	\$\$\$FIS	EQU	X'0E00'			
		0009	1178+	\$\$@FIS	EQU	09			
			1179+	*					
		1724	1180+	#\$FILN	EQU	X'1724'			
		0E00	1181+	\$\$\$FIL	EQU	X'0E00'			
		0009	1182+	\$\$@FIL	EQU	09			
			1183+	*					
		1780	1184+	\$\$#RSP	EQU	X'1780'			
		0000	1185+	\$\$\$#RS	EQU	X'0000'			
		0030	1186+	\$\$@#RS	EQU	48			
			1187+	*					
		1780	1188+	#\$BOLV	EQU	X'1780'			
		0800	1189+	\$\$\$BOV	EQU	X'0800'			
		0018	1190+	\$\$@BOV	EQU	24			
			1191+	*					
		1800	1192+	\$\$SFSY	EQU	X'1800'			
		0C00	1193+	\$\$\$SFS	EQU	X'0C00'			
		0011	1194+	\$\$@SFS	EQU	17			
			1195+	*					
		1844	1196+	\$\$SFOV	EQU	X'1844'			
		1500	1197+	\$\$\$SFO	EQU	X'1500'			
		0003	1198+	\$\$@SFO	EQU	03			
			1199+	*					
		1850	1200+	\$\$STRO	EQU	X'1850'			
		1600	1201+	\$\$\$STR	EQU	X'1600'			
		0002	1202+	\$\$@STR	EQU	02			
			1203+	*					
		1880	1204+	\$\$#FSP	EQU	X'1880'			
		0000	1205+	\$\$\$#FS	EQU	X'0000'			
		0030	1206+	\$\$@#FS	EQU	48			
			1207+	*					
		1880	1208+	\$\$GUFU	EQU	X'1880'			
		0C00	1209+	\$\$\$GUF	EQU	X'0C00'			
		0010	1210+	\$\$@GUF	EQU	16			
			1211+	*					
		18C0	1212+	\$\$ERRP	EQU	X'18C0'			
		0C00	1213+	\$\$\$ERR	EQU	X'0C00'			
		0003	1214+	\$\$@ERR	EQU	03			
			1215+	*					
		18D4	1216+	\$\$#BLN	EQU	X'18D4'			
		0000	1217+	\$\$\$#BL	EQU	X'0000'			
		0001	1218+	\$\$@#BL	EQU	01			
			1219+	*					
		1900	1220+	\$\$ECMA	EQU	X'1900'			
		0C00	1221+	\$\$\$ECM	EQU	X'0C00'			
		0006	1222+	\$\$@ECM	EQU	06			
			1223+	*					
		1918	1224+	\$\$SFLO	EQU	X'1918'			
		0F00	1225+	\$\$\$SFL	EQU	X'0F00'			
		0005	1226+	\$\$@SFL	EQU	05			
			1227+	*					
		192C	1228+	\$\$SDSY	EQU	X'192C'			
		0C00	1229+	\$\$\$SDS	EQU	X'0C00'			
		0004	1230+	\$\$@SDS	EQU	04			
			1231+	*					

@SPFEQ - SYSTEM PROGRAM FILE EQUATES

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	25/09/15	PAGE 30
		193C	1232+	#\$SFFI EQU	X'193C'			
		0E00	1233+	#\$SFF EQU	X'0E00'			
		0008	1234+	#\$@SFF EQU	08			
			1235+	*				
		1980	1236+	#\$UPAC EQU	X'1980'			
		0C00	1237+	#\$UPA EQU	X'0C00'			
		0004	1238+	#\$@UPA EQU	04			
			1239+	*				
		1990	1240+	#\$EFKE EQU	X'1990'			
		0C00	1241+	#\$EFK EQU	X'0C00'			
		0002	1242+	#\$@EFK EQU	02			
			1243+	*				
		19B8	1244+	#\$UCNF EQU	X'19B8'			
		0C00	1245+	#\$UCN EQU	X'0C00'			
		0009	1246+	#\$@UCN EQU	09			
			1247+	*				
		19DC	1248+	#\$UCPL EQU	X'19DC'			
		0700	1249+	#\$UCP EQU	X'0700'			
		000F	1250+	#\$@UCP EQU	15			
			1251+	*				
		1A38	1252+	#\$UATR EQU	X'1A38'			
		0900	1253+	#\$UAT EQU	X'0900'			
		000C	1254+	#\$@UAT EQU	12			
			1255+	*				
		1A88	1256+	#\$UINI EQU	X'1A88'			
		0C00	1257+	#\$UIN EQU	X'0C00'			
		000F	1258+	#\$@UIN EQU	15			
			1259+	*				
		1AD8	1260+	#\$UCDI EQU	X'1AD8'			
		0900	1261+	#\$UCD EQU	X'0900'			
		000B	1262+	#\$@UCD EQU	11			
			1263+	*				
		1B24	1264+	#\$UDEL EQU	X'1B24'			
		0C00	1265+	#\$UDE EQU	X'0C00'			
		000E	1266+	#\$@UDE EQU	14			
			1267+	*				
		1B5C	1268+	#\$UDIS EQU	X'1B5C'			
		0C00	1269+	#\$UDI EQU	X'0C00'			
		0008	1270+	#\$@UDI EQU	08			
			1271+	*				
		1B9C	1272+	#\$ZTRA EQU	X'1B9C'			
		1000	1273+	#\$ZTR EQU	X'1000'			
		0001	1274+	#\$@ZTR EQU	01			
			1275+	*				
		1BA4	1276+	#\$ZDUM EQU	X'1BA4'			
		1100	1277+	#\$ZDU EQU	X'1100'			
		0008	1278+	#\$@ZDU EQU	08			
			1279+	*				
		1BC4	1280+	#\$ZLOA EQU	X'1BC4'			
		1100	1281+	#\$ZLO EQU	X'1100'			
		000C	1282+	#\$@ZLO EQU	12			
			1283+	*				
		1C14	1284+	#\$ZUTM EQU	X'1C14'			
		0C00	1285+	#\$ZUT EQU	X'0C00'			
		0014	1286+	#\$@ZUT EQU	20			
			1287+	*				

@SPFEQ - SYSTEM PROGRAM FILE EQUATES

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	25/09/15	PAGE 31
				1C84	1288+	#\$INLN	EQU X'1C84'			DISK ADDR OF #INLNG
				0600	1289+	\$\$\$INL	EQU X'0600'			CORE LOAD ADDRESS OF #INLNG
				0010	1290+	\$\$@INL	EQU 16			SECTOR COUNT OF #INLNG
					1291+	*				
				1CC4	1292+	#\$KCAL	EQU X'1CC4'			DISK ADDR OF #KCALL
				0C00	1293+	\$\$\$KCA	EQU X'0C00'			CORE LOAD ADDRESS OF #KCALL
				000C	1294+	\$\$@KCA	EQU 12			SECTOR COUNT OF #KCALL
					1295+	*				
				1D24	1296+	#\$KRSU	EQU X'1D24'			DISK ADDR OF #KRSUM
				0C00	1297+	\$\$\$KRS	EQU X'0C00'			CORE LOAD ADDRESS OF #KRSUM
				000A	1298+	\$\$@KRS	EQU 10			SECTOR COUNT OF #KRSUM
					1299+	*				
				1D5C	1300+	#\$UPTF	EQU X'1D5C'			DISK ADDR OF #UPTFI
				0C00	1301+	\$\$\$UPT	EQU X'0C00'			CORE LOAD ADDRESS OF #UPTFI
				0012	1302+	\$\$@UPT	EQU 18			SECTOR COUNT OF #UPTFI
					1303+	*				
				1D24	1304+	#\$UPOV	EQU X'1D24'			DISK ADDR OF #UPOVL
				0C00	1305+	\$\$\$UPO	EQU X'0C00'			CORE LOAD ADDRESS OF #UPOVL
				0005	1306+	\$\$@UPO	EQU 05			SECTOR COUNT OF #UPOVL
					1307+	*				
				1E00	1308+	#\$FMLN	EQU X'1E00'			DISK ADDR OF #FMLNG
				0200	1309+	\$\$\$FML	EQU X'0200'			CORE LOAD ADDRESS OF #FMLNG
				0052	1310+	\$\$@FML	EQU 82			SECTOR COUNT OF #FMLNG
					1311+	*				
				2000	1312+	\$\$\$CNF	EQU X'2000'			DISK ADDR OF ##CNFI
				0000	1313+	\$\$\$CN	EQU X'0000'			CORE LOAD ADDRESS OF ##CNFI
				0001	1314+	\$\$@CN	EQU 01			SECTOR COUNT OF ##CNFI
					1315+	*				
				2004	1316+	#\$KLLA	EQU X'2004'			DISK ADDR OF #KLLAY
				0920	1317+	\$\$\$KLL	EQU X'0920'			CORE LOAD ADDRESS OF #KLLAY
				0001	1318+	\$\$@KLL	EQU 01			SECTOR COUNT OF #KLLAY
					1319+	*				
				2008	1320+	#\$ZLBM	EQU X'2008'			DISK ADDR OF #ZLBMA
				1100	1321+	\$\$\$ZLB	EQU X'1100'			CORE LOAD ADDRESS OF #ZLBMA
				0002	1322+	\$\$@ZLB	EQU 02			SECTOR COUNT OF #ZLBMA
					1323+	*				
				2010	1324+	#\$ZL1M	EQU X'2010'			DISK ADDR OF #ZL1MA
				0F00	1325+	\$\$\$ZL1	EQU X'0F00'			CORE LOAD ADDRESS OF #ZL1MA
				0007	1326+	\$\$@ZL1	EQU 07			SECTOR COUNT OF #ZL1MA
					1327+	*				
				2030	1328+	#\$ZL2M	EQU X'2030'			DISK ADDR OF #ZL2MA
				0F00	1329+	\$\$\$ZL2	EQU X'0F00'			CORE LOAD ADDRESS OF #ZL2MA
				000D	1330+	\$\$@ZL2	EQU 13			SECTOR COUNT OF #ZL2MA
					1331+	*				
				2088	1332+	#\$ZL3M	EQU X'2088'			DISK ADDR OF #ZL3MA
				0C00	1333+	\$\$\$ZL3	EQU X'0C00'			CORE LOAD ADDRESS OF #ZL3MA
				000A	1334+	\$\$@ZL3	EQU 10			SECTOR COUNT OF #ZL3MA
					1335+	*				
				20B0	1336+	#\$ZLVR	EQU X'20B0'			DISK ADDR OF #ZLVRL
				0F00	1337+	\$\$\$ZLV	EQU X'0F00'			CORE LOAD ADDRESS OF #ZLVRL
				0006	1338+	\$\$@ZLV	EQU 06			SECTOR COUNT OF #ZLVRL
					1339+	*				
				2100	1340+	#\$KKEY	EQU X'2100'			DISK ADDR OF #KKEYS
				0C00	1341+	\$\$\$KKE	EQU X'0C00'			CORE LOAD ADDRESS OF #KKEYS
				0006	1342+	\$\$@KKE	EQU 06			SECTOR COUNT OF #KKEYS
					1343+	*				



@SPFEQ - SYSTEM PROGRAM FILE EQUATES

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	25/09/15	PAGE 32
		2118	1344+	##CKT EQU	X'2118'			
		0000	1345+	##\$CK EQU	X'0000'			
		0004	1346+	##\$@CK EQU	04			
			1347+	*				
		212C	1348+	##\$INV EQU	X'212C'			
		0000	1349+	##\$IN EQU	X'0000'			
		003A	1350+	##\$@IN EQU	58			
			1351+	*				
		2300	1352+	##\$PWR EQU	X'2300'			
		0000	1353+	##\$PW EQU	X'0000'			
		00C0	1354+	##\$@PW EQU	192			
			1355+	*	END OF SYSTEM PROGRAM FILE EQUATES			
			1356+		PRINT ON			
			1357	*	@CAN EXP-Y			
			1359+		PRINT ON			

@CANEQ - COMMON CORE LOCATIONS OUTSIDE NUCLEUS

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 25/09/15 PAGE 33
		1361+	*****		
		1362+	*	INPUT LINE HEADER	*
		1363+	*****		
0600		1364+	\$\$\$ILHD EQU	\$ENDNU	FIRST BYTE OF INPUT LINE HEADER
		1365+	*		
0601		1366+	\$\$\$ILEN EQU	\$\$\$ILHD+1	SECOND BYTE OF SDF LENGTH FIELD
		1367+	*		
0602		1368+	\$\$\$UPAR EQU	\$\$\$ILEN+1	UP ARROW LOCATION IN LAST LINE
		1369+	*		
0603		1370+	\$\$\$CKEY EQU	\$\$\$UPAR+1	CMD KEY FUNCTION CODE
		1371+	*		* EXECUTABLE CMD KEYS
0605		1372+	\$\$\$BNLN EQU	\$\$\$ILEN+4	SECOND BYTE OF BINARY LINE NO.
		1373+	*		
0606		1374+	\$\$\$TPCD EQU	\$\$\$BNLN+1	TYPE CODE FIELD
		1376+	*****		
		1377+	*	INPUT LINE TEXT	*
		1378+	*****		
0607		1379+	\$\$\$INLN EQU	\$\$\$TPCD+1	FIRST BYTE CHAR OF INPUT LINE
		1380+	*		
0666		1381+	\$\$\$CDND EQU	\$\$\$INLN+@CARDL-1	LAST CHAR OF CARD INPUT
		1382+	*		
06FA		1383+	\$\$\$INND EQU	\$\$\$INLN+@LINSZ-1	LAST CHAR OF INPUT LINE BUFFER
		1385+	*****		
		1386+	*	KEYBOARD ROUTINE LOCATIONS AND MASKS	*
		1387+	*****		
0890		1388+	\$\$\$PRES EQU	\$ENDNU+X'0290'	ENABLE KEYBOARD ENTRY TO DEPRES
		1389+	*		
09E1		1390+	\$\$\$KBDT EQU	\$\$\$PRES+X'0151'	DATA BYTE FROM KEYBOARD
0081		1391+	\$\$\$STD EQU	B'10000001'	CLI MASK FOR START KEY DATA
0091		1392+	\$\$\$EPL EQU	B'10010001'	CLI MASK FOR ENTER PLUS KEY
		1393+	*		
09E2		1394+	\$\$\$KBSN EQU	\$\$\$KBDT+1	TYPE BYTE FROM KEYBOARD
0040		1395+	\$\$\$DAT EQU	B'01000000'	TBM MASK FOR DATA KEY
0020		1396+	\$\$\$CMD EQU	B'00100000'	TBM MASK FOR COMMAND KEY
0010		1397+	\$\$\$FUN EQU	B'00010000'	TBM MASK FOR FUNCTION KEY
		1398+	*		
09EB		1399+	\$\$\$LPOS EQU	\$\$\$KBSN+9	PRINT HEAD POSITION ADDR
0AFE		1400+	\$\$\$EOSA EQU	\$\$\$PRES+X'026E'	LOCATION OF EOS ADDR
0B44		1401+	\$\$\$COFF EQU	\$\$\$PRES+X'02B4'	ENTRY TO TURN OFF CMD LIGHTS
0B3D		1402+	\$\$\$CKFF EQU	\$\$\$PRES+X'02AD'	ENTRY TO TURN OFF CMD LIGHTS 1-1
0BBF		1403+	\$\$\$DATB EQU	\$\$\$PRES+X'032F'	ADDR OF DATA TABLE TYPE INDR IN
		1404+	*		* DEPRES (VALUE: 1-9)
		1406+	*****		
		1407+	*	MATRIX PRINTER ROUTINE ENTRY POINT	*
		1408+	*****		
0707		1409+	\$\$\$PRNT EQU	\$ENDNU+X'0100'+@HDRLN	DPRINT ENTRY
0782		1410+	\$\$\$PRTN EQU	\$\$\$PRNT+X'007B'	ADDR OF CARRIER RETURN TEST IN
		1411+	*		* DPRINT. MASKS FOLLOE
		1412+	*		* @NOP - NO TEST MADE
		1413+	*		* @BNL - TEST WILL BE MADE
07CE		1414+	\$\$\$PSIO EQU	\$\$\$PRNT+X'00C7'	ADDR OF SIO CTRL IN DPRINT
07E9		1415+	\$\$\$PCNT EQU	\$\$\$PRNT+X'00E2'	ADDR OF PPL CNT IN DPRINT

@CANEQ - COMMON CORE LOCATIONS OUTSIDE NUCLEUS

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	25/09/15	PAGE	34
					1417+	*****	*****				
					1418+	*	CARD READER LOCATIONS				*
					1419+	*****	*****				
	0890				1420+	\$\$\$CDRD EQU	\$\$PRES				ENTRY POINT TO READ CARDS
					1421+	*					
	08C0				1422+	\$\$\$CDBS EQU	\$\$\$CDRD+X'0030'				ENTRY POINT TO WAIT FOR READ
					1424+	*****	*****				
					1425+	*	CRT OUTPUT ROUTINE LOCATIONS				*
					1426+	*****	*****				
	2000				1427+	\$\$\$PYMP EQU	\$\$ZERO+X'2000'				ENTRY POINT TO CRT PLUS PRINT
					1428+	*					
	2004				1429+	\$\$\$PLYN EQU	\$\$\$PYMP+4				ENTRY POINT TO CRT ONLY
					1430+	*					
	209C				1431+	\$\$\$CSNS EQU	\$\$\$PYMP+X'009C'				LOCATION OF SENSE BYTE IN
					1432+	*					* DSPLYN
	2143				1433+	\$\$\$PRFL EQU	\$\$\$PYMP+X'0143'				ENTRY POINT FOR PRINTER FAILURE
					1434+	*					
	2200				1435+	\$\$\$PYCD EQU	\$\$\$PYMP+X'0200'				ENTRY POINT FOR COMMAND KEYS
					1436+	*					* OR CLEAR CRT FUNCTION
					1438+	*****	*****				
					1439+	*	MISCELLANEOUS LOCATIONS				*
					1440+	*****	*****				
	1C00				1441+	\$\$\$ERSK EQU	X'1C00'				START ADDR OF ERROR CODE STACK
	00A0				1442+	\$\$\$NLN EQU	X'00A0'				HIGH ORDER BYTE OF LINE NUMBER
					1443+	*					* IN STACK IF NO. NOT DESIRED
	1C00				1444+	\$\$\$SLIB EQU	X'1C00'				SECONDARY LINE INPUT BUFFER
	06FF				1445+	\$\$\$XIND EQU	\$\$ENDNU+X'00FF'				EXEC INDR PASS AREA
	0080				1446+	\$\$\$ERN EQU	B'10000000'				RUN FUNC SAVED FILE INDR MASK
	1E00				1447+	\$\$\$WSPB EQU	X'1E00'				LOCATION OF BAGETC BUFFER
	06FF				1448+	\$\$\$FLIB EQU	\$\$\$XIND				FILE LIB ADDR PASS AREA
	1D00				1449+	\$\$\$FITS EQU	X'1D00'				LOCATION OF FIT
					1451+	*****	*****				
					1452+	*	KEYWORD COMMAND LOAD ADDRESSES				*
					1453+	*****	*****				
	0600				1454+	\$\$\$KLD1 EQU	\$\$ENDNU				PROGRAMS THAT LOAD BEHIND
					1455+	*					* SYSNUC
	0700				1456+	\$\$\$KLD2 EQU	\$\$ENDNU+X'0100'				PROGRAMS THAT LOAD BEHIND
					1457+	*					* THE INPUT LINE BUFFER
	0C00				1458+	\$\$\$KLD3 EQU	\$\$ENDNU+X'0600'				STANDARD LOAD ADDRESS BEHIND
					1459+	*					* I/O ROUTINES
					1460+	*	END OF COMMON CORE LOCATIONS EQUATES				
					1461+	*	PRINT ON				
					1462	*	@VOL EXP-Y				
					1464+	*	PRINT ON				

@VOLEQ - VOLUME LABEL EQUATES

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00 25/09/15 PAGE 35
				1466+	*****			
				1467+	*	VOLUME LABEL EQUATES	*	
				1468+	*****			
	0002			1469+	\$#TVOL	EQU	X'02'	START OF VOLUME LABEL ('VOL')
	0008			1470+	\$#TLBL	EQU	X'08'	VOLUME LABEL
	000A			1471+	\$#TVTC	EQU	X'0A'	VTOC POINTER
	005B			1472+	\$#TOID	EQU	X'5B'	OWNER ID
	005C			1473+	\$#TCYL	EQU	X'5C'	NUMBER OF CYLINDERS ON DISK
	0069			1474+	\$#TCET	EQU	X'69'	CE TRACK INDICATOR 1-5
	0075			1475+	\$#TALT	EQU	X'75'	ALTERNATE TRACK ASSIGNMENT
	00A8			1476+	\$#TUSE	EQU	X'A8'	TACK USAGE MASK
	00EF			1477+	\$#TSUS	EQU	X'EF'	SUSPECTED DEFECTIVE TRACKS
	00F0			1478+	\$#THVT	EQU	X'F0'	HELP FILE VTOC TAG NO.
	00F2			1479+	\$#THAD	EQU	X'F2'	HELP FILE DADDR
	00F3			1480+	\$#TPTF	EQU	X'F3'	PTF VTOC TAG NO.
	00F4			1481+	\$#TPSZ	EQU	X'F4'	PTF SIZE
	00F6			1482+	\$#TPAD	EQU	X'F6'	PTF DADDR
	00F7			1483+	\$#TLSZ	EQU	X'F7'	PTF SIZE
	00F8			1484+	\$#TLIB	EQU	X'F8'	LIBRARY VTOC TAG NO.
	00F9			1485+	\$#TWRK	EQU	X'F9'	WORK AREA VTOC TAG NO.
	00FA			1486+	\$#TSYS	EQU	X'FA'	SYSTEM PGM FILE VTOC TAG NO.
	00FC			1487+	\$#TBIS	EQU	X'FC'	BIS SYSTEM FILE DADDR
	00FE			1488+	\$#TLAD	EQU	X'FE'	BIS USER LIBRARY DADDR
	00FF			1489+	\$#TIDR	EQU	X'FF'	BIS FILES INDICATOR
	00D7			1490+	\$#TWAL	EQU	215	DISP TO WKAREA RELEASE LEVEL
	00D7			1491+	\$#TRES	EQU	215	DISP TO END OF BIS RESERVED AREA
				1493+	*			BIS FILES INDR BYTE:
	0080			1494+	\$#TSYM	EQU	X'80'	BIT 0 - SYSTEM PROGRAM FILE.
	0040			1495+	\$#TWR1	EQU	X'40'	* 1 - WORK AREA R1
	0020			1496+	\$#TWF1	EQU	X'20'	* 2 - WORK AREA F1
	0010			1497+	\$#TLIF	EQU	X'10'	* 3 - LIBRARY FILE
	0008			1498+	\$#TPFL	EQU	X'08'	* 4 - PTF DATA FILE
	0004			1499+	\$#THEL	EQU	X'04'	* 5 - HELP FILE
				1500+	*	END OF VOLUME LABEL EQUATES		
				1501+		PRINT ON		
				1502	*	@HLT EXP-Y		
				1504+		PRINT ON		

@HLTEQ - HALT INDICATOR EQUATES

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00 25/09/15 PAGE 36
		1506+	*****			
		1507+	*		THESE EQUATES, WHEN USED WITH THE HPL INSTRUCTION AS A TWO	*
		1508+	*		ADDRESS CONSTANT REPLACING THE Q AND R FIELDS, WILL CAUSE THE	*
		1509+	*		CORRESPONDING HALT INDICATORS TO BE LIT.	*
		1510+	*****			
		2040	1512+	@HKBER EQU	X'2040'	KEYBOARD PARITY ERROR SOFT HALT
			1513+	*		* CODE ' B 1 '
	0070	1514+	@HPRER EQU	X'0070'		MATRIX PRINTER ERROR SOFT HALT
			1515+	*		* CODE ' 123 '
	1040	1516+	@HDTRD EQU	X'1040'		DATA RECORDER ERROR SOFT HALT
			1517+	*		* CODE ' C 1 '
	1010	1518+	@HDTRJ EQU	X'1010'		DATA RECORDER TRANSPORT JAM
			1519+	*		* CODE ' C 3 '
	1008	1520+	@HDNRY EQU	X'1008'		DATA RECORDER NOT READY
			1521+	*		* CODE ' C 4 '
	087C	1522+	@HERPG EQU	X'087C'		HARD HALT AFTER ERROR MESSAGE
			1523+	*		* CODE ' D12345'
	1844	1524+	@HLOGE EQU	X'1844'		HARD DISK ERROR WHILE LOGGING
			1525+	*		* AN I/O ERROR
			1526+	*		* CODE ' CD1 5'
	1850	1527+	@HUNSF EQU	X'1850'		HARD DISK UNSAFE ERROR
			1528+	*		* CODE ' CD1 3 '
	006C	1529+	@HIPLE EQU	X'006C'		HARD HALT WHEN NO SYSTEM PGM
			1530+	*		* FILE FOUND ON IPL'D DISK
			1531+	*		* CODE ' 12 45'
	003C	1532+	@HCEPK EQU	X'003C'		HARD HALT FOR CE PACK
			1533+	*		* CODE ' 2345'
	081C	1534+	@HCOPY EQU	X'081C'		HARD HALT ON TERMINATION OF
			1535+	*		* COPY DISK FUNCTION
			1536+	*		* CODE ' D 345'
	0804	1537+	@HFEHT EQU	X'0804'		HARD HALT ON ZUTMON 'H' OPTION
			1538+	*		* CODE ' D 5'
	001C	1539+	@HCOPS EQU	X'001C'		SOFT HALT ON INTERMEDIATE COPY
			1540+	*		* DISK FUNCTION
			1541+	*		* CODE ' 345'
			1542+	*		
		1543+	***		HARD I/O ERROR HALTS	
			1544+	*		
	7840	1545+	@HDRV1 EQU	X'7840'		HARD ERROR ON DRIVE 1
			1546+	*		* CODE 'ABCD1 '
	7844	1547+	@HDRV2 EQU	X'7844'		HARD ERROR ON DRIVE 2
			1548+	*		* CODE 'ABCD1 5'
	7848	1549+	@HKBHE EQU	X'7848'		HARD KEYBOARD ERROR
			1550+	*		* CODE 'ABCD1 4 '
	784C	1551+	@HPRHE EQU	X'784C'		HARD PRINTER ERROR
			1552+	*		* CODE 'ABCD1 45'
	7854	1553+	@HDRHE EQU	X'7854'		HARD DATA RECORDER ERROR
			1554+	*		* CODE 'ABCD1 3 5'
	7858	1555+	@HCRHE EQU	X'7858'		HARD CRT ERROR
			1556+	*		* CODE 'ABCD1 34 '
		1557+	*		END OF HALT EQUATES	
		1558+	*		PRINT ON	
		1559	*		@GMAC EXP-Y	
		1561+	*		PRINT ON	

## S/3 BASIC INTERPRETER FIXED ADDRESS EQUATES

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00 25/09/15 PAGE 37
		1563+	*****			
		1564+	*	FIXED REGION CORE ADDRESS EQUATES		*
		1565+	*****			
		2000	1567+	I\$CSXA EQU	X'2000'	CORE EXTENSION STARTING ADDR
		1600	1568+	I\$CPG1 EQU	X'1600'	CORE PAGE REGION STARTING ADDR
		1570+	*****			
		1571+	*	CORE RESIDENT ROUTINE ENTRY POINTS AND PARAMETER ADDRESSES		*
		1572+	*****			
		0607	1574+	I\$INIT EQU	X'0607'	ENTRY - INTERPRETER INITIATOR
		0607	1575+	I\$FWRK EQU	X'0607'	INTERPRETER FUNCTION WORK AREA
		0639	1576+	I\$STKB EQU	X'0639'	RUN-TIME STACK BASE CORE ADDR
			1577+			*
		0C5C	1578+	I\$INTR EQU	X'0C5C'	ENTRY - INTERPRETER EXECUTIVE
		0C68	1579+	I\$NPAG EQU	X'0C68'	ENTRY - RESET EXECUTION CONTROL
		0C74	1580+	I\$XAD4 EQU	X'0C74'	ENTRY - INCR IAR, 4-BYTE INSTR
		0C7B	1581+	I\$XAD3 EQU	X'0C7B'	ENTRY - INCR IAR, 3-BYTE INSTR
		0C82	1582+	I\$XAD2 EQU	X'0C82'	ENTRY - INCR IAR, 2-BYTE INSTR
		0C89	1583+	I\$XAD1 EQU	X'0C89'	ENTRY - INCR IAR, 1-BYTE INSTR
		0C9D	1584+	I\$ADST EQU	X'0C9D'	ENTRY - INCR STACK POINTER RTN
		0CAB	1585+	I\$XERR EQU	X'0CAB'	ENTRY - EXECUTION ERROR RTN
		0C60	1586+	I\$BASE EQU	X'0C60'	INTERPRETER BASE CORE ADDR
		0D4E	1587+	I\$STAK EQU	X'0D4E'	RUN-TIME STACK POINTER
		0D4F	1588+	I\$STKI EQU	X'0D4F'	RUN-TIME STACK POINTER INCR
		0C61	1589+	I\$XPAG EQU	X'0C61'	CURRENT PSEUDO INSTRUCTION PAGE
		0D4C	1590+	I\$XIAR EQU	X'0D4C'	PSEUDO INSTRUCTION CORE ADDR
		0CBC	1591+	I\$ERRC EQU	X'0CBC'	EXECUTION ERROR CODE BYTE
		0D53	1592+	I\$DATA EQU	X'0D53'	INTERNAL DATA FILE VADDR RTN
		0D55	1593+	I\$DAT1 EQU	X'0D55'	DATA FILE 1ST ELEMENT VADDR
		0D57	1594+	I\$PARM EQU	X'0D57'	PARAMETER COMMUNICATION AREA
		0D59	1595+	I\$WRK1 EQU	X'0D59'	GENERAL PURPOSE WORK AREA 1
		0D5B	1596+	I\$WRK2 EQU	X'0D5B'	GENERAL PURPOSE WORK AREA 2
		0D51	1597+	I\$STHA EQU	X'0D51'	CURRENT STMT STH INST VADDR
		0CDE	1598+	I\$IRSW EQU	X'0CDE'	IMAGE STMT REFERENCE SWITCH
		0CE9	1599+	I\$RESW EQU	X'0CE9'	STATEMENT RECURSION ERR SWITCH
		0D2B	1600+	I\$TFSW EQU	X'0D2B'	INTERPRETER TRACE FLOW SWITCH
			1601+			*
		0D3E	1602+	I\$CBN1 EQU	X'0D3E'	BINARY CONSTANT, +1
		0D3F	1603+	I\$CBN2 EQU	X'0D3F'	BINARY CONSTANT, +2
		0D40	1604+	I\$CBN3 EQU	X'0D40'	BINARY CONSTANT, +3
		0D41	1605+	I\$CBN4 EQU	X'0D41'	BINARY CONSTANT, +4
		0D43	1606+	I\$CBM1 EQU	X'0D43'	BINARY CONSTANT, -1
		0D44	1607+	I\$CL1F EQU	X'0D44'	LENGTH CONSTANT, 1 FLT VALUE
		0D45	1608+	I\$CL2F EQU	X'0D45'	LENGTH CONSTANT, 2 FLT VALUES
		0D46	1609+	I\$CL1C EQU	X'0D46'	LENGTH CONSTANT, 1 CHAR VALUE
		0D47	1610+	I\$CL2C EQU	X'0D47'	LENGTH CONSTANT, 2 CHAR VALUES
		0D49	1611+	I\$CLVA EQU	X'0D49'	LENGTH CONSTANT, VIRTUAL ADDR
		0D4A	1612+	I\$CLFA EQU	X'0D4A'	LENGTH CONSTANT, FLT VAL, VADDR
			1613+			*
		0D5C	1614+	I\$RNSW EQU	X'0D5C'	RANDOM NUMBER INITIALIZE SWITCH
		0001	1615+	I\$RNMK EQU	X'0001'	RANDOM NUMBER INITIALIZE SW MASK
			1616+			*
		0DC5	1617+	I\$INDR EQU	X'0DC5'	PRINT USAGE INDICATOR BYTE
		0DC6	1618+	I\$IMLN EQU	X'0DC6'	IMAGE ASSEMBLY BYTE LENGTH



## S/3 BASIC INTERPRETER FIXED ADDRESS EQUATES

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00 25/09/15 PAGE 38
		0DC8	1619+I\$PUB1	EQU	X'0DC8'	IMAGE BUFFER 1 CORE ADDRESS
		0DCA	1620+I\$PUB2	EQU	X'0DCA'	IMAGE BUFFER 2 CORE ADDRESS
		0DCC	1621+I\$IMPT	EQU	X'0DCC'	IMAGE SCAN POINTER
		0DCE	1622+I\$IMC1	EQU	X'0DCE'	IMAGE CONV SPEC 1ST CHAR PNTR
		0DD0	1623+I\$SDPT	EQU	X'0DD0'	IMAGE CONV SPEC DECIMAL POINT PN
		0D5A	1624+I\$CSCT	EQU	X'0D5A'	IMAGE CONV SPEC COUNTERS
		0D58	1625+I\$SSCT	EQU	X'0D58'	IMAGE CONV SPEC CHAR COUNT
		0D59	1626+I\$SDCT	EQU	X'0D59'	IMAGE CONV SPEC DIGIT COUNT
		0D5A	1627+I\$SFCT	EQU	X'0D5A'	IMAGE CONV SPEC FRACTION COUNT
		0D5B	1628+I\$SICT	EQU	X'0D5B'	IMAGE CONV SPEC INTEGER COUNT
		0D5B	1629+I\$ADJX	EQU	X'0D5B'	ADJUSTED EXPONENT FOR E-FORMAT
			1630+*			
		0DD1	1631+I\$FACT	EQU	X'0DD1'	FUNCTION ACTIVITY TABLE BASE ADD
		0DE6	1632+I\$FATE	EQU	X'0DE6'	FUNCTION ACTIVITY TABLE END ADDR
		0DE8	1633+I\$FATP	EQU	X'0DE8'	FUNCTION ACTIVITY TABLE POINTER
			1634+*			
		0D5D	1635+I\$SFFO	EQU	X'0D5D'	FILE EXEC RTNS OVERLAY CADDR
			1636+*			
		119D	1637+I\$BSET	EQU	X'119D'	ENTRY - SET BRANCH EXEC ADDR
		117B	1638+I\$BRCN	EQU	X'117B'	BRANCH CONDITION STATUS BYTE
			1639+*			
		0B50	1640+I\$STCK	EQU	X'0B50'	ENTRY - STACK VM ELEMENT
		0BA2	1641+I\$SLNG	EQU	X'0BA2'	ELEMENT LENGTH INPUT PARAMETER
		0BA1	1642+I\$SLLC	EQU	X'0BA1'	STACKED ELEMENT LENGTH PARM
			1643+*			
		0BB0	1644+I\$USTK	EQU	X'0BB0'	ENTRY - UNSTACK VM ELEMENT
		0C3A	1645+I\$ULNG	EQU	X'0C3A'	ELEMENT LENGTH INPUT PARAMETER
		0BC1	1646+I\$DMSW	EQU	X'0BC1'	DATA TYPE MATCHING SWITCH
			1647+*			
		0A27	1648+I\$CPUF	EQU	X'0A27'	ENTRY - FLOATING POINT UNPACKER
		0A85	1649+I\$CUPF	EQU	X'0A85'	ENTRY - FLOATING POINT PACKER
		0AE3	1650+I\$CFBS	EQU	X'0AE3'	ENTRY - FLT TO BIN SUBSC CONV
		075D	1651+I\$FADD	EQU	X'075D'	ENTRY - FLOATING ADDITION RTN
		0751	1652+I\$FSUB	EQU	X'0751'	ENTRY - FLOATING SUBTRACT RTN
		082A	1653+I\$FMPY	EQU	X'082A'	ENTRY - FLOATING MULTIPLY RTN
		0919	1654+I\$FDVD	EQU	X'0919'	ENTRY - FLOATING DIVISION RTN
			1655+*			
		0E24	1656+I\$I700	EQU	X'0E24'	ENTRY - LINE PRINTER CONFIG. CK.
		130B	1657+I\$MOD4	EQU	X'130B'	STACK MODIFICATION ENTRY
			1658+*			
		1358	1659+I\$CVAD	EQU	X'1358'	ENTRY - PAGING RTN CONV VADDR
		1349	1660+I\$MDFY	EQU	X'1349'	ENTRY - CONVERT VADDR FOR MODIFY
		1354	1661+I\$LOCK	EQU	X'1354'	ENTRY - LOCK AND CONVERT VADDR
		1350	1662+I\$UNLK	EQU	X'1350'	ENTRY - UNLOCK A VIRTUAL PAGE
		1329	1663+I\$LDBR	EQU	X'1329'	ENTRY - CONVERT VADDR, LOAD BR
		1330	1664+I\$LDXR	EQU	X'1330'	ENTRY - CONVERT VADDR, LOAD XR
		12B1	1665+I\$CALL	EQU	X'12B1'	ENTRY - CALL VIRTUAL MEMORY RTN
		12B6	1666+I\$LBFR	EQU	X'12B6'	FORCE LINE PRINTER BUFFER UNLOCK
		12D3	1667+I\$RTRN	EQU	X'12D3'	ENTRY - RETURN FROM VM ROUTINE
		1449	1668+I\$PGNO	EQU	X'1449'	VIRTUAL PAGE INPUT PARAMETER
		144A	1669+I\$PGDS	EQU	X'144A'	VIRTUAL PAGE DISP INPUT PARM
		144A	1670+I\$VADR	EQU	X'144A'	VIRTUAL ADDRESS INPUT PARM
		144C	1671+I\$CADR	EQU	X'144C'	CORE ADDRESS OUTPUT PARAMETER
		14CA	1672+I\$PGTB	EQU	X'14CA'	PAGE REFERENCE TABLE CORE ADDR
		15E2	1673+I\$PLRT	EQU	X'15E2'	CORE PAGE STATUS TABLES ADDR
		15CA	1674+I\$PSTK	EQU	X'15CA'	PAGE LINKAGE STACK CORE ADDR



## S/3 BASIC INTERPRETER FIXED ADDRESS EQUATES

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 25/09/15 PAGE 39
		1676+	*****		
		1677+	*	SPECIAL INTERPRETER EQUATES REFERENCING CONSTANTS	*
		1678+	*****		
		000A	1680+	I\$NCPG EQU 10	NO. OF 8K SYSTEM PAGES (MAX)
		1682+	*****		
		1683+	*	PRINT USING FUNCTION INDICATORS (USED WITH INDR BYTE I\$INDR)	*
		1684+	*****		
		0080	1686+	I\$NISW EQU X'80'	NULL IMAGE INDICATOR
			1687+	*	* 1 = CURRENT IMAGE NULL
		0040	1688+	I\$B1SW EQU X'40'	IMAGE BUFFER 1 INDICATOR
			1689+	*	* 1 = BUFFER 1 LOCKED IN CORE
		0020	1690+	I\$B2SW EQU X'20'	IMAGE BUFFER 2 INDICATOR
			1691+	*	* 1 = BUFFER 2 LOCKED IN CORE
		0010	1692+	I\$CSSW EQU X'10'	CONVERSION SPECIFICATION INDR
			1693+	*	* 1 = AT LEAST 1 C/S IN IMAGE
		0008	1694+	I\$STSW EQU X'08'	CHARACTER STRING INDICATOR
			1695+	*	* 1 = CONV SPEC NOT FILLED
		0004	1696+	I\$ECSW EQU X'04'	E-FORMAT CONVERSION INDICATOR
			1697+	*	* 1 = E-FORMAT CONV SPEC
		0002	1698+	I\$NDSW EQU X'02'	END-OF-IMAGE INDICATOR
			1699+	*	* 1 = IMAGE END ENCOUNTERED
		0001	1700+	I\$SNSW EQU X'01'	IMPLIED NEGATIVE SIGN INDR
			1701+	*	* 1 = NEG VALUE W/ NO C/S SIGN
			1702+	*	
				END OF SYSTEM PROGRAM FILE EQUATES	
			1703+	PRINT ON	
0000			1704	ORG X'0000'	

## DKDISK - MODULE PROLOG

ERR	LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00	25/09/15	PAGE 40
			1706	*	*****			
			1707	*				*
			1708	*	5703-XM1 COPYRIGHT IBM CORP. 1970			*
			1709	*	REFER TO INSTRUCTIONS ON COPYRIGHT NOTICE 120-2083			*
			1710	*				*
			1711	*	*****			*
			1712	*	*STATUS			*
			1713	*	VERSION 1 MODIFICATION 0			*
			1714	*				*
			1715	*	*FUNCTION			*
			1716	*	DKDISK PROVIDES THE PHYSICAL I/O ON THE DOLPHIN ATTACHMENT.			*
			1717	*	I/O FUNCTIONS INCLUDING SEEKING, READING, WRITE-VERIFYING AND			*
			1718	*	ERROR RECOVERY. ALL FUNCTIONS PROVIDE FOR I/O OVERLAP.			*
			1719	*				*
			1720	*	*ENTRY POINTS			*
			1721	*	DKDISK (\$DISKN)			*
			1722	*	ONE ENTRY POINT FOR ALL FUNCTIONS IS PROVIDED. THE CALLING			*
			1723	*	SEQUENCE IS:			*
			1724	*	B \$DISKN			*
			1725	*	DC AL2(DPLA)			*
			1726	*	WHERE DPLA IS THE ADDRESS OF THE LEFT BYTE OF A 6 BYTE PARAMETER			*
			1727	*	LIST.			*
			1728	*				*
			1729	*	*INPUT			*
			1730	*	INPUT TO DKDISK IS A 6 BYTE PARAMETER LIST WITH THE FOLLOWING			*
			1731	*	FORMAT			*
			1732	*	BYTE 0 X'00' SEEK CONTROL.			*
			1733	*	X'01' READ DATA CONTROL.			*
			1734	*	X'02' WRITE AND VERIFY CONTROL.			*
			1735	*	X'FF' WAIT AND CHECK FOR ERRORS CONTROL.			*
			1736	*	BYTE 1 CYLINDER ADDRESS			*
			1737	*	BYTE 2 HEAD, SECTOR AND DRIVE ADDRESSES.			*
			1738	*	BYTE 3 SECTOR COUNT.			*
			1739	*	BYTES 4 & 5 CORE ADDRESS OF I/O BUFFER.			*
			1740	*	BYTES 3-5 ARE NOT REQUIRED FOR SEEK FUNCTIONS.			*
			1741	*	BYTES 1-5 ARE NOT REQUIRED FOR WAIT FUNCTIONS.			*
			1742	*	FOR WRITE FUNCTIONS, THE DATA TO BE WRITTEN MUST REMAIN			*
			1743	*	UNCHANGED UNTIL A WAIT FUNCTION IS ISSUED.			*
			1744	*				*
			1745	*	*OUTPUT			*
			1746	*	INDEX REGISTERS 1 AND 2 ARE RESTORED UPON EXIT. THE PREVIOUS			*
			1747	*	(IF ANY) DISK OPERATION IS COMPLETED AND THE CURRENT ONE (IF ANY)			*
			1748	*	STARTED.			*
			1749	*				*
			1750	*	*EXTERNAL REFERENCES			*
			1751	*	\$CIMSK - INQUIRY REQUEST MASK INDICATOR.			*
			1752	*	\$PLST1 - I/O PARAMETER LIST SAVE AREA.			*
			1753	*	\$PLST2 - I/O PARAMETER LIST SAVE AREA.			*
			1754	*	\$PLST3 - I/O PARAMETER LIST SAVE AREA.			*
			1755	*	\$INDR2 - I/O ERROR PENDING INDICATOR.			*
			1756	*	\$ERLOG - ENTRY TO LOG I/O ERROR.			*
			1757	*	\$PKERT - INDIVIDUAL DISK READ/WRITE COUNTERS.			*
			1758	*	\$HISTE - OBR ENTRY.			*
			1759	*	\$IOIND - HARD I/O ERROR INDICATOR.			*
			1760	*				*
			1761	*	*EXITS, NORMAL			*

## DKDISK - MODULE PROLOG

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 25/09/15 PAGE 41
		1762	*	NORMAL EXIT IS TO THE INSTRUCTION FOLLOWING THE IN-LINE DPL	*
		1763	*	ADDRESS CONSTANT. THE GENERAL REGISTERS ARE RESTORED.	*
		1764	*		*
		1765	*	*EXITS, ERROR	*
		1766	*	IF AN I/O ERROR HAS OCCURRED, EXIT IS MADE TO NERLOG FOR ERROR	*
		1767	*	LOGGING. THE ONE EXCEPTION IS AN UNRECOVERABLE EQUIPMENT CHECK.	*
		1768	*	IN THIS CASE NO EXIT IS MADE.	*
		1769	*		*
		1770	*	*TABLES/WORK AREAS	*
		1771	*	N/A	*
		1772	*		*
		1773	*	*ATTRIBUTES	*
		1774	*	RELOCATABLE	*
		1775	*	CORE RESIDENT	*
		1776	*		*
		1777	*	*CHARACTER CODE DEPENDENCY	*
		1778	*	N/A	*
		1779	*		*
		1780	*	*NOTES	*
		1781	*	ERROR PROCEDURES	*
		1782	*	A COMPLETE ERROR RECOVERY PROCEDURE (ERP) IS INCL. IN DFDISK	*
		1783	*	TO FACILITATE RECOVERY FROM DISK ERROR. AN INITIAL DISK ERROR	*
		1784	*	WILL RESULT IN AN OBR (ERROR LOG) ENTRY FOR THE PARTICULAR	*
		1785	*	ERROR BEING PLACED AT LOCATION \$HISTE. ERROR RECOVERY IS THEN	*
		1786	*	BROKEN DOWN INTO THE FOLLOWING PROCEDURES.	*
		1787	*	ACTION 1 SET HARD ERROR INDICATOR AND EXIT TO NERLOG.	*
		1788	*	ACTION 2 REISSUE THE DISK OPERATION. ON THE SECOND	*
		1789	*	OCCURRENCE EXECUTE A HARD HALT.	*
		1790	*	ACTION 4 READ THE CURRENT TRACK ID. IF AN ERROR OCCURS	*
		1791	*	PERFORM ACTION 7 FOR THE ORIGINAL OPERATION.	*
		1792	*	CHECK TO DETERMINE IF POSITIONED ON THE CORRECT	*
		1793	*	TRACK. IF SO, DO ACTION 7.	*
		1794	*	CHECK TO DETERMINE IF HEAD SWITCHING FROM AN	*
		1795	*	ALTERNATE TRACK HAS TAKEN PLACE. IF SO SET TRACK	*
		1796	*	ADDRESS FOR NEXT GOOD TRACK AND PERFORM ACTION 5.	*
		1797	*	CHECK IF THIS TRACK HAS BEEN FLAGGED DEFECTIVE.	*
		1798	*	IF SO, SET ALTERNATE TRACK ADDRESS AND CONTINUE	*
		1799	*	THE ORIGINAL OPERATION ON THE ALTERNATE.	*
		1800	*	IF NOT DEFECTIVE DO ACTION 6.	*
		1801	*	ACTION 5 DECREASE THE NUMBER OF SECTORS BY THE NUMBER OF	*
		1802	*	SECTORS SUCCESSFULLY COMPLETED. CALCULATE THE	*
		1803	*	NEXT SECTOR AND DATA ADDRESSES. SEEK TO THE	*
		1804	*	DESIRED TRACK AND RESTART THE OPERATION.	*
		1805	*	ACTION 6 RECALIBRATE TO CYLINDER 0.	*
		1806	*	SEEK TO THE ORIGINAL TRACK.	*
		1807	*	PERFORM ACTION 7 IF REQUEST WAS NOT FOR SEEK.	*
		1808	*	IF THE ERROR PERSISTS, REPEAT THE ABOVE.	*
		1809	*	AFTER 16 UNSUCCESSFUL ATTEMPS, DO ACTION 1.	*
		1810	*	ACTION 7 IF THE OPERATION WAS FOR A WRITE-VERIFY, REPEAT	*
		1811	*	THE OPERATION. EIGHT FAILURES CAUSE EXECUTION OF	*
		1812	*	ACTION 1. IF OTHER THAN A WRITE-VERIFY, REPEAT	*
		1813	*	THE OPERATION UP TO 16 TIMES, THEN DO ACTION 6.	*
		1814	*	ACTION 8 REPEAT ORIGINAL OPERATION.	*
		1815	*		*
		1816	*	A LIST OF ERRORS AND THEIR INITIAL ACTIONS FOLLOW:	*
		1817	*	EQUIPMENT CHECK PERFORM ACTION 8	*

## DKDISK - MODULE PROLOG

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00	25/09/15	PAGE 42
		1818	*	INTERVENTION REQUIRED	PERFORM ACTION 4		*
		1819	*	OVERRUN	PERFORM ACTION 4		*
		1820	*	NO RECORD FOUND	PERFORM ACTION 4		*
		1821	*	MISSING ADDRESS MARK	PERFORM ACTION 4		*
		1822	*	DATA CHECK	PERFORM ACTION 4		*
		1823	*	TRACK CONDITION CHECK	PERFORM ACTION 4		*
		1824	*	SEEK CHECK	PERFORM ACTION 6		*
		1825	*	END OF CYLINDER	PERFORM ACTION 5		*
		1826	*	UNRECOGNIZABLE ERROR	PERFORM ACTION 1		*
		1827	*				*
		1828	*	REGISTER USAGE			*
		1829	*	INDEX REGISTER 1 AND 2 ARE USED FOR BASE ADDRESSING.			*
		1830	*				*
		1831	*	SAVED/RESTORED AREAS			*
		1832	*	N/A			*
		1833	*				*
		1834	*	MODIFICATION CONSIDERATIONS			*
		1835	*	* DKDISK HAS BEEN OPTIMISED FOR FUNCTION AND CORE REQUIREMENTS.			*
		1836	*	VARIOUS CODING TECHNIQUES WERE APPLIED TO MINIMIZE PHYSICAL			*
		1837	*	SIZE. THE DPL WAS DESIGNED FOR OPTIMUM INTERFACE WITH THE			*
		1838	*	DOLPHIN ATTACHMENT.			*
		1839	*	* SEVERAL SYSTEM MODULES (PRIMARILY UINITL AND UATRCK) WERE			*
		1840	*	DESIGNED TO DIRECTLY USE CERTAIN INTERNAL LOGICAL SECTIONS			*
		1841	*	OF DKDSIK. MODIFICATIONS TO THESE AREAS WILL MOST LIKELY			*
		1842	*	IMPACT THESE MODULES. THE FOLLOWING TABLE LISTS THE LABELS			*
		1843	*	OF VARIOUS INSTRUCTION INTERNAL TO DKDISK USED BY OTHER			*
		1844	*	MODULES.			*
		1845	*	DKD030 THIS INSTRUCTION NORMALLY RESETS THE FLAG BYTE			*
		1846	*	IN THE DCF BACK TO A NORMAL TRACK INDICATION.			*
		1847	*	IT IS A SBF INSTRUCTION SO THAT IT CAN BE NOP'D			*
		1848	*	UINITL.			*
		1849	*	DKDDPL THIS FIELD IS REFERENCED BY NAME BY UINITL AND			*
		1850	*	UATRCK.			*
		1851	*	DKDDDR SAME AS ABOVE.			*
		1852	*	DKDCF1 SAME AS ABOVE.			*
		1853	*	DKDSTA REFERENCED BY UATRCK.			*
		1854	*	DKDRST THIS SENSE BYTE IS MODIFIED BY UATRCK TO DUMMY			*
		1855	*	UP A SEEK ERROR.			*
		1856	*	DKD250 DKDISK IS REENTERED HERE TO FACILITATE SEEK ERROR			*
		1857	*	RECOVERY BY UATRCK.			*
		1858	*	DKD270 THIS INSTRUCTION IS MODIFIED BY UINETL TO CAUSE A			*
		1859	*	RETURN TO A SPECIAL ERP CONTAINED WITHIN UINITL.			*
		1860	*	DKD290 THIS INSTRUCTION IS MODIFIED BY MINITL, UINITL,			*
		1861	*	AND UATRCK TO TRAP HARD DISK ERRORS, THE BRANCH			*
		1862	*	ADDRESS IS CHANGED TO EFFECT A RETURN TO THE			*
		1863	*	MODIFYING PROGRAM.			*
		1864	*	DKDAC4 THIS ADDRESS IS REFERENCED BY NAME TO RESTORE			*
		1865	*	THE INSTRUCTION AT DKD270.			*
		1866	*	* THE TECHNIQUE USED IN DEFINING THE DPL CONTROL BYTE IS USED			*
		1867	*	BY UINITL AND UATRCK FOR GENERATING CONTROL CODES FOR			*
		1868	*	READING AND WRITING TRACK IDS. THESE CODES ARE X'11' FOR			*
		1869	*	READ-ID AND X'12' FOR WRITE-ID.			*
		1870	*				*
		1871	*	REQUIRED MODULES			*
		1872	*	@SYSEQ - GENERAL SYSTEM EQUATES.			*
		1873	*	@FXDEQ - NUCLEUS LOCATION EQUATES.			*

## DKDISK - MODULE PROLOG

ERR	LOC	OBJECT CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	25/09/15	PAGE 43
				1874	*	@HDWEQ - HARDWARE VALUE EQUATES.			*
				1875	*	@SPFEQ - SYSTEM PROGRAM FILE.			*
				1876	*	@CANEQ - COMMON CORE LOCATIONS.			*
				1877	*	@CY0EQ - CYLINDER ZERO EQUATES.			*
				1878	*	@VOLEQ - VOLUME LABEL EQUATES.			*
				1879	*	@HLTEQ - HALT CODES EQUATES.			*
				1880	*				*
				1881	*	OTHER			*
				1882	*	NONE			*
				1883	*				*
				1884	*	*****			*
0000				1886		ORG X'0000'			IT ALL START AT ZERO
0000	C0 87 0600			1887	DKD010 B	MOPPET			WILL BRANCH TO DUMPC
				1888	*				* MODIFIED BY NIPPER
0004	F5F7F0F360E7D4F1	0024	1889		DC	CL33'5703-XM1 COPYRIGHT IBM CORP. 1970'			
		0004	1890	DKDTLN	EQU	4			SIZE OF ERROR STATUS FIELD
		007F	1891	DKDX7F	EQU	X'7F'			MASK FOR SECTOR ADDRESS
		0080	1892	DKDX80	EQU	X'80'			MASK FOR HEAD BIT IN DADDR
		00FF	1893	DKDXFF	EQU	X'FF'			MASK FOR WAIT ONLY CALL
		0003	1894	DKDX03	EQU	X'03'			MASK FOR DEVC BITS IN SCTR ADDR
		00FC	1895	DKDXFC	EQU	X'FC'			MASK FOR SECTOR ADDR
		005C	1896	DKDETR	EQU	X'5C'			BITS ON -> LAST SCTR ON TRACK
			1897	*					
			1898	***	ERP RETRY COUNTS				
			1899	*					
		0002	1900	DKDCNE	EQU	2			EQUIPMENT CHECK RETRY COUNT
		0002	1901	DKDCCE	EQU	2			TRACK COND CHECK RETRY COUNT
		0010	1902	DKDKCE	EQU	16			SEEK RETRY COUNT
		0008	1903	DKDRCE	EQU	8			WRITE RETRY COUNT
		0010	1904	DKDDCE	EQU	16			READ RETRY COUNT
			1906	*	*****				*
			1907	*	SUBROUTINE ENTRY				*
			1908	*	*****				*
		0025	1909	DKDISK	EQU	*			ENTRY TO IOCR
		00D6	1910		USING DKDBS2,@BR				BASE REGISTER VALUE
			1911	*					
			1912	***	SUBROUTINE INITIALIZATION				
			1913	*					
0025	34 01 00E5		1914		ST DKD100+@OP1,@BR				SAVE BASE REGISTER
0029	C2 01 00D6		1915		LA DKDBS2,@BR				LOAD BASE REGISTER
002D	74 02 13		1916		ST DKD110+@OP1(,@BR),@XR				SAVE OTHER XR
0030	76 08 38		1917		A DKDONE(,@BR),@ARR				CALC ADDR OF POINTER TO DPL
0033	34 08 0067		1918		ST DKD040+@OP1,@ARR				SAVE ADDR OF ADDR OF PARM LIST
0037	76 08 38		1919		A DKDONE(,@BR),@ARR				CALC RETURN ADDR
003A	74 08 17		1920		ST DKD120+@OP1(,@BR),@ARR				SAVE RETURN ADDR
003D	0C 04 02F1 02F6		1921		MVC DKDTCT(DKDOCT),DKDNIT				INITIALIZE ERROR RETRY COUNTER
			1922	*					
			1923	***	MASK CONSOLE INTERRUPT WHILE IN DISK ROUTINE				
			1924	*					
0043	4C 00 01 0476		1925		MVC DKD080+@Q(1,@BR),%CIMSK				SAVE CURRENT MASK STATUS
0048	3C 80 0476		1926		MVI %CIMSK,@NOP				MASK CONSOLE INTERRUPT
			1927	*					
			1928	***	WAIT FOR LAST READ/WRITE TO BE COMPLETED AND CHECK FOR ERRORS				
			1929	*					

## DKDISK - MODULE PROLOG

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00 25/09/15 PAGE 44
	004C	D0	87 44		1930	DKD020	B DKD130(, @BR)	TEST FOR ERRORS
				004F	1931	DKD025	EQU *	RETURN FROM ERP
	004F	7B	FF 1F		1932	DKD030	SBF DKDCF1+@PFLAG(, @BR), DKDXFF	SET NORMAL TRACK OPERATION
					1933	*		* ABOVE INST CHANGED BY #UINIT
	0052	7D	02 19		1934		CLI DKDDPL+@DCTRL(, @BR), @DPUT	WAS LAST OPERATION A WRITE ?
	0055	F2	01 0C		1935		JNE DKD040	JUMP TO DO NEXT OP IF NO
	0058	5C	06 1E 35		1936		MVC DKDDPL+@DBFR2(@DPLNG+1, @BR), DKDST2(, @BR)	SET OP TO VERIFY
	005C	5E	00 65 3A		1937		ALC DKD170+@D1(1, @BR), DKDWRF(, @BR)	SET POINTER TO READ CNTR
	0060	C0	87 032A		1938		B DKD420	GO DO VERIFY
					1939	*		
					1940	***	GET DISK PARAMETER LIST AND SET UP NEXT OP LIST	
					1941	*		
	0064	35	02 0000		1942	DKD040	L *-*, @XR	XR = ADDR OF DPL
	0068	5C	00 26 28		1943	DKD050	MVC DKDRMA(1, @BR), DKDRMG(, @BR)	SET CURRENT ARM POSITION
	006C	6C	05 1E 05		1944		MVC DKDDPL+@DBFR2(@DPLNG, @BR), @DBFR2(, @XR)	SAVE NEW DPL
	0070	7D	FF 19		1945		CLI DKDDPL+@DCTRL(, @BR), DKDXFF	IS THIS WAIT ONLY OPERATION ?
	0073	F2	81 60		1946		JE DKD080	BRANCH IF YES
	0076	0C	0D 0462 045B		1947		MVC \$PLST3(2*@DPLNG+2), \$PLST2	PUSH DOWN PARM LIST STACK
	007C	1C	06 0454 1E		1948		MVC \$PLST1(@DPLNG+1), DKDDPL+@DBFR2(, @BR)	MOVE LIST TO STACK
	0081	5F	00 1C 38		1949		SLC DKDDPL+@DCNT(1, @BR), DKDONE(, @BR)	GET TRUE HDW COUNT
	0085	58	02 18 19		1950		MNZ DKDSAV(, @BR), DKDDPL+@DCTRL(, @BR)	SAVE SIO CNTL CODE
	0089	7B	F0 19		1951		SBF DKDDPL+@DCTRL(, @BR), X'F0'	CLEAR CNTL BITS IN DPL
					1952	*		
					1953	***	SET CURRENT DISK ARM POSITION FOR SPECIFIED SPINDLE	
					1954	*		
	008C	3C	26 009D		1955		MVI DKD060+@DD2, DKDRMA-DKDBS2	SET UP MOVE OF ARM A
	0090	78	02 1B		1956		TBN DKDDPL+@DSAD(, @BR), @DSPIN	IS THIS SPINDLE A SELECTED ?
	0093	F2	90 04		1957		JF DKD060	BRANCH IF YES
	0096	3C	27 009D		1958		MVI DKD060+@DD2, DKDRMB-DKDBS2	SET UP MOVE OF ARM B
					1959	*		
	009A	5C	00 28 00		1960	DKD060	MVC DKDRMG(1, @BR), *-*(, @BR)	SET CURRENT ARM POSITION
	009E	0C	00 006A 009D		1961		MVC DKD050+@D1(1), DKD060+@DD2	SET UP EXIT INSTRUCTION
					1962	*		
					1963	***	CALCULATE DEVC ADDRESS	
					1964	*		
	00A4	5C	00 29 1B		1965		MVC DKDDDR(1, @BR), DKDDPL+@DSAD(, @BR)	MOVE SECTOR ADDR BYTE
	00A8	7B	FC 29		1966		SBF DKDDDR(, @BR), DKDXFC	MASK OUT SECTOR ADDR
	00AB	7B	03 1B		1967		SBF DKDDPL+@DSAD(, @BR), DKDX03	MASK OUT DEVC SPEC BITS
	00AE	5C	04 35 1E		1968		MVC DKDST2(@DBFR2, @BR), DKDDPL+@DBFR2(, @BR)	SAVE LIST FOR VERIFY
	00B2	5E	00 29 29		1969		ALC DKDDDR(1, @BR), DKDDDR(, @BR)	DOUBLE SPINDLE/DISK SPECIFIED
	00B6	5E	00 29 29		1970		ALC DKDDDR(1, @BR), DKDDDR(, @BR)	SHIFT SPINDLE/DISK BITS
	00BA	5C	00 65 29		1971		MVC DKD170+@D1(1, @BR), DKDDDR(, @BR)	SET DISP TO PACK ERR RATE
	00BE	7D	02 19		1972		CLI DKDDPL+@DCTRL(, @BR), @DPUT	IS FUNCTION A WRITE ?
	00C1	F2	81 04		1973		JE DKD070	SKIP POINTER MODIFICATION
	00C4	5E	00 65 3A		1974		ALC DKD170+@D1(1, @BR), DKDWRF(, @BR)	POINT TO READ CNTL
	00C8	5E	00 29 29		1975	DKD070	ALC DKDDDR(1, @BR), DKDDDR(, @BR)	SHIFT SPINDLE/DISK BITS
	00CC	5E	00 29 3E		1976		ALC DKDDDR(1, @BR), DKDADR(, @BR)	CALC DEVICE ADDRESS
					1977	*		
					1978	***	EXECUTE INITIAL SEEK AND START DATA OPERATION IF DESIRED	
					1979	*		
	00D0	D0	87 81		1980		B DKDSEE(, @BR)	GO EXECUTE SEEK
	00D3	D0	87 56		1981		B DKD160(, @BR)	BRANCH TO START NEW OP IF NO
					1982	*		
					1983	***	UPDATE CURRENT CYLINDER POSITION	
					1984	*		
	00D6				1985	DKDBS2	EQU *	BASE VALUE FOR CALL SECTION



DKDISK - MODULE PROLOG

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15,	MOD 00	25/09/15	PAGE	45
					1986	*						
					1987	***	UNMASK CONSOLE INTERRUPT AND CHECK FOR RECORDED INTERRUPT					
					1988	*						
00D6	3C	00	0476		1989	DKD080	MVI \$CIMSK,*-*				RESTORE MASK STATUS	
00DA	38	04	03D5		1990		TBN \$INDR2,\$ERPND				IS THERE AN ERROR TO BE LOGGED ?	
00DE	C0	10	0345		1991	DKD090	BT \$ERLOG				IF YES DO ERROR LOGGING	
					1992	*						
					1993	***	EXIT FROM SUBROUTINE					
					1994	*						
00E2	C2	01	0000		1995	DKD100	LA *-*,@BR				RESTORE BASE REGISTER	
00E6	C2	02	0000		1996	DKD110	LA *-*,@XR				RESTORE XR	
00EA	C0	FF	0000		1997	DKD120	BC *-*,X'FF'				RETURN TO CALLER	



## DKDISK - MODULE PROLOG

ERR	LOC	OBJECT CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00 25/09/15 PAGE 46
				1999	*****	*****	
				2000	*	WORK AREAS	*
				2001	*****	*****	
			00EE	2002	DKDSAV EQU	* LEFT BYTE OF SAVE AREA	
00EE			00EE	2003	DS	CL1 SAVE AREA FOR SIO CNTL BYTE	
00EE				2004	ORG	DKDSAV SET INSTR COUNTER	
00EE	00		00EE	2005	DC	XL1 '00' INITIALLY ZERO	
				2006	*		
				2007	***	SAVE AREA FOR CURRENT DISK PARAMETER LIST	
				2008	*		
			00EF	2009	DKDDPL EQU	* LEFT BYTE OF SAVE AREA	
00EF			00F4	2010	DS	CL(@DPLNG) PARAMETER LIST SAVE AREA	
00EF				2011	ORG	DKDDPL SET INITIAL VALUE	
00EF	00000000000000		00F4	2012	DC	XL(@DPLNG) '00' INITIALLY ZERO	
				2013	*		
				2014	***	DISK CONTROL FIELD	
				2015	*		
			00F5	2016	DKDCF1 EQU	* LEFT BYTE OF DCF	
00F5			00F8	2017	DS	CL(@DCFLN) DISK CONTROL FIELD	
00F5				2018	ORG	DKDCF1 SET INITIAL VALUE	
00F5	00000000		00F8	2019	DC	XL(@DCFLN) '00' INITIALLY ZERO	
00F9			00FB	2020	DKDZZZ DS	XL(@DCFLN-1) SAVE AREA FOR DCF	
				2021	*		
				2022	***	CURRENT CYL POSITIONING OF DISK ARMS	
				2023	*		
			00FC	2024	DKDRMA DS	CL1 POSITION OF SINDLE A ARM	
00FC				2025	ORG	DKDRMA SET INITIAL VALUE	
00FC	00		00FC	2026	DC	XL1 '00' INITIALLY AT CYL 0	
			00FC	2027	DKDMPS EQU	DKDRMA POSITION OF DRIVE 1 ARM	
				2029	DKDRMB DS	CL1 POSITION OF SINDLE B ARM	
00FD				2030	ORG	DKDRMB SET INITIAL VALUE	
00FD	FF		00FD	2031	DC	XL1 'FF' INITIALLY SET TO CAUSE RECAL.	
				2033	DKDRMG DS	CL1 CYL POSITION FOR CURRENT DISK	
00FE			00FE	2034	ORG	DKDRMG SET INITAIL VALUE	
00FE	00		00FE	2035	DC	XL1 '00' INITIALLY ZERO	
				2036	*		
				2037	***	CURRENT SPINDLE/DISK ADDRESS	
				2038	*		
			00FF	2039	DKDDDR DS	CL1 SAVE AREA FOR DEVC ADDRESS	
00FF				2040	ORG	DKDDDR SET INITIAL VALUE	
00FF	A0		00FF	2041	DC	AL1(@SPINA) SPINDLE A ADDRESS	
				2042	*		
				2043	***	CURRENT ERROR STATUS SAVE AREA	
				2044	*		
			0100	2045	DKDRST EQU	* LEFT BYTE OF SNS - BYTE 0	
0100			0103	2046	DS	CL(DKDTLN) CURRENT ERROR STATUS SAVE	
0104	00		0104	2047	DKDOFT DC	AL1(@DHARD) HISTORY LOG TYPE INDR	
0105	0301		0106	2048	DC	XL2 '0301' VERIFY CONTROL	
0107			010B	2049	DKDST2 DS	CL(@DBFR2) SAVE AREA FOR VERIFY LIST	
				2050	*		

DISK IOCS, COMMON STORAGE AND CONSTANT AREA

ERR LOC		OBJECT CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	25/09/15	PAGE	47
				2052	*****					
				2053	*	CONSTANTS				*
				2054	*****					
			010C	2055	DKDEST	EQU	*			START OF SEEK ZERO CYLS DCF TO
				2056	*					* RESET ERROR LATCH ON DRIVE 1
	010C	000001		2057	DKDONE	DC	XL3'000001'			CONSTANT 1
			010E	2058	DKDRTR	EQU	DKDONE			ERROR RETRY DECREMENT
	010F	0002		2059	DKDWRF	DC	XL2'0002'			CNTR, FUNC CODES FOR WRITE OPER.
	0111	00F5		2060	DKDCFP	DC	AL2(DKDCF1)			POINTER TO DISK CONTROL FIELD
	0113	00A0		2061	DKDADR	DC	AL2(@SPINA)			DEVC ADDR OF SPINDLE A (FIXED)
	0115	0000FF		2062	DKDECL	DC	XL3'0000FF'			RECALIBRATE SEEK DCF
				2063	*					
	0118	010C		2064	DKDSTA	DC	AL2(DKDEST)			ADDR OF RESET ERROR DCF
				2065	*					

## DISK IOCS, COMMON ROUTINES

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00 25/09/15 PAGE 48
					2067	*****	*****	
					2068	*	ROUTINE TO CHECK FOR ERRORS	*
					2069	*****	*****	
011A	74	08	55		2070	DKD130	ST DKD150+@OP1(,@BR),@ARR	SAVE RETURN ADDRESS
011D	58	00	4F 29		2071		MZZ DKD140+@Q(,@BR),DKDDDR(,@BR)	SET DEVC ADDR IN TIO
0121	F1	A2	00		2072		APL @SPINA+@DBUSY	WAIT UNTIL NOT BUSY
0124	C1	00	0189		2073	DKD140	TIO DKDERP,@DERR	TEST FOR ERROR, BR TO RECOVERY
0128	C0	87	0000		2074	DKD150	B *-*	RETURN TO CALLER
					2076	*****	*****	
					2077	*	ROUTINE TO EXECUTE NON-SEEK I/O OPERATION	*
					2078	*****	*****	
				00D6	2079		USING DKDBS2,@BR	BASE ADDR FOR DKOPER
012C	74	08	80		2080	DKD160	ST DKD190+@OP1(,@BR),@ARR	SAVE RETURN ADDR
012F	7D	00	19		2081		CLI DKDDPL+@DCTRL(,@BR),@DPOS	IS OPERATION A SEEK ?
0132	F2	81	1E		2082		JE DKD190	BRANCH IF YES
0135	C2	02	0417		2083		LA \$PKERT-2,@XR	POINT TO ERR/RATE TABLE
0139	9E	01	00 38		2084	DKD170	ALC *-*(2,@XR),DKDONE(,@BR)	BUMP ERROR RATE COUNTER
013D	71	A4	1E		2085		LIO DKDDPL+@DBFR2(,@BR),@DFDR+@SPINA	LOAD DFDR
0140	5C	02	22 1C		2086		MVC DKDCF1+@PCNT(@DCFLN-1,@BR),DKDDPL+@DCNT(,@BR)	SET UP DCF
0144	5C	00	7B 29		2087		MVC DKD180+@Q(1,@BR),DKDDDR(,@BR)	SET DEVC ADDR IN SIO
0148	5E	00	7B 19		2088		ALC DKD180+@Q(1,@BR),DKDDPL+@DCTRL(,@BR)	SET FUNC BITS IN SIO
014C	5C	00	7C 18		2089		MVC DKD180+@D1(1,@BR),DKDSAV(,@BR)	SET SIO CNTL BYTE
0150	F3	00	00		2090	DKD180	SIO *-*,*-*	EXECUTE I/O FUNCTION
0153	C0	87	0000		2091	DKD190	B *-*	RETURN TO CALLER
					2093	*****	*****	
					2094	*	ROUTINE TO SEEK TO TRACK SPECIFIED IN DPL	*
					2095	*****	*****	
				00D6	2096		USING DKDBS2,@BR	BASE ADDR FOR DSEEK
				0157	2097	DKDSEE	EQU *	ENTRY FOR NORMAL SEEK
0157	5C	01	21 1B		2098		MVC DKDCF1+@PSAD(@DADDR,@BR),DKDDPL+@DSAD(,@BR)	SET UP DCF
				015B	2099	DKDLSK	EQU *	ENTRY TO SEEK TO ALTERNATE TRACK
015B	74	08	B2		2100		ST DKD220+@OP1(,@BR),@ARR	SAVE RETURN ADDRESS
015E	5C	00	22 20		2101		MVC DKDCF1+@PCNT(1,@BR),DKDCF1+@PCYL(,@BR)	SET UP CYL CALC
0162	5F	00	22 28		2102		SLC DKDCF1+@PCNT(1,@BR),DKDRMG(,@BR)	SUB DISK ARM POSITION
0166	7A	01	21		2103		SBN DKDCF1+@PSAD(,@BR),@DCYMV	SET BIT FOR FORWARD MOVEMENT
0169	F2	02	0B		2104		JNL DKD200	BRANCH IF FORWARD MOVEMENT
					2105	*		
016C	5C	00	22 28		2106		MVC DKDCF1+@PCNT(1,@BR),DKDRMG(,@BR)	REVERSE SUBTRACT ORDER
0170	5F	00	22 20		2107		SLC DKDCF1+@PCNT(1,@BR),DKDCF1+@PCYL(,@BR)	CALC CYL DIFF
0174	7B	01	21		2108		SBF DKDCF1+@PSAD(,@BR),@DCYMV	SET BIT FOR REVERSE MOVEMENT
0177	5C	00	A9 29		2109	DKD200	MVC DKD210+@Q(1,@BR),DKDDDR(,@BR)	SET DEVC ADDRIN SIO SEEK
017B	71	A6	3C		2110		LIO DKDCFP(,@BR),@SPINA+@DFCR	LOAD DFCR
017E	F3	00	00		2111	DKD210	SIO @SKCTL,@DSEEK	EXECUTE SEEK
0181	5C	00	28 20		2112		MVC DKDRMG(1,@BR),DKDCF1+@PCYL(,@BR)	SET NEW ARM POSITION
0185	C0	87	0000		2113	DKD220	B *-*	RETURN TO CALLER
					2114	*		

## DKDISK - ERP SECTION

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00 25/09/15 PAGE 49
		2116	*****		*****	
		2117	*		THIS ROUTINE SENSES THE ERROR BITS. IF AN UNSAFE CONDITION IS	*
		2118	*		DETERMINED, DKUNSF IS CALLED TO EXECUTE THE HARD HALT.	*
		2119	*****		*****	
		00D6 2120			USING DKDBS2,@BR	CALL SECTION BASE VALUE
		0189 2121	DKDERP	EQU	*	ENTRY TO ERP SECTION
0189 58 00 BC 29		2122		MZZ	DKD230+@Q(,@BR),DKDDDR(,@BR)	SET DEVICE ADDR IN 1ST SNS
018D 58 00 BF 29		2123		MZZ	DKD240+@Q(,@BR),DKDDDR(,@BR)	SET DEVICE ADDR IN 2ND SNS
0191 70 02 2B		2124	DKD230	SNS	DKDRST+@SNSB1(,@BR),@DVST1	SENSE BYTES 0 & 1
0194 70 03 2D		2125	DKD240	SNS	DKDRST+@SNSB3(,@BR),@DVST2	SENSE BYTES 2 & 3
		2126	*			
0197 5C 02 25 22		2127	DKD250	MVC	DKDZZZ(@DADDR+1,@BR),DKDCF1+@PCNT(,@BR)	SAVE DCF INFO
		019B 2128	DKD265	EQU	*	ENTRY TO SKIP ERROR LOG SET UP
019B C2 02 01F4		2129		LA	DKDBS1,@XR	SET BASE REG FOR ERP
		2130	*			
		2131	***		BRANCH TO PROCESS ERROR	
		2132	*			
019F 78 10 2A		2133		TBN	DKDRST(,@BR),@DEREQ	EQUIPMENT CHECK ?
01A2 F2 10 34		2134		JT	DKDAC2	DO ACTION 2 IF YES
01A5 78 40 2A		2135		TBN	DKDRST(,@BR),@DERIN	INTERVENTION REQUIRED ?
01A8 C0 10 02F7		2136		BT	DKDAC8	DO ACTION 8 IF YES
01AC 78 04 2B		2137		TBN	DKDRST+@SNSB1(,@BR),@OVRUN	OVERRUN ?
01AF F2 10 33		2138		JT	DKDAC4	DO ACTION 4 IF YES
01B2 79 2E 2A		2139		TBF	DKDRST(,@BR),@DERNR+@DERTC+@DERD2+@DERMA	NO RCD FND,
		2140	*			* TRK COND CHK, MISSING ADDR
		2141	*			* MARK OR DATA CHK ?
01B5 C0 90 01E5		2142	DKD270	BF	DKDAC4	DO ACTION 4 IF YES
01B9 78 01 2A		2143		TBN	DKDRST(,@BR),@DERSC	SEEK CHECK ?
01BC F2 10 CF		2144		JT	DKDAC6	DO ACTION 6 IF YES
01BF 78 20 2B		2145		TBN	DKDRST+@SNSB1(,@BR),@DERCE	CYLINDER END ?
01C2 F2 10 9C		2146		JT	DKDAC5	DO ACTION 5 IF YES
		2147	*			
		2148	***		ERROR UNRECOGNIZABLE - FALL THROUGH TO DO ACTION 1	
		2149	*			

DKDISK - ACTION 1 - ERP SECTION

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	25/09/15	PAGE	50
		2151			*****				
		2152	*		ACTION 1 - HARD ERROR ROUTINE				*
		2153			*****				
		01C5	2154	DKDAC1	EQU *				ENTRY TO DO ACTION 1
01C5	3A 20 03D2	2155	DKD280	SBN	\$IOIND,\$HRDER				SET HARD ERROR INDICATOR
01C9	71 A6 43	2156		LIO	DKDSTA(,@BR),@SPINA+@DFCR				LOAD DFCR WITH SEEK 0 CYLS DCF
01CC	F3 A0 00	2157		SIO	@SKCTL,@DSEEK+@SPINA				RESET ERROR LATCH
01CF	7C 00 19	2158		MVI	DKDDPL+@PCTRL(,@BR),@DPOS				INSURE NO VERIFY OPERATION
01D2	F2 87 0C	2159		J	DKD295				SETUP ERROR ENTRY
01D5	C0 87 0345	2160	DKD290	B	\$ERLOG				GO LOG ERROR AND HALT
		2161	*						

DKDISK - ACTION 2 - ERP SECTION

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	25/09/15	PAGE	51
		2163			*****				
		2164	*		ACTION 2 - RETRY OPERATION ONCE				*
		2165			*****				
		00D6	2166		USING DKDBS2,@BR				CALL SECTION BASE VALUE
		01D9	2167	DKDAC2	EQU *				ENTRY TO DO ACTION 2
01D9	9F 00 F9 38		2168		SLC DKDERT-DKDBS1(1,@XR),DKDRTR(,@BR)				DECR EQUIP CHECK RETRY
01DD	C0 81 0337		2169		BZ DKDNSF				DO HARD EQUIPMENT CHECK ERROR
01E1	C0 87 02FD		2170	DKD295	B DKD410				GO RETRY OPERATION
		2171	*						

## DKDISK - ACTION 4 - ERP SECTION

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00 25/09/15 PAGE 52
					2173	*****	*****	
					2174	*	ACTION 4 - NO RECORD FOUND / TRACK CONDITION CHECK	*
					2175	*****	*****	
				01F4	2176		USING DKDBS1,@XR CALL SECTION BASE VALUE	
				01E5	2177	DKDAC4 EQU *	ENTRY TO DO ACTION 4	
01E5	9C	01	F2	21	2178	MVC	DKDSAD(@DADDR,@XR),DKDCF1+@PSAD-DKDBS2(,@BR)	
01E9	9C	03	ED	22	2179	MVC	DKDCF2+@PCNT(@DCFLN,@XR),DKDCF1+@PCNT(,@BR) SAVE PCF	
01ED	9C	00	01	29	2180	MVC	DKD300+@Q(1,@XR),DKDDDR-DKDBS2(,@BR) SET DEVICE ADDR	
01F1	BA	01	01		2181	SBN	DKD300+@Q(,@XR),@DREAD SET READ ID FUNCTION BIT	
				01F4	2182	DKDBS1 EQU *	BASE VALUE FOR ERP (XR)	
01F4	F3	00	01		2183	DKD300 SIO	@DCRID,*-* READ ID	
01F7	98	00	0B	29	2184	MZZ	DKD310+@Q(,@XR),DKDDDR-DKDBS2(,@BR) SET SPINDLE ADDR IN TIO	
01FB	F1	A2	00		2185	APL	@SPINA+@DBUSY WAIT FOR OPERATION COMPLETION	
01FE	E1	00	B8		2186	DKD310 TIO	DKDAC7(,@XR),@DERR JUMP TO ACTION 7 ON ERROR	
0201	9C	01	F4	21	2187	MVC	DKDDSV(@DADDR,@XR),DKDCF1+@PSAD-DKDBS2(,@BR) SAVE CURRENT	
					2188	*	* DISK ADDRESS	
0205	BB	7F	F2		2189	SBF	DKDSAD(,@XR),DKDX7F SET SECTOR BITS OFF	
0208	BB	7F	F4		2190	SBF	DKDDSV(,@XR),DKDX7F SET SECTOR BITS OFF	
020B	AD	01	F2	F4	2191	CLC	DKDSAD(@DADDR,@XR),DKDDSV(,@XR) IS THIS THE CORRECT TRK ?	
020F	F2	81	9A		2192	JE	DKDAC7 DO ACTION 7 IF YES	
0212	B0	A4	F6		2193	SNS	DKDFDR(,@XR),@SPINA+@DFDR SENSE DATA ADDR	
0215	5D	00	1C	22	2194	CLC	DKDDPL+@DCNT(1,@BR),DKDCF1+@PCNT(,@BR) 1ST SECTOR ERROR ?	
0219	F2	81	12		2195	JE	DKD320 JUMP IF YES	
021C	5E	00	22	38	2196	ALC	DKDCF1+@PCNT(1,@BR),DKDONE(,@BR) GET SECTOR COUNT	
0220	9E	00	ED	38	2197	ALC	DKDCF2+@PCNT(1,@XR),DKDONE(,@BR) GET SAVED COUNT TOO	
0224	7D	02	19		2198	CLI	DKDDPL+@DCTRL(,@BR),@DPUT WRITE OPERATION ?	
0227	F2	81	04		2199	JE	DKD320 JUMP IF YES	
022A	9F	00	F5	38	2200	SLC	DKDFDR-1(1,@XR),DKDONE(,@BR) DECREMENT DATA ADDR 1 SCTR	
022E	AD	01	EB	EF	2201	DKD320 CLC	DKDCF2+@PCYL(2,@XR),DKDCF3+@PCYL(,@XR) CHECK IF LAST	
					2202	*	* OPERATION WAS FOR ALTERNATE	
					2203	*	* ON CORRECT CYLINDER	
0232	F2	01	11		2204	JNE	DKD330 JUMP IF NO	
0235	AD	01	F0	F2	2205	CLC	DKDCF3+@PSAD(2,@XR),DKDSAD(,@XR) DID HEAD SWITCH OCCURE ?	
0239	F2	81	0A		2206	JE	DKD330 JUMP IF NOT	
023C	6C	01	1E	F6	2207	MVC	DKDDPL+@DBFR2(@CADDR,@BR),DKDFDR(,@XR) SET NEXT DATA ADDR	
0240	7C	01	1F		2208	MVI	DKDCF1+@PFLAG(,@BR),@ALTFL SET ALTERNATE INDICATOR	
0243	F2	87	2B		2209	J	DKD345 DO ACTION 5	
					2210	*		
					2211	***	CHECK FOR DEFECTIVE TRACK	
					2212	*		
0246	7D	02	1F		2213	DKD330 CLI	DKDCF1+@PFLAG-DKDBS2(,@BR),@DEFLG IS TRACK DEFECTIVE ?	
0249	F2	01	42		2214	JNE	DKDAC6 IF NO, DO ACTION 6	
024C	7C	01	1F		2215	MVI	DKDCF1+@PFLAG-DKDBS2(,@BR),@ALTFL SET FLAG FOR ALTERNATE	
024F	D0	87	85		2216	B	DKDLK-DKDBS2(,@BR) SEEK TO ALTERNATE	
0252	6C	02	1C	ED	2217	MVC	DKDDPL+@PCNT-DKDBS2(@DADDR+1,@BR),DKDCF2+@PCNT(,@XR) SET	
					2218	*	* UP NEW PARM LIST	
0256	AC	01	F0	F2	2219	MVC	DKDCF3+@PSAD(@DADDR,@XR),DKDSAD(,@XR) SAVE ORIGINAL DADDR	
025A	6C	01	1E	F6	2220	MVC	DKDDPL+@DBFR2(@CADDR,@BR),DKDFDR(,@XR) SET NEXT DATA ADDR	
025E	F2	87	CC		2221	J	DKD430 GO RETRY WITH ALTERNATE TRACK	
					2222	*		



## DKDISK - ACTION 5 - ERP SECTION

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 25/09/15 PAGE 53
		2224	*****	*****	
		2225	*	ACTION 5 - END OF CYLINDER PROCESSING	*
		2226	*****	*****	
		00D6 2227		USING DKDBS2,@BR	CALL SECTION BASE VALUE
		0261 2228	DKDAC5 EQU *		ENTRY TO DO ACTION 5
0261 70 A4 1E		2229	SNS	DKDDPL+@DBFR2(,@BR),@SPINA+@DFDR	SET NEXT DATA ADDRESS
0264 5D 00 1E 35		2230	CLC	DKDDPL+@DBFR2(1,@BR),DKDST2(,@BR)	OPERATION COMPLETE ?
0268 F2 01 41		2231	JNE	DKDAC7	NO, DO ACTION 7
026B 78 5C 21		2232	TBN	DKDCF1+@PSAD(,@BR),DKDETR	END OF TRACK ?
026E F2 90 3B		2233	JF	DKDAC7	NO, DO ACTION 7
0271 5C 00 1C 22		2234	DKD345 MVC	DKDDPL+@PCNT(1,@BR),DKDCF1+@PCNT(,@BR)	SET LIST COUNT
		2235	*		* FROM DCF
0275 7B 7F 1B		2236	SBF	DKDDPL+@PSAD(,@BR),DKDX7F	SET SECTOR BITS OFF
0278 6E 01 1B F8		2237	DKD350 ALC	DKDDPL+@PSAD(@DADDR,@BR),DKDDSW-DKDBS1(,@XR)	BUMP TRACK
027C 7D 01 1F		2238	CLI	DKDCF1+@PFLAG(,@BR),@ALTFL	ALTERNATE TRACK OPERATION ?
027F F2 81 06		2239	JE	DKD360	JUMP IF YES
0282 78 80 1B		2240	TBN	DKDDPL+@PSAD(,@BR),DKDX80	IS THIS START OF NEW CYLINDER ?
0285 E0 10 84		2241	BT	DKD350-DKDBS1(,@XR)	BRANCH IF NO
0288 7C 00 1F		2242	DKD360 MVI	DKDCF1+@PFLAG(,@BR),@NORFL	SET FLAG FOR NORMAL OPERATION
028B F2 87 9C		2243	J	DKD420	GO FINISH OPERATION
		2244	*		

DKDISK - ACTION 6 - ERP SECTION

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 25/09/15 PAGE 54
		2246		*****	
		2247	*	ACTION 6 - RESEEK TO DESIRED TRACK	*
		2248		*****	
		01F4 2249		USING DKDBS1,@XR	CALL SECTION BASE VALUE
		028E 2250	DKDAC6 EQU *		ENTRY TO DO ACTION 6
028E 9F 00 FB 38		2251	SLC	DKDKRT(1,@XR),DKDRTR-DKDBS2(,@BR)	DECR RESEEK COUNT
0292 F2 81 36		2252	JZ	DKD390	DO ACTION 1 IF NO MORE TRIES
0295 98 00 AA 29		2253	MZZ	DKD370+@Q(,@XR),DKDDDR-DKDBS2(,@BR)	SET DEVC ADDRESS
0299 5C 02 22 41		2254	MVC	DKDCF1+@PCNT-DKDBS2(DKDA03,@BR),DKDECL-DKDBS2(,@BR)	
029D F3 00 00		2255	DKD370 SIO	@SKCTL,@DSEEK	RECALIBRATE SEEK
02A0 7C 00 28		2256	MVI	DKDRMG-DKDBS2(,@BR),@ZERO	SET ARM POSITION TO ZERO
02A3 7D 00 19		2257	CLI	DKDDPL+@DCTRL-DKDBS2(,@BR),@DPOS	SEEK ONLY ?
02A6 F2 81 54		2258	JE	DKD410	IF YES GO RETRY SEEK
02A9 D0 87 81		2259	B	DKDSEE-DKDBS2(,@BR)	SEEK TO ORIGINAL CYLINDER
		2260	*		

## DKDISK - ACTION 7 - ERP SECTION

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 25/09/15 PAGE 55
		2262	*****		
		2263	*	ACTION 7 - RETRY READ/WRITE OPERATION	*
		2264	*****		
		01F4 2265		USING DKDBS1,@XR	CALL SECTION BASE VALUE
		02AC 2266	DKDAC7	EQU *	ENTRY TO DO ACTION 7
02AC 7D 01 19		2267		CLI DKDDPL+@DCTRL-DKDBS2(,@BR),@DREAD	IS IT A READ OP ?
02AF F2 01 12		2268		JNE DKD380	JUMP IF NO
		2269	*		
		2270	***	RETRY READ/VERIFY OPERATION	
		2271	*		
		2272	*		
		2273	***	CHECK FOR VERIFY OPERATION	
		2274	*		
02B2 7D 03 7C		2275		CLI DKD180+@D1-DKDBS2(,@BR),@DVERY	IS THIS A VERIFY OP ?
02B5 F2 01 17		2276		JNE DKD400	JUMP IF NO
		2277	*		
		2278	***	SET RETRY OF WRITE-VERIFY OPERATION	
		2279	*		
02B8 5C 04 1E 35		2280		MVC DKDDPL+@DBFR2-DKDBS2(@DBFR2,@BR),DKDST2-DKDBS2(,@BR)	SET
		2281	*		* FUNCTION AND LIST FOR REWRITE
02BC 5C 01 19 3A		2282		MVC DKDDPL+@DCTRL-DKDBS2(2,@BR),DKDWRF-DKDBS2(,@BR)	SET WRITE
02C0 5F 00 65 3A		2283		SLC DKD170+@D1(1,@BR),DKDWRF(,@BR)	POINT TO WRITE COUNTER
		2284	*		
		2285	***	RETRY WRITE OPERATION	
		2286	*		
02C4 9F 00 FC 38		2287	DKD380	SLC DKDRRT(1,@XR),DKDRTR-DKDBS2(,@BR)	DECR WRITE RETRY COUNTER
02C8 F2 01 32		2288		JNZ DKD410	RETRY IF MORE ALLOWED
02CB C0 87 01C5		2289	DKD390	B DKD280	OTHERWISE DO ACTION 1
		2290	*		
		2291	***	RETRY READ OPERATION	
		2292	*		
02CF 9F 00 FD 38		2293	DKD400	SLC DKDDRT(1,@XR),DKDRTR-DKDBS2(,@BR)	DECR READ RETRY CNTR
02D3 F2 01 27		2294		JNZ DKD410	RETRY IF MORE ALLOWED
02D6 8C 00 FD 02F6		2295		MVC DKDDRT(1,@XR),DKDDCT	RESET READ RETRY COUNTER
02DB E0 87 9A		2296		B DKDAC6(,@XR)	DO ACTION 6
		2297	*		

## ERP SECTION - CONSTANTS AND EQUATES

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00 25/09/15 PAGE 56
		2299			*****	
		2300	*		STORAGE AND CONSTANT AREAS	*
		2301			*****	
		2302	*			
		2303	***		ACTION 4 STORAGE AREAS	
		2304	*			
		0003	2305	DKDA03 EQU	@DADDR+1	LENGTH OF C, S, AND N BYTES
		02DE	2306	DKDCF2 EQU	*	START OF SAVE DCF
02DE		02E1	2307	DS	CL4	SAVED DCF
02E2	01	02E2	2308	DKDCF3 DC	AL1(@ALTFL)	ALTERNATE TRACK FLAG BYTE
02E3		02E4	2309	DS	CL2	C AND S DCF BYTES
02E5		02E6	2310	DKDSAD DS	CL2	SAVE AREA FOR C AND S BYTES
02E7		02E8	2311	DKDDSV DS	CL(@DADDR)	DISK ADDRESS SAVE AREA
02E9		02EA	2312	DKDFDR DS	CL(@CADDR)	SAVE AREA FOR DFDR DATA ADDRESS
02EB	0080	02EC	2313	DKDDSW DC	XL2'0080'	INCREMENT VALUE TO NEXT TRACK
		2314	*			
		2315	***		ERROR RETRY COUNTERS	
		2316	*			
02ED		0005	2317	DKDOCT EQU	5	NO. ERROR RETRY COUNTERS
		02F1	2318	DKDTCT DS	CL(DKDOCT)	ERROR RETRY COUNTERS
		2319	*			
		02F1	2320	DKDDRT EQU	DKDTCT	READ RETRY COUNT
		02F0	2321	DKDRRT EQU	DKDDRT-1	WRITE RETRY COUNT
		02EF	2322	DKDKRT EQU	DKDRRT-1	SEEK RETRY COUNT
		02EE	2323	DKDCRT EQU	DKDKRT-1	TRACK CONDITION CHECK COUNT
		02ED	2324	DKDERT EQU	DKDCRT-1	EQUIPMENT CHECK RETRY COUNT
		2325	*			
		2326	***		ERROR RETRY INITIAL VALUES	
		2327	*			
02F2	02	02F2	2328	DC	AL1(DKDCNE)	INITIAL EQ CHECK RETRY COUNT
02F3	02	02F3	2329	DC	AL1(DKDCCE)	INITIAL TRK COND CHK RETRY COUNT
02F4	10	02F4	2330	DC	AL1(DKDKCE)	INITIAL SEEK RETRY COUNT
02F5	08	02F5	2331	DC	AL1(DKDRCE)	INITIAL WRITE RETRY COUNT
02F6	10	02F6	2332	DKDDCT DC	AL1(DKDDCE)	INITIAL READ RETRY COUNT
		02F6	2333	DKDNIT EQU	*-1	INITIAL ERROR COUNT FIELD
		2334	*			

## DKDISK - ACTION 8 - ERP SECTION

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	25/09/15	PAGE	57
			2336		*****				
			2337	*	ACTION 8 - INTERVENTION REQUIRED				*
			2338		*****				
		00D6	2339		USING DKDBS2,@BR				CALL SECTION BASE VALUE
		02F7	2340	DKDAC8	EQU *				ENTRY TO DO ACTION 8
02F7	7C 00 28		2341		MVI DKDRMG-DKDBS2(,@BR),@ZERO				RESET CYL POSITION TO ZERO
02FA	F2 87 36		2342		J DKD440				RETURN TO CALL SECTION
			2343	*					
			2344	***	RETRY ORIGINAL OPERATION				
			2345	*					
02FD	C2 02 03C0		2346	DKD410	LA \$NUCBS,@XR				POINT XR TO COMMUNICATION AREA
0301	B8 0C 15		2347		TBN \$INDR2-\$NUCBS(,@XR),\$ERPND+\$DKERR				ERROR ENTRY SET UP ?
0304	F2 10 19		2348		JT DKD415				IF YES, DON'T REDO
0307	BA 0C 15		2349		SBN \$INDR2-\$NUCBS(,@XR),\$ERPND+\$DKERR				SET ERROR INDR
			2350	*					
030A	9C 01 6F 7C		2351		MVC \$HISTE-\$NUCBS+@HSTQR(2,@XR),DKD180+@D1(,@BR)				OPER SIO Q+R
030E	78 01 2A		2352		TBN DKDRST(,@BR),@DERSC				WAS ERROR A SEEK CHECK ?
0311	F2 90 04		2353		JF DKD412				BRANCH IF NO
			2354	*					
0314	9C 01 6F AA		2355		MVC \$HISTE-\$NUCBS+@HSTQR(2,@XR),DKD210+@D1(,@BR)				SEEK SIO Q+R
0318	9C 04 74 2E		2356	DKD412	MVC \$HISTE-\$NUCBS+#HISCT(DKDTLN+1,@XR),DKDOFT(,@BR)				SENSE TYPE
031C	9C 02 77 25		2357		MVC \$HISTE-\$NUCBS+#HISTN(3,@XR),DKDZZZ(,@BR)				MOVE ADDR TO LOG
			2358	*					
0320	B8 20 12		2359	DKD415	TBN \$IOIND-\$NUCBS(,@XR),\$HRDER				IS THIS A HARD ERROR ?
0323	D0 10 FF		2360		BT DKD290(,@BR)				YES, GO LOG ERROR AND HALT
0326	9E 00 74 38		2361		ALC \$HISTE-\$NUCBS+#HISCT(1,@XR),DKDONE(,@BR)				BUMP OBR ENTRIES
032A	D0 87 81		2362	DKD420	B DKDSEE(,@BR)				SEEK TO DESIRED TRACK
032D	D0 87 56		2363	DKD430	B DKD160(,@BR)				START I/O OPERATION
0330	D0 87 44		2364		B DKD130(,@BR)				WAIT AND CHECK FOR ERRORS
0333	C0 87 004F		2365	DKD440	B DKD025				RETURN TO CALL SECTION
			2366	*					

DKDISK - DISK UNSAFE ROUTINE

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 25/09/15 PAGE 58
		2368		*****	
		2369	*	THIS ROUTINE IS EXECUTED IF THE DISK IS FOUND UNSAFE.	*
		2370	*	IT WILL HALT INDICATING TO THE OPERATOR THAT AN UNSAFE	*
		2371	*	CONDITION IS PRESENT. WHEN THE OPERATOR PRESSES -START-,	*
		2372	*	BYTE 3 OF THE SENSE DATA WILL BE DISPLAYED.	*
		2373		*****	
		00D6 2374		USING DKDBS2,@BR	WORK AREA BASE VALUE
0337 75 08 2D		0337 2375	DKDNSF EQU *		ENTRY TO UNSAFE ROUTINE
		2376	L	DKDRST+@SNSB3(,@BR),@ARR	DISPLAY SENSE BYTES IN ARR
		2377	*KD450 \$HPL	CODE-@HUNSF	HARD DISK UNSAFE ERROR
		033A 2378+DKD450	EQU *		
033A F0		033A 2379+	DC	XL1 'F0'	INLINE HPL INSTRUCTION
033B 1850		033C 2380+	DC	AL2(@HUNSF)	HALT CODE
033D C0 87 033A		2381	B	DKD450	SORRY, HARD HALT
		2382	*		
		0341 2383	DKDND EQU *		END OF DISK IOCS
		2384	*		

## NERLOG - MODULE PROLOG

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00	25/09/15	PAGE 59
2386				*****			
2387				*			
2388	*	5703-XM1		COPYRIGHT IBM CORP. 1970			
2389	*			REFER TO INSTRUCTIONS ON COPYRIGHT NOTICE 120-2083			
2390	*			*			
2391				*****			
2392				*STATUS			
2393	*			VERSION 1 MODIFICATION 0			
2394	*			*			
2395				*FUNCTION			
2396	*			* NERLOG IS DIVIDED INTO TWO SECTIONS: A CORE RESIDENT INTERFACE			
2397	*			WHICH SAVES AND RESTORES AN AREA OF CORE AND LOADS AND EXECUTES			
2398	*			THE OVERLAY; AN OVERLAY SECTION WHICH UPDATES THE ERROR LOGS			
2399	*			AND INDIVIDUAL VOLUME STATISTICS.			
2400	*			* NERLOG UPDATES THE OBR AND SDR ERROR LOGS.			
2401	*			* IF AN UNRECOVERABLE I/O ERROR OCCURS, A HARD HALT IS GENERATED			
2402	*			WITH A CORRESPONDING DEVICE CODE.			
2403	*			* NERLOG SAVES DKDISK STATUS SO THAT IT MAY RE-ENTER THE IOCS FOR			
2404	*			IT'S FUNCTIONS.			
2405	*			*			
2406				*ENTRY POINTS			
2407	*			NERLOG			
2408	*			A SINGLE ENTRY POINT (REFERENCED \$ERLOG) IS PROVIDED FOR LOGGING			
2409	*			ALL I/O ERRORS. NERLOG IS ALWAYS CALLED BY DKDISK. OTHER IOCS			
2410	*			REQUIRING ERROR LOGGING MUST CALL DKDISK WITH THE INDICATOR			
2411	*			\$ERPND SET TO INDICATE AN ERROR IS PENDING.			
2412	*			*			
2413				*INPUT			
2414	*			INPUT TAKE THE FORM OF AN OBR ENTRY PLACED AT \$HISTE AND THE			
2415	*			NUCLEUS INDICATORS \$ERPND, INDICATING AN ENTRY IS PENDING, AND			
2416	*			\$HRDR IF THE ERROR WAS UNRECOVERABLE.			
2417	*			*			
2418				*OUTPUT			
2419	*			TEMPORARY ERRORS ARE LOGGED, SYSTEM STATUS RESTORED, AND NORMAL			
2420	*			EXECUTION CONTINUED. HARD ERRORS ARE LOGGED AND A HARD HALT,			
2421	*			WITH APPROPRIATE HALT CODE, EXECUTED.			
2422	*			*			
2423				*EXITS, NORMAL			
2424	*			NORMAL EXIT IS TO THE PROGRAM CALLING DKDISK UNLESS A HARD I/O			
2425	*			ERROR OCCURRED. IN THIS CASE, A HARD HALT IS EXECUTED.			
2426				*EXITS, ERROR			
2427	*			*			
2428	*			IF AN UNRECOVERABLE DISK ERROR OCCURS WHILE LOGGING, A SPECIAL			
2429	*			HARD HALT IS EXECUTED. RE-IPL IS THE ONLY RECOVERY.			
2430	*			*			
2431				*TABLES/WORK AREAS			
2432	*			A 14 BYTES TABLE IS USED FOR DETERMINING THE HALT CODE FOR HARD			
2433	*			I/O ERRORS. EACH ENTRY IN THE TABLE CONSISTS OF A ONY BYTE DEVICE*			
2434	*			CODE FOLLOWED BY A ONE BYTE HALT CODE. THE DEVICE CODE IS SCANNED*			
2435	*			AND A MATCH RESULTS IN THE HALT CODE BEING MOVED TO THE CONTROL			
2436	*			BYTE OF A HPL INSTRUCTION.			
2437	*			*			
2438				*ATTRIBUTES			
2439	*			RELOCATABLE			
2440	*			CORE RESIDENT WITH OVERLAY			
2441	*			*			



ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00	25/09/15	PAGE 60
		2442	*	CHARACTER CODE DEPENDENCY			*
		2443	*	N/A			*
		2444	*				*
		2445	*	NOTES			*
		2446	*	ERROR PROCEDURES			*
		2447	*	THE HARD ERROR EXIT IN DKDISK IS MODIFIED TO CAUSE A BRANCH			*
		2448	*	TO A SPECIAL HARD HALT IN NERLOG. AN UNRECOVERABLE DISK ERROR			*
		2449	*	WHILE LOGGING RESULTS IN THIS HALT.			*
		2450	*				*
		2451	*	REGISTER USAGE			*
		2452	*	INDEX REG 1 (@BR) IS USED FOR BASE ADDRESSING. INDEX REG 2 (XR)			*
		2453	*	IS USED FOR INDEXING THE ERROR LOGS.			*
		2454	*				*
		2455	*	SAVED/RESTORED AREAS			*
		2456	*	FIVE SECTORS OF CORE BEGINNING AT THE CORE LOCATION X'0700' ARE			*
		2457	*	SAVED IN THE UNUSED SDR/OBR AREA ON THE REMOVABLE DISK ON			*
		2458	*	DRIVE 1. THE OVERLAY PORTION IS BOUGHT INTO THIS AREA.			*
		2459	*	UPON COMPLETION OF ERROR LOGGING, THESE SECTORS ARE RESTORED.			*
		2460	*				*
		2461	*	MODIFICATION CONSIDERATIONS			*
		2462	*	NERLOG SAVES THE EXIT VALUES OF DKDISK SO THAT IT MAY RE-ENTER			*
		2463	*	THE IOCS. CARE SHOULD BE TAKEN WHEN MODIFING EITHER DKDISK			*
		2464	*	OR NERLOG SO THAT THE INTERFACING IS NOT DISTURBED.			*
		2465	*				*
		2466	*	REQUIRED MODULES			*
		2467	*	@SYSEQ - GENERAL SYSTEM EQUATES.			*
		2468	*	@HOWEQ - HARDWARE VALUE EQUATES.			*
		2469	*	@FXDEQ - NUCLEUS LOCATION EQUATES.			*
		2470	*	@HLTEQ - HALT INDICATOR EQUATES.			*
		2471	*	@CY0EQ - CYLINDER ZERO EQUATES.			*
		2472	*	@CANEQ - TRANSCIENT LOCATION EQUATES.			*
		2473	*	@CVOLQ - VOLUME LABEL EQUATES			*
		2474	*	DKDISK - DISK IOCS			*
		2475	*				*
		2476	*	OTHER			*
		2477	*	NONE			*
		2478	*				*
		2479	*	*****			*

## NERLOG - I/O ERROR LOGGING INTERFACE

ERR LOC OBJECT CODE ADDR STMT SOURCE STATEMENT VER 15, MOD 00 25/09/15 PAGE 61

```

2482 *****
2483 *   NERLOG
2484 *
2485 *   THIS INTERFACE SAVES ALL PERTINENT DATA REQUIRED FOR DKDISK
2486 *   RE-ENTRY. IT CALLS DKDISK TO SAVE 5 SECTORS OF CORE ON CYL 0.
2487 *   NBLOAD IS THEN CALLED TO EXECUTE THE ERROR LOGGING OVERLAY.
2488 *   UPON COMPLETION OF ERROR LOGGING, CORE AND DKDISK ARE RESTORED
2489 *   TO ORIGINAL VALUES.
2490 *   NERLOG
2491 *****

0345 2493 ORG $ERLOG EXECUTION ADDRESS
000C 2494 NERCDE EQU 3*4 LENGTH OF 3 4-BYTE INSTRUCTIONS
00D6 2495 USING DKDBS2,@BR DKDISK BASE VALUE
0345 2496 NERLOG EQU * ENTRY TO LOG AN ERROR
0345 1C 0B 039D 17 2497 MVC NERTRN(NERCDE),DKD120+@OP1(,@BR) SAVE DISK RETURN + REGS
034A 7C 80 09 2498 MVI DKD090+@Q(,@BR),@NOP NOP BRANCH TO NERLOG
034D 0C 01 01D8 03A6 2499 MVC DKD290+@OP1(@CADDR),NERHRD SET HARD ERROR BRANCH
0353 F3 10 00 2500 SIO NERDSL,@KEYBD DISABLE KB INTERRUPTS
0356 F1 E2 00 2501 APL @PBUSY WAIT FOR PRINTER
0359 38 40 03D2 2502 TBN $IOIND,$DTRDR IS DATA RECORDER ON SYSTEM ?
035D F2 90 03 2503 JF NER050 SKIP WAIT IF NOT
0360 F1 F2 00 2504 APL @BZ37B WAIT ON DATA RECORDER
0363 C0 87 0025 2505 NER050 B $DISKN WRITE CORE TO SAVE AREA
0367 03A9 0368 2506 DC AL2(NERWRT) DPL ADDRESS
0369 C0 87 0025 2507 B $DISKN LOAD + EXEC OVERLAY AT X'0700'
036D 03AF 036E 2508 DC AL2(NEROVL) DPL ADDRESS
036F C0 87 0025 2509 B $DISKN WAIT FOR OP COMPLETION
0373 057F 0374 2510 DC AL2($WAITF) WAIT DPL ADDRESS
0375 C0 87 0700 2511 B NEROVR EXECUTE ERROR LOGGING
2512 *
2513 *** RETURN IS MADE HERE FROM OVERLAY AFTER STARTING RESTORE CORE OP
2514 *
0379 2515 NER100 EQU * RETURN POINT FROM OVERLAY
037D 07D9 037E 2516 B $DISKN START CORE RESTORE
037F C0 87 0025 2517 DC AL2(NERSTR) DPL ADDRESS
0383 057F 0384 2518 B $DISKN WAIT FOR OP COMPLETION
0385 3C 10 00DF 2519 DC AL2($WAITF) WAIT DPL ADDRESS
0389 0C 01 01D8 03A8 2520 MVI DKD090+@Q,@BT RESTORE NERLOG BRANCH
038F F3 10 02 2521 MVC DKD290+@OP1(@CADDR),NERLGA RESTORE HARD ERROR BRANCH
0392 039D 2522 SIO NERENL,@KEYBD ENABLE KB INTERRUPTS
2523 NERTRN DS CL(NERCDE) EXECUTE OLD DKDISK REGISTER
2524 * * RESTORE AND RETURN BRANCH

039E 2526 NERHLT EQU * ENTRY TO HARD ERROR ON ERROR
2527 * * LOGGING
2528 * $HPL CODE-@HLOGE HARD DISK ERROR WHILE LOGGING
039E F0 039E 2529+ DC XL1'F0' INLINE HPL INSTRUCTION
039F 1844 03A0 2530+ DC AL2(@HLOGE) HALT CODE
03A1 C0 87 039E 2531 B NERHLT SORRY, HARD ERROR

03A5 039E 03A6 2533 NERHRD DC AL2(NERHLT) ADDRESS OF HARD ERROR ROUTINE
03A7 0345 03A8 2534 NERLGA DC AL2($ERLOG) ADDRESS OF ENTRY TO NERLOG
2535 *
2536 *ERWRT $DPL FUNC-@DPUT,DADDR-#CORSV,CNT-#@CORS,CADDR-$$KLD2
03A9 2537+NERWRT EQU * DISK PARAMETER LIST

```

NERLOG - I/O ERROR LOGGING INTERFACE

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	25/09/15	PAGE 62
03A9	02	03A9	2538+	DC	AL1(@DPUT)			REQUESTED FUNCTION
03AA	0010	03AB	2539+	DC	AL2(#CORSV)			DISK ADDRESS
03AC	05	03AC	2540+	DC	AL1(#@CORS)			SECTOR COUNT
03AD	0700	03AE	2541+	DC	AL2(\$\$KLD2)			BUFFER ADDRESS
			2542+	***	END OF EXPANSION	***		
			2543	*				
			2544	*EROVL \$DPL	FUNC-@DGET,DADDR-#NEROV,CNT-#@NERO,CADDR-\$\$KLD2			
		03AF	2545+	NEROVL EQU	*			DISK PARAMETER LIST
03AF	01	03AF	2546+	DC	AL1(@DGET)			REQUESTED FUNCTION
03B0	009C	03B1	2547+	DC	AL2(#NEROV)			DISK ADDRESS
03B2	03	03B2	2548+	DC	AL1(#@NERO)			SECTOR COUNT
03B3	0700	03B4	2549+	DC	AL2(\$\$KLD2)			BUFFER ADDRESS
			2550+	***	END OF EXPANSION	***		
		0000	2552	NERDSL EQU	X'00'			DISABLE INTERRUPTS CNTL
		0002	2553	NERENL EQU	X'02'			ENABLE INTERRUPTS CNTL
			2554	*				

NUCLEUS - PERMANENT STORAGE AND CONSTANT AREAS

ERR	LOC	OBJECT CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00 25/09/15 PAGE 63
				2556	*****	*****	
				2557	*	PERMANENT SYSTEM STORAGE AREAS	*
				2558	*****	*****	
	03C0			2559	ORG	\$NUCBS	SET LOCATION COUNTER
			03C0	2560	NNUCBS	EQU *	START OF COMMUNICATION AREA
	03C0		03C0	2561	NRMRGN	DS CL1	SOFTWARE RIGHT MARGIN VALUE
	03C0			2562	ORG	*-1	SET INITIAL VALUE FOR RIGHT MG
	03C0	DC	03C0	2563	DC	IL1'220'	*
	03C1		03C1	2564	NLMRGN	DS CL1	SOFTWARE LEFT MARGIN VALUE
	03C1			2565	ORG	*-1	SET INITIAL VALUE FOR LEFT MG
	03C1	00	03C1	2566	DC	IL1'00'	*
				2567	*		
	03C2		03C2	2568	NPRPOS	DS CL1	PRINTER POSITION WORK AREA
	03C2			2569	ORG	*-1	SET INITIAL PRINTER POSITION
	03C2	00	03C2	2570	DC	IL1'00'	*
				2571	*		
	03C3		03C3	2572	NKEYCD	DS CL1	INPUT SOURCE INDICATOR
	03C3			2573	ORG	*-1	INITIALIZE IT FOR
	03C3	00	03C3	2574	DC	IL1'00'	* KEYBOARD INPUT
				2575	*		
				2576	***	REGISTER SAVE AREAS FOR CONSOLE INTERRUPT ROUTINE	
				2577	*		
	03C4		03C5	2578	NBRSAV	DS CL(@REGL)	BASE REG SAVE AREA
	03C6		03C7	2579	NXRSAV	DS CL(@REGL)	OTHER INDEX REG SAVE AREA
				2580	*		
	03C8		03CB	2581	NTABLN	DS CL4	AUTOMATIC LINE NUMBER TO BE
	03C8			2582	ORG	*-4	* INSERTED IF TAB KEY IS FIRST
	03C8	F0F1F0F0	03CB	2583	DC	CL4'0100'	* KEY PRESSED ON KEYBOARD INPUT
	03CC	40	03CC	2584	DC	CL1' '	BLANK FOLLOWING AUTO LINE NO.
	03CD		03CD	2585	NCAERR	DS CL1	ERROR CODE SAVE FOR INTERFACE
	03CD			2586	ORG	*-1	* TO CMD ANALYZER ROUTINE
	03CD	00	03CD	2587	DC	XL1'00'	INITIALIZE TO NO ERROR
	03CE		03CF	2588	NINLNO	DS CL2	EX THE LINE NO. FOR INTERPRETER
	03CE			2589	ORG	*-2	INITIALIZE INDICATOR
	03CE	8000	03CF	2590	DC	XL2'8000'	SET TO NORMAL CONDITION
	03D0		03D0	2591	NXIND1	DS CL1	PRIMARY EXECUTION MODE INDRS
	03D1		03D1	2592	NXIND2	DS CL1	SECODARY EXECUTION MODE INDRS
	03D2		03D2	2593	NIOIND	DS CL1	I/O STATUS INDICATOR
	03D3		03D3	2594	NCRTIN	DS CL1	INDICATORS
	03D4		03D4	2595	NINDR1	DS CL1	INDICATORS
	03D0			2596	ORG	\$XIND1	INITIALIZE INDRS
	03D0	0000000000	03D4	2597	DC	XL5'00'	* TO ZERO
	03D5		03D5	2598	NINDR2	DS CL1	INDICATORS
	03D5			2599	ORG	\$INDR2	INITIALIZE INDRS
	03D5	00	03D5	2600	DC	XL1'00'	* TO ZERO
	03D6		03D6	2601	NINDR3	DS CL1	INDICATOR
	03D6			2602	ORG	\$INDR3	INITIALIZE INDR
	03D6	00	03D6	2603	DC	XL1'00'	* TO ZERO
	03D7		03D7	2604	NDKSIZ	DS CL1	DISK SIZE INDR
	03D7			2605	ORG	\$DKSIZ	INITIALIZE INDR
	03D7	02	03D7	2606	DC	AL1(\$DK200)	* TO MINIMUM DISK SIZE
	03D8		03D8	2607	NXIND3	DS CL1	PAST PRIMARY EXEC MODE INDR
	03D8			2608	ORG	\$XIND3	INITIALIZE INDR
	03D8	00	03D8	2609	DC	XL1'00'	* TO ZERO
	03D9		03DA	2610	NFILIB	DS CL(@DADDR)	DADDR OF CURRENT FILE LIBRARY
	03D9			2611	ORG	\$FILIB-1	PLACE INITIAL VALUE

NUCLEUS - PERMANENT STORAGE AND CONSTANT AREAS

ERR	LOC	OBJECT CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00 25/09/15 PAGE 64
	03D9	0000	03DA	2612	DC	XL2'0000'	INITIALLY ZERO
	03DB		03DC	2613	NUSRDR DS	CL(@DADDR)	REL DISP TO 1ST USER BLOCK
	03DD		03DD	2614	NCOFNG DS	CL1	CONFIGURATION INDRS
	03DD			2615	ORG	\$CONFIG	PLACE INITIAL VALUE
	03DD	00	03DD	2616	DC	XL1'00'	SET ALL DEVICES OFF
	03DE	0005	03DF	2617	NLEVEL DC	AL2(@SYLVL)	CURRENT SYSTEM LEVEL NUMBER
	03E0		03E0	2618	NDBGUF DS	CL1	GUFUDI DEBUG INDR
	03E0			2619	ORG	\$DBGUF	PLACE INITIAL VALUE
	03E0	00	03E0	2620	DC	XL1'00'	SET GUFUDI TO CRUSH + REORDER
	03E1		03E1	2621	NKEYBD DS	CL1	KEYBOARD TYPE
	03E1			2622	ORG	\$KEYBD	PLACE INITIAL VALUE
	03E1	01	03E1	2623	DC	XL1'01'	SET INDR FOR KB1
	03E2		03E2	2624	NCRPOS DS	CL1	CURRENT CURSOR POSITION
	03E2			2625	ORG	\$CRPOS	PLACE INITIAL VALUE
	03E2	00	03E2	2626	DC	XL1'00'	SET TO ZERO
	03E3		03E3	2627	NBUFPT DS	CL1	LINE PRINTER BUFFER POINTER
	03E3			2628	ORG	\$BUFPT	PLACE INITIAL VALUE
	03E3	00	03E3	2629	DC	XL1'00'	SET TO ZERO
	03E4		03E4	2630	NLPRP3 DS	CL1	LINE PRINTER INDICATORS
	03E4			2631	ORG	\$LPRP3	PLACE INITIAL VALUE
	03E4	00	03E4	2632	DC	XL1'00'	SET TO ZERO
	03E5		03E5	2633	NLPROS DS	CL1	LINE PRINTER PRINT POSITION
	03E5			2634	ORG	\$LPROS	PLACE INITIAL VALUE
	03E5	00	03E5	2635	DC	XL1'00'	SET TO ZERO
	03E6		03E6	2636	NNEXTB DS	CL1	RELATIVE DADDR PROCEDURE CALL
	03E6			2637	ORG	\$NEXTB	PLACE INITIAL VALUE
	03E6	00	03E6	2638	DC	XL1'00'	SET TO ZERO
	03E7		03E7	2639	NNEXTL DS	CL1	DISPLACEMENT WITHIN DB
	03E7			2640	ORG	\$NEXTL	PLACE INITIAL VALUE
	03E7	00	03E7	2641	DC	XL1'00'	SET TO ZERO
	03E8		03E8	2642	NDFDET DS	CL1	GRAPRP INTERNAL INDR.
	03E8			2643	ORG	\$DFDET	PLACE INITIAL VALUE
	03E8	00	03E8	2644	DC	XL1'00'	SET ZERO
	03E9		03EA	2645	NLPRI0 DS	CL2	LINE PRT BUFFER INCR & PDAR
	03EA			2646	ORG	\$LPRI0	PLACE INITIAL VALUE
	03EA	0000	03EB	2647	DC	XL2'00'	SET TO ZERO
	03EC		03F6	2648	NPTCH1 DS	CL11	COMMUNICATION PATCH AREA
				2649	*		
				2650	***	SAVE AREAS FOR REMOVABLE DISK VOLID'S	
				2651	*		
			03F7	2652	NVOLID EQU	*	LEFT BYTE OF VOLID TABLE
	03F7		03FE	2653	DS	CL(#VLTBE)	VOLID REMOVABLE DISK 1
	03FF		0406	2654	DS	CL(#VLTBE)	VOLID FIXED DISK 1
	0407		040E	2655	DS	CL(#VLTBE)	VOLID REMOVABLE DISK 2
	040F		0416	2656	DS	CL(#VLTBE)	VOLID FIXED DISK 2
			0417	2657	NPKERT EQU	*	LEFT BYTE OF ERROR RATE TABLE
	0417		041A	2658	DS	CL(#PKRTL)	PACK ERR RATE CNTR FOR R1
	041B		041E	2659	DS	CL(#PKRTL)	PACK ERR RATE CNTR FOR F1
	041F		0422	2660	DS	CL(#PKRTL)	PACK ERR RATE CNTR FOR R2
	0423		0426	2661	DS	CL(#PKRTL)	PACK ERR RATE CNTR FOR F2
	03F6			2662	ORG	\$VOLID	INIT VOLID & ERR RATE TABLES
	03F6	00000000000000	03FB	2663	DC	XL(#VOLNG)'00'	INITIALIZE R1 VOLID
	03FC	0000	03FD	2664	DC	XL(@DADDR)'0000'	INITIALIZE FILE LIBR ADDR
	03FE	00000000000000	0403	2665	DC	XL(#VOLNG)'00'	INITIALIZE F1 VOLID
	0404	0001	0405	2666	DC	XL(@DADDR)'0001'	INITIALIZE FILE LIBR ADDR
	0406	00000000000000	040B	2667	DC	XL(#VOLNG)'00'	INITIALIZE R2 VOLID

NUCLEUS - PERMANENT STORAGE AND CONSTANT AREAS

ERR	LOC	OBJECT CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00 25/09/15 PAGE 65
040C	0002		040D	2668	DC	XL(@DADDR)'0002'	INITIALIZE FILE LIBR ADDR
040E	00000000000000		0413	2669	DC	XL(#VOLNG)'00'	INITIALIZE F2 VOLID
0414	0003		0415	2670	DC	XL(@DADDR)'0003'	INITIALIZE FILE LIBR ADDR
				2671	*		
0416	00000000		0419	2672	DC	XL(#PKRTL)'00'	SET ERR COUNTERS R1 TO ZERO
041A	00000000		041D	2673	DC	XL(#PKRTL)'00'	SET ERR COUNTERS F1 TO ZERO
041E	00000000		0421	2674	DC	XL(#PKRTL)'00'	SET ERR COUNTERS R2 TO ZERO
0422	00000000		0425	2675	DC	XL(#PKRTL)'00'	SET ERR COUNTERS F2 TO ZERO
				2676	*		
0426			042D	2677	NPASWD DS	CL8	CURRENT PASSWORD
0426				2678	ORG	\$PASWD-7	INITIALIZE PASSWORD
0426	4040404040404040		042D	2679	DC	CL8' '	INITIALIZE TO BLANKS
			042E	2680	NHISTE EQU	*	1ST BYTE OF HISTORY ENTRY
042E			0435	2681	NHIST1 DS	CL(#HISLN)	1ST ENTRY OF HISTORY LOG
0436			0437	2682	DS	CL(#DKEXT)	DISK HIST ENTRY EXTENSION
0438			043A	2683	NDATE DS	CL3	CURRENT DATE
043B			043B	2684	NEXFTR DS	CL1	CORE EXPANSION FACTOR
043C			0443	2685	NWFNME DS	CL8	WORK FILE NAME
043B				2686	ORG	\$EXFTR	PLACE INITIAL VALUE
043B	0000000000000000		0443	2687	DC	XL9'00'	INITIALIZE EXPANSION + FILE NAME
0444			0449	2688	NDPLSV DS	CL6	DSL SAVE AREA
044A			044B	2689	NPRDEV DS	CL(@CADDR)	ADDRESS OF SYSTEM PRINTER IOCR
044A				2690	ORG	\$PRDEV-1	PLACE INITIAL VALUE
044A	0707		044B	2691	DC	AL2(\$\$PRNT)	INITIALIZE IT TO DPRINT
044C			044D	2692	NCRTAD DS	CL(@CADDR)	ENTRY ADDR FOR CRT LOADER
044E			0454	2693	NPLST1 DS	CL(@DPLNG+1)	LAST I/O PARM LIST EXECUTED
0455			045B	2694	NPLST2 DS	CL(@DPLNG+1)	2ND TO LAST I/O PARM LIST
045C			0462	2695	NPLST3 DS	CL(@DPLNG+1)	3RD TO LAST I/O PARM LIST
				2696	*		
0463	0001		0464	2697	NC0001 DC	XL2'0001'	CONSTANT OF ONE
				2698	*		

SYSNUC - SYSTEM PRINTER INTERFACE ROUTINE

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	25/09/15	PAGE 66
					2700		*****			
					2701	*	INTERFACE TO THE SYSTEM PRINTER IOCR			*
					2702	*				*
					2703	*	\$PRDEVM WILL HOLD THE CORE ADDRESS OF THE SYSTEM PRINTER IOCR.			*
					2704	*	EITHER DPRINT (MATRIX PRINTER) OR DSPLYN (CRT).			*
					2705		*****			
0465					2706	ORG	\$SPRNT			SET LOCATION COUNTER
				0465	2707	NSPRNT	EQU	*		ENTRY TO THE SYSTEM PRINTER IOCR
0465	35	10	044B		2708	L	\$PRDEV,@IAR			BRANCH TO THE CORRECT IOCR
					2709	*				



SYSNUC - ERROR ROUTINE INTERFACE

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	25/09/15	PAGE	67
		2711			*****				
		2712	*		INTERFACE TO ERROR ROUTINE IN CMD ANALYZER FOR KEYWORD PGM'S	*			
		2713			*****				
0469	C0 87 0522	0469	2714	NCAERK	EQU	*			
			2715		B	\$BLOAD			
046D	046F	046E	2716		DC	AL2(\$ERDPL)			
			2717	*					
		046F	2718	NERDPL	EQU	*			
046F	01	046F	2719		DC	XL1'01'			
0470	18C0	0471	2720	NERMAD	DC	AL2(\$ERRP)			
0472	03	0472	2721		DC	AL1(\$@ERR)			
0473	0C00	0474	2722		DC	XL2'0C00'			
			2723	*					

NQUIRY - CONSOLE INTERRUPT ROUTINE

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00 25/09/15 PAGE 68
				2725		*****	*****	
				2726	*	CONSOLE INTERRUPT PROCESSING	*	
				2727		*****	*****	
0475	F2	80	21	2728	NQU050	JC	NAB200,@NOP	BRANCH IF NOT 'DISABLED'
				0476 2729	NCIMSK	EQU	NQU050+@Q	INDR FOR CI MASKED
0478	3C	80	0496	2730		MVI	\$CISUS,@NOP	SET INDR FOR SUSPENDED CI
047C	3A	20	03E0	2731		SBN	\$DBGUF,\$IRKEY	SET INPUT KEYBOARD GUFUDI INDR
0480	F3	10	1B	2732	NCIEXT	SIO	@KELOK,@KEYBD	UNLOCK, ENABLE KB, EXIT OFF
				2733	*			
				2734	***	HARDWARE CONSOLE INTERRUPT ENTRY POINT		
				2735	*			
				0483 2736	NCIENT	EQU *		ENTRY FROM INTERRUPT
0483	30	10	048C	2737		SNS	NQUSNS,@KEYBD	SENSE DATA FROM KEYBOARD
0487	C0	87	0475	2738		B	NQU050	BRANCH TO PROCESS CI
048B				048C 2739	NQUSNS	DS	CL(@REGL)	KEYBOARD SENSE DATA
				2740	*			
				2741	***	ENTRY TO UNSUSPENDED AND UNMASK CI		
				2742	*			
048D				2743		ORG	\$UNMSK	
				048D 2744	NUNMSK	EQU	*	ENTRY TO UNMASK CI
048D	34	08	0498	2745		ST	NQU100+@OP1,@ARR	SAVE RETURN ADDRESS
0491	3C	87	0476	2746		MVI	\$CIMSK,@UCB	ENABLE CI
0495	C0	87	0000	2747	NQU100	BC	*-*,@UCB	RETRUN FROM INTERRUPT
				0496 2748	NCISUS	EQU	NQU100+@Q	INDR FOR SUSPENDED IR
				2749	*			

NABORT - SUBROUTINE LINKAGE ROUTINES

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	25/09/15	PAGE 69
				2751		*****				
				2752	*	ABORT CURRENT OPERATION				*
				2753		*****				
				0499	2754	NAB200 EQU	*			ENTRY TO ABORT CURRENT OP
0499	C0	87	05A8		2755	B	NAB300			TEST LINE PRINTER CONFIG
049D	3B	07	03C3		2756	NCAIPL SBF	\$KEYCD,\$NOLST+\$CARDI+\$IOYES			SET INPUT TO KEYBOARD AND
					2757	*				* INDICATE I/O ROUTINES NOT IN CORE
04A1	3C	87	0496		2758	NCARPL MVI	\$CISUS,@UCB			RESET INDR FOR NO SUSPENDED CI
04A5	3C	87	0476		2759		MVI \$CIMSK,@UCB			UNMASK IR
04A9	3B	08	03D2		2760		SBF \$IOIND,\$CMDKY			SET COMMAND KEY ONLY OFF
04AD	38	01	03D1		2761		TBN \$XIND2,\$EXCMD			IN EXECUTION ?
04B1	F2	10	0A		2762		JT NPAUS1			IF YES SAVE ALL OF CORE
					2763	*				
04B4	C0	87	0522		2764	NCABLD B	\$BLOAD			GO LOAD AND EXECUTE GUFUDI
04B8	0580			04B9	2765		DC AL2(NBL100)			GUFUDI DPL ADDRESS
					2766	*				

## NPAUSE - ROUTINE TO SAVE/RESTORE CORE

ERR LOC OBJECT CODE ADDR STMT SOURCE STATEMENT VER 15, MOD 00 25/09/15 PAGE 70

```

2768 *****
2769 * ROUTINE TO SAVE/RESTORE CORE AND EXECUTE #EXMSG (PGM INT PROC) *
2770 *****
04BA 2771 ORG $PAUSD SET LOCATION COUNTER
048D 2772 USING $UNMSK,@BR BASE ADDRESSING
04BA 2773 NPAUSE EQU * ENTRY TO SAVE CORE
04BA 34 08 0498 2774 ST NQU100+@OP1,@ARR SAVE RETURN ADDRESS
04BE 2775 NPAUS1 EQU * ENTRY FOR EXECUTION INTERRUPT
04BE 34 01 04FA 2776 NPA010 ST NPA060+@OP1,@BR SAVE BASE REGISTER
04C2 C2 01 048D 2777 LA $UNMSK,@BR LOAD BASE REGISTER
04C6 5C 01 71 0B 2778 MVC NPA090+@OP1(@CADDR,@BR),NQU100+@OP1(@BR) SAVE RETURN ADDR
04CA 74 02 65 2779 NPA015 ST NPA050+@OP1(@BR),@XR SAVE INDEX REGISTER
04CD 7C 87 60 2780 MVI NPA040+@Q(@BR),@UCB SET UP BRANCH FOR BLOAD
04D0 7C 02 81 2781 MVI $CSDPL+@DCTRL(@BR),@DPUT SET WRITE FUNCTION
04D3 F2 87 0A 2782 J NPA020 SKIP

04D6 C2 01 048D 2784 NRSTR LA $UNMSK,@BR LOAD BASE REGISTER
04DA 7C 01 81 2785 MVI $CSDPL+@DCTRL(@BR),@DGET SET READ FUNCTION
04DD 7C 80 60 2786 MVI NPA040+@Q(@BR),@NOP SET UP NO-OP BR FOR BLOAD
04E0 C0 87 0025 2787 NPA020 B $DISKN START DISK I/O
04E4 050E 04E5 2788 DC AL2($CSDPL) ADDR OF DPL
04E6 C0 87 0025 2789 NPA030 B $DISKN WAIT FOR DISK I/O COMPLETION
04EA 057F 04EB 2790 DC AL2($WAITF) WAIT DPL ADDRESS
04EC F2 00 17 2791 NPA040 JC NPA100,*-* BRANCH TO BLOAD FOR SAVE OP
2792 * * ELSE RETURN AFTER RESTORE
04EF C2 02 0000 2793 NPA050 LA *-* ,@XR RESTORE INDEX REGISTER
04F3 5D 01 71 80 2794 CLC NPA090+@OP1(@CADDR,@BR),NPAIDF(@BR) RETURN FROM FE AID ?
04F7 C2 01 0000 2795 NPA060 LA *-* ,@BR RESTORE BASE REGISTER
04FB C0 01 0000 2796 NPA090 BNE *-* RETURN IF NOT A CALL FROM UTMON
2797 * WAIT FOR FE TO DO SOMETHING
04FF F0 04FF 2798 * $HPL CODE-@HFEHT OPTION 'H' FE HALT
0500 0804 0501 2799+ DC XL1'F0' INLINE HPL INSTRUCTION
0502 C0 87 04FB 2800+ DC AL2(@HFEHT) HALT CODE
2801 B NPA090 GO BACK FOR ANOTHER TRY
2802 *
0506 C0 87 051E 2803 NPA100 B $RLOAD BRANCH TO LOAD PRGRAM
050A 0514 050B 2804 DC AL2(NEXDPL) ADDR EX TIME MESSAGE PROGRAM
050C 0004 050D 2805 NPAIDF DC AL2($FEARR) ARR ADDR FROM FE LOAD
2806 *
050E 050E 2807 NCSDDL EQU * SAVE CORE DPL
050F 050E 2808 DS CL1 CNTL BYTE
050F 0511 2809 NSWPCR DS CL(@DADDR+1) SCTR ADDRESS AND COUNT SAVE AREA
050F 2810 ORG $SWPCR-2
050F 0800 0510 2811 DC AL2(##$COR) CURRENT SAVE ADDR
0511 1A 0511 2812 DC IL1'26' SCTR COUNT FOR 8K MACHINE
0512 0600 0513 2813 DC AL2($ENDNU) START OF CORE TO BE SAVED
2814 *
0514 2815 NEXDPL EQU * LOAD 'SUPER' DPL
0514 01 0514 2816 DC XL1'01' CNTL BYTE
0515 07D4 0516 2817 NEXADR DC AL2(##$EXMS) DISK ADDR OF EXMSG
0517 03 0517 2818 DC AL1(##$EXM) SECTOR COUNT
0518 0C00 0519 2819 DC AL2($$KLD3) LOAD ADDRESS
2820 *

```

## MOPPET - MODULE PROLOG

ERR	LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00	25/09/15	PAGE 71
			2822		*****			
			2823	*				*
			2824	*	5703-XM1 COPYRIGHT IBM CORP. 1970			*
			2825	*	REFER TO INSTRUCTIONS ON COPYRIGHT NOTICE 120-2083			*
			2826	*				*
			2827		*****			*
			2828	*	STATUS			*
			2829	*	VERSION 1 MODIFICATION 0			*
			2830	*				*
			2831	*	FUNCTION			*
			2832	*	* NERLOG IS USED FOR LOADING OVERLAYS. ONE OF THREE TYPES OF			*
			2833	*	LOADS ARE PERFORMED:			*
			2834	*	- LOAD AND EXECUTE USING ABSOLUTE DISK ADDRESS.			*
			2835	*	- LOAD AND EXECUTE USING RELATIVE DISK ADDRESS.			*
			2836	*	- LOAD AND RETURN USING RELATIVE DISK ADDRESS.			*
			2837	*	* IF OVERLAY TRACING IS REQUESTED, NBLOAD WILL LOAD AND EXECUTE			*
			2838	*	ZTRACE.			*
			2839	*				*
			2840	*	ENTRY POINTS			*
			2841	*	THREE ENTRY POINTS ARE PROVIDED FOR THE THREE TYPES OF LOADS.			*
			2842	*	\$LOADR - LOAD RELATIVE AND RETURN.			*
			2843	*	\$RLOAD - LOAD RELATIVE AND EXECUTE.			*
			2844	*	\$BLOAD - LOAD ABSOLUTE AND EXECUTE.			*
			2845	*	THE CALLING SEQUENCE IS THE SAME FOR ALL THREE ENTRY POINTS.			*
			2846	*	B \$BLOAD (OR OTHER ENTRY)			*
			2847	*	DC AL2(DPLA)			*
			2848	*	WHERE DPLA IS THE CORE ADDRESS OF THE DPL USED FOR LOADING.			*
			2849	*				*
			2850	*	INPUT			*
			2851	*	INPUT TO NBLOAD TAKES THE FORM OF THE STANDARD DISK PARAMETER			*
			2852	*	LIST (DPL).			*
			2853	*				*
			2854	*	OUTPUT			*
			2855	*	N/A			*
			2856	*				*
			2857	*	EXTERNAL REFERENCES			*
			2858	*	\$C0001 - CONSTANT OF ONE.			*
			2859	*	\$XRSV - REGISTER 2 SAVE AREA.			*
			2860	*	\$DISKN - ENTRY TO DISK IOCS.			*
			2861	*	\$CIEXT - INTERRUPT LEVEL 1 EXIT INSTRUCTION.			*
			2862	*				*
			2863	*	EXITS, NORMAL			*
			2864	*	DEPENDING ON THE FUNCTION, EXIT IS MADE EITHER TO THE CALLING			*
			2865	*	PROGRAM (ENTRY AT \$LOADR) OR THE THE FIRST BYTE PAST THE PROGRAM			*
			2866	*	HEADER IN THE LOADED OVERLAY.			*
			2867	*				*
			2868	*	EXITS, ERROR			*
			2869	*	N/A			*
			2870	*				*
			2871	*	TABLES/WORK AREAS			*
			2872	*	N/A			*
			2873	*				*
			2874	*	ATTRIBUTES			*
			2875	*	RELOCATABLE			*
			2876	*	CORE RESIDENT IN SYSTEM NUCLEUS			*
			2877	*				*

MOPPET - MODULE PROLOG

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00	25/09/15	PAGE 72	
			2878	*CHARACTER CODE DEPENDENCY				*
			2879	* N/A				*
			2880	*				*
			2881	*NOTES				*
			2882	* ERROR PROCEDURES				*
			2883	* N/A				*
			2884	*				*
			2885	* REGISTER USAGE				*
			2886	* INDEX REG 2 (@XR) IS SAVED AT \$XRSV AND THE USED FOR BASE				*
			2887	* ADDRESSING. INDEX REG 1 (@BR) IS DESTROYED IN ITS USE AS AN				*
			2888	* INDEX POINTER.				*
			2889	*				*
			2890	* SAVED/RESTORED AREAS				*
			2891	* N/A				*
			2892	*				*
			2893	* MODIFICATION CONSIDERATIONS				*
			2894	* N/A				*
			2895	*				*
			2896	* REQUIRED MODULES				*
			2897	* @SYSEQ - GENERAL SYSTEM EQUATES.				*
			2898	* @FXDEQ - NUCLEUS LOCATION EQUATES.				*
			2899	* @SPFEQ - SYSTEM PROGRAM DISK ADDRESSES.				*
			2900	*				*
			2901	* OTHER				*
			2902	* NONE				*
			2903	*				*
			2904	*****				*

## NBLOAD - BLAST LOADER ROUTINES

ERR LOC OBJECT CODE ADDR STMT SOURCE STATEMENT VER 15, MOD 00 25/09/15 PAGE 73

```

2907 *****
2908 * ROUTINE TO BLAST LOAD REQUESTED PROGRAM AND EXECUTE IT. *
2909 * THE EXECUTION ADDRESS IS THE LOAD ADDRESS. *
2910 * THE REQUESTING PROGRAM MAY OVERLAY ITSELF. *
2911 *****
051A 2912 ORG $LOADR
0522 2913 USING NBLOAD,@XR BASE VALUE FOR NBLOAD
051A 2914 NLOADR EQU * ENTRY FOR RELOCATABLE PROGRAM
2915 * * LOAD (NO EXECUTION)
051A 3C 80 0569 2916 MVI NBL065+@Q,@NOP SET RETURN INDR
051E 2917 NRLOAD EQU * ENTRY FOR RELOCATBLE BLAST
051E 3C 80 0541 2918 MVI NBL050+@Q,@NOP SET UP FOR RELOCATE LOAD
0522 2919 NBLOAD EQU * ENTRY TO BLAST LOADER
0522 36 08 0464 2920 A $C0001,@ARR CALC ADDR OF PARAMETER
0526 34 02 03C7 2921 ST $XRSV,@XR SAVE XR FOR KEYWORD PROGRAMS
052A C2 02 0522 2922 LA NBLOAD,@XR LOAD BASE REGISTER
052E B4 08 19 2923 ST NBL020+@OP1(,@XR),@ARR CALC ADDR OF PARAMETER
0531 36 08 0464 2924 A $C0001,@ARR CALCULATE POSSIBLE RETURN ADDR
0535 B4 08 4F 2925 ST NBL066+@OP1(,@XR),@ARR SAVE RETURN ADDRESS
0538 35 01 0000 2926 NBL020 L *-*,@BR XR POINTS TO DPL FOR LOAD FUNC.
053C 9C 05 5C 05 2927 MVC NBL080(@DPLNG,@XR),@DBFR2(,@BR) MOVE IN DPL
0540 F2 87 07 2928 NBL050 JC NBL060,@UCB JUMP IF ABSOLUTE LOAD
0543 AE 01 59 65 2929 ALC NBL070+@PSAD(@DADDR,@XR),$BSADR(,@XR) ADD RELOCATE DADDR
0547 BC 87 1F 2930 MVI NBL050+@Q(,@XR),@UCB SET INDR FOR ABSOLUTE LOAD
054A B5 01 5C 2931 NBL060 L NBL080(,@XR),@BR BR = LOAD ADDRESS
054D F2 80 3E 054A 2932 NLOADB EQU NBL060 SPECIAL ENTRY FOR SFLOAD/SFFIND
2933 JC NBLZTR,@NOP JUMP IF TRACE REQUESTED
054E 2934 NTROVR EQU *-2 ADDR OF FE TRACE INDR
2935 *
0550 C0 87 0025 2936 NBLRTN B $DISKN START DISK READ OF PROGRAM
0554 0579 0555 2937 DC AL2(NBL070) ADDR OF DPL
0556 C0 87 0025 2938 B $DISKN WAIT FOR DISK I/O COMPLETION
055A 057F 055B 2939 DC AL2($WAITF) WAIT DPL ADDRESS
2940 *
055C AC 04 6B 6A 2941 MVC $FEMAP+@MAPEN(@MAPEN,@XR),$FEMAP+@MAPEN-1(,@XR)
2942 * * PUSH DOWN FE CORE MAP
0560 9C 00 66 06 2943 MVC $FEMAP(1,@XR),6(,@BR) SET PGM NUMBER IN FE MAP
0564 AE 01 5C 85 2944 ALC NBL080(@CADDR,@XR),NBL07(,@XR) SET EXECUTION ADDR
0568 F2 87 07 2945 NBL065 JC NBL067,@UCB GO TO EXECUTE PROGRAM
0569 2946 NBLNOE EQU NBL065+@Q ADDR OF NO EXECUTE INDR
056B BC 87 47 2947 MVI NBL065+@Q(,@XR),@UCB RESET SWITCH
056E C0 87 0000 2948 NBL066 B *-* RETURN TO CALLING PROGRAM
0571 2949 NLDRTN EQU NBL066+@OP1 ADDR OF RETURN ADDRESS
0572 B5 20 5C 2950 NBL067 L NBL080(,@XR),@P1IAR LOAD MAINLINE IAR
0575 C0 87 0480 2951 B $CIEXT
2952 *
2953 *** DPL'S FOR LOAD ROUTINE
2954 *
0579 0579 2955 NBL070 EQU * START OF PGM DPL
0579 2956 NBLDPL EQU NBL070 ADDR OF DPL TO LOAD PGM
057E 2957 NBL080 DS CL(@DPLNG) PROGRAM DPL
2958 *
057F FF 057F 2959 NWAITF DC XL1'FF' DISK WAIT I/O COMPLETE FUNC CODE
0580 01 0580 2960 NBL100 DC XL1'01' READ DISK FUNCTION CODE
0581 1880 0582 2961 DC AL2($GUFU) DISK ADDR OF GGUFUDI
0583 10 0583 2962 NGUFIO DC AL1($@GUF) SECTOR COUNT FOR GUFIDI

```



## NBLOAD - BLAST LOADER ROUTINES

ERR LOC OBJECT CODE ADDR STMT SOURCE STATEMENT VER 15, MOD 00 25/09/15 PAGE 74

0584	0C00	0585	2963	DC	AL2(\$\$KLD3)	LOAD ADDRESS FOR GUFUDI
0586		0587	2964	NDSADR DS	CL(@DADDR)	DISK ADDR RELOCATION FACTOR
0586			2965	ORG	*-2	PLACE INITIAL VALUE
0586	0000	0587	2966	DC	XL2'0000'	INITIAL VALUE
		0588	2967	NFEMAP EQU	*	1ST BYTE CORE MAP
0588		058D	2968	DS	CL(@MAPEN+1)	THE MAP
0588			2969	ORG	NFEMAP	PLACE INITIAL VALUE
0588	00000000000000	058D	2970	DC	XL(@MAPEN+1)'00'	INITIAL VALUE
		058E	2971	NBLZTR EQU	*	ENTRY TO LOAD + EXEC #ZTRAC
058E	B4 01 83		2972	ST	NBLZPL+@DBFR2(,@XR),@BR	SET LOAD ADDR IN DPL
0591	C0 87 0025		2973	B	\$DISKN	START DISK I/O OPERATION
0595	05A0	0596	2974	DC	AL2(NBLZPL)	ADDR OF DPL
0597	C0 87 0025		2975	B	\$DISKN	WAIT FOR DISK I/O COMPLETION
059B	057F	059C	2976	DC	AL2(\$WAITF)	WAIT DPL ADDRESS
			2977	*		
059D	D0 87 07		2978	B	@HDRLN(,@BR)	EXECUTE #ZTRAC
			2980	*BLZPL \$DPL	FUNC-@DGET,DADDR-#\$ZTRA,CNT-#\$@ZTR,CADDR-'*-*'	
		05A0	2981	+NBLZPL EQU	*	DISK PARAMETER LIST
05A0	01	05A0	2982	+	DC AL1(@DGET)	REQUESTED FUNCTION
05A1	1B9C	05A2	2983	+	DC AL2(#\$ZTRA)	DISK ADDRESS
05A3	01	05A3	2984	+	DC AL1(#\$@ZTR)	SECTOR COUNT
05A4	0000	05A5	2985	+	DC AL2(*-*)	BUFFER ADDRESS
			2986	+	*** END OF EXPANSION ***	
			2987	*		
05A6	0007	05A7	2988	NBLC07 DC	IL2'7'	EXECUTION ADDR DISP
			2989	***	ENTRY FROM NABORT	
05A8	38 01 03D1	2990	NAB300 TBN		\$XIND2,\$EXCMD	TEST PGM IN EXECUTION
05AC	F2 90 04	2991		JF	NAB400	JUMP IF NO
05AF	C0 87 0E24	2992		B	IS\$I700	GO TEST LINE PRINTER CONFIG.
05B3	3A 20 03C3	2993	NAB400 SBN		\$KEYCD,\$INRPT	SET PGM INTERRUPT INDR
05B7	C0 87 049D	2994		B	NCAIPL	RETURN TO ABORT ROUTINE
			2995	*		

## MOPPET - MODULE PROLOG

ERR	LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00	25/09/15	PAGE 75
				2997	*****			
				2998	*			*
				2999	* 5703-XM1 COPYRIGHT IBM CORP. 1970			*
				3000	* REFER TO INSTRUCTIONS ON COPYRIGHT NOTICE 120-2083			*
				3001	*			*
				3002	*****			*
				3003	*STATUS			*
				3004	* VERSION 1 MODIFICATION 0			*
				3005	*			*
				3006	*FUNCTION			*
				3007	* * MOPPET PROVIDES THE INTERFACE BETWEEN THE IPL PROGRAM, @MLOAD,			*
				3008	* AND THE NUCLEUS INITIALIZATION PROGRAM, #MIPPE.			*
				3009	* * THE SPF BASE DISK ADDRESS IS CALCULATED AND PLACED AT \$BSADR			*
				3010	* * NUCLEUS DPL DISK ADDRESS ARE UPDATED WITH THE SPF BASE ADDRESS.			*
				3011	* * NERLOG UPDATES THE OBR AND SDR ERROR LOGS.			*
				3012	* * #MIPPE IS LOADED AND EXECUTED.			*
				3013	*			*
				3014	*ENTRY POINTS			*
				3015	* MOPPET			*
				3016	* THIS ENTRY POINT IS THE ONLY ONE PROVIDED.			*
				3017	*			*
				3018	*INPUT			*
				3019	* MLOADS MUST SET CORE LOCATION \$IPLDV TO ZERO IF R1 IS IPL-ED, OR			*
				3020	* TO ONE OF F1 IS IPL-ED.			*
				3021	*			*
				3022	*OUTPUT			*
				3023	* N/A			*
				3024	*			*
				3025	*EXTERNAL REFERENCES			*
				3026	* \$NUCBS - BASE VALUE FOR COMMUNICATION AREA.			*
				3027	* \$BSADR - BASE DISK ADDRESS FOR SPF.			*
				3028	* \$CSDPL - GUFUDI DISK ADDRESS.			*
				3029	* NBLZPL - #ZTRAC DISK ADDRESS.			*
				3030	* NEROVL - NERLOG OVERLAY DISK ADDRESS.			*
				3031	*			*
				3032	*EXITS, NORMAL			*
				3033	* EXIT IS TO \$RLOAD TO LOAD AND EXECUTE #MIPPE.			*
				3034	*			*
				3035	*EXITS, ERROR			*
				3036	* N/A			*
				3037	*			*
				3038	*TABLES/WORK AREAS			*
				3039	* N/A			*
				3040	*			*
				3041	*ATTRIBUTES			*
				3042	* RELOCATABLE			*
				3043	*			*
				3044	*CHARACTER CODE DEPENDENCY			*
				3045	* N/A			*
				3046	*			*
				3047	*NOTES			*
				3048	* ERROR PROCEDURES			*
				3049	* N/A			*
				3050	*			*
				3051	* REGISTER USAGE			*
				3052	* INDEX REG 1 AND 2 ARE USED FOR BASE ADDRESSING.			*

MOPPET - MODULE PROLOG

ERR	LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00	25/09/15	PAGE	76
				3053	*				*
				3054	* SAVED/RESTORED AREAS				*
				3055	* N/A				*
				3056	*				*
				3057	* MODIFICATION CONSIDERATIONS				*
				3058	* N/A				*
				3059	*				*
				3060	* REQUIRED MODULES				*
				3061	* @SYSEQ - GENERAL SYSTEM EQUATES.				*
				3062	* @FXDEQ - NUCLEUS LOCATION EQUATES.				*
				3063	* NERLOG - ERROR LOGGNG ROUTINE.				*
				3064	* NBLOAD - BLAST LOADER.				*
				3065	* NPAUSE - PAUSE ROUTINE.				*
				3066	*				*
				3067	* OTHER				*
				3068	* NONE				*
				3069	*				*
				3070	*****				*

## MUPPET - IPL INTERFACE WITH MIPPER

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	25/09/15	PAGE 77
	05FF				3073	ORG	\$ENDNU-1			SET IPL INDR
	05FF	00		05FF	3074	DC	AL1(MOPREM)			SET TO REMOVABLE DISK
				03C0	3075	USING	\$NUCBS,@XR			INDEX BASE VALUE
				0622	3076	USING	MOPBSE,@BR			BASE ADDRESS SPECIFICATION
				0600	3077	MOPPET EQU	*			MODULE ENTRY POINT
	0600	C2	01	0622	3078	LA	MOPBSE,@BR			LOAD BASE REGISTER
	0604	C2	02	03C0	3079	LA	\$NUCBS,@XR			INITIALIZE XR
	0608	75	C0	40	3080	L	MOPILA(,@BR),@I1IAR			
	060B	1C	01	0003	3081	MVC	\$\$ZERO+@OP1(@CADDR),MOPRSS(,@BR)			SET EARLY RESET BR ADDR
	0610	3D	00	05FF	3082	CLI	MOPDSK,MOPREM			IPL FROM R1 ?
	0614	F2	81	0B	3083	JE	MOP030			JUMP IF YES
	0617	7A	01	33	3084	SBN	MOPDK1+@DSAD(,@BR),MOPSF1			SET DISK ADDRESS FOR FIXED
	061A	3A	01	03B1	3085	SBN	NEROVL+@DSAD,MOPSF1			SET NERLOG ADDRESS FOR FIXED
	061E	3A	01	0587	3086	SBN	\$BSADR,MOPSF1			SET FIXED DISK BIT IN SPF DADDR
	0622	C0	87	0025	3087	MOP030 B	\$DISKN			START DISK I/O OPERATION
	0626	0653		0627	3088	DC	AL2(MOPDK1)			ADDR OF DPL
	0628	C0	87	0025	3089	B	\$DISKN			WAIT FOR DISK I/O COMPLETION
	062C	057F		062D	3090	DC	AL2(\$WAITF)			WAIT DPL ADDRESS
	062E	38	80	0CFF	3091	TBN	MOPBF1+\$#TIDR,\$#TSYM			SYSTEM PROGRAM FILE ?
	0632	F2	10	06	3092	JT	MOP040			JUMP IF ON VOLUME
				0635	3093	MOP035 EQU	*			
					3094	*	\$HPL			NO SYSTEM PROGRAM FILE
	0635	F0		0635	3095+	DC	XL1'F0'			INLINE HPL INSTRUCTION
	0636	006C		0637	3096+	DC	AL2(@HIPLE)			HALT CODE
	0638	D0	87	13	3097	B	MOP035(,@BR)			SORRY, HARD HALT
	063B	0E	01	0587	0CFC	3099	MOP040 ALC			\$BSADR(@DADDR),MOPBF1+\$#TBIS SET SYSTEM DISK ADDRESS
	0641	0E	01	0510	0587	3100	ALC			\$CSDPL+@DSAD(@DADDR),\$BSADR SET TRUE DADDR FOR CORE SAVE
	0647	0E	01	05A2	0587	3101	ALC			NBLZPL+@DSAD(@DADDR),\$BSADR SET ZTRACE SPF DADDR
					3102	*	RLOAD			MOPDK2
	064D	C0	87	051E	3103	B	\$RLOAD			LOAD AND EXECUTE OVERLAY
	0651	0659		0652	3104	DC	AL2(MOPDK2)			DPL ADDRESS
				0622	3106	MOPBSE EQU	MOP030			MIPPER BASE ADDRESS
				05FF	3107	MOPDSK EQU	X'05FF'			IPL DISK INDR
				0000	3108	MOPREM EQU	0			INDR FOR IPL FROM REMOVABLE
					3109	*				
					3110	*OPDK1 \$DPL	FUNC-@DGET,DADDR-#VOLR1,CNT-#@VLAB,CADDR-MOPBF1			
				0653	3111+MOPDK1	EQU	*			DISK PARAMETER LIST
	0653	01		0653	3112+	DC	AL1(@DGET)			REQUESTED FUNCTION
	0654	0008		0655	3113+	DC	AL2(#VOLR1)			DISK ADDRESS
	0656	01		0656	3114+	DC	AL1(#@VLAB)			SECTOR COUNT
	0657	0C00		0658	3115+	DC	AL2(MOPBF1)			BUFFER ADDRESS
					3116+***	END OF EXPANSION ***				
					3117	*				
					3118	*OPDK2 \$DPL	FUNC-@DGET,DADDR-#\$MIPP,CNT-MOPCNT,CADDR-MOPBF1			
				0659	3119+MOPDK2	EQU	*			DISK PARAMETER LIST
	0659	01		0659	3120+	DC	AL1(@DGET)			REQUESTED FUNCTION
	065A	0A80		065B	3121+	DC	AL2(#\$MIPP)			DISK ADDRESS
	065C	14		065C	3122+	DC	AL1(MOPCNT)			SECTOR COUNT
	065D	0C00		065E	3123+	DC	AL2(MOPBF1)			BUFFER ADDRESS
					3124+***	END OF EXPANSION ***				
					3125	*				
				0001	3126	MOPSF1 EQU	X'01'			FIXED DISK SECTOR BIT
				0C00	3127	MOPBF1 EQU	\$\$KLD3			BUFFER ADDRESS
	065F	04BA		0660	3128	MOPRSS DC	AL2(\$PAUSD)			FE HALT ADDRESS

[illegible]

VER 15, MOD 00 25/09/15 PAGE 78

0661	0483	0014	3129	MOPCNT	EQU	020	COUNT TO LOAD MIPPER AND CLEAR
		0662	3130	MOPILA	DC	AL2(\$CIENT)	INTERRUPT RETURN ENTRY ADDRESS
			3131	*			

## NERLOG - OVERLAY ROUTINE

ERR LOC OBJECT CODE ADDR STMT SOURCE STATEMENT VER 15, MOD 00 25/09/15 PAGE 79

```

3133 *****
3134 *      NERLOG - OVERLAY ROUTINE      *
3135 *****
0700      3136      ORG  $$KLD2      LOCATE ROUTINE
          071A 3137      USING NERBSE,@BR      BASE REGISTER SPEC
          3138 *
0700 3139 NEROVR EQU *      ENTRY TO OVERLAY
0700 08 00 07C1 042E 3140      MZZ      NERDEV,$HISTE+#HISTQ      SAVE DEVICE ADDRESS
0706 C0 87 07DF      3141      B      NERSDR      DO SDR UPDATES
070A C2 01 071A      3142 NER200 LA      NERBSE,@BR      LOAD BASE REGISTER
070E C0 87 0025      3143      B      $DISKN      READ OBR TO BUFFER
0712 077F      0713 3144      DC      AL2(NERPL2)      DPL ADDRESS
0714 C0 87 0025      3145      B      $DISKN      WAIT FOR OBR
0718 057F      0719 3146      DC      AL2($WAITF)      WAIT DPL ADDRESS
071A C2 02 0A00      3147 NER220 LA      NERBUF,@XR      POINT TO BUFFER
071E B8 07 01      3148      TBN      #HSENT(,@XR),NERSCP      HAVE THE OBR POINTERS BEEN
0721 F2 10 0A      3149      JT      NER225      * ADJUSTED CORRECTLY ? YES-GO
0724 8F 01 01 0464 3150      SLC      #HSENT(2,@XR),$C0001      SUBTR 1 FROM NEXT AVAILABLE
0729 8F 01 03 0464 3151      SLC      #HISDX(2,@XR),$C0001      * ENTRY AND LAST ENTRY
072E AD 01 01 03      3152 NER225 CLC      #HSENT(2,@XR),#HISDX(,@XR)      END OF OBR TABLE ?
0732 F2 01 04      3153      JNE      NER230      NO - GO PREPARE TO LOG
0735 9C 01 01 8F      3154      MVC      #HSENT(2,@XR),NERENT(,@BR)      RESET PTR TO 1ST ENTRY OF TABLE
0739 B6 02 01      3155 NER230 A      #HSENT(,@XR),@XR      ADD DISPLACEMENT TO NEXT
073C 1E 01 0A01 92 3156      ALC      NERBUF+#HSENT(2),NERIDX(,@BR)      INCR POINTER TO NEXT
0741 8C 07 08 0435 3157      MVC      #HISLN(#HISLN,@XR),$HIST1      MOVE IN CURRENT OBR ENTRY
0746 38 08 03D5      3158      TBN      $INDR2,$DKERR      ANOTHER ENTRY TO BE PLACED
074A 3B 0C 03D5      3159      SBF      $INDR2,$DKERR+$ERPND      TURN OFF ERROR INDRS
074E F2 90 1A      3160      JF      NER250      JUMP IF OBR COMPLETE
          03FB 3161      USING $VOLID+#VOLNG-1,@XR      BASE VALUE TO $VOLID
0751 C2 02 03FB      3162      LA      $VOLID+#VOLNG-1,@XR      LOAD INDEX REGISTER
0755 6C 00 49 33      3163      MVC      NER240+@DD2(1,@BR),$HISTE+#HISTQ(,@XR)      SAVE DEVICE ADDR
0759 7B 07 49      3164      SBF      NER240+@DD2(,@BR),X'07'      GET DEVICE ADDRESS
075C 5F 00 49 93      3165      SLC      NER240+@DD2(1,@BR),NERVOL(,@BR)      CALC VOLID TABLE DISP
0760 AC 05 3A 00      3166 NER240 MVC      $HISTE+#HSEND(#VOLNG,@XR),*-(,@XR)      MOVE VOLID TO ENTRY
0764 AC 01 34 3C      3167      MVC      $HISTE+#HISTR(#DKEXT,@XR),$HISTE+#HISTN(,@XR)      PUT 2ND ENTRY
          3168 *      * AT $HISTE IN FIRST ENTRY
0768 D0 87 00      3169      B      NER220(,@BR)      GO LOG 2ND ENTRY

076B 7C 02 65      3171 NER250 MVI      NERPL2+@DCTRL(,@BR),@DPUT      SET DPL TO WRITE
076E C0 87 0025      3172      B      $DISKN      GO WRITE OBR
0772 077F      0773 3173      DC      AL2(NERPL2)      DPL ADDRESS
0774 38 20 03D2      3174      TBN      $IOIND,$HRDER      WAS THIS A HARD ERROR ?
0778 F2 10 41      3175      JT      NER260      DO HARD HALT IF YES
077B C0 87 0379      3176      B      NER100      RETURN TO CORE RESIDENT SECTION
          3177 *      * BEFORE DISK READ OVERTAKES US
          3178 ***      CONSTANTS AND EQUATES
          3179 *
          077F 3180 NERPL2 EQU *      ADDRESS OF OBR DPL
077F 01      077F 3181      DC      AL1(@DGET)      READ CNTL
0780 001D      0781 3182      DC      AL2(#OBRAD)      DADDR OF OBR
0782 02      0782 3183      DC      AL1(#@OBRA)      SECTOR COUNT OF OBR
0783 0A00      0784 3184      DC      AL2(NERBUF)      BUFFER ADDRESS
          0785 3185 NERPL3 EQU *      ADDRESS OF DPL
0785 01      0785 3186      DC      AL1(@DGET)      REQUESTED FUNCTION
0786 0011      0787 3187      DC      AL2(#SDRDK)      DISK ADDRESS
0788 01      0788 3188      DC      AL1(1)      SECTOR COUNT

```

## NERLOG - OVERLAY ROUTINE

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00 25/09/15 PAGE 80
0789	0A00			078A	3189	DC	AL2(NERBF1)	BUFFER ADDRESS
				078B	3190	NERPL4 EQU	*	ADDRESS OF DPL
078B	02			078B	3191	DC	AL1(@DPUT)	REQUESTED FUNCTION
078C	0011			078D	3192	DC	AL2(#SDRDK)	DISK ADDRESS
078E	01			078E	3193	DC	AL1(1)	SECTOR COUNT
078F	0A00			0790	3194	DC	AL2(NERBF1)	BUFFER ADDRESS
				0791	3195	NERPL5 EQU	*	ADDRESS OF DPL
0791	01			0791	3196	DC	AL1(@DGET)	REQUESTED FUNCTION
0792	000C			0793	3197	DC	AL2(#VLSDR)	DISK ADDRESS
0794	01			0794	3198	DC	AL1(1)	SECTOR COUNT
0795	0A00			0796	3199	DC	AL2(NERBF1)	BUFFER ADDRESS
				0797	3200	NERPL6 EQU	*	ADDRESS OF DPL
0797	01			0797	3201	DC	AL1(@DGET)	REQUESTED FUNCTION
0798	0019			0799	3202	DC	AL2(#IOSDR)	DISK ADDRESS
079A	01			079A	3203	DC	AL1(1)	SECTOR COUNT
079B	0A00			079C	3204	DC	AL2(NERBF2)	BUFFER ADDRESS
				079D	3205	NERPL7 EQU	*	ADDRESS OF DPL
079D	01			079D	3206	DC	AL1(@DGET)	REQUESTED FUNCTION
079E	0008			079F	3207	DC	AL2(#VOLR1)	DISK ADDRESS
07A0	01			07A0	3208	DC	AL1(1)	SECTOR COUNT
07A1	0A00			07A2	3209	DC	AL2(NERBF1)	BUFFER ADDRESS
07A3	0C			07A3	3211	NERLCT DC	IL1'12'	SUSPECT TRACK TABLE COUNT
07A4	00			07A4	3212	NERDSP DC	AL1(*-*)	DSL DISK BITS FOR DSAD
07A5				07A6	3213	NERSHF DS	XL2	SNS BYTE SHIFT WORK AREA
07A7	02			07A7	3214	NERTWO DC	IL1'2'	TWO
07A8	0007			07A9	3215	NERENT DC	AL2(#HSEND)	DISPLACEMENT TO FIRST ENTRY
07AA	00			07AA	3216	NERDSB DC	AL1(*-*)	DISK BITS (DEFAULT R1)
07AB	0008			07AC	3217	NERIDX DC	AL2(#HISLN)	LENGTH OF AN OBR ENTRY
07AD	A0			07AD	3218	NERVOL DC	AL1(@SPINA)	MASK OUT DEVICE
				071A	3219	NERBSE EQU	NER220	BASE ADDRESS
				0002	3220	NERDEL EQU	2	LENGTG SDR ENTRY
				0010	3221	NERPED EQU	X'10'	DISP TO PERMANENT DISK SDR
				0040	3222	NERD2D EQU	X'40'	DISP TO DRIVE 1 SDR COUNTERS
				0008	3223	NERFIX EQU	X'08'	FIXED DISK BIT IN Q CODE
				0020	3224	NERFXD EQU	X'20'	DISP TO FIXED DISK SDR COUNTERS
				0001	3225	NERKYD EQU	X'01'	DISP TO KEYBOARD SDR COUNTERS
				0009	3226	NERCTD EQU	X'09'	DISP TO CRT SDR COUNTERS
				0045	3227	NERDCD EQU	X'45'	DISP TO DTRCDR COMPARE SDR CNTR
				0041	3228	NERDSD EQU	X'41'	DISP TO DTRCDR NOT READY SDR
				0011	3229	NERPHD EQU	X'11'	DISP TO MP HORZ CYCLE SDR CNTR
				0007	3230	NERSCP EQU	X'07'	BITS 5-7 ON IN OBR POINTER
				3231	*			* SCP INITIALIZED BEFORE MOD 3
				000C	3232	NERPPE EQU	X'0C'	DISP TO MP PERMANENT HORZ CYCLE
				0A00	3233	NERBF1 EQU	NEROVR+3*256	START OF DISK SDR BUFFER
				0A00	3234	NERBF2 EQU	NERBF1	START OF NON-DISK SDR BUFFER
				0A00	3235	NERBUF EQU	NEROVR+3*256	BUFFER ADDRESS
				0078	3236	NERHLQ EQU	X'78'	HALT INDR ABCD
				0040	3237	NERHD1 EQU	X'40'	DISK 1 INDR (1)
				0044	3238	NERHD2 EQU	X'44'	DISK 2 INDR (15)
				0048	3239	NERHKY EQU	X'48'	KEYBOARD INDR (14)
				004C	3240	NERHPR EQU	X'4C'	PRINTER INDER (145)
				0054	3241	NERHPU EQU	X'54'	PUNCH INDR (135)
				0058	3242	NERHCR EQU	X'58'	CRT INDR (134)
				00FF	3243	NEREND EQU	X'FF'	TERMINATOR
				07AE	3244	NERTBL EQU	*	DEVICE ADDR, HALT TABLE



## NERLOG - OVERLAY ROUTINE

ERR LOC OBJECT CODE ADDR STMT SOURCE STATEMENT VER 15, MOD 00 25/09/15 PAGE 81

07AE	A0		07AE	3245	DC	AL1(@SPINA)	DISK 1
07AF	40		07AF	3246	DC	AL1(NERHD1)	HALT Q CODE
07B0	B0		07B0	3247	DC	AL1(@SPINB)	DISK 2
07B1	44		07B1	3248	DC	AL1(NERHD2)	HALT Q CODE
07B2	10		07B2	3249	DC	AL1(@KEYBD)	KEYBOARD
07B3	48		07B3	3250	DC	AL1(NERHKY)	HALT Q CODE
07B4	E0		07B4	3251	DC	AL1(@PSIOQ)	PRINTER
07B5	4C		07B5	3252	DC	AL1(NERHPR)	HALT Q CODE
07B6	F0		07B6	3253	DC	AL1(@CD37B)	PUNCH
07B7	54		07B7	3254	DC	AL1(NERHPU)	HALT Q CODE
07B8	90		07B8	3255	DC	AL1(@CRTQ)	CRT
07B9	58		07B9	3256	DC	AL1(NERHCR)	HALT Q CODE
07BA	FFFF		07BB	3257	DC	2AL1(NEREND)	END OF TABLE
				3258	*		
07BC	C2 02	07AE		3259	NER260 LA	NERTBL,@XR	POINT TO DEVICE HALT TABLE
07C0	BD 00	00		3260	NER270 CLI	0(,@XR),*-*	IS THIS THE DEVICE ?
			07C1	3261	NERDEV EQU	NER270+@Q	DEVICE
07C3	F2 81	09		3262		JE NER280	JUMP IF YES
07C6	E2 02	02		3263		LA 2(,@XR),@XR	POINT TO NEXT ENTRY
07C9	BD FF	00		3264		CLI 0(,@XR),NEREND	END OF TABLE ?
07CC	D0 01	A6		3265		BNE NER270(,@BR)	GO CHECK NEXT DEVICE
				3266	*	ENTRY FOUND, SET HALT LIGHTS	
07CF	6C 00	BB 01		3267	NER280 MVC	NER290+@D1(,@BR),1(1,@XR)	SET HALT LIGHTS
			07D3	3268	NER290 EQU	*	
				3269	*	\$HPL CODE-NERHLQ	HALT I/O ERROR
07D3	F0		07D3	3270+	DC	XL1'F0'	INLINE HPL INSTRUCTION
07D4	0078		07D5	3271+	DC	AL2(NERHLQ)	HALT CODE
07D6	D0 87	B9		3272	B	NER290(,@BR)	SORRY, HARD HALT
				3274	*ERSTR \$DPL	FUNC-@DGET,DADDR-#CORSV,CNT-#@CORS,CADDR-\$\$KLD2	
			07D9	3275+	NERSTR EQU	*	DISK PARAMETER LIST
07D9	01		07D9	3276+	DC	AL1(@DGET)	REQUESTED FUNCTION
07DA	0010		07DB	3277+	DC	AL2(#CORSV)	DISK ADDRESS
07DC	05		07DC	3278+	DC	AL1(#@CORS)	SECTOR COUNT
07DD	0700		07DE	3279+	DC	AL2(\$\$KLD2)	BUFFER ADDRESS
				3280+	***	END OF EXPANSION ***	
				3281	*		

## NERLOG - UPDATE SDR TABLES

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	25/09/15	PAGE 82
				03C0	3283		USING \$NUCBS,@BR			BASE REGISTER SPECIFICATION
				07DF	3284	NERSDR	EQU *			ENTRY OF SDR UPDATE
07DF	C2	01	03C0		3285		LA \$NUCBS,@BR			LOAD BASE REGISTER
07E3	78	08	15		3286		TBN \$INDR2(,@BR), \$DKERR			IS THIS A DISK ERROR ?
07E6	C0	90	08E0		3287		BF NER500			IF YES DO OTHER DEVICES
07EA	C0	87	0025		3288		B \$DISKN			READ MASTER DISK SDR
07EE	0785			07EF	3289		DC AL2(NERPL3)			DPL ADDRESS
07F0	C2	02	0A01		3290		LA NERBF1+1,@XR			POINT XR TO FIRST SDR ENTRY
07F4	78	04	71		3291		TBN \$HISTE+#HISN1(,@BR),@OVRUN			OVERRUN ERROR ?
07F7	F2	10	30		3292		JT NER320			UPDATE IF YES
07FA	E2	02	04		3293		LA 2*NERDEL(,@XR),@XR			POINT TO DATA CHECK ENTRY
07FD	78	08	70		3294		TBN \$HISTE+#HISN1-1(,@BR),@DERD2			DATA CHECK ERROR ?
0800	F2	90	09		3295		JF NER310			SKIP TO NO RECORD FOUND IF NO
0803	78	01	71		3296		TBN \$HISTE+#HISN1(,@BR),@DREAD			WAS IT A READ OPERATION ?
0806	F2	90	21		3297		JF NER320			UPDATE WRITE CNTR IF NO,
0809	F2	87	1B		3298		J NER315			ELSE UPDATE READ COUNTER
					3299	*				
080C	E2	02	04		3300	NER310	LA 2*NERDEL(,@XR),@XR			POINT TO NO-RECORD-FOUND ENTRY
080F	78	04	71		3301		TBN \$HISTE+#HISN1(,@BR),@DERNR			NO-RECORD-FOUNF ERROR ?
0812	F2	10	15		3302		JT NER320			UPDATE IF YES
0815	E2	02	02		3303		LA NERDEL(,@XR),@XR			POINT TO EQUIPMENT CHECK
0818	78	10	70		3304		TBN \$HISTE+#HISN1-1(,@BR),@DEREQ			EQUIP CHECK ERROR ?
081B	F2	10	0C		3305		JT NER320			UPDATE IF YES
081E	E2	02	02		3306		LA NERDEL(,@XR),@XR			POINT TO MISSING ADDR MARK
0821	78	20	70		3307		TBN \$HISTE+#HISN1-1(,@BR),@DERMA			MISSING ADDR MARK ERROR ?
0824	F2	10	03		3308		JT NER320			UPDATE IF YES
0827	E2	02	02		3309	NER315	LA NERDEL(,@XR),@XR			POINT TO NEXT COUNTER
082A	78	20	12		3310	NER320	TBN \$IOIND(,@BR), \$HRDER			IS IT PERMANENT ERROR ?
082D	F2	90	06		3311		JF NER330			GO CHECK DISK IF NO
0830	7C	00	74		3312		MVI \$HISTE+#HISCT(,@BR),@ZERO			INDICATE HARD ERROR IN OBR
0833	E2	02	10		3313		LA NERPED(,@XR),@XR			POINT TO PERMANENT COUNTERS
0836	3D	A0	07C1		3314	NER330	CLI NERDEV,@SPINA			IS IT DRIVE 1 ?
083A	F2	81	07		3315		JE NER340			CHECK DISK IF YES
083D	3C	02	07AA		3316		MVI NERDSB,2*@B1			SET DISK BIT FOR DRIVE 2
0841	E2	02	40		3317		LA NERD2D(,@XR),@XR			POINT TO DRIVE 2 COUNTERS
0844	78	08	6E		3318	NER340	TBN \$HISTE+#HISTQ(,@BR),NERFIX			IS IT A FIXED DISK ?
0847	F2	90	07		3319		JF NER350			GO DO UPDATE IF NOT
084A	3A	01	07AA		3320		SBN NERDSB,@B1			SET FIXED DISK BIT
084E	E2	02	20		3321		LA NERFXD(,@XR),@XR			POINT TO FIXED DISK COUNTERS
0851	C0	87	0025		3322	NER350	B \$DISKN			WAIT ON DISK SDR READ
0855	057F			0856	3323		DC AL2(\$WAITF)			WAIT DPL ADDRESS
0857	9E	01	00 A4		3324		ALC 0(2,@XR), \$C0001(,@BR)			ADD ONE TO DISK ERROR COUNTER
085B	C0	87	0025		3325		B \$DISKN			WRITE DISK SDR
085F	078B			0860	3326		DC AL2(NERPL4)			DPL ADDRESS
0861	78	20	12		3327		TBN \$IOIND(,@BR), \$HRDER			WAS THIS A HARD DISK ERROR ?
0864	F2	10	24		3328		JT NER400			LOG SUSPECTED TRACK IF YES
0867	0E	00	0793 07AA		3329		ALC NERPL5+@DSAD(1),NERDSB			SET DISK BITS
086D	C0	87	0025		3330		B \$DISKN			READ VOLUME STAT COUNTERS
0871	0791			0872	3331		DC AL2(NERPL5)			DPL ADDRESS
0873	C0	87	0025		3332		B \$DISKN			WAIT ON OP COMPLETION
0877	057F			0878	3333		DC AL2(\$WAITF)			WAIT ADDRESS
0879	1E	01	0A03 A4		3334		ALC NERBF1+3, \$C0001(2,@BR)			BUMP COUNTER
087E	3C	02	0791		3335		MVI NERPL5+@DCTRL,@DPUT			SET WRITE CNTL
0882	C0	87	0025		3336		B \$DISKN			WRITE VOLUME STAT COUNTERS
0886	0785			0887	3337		DC AL2(NERPL3)			DPL ADDRESS
0888	F2	87	51		3338		J NER450			GO DO OBR ENTRY

## NERLOG - UPDATE SDR TABLES

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	25/09/15	PAGE 83
					3339	*				
				088B	3340	NER400 EQU	*			
088B	0E	00	079F 07AA		3341	ALC	NERPL7+@DSAD,NERDSB(1)		SET DISK BITS	
0891	C0	87	0025		3342	B	\$DISKN		READ VOLUME	
0895	079D			0896	3343	DC	AL2(NERPL7)		DPL ADDRESS	
0897	C0	87	0025		3344	B	\$DISKN		WAIT ON OP COMPLETION	
089B	057F			089C	3345	DC	AL2(\$WAITF)		WAIT ADDRESS	
089D	4D	00	75 0A5C		3346	CLC	\$HISTE+#HISTC(1,@BR),NERBF1+\$#TCYL		CHECK IF INVALID CYL	
08A2	F2	84	37		3347	JH	NER450		HIGH INVALID, DO NOT LOG	
08A5	C2	02	0AD7		3348	LA	NERBF1+\$#TSUS-12*@DADDR,@XR		POINT TO SUSPECT TRK TABLE	
08A9	1C	01	07A6 76		3349	MVC	NERSHF,\$HISTE+#HISTS(@DADDR,@BR)		MOVE TRACK ADDR TO WORK	
08AE	3B	7F	07A6		3350	SBF	NERSHF,X'7F'		* AREA AND SET SECTOR BITS OFF	
08B2	E2	02	02		3351	NER420 LA	@DADDR(,@XR),@XR		POINT TO NEXT ENTRY	
08B5	8D	01	00 07A6		3352	CLC	0(@DADDR,@XR),NERSHF		IS TRACK ALREADY LOGGED ?	
08BA	F2	81	1F		3353	JE	NER450		SKIP LOG OPERATION IF YES	
08BD	1F	00	07A3 A4		3354	SLC	NERLCT,\$C0001(1,@BR)		DECREMENT LOOP COUNTER	
08C2	C0	84	08B2		3355	BH	NER420		IF MORE -- BRANCH	
08C6	0C	15	0AEF 0AED		3356	MVC	NERBF1+\$#TSUS(11*@DADDR),NERBF1+\$#TSUS-@DADDR		PUSH DOWN	
08CC	0C	01	0AD9 07A6		3357	MVC	NERBF1+\$#TSUS-11*@DADDR(@DADDR),NERSHF		TABLE & PLACE ENTRY	
08D2	3C	02	079D		3358	MVI	NERPL7+@DCTRL,@DPUT		SET CNTL TO WRITE	
08D6	C0	87	0025		3359	B	\$DISKN		WRITE VOLUME	
08DA	079D			08DB	3360	DC	AL2(NERPL7)		DPL ADDRESS	
08DC	C0	87	070A		3361	NER450 B	NER200		GO DO OBR UPDATE	

## NERLOG - UPDATE SDR TABLES

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	25/09/15	PAGE 84
				08E0	3363	NER500	EQU *			ENTRY TO LOG NON-DISK SDR
	08E0	C0 87 0025			3364		B \$DISKN			READ NON-DISK SDR
	08E4	0797		08E5	3365		DC AL2(NERPL6)			DPL ADDRESS
	08E6	C0 87 0025			3366		B \$DISKN			WAIT ON OP COMPLETION
	08EA	057F		08EB	3367		DC AL2(\$WAITF)			WAIT ADDRESS
	08EC	C2 02 0A00			3368		LA NERBF2,@XR			POINT TO BUFFER
	08F0	3D 10 07C1			3369		CLI NERDEV,@KEYBD			IS IT FROM KEYBOARD ?
	08F4	F2 01 04			3370		JNE NER510			IF NO CHECK NEXT DEVICE
	08F7	9E 01 01 A4			3371		ALC NERKYD(2,@XR),\$C0001(,@BR)			BUMP KB ERROR COUNTER
	08FB	3D 90 07C1			3372	NER510	CLI NERDEV,@CRTQ			IS IT FROM CRT ?
	08FF	F2 01 04			3373		JNE NER520			IF NO CHECK NEXT DEVICE
	0902	9E 01 09 A4			3374		ALC NERCTD(2,@XR),\$C0001(,@BR)			BUMP CRT ERROR COUNTER
	0906	3D F0 07C1			3375	NER520	CLI NERDEV,@CD37B			IS IT FROM DATA RECORDER ?
	090A	F2 01 11			3376		JNE NER530			IF NO CHECK NEXT DEVICE
	090D	78 04 71			3377		TBN \$HISTE+#HISN1(,@BR),@CP37B			COMPARE ERROR
	0910	F2 90 07			3378		JF NER525			DO NOT COUNT READY
	0913	9E 01 45 A4			3379		ALC NERDCD(2,@XR),\$C0001(,@BR)			BUMP DATA RECORDER COUNTER
	0917	F2 87 36			3380		J NER560			GO WRITE SDR
					3381	*				
	091A	9E 01 45 A4			3382	NER525	ALC NERDCD(2,@XR),\$C0001(,@BR)			BUMP 'NOT READY' COUNTER
	091E	3D E0 07C1			3383	NER530	CLI NERDEV,@PSIOQ			IS IT FROM PRINTER ?
	0922	F2 01 2B			3384		JNE NER560			IF NO GO WRITE SDR
	0925	E2 02 11			3385		LA NERPHD(,@XR),@XR			POINT TO TEMP HORZ CYCLE CHK
	0928	1C 00 07A6 71			3386		MVC NERSHF,\$HISTE+#HISN1(1,@BR)			SET SNS BYTE FOR SHIFT
	092D	0E 00 07A6 07A6			3387	NER533	ALC NERSHF(1),NERSHF			SHIFT SNS BITS
	0933	F2 A0 0A			3388		JOL NER540			SKIP OUT WHEN BIT INCOUNTED
	0936	0E 00 0942 07A7			3389		ALC NER540+@D1(1),NERTWO			BUMP INDEX COUNTER
	093C	C0 20 092D			3390		BNOL NER533			GO LOOP (FALL THROUGH IF ERROR)
	0940	E2 02 00			3391	NER540	LA *-*(,@XR),@XR			POINT TO TEMP CNTR
	0943	78 20 12			3392		TBN \$IOIND(,@BR),\$HRDER			IS IT A PERMANENT ERROR ?
	0946	F2 90 03			3393		JF NER550			IF NO UPDATE TEMP COUNTERS
	0949	E2 02 0C			3394		LA NERPPE(,@XR),@XR			POINT TO PERMANENT COUNTERS
	094C	9E 01 00 A4			3395	NER550	ALC 0(,@XR),\$C0001(2,@BR)			BUMP PRINTER ERROR COUNTER
	0950	3C 02 0797			3396	NER560	MVI NERPL6+@DCTRL,@DPUT			SET WRITE OP
	0954	C0 87 0025			3397		B \$DISKN			WRITE SDR
	0958	0797		0959	3398		DC AL2(NERPL6)			DPL ADDRESS
	095A	C0 87 070A			3399		B NER200			GO DO OBR UPDATE
					3400	*				
	0C00				3402		ORG X'0C00'			CYL 0, TRK 1, SCTR 12
					3403	*				
					3404	***	INITIALIZE TO ZERO			
					3405	*				
	0C00	0000000000000000	0EFF	3406			DC 3XL256'00'			CYL 0, TRK 1, SCTR 13 - 15
	0F00	0000000000000000	16FF	3407			DC 8XL256'00'			CYL 0, TRK 1, SCTR 16 - 23
	1700	0000000000000000	17FF	3408			DC 64XL4'00'			CYL 0, TRK 1, SCTR 24
			FFFF	3409			END			

TOTAL STATEMENTS IN ERROR IN THIS ASSEMBLY = 0

## CROSS REFERENCE

VER 15, MOD 00 25/09/15 PAGE 85

SYMBOL	LEN	VALUE	DEFN	REFERENCES
\$\$\$CMD	001	0020	1396	
\$\$\$DAT	001	0040	1395	
\$\$\$EPL	001	0091	1392	
\$\$\$ERN	001	0080	1446	
\$\$\$FUN	001	0010	1397	
\$\$\$NLN	001	00A0	1442	
\$\$\$STD	001	0081	1391	
\$\$BNLN	001	0605	1372	1374
\$\$CDBS	001	08C0	1422	
\$\$CDND	001	0666	1381	
\$\$CDRD	001	0890	1420	1422
\$\$CKEY	001	0603	1370	
\$\$CKFF	001	0B3D	1402	
\$\$COFF	001	0B44	1401	
\$\$CSNS	001	209C	1431	
\$\$DATB	001	0BBF	1403	
\$\$EOSA	001	0AFE	1400	
\$\$ERSK	001	1C00	1441	
\$\$FITS	001	1D00	1449	
\$\$FLIB	001	06FF	1448	
\$\$ILEN	001	0601	1366	1368 1372
\$\$ILHD	001	0600	1364	1366
\$\$INLN	001	0607	1379	1381 1383
\$\$INND	001	06FA	1383	
\$\$KBDT	001	09E1	1390	1394
\$\$KBSN	001	09E2	1394	1399
\$\$KLD1	001	0600	1454	
\$\$KLD2	001	0700	1456	2541 2549 3136 3279
\$\$KLD3	001	0C00	1458	2819 2963 3127
\$\$LPOS	001	09EB	1399	
\$\$PCNT	001	07E9	1415	
\$\$PLYN	001	2004	1429	
\$\$PRES	001	0890	1388	1390 1400 1401 1402 1403 1420
\$\$PRFL	001	2143	1433	
\$\$PRNT	001	0707	1409	1410 1414 1415 2691
\$\$PRTN	001	0782	1410	
\$\$PSIO	001	07CE	1414	
\$\$PYCD	001	2200	1435	
\$\$PYMP	001	2000	1427	1429 1431 1433 1435
\$\$SLIB	001	1C00	1444	
\$\$TPCD	001	0606	1374	1379
\$\$UPAR	001	0602	1368	1370
\$\$WSPB	001	1E00	1447	
\$\$XIND	001	06FF	1445	1448
\$\$ZERO	001	0000	0232	0233 0235 0236 0237 0241 1427 3081*
\$#TALT	001	0075	1475	
\$#TBIS	001	00FC	1487	3099
\$#TCET	001	0069	1474	
\$#TCYL	001	005C	1473	3346
\$#THAD	001	00F2	1479	
\$#THEL	001	0004	1499	
\$#THVT	001	00F0	1478	
\$#TIDR	001	00FF	1489	3091
\$#TLAD	001	00FE	1488	
\$#TLBL	001	0008	1470	
\$#TLIB	001	00F8	1484	

## CROSS REFERENCE

VER 15, MOD 00 25/09/15 PAGE 86

SYMBOL	LEN	VALUE	DEFN	REFERENCES
\$#TLIF	001	0010	1497	
\$#TLSZ	001	00F7	1483	
\$#TOID	001	005B	1472	
\$#TPAD	001	00F6	1482	
\$#TPFL	001	0008	1498	
\$#TPSZ	001	00F4	1481	
\$#TPTF	001	00F3	1480	
\$#TRES	001	00D7	1491	
\$#TSUS	001	00EF	1477	3348 3356 3356* 3357*
\$#TSYM	001	0080	1494	3091
\$#TSYS	001	00FA	1486	
\$#TUSE	001	00A8	1476	
\$#TVOL	001	0002	1469	
\$#TVTC	001	000A	1471	
\$#TWAL	001	00D7	1490	
\$#TWF1	001	0020	1496	
\$#TWRK	001	00F9	1485	
\$#TWR1	001	0040	1495	
\$ABORT	001	0010	0345	
\$BASIC	001	0080	0403	
\$BIGCD	001	0080	0479	
\$BLDPL	001	0579	0612	0614
\$BLNOE	001	0569	0602	
\$BLOAD	001	0522	0593	0595 0598 0611 0612 2715 2764
\$BLRTN	001	0550	0601	0602
\$BRSAV	001	03C5	0290	0291
\$BSADR	001	0587	0617	0619 2929 3086* 3099* 3100 3101
\$BUFPT	001	03E3	0498	0499 2628
\$CABLD	001	04B4	0571	0572
\$CAERK	001	0469	0548	0551
\$CAERR	001	03CD	0296	0298
\$CAIPL	001	049D	0567	0569
\$CALLI	001	0008	0488	
\$CARDI	001	0001	0259	2756
\$CARPL	001	04A1	0569	0571
\$CIENT	001	0483	0558	0559 3130
\$CIEXT	001	0480	0557	0558 2951
\$CIMSK	001	0476	0554	0557 1925 1926* 1989* 2746* 2759*
\$CISUS	001	0496	0562	0567 2730* 2758*
\$CLBFR	001	0010	0446	
\$CMDKY	001	0008	0358	2760
\$CMODE	001	0002	0408	
\$CONFIG	001	03DD	0471	0481 2615
\$CRPOS	001	03E2	0497	0498 2625
\$CRTAD	001	044D	0536	0537
\$CRTAV	001	0002	0352	
\$CRTDN	001	0002	0376	
\$CRTIN	001	03D3	0373	0380
\$CRTNO	001	0004	0355	
\$CRTPU	001	0004	0377	
\$CRTSP	001	0008	0378	
\$CRTUP	001	0001	0375	
\$CRUSH	001	0080	0484	
\$CSDPL	001	050E	0583	0584 2781* 2785* 2788 3100*
\$C0001	001	0464	0540	0546 2920 2924 3150 3151 3324 3334 3354 3371 3374 3379 3382 3395



VER 15, MOD 00 25/09/15 PAGE 87

SYMBOL	LEN	VALUE	DEFN	REFERENCES				VER 15, MOD 00					25/09/15		PAGE	87
\$DATE	001	043A	0521	0522												
\$DBGUF	001	03E0	0483	0492	2619	2731*										
\$DBLOK	001	0001	0433													
\$DFDET	001	03E8	0504	0505 2643												
\$DISKN	001	0025	0235	2505	2507	2509	2516	2518	2787	2789	2936	2938	2973	2975	3087	
				3089	3143	3145	3172	3288	3322	3325	3330	3332	3336	3342	3344	
				3359	3364	3366	3397									
\$DKERR	001	0008	0414	2347	2349	3158	3159	3286								
\$DKSIZ	001	03D7	0458	0466	0507	2605										
\$DK100	001	0001	0460													
\$DK200	001	0002	0461	2606												
\$DK400	001	0004	0462													
\$DK600	001	0008	0463													
\$DK800	001	0010	0464													
\$DPLSV	001	0449	0532	0534												
\$DTNMB	001	0040	0279													
\$DTRDR	001	0040	0367	2502												
\$ENDNU	001	0600	0626	1364	1388	1409	1445	1454	1456	1458	2813	3073				
\$ERDPL	001	046F	0551	0553	2716											
\$ERFIL	001	0040	0306													
\$ERHRD	001	0004	0438													
\$ERKEY	001	0080	0310													
\$ERLOG	001	0345	0240	1991	2160	2493	2534									
\$ERMAD	001	0472	0553	0554												
\$ERPND	001	0004	0411	1990	2347	2349	3159									
\$ERRCT	001	03CF	0312													
\$ERRPG	001	03CE	0300													
\$ERSFL	001	0035	0305													
\$ERSTK	001	0030	0303													
\$ER050	001	0363	0241													
\$ER1N2	001	0050	0308													
\$EXADR	001	0517	0586	0588												
\$EXCMD	001	0001	0340	2761	2990											
\$EXFTR	001	043B	0522	0527	2686											
\$FCIND	001	0010	0418													
\$FDIND	001	0040	0425													
\$FEARR	001	0004	0233	2805												
\$FEMAP	001	0588	0619	0620	2941	2941*	2943*									
\$FILIB	001	03DA	0469	0470	2611											
\$FITIN	001	0010	0394													
\$FUIND	001	0020	0423													
\$GUFIO	001	0583	0616	0617												
\$GUFIR	001	0008	0268													
\$HISTE	001	042E	0519	0520	2351*	2355*	2356*	2357*	2361*	3140	3163	3166*	3167	3167*	3291	
				3294	3296	3301	3304	3307	3312*	3318	3346	3349	3377	3386		
\$HIST1	001	0435	0520	0521	3157											
\$HRDER	001	0020	0364	2155	2359	3174	3310	3327	3392							
\$INDR1	001	03D4	0380	0406												
\$INDR2	001	03D5	0406	0431	1990	2347	2349*	2599	3158	3159*						
\$INDR3	001	03D6	0431	0458	2602											
\$INLNO	001	03CF	0298	0300	0312	0319										
\$INRPT	001	0020	0276	2993												
\$IOIND	001	03D2	0347	0373	2155*	2359	2502	2760*	3174	3310	3327	3392				
\$IOPGS	001	0010	0487													
\$IOYES	001	0002	0262	2756												
\$IPLDV	001	05FF	0623	0626												



## CROSS REFERENCE

VER 15, MOD 00 25/09/15 PAGE 88

SYMBOL	LEN	VALUE	DEFN	REFERENCES
\$IRKEY	001	0020	0486	2731
\$KEYBD	001	03E1	0492	0497 2622
\$KEYCD	001	03C3	0256	0290 2756* 2993*
\$KEYDT	001	0040	0400	
\$KE090	001	00DE	0236	
\$KE130	001	01D5	0237	
\$KYBSY	001	0010	0273	
\$LDRTN	001	0571	0611	
\$LEVEL	001	03DF	0481	0483
\$LIST	001	0002	0435	
\$LMRGN	001	03C1	0251	0253
\$LNPTR	001	0080	0370	
\$LOADB	001	054A	0595	
\$LOADR	001	051A	0588	0591 2912
\$LPRIO	001	03EA	0505	2646
\$LPROS	001	03E5	0500	0502 2634
\$LPRP3	001	03E4	0499	0500 2631
\$MOUNT	001	0020	0449	
\$MPDWN	001	0001	0349	
\$NEXTB	001	03E6	0502	0503 2637
\$NEXTL	001	03E7	0503	0504 2640
\$NOENB	001	0008	0441	
\$NOLST	001	0004	0265	2756
\$NUCBS	001	03C0	0248	0249 2346 2347 2349* 2351* 2355* 2356* 2357* 2359 2361* 2559 3075 3079 3283 3285
\$NWRKF	001	0080	0454	
\$NWRKR	001	0040	0451	
\$PASWD	001	042D	0518	0519 2678
\$PAUSD	001	04BA	0572	0574 2771 3128
\$PAUSE	001	0002	0342	
\$PGMDT	001	0020	0397	
\$PGMST	001	0010	0361	
\$PKERT	001	0419	0516	0518 2083
\$PLST1	001	0454	0537	0538 1948*
\$PLST2	001	045B	0538	0539 1947
\$PLST3	001	0462	0539	0540 1947*
\$PRDEV	001	044B	0534	0536 2690 2708
\$PRESN	001	0002	0385	
\$PROCI	001	0001	0382	
\$PRPOS	001	03C2	0253	0256
\$PSDBR	001	04FA	0577	
\$PSDXR	001	04F2	0576	0577
\$PSTEP	001	0004	0343	
\$PSTMT	001	0008	0344	
\$PTCH1	001	03F5	0507	0511
\$READY	001	0080	0427	
\$REORD	001	0040	0485	
\$RLOAD	001	051E	0591	0593 2803 3103
\$RMRGN	001	03C0	0249	0251
\$RSTR	001	04D6	0574	0576 0578 0583
\$RUNIT	001	0001	0321	
\$SFAID	001	050D	0579	
\$SPRNT	001	0465	0546	0548 2706
\$SRTRN	001	04FE	0578	0579
\$STEPT	001	0002	0322	
\$SWPCR	001	0511	0584	0586 2810

## CROSS REFERENCE

VER 15, MOD 00 25/09/15 PAGE 89

SYMBOL	LEN	VALUE	DEFN	REFERENCES
\$TABLN	001	03CB	0293	0296
\$TFLOW	001	0008	0328	
\$TRACE	001	0004	0323	
\$TRALL	001	0010	0329	
\$TROVR	001	054E	0598	0601
\$TRUNK	001	0080	0281	
\$TRVAR	001	0020	0330	
\$UNMSK	001	048D	0559	0562 2743 2772 2777 2784
\$USRDR	001	03DC	0470	0471
\$VMDEF	001	0080	0334	
\$VOLF1	001	03FE	0513	0514
\$VOLF2	001	040E	0515	
\$VOLID	001	03F6	0511	0512 0516 2662 3161 3162
\$VOLR1	001	03F6	0512	0513
\$VOLR2	001	0406	0514	0515
\$WAITF	001	057F	0614	0616 2510 2519 2790 2939 2976 3090 3146 3323 3333 3345 3367
\$WFDEF	001	0040	0528	
\$WFLOK	001	0008	0391	
\$WFNME	001	0443	0527	0532
\$WSIND	001	0004	0388	
\$XIND1	001	03D0	0319	0338 2596
\$XIND2	001	03D1	0338	0347 2761 2990
\$XIND3	001	03D8	0466	0469 2608
\$XPREC	001	0040	0331	
\$XRSAB	001	03C7	0291	0293 2921*
\$ZTRAD	001	05A2	0620	
\$12K	001	0004	0475	
\$16CKY	001	0008	0477	
\$16K	001	0002	0474	
\$22IMP	001	0001	0472	
###BL	001	0000	1217	
###CK	001	0000	1345	
###CN	001	0000	1313	
###CO	001	0000	1105	
###CS	001	0000	1165	
###DR	001	0000	0909	
###ER	001	0000	1109	
###FS	001	0000	1205	
###IN	001	0000	1349	
###PW	001	0000	1353	
###RS	001	0000	1185	
###SA	001	0000	1173	
###SS	001	0000	1169	
###VU	001	0600	1129	
###0T	001	0700	0901	
###1T	001	0000	0905	
###BCO	001	0600	0917	
###BOV	001	0800	1189	
###DPR	001	0700	0925	
###DRE	001	0889	0941	
###DSP	001	2800	0961	
###ECM	001	0C00	1221	
###EFK	001	0C00	1241	
###ERR	001	0C00	1213	
###EXM	001	0C00	1101	
###FIL	001	0E00	1181	

## CROSS REFERENCE

SYMBOL   LEN   VALUE   DEFN   REFERENCES   VER 15, MOD 00   25/09/15   PAGE   90

\$\$\$FIS	001	0E00	1177	
\$\$\$FML	001	0200	1309	
\$\$\$FMS	001	0200	1149	
\$\$\$GRA	001	0889	1073	
\$\$\$GUF	001	0C00	1209	
\$\$\$INL	001	0600	1289	
\$\$\$INS	001	0600	0913	
\$\$\$KAL	001	0C00	1077	
\$\$\$KCA	001	0C00	1293	
\$\$\$KCH	001	0C00	1045	
\$\$\$KCN	001	0C00	1161	
\$\$\$KCT	001	0C00	1013	
\$\$\$KDE	001	0C00	1009	
\$\$\$KDI	001	0D00	1089	
\$\$\$KDN	001	0C00	0997	
\$\$\$KDO	001	0E00	1093	
\$\$\$KED	001	0C00	0933	
\$\$\$KEN	001	0C00	0937	
\$\$\$KEX	001	0C00	0957	
\$\$\$KGO	001	0C00	0929	
\$\$\$KHE	001	0C00	1113	
\$\$\$KKE	001	0C00	1341	
\$\$\$KLI	001	0C00	1017	
\$\$\$KLL	001	0920	1317	
\$\$\$KLO	001	0C00	1021	
\$\$\$KME	001	0D00	1001	
\$\$\$KMO	001	0C00	0945	
\$\$\$KNA	001	0C00	1057	
\$\$\$KOV	001	0E00	0977	
\$\$\$KPA	001	0C00	0953	
\$\$\$KPO	001	0C00	1041	
\$\$\$KPR	001	0C00	1065	
\$\$\$KRE	001	0C00	0985	
\$\$\$KRL	001	0700	1081	
\$\$\$KRM	001	0C00	0949	
\$\$\$KRN	001	0700	0969	
\$\$\$KRO	001	0D00	0973	
\$\$\$KRS	001	0C00	1297	
\$\$\$KRU	001	0C00	0993	
\$\$\$KRV	001	0800	1085	
\$\$\$KSA	001	0C00	1029	
\$\$\$KSE	001	0E00	1069	
\$\$\$KSO	001	0C20	1121	
\$\$\$KSS	001	0C00	1053	
\$\$\$KSV	001	0980	1049	
\$\$\$KSY	001	0C00	1061	
\$\$\$KWI	001	0C00	0989	
\$\$\$KWR	001	0C00	0981	
\$\$\$LOA	001	0600	0921	
\$\$\$MIP	001	0C00	1117	
\$\$\$SDS	001	0C00	1229	
\$\$\$SFF	001	0E00	1233	
\$\$\$SFL	001	0F00	1225	
\$\$\$SFO	001	1500	1197	
\$\$\$SFS	001	0C00	1193	
\$\$\$SPA	001	0C00	1033	

## CROSS REFERENCE

SYMBOL LEN VALUE DEFN REFERENCES VER 15, MOD 00 25/09/15 PAGE 91

###SPO	001	0806	1037	
###SPS	001	0C00	1025	
###STR	001	1600	1201	
###TDC	001	1000	1005	
###TSY	001	1000	0965	
###TVK	001	0FC0	1141	
###UAL	001	0C00	1157	
###UAT	001	0900	1253	
###UCD	001	0900	1261	
###UCN	001	0C00	1245	
###UCP	001	0700	1249	
###UDE	001	0C00	1265	
###UDI	001	0C00	1269	
###UEX	001	0C00	1153	
###UIN	001	0C00	1257	
###UPA	001	0C00	1237	
###UPO	001	0C00	1305	
###UPT	001	0C00	1301	
###VCR	001	2000	1097	
###VLO	001	0600	1133	
###VOD	001	0600	1137	
###VVM	001	0000	1145	
###VXI	001	0600	1125	
###ZDU	001	1100	1277	
###ZLB	001	1100	1321	
###ZLO	001	1100	1281	
###ZLV	001	0F00	1337	
###ZL1	001	0F00	1325	
###ZL2	001	0F00	1329	
###ZL3	001	0C00	1333	
###ZTR	001	1000	1273	
###ZUT	001	0C00	1285	
##BLN	001	18D4	1216	
##CKT	001	2118	1344	
##CNF	001	2000	1312	
##COR	001	0800	1104	2811
##CSA	001	1000	1164	
##DRT	001	0000	0908	
##ERM	001	0928	1108	
##FSP	001	1880	1204	
##INV	001	212C	1348	
##PWR	001	2300	1352	
##RSP	001	1780	1184	
##SAV	001	1180	1172	
##SSA	001	1128	1168	
##VUF	001	0B08	1128	
##0TR	001	0000	0900	
##1TR	001	0080	0904	
##@BL	001	0001	1218	
##@CK	001	0004	1346	
##@CN	001	0001	1314	
##@CO	001	003A	1106	
##@CS	001	003A	1166	
##@DR	001	0008	0910	
##@ER	001	0032	1110	
##@FS	001	0030	1206	

## CROSS REFERENCE

SYMBOL   LEN   VALUE   DEFN   REFERENCES   VER 15, MOD 00   25/09/15   PAGE   92

##\$@#IN	001	003A	1350	
##\$@#PW	001	00C0	1354	
##\$@#RS	001	0030	1186	
##\$@#SA	001	0108	1174	
##\$@#SS	001	0001	1170	
##\$@#VU	001	0002	1130	
##\$@#0T	001	0018	0902	
##\$@#1T	001	0018	0906	
##\$@BCO	001	0018	0918	
##\$@BOV	001	0018	1190	
##\$@DPR	001	0005	0926	
##\$@DRE	001	0001	0942	
##\$@DSP	001	0004	0962	
##\$@ECM	001	0006	1222	
##\$@EFK	001	0002	1242	
##\$@ERR	001	0003	1214	2721
##\$@EXM	001	0003	1102	2818
##\$@FIL	001	0009	1182	
##\$@FIS	001	0009	1178	
##\$@FML	001	0052	1310	
##\$@FMS	001	0052	1150	
##\$@GRA	001	0003	1074	
##\$@GUF	001	0010	1210	2962
##\$@INL	001	0010	1290	
##\$@INS	001	0010	0914	
##\$@KAL	001	000F	1078	
##\$@KCA	001	000C	1294	
##\$@KCH	001	000C	1046	
##\$@KCN	001	0010	1162	
##\$@KCT	001	0009	1014	
##\$@KDE	001	0010	1010	
##\$@KDI	001	0005	1090	
##\$@KDN	001	0010	0998	
##\$@KDO	001	000C	1094	
##\$@KED	001	000E	0934	
##\$@KEN	001	0006	0938	
##\$@KEX	001	0003	0958	
##\$@KGO	001	0002	0930	
##\$@KHE	001	000C	1114	
##\$@KKE	001	0006	1342	
##\$@KLI	001	0011	1018	
##\$@KLL	001	0001	1318	
##\$@KLO	001	0008	1022	
##\$@KME	001	0003	1002	
##\$@KMO	001	0004	0946	
##\$@KNA	001	0008	1058	
##\$@KOV	001	0009	0978	
##\$@KPA	001	0005	0954	
##\$@KPO	001	000D	1042	
##\$@KPR	001	0009	1066	
##\$@KRE	001	0002	0986	
##\$@KRL	001	0004	1082	
##\$@KRM	001	0003	0950	
##\$@KRN	001	0003	0970	
##\$@KRO	001	000A	0974	
##\$@KRS	001	000A	1298	

## CROSS REFERENCE

SYMBOL LEN VALUE DEFN REFERENCES VER 15, MOD 00 25/09/15 PAGE 93

#\$@KRU	001	0003	0994
#\$@KRV	001	000D	1086
#\$@KSA	001	0011	1030
#\$@KSE	001	0004	1070
#\$@KSO	001	000D	1122
#\$@KSS	001	000B	1054
#\$@KSV	001	0002	1050
#\$@KSY	001	000F	1062
#\$@KWI	001	0002	0990
#\$@KWR	001	0002	0982
#\$@LOA	001	0013	0922
#\$@MIP	001	000D	1118
#\$@SDS	001	0004	1230
#\$@SFF	001	0008	1234
#\$@SFL	001	0005	1226
#\$@SFO	001	0003	1198
#\$@SFS	001	0011	1194
#\$@SPA	001	0004	1034
#\$@SPO	001	0003	1038
#\$@SPS	001	0001	1026
#\$@STR	001	0002	1202
#\$@TDC	001	0003	1006
#\$@TSY	001	0003	0966
#\$@TVK	001	0001	1142
#\$@UAL	001	0011	1158
#\$@UAT	001	000C	1254
#\$@UCD	001	000B	1262
#\$@UCN	001	0009	1246
#\$@UCP	001	000F	1250
#\$@UDE	001	000E	1266
#\$@UDI	001	0008	1270
#\$@UEX	001	000E	1154
#\$@UIN	001	000F	1258
#\$@UPA	001	0004	1238
#\$@UPO	001	0005	1306
#\$@UPT	001	0012	1302
#\$@VCR	001	0008	1098
#\$@VLO	001	0002	1134
#\$@VOD	001	0016	1138
#\$@VVM	001	0030	1146
#\$@VXI	001	0002	1126
#\$@ZDU	001	0008	1278
#\$@ZLB	001	0002	1322
#\$@ZLO	001	000C	1282
#\$@ZLV	001	0006	1338
#\$@ZL1	001	0007	1326
#\$@ZL2	001	000D	1330
#\$@ZL3	001	000A	1334
#\$@ZTR	001	0001	1274
#\$@ZUT	001	0014	1286
#\$BCOM	001	0080	0916
#\$BOLV	001	1780	1188
#\$DPRI	001	014C	0924
#\$DREA	001	0200	0940
#\$DSPL	001	0240	0960
#\$ECMA	001	1900	1220

2984

## CROSS REFERENCE

VER 15, MOD 00 25/09/15 PAGE 94

SYMBOL	LEN	VALUE	DEFN	REFERENCES
#\$EFKE	001	1990	1240	
#\$ERRP	001	18C0	1212	2720
#\$EXMS	001	07D4	1100	2817
#\$FILN	001	1724	1180	
#\$FIST	001	1700	1176	
#\$FMLN	001	1E00	1308	
#\$FMST	001	0D00	1148	
#\$GRAP	001	0690	1072	
#\$GUFU	001	1880	1208	2961
#\$INLN	001	1C84	1288	
#\$INST	001	0020	0912	
#\$KALL	001	06A4	1076	
#\$KCAL	001	1CC4	1292	
#\$KCHA	001	053C	1044	
#\$KCND	001	0F80	1160	
#\$KCTL	001	03BC	1012	
#\$KDEL	001	035C	1008	
#\$KDIS	001	0744	1088	
#\$KDNT	001	0300	0996	
#\$KDOV	001	0780	1092	
#\$KEDI	001	0188	0932	
#\$KENA	001	01C4	0936	
#\$KEXT	001	0234	0956	
#\$KGOS	001	0180	0928	
#\$KHEL	001	0A30	1112	
#\$KKEY	001	2100	1340	
#\$KLIS	001	0400	1016	
#\$KLLA	001	2004	1316	
#\$KLOG	001	0444	1020	
#\$KMER	001	030C	1000	
#\$KMOU	001	0204	0944	
#\$KNAM	001	05C0	1056	
#\$KOVN	001	0290	0976	
#\$KPAS	001	0220	0952	
#\$KPOO	001	0508	1040	
#\$KPRT	001	063C	1064	
#\$KREA	001	02BC	0984	
#\$KRLA	001	0700	1080	
#\$KRMO	001	0214	0948	
#\$KRNU	001	0280	0968	
#\$KROV	001	028C	0972	
#\$KRSU	001	1D24	1296	
#\$KRUN	001	02CC	0992	
#\$KRVL	001	0710	1084	
#\$KSAV	001	0488	1028	
#\$KSET	001	0680	1068	
#\$KSOV	001	0AC8	1120	
#\$KSSP	001	0594	1052	
#\$KSVL	001	058C	1048	
#\$KSYM	001	0600	1060	
#\$KWID	001	02C4	0988	
#\$KWRI	001	02B4	0980	
#\$LOAD	001	0100	0920	
#\$MIPP	001	0A80	1116	3121
#\$SDSY	001	192C	1228	
#\$SFFI	001	193C	1232	



## CROSS REFERENCE

VER 15, MOD 00 25/09/15 PAGE 95

SYMBOL	LEN	VALUE	DEFN	REFERENCES
##\$SFLO	001	1918	1224	
##\$SFOV	001	1844	1196	
##\$SFSY	001	1800	1192	
##\$SPAC	001	04CC	1032	
##\$SPOV	001	04DC	1036	
##\$SPSY	001	0484	1024	
##\$STRO	001	1850	1200	
##\$TDCK	001	0350	1004	
##\$TSYK	001	0250	0964	
##\$TVKB	001	0BAC	1140	
##\$UALL	001	0F00	1156	
##\$UATR	001	1A38	1252	
##\$UCDI	001	1AD8	1260	
##\$UCNF	001	19B8	1244	
##\$UCPL	001	19DC	1248	
##\$UDEL	001	1B24	1264	
##\$UDIS	001	1B5C	1268	
##\$UEXL	001	0EA8	1152	
##\$UINI	001	1A88	1256	
##\$UPAC	001	1980	1236	
##\$UPOV	001	1D24	1304	
##\$UPTF	001	1D5C	1300	
##\$VCRT	001	07B4	1096	
##\$VLOA	001	0B80	1132	
##\$VODK	001	0B88	1136	
##\$VVMR	001	0C00	1144	
##\$VXIT	001	0B00	1124	
##\$ZDUM	001	1BA4	1276	
##\$ZLBM	001	2008	1320	
##\$ZLOA	001	1BC4	1280	
##\$ZLVR	001	20B0	1336	
##\$ZL1M	001	2010	1324	
##\$ZL2M	001	2030	1328	
##\$ZL3M	001	2088	1332	
##\$ZTRA	001	1B9C	1272	2983
##\$ZUTM	001	1C14	1284	
##1TRK	001	0000	0003	
##@CORS	001	0005	0866	2540 3278
##@MVSD	001	0001	0874	
##@NERO	001	0003	0868	2548
##@OBRA	001	0002	0870	3183
##@PTFL	001	0006	0889	
##@PTFS	001	0001	0888	
##@VCNT	001	0002	0886	
##@VLAB	001	0001	0881	3114
##@VLSD	001	0001	0872	
##CNDIS	001	0001	0841	
##CNFIG	001	0005	0877	
##CORSV	001	0010	0865	2539 3277
##DKEXT	001	0002	0848	2682 3167
##FIGSC	001	0001	0878	
##HISCT	001	0006	0855	2356* 2361* 3312*
##HISDX	001	0003	0850	3151* 3152
##HISLN	001	0008	0847	0848 2681 3157 3157* 3217
##HISN1	001	0003	0853	3291 3294 3296 3301 3304 3307 3377 3386
##HISN2	001	0005	0854	

## CROSS REFERENCE

VER 15, MOD 00 25/09/15 PAGE 96

SYMBOL	LEN	VALUE	DEFN	REFERENCES
#HISTC	001	0007	0857	3346
#HISTN	001	0009	0859	2357* 3167
#HISTQ	001	0000	0851	3140 3163 3318
#HISTR	001	0001	0852	3167*
#HISTS	001	0008	0858	3349
#HISTV	001	000F	0860	
#HSEND	001	0007	0856	3166* 3215
#HSENT	001	0001	0849	3148 3150* 3152 3154* 3155 3156*
#IOSDR	001	0019	0876	3202
#MVSDR	001	000D	0873	
#NEROV	001	009C	0867	2547
#OBRAD	001	001D	0869	3182
#PKCNT	001	0002	0834	
#PKMRW	001	002B	0835	
#PKRDD	001	0003	0832	
#PKRTD	001	0003	0831	
#PKRTL	001	0004	0838	2658 2659 2660 2661 2672 2673 2674 2675
#PKVRD	001	000B	0836	
#PKVWD	001	0007	0837	
#PKWTD	001	0001	0833	
#PTFDA	001	00DC	0887	
#RDWTL	001	0004	0839	
#SDRDK	001	0011	0875	3187 3192
#VLSDR	001	000C	0871	3197
#VLTBE	001	0008	0826	2653 2654 2655 2656
#VOLF1	001	0009	0879	
#VOLNG	001	0006	0824	0826 0848 2663 2665 2667 2669 3161 3162 3166
#VOLOC	001	0005	0825	
#VOLR1	001	0008	0880	3113 3207
#VTCF1	001	0025	0883	
#VTCF2	001	0027	0885	
#VTCR1	001	0024	0882	
#VTCR2	001	0026	0884	
@ALTFL	001	0001	0667	2208 2215 2238 2308
@ARR	001	0008	0024	1917* 1918 1919* 1920 2070 2080 2100 2376* 2745 2774 2920* 2923 2924* 2925
@ASIGN	001	007C	0079	
@ASTER	001	005C	0077	
@BCRDL	001	0050	0096	
@BE	001	0081	0051	
@BF	001	0090	0060	
@BH	001	0084	0049	
@BKSPC	001	0010	0764	
@BL	001	0082	0050	
@BLANK	001	0040	0073	
@BM	001	0082	0062	
@BNE	001	0001	0054	
@BNH	001	0004	0052	
@BNL	001	0002	0053	
@BNM	001	0002	0065	
@BNOL	001	0020	0058	
@BNOZ	001	0008	0057	
@BNP	001	0004	0064	
@BNZ	001	0001	0066	
@BOL	001	00A0	0056	
@BOZ	001	0088	0055	

CROSS REFERENCE																					
SYMBOL	LEN	VALUE	DEFN	REFERENCES												VER 15, MOD 00	25/09/15	PAGE	97		
@BP	001	0084	0061																		
@BR	001	0001	0021	1910	1914	1915*	1916	1917	1919	1920	1925	1930	1932	1934	1936						
				1936	1937	1937	1943	1943	1944	1945	1948	1949	1949	1950	1950						
				1951	1956	1960	1960	1965	1965	1966	1967	1968	1968	1969	1969						
				1970	1970	1971	1971	1972	1974	1974	1975	1975	1976	1976	1980						
				1981	1995*	2070	2071	2071	2079	2080	2081	2084	2085	2086	2086						
				2087	2087	2088	2088	2089	2089	2096	2098	2098	2100	2101	2101						
				2102	2102	2103	2106	2106	2107	2107	2108	2109	2109	2110	2112						
				2112	2120	2122	2122	2123	2123	2124	2125	2127	2127	2133	2135						
				2137	2139	2143	2145	2156	2158	2166	2168	2178	2179	2180	2184						
				2187	2194	2194	2196	2196	2197	2198	2200	2207	2208	2213	2215						
				2216	2217	2220	2227	2229	2230	2230	2232	2234	2234	2236	2237						
				2238	2240	2242	2251	2253	2254	2254	2256	2257	2259	2267	2275						
				2280	2280	2282	2282	2283	2283	2287	2293	2339	2341	2351	2352						
				2355	2356	2357	2360	2361	2362	2363	2364	2374	2376	2495	2497						
				2498	2772	2776	2777*	2778	2778	2779	2780	2781	2784*	2785	2786						
				2794	2794	2795*	2926*	2927	2931*	2943	2972	2978	3076	3078*	3080						
				3081	3084	3097	3137	3142*	3154	3156	3163	3164	3165	3165	3169						
				3171	3265	3267	3272	3283	3285*	3286	3291	3294	3296	3301	3304						
				3307	3310	3312	3318	3324	3327	3334	3346	3349	3354	3371	3374						
				3377	3379	3382	3386	3392	3395												
@BT	001	0010	0059	2520																	
@BZ	001	0081	0063																		
@BZ37B	001	00F2	0777	2504																	
@B1	001	0001	0071	3316	3320																
@CADDR	001	0002	0150	2207	2220	2312	2499	2521	2689	2692	2778	2794	2944	3081							
@CARDL	001	0060	0095	1381																	
@CC37B	001	0000	0773																		
@CD37B	001	00F0	0791	3253	3375																
@CHARA	001	00C1	0080																		
@CHARF	001	00C6	0081																		
@CHARR	001	00D9	0082																		
@CHARZ	001	00E9	0083																		
@CKY01	001	0001	0725																		
@CKY02	001	0002	0726																		
@CKY03	001	0003	0727																		
@CKY04	001	0004	0728																		
@CKY05	001	0005	0729																		
@CKY06	001	0006	073																		

## CROSS REFERENCE

VER 15, MOD 00 25/09/15 PAGE 98

SYMBOL	LEN	VALUE	DEFN	REFERENCES
@CRERR	001	0090	0759	
@CRPRY	001	0004	0763	
@CRTDS	001	0092	0756	
@CRTQ	001	0090	0758	3255 3372
@CURSR	001	0040	0760	
@DADDR	001	0002	0148	2098 2127 2178 2187 2191 2217 2219 2237 2305 2311 2610 2613 2664 2666 2668 2670 2809 2929 2964 3099 3100 3101 3348 3349 3351 3352 3356 3356 3357 3357*
@DBFR1	001	0004	0137	
@DBFR2	001	0005	0138	1936* 1944 1944* 1948 1968 1968 2049 2085 2207* 2220* 2229* 2230 2280 2280* 2927 2972* 2072 2185
@DBUSY	001	0002	0661	
@DCALK	001	0001	0089	
@DCBCY	001	0009	0123	
@DCBT1	001	0050	0125	
@DCFLN	001	0004	0645	2017 2019 2020 2086 2179
@DCNT	001	0003	0136	1949* 2086 2194
@DCRID	001	0001	0659	2183
@DCST1	001	0040	0124	
@DCTRL	001	0000	0133	1934 1945 1950 1951* 1972 2081 2088 2198 2257 2267 2282* 2781* 2785* 3171* 3335* 3358* 3396*
@DCTRW	001	0000	0658	
@DCWID	001	0001	0655	
@DCYL	001	0001	0134	
@DCYMV	001	0001	0646	2103 2108
@DD2	001	0003	0038	1955* 1958* 1961 3163* 3164* 3165*
@DEFLG	001	0002	0668	2213
@DERCE	001	0020	0698	2145
@DERD2	001	0008	0690	2139 3294
@DEREQ	001	0010	0689	2133 3304
@DERIN	001	0040	0687	2135
@DERMA	001	0020	0688	2139 3307
@DERNR	001	0004	0691	2139 3301
@DERR	001	0000	0662	2073 2186
@DERSC	001	0001	0693	2143 2352
@DERTC	001	0002	0692	2139
@DFCR	001	0006	0648	2110* 2156*
@DFDR	001	0004	0649	2085* 2193 2229
@DGET	001	0001	0142	2546 2785 2982 3112 3120 3181 3186 3196 3201 3206 3276
@DHARD	001	0000	0676	2047
@DLNCT	001	000F	0762	
@DLNLG	001	0040	0761	
@DOLAR	001	005B	0076	
@DOP2	001	0004	0036	
@DPLNG	001	0006	0140	1936 1944 1947 1948 2010 2012 2693 2694 2695 2927 2957
@DPOS	001	0000	0141	2081 2158 2257
@DPUT	001	0002	0143	1934 1972 2198 2538 2781 3171 3191 3335 3358 3396
@DREAD	001	0001	0652	2181 2267 3296
@DSAD	001	0002	0135	1956 1965 1967* 2098 3084* 3085* 3100* 3101* 3329* 3341*
@DSBCY	001	0004	0114	
@DSBSY	001	0092	0757	
@DSCS1	001	0000	0115	
@DSEEK	001	0000	0651	2111 2157 2255
@DSIVF	001	0003	0146	
@DSPIN	001	0002	0139	1956
@DTRSZ	001	0018	0093	

## CROSS REFERENCE

VER 15, MOD 00 25/09/15 PAGE 99

SYMBOL	LEN	VALUE	DEFN	REFERENCES
@DUNSF	001	0080	0694	
@DVBCY	001	0007	0116	
@DVERY	001	0003	0657	2275
@DVRFY	001	0031	0144	
@DVST1	001	0002	0663	2124
@DVST2	001	0003	0664	2125
@DWAIT	001	00FF	0145	
@DWBCY	001	0005	0111	
@DWRTT	001	0002	0653	
@DWSIZ	001	00C0	0113	
@DWTB1	001	0003	0112	
@DZERO	001	00F0	0072	
@D1	001	0002	0034	1937* 1961* 1971* 1974* 2089* 2275 2283* 2351 2355 3267* 3389*
@EOF	001	001C	0085	
@EOFTC	001	0075	0170	
@EOS	001	001E	0084	
@ER37B	001	00F0	0778	
@FDDBC	001	0000	0203	
@FDE1	001	000C	0208	
@FDFNA	001	000B	0206	
@FDHLN	001	0002	0216	
@FDLNC	001	0002	0201	
@FDNSC	001	0003	0218	
@FDSD	001	0000	0214	
@FLACE	001	0009	0205	
@FLDBC	001	0001	0204	
@FLDIN	001	0012	0750	
@FLENT	001	0004	0209	
@FLFNA	001	0002	0207	
@FLHLN	001	0002	0217	
@FLLNC	001	0002	0202	
@FLNSC	001	0001	0219	
@FLSD	001	0001	0215	
@HCEPK	001	003C	1532	
@HCOPS	001	001C	1539	
@HCOPY	001	081C	1534	
@HCRHE	001	7858	1555	
@HDNRY	001	1008	1520	
@HDRHE	001	7854	1553	
@HDRLN	001	0007	0100	1409 2978
@HDRV1	001	7840	1545	
@HDRV2	001	7844	1547	
@HDTRD	001	1040	1516	
@HDTRJ	001	1010	1518	
@HERPG	001	087C	1522	
@HFEHT	001	0804	1537	2800
@HIPLE	001	006C	1529	3096
@HKBER	001	2040	1512	
@HKBHE	001	7848	1549	
@HLOGE	001	1844	1524	2530
@HPRER	001	0070	1514	
@HPRHE	001	784C	1551	
@HSTAD	001	0009	0674	
@HSTEN	001	0007	0673	
@HSTPE	001	0006	0672	
@HSTQR	001	0001	0670	2351* 2355*

## CROSS REFERENCE

VER 15, MOD 00 25/09/15 PAGE 100

SYMBOL	LEN	VALUE	DEFN	REFERENCES
@HSTSN	001	0005	0671	
@HSTVI	001	000F	0675	
@HUNSF	001	1850	1527	2380
@IAR	001	0010	0025	2708*
@ID37B	001	0040	0814	
@INDEX	001	0001	0164	0165
@INST3	001	0003	0040	
@INST4	001	0004	0041	
@INST5	001	0005	0042	
@INST6	001	0006	0043	
@IP37B	001	00C0	0813	
@I1IAR	001	00C0	0028	3080*
@KCMDK	001	0020	0724	
@KELOK	001	001B	0723	2732
@KENAB	001	001E	0721	
@KEXIT	001	001F	0722	
@KEYBD	001	0010	0741	2500 2522 2732 2737 3249 3369
@KFUNK	001	0010	0744	
@KHARD	001	0011	0749	
@KLEAR	001	000D	0745	
@LINSZ	001	00F4	0092	1383
@LO37B	001	00F0	0782	
@MAPEN	001	0005	0097	2941 2941 2941* 2968 2970
@MINCR	001	2000	0091	
@MINUS	001	0060	0088	
@NOP	001	0080	0048	1926 2498 2728 2730 2786 2916 2918 2933
@NORFL	001	0000	0669	2242
@NTRDY	001	00A0	0806	
@NUMBR	001	007B	0078	
@OPD2	001	0004	0037	
@OP1	001	0003	0035	1914* 1916* 1918* 1920* 2070* 2080* 2100* 2497 2499* 2521* 2745* 2774* 2776* 2778 2778* 2779* 2794 2923* 2925* 2949 3081*
@OP2	001	0005	0039	
@OVRUN	001	0004	0699	2137 3291
@PBUSY	001	00E2	0711	2501
@PCAR	001	00E6	0708	
@PCNT	001	0003	0643	2086* 2101* 2102* 2106* 2107* 2127 2179 2179* 2194 2196* 2197* 2217 2217* 2234 2234* 2254*
@PCTRL	001	0000	0157	2158*
@PCYL	001	0001	0641	2101 2107 2112 2201 2201
@PC37B	001	00F2	0798	
@PDAR	001	00E4	0707	
@PDATA	001	0003	0159	
@PD37B	001	0080	0812	
@PERR	001	00E0	0714	
@PFLAG	001	0000	0640	1932* 2208* 2213 2215* 2238 2242*
@PFORM	001	00E1	0712	
@PGCSZ	001	0020	0090	0091
@PLITE	001	00E2	0713	
@PLNGH	001	0004	0704	
@PMGCK	001	0020	0715	
@PN37B	001	00F0	0797	
@PPLNG	001	0004	0156	
@PRCNT	001	0001	0158	
@PRETR	001	00C0	0162	
@PRINT	001	0040	0160	0162

## CROSS REFERENCE

VER 15, MOD 00 25/09/15 PAGE 101

SYMBOL	LEN	VALUE	DEFN	REFERENCES
@PRITY	001	0080	0748	
@PSAD	001	0002	0642	2098* 2103* 2108* 2178 2187 2205 2219* 2232 2236* 2237* 2240 2929*
@PSIOQ	001	00E0	0710	3251 3383
@PSIOR	001	0000	0709	
@PSNSQ	001	00E2	0716	
@PSR	001	0004	0023	
@PWAIT	001	00FF	0166	
@P1IAR	001	0020	0026	2950*
@P2IAR	001	0040	0027	
@Q	001	0001	0032	1925* 2071* 2087* 2088* 2109* 2122* 2123* 2180* 2181* 2184* 2253* 2498* 2520* 2729 2748 2780* 2786* 2916* 2918* 2930* 2946 2947* 3261
@RD37B	001	00F1	0792	
@REGL	001	0002	0020	2578 2579 2739
@RETRN	001	0080	0161	0162
@RLDWN	001	004F	0167	
@RTCNT	001	0003	0706	
@RTRNC	001	0080	0169	
@RT37B	001	0005	0805	
@SBLN	001	0005	0178	
@SBLNL	001	0002	0192	
@SCTSΖ	001	0100	0108	
@SDFLN	001	0007	0098	
@SDF0	001	0000	0174	
@SDF1	001	0001	0175	
@SDF2	001	0002	0176	
@SDF3	001	0003	0177	
@SECCY	001	0030	0094	
@SIST	001	0001	0189	
@SKCTL	001	0000	0656	2111 2157 2255
@SLASH	001	0061	0075	
@SLAST	001	0002	0191	
@SMIDL	001	0003	0190	
@SNSB0	001	0000	0680	
@SNSB1	001	0001	0681	2124* 2137 2145
@SNSB2	001	0002	0682	
@SNSB3	001	0003	0683	2125* 2376
@SNULL	001	0080	0181	
@SN37B	001	00F2	0786	
@SONLY	001	0000	0188	
@SPINA	001	00A0	0665	2041 2061 2072 2085* 2110* 2156* 2157 2185 2193 2229 3218 3245 3314
@SPINB	001	00B0	0666	3247
@STEXT	001	0007	0180	
@STYPE	001	0006	0179	
@SYCNT	001	0002	0705	
@SYLVL	001	0005	0008	2617
@TBCNT	001	0000	0168	
@TBLEF	001	0010	0163	0165
@TBLIX	001	0011	0165	
@TJ37B	001	0040	0803	
@TYPAM	001	0002	0747	
@TYPO	001	001C	0746	
@UCB	001	0087	0047	2746 2747 2758 2759 2780 2928 2930 2945 2947
@UPARW	001	005A	0086	
@VADDR	001	0002	0149	
@VENTA	001	0056	0121	



## CROSS REFERENCE

VER 15, MOD 00 25/09/15 PAGE 102

SYMBOL	LEN	VALUE	DEFN	REFERENCES
@VMDDV	001	00FE	0122	
@VMFD1	001	0000	0117	
@VMFD2	001	0001	0118	
@VMRS3	001	0002	0120	
@VMTRL	001	0001	0119	
@VOLID	001	0006	0099	
@VQ	001	0001	0033	
@WA37B	001	00FF	0811	
@WSFIT	001	0500	0109	
@WSTBL	001	0503	0110	
@XR	001	0002	0022	1916 1942* 1944 1996* 2083* 2084 2129* 2168 2176 2178 2179 2180 2181 2184 2186 2187 2189 2190 2191 2191 2193 2197 2200 2201 2201 2205 2205 2207 2217 2219 2219 2220 2237 2241 2249 2251 2253 2265 2287 2293 2295 2296 2346* 2347 2349 2351 2355 2356 2357 2359 2361 2779 2793* 2913 2921 2922* 2923 2925 2927 2929 2929 2930 2931 2941 2941 2943 2944 2944 2947 2950 2972 3075 3079* 3147* 3148 3150 3151 3152 3152 3154 3155 3155* 3157 3161 3162* 3163 3166 3166 3167 3167 3259* 3260 3263 3263* 3264 3267 3290* 3293 3293* 3300 3300* 3303 3303* 3306 3306* 3309 3309* 3313 3313* 3317 3317* 3321 3321* 3324 3348* 3351 3351* 3352 3368* 3371 3374 3379 3382 3385 3385* 3391 3391* 3394 3394* 3395
@ZERO	001	0000	0070	2256 2341 3312
@4K	001	0010	0765	
DKDAC1	001	01C5	2154	
DKDAC2	001	01D9	2167	2134
DKDAC4	001	01E5	2177	2138 2142
DKDAC5	001	0261	2228	2146
DKDAC6	001	028E	2250	2144 2214 2296
DKDAC7	001	02AC	2266	2186 2192 2231 2233
DKDAC8	001	02F7	2340	2136
DKDADR	002	0114	2061	1976
DKDA03	001	0003	2305	2254
DKDBS1	001	01F4	2182	2129 2168* 2176 2237 2241 2249 2265
DKDBS2	001	00D6	1985	1910 1915 1955 1958 2079 2096 2120 2166 2178 2180 2184 2187 2213 2215* 2216 2217* 2227 2251 2253 2254 2254* 2256* 2257 2259 2267 2275 2280 2280* 2282 2282* 2287 2293 2339 2341* 2374 2495
DKDCCE	001	0002	1901	2329
DKDCFP	002	0112	2060	2110
DKDCF1	001	00F5	2016	1932* 2018 2060 2086* 2098* 2101 2101* 2102* 2103* 2106* 2107 2107* 2108* 2112 2127 2178 2179 2187 2194 2196* 2208* 2213 2215* 2232 2234 2238 2242* 2254*
DKDCF2	001	02DE	2306	2179* 2197* 2201 2217
DKDCF3	001	02E2	2308	2201 2205 2219*
DKDCNE	001	0002	1900	2328
DKDCRT	005	02EE	2323	2324
DKDDCE	001	0010	1904	2332
DKDDCT	001	02F6	2332	2295
DKDDDR	001	00FF	2039	1965* 1966* 1969 1969* 1970 1970* 1971 1975 1975* 1976* 2040 2071 2087 2109 2122 2123 2180 2184 2253
DKDDPL	001	00EF	2009	1934 1936* 1944* 1945 1948 1949* 1950 1951* 1956 1965 1967* 1968 1972 2011 2081 2085 2086 2088 2098 2158* 2194 2198 2207* 2217* 2220* 2229* 2230 2234* 2236* 2237* 2240 2257 2267 2280* 2282*
DKDDRT	005	02F1	2320	2293* 2295* 2321
DKDDSV	002	02E8	2311	2187* 2190* 2191
DKDDSW	002	02EC	2313	2237
DKDECL	003	0117	2062	2254

## CROSS REFERENCE

VER 15, MOD 00 25/09/15 PAGE 103

SYMBOL	LEN	VALUE	DEFN	REFERENCES
DKDERP	001	0189	2121	2073
DKDERT	005	02ED	2324	2168*
DKDEST	001	010C	2055	2064
DKDETR	001	005C	1896	2232
DKDFDR	002	02EA	2312	2193* 2200* 2207 2220
DKDISK	001	0025	1909	
DKDKCE	001	0010	1902	2330
DKDKRT	005	02EF	2322	2251* 2323
DKDLSK	001	015B	2099	2216
DKDMPS	001	00FC	2027	
DKDND	001	0341	2383	
DKDNIT	001	02F6	2333	1921
DKDNSF	001	0337	2375	2169
DKDOCT	001	0005	2317	1921 2318
DKDOFT	001	0104	2047	2356
DKDONE	003	010E	2057	1917 1919 1949 2058 2084 2196 2197 2200 2361
DKDRCE	001	0008	1903	2331
DKDRMA	001	00FC	2024	1943* 1955 2025 2027
DKDRMB	001	00FD	2029	1958 2030
DKDRMG	001	00FE	2033	1943 1960* 2034 2102 2106 2112* 2256* 2341*
DKDRRT	005	02F0	2321	2287* 2322
DKDRST	001	0100	2045	2124* 2125* 2133 2135 2137 2139 2143 2145 2352 2376
DKDRTR	003	010E	2058	2168 2251 2287 2293
DKDSAD	002	02E6	2310	2178* 2189* 2191 2205 2219
DKDSAV	001	00EE	2002	1950* 2004 2089
DKDSEE	001	0157	2097	1980 2259 2362
DKDSTA	002	0119	2064	2156
DKDST2	005	010B	2049	1936 1968* 2230 2280
DKDTCT	005	02F1	2318	1921* 2320
DKDTLN	001	0004	1890	2046 2356
DKDWRF	002	0110	2059	1937 1974 2282 2283
DKDXFC	001	00FC	1895	1966
DKDXFF	001	00FF	1893	1932 1945
DKDX03	001	0003	1894	1967
DKDX7F	001	007F	1891	2189 2190 2236
DKDX80	001	0080	1892	2240
DKDZZZ	003	00FB	2020	2127* 2357
DKD010	004	0000	1887	
DKD020	003	004C	1930	
DKD025	001	004F	1931	2365
DKD030	003	004F	1932	
DKD040	004	0064	1942	1918* 1935
DKD050	004	0068	1943	1961*
DKD060	004	009A	1960	1955* 1957 1958* 1961
DKD070	004	00C8	1975	1973
DKD080	004	00D6	1989	1925* 1946
DKD090	004	00DE	1991	2498* 2520*
DKD100	004	00E2	1995	1914*
DKD110	004	00E6	1996	1916*
DKD120	004	00EA	1997	1920* 2497
DKD130	003	011A	2070	1930 2364
DKD140	004	0124	2073	2071*
DKD150	004	0128	2074	2070*
DKD160	003	012C	2080	1981 2363
DKD170	004	0139	2084	1937* 1971* 1974* 2283*
DKD180	003	0150	2090	2087* 2088* 2089* 2275 2351

## CROSS REFERENCE

VER 15, MOD 00 25/09/15 PAGE 104

SYMBOL	LEN	VALUE	DEFN	REFERENCES
DKD190	004	0153	2091	2080* 2082
DKD200	004	0177	2109	2104
DKD210	003	017E	2111	2109* 2355
DKD220	004	0185	2113	2100*
DKD230	003	0191	2124	2122*
DKD240	003	0194	2125	2123*
DKD250	004	0197	2127	
DKD265	001	019B	2128	
DKD270	004	01B5	2142	
DKD280	004	01C5	2155	2289
DKD290	004	01D5	2160	2360 2499* 2521*
DKD295	004	01E1	2170	2159
DKD300	003	01F4	2183	2180* 2181*
DKD310	003	01FE	2186	2184*
DKD320	004	022E	2201	2195 2199
DKD330	003	0246	2213	2204 2206
DKD345	004	0271	2234	2209
DKD350	004	0278	2237	2241
DKD360	003	0288	2242	2239
DKD370	003	029D	2255	2253*
DKD380	004	02C4	2287	2268
DKD390	004	02CB	2289	2252
DKD400	004	02CF	2293	2276
DKD410	004	02FD	2346	2170 2258 2288 2294
DKD412	004	0318	2356	2353
DKD415	003	0320	2359	2348
DKD420	003	032A	2362	1938 2243
DKD430	003	032D	2363	2221
DKD440	004	0333	2365	2342
DKD450	001	033A	2378	2381
I\$ADJX	001	0D5B	1629	
I\$ADST	001	0C9D	1584	
I\$BASE	001	0C60	1586	
I\$BRCN	001	117B	1638	
I\$BSET	001	119D	1637	
I\$B1SW	001	0040	1688	
I\$B2SW	001	0020	1690	
I\$CADR	001	144C	1671	
I\$CALL	001	12B1	1665	
I\$CBM1	001	0D43	1606	
I\$CBN1	001	0D3E	1602	
I\$CBN2	001	0D3F	1603	
I\$CBN3	001	0D40	1604	
I\$CBN4	001	0D41	1605	
I\$CFBS	001	0AE3	1650	
I\$CLFA	001	0D4A	1612	
I\$CLVA	001	0D49	1611	
I\$CL1C	001	0D46	1609	
I\$CL1F	001	0D44	1607	
I\$CL2C	001	0D47	1610	
I\$CL2F	001	0D45	1608	
I\$CPG1	001	1600	1568	
I\$CPUF	001	0A27	1648	
I\$CSCT	001	0D5A	1624	
I\$CSSW	001	0010	1692	
I\$CSXA	001	2000	1567	

## CROSS REFERENCE

VER 15, MOD 00 25/09/15 PAGE 105

SYMBOL	LEN	VALUE	DEFN	REFERENCES
I\$CUPF	001	0A85	1649	
I\$CVAD	001	1358	1659	
I\$DATA	001	0D53	1592	
I\$DAT1	001	0D55	1593	
I\$DMSW	001	0BC1	1646	
I\$ECSW	001	0004	1696	
I\$ERRC	001	0CBC	1591	
I\$FACT	001	0DD1	1631	
I\$FADD	001	075D	1651	
I\$FATE	001	0DE6	1632	
I\$FATP	001	0DE8	1633	
I\$FDVD	001	0919	1654	
I\$FMPY	001	082A	1653	
I\$FSUB	001	0751	1652	
I\$FWRK	001	0607	1575	
I\$IMC1	001	0DCE	1622	
I\$IMLN	001	0DC6	1618	
I\$IMPT	001	0DCC	1621	
I\$INDR	001	0DC5	1617	
I\$INIT	001	0607	1574	
I\$INTR	001	0C5C	1578	
I\$IRSW	001	0CDE	1598	
I\$I700	001	0E24	1656	2992
I\$LBFR	001	12B6	1666	
I\$LDBR	001	1329	1663	
I\$LDXR	001	1330	1664	
I\$LOCK	001	1354	1661	
I\$MDFY	001	1349	1660	
I\$MOD4	001	130B	1657	
I\$NCPG	001	000A	1680	
I\$NDSW	001	0002	1698	
I\$NISW	001	0080	1686	
I\$NPAG	001	0C68	1579	
I\$PARM	001	0D57	1594	
I\$PGDS	001	144A	1669	
I\$PGNO	001	1449	1668	
I\$PGTB	001	14CA	1672	
I\$PLRT	001	15E2	1673	
I\$PSTK	001	15CA	1674	
I\$PUB1	001	0DC8	1619	
I\$PUB2	001	0DCA	1620	
I\$RESW	001	0CE9	1599	
I\$RNMK	001	0001	1615	
I\$RNSW	001	0D5C	1614	
I\$RTRN	001	12D3	1667	
I\$SDCT	001	0D59	1626	
I\$SDPT	001	0DD0	1623	
I\$SFCT	001	0D5A	1627	
I\$SFFO	001	0D5D	1635	
I\$SICT	001	0D5B	1628	
I\$SLLC	001	0BA1	1642	
I\$SLNG	001	0BA2	1641	
I\$SNSW	001	0001	1700	
I\$SSCT	001	0D58	1625	
I\$STAK	001	0D4E	1587	
I\$STCK	001	0B50	1640	

## CROSS REFERENCE

VER 15, MOD 00 25/09/15 PAGE 106

SYMBOL	LEN	VALUE	DEFN	REFERENCES
I\$STHA	001	0D51	1597	
I\$STKB	001	0639	1576	
I\$STKI	001	0D4F	1588	
I\$STSW	001	0008	1694	
I\$TFSW	001	0D2B	1600	
I\$ULNG	001	0C3A	1645	
I\$UNLK	001	1350	1662	
I\$USTK	001	0BB0	1644	
I\$VADR	001	144A	1670	
I\$WRK1	001	0D59	1595	
I\$WRK2	001	0D5B	1596	
I\$XAD1	001	0C89	1583	
I\$XAD2	001	0C82	1582	
I\$XAD3	001	0C7B	1581	
I\$XAD4	001	0C74	1580	
I\$XERR	001	0CAB	1585	
I\$XIAR	001	0D4C	1590	
I\$XPAG	001	0C61	1589	
MOPBF1	001	0C00	3127	3091 3099 3115 3123
MOPBSE	004	0622	3106	3076 3078
MOPCNT	001	0014	3129	3122
MOPDK1	001	0653	3111	3084* 3088
MOPDK2	001	0659	3119	3104
MOPDSK	001	05FF	3107	3082
MOPILA	002	0662	3130	3080
MOPPET	001	0600	3077	1887
MOPREM	001	0000	3108	3074 3082
MOPRSS	002	0660	3128	3081
MOPSF1	001	0001	3126	3084 3085 3086
MOP030	004	0622	3087	3083 3106
MOP035	001	0635	3093	3097
MOP040	006	063B	3099	3092
NAB200	001	0499	2754	2728
NAB300	004	05A8	2990	2755
NAB400	004	05B3	2993	2991
NBLC07	002	05A7	2988	2944
NBLDPL	001	0579	2956	
NBLNOE	003	0569	2946	
NBLOAD	001	0522	2919	2913 2922
NBLRTN	004	0550	2936	
NBLZPL	001	05A0	2981	2972* 2974 3101*
NBLZTR	001	058E	2971	2933
NBL020	004	0538	2926	2923*
NBL050	003	0540	2928	2918* 2930*
NBL060	003	054A	2931	2928 2932
NBL065	003	0568	2945	2916* 2946 2947*
NBL066	004	056E	2948	2925* 2949
NBL067	003	0572	2950	2945
NBL070	001	0579	2955	2929* 2937 2956
NBL080	006	057E	2957	2927* 2931 2944* 2950
NBL100	001	0580	2960	2765
NBRSAV	002	03C5	2578	
NBUFPT	001	03E3	2627	
NCABLD	004	04B4	2764	
NCAERK	001	0469	2714	
NCAERR	001	03CD	2585	

## CROSS REFERENCE

VER 15, MOD 00 25/09/15 PAGE 107

SYMBOL	LEN	VALUE	DEFN	REFERENCES
NCAIPL	004	049D	2756	2994
NCARPL	004	04A1	2758	
NCIENT	001	0483	2736	
NCIEXT	003	0480	2732	
NCIMSK	003	0476	2729	
NCISUS	004	0496	2748	
NCOFNG	001	03DD	2614	
NCRPOS	001	03E2	2624	
NCRTAD	002	044D	2692	
NCRTIN	001	03D3	2594	
NCSDPL	001	050E	2807	
NC0001	002	0464	2697	
NDATE	003	043A	2683	
NDBGUF	001	03E0	2618	
NDFDET	001	03E8	2642	
NDKSIZ	001	03D7	2604	
NDPLSV	006	0449	2688	
NDSADR	002	0587	2964	
NERBF1	001	0A00	3233	3189 3194 3199 3209 3234 3290 3334* 3346 3348 3356 3356* 3357*
NERBF2	001	0A00	3234	3204 3368
NERBSE	004	071A	3219	3137 3142
NERBUF	001	0A00	3235	3147 3156* 3184
NERCDE	001	000C	2494	2497 2523
NERCTD	001	0009	3226	3374*
NERDCD	001	0045	3227	3379* 3382*
NERDEL	001	0002	3220	3293 3300 3303 3306 3309
NERDEV	003	07C1	3261	3140* 3314 3369 3372 3375 3383
NERDPL	001	046F	2718	
NERDSB	001	07AA	3216	3316* 3320* 3329 3341
NERDSD	001	0041	3228	
NERDSL	001	0000	2552	2500
NERDSP	001	07A4	3212	
NERD2D	001	0040	3222	3317
NEREND	001	00FF	3243	3257 3264
NERENL	001	0002	2553	2522
NERENT	002	07A9	3215	3154
NERFIX	001	0008	3223	3318
NERFXD	001	0020	3224	3321
NERHCR	001	0058	3242	3256
NERHD1	001	0040	3237	3246
NERHD2	001	0044	3238	3248
NERHKY	001	0048	3239	3250
NERHLQ	001	0078	3236	3271
NERHLT	001	039E	2526	2531 2533
NERHPR	001	004C	3240	3252
NERHPU	001	0054	3241	3254
NERHRD	002	03A6	2533	2499
NERIDX	002	07AC	3217	3156
NERKYD	001	0001	3225	3371*
NERLCT	001	07A3	3211	3354*
NERLGA	002	03A8	2534	2521
NERLOG	001	0345	2496	
NERMAD	002	0471	2720	
NEROVL	001	03AF	2545	2508 3085*
NEROVR	001	0700	3139	2511 3233 3235
NERPED	001	0010	3221	3313

## CROSS REFERENCE

VER 15, MOD 00 25/09/15 PAGE 108

SYMBOL	LEN	VALUE	DEFN	REFERENCES
NERPHD	001	0011	3229	3385
NERPL2	001	077F	3180	3144 3171* 3173
NERPL3	001	0785	3185	3289 3337
NERPL4	001	078B	3190	3326
NERPL5	001	0791	3195	3329* 3331 3335*
NERPL6	001	0797	3200	3365 3396* 3398
NERPL7	001	079D	3205	3341* 3343 3358* 3360
NERPPE	001	000C	3232	3394
NERSCP	001	0007	3230	3148
NERSDR	001	07DF	3284	3141
NERSHF	002	07A6	3213	3349* 3350* 3352 3357 3386* 3387 3387*
NERSTR	001	07D9	3275	2517
NERTBL	001	07AE	3244	3259
NERTRN	012	039D	2523	2497*
NERTWO	001	07A7	3214	3389
NERVOL	001	07AD	3218	3165
NERWRT	001	03A9	2537	2506
NER050	004	0363	2505	2503
NER100	001	0379	2515	3176
NER200	004	070A	3142	3361 3399
NER220	004	071A	3147	3169 3219
NER225	004	072E	3152	3149
NER230	003	0739	3155	3153
NER240	004	0760	3166	3163* 3164* 3165*
NER250	003	076B	3171	3160
NER260	004	07BC	3259	3175
NER270	003	07C0	3260	3261 3265
NER280	004	07CF	3267	3262
NER290	001	07D3	3268	3267* 3272
NER310	003	080C	3300	3295
NER315	003	0827	3309	3298
NER320	003	082A	3310	3292 3297 3302 3305 3308
NER330	004	0836	3314	3311
NER340	003	0844	3318	3315
NER350	004	0851	3322	3319
NER400	001	088B	3340	3328
NER420	003	08B2	3351	3355
NER450	004	08DC	3361	3338 3347 3353
NER500	001	08E0	3363	3287
NER510	004	08FB	3372	3370
NER520	004	0906	3375	3373
NER525	004	091A	3382	3378
NER530	004	091E	3383	3376
NER533	006	092D	3387	3390
NER540	003	0940	3391	3388 3389*
NER550	004	094C	3395	3393
NER560	004	0950	3396	3380 3384
NEXADR	002	0516	2817	
NEXDPL	001	0514	2815	2804
NEXFTR	001	043B	2684	
NFEMAP	001	0588	2967	2969
NFILIB	002	03DA	2610	
NGUFIO	001	0583	2962	
NHISTE	001	042E	2680	
NHIST1	008	0435	2681	
NINDR1	001	03D4	2595	



## CROSS REFERENCE

VER 15, MOD 00 25/09/15 PAGE 109

SYMBOL	LEN	VALUE	DEFN	REFERENCES
NINDR2	001	03D5	2598	
NINDR3	001	03D6	2601	
NINLNO	002	03CF	2588	
NIOIND	001	03D2	2593	
NKEYBD	001	03E1	2621	
NKEYCD	001	03C3	2572	
NLDRTN	004	0571	2949	
NLEVEL	002	03DF	2617	
NLMRGN	001	03C1	2564	
NLOADB	003	054A	2932	
NLOADR	001	051A	2914	
NLPRIO	002	03EA	2645	
NLPROS	001	03E5	2633	
NLPRP3	001	03E4	2630	
NNEXTB	001	03E6	2636	
NNEXTL	001	03E7	2639	
NNUCBS	001	03C0	2560	
NPAIDF	002	050D	2805	2794
NPASWD	008	042D	2677	
NPAUSE	001	04BA	2773	
NPAUS1	001	04BE	2775	2762
NPA010	004	04BE	2776	
NPA015	003	04CA	2779	
NPA020	004	04E0	2787	2782
NPA030	004	04E6	2789	
NPA040	003	04EC	2791	2780* 2786*
NPA050	004	04EF	2793	2779*
NPA060	004	04F7	2795	2776*
NPA090	004	04FB	2796	2778* 2794 2801
NPA100	004	0506	2803	2791
NPKERT	001	0417	2657	
NPLST1	007	0454	2693	
NPLST2	007	045B	2694	
NPLST3	007	0462	2695	
NPRDEV	002	044B	2689	
NPRPOS	001	03C2	2568	
NPTCH1	011	03F6	2648	
NQUSNS	002	048C	2739	2737*
NQU050	003	0475	2728	2729 2738
NQU100	004	0495	2747	2745* 2748 2774* 2778
NRLOAD	001	051E	2917	
NRMRGN	001	03C0	2561	
NRSTR	004	04D6	2784	
NSPRNT	001	0465	2707	
NSWPCR	003	0511	2809	
NTABLN	004	03CB	2581	
NTROVR	001	054E	2934	
NUNMSK	001	048D	2744	
NUSRDR	002	03DC	2613	
NVOLID	001	03F7	2652	
NWAITF	001	057F	2959	
NWFNME	008	0443	2685	
NXIND1	001	03D0	2591	
NXIND2	001	03D1	2592	
NXIND3	001	03D8	2607	
NXRSAV	002	03C7	2579	

TOTAL STATEMENTS IN ERROR IN THIS ASSEMBLY = 0

OL105 I THE CODE LENGTH OF ##1TRK IS 6144 DECIMAL.

OL103 I TOTAL NUMBER OF LIBRARY SECTORS REQUIRED IS 34  
NAME-##1TRK,PACK-R1R1R1,UNIT-R1,RETAIN-P,LIBRARY-R,CATEGORY-000

START ADDRESS	CATEGORY	NAME AND ENTRY	CODE LENGTH HEXADECIMAL DECIMAL	
0000	0	##1TRK	1800 6144	
OL100 I		THE TOTAL CORE USED BY ##1TRK IS 6144 DECIMAL.		
OL101 I		THE START CONTROL ADDRESS OF THIS MODULE IS 0000.		
OL104 I		TOTAL NUMBER OF LIBRARY SECTORS REQUIRED IS 25		
0		NAME-##1TRK,PACK-R1R1R1,UNIT-R1,RETAIN-P,LIBRARY-O GO GET TO CARD FILE PART 2		
			7628 *	
			7629 *	DATA ITEM IN STACK MUST BE CONVERTED TO CARD OUTPUT
			7630 *	FORMAT AND ITS LENGTH DETERMINED.
			7631 *	
1DA3 35 02 0D4E			7632 SFP220 L	I\$STAK,@XR POINT @XR AT DATA ITEM
1DA7 B8 40 00			7633 TBN	I@STAT(,@XR),B@DTYP CHARACTER CONSTANT ?
1DAA F2 90 30			7634 JF	SFP320 NO, GO CONVERT TO UNPACKED FLT
			7635 *	
			7636 *	CHARACTER CONSTANT TO BE FORMATED
			7637 *	
1DAD BC 7D 00			7638 MVI	I@STAT(,@XR),B@SQUO MOVE A QUOTE MARK TO LEFT END
1DB0 7C 13 EB			7639 MVI	SFPDIC(,@BR),I@LCRF+1 INITLZ CHAR ELEMENT LENGTH + 1
1DB3 E2 02 01			7640 SFP230 LA	@B1(,@XR),@XR INCR @XR BY 1
1DB6 5F 00 EB EC			7641 SLC	SFPDIC(,@BR),SFPONE(1,@BR) DECR COUNTER BY ONE
1DBA F2 81 13			7642 JZ	SFP250 JUMP OUT OF LOOP IF ZERO
1DBD BD 7D 00			7643 CLI	@ZERO(,@XR),B@SQUO IS REF. CHAR. A QUOTE MARK ?
1DC0 D0 01 B3			7644 BNE	SFP230(,@BR) NO, GO INCR TO NEXT CHARACTER
1DC3 AC 10 12 11			7645 MVC	I@LCRF(,@XR),I@LCRF-1(I@LCRF-1,@XR) MOVE CONT TO RIGHT
1DC7 E2 02 01			7646 LA	@B1(,@XR),@XR INCR @XR BY ONE
1DCA BC 7D 00			7647 MVI	@ZERO(,@XR),B@SQUO MOVE IN A MATCHING QUOTE
1DCD D0 87 B3			7648 B	SFP230(,@BR) GO INCR TO NEXT CHARACTER
1DD0 BC 7D 00			7649 SFP250 MVI	@ZERO(,@XR),B@SQUO MOVE IN RIGHT CLOSING QUOTE
			7650 *	
			7651 *	ENTIRE CHARACTER ELEMENT HAS BEEN FORMATTED
			7652 *	
1DD3 34 02 0D59			7653 ST	I\$WRK1,@XR SAVE CHAR-CON RIGHT END ADDR
1DD7 0F 01 0D59 0D4E			7654 SLC	I\$WRK1,I\$STAK(@CADDR) DECR BY START CADDR - LENGTH-1
			7655 *	
			7656 *	ACCESS PART 2 OR PUT TO CARD FILE ROUTINE
			7657 *	
1DDD C2 02 0000			7658 SFP320 LA	*-*,@XR RESTORE D2 ENTRY ROUTINE
	1DE0		7659 SFPCXI EQU	SFP320+@OP1 * FROM POINTER SAVE AREA
			7660 *	
1DE1 C0 87 12B1			7661 B	I\$CALL GO TO CARD FILE ENTRY IN
1DE5 2000	1DE6		7662 DC	AL(@VADDR)(V\$XSPT+3*B@BLSZ) * SFPUTR - PART 4
			7663 *	
			7664 *	CARD OUTPUT COMPLETED - GO TO SFPUTR GENERAL RETURN
			7665 *	
1DE7 D0 87 7E			7666 B	SFP150(,@BR) GO TO UNLOCK D2 RCD
			7667 *	
			7668 *	PART 1 - WORKAREAS, EQUATES AND CONSTANTS
			7669 *	
1DEA	1DEB		7670 SFPDEV DS	CL(@CADDR) SAVE SYS OUTPUT DEVICE STATUS
			7671 *	
	1DEB		7672 SFPDIC EQU	SFPDEV CHAR-CON ELEMENT COUNTER
			7673 *	

SFPUTR - PROLOGUE - VM PUT ROUTINE

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	31/05/21	PAGE 100
	1DEC	01		1DEC	7674	SFPONE	DC XL1'01'			
	1DED	0707		1DEE	7675	SFPMPT	DC AL2(\$\$PRNT)			INDR FOR OUTPUT TO PRINTER
	1DEF	2004		1DF0	7676	SFPCRT	DC AL2(\$\$PLYN)			INDR FOR OUTPUT TO CRT
	1DF1	0100		1DF2	7677	SFPVD2	DC AL2(V\$SFD2)			VADDR OF VM DIRECTORY 2

## SFPUTR - PROLOGUE - VM PUT ROUTINE

ERR LOC OBJECT CODE ADDR STMT SOURCE STATEMENT VER 15, MOD 00 31/05/21 PAGE 101

1E00			7679	ORG	*,B@LVPG,0	PLACE MODULE AT PAGE BOUNDARY
			1E00 7680	SFPBS2 EQU	*	ESTABLISH
			1E00 7681	USING	SFPBS2,@BR	* BASE RESISTER
			7682 *			
			7683 *		THE FILE SPECIFIED IS A DISK FILE.	
			7684 *		@XR POINTS TO THE CURRENT D2 ENTRY.	
			7685 *		THE CONSTANT TO BE PUT IN THE FILE IS IN THE	
			7686 *		STACK IN PACKED FORMAT, IF NUMERIC.	
			7687 *			
1E00	AD 00 04 03		7688	CLC	@\$D2CS(@\$L2VB,@XR),@\$D2BS(@XR)	IS BUFFER FULL ?
1E04	F2 01 06		7689	JNE	SFP350	NO, BYPASS BUFFER WRITE
1E07	D0 87 E0		7690	B	SFP580(@BR)	GO WRITE BUFFER
1E0A	F2 87 0E		7691	J	SFP370	BYPASS BFR ACCESS
1E0D	2C 01 144A 05		7693	SFP350 MVC	I\$VADR,@\$D2CP(@\$L2CP,@XR)	COMBINE CURR POINTER & VM BASE
1E12	2E 00 1449 02		7694	ALC	I\$VADR-1,@\$D2VB(@\$L2VB,@XR)	* PAGE TO SET CURR BFR PAGE
1E17	C0 87 1349		7695	B	I\$MDFY	ACCESS & SET CURR BFR MODIFIED
1E1B	7C 00 F4		7696	SFP370 MVI	SFPWRK(@BR),@ZERO	INITLZ SPACE LEFT COUNTER
1E1E	6F 00 F4 05		7697	SLC	SFPWRK(@BR),@\$D2CB(@\$L2CB,@XR)	CALC SPACE LEFT IN BFR
1E22	6C 00 F6 01		7698	MVC	SFPSIO(@BR),@\$D2IO(@\$L2IO,@XR)	SAVE FILE PREC
1E26	74 02 7C		7699	ST	SFPXR1(@BR),@XR	SAVE POINTER TO D2 ENTRY
1E29	B8 02 01		7700	TBN	@\$D2IO(@XR),@\$M2EF	END OF FILE INDR ON ?
1E2C	F2 90 10		7701	JF	SFP385	NO, GO SET POINTER TO STACK
1E2F	35 02 144C		7702	L	I\$CADR,@XR	SET POINTER TO CURR BFR LOCO.
1E33	BC 1C 00		7703	MVI	@ZERO(@XR),@EOF	MOVE IN AN EOF CODE
1E36	75 02 7C		7704	L	SFPXR1(@BR),@XR	RELOAD D2 ENTRY POINTER
1E39	D0 87 E0		7705	B	SFP580(@BR)	GO WRITE BUFFER
1E3C	D0 87 A9		7706	B	SFP490(@BR)	GO TO RETURN
1E3F	35 02 0D4E		7708	SFP385 L	I\$STAK,@XR	SET POINTER TO STACK
1E43	B8 40 00		7709	TBN	I@STAT(@XR),B@DTYP	CHARACTER CONSTANT ?
1E46	F2 10 2A		7710	JT	SFP430	YES, SO SET LENGTH
1E49	78 20 F6		7711	TBN	SFPSIO(@BR),@\$M2FP	IS FILE PREC = LONG ?
1E4C	F2 10 10		7712	JT	SFP400	YES, GO CHECK RUN PREC.
1E4F	7C 05 F2		7713	MVI	SFPCNL(@BR),I@LPFS	SET CON LNG FOR SHORT PREC.
			7714 *		IF RUN PREC = SHORT, NEXT INSTR IS A JUMP TO RELOAD POINTER	
			7715 *		TO D2 ENTRY. IF RUN PREC = LONG, INSTR IS A NOP	
1E52	F2 87 21		7716	JC	SFP450,I@PRSW	JUMP IF RUN PREC = SHORT
1E55	AC 00 04 08		7717	MVC	I@PEXS(@XR),I@PEXL(@B1,@XR)	SET EXPONENT TO SHORT PREC.
1E59	BB 20 00		7718	SBF	I@STAT(@XR),B@PREC	SET CON INDR TO SHORT PREC.
1E5C	F2 87 17		7719	J	SFP450	GO RELOAD D2 ENTRY POINTER
1E5F	7C 09 F2		7720	SFP400 MVI	SFPCNL(@BR),I@LPFL	SET CON LNG FOR LONG PREC.
1E62	F2 80 11		7721	JC	SFP450,@UCB-I@PRSW+@NOP	IF RUN = LONG, GO RELOAD D2 PTR
1E65	BA 20 00		7722	SFP410 SBN	I@STAT(@XR),B@PREC	SET CON INDR TO LONG PREC
1E68	AC 00 08 04		7723	MVC	I@PEXL(@XR),I@PEXS(@B1,@XR)	SET EXPONENT FOR LONG PREC.
1E6C	AF 03 07 07		7724	SLC	I@PEXL-1(@XR),I@PEXL-1(SFPDLS,@XR)	SET EXTRA DIGITS TO 0
1E70	F2 87 03		7725	J	SFP450	GO RELOAD D2 ENTRY POINTER
1E73	7C 13 F2		7727	SFP430 MVI	SFPCNL(@BR),I@LCRV	SET CON LNG FOR CHAR CONSTANT
1E76	74 02 CA		7728	SFP450 ST	SFPSTK(@BR),@XR	INITLZ CON-MOVE STACK CADDR
1E79	C2 02 0000		7729	SFP460 LA	*-*,@XR	RESTORE CURR D2 ENTRY POINTER
		1E7C	7730	SFPXR1 EQU	SFP460+@OP1	CADDR OF CURRENT D2 ENTRY
1E7D	7D 00 F4		7731	CLI	SFPWRK(@BR),@ZERO	IS SPACE LEFT AT MAXIMUM ?
1E80	F2 81 2A		7732	JE	SFP500	YES, BYPASS REST OF TESTING
1E83	5C 00 F8 F4		7733	MVC	SFPprt(@BR),SFPWRK(1,@BR)	MOVE SPACE LEFT TO MOVE LENGTH
1E87	5F 00 F4 F2		7734	SLC	SFPWRK(@B1,@BR),SFPCNL(@BR)	DECR SPACE LEFT BY CON-LNG

## SFPUTR - PROLOGUE - VM PUT ROUTINE

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00 31/05/21 PAGE 102
	1E8B	F2	84 1F		7735	JH	SFP500	GO MOVE CONSTANT IF SPACE
	1E8E	6C	01 F6 0B		7736	MVC	SFPWK2(@\$L2FS,@BR),@\$D2FS(@XR)	MOVE FILE SIZE TO WORKFLD
	1E92	6C	01 F4 09		7737	MVC	SFPWRK(@BR),@\$D2DD(@\$L2DD,@XR)	DECR BY
	1E96	5F	01 F6 F4		7738	SLC	SFPWK2(@BR),SFPWRK(@\$L2FS,@BR)	* SECTORS USED
	1E9A	6F	00 F6 04		7739	SLC	SFPWK2(@BR),@\$D2CS(@\$L2CS,@XR)	CALC SECTORS LEFT IN
	1E9E	5F	01 F6 FC		7740	SLC	SFPWK2(@\$L2FS,@BR),SFPC01(@BR)	* FILE
	1EA2	F2	84 0C		7741	JH	SFP510	MORE, GO SET UP PART-CON MOVE
	1EA5	3C	BA 0CBC		7742	SFP480 MVI	I\$ERRC,@E715	SET EOF ERROR CODE
	1EA9	C0	87 12D3		7743	SFP490 B	I\$RTRN	RETURN
	1EAD	5C	00 F8 F2		7745	SFP500 MVC	SFPprt(@BR),SFPCNL(1,@BR)	SET LENGTH OF CON-MOVE
	1EB1	4C	01 C8 144C		7746	SFP510 MVC	SFPBFR(@CADDR,@BR),I\$CADR	INITLZ BUFFER CADDR FOR MOVE
	1EB6	7C	FF C6		7747	MVI	SFPMVL(@BR),SFPMs1	CALC LENGTH
	1EB9	5E	00 C6 F8		7748	ALC	SFPMVL(@B1,@BR),SFPprt(@BR)	* OF MOVE
	1EBD	5E	00 CA C6		7749	ALC	SFPSTK(@B1,@BR),SFPMVL(@BR)	INCR STACK BUFFER CADDR'S
	1EC1	5E	00 C8 C6		7750	ALC	SFPBFR(@B1,@BR),SFPMVL(@BR)	* BY LENGTH OF MOVE
	1EC5	0C	00 0000 0000		7751	SFP550 MVC	*-*(@VQ),*-*	MOVE CONSTANT FROM STACK TO BFR
				1EC8	7752	SFPBFR EQU	SFP550+@OP1	CADDQ OF BUFFER LOCATION
				1EC6	7753	SFPMVL EQU	SFP550+@Q	LENGTH OF MOVE
				1ECA	7754	SFPSTK EQU	SFP550+@OP2	CADDR OF STACK LOCATION
	1ECB	9E	01 05 F8		7755	ALC	@\$D2CP(@\$L2CP,@XR),SFPprt(@BR)	INCR D2 ENTRY BFR POINTER
	1ECF	5F	00 F2 F8		7756	SLC	SFPCNL(@B1,@BR),SFPprt(@BR)	DECR CON-LNG BY LNG MOVED
	1ED3	D0	81 A9		7757	BZ	SFP490(@BR)	IF ALL MOVED, GO TO RETURN
	1ED6	5E	00 CA FC		7758	SFP560 ALC	SFPSTK(@B1,@BR),SFPC01(@BR)	INCR STACK CADDR FOR NXT MVE
	1EDA	D0	87 E0		7759	B	SFP580(@BR)	
	1EDD	D0	87 AD		7760	B	SFP500(@BR)	
					7761	*		
					7762	*	VM BUFFER MUST BE WRITTEN TO THE SAVED FILE	
					7763	*		
	1EE0	74	08 F1		7764	SFP580 ST	SFP590+@OP1(@BR),@ARR	SAVE RETURN
	1EE3	C0	87 12B1		7765	B	I\$CALL	EXECUTE PART 3
	1EE7	1E02		1EE8	7766	DC	AL(@VADDR)(V\$XSPT+2+B@BLSZ)	* OF SFPUTR
	1EE9	1C	01 144C FA		7767	MVC	I\$CADR,SFPSCA(@CADDR,@BR)	
	1EEE	C0	87 0000		7768	SFP590 B	*-*	RETURN TO CALLING LOCATION
					7769	*		
					7770	*	PART 2 - WORK AREAS AND CONSTANTS	
					7771	*		
	1EF2			1EF2	7772	SFPCNL DS	CL1	SAVE AREA FOR CONSTANT LENGTH
	1EF2				7773	ORG	SFPCNL	* INITIALIZE TO
	1EF2	00		1EF2	7774	DC	XL1'0'	* ZERO
	1EF3	00		1EF3	7775	DC	XL1'00'	INITLZ TO '0' TO USE WRK AS 2
	1EF4			1EF4	7776	SFPWRK DS	CL1	WORKAREA TO CALC SPACE LEFT
	1EF5			1EF6	7777	SFPWK2 DS	CL2	WORKAREA TO CHECK FOR EOF
	1EF7			1EF8	7778	SFPprt DS	CL2	WORKAREA TO FOR MOVE LENGTH
	1EF7				7779	ORG	SFPprt-1	* INITLZ TO ZERO, 1ST BYTE WILL
	1EF7	0000		1EF8	7780	DC	XL2'0'	* ALWAYS BE 0.
	1EF9			1EFA	7781	SFPSCA DS	CL2	SAVE AREA FOR VM BUFFER CADDR
	1EFB	0001		1EFC	7782	SFPC01 DC	IL2'1'	ONE CONSTANT
					7783	*		
					7784	*	PART 2 - EQUATES	
					7785	*		
				1EF6	7786	SFPsIO EQU	SFPWK2	SAVE AREA FOR FILE PRECISION
				0004	7787	SFPDLS EQU	I@LPFL-I@LPFS	LNG LONG PREC, EXTRA SIGNIFICNCE
				00FF	7788	SFPMS1 EQU	X'FF'	MINUS 1



## SFPUTR - PROLOGUE - VM PUT ROUTINE

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00 31/05/21 PAGE 103
	1F00				7790	ORG	*,B@LVP,0	PLACE MODULE_AT PAGE BOUNDARY
				1F00	7791	SFPBS3 EQU	*	* FOR THIS PAGE
				1F00	7792	USING	SFPBS3,@BR	ESTABLISH BASE REGISTER USAGE
	1F00	1C 01	144A 8B		7793	MVC	I\$VADR,SFPRT2(@CADDR,@BR)	REFERENCE PAGE 2 OF
	1F05	C0 87	1358		7794	B	I\$CVAD	* THE SFPUTR ROUTINE
	1F09	4C 00	76 144B		7795	MVC	SFP680+@OP1-1(1,@BR),I\$CADDR-1	SET UP RTRN CADDR FOR PG CA
	1F0E	AD 00	04 03		7796	CLC	@\$D2CS(@\$L2VB,@XR),@\$D2BS(@XR)	IS VM BUFFER FULL ?
	1F12	F2 81	09		7797	JE	SFP610	YES, GO SET UP FOR SFLOAD
	1F15	B8 02	01		7798	TBN	@\$D2IO(@XR),@\$M2EF	EOF INDR SET ?
	1F18	F2 10	03		7799	JT	SFP610	YES, GO SET UP FOR SFLOAD
					7800	*		
					7801	*		
					7802	*	CURRENT BUFFER PAGE IN CORE IS FILLED, BUT VM BUFFER	
					7803	*	NOT FULL. GET NEXT PAGE OF BUFFER INTO CORE.	
	1F1B	F2 87	48		7804	J	SFP675	
					7805	*		
					7806	*	VM BUFFER MUST BE PUT TO SAVED FILE. SET UP & EXEC SFLOAD	
					7807	*		
	1F1E	4C 00	2C 144B		7808	SFP610 MVC	SFP625+@OP2-1(1,@BR),I\$CADDR-1	SET UP CADDR FOR CON-LNG
	1F23	4C 00	32 144B		7809	MVC	SFP630+@OP2-1(1,@BR),I\$CADDR-1	SET UP CADDR FOF CON-CADDR
	1F28	0C 00	0D59 1EF2		7810	SFP625 MVC	I\$WRK1,SFPCNL(1)	MOVE CON-LNG FOR SFLOAD
	1F2E	0C 01	0D5B 1ECA		7811	SFP630 MVC	I\$WRK2,SFPSTK(@CADDR)	MOVE CADDR OF CON FOR SFLOAD
	1F34	3C 02	0D58		7812	MVI	I\$WRK1-1,@DPUT	SET OUTPUT INDR FOR SFLOAD
	1F38	74 01	43		7813	ST	SFP635+SFP5(@BR),@BR	SET UP CADDR OF DPL TO
	1F3B	7C 7E	43		7814	MVI	SFP635+SFP5(@BR),SFPD1D	* WRITE OUT INTERPRETER
					7815	*SFP635 DISK	@ZERO	WRITE IT OUT
	1F3E	C0 87	0025		7816	SFP635 B	\$DISKN	PERFORM PHYSICAL DISK OP
	1F42	0000		1F43	7817	DC	AL2(@ZERO)	DPL ADDRESS
					7818	***	END OF EXPANSION ***	
	1F44	74 01	55		7820	ST	SFP640+SFP5(@BR),@BR	SET UP CADDR OF DPL TO
	1F47	7C 84	55		7821	MVI	SFP640+SFP5(@BR),SFPD2D	* READ & EXECUTE SFLOAD
	1F4A	74 01	5B		7822	ST	SFP650+@OP1(@BR),@BR	SAVE BASE REGISTER
	1F4D	74 02	5F		7823	ST	SFP655+@OP1(@BR),@XR	SAVE POINTER TO D2 ENTRY
					7824	*SFP640 BLOAD	@ZERO	GO EXECUTE SFLOAD
	1F50	C0 87	0522		7825	SFP640 B	\$BLOAD	LOAD AND EXECUTE WORK AREA PGM
	1F54	0000		1F55	7826	DC	AL2(@ZERO)	DPL ADDRESS
					7827	***	END OF EXPANSION ***	
					7829	*		
					7830	*	RETURN FROM SFLOAD	
					7831	*		
	1F56	0444		1F57	7832	DC	AL(@CADDR)(\$DPLSV-5)	CADDR OF INTERPRETER READ DPL
	1F58	C2 01	0000		7833	SFP650 LA	*-*,@BR	RESTORE BASE
	1F5C	C2 02	0000		7834	SFP655 LA	*-*,@XR	RESTORE D2 POINTER
					7835	*	DISK \$WAITF	WAIT FOR INTERPRETER
	1F60	C0 87	0025		7836	B	\$DISKN	PERFORM PHYSICAL DISP OP
	1F64	057F		1F65	7837	DC	AL2(\$WAITF)	DPL ADDRESS
					7838	***	END OF EXPANSION ***	
	1F66	2C 01	144A 05		7840	SFP675 MVC	I\$VADR,\$D2CP(@\$L2CP,@XR)	SET UP VADDR OF NEXT PAGE
	1F6B	2E 00	1449 02		7841	ALC	I\$VADR-1,\$D2VB(@\$L2VB,@XR)	* OF VM BUFFER
	1F70	C0 87	1349		7842	B	I\$MDFY	ACCESS & SET FOR MODIFICATION
					7843	*		
					7844	*	THE NEXT INST HAS BEEN MODIFIED TO UP THE CADDR OF THE	
					7845	*	BUFFER INTO A SAVE AREA IN PART 2 OF SFPUTR	



## SFPUTR - PROLOGUE - VM PUT ROUTINE

ERR LOC OBJECT CODE ADDR STMT SOURCE STATEMENT VER 15, MOD 00 31/05/21 PAGE 104

```

      7846 *
1F74 0C 01 1EFA 144C 7847 SFP680 MVC SFPSCA,I$CADR(@CADDR)
1F7A C0 87 12D3 7848 B I$RTRN RETURN TO SFPUTR - PART 2
      7849 *
      7850 * PART 3 - DPL'S
      7851 *
      7852 *SFPD1 DPL FUNC-@DPUT,DADDR-#@VSFI,CNT-#@@VSL,CADDR-#$$INS
1F7E 7853 SFPD1 EQU * DISK PARAMETER LIST
1F7E 02 1F7E 7854 DC AL1(@DPUT) REQUESTED FUNCTION
1F7F 09A1 1F80 7855 DC AL2(@VSFI) DISK ADDRESS
1F81 0F 1F81 7856 DC AL1(@@VSL) SECTOR COUNT
1F82 0600 1F83 7857 DC AL2($$INS) BUFFER ADDRESS
      7858 *** END OF EXPANSION ***

007E 7860 SFPD1D EQU SFPD1-SFPBS3 DISP TO INTRP WRITE DPL
      7861 *
      7862 *SFPD2 DPL FUNC-@DGET,DADDR-#@SFLO,CNT-#@@SFL,CADDR-#$$SFL
1F84 7863 SFPD2 EQU * DISK PARAMRMI RST
1F84 01 1F84 7864 DC AL1(@DGET) REQUESTED FUNCTION
1F85 0499 1F86 7865 DC AL2(@SFLO) DISK ADDRESS
1F87 05 1F87 7866 DC AL1(@@SFL) SECTOR COUNT
1F88 0F00 1F89 7867 DC AL2($$SFL) BUFFER ADDRESS
      7868 *** END OF EXPANSION ***

0084 7870 SFPD2D EQU SFPD2-SFPBS3 DISP TO SFLOAD READ DPL
      7871 *
      7872 * PART 3 - CONSTANTS, SAVE AREAS & EQUATES
      7873 *
1F8A 1E00 1F8B 7874 SFPRT2 DC AL(@VADDR)(V$XSPT+B@LVPG) VADDR OF SFPUTR PART 2
      7875 *
0005 7876 SFP5 EQU 5 DISP TO DPL-ADDR
0F00 7877 SFPSAO EQU $$KLD1+X'0900' CORE LOAD ADDR OF #SFLOA

```

## SFPUTR - PROLOGUE - VM PUT ROUTINE

ERR LOC OBJECT CODE ADDR STMT SOURCE STATEMENT VER 15, MOD 00 31/05/21 PAGE 105

2000				7879	ORG	*,B@LVPG,0	PLACE MODULE AT PAGE BOUNDARY
			2000	7880	SFPBS4 EQU	*	* REGISTER USAGE
			2000	7881	USING	SFPBS4,@BR	ESTABLISH BASE
				7882	*		
				7883	*	ENTRY TO PART 2 OR PT TO CARD FILE ROUTINE	
				7884	*		
2000	74	02	AD	7885	ST	SFP830+@OP1(,@BR),@XR	SAVE D2 POINTER
2003	6C	01	C0 05	7886	MVC	SFPVCA(,@BR),@\$D2CP(@\$L2CP,@XR)	SAVE CURRENT BFR POINTER
2007	6C	00	94 02	7887	MVC	SFPCBP(,@BR),@\$D2VB(@\$L2VB,@XR)	SAVE BFR BASE PAGE NUM.
				7888	*		
200B	B8	02	01	7889	TBN	@\$D2IO(,@XR),@\$M2EF	END OF FILE INDR ON ?
200E	F2	90	7F	7890	JF	SFP800	YES, GO CLEAR & FLUSH BUFFER
				7891	*		
2011	35	02	0D4E	7892	L	I\$STAK,@XR	POINT @XR AT STACKED CONSTANT
2015	74	02	8E	7893	ST	SFPSTC(,@BR),@XR	SAVE STACK POINTER
2018	4C	00	DE 0D59	7894	MVC	SFPCFL(@\$L2CB,@BR),I\$WRK1	SET CHAR-CON LENGTH-1
201D	BD	7D	00	7895	CLI	I@STAT(,@XR),B@SQUO	IS IT A CHAR-CON ?
2020	F2	81	55	7896	JE	SFP785	YES, GO CHECK FOR FIT
				7897	*		
				7898	*	DATA ITEM IS NUMERIC - CONVERT IT TO OUTPUT FORMAT	
				7899	*		
2023	C0	87	0A27	7900	B	I\$CPUF	UNPACK THE NUMBER
2027	AC	06	08 07	7901	MVC	I@PREC+1(,@XR),I@PREC(I@PREC,@XR)	SHIFT MANTISSA RIGHT 1
202B	BC	4B	01	7902	MVI	I@STAT+1(,@XR),B@DPNT	INSERT DECIMAL POINT
				7903	*		
				7904	*	CONVERT EXPONENT TO OUTPUT FORMAT	
				7905	*		
202E	9C	03	0C FA	7906	MVC	I@PREC+5(,@XR),SFPEXI(SFPLEX,@BR)	MOVE IN EXPONENT IMAGE
2032	9F	00	00 FB	7907	SLC	I@STAT(,@XR),SFPEZR(1,@BR)	DETERMINE EXP. MAGNITUDE
2036	F2	81	2D	7908	JZ	SFP750	BYPASS CONVERSION TO DEC IF 0
2039	F2	84	0E	7909	JH	SFP720	BYPASS RE-COMPLEMENT IF POSITIVE
				7910	*		
				7911	*	NEGATIVE EXPONENT - MODIFY SIGN AND RECOMPUTE BINARY EXPONENT	
				7912	*		
203C	7C	00	FD	7913	MVI	SFPNGE(,@BR),@ZERO	DETERMINE BINARY MAGNITUDE OF
203F	6F	00	FD 00	7914	SLC	SFPNGE(1,@BR),I@STAT(,@XR)	* NEGATIVE EXPONENT
2043	9C	00	00 FD	7915	MVC	I@STAT(,@XR),SFPNGE(1,@BR)	PUT ABS VALUE IN STACK EXP POS
2047	BC	60	0A	7916	MVI	I@PREC+3(,@XR),B@MINS	MAKE EXPONENT SIGN NEGATIVE
				7917	*		
				7918	*	CONVERT BINARY EXPONENT MAGNITUDE TO ZONED DECIMAL	
				7919	*		
204A	54	10	FD F6	7920	SFP720 ZAZ	SFPDAC(SFPLXM,@BR),SFPZD1(1,@BR)	SET ACCUMULATOR - 1
204E	7C	01	52	7921	MVI	SFP725+@Q(,@BR),@B1	SET BINARY MASK FOR 2**6 BIT
2051	B8	01	00	7922	SFP725 TBN	I@STAT(,@XR),@VQ	TEST BINARY EXP MAGNITUDE BIT
2054	F2	90	04	7923	JF	SFP730	BYPASS DEC. EXP. INCR IF ZERO
2057	96	01	0C FD	7924	AZ	I@PREC+5(SFPLXM,@XR),SFPDAC(SFPLXM,@BR)	INCR DECIMAL EXP
205B	5E	00	52 52	7925	SFP730 ALC	SFP725+@Q(,@BR),SFP725+@Q(1,@BR)	SHIFT BINARY MASK LEFT
205F	56	01	FD FD	7926	AZ	SFPDAC(SFPLXM,@BR),SFPDAC(SFPLXM,@BR)	DOUBLE DEC ACCUM
2063	D0	08	51	7927	BNOZ	SFP725(,@BR)	REPEAT LOOP UNTIL ACCUM > 64
				7928	*		
				7929	*	DETERMINE AND SET SIGN OF MANTISSA	
				7930	*		
2066	BC	40	00	7931	SFP750 MVI	I@STAT(,@XR),B@BLNK	INITLZ SIGN TO POSITIVE
2069	B8	F0	08	7932	TBN	I@PREC+1(,@XR),B@ZPOS	IS MANTISSA POSITIVE ?
206C	F2	10	06	7933	JT	SFP760	YES, BYPASS NEGATIVE HANDLING
206F	BC	60	00	7934	MVI	I@STAT(,@XR),B@MINS	SET SIGN TO NEGATIVE

## SFPUTR - PROLOGUE - VM PUT ROUTINE

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	31/05/21	PAGE 106
	2072	BA	F0 08		7935	SBN	I@PREC+1(,@XR),B@ZPOS			CLEAR MANTISSA SIGN INDR.
					7936	*				
					7937	*	SET LENGTH OF NUMERIC VALUE IN STACK			
					7938	*				
	2075	7C	0C DE		7939	SFP760 MVI	SFPCFL(,@BR),I@PREC+SFPENC			SET EXTERNAL LENGTH OF NUM-CON
					7940	*				
					7941	*	DETERMNE IF THIS DATA ITEM WILL FIT ON CURRENT CARD			
					7942	*				
	2078	5C	00 FD C0		7943	SFP785 MVC	SFPCPW(,@BR),SFPCPT(1,@BR)			MOVE CURR PT TO WORK AREA
	207C	5E	00 FD DE		7944	ALC	SFPCPW(@\$L2CB,@BR),SFPCFL(,@BR)			CALC NEXT BFR SPACE
	2080	38	80 03DD		7945	TBN	\$CONFIG,\$BIGCD			IS IBM 129 CONFIGURED ? 1-4
	2084	F2	90 03		7946	JF	SFP790			JUMP IF NOT 1-4
	2087	7C	4E 8B		7947	MVI	SFP790+1(,@BR),@BCRDL-2			CHANGE LENGTH TO BIG CARD 1-4
	208A	7D	00 FD		7948	SFP790 CLI	SFPCPW(,@BR),*-*			WILL ADJ LNG FIT ON CARD ? 1-4
	208B				7949	ORG	SFP790+@Q			POINT TO INITLZ LENGTH 1-4
	208B	5E		208B	7950	DC	AL1(@CARDL-2)			INITIALIZE FOR 96 COL CARD 1-4
	208D				7951	ORG	SFP790+@INST3			RESTORE INSTRUCTION ADDR REG 1-4
	208D	F2	04 27		7952	JNH	SFP850			YES, GO PUT IT IN BUFFER
					7953	*				
					7954	*	CARD SUFFER CAN NOT CONTAIN THE ENTIRE NEXT CONSTANT OR			
					7955	*	END OF FILE HAS OCCURRED.			
					7956	*				
					7957	*	CLEAR THE UNUSED PORTION OF THE BUFFER			
					7958	*				
	2090	C0	87 1330		7959	SFP800 B	I\$LDXR			POINT @XR TO 1ST BYTE
	2094			2095	7960	SFPCBV DS	CL(@VADDR)			VADDR OF CARD BUFFER PAGE
				2094	7961	SFPCBP EQU	SFPCBV-1			PAGE ADDRESS
	2095				7962	ORG	SFPCBV			INITLZ THE DISPLACEMENT TO
	2095	00		2095	7963	DC	XL(@\$L2CB)'00'			* THE FIRST BYTE OF THE BUFFER
	2096	76	02 C0		7964	A	SFPCPT(,@BR),@XR			INCR @XR TO FIRST UNUSED BYTE
	2099	BC	40 60		7965	MVI	@CARDL(,@XR),B@BLNK			BLANK OUT THE UNUSED PORTION
	209C	AC	5F 5F 60		7966	MVC	@CARDL-1(,@XR),@CARDL(@CARDL,@XR)			* OF THE CARD BUFFER
					7967	*				
					7968	*	CALL THE I/O ROUTINE TO PUNCH THE CARD			
					7969	*				
	20A0	35	02 144C		7970	L	I\$CADR,@XR			POINT @XR TO BUFFER
	20A4	C0	87 12B1		7971	B	I\$CALL			EXECUTE THE CARD
	20A8	2A96		20A9	7972	DC	AL(@VADDR)(V\$SCDO)			* PUT ROUTINE
					7973	*				
					7974	*	POINT @XR BACK TO D2 ENTRY AND CHECK FOR END OF FILE			
					7975	*				
	20AA	C2	02 0000		7976	SFP830 LA	*-*,@XR			POINT @XR TO D2 ENTRY
	20AE	B8	02 01		7977	TBN	@\$D2IO(,@XR),@\$M2EF			END OF FILE INDR ON ?
	20B1	D0	10 F1		7978	BT	SFP950(,@BR)			YES, GO TO RETURN
	20B4	7C	00 C0		7979	MVI	SFPCPT(,@BR),@ZERO			MOVE A ZERO TO CURR BYTE PT
					7980	*				
					7981	*	PUT DATA ITEM IN CARD BUFFER			
					7982	*				
	20B7	5E	00 C0 94		7983	SFP850 ALC	SFPVCA(,@BR),SFPCBP(@\$L2VB,@BR)			INCR BFR DISP BY VM BASE
	20BB	C0	87 1330		7984	B	I\$LDXR			POINT @XR AT
	20BF			20C0	7985	SFPVCA DS	CL(@VADDR)			* 1ST UNUSED BUFFER POSITION
				20C0	7986	SFPCPT EQU	SFPVCA			DISP. TO 1ST UNUSED SPACE
	20C1	C0	87 1349		7987	B	I\$MDFY			SET BUFFER MODIFIED 1-4
	20C5	7D	00 C0		7988	CLI	SFPCPT(,@BR),@ZERO			1ST DATA ITEM ON CARD
	20C8	F2	81 0A		7989	JE	SFP865			YES, BYPASS SEPERATOR PLACEMENT
					7990	*				

## SFPUTR - PROLOGUE - VM PUT ROUTINE

ERR LOC OBJECT CODE ADDR STMT SOURCE STATEMENT VER 15, MOD 00 31/05/21 PAGE 107

```

7991 * PLACE A SEPERATOR BEFORE THE CURRENT DATA ITEM
7992 *
20CB BC 6B 00 7993 MVI @ZERO(,@XR),B@CMMA PUT SEPORATOR IN CARD BUFFER
20CE E2 02 01 7994 LA @B1(,@XR),@XR INCR @XR TO NEXT UNUSED BYTE
20D1 5E 00 C0 F5 7995 ALC SFPCPT(,@BR),SFPX01(1,@BR) INCR CARD BUFR PT BY 1
7996 *
7997 * PLACE THE DATA ITEM IN THE CARD BUFFER
7998 *
20D5 5E 00 8E DE 7999 SFP865 ALC SFPSTC(1,@BR),SFPCFL(,@BR) * OF DATA ITEM
20D9 5C 00 DF DE 8000 MVC SFPDCA(,@BR),SFPCFL(1,@BR) SET BUFFER DISP TO CON LEFTEND
20DD 8C 00 00 0000 8001 SFP875 MVC 0(@VQ,@XR),*-* MOVE DATA ITEM TO BUFFER
20DF 8002 SFPDCA EQU SFP875+@D1 DISP FROM @XR TO LH BFR ADDR
20DE 8003 SFPCFL EQU SFP875+@Q EXTEWAL FORMAT LENGTH MINUS 1
208E 8004 SFPSTC EQU SFP790+@DOP2 LEFT HAND STACK CADDR OF ITEM
8005 *
8006 * UPDATE D2 CURRENT ENTRY BUFFER POINTER AND RETURN
8007 *
20E2 75 02 AD 8008 L SFP830+@OP1(,@BR),@XR RESTORE D22 ENTRY POINTER
20E5 9C 00 05 C0 8009 MVC @$D2CB(@$L2CB,@XR),SFPCPT(,@BR) SET IN STARTING DISP.
20E9 9E 00 05 DE 8010 ALC @$D2CB(@$L2CB,@XR),SFPCFL(,@BR) INCR TO ENDING DISP.
20ED 9E 00 05 F5 8011 ALC @$D2CB(@$L2CB,@XR),SFPX01(,@BR) INCR TO NEXT UNUSED SPACE
20F1 C0 87 12D3 8012 SFP950 B I$RTRN RETURN
8013 *
8014 * PART 4 - EQUATES, WORKAREAS AND CONSTANTS
8015 *
0004 8016 SFPLEX EQU 4 LENGTH OF EXPONENT IMAGE
0002 8017 SFPLXM EQU 2 LENGTH OF ZONED DEC. EXPONENT
0005 8018 SFPENC EQU SFPLEX+2-1 LENGTH-1 OF EXTRA NON-PREC CHAR
8019 *
20F5 01 20F5 8020 SFPX01 DC XL1'1' BINARY 1
20F6 F1 20F6 8021 SFPZD1 DC DL1'1' ZONED DECIMAL 1
20F7 C54EF0F0 20FA 8022 SFPEXI DC CL(SFPLEX)'E+00' EXPONENT IMAGE, INITLZ ZERO
20FB 80 20FB 8023 SFPEZR DC AL1(B@NXZR) ZERO NORMALIZED EXPONENT
8024 *
20FC 20FD 8025 SFPDAC DS CL(@CADDR) DECIMAL ACCUMULATOR WORK AREA
20FD 8026 SFPNGE EQU SFPDAC NEGATIVE EXPONENT WORK AREA
20FD 8027 SFPCPW EQU SFPNGE BUFFER OVERFLOW CALC WORK AREA
8028 *
8029 * END OF SFPUTR ROUTINE
8030 *

```



## SFGETR - PROLOGUE - VM GET ROUTINE

ERR	LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00	31/05/21	PAGE 109
			8033		*****			
			8034	*	5703-XM1 COPYRIGHT IBM CORP. 1970			*
			8035	*	REFER TO INSTRUCTIONS ON COPY RIGHT NOTICE, 120-2083			*
			8036	*				*
			8037		*****			
			8038	*	STATUS			*
			8039	*	VERSION 1 MODIFICATION 0			*
			8040	*				*
			8041	*	FUNCTION			*
			8042	*	* SFGETR IS CALLED TO INPUT THE NEXT SEQUENTIAL DATA ELEMENT FROM			*
			8043	*	AN EXTERNAL DATA FILE. THIS DATA FILE MAY BE ON DISK OR CARD.			*
			8044	*	* THE NEXT SEQUENTIAL DATA ELEMENT, ARITHMETIC OR CHARACTER, IS			*
			8045	*	ACCESSED AND PLACED IN THE RUN-TIME STACK AREA.			*
			8046	*	* IF INPUT IS FROM THE CARD READER, THE DATA ELEMENT MUST BE CON-			*
			8047	*	VERTED TO INTERNAL NOTATION BEFORE IT IS PASSED			*
			8048	*	* WHEN ALL DATA ELEMENTS IN THE BUFFERS ALLOCATED TO THE FILE ARE			*
			8049	*	DEPLETED, A CALL IS MADE TO THE APPROPRIATE ROUTINE TO REFILL			*
			8050	*	THE BUFFERS IN VIRTUAL MEMORY.			*
			8051	*				*
			8052	*	ENTRY POINTS			*
			8053	*	* THE ENTRY IS SFGETR. THE FORMAT OF THE CALLING SEQUENCE IS AS			*
			8054	*	FOLLOWS:			*
			8055	*	B I\$CALL			*
			8056	*	DC AL2(V\$XSGT)			*
			8057	*				*
			8058	*	INPUT:			*
			8059	*	* THE DISPLACEMENT TO THE CURRENT FILE ENTRY IS IN THE D2 HEADER.			*
			8060	*	* THE ADDRESS OR THE LOCATION TO PLACE THE NEXT CONSTANT FROM THE			*
			8061	*	FILE IS IN I\$STAK.			*
			8062	*				*
			8063	*	OUTPUT			*
			8064	*	* THE CONSTANTS ARE MOVED INTO THE STACK REFERENCED IN THE ADDR			*
			8065	*	AT I\$STAK			*
			8066	*				*
			8067	*	EXTERNAL REFERENCES			*
			8068	*	I\$MDFY - PAGING ENTRY TO SET BASE PAGE WRITE BACH INDICATOR			*
			8069	*	I\$LOCK - PAGING ENTRY TO LOCK PAGE IN CORE			*
			8070	*	I\$CVAD - PAGING ENTRY TO ACCESS PAGE			*
			8071	*	I\$CALL - PAGING ENTRY TO CALL ANOTHER ROUTINE			*
			8072	*	I\$UNLK - PAGING ENTRY TO UNLOCK PAGE			*
			8073	*	I\$RTRN - PAGING ENTRY TO RETURN TO USER			*
			8074	*	DKDISK - DISK IOCR			*
			8075	*	SFLOAD - BUFFER TRANSFER ROUTINE			*
			8076	*	DFRDIN - CARD READ ROUTINE			*
			8077	*	I\$STAK - LOCATION OF ADDRESS OF STACK TO PLACE CONSTANT			*
			8078	*	I\$ERRC - ERROR CODE SAVE AREA			*
			8079	*				*
			8080	*	EXITS, NORMAL			*
			8081	*	THE CONSTANTS ARE PLACED IN THE RUN-TIME STACK, AND CONTROL			*
			8082	*	RETURNED VIA I\$RTRN.			*
			8083	*				*
			8084	*	EXITS, ERROR			*
			8085	*	IN THE EVENT OF AN ERROR, AN ERROR CODE IS PLACED IN I\$ERRC, AND			*
			8086	*	CONTROL RETURNED TO THE USER VIA I\$RTRN.			*
			8087	*				*
			8088	*	TABLES/WORK AREAS			*



## SFGETR - PROLOGUE - VM GET ROUTINE

ERR LOC OBJECT CODE ADDR STMT SOURCE STATEMENT VER 15, MOD 00 31/05/21 PAGE 110

```

8089 * D2 - VM DIRECTORY 2 - CURRENT USAGE I/O INFORMATION *
8090 * *
8091 *ATTRIBUTES *
8092 * REUSABLE, RELOCATABLE *
8093 * *
8094 *CHARACTER CODE DEPENDENCY *
8095 * THE OPERATION OF THIS MODULE DEPENDS UPON AN INTERNAL REPRESENTA- *
8096 * TION OF THE EXTERNAL CHARACTER SET WHICH IS EQUIVALENT TO THE ONE *
8097 * USED AT ASSEMBLY TIME. *
8098 * *
8099 *NOTES *
8100 * ERROR PROCEDURES *
8101 * AN ERROR CODE IS PLACED IN I$ERRC, AND CONTROL RETURNED TO THE *
8102 * USER *
8103 * *
8104 * REGISTER USAGE *
8105 * REGISTER 1 (@BR) IS USED AS THE BASE REGISTER *
8106 * REGISTER 2 (@XR) IS USED TO REFERENCE THE CURRENT FILE ENTRY *
8107 * AND D2, AND TO REFERENCE THE RUN-TIME STACK. *
8108 * *
8109 * SAVED/RESTORED AREAS *
8110 * NONE. *
8111 * *
8112 * MODIFICATION CONSIDERATIONS *
8113 * NONE. *
8114 * *
8115 * REQUIRED NODULES *
8116 * @SYSEQ - SYSTEM EQUATES *
8117 * @FXDEQ - FIXED ADDRESSES FOR SYSTEM NUCLEUS *
8118 * @CANEQ - SYSTEM LOCATION EQUATES *
8119 * @WKAEQ - SYSTEM WORKAREA DADDR EQUATES *
8120 * @CYOEQ - CYLINDER ZERO EQUATES *
8121 * @VMDEQ - VM DIRECTORY EQUATES *
8122 * @ERMEQ - ERROR MESSAGE EQUATES *
8123 * @SPFEQ - SYSTEM PROGRAM FILE EQUATES *
8124 * $V$EQU - FIXED VIRTUAL ADDRESSES *
8125 * $B@EQU - BASIC COMPILER EQUATES *
8126 * $I@EQU - BASIC INTERPRETER EQUATES *
8127 * $I@SEQ/$I@LEQ - STANDARD/LONG PRECISION EXECUTION EQUATES *
8128 * *
8129 * OTHER *
8130 * NONE *
8131 * *
8132 *****
2100 2100 8133 ORG *,B@LVPG,0 PLACE AT A PAGE BOUNDARY
2100 2100 8134 SFGBS1 EQU * ESTABLISH BASE REGISTER USE
2100 2100 8135 USING SFGBS1,@BR
2100 2100 8136 SFGETR EQU * ENTRY POINT
2100 1C 01 144A FA 8137 MVC I$VADR,SFGVD2(@VADDR,@BR) ACCESS VM DIRECTORY 2 (D2) AND
2105 C0 87 1349 8138 B I$MDFY * SET FOR WRITE BACK
2109 C0 87 1354 8139 B I$LOCK LOCK IT IN CORE ALSO
210D 35 02 144C 8140 L I$CADR,@XR POINT XR TO CURRENT FILE ENTRY
2111 B6 02 01 8141 A @$D2CF(,@XR),@XR *
2114 B8 08 01 8142 TBN @$D2IO(,@XR),@$M2CI CURRENT USAGE = INPUT ?
2117 F2 10 16 8143 JT SFG200 YES, GO CHECK DEVICE CODE
211A B8 04 01 8144 TBN @$D2IO(,@XR),@$M2CO CURRENT USAGE = OUTPUT ?

```



## SFGETR - PROLOGUE - VM GET ROUTINE

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	31/05/21	PAGE 111
211D	F2	10	06		8145	JT	SFG120			YES, GO SET USAGE ERR
2120	B8	80	01		8146	TBN	@\$D2IO(, @XR), @\$M2FI			FILE USAGE - INPUT ?
2123	F2	10	07		8147	JT	SFG150			YES, GO SET CURRENT USAGE
2126	3C	B7	0CBC		8148	SFG120 MVI	I\$ERRC, @@E712			SET INVALID FILE USAGE ERR CODE
212A	F2	87	BF		8149	J	SFG295			RETURN
212D	BA	08	01		8150	SFG150 SBN	@\$D2IO(, @XR), @\$M2CI			SET CURRENT USAGE TO INPUT
					8151	*				
					8152	*	FILE MAY CURRENTLY BE USED FOR INPUT. CHECK INPUT DEVICE.			
					8153	*				
2130	B8	20	00		8154	SFG200 TBN	@\$D2DC(, @XR), @\$MBCD			CARD FILE ?
2133	F2	90	B0		8155	JF	SFG290			NO, GO ACCESS DISK ROUTINE
					8156	*				
					8157	*	CARD FILE - ACCESS BUFFER AND LOCK IT IN CORE			
					8158	*				
2136	BD	FF	05		8159	CLI	@\$D2CB(, @XR), SFGNFM			FIRST ACCESS OF THIS FILE ?
2139	F2	81	06		8160	JE	SFG205			NO, BYPASS INITIALIZATION
213C	7C	80	5B		8161	MVI	SFG210+@Q(, @BR), @NOP			FORCE AN INITIAL CARD RFAD
213F	BC	FF	05		8162	MVI	@\$D2CB(, @XR), SFGNFM			SET OFF FIRST ACCESS INDICATOR
2142	6C	00	4A 02		8163	SFG205 MVC	SFGBVA-1(, @BR), @\$D2VB(@\$L2VB, @XR)			SET UP VADDR OF BUFFER
2146	C0	87	1330		8164	B	I\$LDXR			ACCESS BFR & POINT @XR TO IT
214A				214B	8165	SFGBVA DS	CL(@VADDR)			VADDR OF BUFFER
214B					8166	ORG	SFGBVA			* INITIALIZE BYTE DISPLACEMENT
214B	00			214B	8167	DC	XL1'0'			* TO ZERO
214C	74	02	FC		8168	ST	SFGCBA(, @BR), @XR			SAVE BUFFER CORE ADDRESS
214F	C0	87	1349		8169	B	I\$MDFY			SET FOR WRITE BACK 1-3
2153	C0	87	1354		8170	B	I\$LOCK			* IN CORE
2157	BD	60	FF		8171	CLI	SFGCBP(, @XR), @CARDL			MORE VALUES ON CARD ?
215A	F2	01	70		8172	SFG210 JC	SFG280, @BNE			YES, BYPASS CARD READ-FORCE 1ST
215D	7C	01	5B		8173	MVI	SFG210+@Q(, @BR), @BNE			RESET FORCE 1ST READ SWITCH
					8174	*				
					8175	*	A CARD READ IS REQUIRED			
					8176	*				
2160	C0	87	12B1		8177	SFG215 B	I\$CALL			GO READ A
2164	2A00			2165	8178	DC	AL(@VADDR)(V\$SCDI)			* CARD
					8179	*				
2166	BC	1E	60		8180	MVI	@CARDL(, @XR), @EOS			PUT AN EOS FOLLOWING CARD
2169	7C	40	74		8181	MVI	SFG225+@D1(, @BR), SFGICR			INITLX FOR BLANKS VALID FIRST
216C	7C	87	C5		8182	MVI	SFG270+@Q(, @BR), @UCB			INITLX FOR A BLANK CARD
					8183	*				
					8184	*	FORCE A COMMA DELIMITER BETWEEN DATA ITEMS			
					8185	*				
216F	BD	40	00		8186	SFG220 CLI	@ZERO(, @XR), @BLANK			THIS CHARACTER A BLANK ?
2172	F2	81	40		8187	SFG225 JE	SFG255			YES, GO INCR IF TRANSPARENT NOW
					8188	*				
2175	BD	6B	00		8189	SFG227 CLI	@ZERO(, @XR), @COMMA			THIS CHARACTER A COMMA ?
2178	F2	81	40		8190	JE	SFG260			YES, GO SET TRANSPARENT BLANK SW
					8191	*				
217B	BD	1E	00		8192	CLI	@ZERO(, @XR), @EOS			THIS CHARACTER A EOS ?
217E	F2	81	40		8193	JE	SFG265			YES, GO RELOAD BUFFER POINTER
					8194	*				
2181	7C	80	C5		8195	MVI	SFG270+@Q(, @BR), @NOP			SET NON-BLANK CARD SWITCH
					8196	*				
					8197	*	CHECK FOR AND HANDLE CHARACTER CONSTANTS			
					8198	*				
2184	BD	7D	00		8199	CLI	@ZERO(, @XR), B@SQUO			THIS CHARACTER A QUOTE ?
2187	F2	01	15		8200	JNE	SFG235			NO, BYPASS CHAR CON HANDLING

## SFGETR - PROLOGUE - VM GET ROUTINE

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	31/05/21	PAGE 112
					8201	*				
	218A	E2	02	01	8202	SFG230	LA @B1(, @XR), @XR			INCR BFR POINTER TO NEXT CHAR
					8203	*				
	218D	BD	1E	00	8204		CLI @ZERO(, @XR), @EOS			THIS CHARACTER THE OEOS ?
	2190	F2	81	4C	8205		JE SFG285			YES, GO SET ERROR CODE
					8206	*				
	2193	BD	7D	00	8207		CLI @ZERO(, @XR), @SQUO			THIS CHARACTER A QUOTE ?
	2196	D0	01	8A	8208		BNE SFG230(, @BR)			NO, GO INCR TO NEXT
					8209	*				
	2199	BD	7D	01	8210		CLI @B1(, @XR), @SQUO			NEXT CHARACTER A QUOTE ?
	219C	F2	81	16	8211		JE SFG255			YES, GO INCR TWICE
					8212	*				
					8213	*	CHECK FOR COMMA DELIMITER REQUIRED			
					8214	*				
	219F	7D	3D	74	8215	SFG235	CLI SFG225+@D1(, @BR), SFGBLK			IS A BLANK(S) THE ONLY DELIMITER
	21A2	7C	3A	74	8216		MVI SFG225+@D1(, @BR), SFG255			SET SW TO: NEXT BLANK A DELIMITR
	21A5	F2	01	0D	8217		JNE SFG255			NOT ONLY DELIM, GO INCR TO NEXT
					8218	*				
	21A8	3C	6B	0000	8219	SFG240	MVI *-*, @COMMA			REPLACE BLANK DELIM WITH COMMA
	21AC	F2	87	06	8220		J SFG255			GO INCR TO NEXT CHARACTER
					8221	*				
	21AF	7C	3D	74	8222	SFG245	MVI SFG225+@D1(, @BR), SFGBLK			SET SW TO: BLANK ONLY DELIMITER
					8223	*				
	21B2	74	02	AB	8224	SFG250	ST SFG240+@OP1(, @BR), @XR			SAVE BLANK DELIMITER LOCATION
					8225	*				
					8226	*	INCREMENT BUFFER POINTER TO NEXT CHARACTER			
					8227	*				
	21B5	E2	02	01	8228	SFG255	LA @B1(, @XR), @XR			INCR POINTER
	21B8	D0	87	6F	8229		B SFG220(, @BR)			GO CHECK NEXT CHARACTER
					8230	*				
					8231	*	COMMA RECOGNIZED AS THE REAL DELIMITER			
					8232	*				
	21BB	7C	40	74	8233	SFG260	MVI SFG225+@D1(, @BR), SFGICR			SET SW TO: BLANKS TRANSPARENT
	21BE	D0	87	B5	8234		B SFG255(, @BR)			GO INCR TO NEXT CHARACTER
					8235	*				
					8236	*	ENTIRE CARD HAS BEEN CORRECTLY DELIMITED			
					8237	*				
	21C1	75	02	FC	8238	SFG265	L SFGCBA(, @BR), @XR			SET POINTER TO BUFFER START
					8239	*				
	21C4	D0	87	60	8240	SFG270	BC SFG215(, @BR), @UCB			GO READ NEXT CARD IF THIS BLANK
					8241	*				
					8242	*	GO TO PAGE 3 OF SFGETR TO SYNTAX CHECK THE DATA ITEMS			
					8243	*				
	21C7	C0	87	12B1	8244		B I\$CALL			GO TO PAGE 3 OF STGETR
	21CB	23B7			21CC 8245		DC AL(@VADDR)(V\$XSGT+SFG920-SFGBS1) * FOR SYNTAX CHECK			
					8246	*				
					8247	*	GO TO PAGE 3 OF SFGETR TO CONVERT AND STACK THE NEXT DATA ITEM			
					8248	*				
	21CD	C0	87	12B1	8249	SFG280	B I\$CALL			GO TO PAGE 3 OF SFGETR TO
	21D1	23C6			21D2 8250		DC AL(@VADDR)(V\$XSGT+SFG930-SFGBS1) * CONVERT & STACK ITEM			
					8251	*				
	21D3	1C	01	144A 4B	8252	SFG282	MVC I\$VADR, SFGBVA(@VADDR, @BR)			UNLOCK PAGE 1-3
	21D8	C0	87	1350	8253		B I\$UNLK			* BUFFER PAGE 1-3
	21DC	F2	87	0D	8254		J SFG295			GO TO GENERAL SFGETR EXIT
					8255	*				
	21DF	3C	BD	0CBC	8256	SFG285	MVI I\$ERRC, @@E718			SET ERROR CODE

## SFGETR - PROLOGUE - VM GET ROUTINE

ERR	LOC	OBJECT CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	31/05/21	PAGE 113
	21E3	D0 87 D3		8257	B	SFG282(,@BR)		GO UNLOCK BUFFER	1-3
				8258	*				
				8259	*	DISK FILE - GO TO NEXT PAGE OF SFGETR			
				8260	*				
	21E6	C0 87 12B1		8261	SFG290 B	I\$CALL		EXEC PART 2 OF SFGETR - DISK	
	21EA	2200	21EB	8262	DC	AL(@VADDR)(V\$XSGT+B@BLSZ)	*	FILE DATA ACCESS	
				8263	*				
				8264	*	GENERAL EXIT FROM SFGETR			
				8265	*				
	21EC	1C 01 144A FA		8266	SFG295 MVC	I\$VADR,SFGVD2(@VADDR,@BR)		UNLOCK	
	21F1	C0 87 1350		8267	B	I\$UNLK		* DIRECTORY 2	
	21F5	C0 87 12D3		8268	B	I\$RTRN		RETURN TO CALLER	
				8269	*				
				8270	*	CONSTANTS, WORKAREAS & EQUATES			
				8271	*				
	21F9	0100	21FA	8272	SFGVD2 DC	AL(@VADDR)(V\$SFD2)		VADDR OF VM DIRECTORY 2	
				8273	*				
	21FB		21FC	8274	SFGCBA DS	CL(@CADDR)		SAVE FLD FOR CORE BFR ADDR	
				8275	*				
			00FF	8276	SFGNFM EQU	X'FF'		NOT FIRST CARD FILE ACCESS MASK	
			00FF	8277	SFGCBP EQU	X'FF'		DISP. TO CARD BUFFER POINTER	
			0040	8278	SFGICR EQU	SFG255-SFG227		DISP. TO BLANK TRANSPARENT	
			003D	8279	SFGBLK EQU	SFG250-SFG227		DISP. TO BLANK DEAMITER	
			003A	8280	SFGRST EQU	SFG245-SFG227		DISP. TO BLANK RESET	
				8281	*				
				8282	*	END OF SFGETR - PART 1			
				8283	*				

## SFGETR - PROLOGUE - VM GET ROUTINE

ERR LOC OBJECT CODE ADDR STMT SOURCE STATEMENT VER 15, MOD 00 31/05/21 PAGE 114

```

2200          8285      ORG    *,B@LVP,0          PLACE AT A PAGE BOUNDARY
          2200 8286 SFGBS2 EQU    *          ESTABLISH BASE
          2200 8287      USING SFGBS2,@BR        * REGISTER USAGE
          8288 *-----
          8289 *          UPON ENTRY TO PART 2:
          8290 *
          8291 *          1. A DISK BEEN SET FOR INPUT IS TO BE ACCESSED.
          8292 *          2. D2 HAS BEEN LOCKED IN CORE WITH THE MODIFY
          8293 *          INDR SET ON.
          8294 *          3. @XR POINTS TO THE CURRENT D2 ENTRY.
          8295 *-----
          8296 *
          8297 *          TEST IF CURRENT SEGMENT EMPTY
          8298 *
2200 9D 01 0D E2 8299      CLC    @$D2LC(@$L2LC,@XR),SFGZRO(@BR) THIS SEGMENT = ZERO ?
2204 F2 01 06    8300      JNE    SFG450          YES, BYPASS NEXT SEGMENT ACCESS
          8301 *
          8302 *          CALL PART 3 OF SFGETR TO ACCESS NEXT SEGMENT
          8303 *
2207 C0 87 12B1 8304      B      I$CALL          EXECUTE PART 3 OF SFGETR TO
220B 2300        220C 8305      DC    AL(@VADDR)(V$XSGT+2*B@BLSZ) * ACCESS NEXT SEGMENT
          8306 *
          8307 *          ACCESS CURRENT BUFFER PAGE AND CHECK FOR EOS
          8308 *
220D 7C 05 E8    8309 SFG450 MVI    SFGCNL(@BR),I@LPFS      SET DATA ITEM LNG TO SHORT PREC
2210 7C 80 B0    8310      MVI    SFG575+@Q(@BR),@NOP      SET PREC ADJ. SWITCH FOR SHORT
2213 B8 20 01    8311      TBN    @$D2IO(@XR),@$M2FP      LONG PRECISION ?
2216 F2 90 07    8312      JF     SFG470          NO, BYPASS ADD TO LONG PRECISION
2219 7C 87 B0    8313      MVI    SFG575+@Q(@BR),@UCB      SET PREC ADJ. SWITCH FOR LONG
221C 5E 00 E8 E3 8314      ALC    SFGCNL(@BR),SFGDLS(1,@BR) INCR DATA ITEM LNG TO LONG PRC
2220 6C 01 E6 0D 8315 SFG470 MVC    SFGSDF(@BR),@$D2LC(@$L2LC,@XR) SAVE SDF COUNT
2224 74 02 76    8316      ST     SFGD2P(@BR),@XR          SAVE D2 ENTRY POINTER
2227 6C 01 34 05 8317      MVC    SFGVCB(@BR),@$D2CP(@$L2CP,@XR) SET UP VADDR OF CURRENT
222B 6E 00 33 02 8318      ALC    SFGVCB-1(@BR),@$D2VB(@$L2VB,@XR) * DATA ITEM IN VM BUFFER
222F C0 87 1330 8319      B      I$LDXR          ACCESS AND POINT
2233          2234 8320 SFGVCB DS    CL(@VADDR)          * @XR TO IT
2235 BD 1C 00    8321      CLI    @ZERO(@XR),@EOF          END OF FILE ?
2238 F2 81 95    8322      JE     SFG690          YES, GO SET ERROR CODE
          8323 *
          8324 *          DETERMINE LENGTH OF DATA ITEM & PLACE IT IN STACK
          8325 *
223B 4C 01 70 0D4E 8326      MVC    SFGMTA(@CADDR,@BR),I$STAK INITIALIZE MOVE TO ADDRESS
2240 B8 40 00    8327      TBN    @ZERO(@XR),B@DTYP      CHARACTER CONSTANT ?
2243 F2 90 03    8328      JF     SFG500          NO, NUM LNG ALREADY SET. BYPASS
2246 7C 13 E8    8329      MVI    SFGCNL(@BR),I@LCRV      SET DATA ITEM LENGTH FOR CHAR.
          8330 *
2249 5C 01 EA E8 8331 SFG500 MVC    SFGPCL(@BR),SFGCNL(2,@BR) INIT FOR FULL DATA ITEM MOVE
224D 5D 01 E6 E8 8332      CLC    SFGSDF(@BR),SFGCNL(2,@BR) ALL OF DATA ITEM IN BUFFER
2251 F2 02 04    8333      JNL    SFG520          YES, GO SET UP MOVE
2254 5C 01 EA E6 8334      MVC    SFGPCL(@BR),SFGSDF(2,@BR) NO, RESET MOVE LNG FOR PARTIAL
          8335 *
2258 7C FF EC    8336 SFG520 MVI    SFGMLQ(@BR),SFGMS1      SET MOVE LENGTH FOR PART OF
225B 5E 00 EC EA 8337      ALC    SFGMLQ(@BR),SFGPCL(1,@BR) * DATA ITEM IN CURRENT BUFFER
225F 5C 00 6E EC 8338      MVC    SFG550+@Q(@BR),SFGMLQ(1,@BR) SET IN MOVE INSTRUCTION
          8339 *
2263 76 02 EC    8340      A      SFGMLQ(@BR),@XR          INCR @XR TO END OR BFR DATA

```

## SFGETR - PROLOGUE - VM GET ROUTINE

ERR LOC OBJECT CODE ADDR STMT SOURCE STATEMENT VER 15, MOD 00 31/05/21 PAGE 115

			8341 *			
2266	74 02 72		8342	ST	SFGMFA(, @BR), @XR	SET CADDR IN MOVE FROM CADDR
			8343 *			
2269	5E 00 70 EC		8344	ALC	SFGMTA(, @BR), SFGMLQ(1, @BR)	INCR MOVE TO CADDR
			8345 *			
226D	0C 00 0000 0000		8346	SFG550 MVC	*-*( @VQ), *-*	MOVE DATA FROM BUFFER TO STACK
		2270	8347	SFGMTA EQU	SFG550+@OP1	* MOVE TO STACK ADDRESS
		2272	8348	SFGMFA EQU	SFG550+@OP2	* MOVE FROM BUFFER ADDRESS
			8349 *			
			8350 *		UPDATE D2 ENTRY POINTERS & CHECK IF ALL OF	
			8351 *		DATA ITEM MOVED	
			8352 *			
2273	C2 02 0000		8353	SFG555 LA	*-*, @XR	POINT @XR BACK TO D2 ENTRY
		2276	8354	SFGD2P EQU	SFG555+@OP1	* D2 ENTRY CADDR_SAVE AREA
2277	9E 01 05 EA		8355	ALC	@\$D2CP(@\$L2CP, @XR), SFGPCL(, @BR)	INCR BFR PT BY MOVE LNG
227B	9F 01 0D EA		8356	SLC	@\$D2LC(@\$L2LC, @XR), SFGPCL(, @BR)	INCR SDF COUNT BY MOV LNG
227F	5F 00 E8 EA		8357	SLC	SFGCNL(, @BR), SFGPCL(1, @BR)	DECR REQ'D BY ACTUAL LENGTH
2283	F2 81 1F		8358	JZ	SFG570	BYPASS BFR REFILL IF DIFRNCE = 0
			8359 *			
			8360 *		ONLY PART OF THE DATA ITEM WAS IN THE CURRENT	
			8361 *		SEGMENT, ACCESS NEXT SEGMENT.	
			8362 *		POINT @XR TO NEW SEGMENT.	
			8363 *		REDO MOVE PROCESSING FOR SECOND PART OF DATA ITEM	
			8364 *		MOVE.	
			8365 *			
2286	C0 87 12B1		8366	B	I\$CALL	EXECUTE PART 3 OF SFGETR TO
228A	2300	228B	8367	DC	AL(@VADDR)(V\$XSGT+2*B@BLSZ)	* ACCESS NEXT SEGMENT
			8368 *			
228C	6C 01 E6 0D		8369	MVC	SFGSDF(, @BR), @\$D2LC(@\$L2LC, @XR)	SET NEW SEG CT IN SAVEFLD
2290	6C 01 9D 05		8370	MVC	SFGVNB(, @BR), @\$D2CP(@\$L2CP, @XR)	SET UP VADDR OF NEW
2294	6E 00 9C 02		8371	ALC	SFGVNB-1(, @BR), @\$D2VB(@\$L2VB, @XR)	* SEGMENT
2298	C0 87 1330		8372	B	I\$LDXR	ACCESS & POINT @XR AT IT
229C		229D	8373	SFGVNB DS	CL(@VADDR)	VADDR OF NEW SEGMENT
229E	5E 00 70 E4		8374	ALC	SFGMTA(, @BR), SFGONE(1, @BR)	INCR MOVE TO ADDR ROR NEXT MOV
22A2	D0 87 49		8375	B	SFG500(, @BR)	SO MOVE REST OF DATA ITEM
			8376 *			
			8377 *		ENTIRE DATA ITEM MOVED TO STACK. SET CORRECT	
			8378 *		PRECISION IF NUMERIC.	
			8379 *			
22A5	35 02 0D4E		8380	SFG570 L	I\$STAK, @XR	POINT @XR TO STACKED DATA ITEM
22A9	B8 40 00		8381	TBN	I@STAT(, @XR), B@DTYP	CHARACTER ITEM ?
22AC	F2 10 25		8382	JT	SFG695	YES, GO TO RETURN
			8383 *			
22AF	F2 80 11		8384	SFG575 JC	SFG585, @NOP	JUMP IF FILE PREC = LONG
22B2	F2 87 1F		8385	JC	SFG695, I@PRSW	JUMP TO EXIT IF RUN PREC = SHORT
			8386 *			
			8387 *		CHANGE PRECISION FROM SHORT TO LONG	
			8388 *			
22B5	BA 20 00		8389	SBN	I@STAT(, @XR), B@PREC	SET PREC = LONG
22B8	AC 00 08 04		8390	MVC	I@PEXL(, @XR), I@PEXS(@B1, @XR)	MOVE EXP TO LONG POSITION
22BC	AF 03 07 07		8391	SLC	I@PEXL-1(, @XR), I@PEXL-1(SFGELS, @XR)	SET EXTRA DIGITS = 0
22C0	F2 87 11		8392	J	SFG695	EXIT
			8393 *			
22C3	F2 80 0E		8394	SFG585 JC	SFG695, @UCB-I@PRSW+@NOP	JUMP TO EXIT IF RUN PREC = LONG
			8395 *			
			8396 *		CHANGE PRECISION FROM LONG TO S4ORT	



## SFGETR - PROLOGUE - VM GET ROUTINE

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	31/05/21	PAGE 116
				8397	*					
22C6	BB	20	00	8398		SBF	I@STAT(,@XR),B@PREC		SET PREC = SHORT	
22C9	AC	00	04 08	8399		MVC	I@PEXS(,@XR),I@PEXL(1,@XR)		MOVE EXP TO SHORT POSITION	
22CD	F2	87	04	8400		J	SFG695		JUMP TO EXIT	
				8401	*					
				8402	*		SET END OF FILE ERROR CODE			
				8403	*					
22D0	3C	B9	0CBC	8404	SFG690	MVI	I\$ERRC,@@E714		SET EOF CODE	
				8405	*					
				8406	*		RETURN TO PART 2 OF SFGETR			
				8407	*					
22D4	1C	01	144A 9D	8408	SFG695	MVC	I\$VADR,SFGVNB(@VADDR,@BR)		MOVE BUFFER PAGE	1-5
22D9	C0	87	1350	8409		B	I\$UNLK		UNLOCK PAGE	1-5
22DD	C0	87	12D3	8410		B	I\$RTRN		EXIT	
				8411	*					
				8412	*		PART 2 - CONSTANTS, WORKAREAS & EQUATES			
				8413	*					
22E1	0000			22E2	8414	SFGZRO	DC	XL(@\$L2CP)'0'	ZERO	
22E3	04			22E3	8415	SFGDLS	DC	AL1(I@LPFL-I@LPFS)	DIFFERENCE IN PRECISION LENGTHS	
22E4	01			22E4	8416	SFGONE	DC	XL1'1'	ONE	
				8417	*					
				00FF	8418	SFGMS1	EQU	X'FF'	MINUS 1	
				0004	8419	SFGELS	EQU	I@LPFL-I@LPFS	LNG LONG PREC EXTRA SIGNIFICNCE	
				8420	*					
22E5				22E6	8421	SFGSDF	DS	CL(@\$L2LC)	SDF COUNT WORKAREA	
22E7				22E8	8422	SFGCNL	DS	CL(@CADDR)	ACTUAL LENGTH OF DATA ITEM	
22E7				8423		ORG	SFGCNL-1		* INITIALIZE TO	
22E7	0000			22E8	8424		DC	XL(@CADDR)'0'	* ZERO	
22E9				22EA	8425	SFGPCL	DS	CL(@CADDR)	BUFFER LNG OF DATA ITEM	
22EB				22EC	8426	SFGMLQ	DS	CL(@CADDR)	PHYS. MOVE LNG & DISPLACEMENT	
22EB				8427		ORG	SFGMLQ-1		* INITIALIZE TO	
22EB	0000			22EC	8428		DC	XL(@CADDR)'0'	* ZERO	
				8429	*					
				8430	*		END OF SFGETR - PART 2			
				8431	*					

## SFGETR - PROLOGUE - VM GET ROUTINE

ERR LOC OBJECT CODE ADDR STMT SOURCE STATEMENT VER 15, MOD 00 31/05/21 PAGE 117

2300			8433	ORG	*,B@LVP,0	PLACE AT A PAGE BOUNDARY
			2300 8434	SFGBS3 EQU	*	ESTABLISH BASE
			2300 8435	USING	SFGBS3,@BR	* REGISTER USAGE
			8436	*	-----	
			8437	*	UPON ENTRY TO PART 3:	
			8438	*		
			8439	*	1. D2 HAS BEEN LOCKED IN CORE WITH THE	
			8440	*	MODIFY INDICATOR SET ON.	
			8441	*	2. @XR POINTS TO THE CURRENT D2 ENTRY	
			8442	*	3. THE CURRENT D2 ENTRY VM BUFFER POINTERS	
			8443	*	MUST BE SET TO THE FIRST DATA ITEM IN THE	
			8444	*	NEXT(FIRST) SEGMENT OR BUFFER.	
			8445	*	-----	
			8446	*		
			8447	*	CHECK FOR MORE SEGMENTS IN CURRENT BUFFER	
			8448	*		
2300	BD 00 05		8449	SFG750 CLI	@\$D2CB(,@XR),@ZERO	ANY SPACE LEFT IN CURR BFR ?
2303	F2 01 52		8450	JNE	SFG830	YES, GO ACCESS BUFFER
2306	BD 00 04		8451	CLI	@\$D2CS(,@XR),@ZERO	INITIAL FILL-UP CALL ?
2309	F2 81 00		8452	JE	SFG760	YES, GO TO GET SFLOAD
230C	AD 00 03 04		8453	SFG760 CLC	@\$D2BS(,@XR),@\$D2CS(@\$L2CS,@XR)	MORE VM BUFFERS ?
2310	F2 84 32		8454	JH	SFG810	YES, GO CHECK DATA FILE TYPE
			8455	*		
			8456	*	VM BUFFERS MUST BE REFILLED. WRITE OUT INTERPRETER	
			8457	*	AND ACCESS & EXECUTE SFLOAD.	
			8458	*		
2313	74 01 1E		8459	SFG780 ST	SFGWPL(,@BR),@BR	SET UP DPL TO WRITE OUT
2316	7C E3 1E		8460	MVI	SFGWPL(,@BR),SFGDWL	* INTERPRETER
			8461	*SFG785 DISK	@ZERO	GO WRITE IT OUT
2319	C0 87 0025		8462	SFG785 B	\$DISKN	PERFORM PHYSICAL DISK OP
231D	0000	231E	8463	DC	AL2(@ZERO)	DPL ADDRESS
			8464	***	END OF EXPANSION ***	
		231E	8466	SFGWPL EQU	SFG785+5	ADDRESS OF WRITE DPL
231F	74 01 34		8467	ST	SFGRPL(,@BR),@BR	SET UP DPL TO READ IN
2322	7C E9 34		8468	MVI	SFGRPL(,@BR),SFGDRL	* #SFLOA
2325	3C 01 0D58		8469	MVI	I\$WRK1-1,@DGET	SET INPUT INDR FOR #SFLOA
2329	74 01 3A		8470	ST	SFGSBR(,@BR),@BR	SAVE BASE REGISTER
232C	74 02 3E		8471	ST	SFGSXR(,@BR),@XR	SAVE D2 POINTER
			8472	*SFG790 BLOAD	@ZERO	GO EXECUTE #SFLOA
232F	C0 87 0522		8473	SFG790 B	\$BLOAD	LOAD AND EXECUTE WORK AREA PGM
2333	0000	2334	8474	DC	AL2(@ZERO)	DPL ADDRESS
			8475	***	END OF EXPANSION ***	
		2334	8477	SFGRPL EQU	SFG790+5	ADDRESS OF READ DPL
			8478	*		
			8479	*	RETURN FROM \$SFLOA	
			8480	*		
2335	0444	2336	8481	DC	AL(@CADDR)(\$DPLSV-5)	CADDR OF DPL TO RELOAD INTERP
2337	C2 01 0000		8482	SFG795 LA	*-*,@BR	RESTORE BASE REGISTER
		233A	8483	SFGSBR EQU	SFG795+@OP1	CADDR OF @BR SAVE AREA
233B	C2 02 0000		8484	SFG800 LA	*-*,@XR	RESTORE D2 POINTER
		233E	8485	SFGSXR EQU	SFG800+@OP1	CADDR OF D2 POINTER SAVE AREA
			8486	*	DISK \$WAITF	WAIT FOR INTERPRETER
233F	C0 87 0025		8487	B	\$DISKN	PERFORM PHYSICAL DISK OP
2343	057F	2344	8488	DC	AL2(\$WAITF)	DPL ADDRESS



## SFGETR - PROLOGUE - VM GET ROUTINE

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	31/05/21	PAGE 118
				8489	***	END OF EXPANSION	***			
				8491	*					
				8492	*	DETERMINE DATA FILE TYPE				
				8493	*					
2345	B8	80	01	8494	SFG810	TBN	@\$D2IO(, @XR), @\$M2FI		PROG-GENERATED DATA FILE ?	
2348	F2	90	0A	8495		JF	SFG825		NO, GO HANDLE KEYBOARD DATA FILE	
234B	BC	00	05	8496		MVI	@\$D2CB(, @XR), @ZERO		SET BYTE POINTER TO ZERO	
234E	9C	01	0D	8497		MVC	@\$D2LC(@\$L2LC, @XR), SFGSSZ(, @BR)		SET SEG COUNT TO MAX.	
2352	F2	87	5E	8498		J	SFG900		GO TO EXIT	
2355	7C	01	05	8500	SFG825	MVI	@\$D2CB(, @BR), @B1		SET BYTE POINTER TO 1ST SDF	
				8501	*					
				8502	*	KEYBOARD DATA FILE - CHECK FIRST/NEXT SDF				
				8503	*					
2358	74	02	3E	8504	SFG830	ST	SFGSXR(, @BR), @XR		SAVE D2 ENTRY POINTER	
235B	6C	01	68	8505		MVC	SFGCBV(, @BR), @\$D2CP(@VADDR, @XR)		SET UP VADDR OF	
235F	6E	00	67	8506		ALC	SFGCBV-1(, @BR), @\$D2VB(@\$L2VB, @XR)		* CURRENT BUFFER	
2363	C0	87	1330	8507		B	I\$LDXR		ACCESS & PT @XR TO NEXT SDF	
2367				2368 8508	SFGCBV	DS	CL(@VADDR)		VADDR OF CURRENT BUFFER	
				8509	*					
2369	BD	80	00	8510		CLI	@SDF0(, @XR), @SNUL		IS THE NEXT SEGMENT NULL ?	
236C	F2	01	0A	8511		JNE	SFG840		NO, GO CHECK SEGMENT TYPE	
				8512	*					
236F	75	02	3E	8513		L	SFGSXR(, @BR), @XR		RESTORE D2 ENTRY POINTER	
2372	9E	00	04	8514		ALC	@\$D2CS(@\$L2CS, @XR), SFGPAF(, @BR)		NULL, INCR PT TO NEXT PAGE	
2376	D0	87	0C	8515		B	SFG760(, @BR)		GO ACCESS NEXT BUFFER	
				8516	*					
				8517	*	TEST NEXT SEGMENT TYPE AND USAGE STATUS				
				8518	*					
2379	6C	06	FA	8519	SFG840	MVC	SFGSHD(SFGHDL, @BR), SFGDEH(, @XR)		MOVE SEG HDR TO SAVE AREA	
237D	75	02	3E	8520		L	SFGSXR(, @BR), @XR		RESTORE D2 ENTRY POINTER	
2380	78	02	F6	8521		TBN	SFGLEH+SDF2(, @BR), @SLAST		PRIMARY SEGMENT ?	
2383	F2	90	0E	8522		JF	SFG860		YES, GO CHECK IF DISABLED	
				8523	*					
				8524	*	SECONDARY SEGMENT				
				8525	*					
2386	F2	80	14	8526	SFG850	JC	SFG870, @NOP		JUMP IF LINE DISABLED	
2389	9E	00	05	8527		ALC	@\$D2CB(@\$L2CB, @XR), SFGSSL(, @BR)		INCR CURR PT BY HDR LNG	
238D	5F	00	F5	8528		SLC	SFGLEH+SDF1(1, @BR), SFGSSL(, @BR)		DECR SEG CT BY HDR LNG	
2391	F2	87	1B	8529		J	SFG890		GO SET ADJ SEG CT IN D2 ENTRY	
				8530	*					
				8531	*	PRIMARY SEGMENT				
				8532	*					
2394	78	80	FA	8533	SFG860	TBN	SFGLEH+@STYPE(, @BR), B@SDMK		STATEMENT DISABLED ?	
2397	F2	90	0A	8534		JF	SFG880		NO, BYPASS BYPASS OF SEG	
239A	7C	87	87	8535		MVI	SFG850+@Q(, @BR), @UCB		SET SWITCH FOR 2NDARY SEGMENTS	
239D	9E	01	05	8536	SFG870	ALC	@\$D2CP(@\$L2CP, @XR), SFGLEH+SDF1(, @BR)		INCR CURR PT BY LNG	
23A1	D0	87	00	8537		B	SFG750(, @BR)		GO ACCESS THE NEXT SEGMENT	
				8538	*					
23A4	7C	80	87	8539	SFG880	MVI	SFG850+@Q(, @BR), @NOP		RESET DISABLED SWITCH	
23A7	9E	00	05	8540		ALC	@\$D2CB(@\$L2CB, @XR), SFGPSL(, @BR)		INCR CURR PT BY HDR LNG	
23AB	5F	00	F5	8541		SLC	SFGLEH+SDF1(1, @BR), SFGPSL(, @BR)		DECR SEG CT BY HDR LNG	
23AF	9C	01	0D	8542	SFG890	MVC	@\$D2LC(@\$L2LC, @XR), SFGLEH+SDF1(, @BR)		SET SEG CT IN ENTRY	
				8543	*					
				8544	*	ALL DONE - GO AWAY				

## SFGETR - PROLOGUE - VM GET ROUTINE

ERR LOC OBJECT CODE ADDR STMT SOURCE STATEMENT VER 15, MOD 00 31/05/21 PAGE 119

```

      23B3 C0 87 12D3      8545 *
                        8546 SFG900 B      I$RTRN      RETURN TO PART 2 OF SFGETR
                        8547 *
                        8548 *      SYNTAX CHECK THE ENTIRE CARD
                        8549 *
      23B7 74 02 FA      8550 SFG920 ST      SFGSB2(,@BR),@XR      SAVE THE BUFFER POINTER
                        8551 *
      23BA C0 87 12B1      8552      B      I$CALL      GO SYNTAY CWECK FOR
      23BE 2E00      23BF 8553      DC      AL(@VADDR)(V$CDSY)      * VALID DATA ITEMS
                        8554 *
      23C0 75 02 FA      8555      L      SFGSB2(,@BR),@XR      RESTORE THE BUFFER POINTER
      23C3 F2 87 16      8556      J      SFG940      JUMP TO CONFIGURE 2ND PASS      1-3
                        8557 *
                        8558 *      CONVERT AND STACK THE NEXT DATA ITEM UNLESS AN ERROR HAS OCCURED
                        8559 *
      23C6 F2 80 07      8560 SFG930 JC      SFG935,@NOP      JOMP FIRST PASS      1-3
      23C9 74 02 FA      8561      ST      SFGSB2(,@BR),@XR      SAVE XR(BUFFER CADDR)      1-3
      23CC 9C 00 FE F9      8562      MVC      SFGXRD(@CADDR-1,@XR),SFGB2-1(,@BR) SET TRUE BUFR CADR 1-3
                        8563 *
      23D0 C0 87 12B1      8564 SFG935 B      I$CALL      SO CONVERT AND STACK NEXT
      23D4 3100      23D5 8565      DC      AL(@VADDR)(V$CDCV)      * DATA ITEM
                        8566 *
      23D6 7C 80 C7      8567      MVI      SFG930+@Q(,@BR),@NOP      FORCE SAVE BUFFER CADDR      1-3
      23D9 F2 87 03      8568      J      SFG945      JUMP TO RETURN TO CALLER      1-3
                        8569 *
      23DC 7C 87 C7      8570 SFG940 MVI      SFG930+@Q(,@BR),@UCB      FORCE NO SAVE OR BUFR ADDR      1-3
                        8571 *
      23DF C0 87 12D3      8572 SFG945 B      I$RTRN      RETURN TO CALLER      1-3
                        8573 *
                        8574 *      PART 3 - DISK PARAMETER LISTS.
                        8575 *
                        8576 *      WRITE OUT INTERPRETER
                        8577 *
                        8578 *SFGPLW DPL      FUNC-@DPUT,DADDR-#@VSFI,CNT=##@VSL,CADDR-#$$INS
      23E3 02      23E3 8579 SFGPLW EQU      *      DISK PARAMETER LIST
      23E4 09A1      23E5 8580      DC      AL1(@DPUT)      REQUESTED FUNCTION
      23E6 0F      23E6 8581      DC      AL2(##@VSFI)      DISK ADDRESS
      23E7 0600      23E6 8582      DC      AL1(##@VSL)      SECTOR COUNT
                        23E8 8583      DC      AL2(##$INS)      BUFFER ADDRESS
                        8584 *** END OF EXPANSION ***
                        00E3 8586 SFGDWL EQU      SFGPLW-SFGBS3      DISP. TO WRITE DPL
                        8587 *
                        8588 *      READ IN SFLOAD
                        8589 *
                        8590 *SFGPLR DPL      FUNC-@DGET,DADDR-#@SFLO,CNT=##@SFL,CADDR-#$$SFL
      23E9 01      23E9 8591 SFGPLR EQU      *      DISK PARAMETER LIST
      23EA 0499      23E9 8592      DC      AL1(@DGET)      REQUESTED FUNCTION
      23EC 05      23EB 8593      DC      AL2(##@SFLO)      DISK ADDRESS
      23ED 0F00      23EC 8594      DC      AL1(##@SFL)      SECTOR COUNT
                        23EE 8595      DC      AL2(##$SFL)      BUFFER ADDRESS
                        8596 *** END OF EXPANSION ***
                        00E9 8598 SFGDRL EQU      SFGPLR-SFGBS3      DISP TO READ DPL
                        8599 *
      8600 *      PART 3 - CONSTANTS, RORKAREAS AND EQUATES.

```

SFGETR - PROLOGUE - VM GET ROUTINE

ERR LOC		OBJECT CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	31/05/21	PAGE 120
				8601	*				
23EF	0100		23F0	8602	SFGSSZ DC	AL(@\$L2LC)(B@BLSZ)			MAX SEGMENT COUNT
23F1	01		23F1	8603	SFGPAF DC	XL(@\$L2CS)'1'			
23F2	04		23F2	8604	SFGSSL DC	XL(@\$L2CB)'4'			LENGTH OF 2NDARY SEG. HDR.
23F3	07		23F3	8605	SFGPSL DC	XL(@\$L2CB)'7'			LENGTH OF PRIMARY SEG. HDR.
				8606	*				
			0F00	8607	SFGSA0 EQU	\$\$KLD1+X'0900'			CORE LOAD ADOR OF #SFLOAD
			0007	8608	SFGHDL EQU	@STEXT			SEGMENT HEADER LENGTH
			0006	8609	SFGDEH EQU	SFGHDL-1			DISP TO RIGHT END OF SEG. HDR.
				8610	*				
23F4			23F4	8611	SFGLEH EQU	*			LEFT END OF HEADER SAVE AREA
			23FA	8612	SFGSHD DS	CL(SFGHDL)			SEGMENT HEADER SAVE AREA
			23FA	8613	SFGSB2 EQU	SFGSHD			SAVE AREA FOR CARD SBR ADDRESS
			00FE	8614	SFGXRD EQU	X'FE'			BUFFER CADDR DISP INTO BFR 1-3
				8615	*				
				8616	*	END OF SFGETR PART 3			
				8617	*				

## SFGETR - PROLOGUE - VM GET ROUTINE

ERR LOC OBJECT CODE ADDR STMT SOURCE STATEMENT VER 15, MOD 00 31/05/21 PAGE 121

2400			8619	ORG	*,B@LVP,0	PLACE MODULE AT PAGE BOUNDARY
			2400 8620	USING	SFRBS1,@BR	ESTABLISH BASE REGISTER
			2400 8621	SFRBS1 EQU	*	IT ON FIRST BYTE OF PAGE
			8622 *			
			8623 *			TERMINATION ENTRY TO CLOSE ALL DATA FILES
			8624 *			
			2400 8625	SFRCAL EQU	*	
2400 7C 80 73			8626	MVI	SFR900+@Q(,@BR),@NOP	SET SW FOR CLOSE ALL LOOP
2403 7C 80 17			8627	MVI	SFR110+@Q(,@BR),@NOP	SET SW FOR CLOSE ALL INITIALIZED
2406 3A 1E 03E4			8628	SBN	\$LPRP3,@KENAB	SET MATRIX PRINTER MODE 1-3
			8629 *			
			8630 *			ENTRY TO CLOSE A SPECIFIED DATA FILE
			8631 *			
			240A 8632	SFRCLS EQU	*	
240A 7C 87 50			8633	MVI	SFR140+@Q(,@BR),@UCB	SET SWITCH FOR CLOSE
			8634 *			
			8635 *			ENTRY TO RESET A SPECIFIED FILE
			8636 *			
			240D 8637	SFRSET EQU	*	
			8638 *			
			8639 *			ACCESS DIRECTORY 2 & REFERENCE SPECIFIED FILE
			8640 *			
240D C0 87 1330			8641	SFR100 B	I\$LDXR	GET VM DIRECTORY 2 AND
2411 0100		2412	8642	SFRVD2 DC	AL(@VADDR)(V\$SFD2)	* POINT @XR TO IT
2413 74 02 84			8643	ST	SFRIXR(,@BR),@XR	SAVE POINTER TO D2 RECORD
			8644 *			
			8645 *			SET TO FIRST ENTRY IF CLOSE ALL
			8646 *			
2416 F2 87 03			8647	SFR110 JC	SFR115,@UCB	JUMP IF NOT CLOSE ALL
2419 BC 02 01			8648	MVI	@\$D2CF(,@XR),@XR	SET DISPLACEMENT TO 1ST ENTRY
			8649 *			
241C B6 02 01			8650	SFR115 A	@\$D2CF(,@XR),@XR	INCR @XR TO SPECIFIED FILE
			8651 *			
			8652 *			DETERMINE IF THE FILE IS INPUT OR OUTPUT
			8653 *			
241F BD 00 00			8654	SFR130 CLI	@\$D2DC(,@XR),@ZERO	THIS FILE ACTIVE ?
2422 F2 81 4D			8655	JE	SFR900	NO, GO CHECK IF CLOSE ALL
2425 B8 08 01			8656	TBN	@\$D2IO(,@XR),@\$M2CI	CURRENT USAGE - INPUT ?
2428 F2 10 24			8657	JT	SFR140	YES, BYPASS CLOSE CALL TO SFPUT
242B B8 04 01			8658	TBN	@\$D2IO(,@XR),@\$M2CO	CURRENT USAGE - OUTPUT ?
242E F2 90 41			8659	JF	SFR900	NO, NOT ACTIVE. GO CHK CLOSE ALL
2431 BA 02 01			8660	SBN	@\$D2IO(,@XR),@\$M2EF	SET END OF FILE INDR
2434 74 02 4B			8661	ST	SFR135+@OP1(,@BR),@XR	SAVE D2 ENTRY POINTER
2437 35 02 0D4E			8662	L	I\$STAK,@XR	MOVE AN END OF FILE CODE
243B BC 1C 00			8663	MVI	I@STAT(,@XR),@EOF	* TO THE STACK
243E C0 87 1354			8664	B	I\$LOCK	LOCK D2 IN CORE
2442 C0 87 12B1			8665	B	I\$CALL	CALL SFPUTR TO EMPTY THE
2446 1D00		2447	8666	DC	AL(@VADDR)(V\$XSPT)	* FILE BUFFER(S)
2448 C2 02 0000			8667	SFR135 LA	*-*,@XR	RESTORE D2 ENTRY POINTER
244C BB 02 01			8668	SBF	@\$D2IO(,@XR),@\$M2EF	SET THE END OF FILE INDR OFF
			8669 *			
			8670 *			CHECK IF RESET OR CLOSE
			8671 *			
244F F2 80 0F			8672	SFR140 JC	SFR300,@NOP	JUMP IF CLOSE

## SFGETR - PROLOGUE - VM GET ROUTINE

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00 31/05/21 PAGE 122
				8674	*			
				8675	*		RESET REQUIRED	
				8676	*			
2452	AF	01	05 05	8677	SFR200	SLC	@\$D2CP(,@XR),@\$D2CP(@\$L2CP,@XR) CLEAR CURRENT POINTER	
2456	AF	01	09 09	8678		SLC	@\$D2DD(@\$L2DD,@XR),@\$D2DD(,@XR) CLEAR DISK DISP	
245A	AF	01	0D 0D	8679		SLC	@\$D2LC(@\$L2LC,@XR),@\$D2LC(,@XR) CLEAR SPF COUNT	
245E	F2	87	11	8680		J	SFR900 GO CHECK IF CLOSE ALL	
				8681	*			
				8682	*		CLOSE REQUIRED	
				8683	*			
2461	BB	0C	01	8684	SFR300	SBF	@\$D2IO(,@XR),@\$M2CI+@\$M2CO SET CURRENT USAGE OFF	
2464	B8	40	00	8685		TBN	@\$D2DC(,@XR),@\$MBSD SCRATCH DISK FILE ?	
2467	D0	10	52	8686		BT	SFR200(,@BR) YES, GO CLEAR CURRENT POINTER'S	
246A	AF	01	01 01	8687		SLC	@\$D2IO(,@XR),@\$D2IO(@\$L2DC+@\$L2IO,@XR) CLEAR ENTRY EXCEPT	
246E	AF	0B	0F 0F	8688		SLC	@\$D2EE(,@XR),@\$D2EE(@\$L2E-@\$D2CS,@XR) * FOR VM BFR BSE&SIZ	
				8689	*			
				8690	*		SPECIFIED FILE HAS SEEN CLOSED OR RESET AS REQUIRED.	
				8691	*		IF CLOSE ALL CONTINUE TILL ALL 8 ENTRIES CLOSED	
				8692	*			
2472	F2	87	17	8693	SFR900	JC	SFR995,@UCB JUMP TO RETURN IF NOT CLOSE ALL	
2475	5F	00	AB AA	8694		SLC	SFRNOE(,@BR),SFRONE(1,@BR) DECR NO. OF ENTRIES COUNTER	
2479	F2	81	10	8695		JZ	SFR995 JUMP TO RETURN IF ZERO.	
247C	1C	01	144A 12	8696		MVC	I\$VADR,SFRVD2(@VADDR,@BR) RESTORE VADDR OF D2 TO PG.RTN.	
2481	C2	02	0000	8697	SFR950	LA	*-*,@XR RESTORE POINTER TO D2 RECORD	
				2484 8698	SFR1XR	EQU	SFR950+@OP1 SAVE AREA FOR D2 RCD POINTER	
2485	9E	00	01 AC	8699		ALC	@\$D2CF(,@XR),SFRX10(1,@BR) INCR FILE PT TO NEXT ENTRY	
2489	D0	87	1C	8700		B	SFR115(,@BR) GO INCR @XR TO NEXT ENTRY & CHK	
				8701	*			
				8702	*		FUNCTION COMPLETE - RESTORE ROUTINE & EXIT	
				8703	*			
248C	7C	80	50	8704	SFR995	MVI	SFR140+@Q(,@BR),@NOP SET RTN FOR RESET FUNCTION	
248F	3B	1E	03E4	8705		SBF	\$LPRP3,@KENAB RESET MATRIX PRINT MODE 1-3	
2493	1C	01	144A 12	8706		MVC	I\$VADR,SFRVD2(@VADDR,@BR) SPECIFY DIRECTORY 2	
2498	C0	87	1349	8707		B	I\$MDFY SET PAGE TO MODIFIED	
249C	7C	87	17	8708		MVI	SFR110+@Q(,@BR),@UCB RESET JUMP CONDITION 1-5	
249F	7C	87	73	8709		MVI	SFR900+@Q(,@BR),@UCB RESET JUMP CONDITION 1-5	
24A2	C0	87	1350	8710		B	I\$UNLK UNLOCK PAGE	
24A6	C0	87	12D3	8711		B	I\$RTRN RETURN TO CALLER	
				8712	*			
				8713	*		CONSTANTS, WORKAREAS & EQUATES	
				8714	*			
24AA	01			24AA 8715	SFRONE	DC	XL1'1' SIMPLY ONE	
				8716	*			
24AB				24AB 8717	SFRNOE	DS	CL1 NUMBER OF D2 ENTRIES COUNTER	
24AB				8718		ORG	SFRNOE * INITIALIZE TO MAXIMUM	
24AB	0C			24AB 8719		DC	AL1(@\$MBEN) * NUMBER OF D2 ENTRIES	
24AC	10			24AC 8720	SFRX10	DC	XL1'10' D2 ENTRY LENGTH	

## SFGETR - PROLOGUE - VM GET ROUTINE

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	31/05/21	PAGE 123
				8722		*****				
				8723	*					*
				8724	*	LINE PRINTER CLOSE OUT ROUTINE				*
				8725	*					*
				8726		*****				*
				2400	8727	USING SFRBS1,@BR	SET BASE REGISTER USAGE			1-4
24AD	F1	E2	00		8728	APL @PBUSY	LOOP PRINTER BUSY			1-4
24B0	3D	00	03E3		8729	CLI \$BUFPT,@ZERO	IS LINE PRINTER BUFFER EMPTY			1-4
24B4	F2	81	13		8730	JE SFR997	JUMP IF BUFFER EMPTY			1-4
24B7	74	02	C6		8731	SFRLPR ST SFR996+@OP1(,@BR),@XR	SAVE XR			1-4
24BA	D2	02	E5		8732	LA SF1000(,@BR),@XR	XR = CADDR PPL			1-4
24BD	C0	87	12B1		8733	B I\$CALL	BRANCH TO CALL ROUTINE			1-4
24C1	2800			24C2	8734	DC AL(@VADDR)(V\$SPRT)	MATRIX PRINTER PAGE			1-4
24C3	C2	02	0000		8735	SFR996 LA *-*,@XR	RESTORE XR			1-4
24C7	F2	87	07		8736	J SFR998	GO RESTORE TRUE POSITION			1-4
24CA	38	01	03E4		8737	SFR997 TBN \$LPRP3,@INDEX	IS DUMMY PRT POSITION IN USE			1-4
24CE	F2	90	06		8738	JF SFR999	JUMP IF NOT			1-4
24D1	0C	00	03C2 03E5		8739	SFR998 MVC \$PRPOS(1),\$LPROS	RESTORE TRUE PRINT POSITION			1-4
24D7	3C	00	03E4		8740	SFR999 MVI \$LPRP3,@ZERO	RESET LINE PRINTER FLAGS			1-4
24DB	F1	E2	00		8741	APL @PBUSY	LOOP IF PRINTER BUSY			1-4
24DE	D1	E0	B7		8742	TIO SFRLPR(,@BR),@PERR	BRANCH IF PRINTER UNIT CHECK			1-4
24E1	C0	87	12D3		8743	B I\$RTRN	RETURN TO CALLER			1-4
					8744	*				1-4
24E5	80			24E5	8745	SF1000 DC XL1'80'	PPL - FORCE CARRAGE RETURN			1-4
					8746	*				
					8747	*****	END SFRSET			*****



DFKEYN - VIRTUAL MEMORY KEYBOARD ROUTINE					
ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 31/05/21 PAGE 124
		8749		*****	*
		8750	*	5703-XM1 COPYRIGHT IBM CORP. 1970	*
		8751	*	REFER TO INSTRUCTIONS ON COPY RIGHT NOTICE, 120-2083	*
		8752	*		*
		8753		*****	*
		8754	*	*STATUS	*
		8755	*	VERSION 1 MODIFICATION 0	*
		8756	*		*
		8757	*	*FUNCTION	*
		8758	*	DFKEYN IS DIVIDED INTO TWO SECTIONS PERFORMING TWO GENERAL	*
		8759	*	FUNCTIONS:	*
		8760	*	* CALL SECTION	*
		8761	*	THE CALL SECTION ENABLES AND UNLOCKS THE KEYBOARD IN	*
		8762	*	PREPARATION FOR LINE INPUT. IT THEN SETS THE INTERRUPT	*
		8763	*	ADDRESS WHICH IS ENTERED ON THE KEYBOARD INTERRUPT LEVEL WHEN	*
		8764	*	A KEY IS DEPRESSED.	*
		8765	*	* INTERRUPT SECTION	*
		8766	*	THE INTERRUPT SECTION SAVES THE SYSTEM STATUS (BR, XR & PSR)	*
		8767	*	AND HANDLES THE INPUT FROM THE KEYBOARD. UPON COMPLETION OF	*
		8768	*	THE INPUT LINE, \$KYBSY IS SET TO ZERO INDICATING THAT THE	*
		8769	*	LINE IS COMPLETE. THEN THE KEYBOARD IS LOCKED.	*
		8770	*	THE INPUT FROM THE KEYBOARD IS CLASSIFIED AND HANDLED AS	*
		8771	*	FOLLOWS:	*
		8772	*	* DATA KEYS -- THE CHARACTER REPRESENTATION IS PLACED IN	*
		8773	*	THE INPUT LINE BUFFER AND PRINTED ON THE	*
		8774	*	SYSTEM PRINTER.	*
		8775	*	* CMD KEYS -- IF THE CRT IS AVAILABLE, DSPLYN IS CALLED	*
		8776	*	TO SET THE FUNCTION FOR KEYS 12-16.	*
		8777	*	AN INDICATOR IS PLACED IN THE INPUT LINE	*
		8778	*	BUFFER (SPECIFIED LOCATION) FOR COMMAND	*
		8779	*	KEYS 1-11.	*
		8780	*	* FUNCTION KEYS -- AS FOLLOWS	*
		8781	*	TAB - IF THE CURRENT POSITION IN THE LINE BUFFER IS	*
		8782	*	POINTING WITHIN AN EYISTING LINE, THE OLD	*
		8783	*	CHARACTER IS PRINTED. IF NOT, A BLANK IS PRINTED	*
		8784	*	THIS POSITIONS THE CARRIER ONE SPACE TO THE RIGHT	*
		8785	*	IF THE KEY IS HELD DOWN, THE TYPOMATIC FEATURE IS	*
		8786	*	ACTIVATED, REPEATING THE ABOVE FUNCTION UNTIL	*
		8787	*	KEY IS RELEASED.	*
		8788	*	BACKSPACE - IF THE SYSTEM PRINTER IS THE MATRIX PRINTER	*
		8789	*	AND IF THIS WAS THE FIRST BACKSPACE FOR THE	*
		8790	*	CURRENT LINE, THE CARRIAGE IS INDEXED AND	*
		8791	*	BACKSPACED ONE POSITION. OTHERWISE, THE INDEX	*
		8792	*	FEATURE IS NOT EXECUTED. IF THE KEY IS MELD DOWN	*
		8793	*	THE TYPOMATIC FEATURE IS ACTIVATED AND THE ABOVE	*
		8794	*	FUNCTION IS REPETED UNTIL THE KEY IS RELEASED.	*
		8795	*	RETURN - THE CARRIAGE IS RETURNED ON THE SYSTEM PRINTER	*
		8796	*	AND \$KYBSY SET TO ZERO INDICATING THE LINE IS	*
		8797	*	COMPLETE. THE KEYBOARD IS THEN LOCKED.	*
		8798	*	ERASE - THE CARRIAGE IS RETURNED ON THE SYSTEM PRINTER	*
		8799	*	ALLOWING THE LINE TO BE RE-ENTERED.	*
		8800	*	INQUIRY REQUEST - THE CURRENT OPERATION IS ABORTED.	*
		8801	*	THIS KEY IS NEVER LOCKED.	*
		8802	*	NOTE: THE ENTER(+) AND PROGRAM START KEYS ARE IGNORED	*

ERR	LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00	31/05/21	PAGE 124
				8749	*****			*
				8750	* 5703-XM1 COPYRIGHT IBM CORP. 1970			*
				8751	* REFER TO INSTRUCTIONS ON COPY RIGHT NOTICE, 120-2083			*
				8752	*			*
				8753	*****			*
				8754	*STATUS			*
				8755	* VERSION 1 MODIFICATION 0			*
				8756	*			*
				8757	*FUNCTION			*
				8758	* DFKEYN IS DIVIDED INTO TWO SECTIONS PERFORMING TWO GENERAL			*
				8759	* FUNCTIONS:			*
				8760	* * CALL SECTION			*
				8761	* THE CALL SECTION ENABLES AND UNLOCKS THE KEYBOARD IN			*
				8762	* PREPARATION FOR LINE INPUT. IT THEN SETS THE INTERRUPT			*
				8763	* ADDRESS WHICH IS ENTERED ON THE KEYBOARD INTERRUPT LEVEL WHEN			*
				8764	* A KEY IS DEPRESSED.			*
				8765	* * INTERRUPT SECTION			*
				8766	* THE INTERRUPT SECTION SAVES THE SYSTEM STATUS (BR, XR & PSR)			*
				8767	* AND HANDLES THE INPUT FROM THE KEYBOARD. UPON COMPLETION OF			*
				8768	* THE INPUT LINE, \$KYBSY IS SET TO ZERO INDICATING THAT THE			*
				8769	* LINE IS COMPLETE. THEN THE KEYBOARD IS LOCKED.			*
				8770	* THE INPUT FROM THE KEYBOARD IS CLASSIFIED AND HANDLED AS			*
				8771	* FOLLOWS:			*
				8772	* * DATA KEYS -- THE CHARACTER REPRESENTATION IS PLACED IN			*
				8773	* THE INPUT LINE BUFFER AND PRINTED ON THE			*
				8774	* SYSTEM PRINTER.			*
				8775	* * CMD KEYS -- IF THE CRT IS AVAILABLE, DSPLYN IS CALLED			*
				8776	* TO SET THE FUNCTION FOR KEYS 12-16.			*
				8777	* AN INDICATOR IS PLACED IN THE INPUT LINE			*
				8778	* BUFFER (SPECIFIED LOCATION) FOR COMMAND			*
				8779	* KEYS 1-11.			*
				8780	* * FUNCTION KEYS -- AS FOLLOWS			*
				8781	* TAB - IF THE CURRENT POSITION IN THE LINE BUFFER IS			*
				8782	* POINTING WITHIN AN EYISTING LINE, THE OLD			*
				8783	* CHARACTER IS PRINTED. IF NOT, A BLANK IS PRINTED			*
				8784	* THIS POSITIONS THE CARRIER ONE SPACE TO THE RIGHT			*
				8785	* IF THE KEY IS HELD DOWN, THE TYPOMATIC FEATURE IS			*
				8786	* ACTIVATED, REPEATING THE ABOVE FUNCTION UNTIL			*
				8787	* KEY IS RELEASED.			*
				8788	* BACKSPACE - IF THE SYSTEM PRINTER IS THE MATRIX PRINTER			*
				8789	* AND IF THIS WAS THE FIRST BACKSPACE FOR THE			*
				8790	* CURRENT LINE, THE CARRIAGE IS INDEXED AND			*
				8791	* BACKSPACED ONE POSITION. OTHERWISE, THE INDEX			*
				8792	* FEATURE IS NOT EXECUTED. IF THE KEY IS MELD DOWN			*
				8793	* THE TYPOMATIC FEATURE IS ACTIVATED AND THE ABOVE			*
				8794	* FUNCTION IS REPETED UNTIL THE KEY IS RELEASED.			*
				8795	* RETURN - THE CARRIAGE IS RETURNED ON THE SYSTEM PRINTER			*
				8796	* AND \$KYBSY SET TO ZERO INDICATING THE LINE IS			*
				8797	* COMPLETE. THE KEYBOARD IS THEN LOCKED.			*
				8798	* ERASE - THE CARRIAGE IS RETURNED ON THE SYSTEM PRINTER			*
				8799	* ALLOWING THE LINE TO BE RE-ENTERED.			*
				8800	* INQUIRY REQUEST - THE CURRENT OPERATION IS ABORTED.			*
				8801	* THIS KEY IS NEVER LOCKED.			*
				8802	* NOTE: THE ENTER(+) AND PROGRAM START KEYS ARE IGNORED			*
				8803	*			*
				8804	*ENTRY POINTS			*



## DFKEYN - VIRTUAL MEMORY KEYBOARD ROUTINE

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 31/05/21 PAGE 125
		8805	*	ENTRY TO DFKEYN IS VIA THE VIRTUAL MEMORY PAGING ROUTINE.	*
		8806	*	THE CALLING SEQUENCE IS:	*
		8807	*	B I\$CALL	*
		8808	*	DC AL(@CADDR)(V\$SKEY)	*
		8809	*	WHERE V\$SKEY IS THE VIRTUAL ADDR OF THE VIRTUAL MEMORY KEYBOARD	*
		8810	*	INPUT IOCR. THE CALL TO DFKEYN INCLUDES THE PASSING IN @XR OF	*
		8811	*	THE ADDRESS OF THE INPUT DATA BUFFER.	*
		8812	*		*
		8813	*	*INPUT	*
		8814	*	INPUT TO THE VIRTUAL MEMORY IOCR IS THE ADDRESS IN @XR OF THE	*
		8815	*	INPUT LINE BUFFER AND THE INPUT DATA.	*
		8816	*		*
		8817	*	*OUTPUT	*
		8818	*	THE OUTPUT FROM THIS ROUTINE IS AN EBCDIC CHARACTER TO THE SYSTEM	*
		8819	*	PRINTER AND THE LINE BUFFER.	*
		8820	*		*
		8821	*	*EXTERNAL REFERENCES	*
		8822	*	\$BRSAB - COMMON SAVE AREA FOR BASE REGISTER	*
		8823	*	\$CIENT - NUCLEUS ENTRY FOR INTERRUPTS	*
		8824	*	\$PRDEV - INDICATOR FOR CURRENT I/O DEVICE	*
		8825	*	\$KEYCD - THUNCATED LINE INDICATOR (\$TRUNK)	*
		8826	*	\$IOIND - HARD I/O ERROR INDICATOR (\$HRDER) SYSTEM STATUS	*
		8827	*	\$INDR2 - ERROR PENDING INDICATOR (\$ERPND)	*
		8828	*	\$HIST1 - ERROR HISTORY LOG	*
		8829	*	\$PLYN - ENTRY TO CRT IOCS	*
		8830	*	I\$CALL - VIRTUAL MEMORY PAGING ROUTINE	*
		8831	*	* INDICATORS FOR VM ROUTINE	*
		8832	*	I\$LDBR	*
		8833	*	I\$LOCK	*
		8834	*	I\$LDXR	*
		8835	*	I\$VADR	*
		8836	*	I\$UNLK	*
		8837	*	I\$RTRN	*
		8838	*	V\$SKEY - VIRTUAL ADDRESS OF DFKEYN	*
		8839	*	V\$SPRT - VIRTUAL ADDRESS OF DFPRNT	*
		8840	*		*
		8841	*	*EXITS, NORMAL	*
		8842	*	EXIT IS TO THE CALLING PROGRAM VIA A BRANCH TO THE VIRTUAL MEMORY	*
		8843	*	PAGING ROUTINE.	*
		8844	*	B I\$RTRN	*
		8845	*		*
		8846	*	*EXITS, ERROR	*
		8847	*	A DATA PARITY ERROR WILL BE RETRIED ONCE. THE SUCCESSIVE PARITY	*
		8848	*	ERRORS WILL CAUSE A SYSTEM GENERATED HARD HALT.	*
		8849	*		*
		8850	*	*TABLES/WORKAREAS	*
		8851	*	DEPTBL - KEYBOARD TABLE CONTAINING THE EBCDIC CHARACTER CODES	*
		8852	*	ARRANGED SUCH THAT AN INDEX VALUE IS SENSED FROM THE	*
		8853	*	KEYBOARD AND USED AS A DISPLACEMENT INTO THE TABLE TO	*
		8854	*	FETCH THE PROPER EBCDIC VALUE. THE TABLE IS INITIALIZED	*
		8855	*	TO KEYBOARD TYPE KB1, BUT MAY BE CHANGED TO REFLECT	*
		8856	*	THE CONFIGURATION RECORD.	*
		8857	*		*
		8858	*	*ATTRIBUTES	*
		8859	*	NATURALLY RELOCATABLE	*
		8860	*		*

## DFKEYN - VIRTUAL MEMORY KEYBOARD ROUTINE

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 31/05/21 PAGE 126
		8861	*	*CHARACTER CODE DEPENDENCY	*
		8862	*	* THE OPERATION OF THIS MODULE DEPENDS UPON AN INTERNAL	*
		8863	*	* REPRESENTATION OF THE EXTERNAL CHARACTER SET WHICH IS EQUIVALENT	*
		8864	*	* TO THE ONE USED AT ASSEMBLY TIME, THE CODING HAS BEEN ARRANGED	*
		8865	*	* SC THAT REDEFINITION OF CHARACTER CONSTANTS, BY REASSEMBLY, WILL	*
		8866	*	* RESULT IN A CORRECT MODULE FOR THE NEW DEFINITIONS.	*
		8867	*		*
		8868	*	*NOTES	*
		8869	*	* ERROR PRJCEDURES	*
		8870	*	* UPON DETECTION OF A DATA REGISTER PARITY ERROR, THE SYSTEM WILL	*
		8871	*	* HALT INDICATING TO THE USER THAT A PARITY ERROR SAS OCCURRED.	*
		8872	*	* TO CONTINUE, OR RETRY THE CHARACTER, THE START SWITCH MUST BE	*
		8873	*	* PRESSED. THE ERROR IS LOGGED IN THE COUNT LOG ON DISK.	*
		8874	*	* IF ANOTHER IS DETECTED, THE HISTORY LOG IS UPDATED AND A HARD	*
		8875	*	* HALT EXECUTED.	*
		8876	*		*
		8877	*	* RESISTER USAGE	*
		8878	*	* * THE @XR IS USED FOR PASSING THE ADDRESS OF THE INPUT DATA	*
		8879	*	* BUFFER.	*
		8880	*	* * THE @XR IS ALSO USED AS A BASE REGISTER FOR PAGE 3	*
		8881	*	* * THE @BR IS USED AS A BASE REGISTER FOR PAGE 2.	*
		8882	*	* * BOTH PliAR AND IliAR ARE USED FOR BRANCHING BETWEEN PROGRAM	*
		8883	*	* AND INTERRUPT LEVEL.	*
		8884	*	* * THE RESISTERS ARE SAVED AND RESTORED.	*
		8885	*		*
		8886	*	* SAVED/RESTORED AREAS	*
		8887	*	* NONE	*
		8888	*		*
		8889	*	* MODIFICATION CONSIDERATIONS	*
		8890	*	* NONE	*
		8891	*		*
		8892	*	* REQUIRED MODULES	*
		8893	*	* FFPRNT - VIRTUAL MEMORY MATRIX PRINTER IOCR	*
		8894	*	* @SYSE0 - GENERAL SYSTEM ELATES	*
		8895	*	* @HDWEQ - HARDWARE VALUE EQUATES	*
		8896	*	* @FXDEQ - NUCLEUS LOCATION EQUATES	*
		8897	*	* @CANEQ - COMMON CORE LOCATION-MKFTS	*
		8898	*	* @CY0EQ - CYLINDER ZERO EQUATES	*
		8899	*	* @HLTEQ - HALT CODE EQUATES	*
		8900	*		*
		8901	*	* OTHER	*
		8902	*	* NONE	*
		8903	*		*
		8904	*	*****	*

## DFKEYN - VIRTUAL MEMORY KEYBOARD ROUTINE

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	31/05/21	PAGE 127
					8906	*****	*****			
					8907	* PAGE 1	*			
					8908	*****	*****			
	2500				8909	ORG	*,256,0			
				2500	8910	USING	DFKEYN,@BR			INITIAL BASE REG FOR PAGE 1
				2700	8911	USING	DFKBS3,@XR			BASE VALUE FOR PAGE 3
					8912	*				
				2500	8913	DFKEYN EQU	*			ENTRY TO ENABLE INPUT
	2500	34	01	03C5	8914	ST	\$BRSAV,@BR			SAVE PAGE 1 ADDRESS
	2504	74	01	68	8915	ST	DFK100+@OP1(,@BR),@BR			SET DFK100 TO
	2507	7C	65	68	8916	MVI	DFK100+@OP1(,@BR),DFKDIO			* BRANCH TO ITSELF
	250A	C0	87	1329	8917	B	I\$LDBR			LOAD PAGE 2 USING BR
	250E	2600			250F	8918	DC	AL2(V\$SKEY+DFKBS2-DFKEYN)		VADDR FOR PAGE 2
	2510	C0	87	1354	8919	B	I\$LOCK			LOCK PAGE 2
				2600	8920	USING	DFKBS2,@BR			BASE VALUE FOR PAGE 2
	2514	7C	20	BC	8921	MVI	DFKP10-1(,@BR),X'20'			INITIALIZE DSPLYN ADDR
	2517	4E	00	BC 043B	8922	ALC	DFKP10-1(1,@BR),\$EXFTR			CALCULATE DSPLYN ENTRY ADDRESS
	251C	74	02	28	8923	ST	DFKLMG(,@BR),@XR			SAVE INPUT LINE ADDRESS
	251F	74	02	26	8924	ST	DFKSTN(,@BR),@XR			SET STARTING DATA ADDRESS
	2522	74	02	2A	8925	ST	DFKRMG(,@BR),@XR			SET STARTING ADDR IN RIGHT ADDR
	2525	4C	00	1F 03C0	8926	MVC	DFKNPS(1,@BR),\$RMGRN			RIGHT JUSTIFY RIGHT MRGN VALUE
	252A	4F	00	1F 03C1	8927	SLC	DFKNPS(1,@BR),\$LMGRN			CALCULATE PRINTER WIDTH
	252F	5E	01	2A 1F	8928	ALC	DFKRMG(@CADDR,@BR),DFKNPS(,@BR)			CALC RIGHT MARGIN ADDR
	2533	D2	02	03	8929	LA	DFKNTR-DFKBS2(,@BR),@XR			PUT INTERRUPT ADDR IN XR
	2536	74	02	15	8930	ST	DFKIAR(,@BR),@XR			SAVE INTERRUPT ADDR FOR LOAD
	2539	7C	00	1F	8931	MVI	DFKNPS(,@BR),@ZERO			SET NO LINE POSITION CHANGE
	253C	D2	02	53	8932	LA	DFKENT-DFKBS2(,@BR),@XR			LOAD MAINLINE ENTRY ADDR
	253F	74	02	32	8933	ST	DFKROS(,@BR),@XR			SAVE MAINLINE ADDR FOR PliAR
	2542	35	02	03C5	8934	L	\$BRSAV,@XR			POINT XR TO PAGE 1
	2546	E2	02	65	8935	LA	DFK100-DFKEYN(,@XR),@XR			XR = HALT ADDRESS
	2549	74	02	30	8936	ST	DFKRET(,@BR),@XR			SAVE MAINLINE RETURN ADDRESS
	254C	E2	02	5B	8937	LA	DFKTBL-DFK100(,@XR),@XR			XR = DATA TABLE ADDRESS
	254F	74	02	17	8938	ST	DFKBLE(,@BR),@XR			SAVE DATA TABLE ADDRESS
	2552	C0	87	1330	8939	B	I\$LDXR			READ IN PAGE 3 USING XR
	2556	2700			2557	8940	DC	AL2(V\$SKEY+DFKBS3-DFKEYN)		VADDR FOR PAGE 3
	2558	C0	87	1354	8941	B	I\$LOCK			LOCK PAGE 3
	255C	74	02	36	8942	ST	DFKXRS(,@BR),@XR			SAVE PAGE 3 ADDRESS
	255F	75	C0	15	8943	L	DFKIAR(,@BR),@IliAR			LOAD INTERRUPT ADDRESS
	2562	F3	10	1E	8944	SIO	@KENAB,@KEYBD			ENABLE, UNLOCK KEYBOARD
				0065	8945	DFKDIO EQU	*-DFKEYN			DISPLACEMENT TO DFK100
	2565	C0	87	0000	8946	DFK100 B	*-*			WAIT FOR LINE
	2569	1C	01	144A 3C	8947	DFK120 MVC	I\$VADR(@VADDR),DFKPG3(,@BR)			SET PAGE 3 VADDR
	256E	C0	87	1350	8948	B	I\$UNLK			UNLOCK PAGE 3
	2572	1C	01	144A 3A	8949	MVC	I\$VADR(@VADDR),DFKPG2(,@BR)			SET PAGE 2 VADDR
	2577	C0	87	1350	8950	B	I\$UNLK			UNLOCK PAGE 2
	257B	75	02	28	8951	L	DFKLMG(,@BR),@XR			RESTORE XR TO DATA ADDRESS
	257E	C0	87	12D3	8952	DFK140 B	I\$RTRN			RETURN TO CALLING PGM
					8953	*****	*****			

## DFKEYN - VIRTUAL MEMORY KEYBOARD ROUTINE

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	31/05/21	PAGE 128
	25C0				8955	ORG	DFKEYN+256-64			PLACE DATA TABLE TO END OF PAGE
				25C0	8956	DFKTBL EQU	*			FIRST BYTE OF DATA TABLE
	25C0	F0		25C0	8957	DC	CL1'0'			0
	25C1	F1		25C1	8958	DC	CL1'1'			1
	25C2	F2		25C2	8959	DC	CL1'2'			2
	25C3	F3		25C3	8960	DC	CL1'3'			3
	25C4	F4		25C4	8961	DC	CL1'4'			4
	25C5	F5		25C5	8962	DC	CL1'5'			5
	25C6	F6		25C6	8963	DC	CL1'6'			6
	25C7	F7		25C7	8964	DC	CL1'7'			7
	25C8	F8		25C8	8965	DC	CL1'8'			8
	25C9	F9		25C9	8966	DC	CL1'9'			9
	25CA	C1		25CA	8967	DC	CL1'A'			A
	25CB	C2		25CB	8968	DC	CL1'B'			B
	25CC	C3		25CC	8969	DC	CL1'C'			C
	25CD	C4		25CD	8970	DC	CL1'D'			D
	25CE	C5		25CE	8971	DC	CL1'E'			E
	25CF	C6		25CF	8972	DC	CL1'F'			F
	25D0	5D		25D0	8973	DC	XL1'5D'			)
	25D1	5A		25D1	8974	DC	AL1(@UPARW)			UP ARROW
	25D2	7C		25D2	8975	DC	XL1'7C'			@
	25D3	78		25D3	8976	DC	XL1'78'			#
	25D4	58		25D4	8977	DC	XL1'58'			\$
	25D5	6C		25D5	8978	DC	XL1'6C'			%
	25D6	4A		25D6	8979	DC	XL1'4A'			CENTS SIGN
	25D7	50		25D7	8980	DC	XL1'50'			&
	25D8	70		25D8	8981	DC	XL1'70'			'
	25D9	4D		25D9	8982	DC	XL1'4D'			(
	25DA	C7		25DA	8983	DC	CL1'G'			G
	25DB	C8		25DB	8984	DC	CL1'H'			H
	25DC	C9		25DC	8985	DC	CL1'I'			I
	25DD	D1		25DD	8986	DC	CL1'J'			J
	25DE	D2		25DE	8987	DC	CL1'K'			K
	25DF	D3		25DF	8988	DC	CL1'L'			L
	25E0	D4		25E0	8989	DC	CL1'M'			M
	25E1	D5		25E1	8990	DC	CL1'N'			N
	25E2	D6		25E2	8991	DC	CL1'O'			O
	25E3	D7		25E3	8992	DC	CL1'P'			P
	25E4	D8		25E4	8993	DC	CL1'Q'			Q
	25E5	D9		25E5	8994	DC	CL1'R'			R
	25E6	E2		25E6	8995	DC	CL1'S'			S
	25E7	E3		25E7	8996	DC	CL1'T'			T
	25E8	E4		25E8	8997	DC	CL1'U'			U
	25E9	E5		25E9	8998	DC	CL1'V'			V
	25EA	E6		25EA	8999	DC	CL1'W'			W
	25EB	E7		25EB	9000	DC	CL1'X'			X
	25EC	E8		25EC	9001	DC	CL1'Y'			Y
	25ED	E9		25ED	9002	DC	CL1'Z'			Z
	25EE	60		25EE	9003	DC	XL1'60'			-
	25EF	7E		25EF	9004	DC	XL1'7E'			= (EQUAL SIGN)
	25F0	4E		25F0	9005	DC	CL1'+'			+ (PLUS)
	25F1	4B		25F1	9006	DC	CL1'.'			PERIOD
	25F2	5C		25F2	9007	DC	CL1'*'			; (SEMICOLON)
	25F3	5C		25F3	9008	DC	CL1'*'			*
	25F4	6B		25F4	9009	DC	CL1','			COMMA
	25F5	4B		25F5	9010	DC	CL1'.'			PERIOD

DFKEYN - VIRTUAL MEMORY KEYBOARD ROUTINE

ERR LOC		OBJECT CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	31/05/21	PAGE 129
25F6	61		25F6	9011	DC	XL1'61'			/
25F7	6F		25F7	9012	DC	XL1'6F'			?
25F8	4F		25F8	9013	DC	XL1'4F'			LOGICAL 'OR'
25F9	40		25F9	9014	DFKLKA	DC	CL1' '		BLANK
25FA	7A		25FA	9015	DC	XL1'7A'			COLON
25FB	7F		25FB	9016	DC	XL1'7F'			NOT EQUAL
25FC	4C		25FC	9017	DC	XL1'4C'			LESS NAN
25FD	6E		25FD	9018	DC	XL1'6E'			> (GREATER THAN)
25FE	6D		25FE	9019	DC	XL1'6D'			UNDER SCORE
25FF	5F		25FF	9020	DC	XL1'5F'			LOGICAL 'NOT'
				9021	*****				
			0039	9022	DFKLNK EQU	DFKLKA-DFKTBL			DISP OF BLANK IN TABLE
				9023	*****				

DFKEYN - VIRTUAL MEMORY KEYBOARD ROUTINE

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	31/05/21	PAGE 130
					9025	*****	*****			
					9026	*	PAGE 2			*
					9027	*				*
					9028	*	ONCE THE KEYBOARD HAS BEEN UNLOCKED, ALL KEYBOARD INTERRUPTS			*
					9029	*	WILL ENTER AT DFKNTR. THE INTERRUPT WILL BE SERVICED AND THE			*
					9030	*	LEVEL EXITED.			*
					9031	*****	*****			
2600					9032	ORG	DFKEYN+256			PLACE PAGE 2
				2600	9033	DFKBS2 EQU	*			PAGE 2 BASE ADDRESS
2600	F3	10	19		9034	DFK160 SIO	DFKEXL,@KEYBD			EXIT LEVEL, LOCK KEYBOARD
					9035	*				
				2603	9036	DFKNTR EQU	*			INTERRUPT ENTRY UDR
2603	75	20	32		9037	L	DFKROS(,@BR),@PLIAR			LOAD PLIAR WITH PROCESSOR ENTRY
2606	70	10	1D		9038	SNS	DFKNSK(,@BR),@KEYBD			SENSE KEYBOARD DATA
2609	5D	01	1D 34		9039	CLC	DFKNSK(@REGL,@BR),DFKIRK(,@BR)			IS IT INQUIRY REQUEST ?
260D	D0	01	00		9040	BNE	DFK160(,@BR)			GO EXIT LEVEL IF NOT
2610	C0	87	0483		9041	B	\$CIENT			GO CHECK MASK STATUS
					9042	*****	*****			

## DFKEYN - VIRTUAL MEMORY KEYBOARD ROUTINE

ERR LOC OBJECT CODE ADDR STMT SOURCE STATEMENT VER 15, MOD 00 31/05/21 PAGE 131

```

          9044 *****
          9045 *   CONSTANTS AND WORK AREAS FOR KEYBOARD IOCR   *
          9046 *****
          9047 *
2614 0000      2615 9048 DFKIAR DC      AL2(*-*)      INTERRUPT ENTRY ADDR
2616 0000      2617 9049 DFKBLE DC      AL2(*-*)      ADDR OF DATA TABLE
2618 0483      2619 9050 DFKIET DC      AL2($CIENT)    ADDR OF CI ENTRY
261A 10        261A 9051          DC      AL1(@KEYBD)   SIO Q BYTE
261B 1E        261B 9052          DC      AL1(@KENAB)   SID R BYTE - ENABLE KEYBOARD
261C          261C 9053 DFKATA DS      CL1             DATA BYTE
261D          261D 9054 DFKNSK DS      CL1             SENSE BYTE
261E 0000      261F 9055 DFKNPS DC      XL2'000'       LINE POSITION CHANGE
2620 0001      2621 9056 DFKC01 DC      XL2'0001'      CONSTANT 1
2622 00        2622 9057          DC      XL1'00'      INDEX PPL CNT BYTE
          2621 9058 DFKIST EQU      DFKC01            OBR ENTRY
          2623 9059 DFKPPL EQU      *                PRINT PPL
2623 40        2623 9060          DC      XL1'40'      PRINT COMMAND
2624          2624 9061 DFKCNT DS      CL1             PRINT COUNT
2625 0000      2626 9062          DC      AL2(*-*)      INITIAL PRINT DOSTION
          2626 9063 DFKSTN EQU      DFKPPL+@PDATA     ADDR OF CURRENT POS IN LINE BUF
2627 0000      2628 9064 DFKLMG DC      AL2(*-*)      ADDR OF LEFT POS OF LINE BUFFER
2629 0000      262A 9065 DFKRMG DC      AL2(*-*)      ADDR OF RIGHT MARGIN IN LINE
262B          262C 9066 DFKIME DS      CL2             100 MS LOOP CNTR
262D 15B3      262E 9067 DFKMCT DC      IL2'5555'     INITIAL CNT FOR 100 MS
262F          2630 9068 DFKRET DS      CL2             INTERRUPT RETURN ADDR
2631 0000      2632 9069 DFKROS DC      AL2(*-*)      MAINLINE ENTRY ADDRESS
2633 11        2633 9070          DC      AL1(DFKRKY)   I R KEY CODE
2634 10        2634 9071 DFKIRK DC      AL1(@KFUNK)     FUNCTION KEY CODE
2635          2636 9072 DFKXRS DS      CL(@CADDR)      PAGE 3 ADDR SAVE AREA
2637 0004      2638 9073 DFKXDP DC      AL2(DFK120-DFK100) INCREMENT TO JUMP HPL
2639 2600      263A 9074 DFKPG2 DC      AL2(V$SKEY+DFKBS2-DFKEYN) VADDR FOR PAGE 2
263B 2700      263C 9075 DFKPG3 DC      AL2(V$SKEY+DFKBS3-DFKEYN) VADDR FOR PAGE 3
          9076 *****
```



DFKEYN - VIRTUAL MEMORY KEYBOARD ROUTINE

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	31/05/21	PAGE 132
				9078		*****				
				9079	*	EQUATES USED FOR KEYBOARD IOCR				*
				9080		*****				
				9081	*					
		0001		9082	DFK001	EQU	1			ONE
		0005		9083	DFKTAB	EQU	X'05'			TAB KEY
		0016		9084	DFKBSP	EQU	X'16'			BACKSPACE KEY
		0015		9085	DFKRTN	EQU	X'15'			RETURN KEY
		0003		9086	DFKERS	EQU	X'03'			ERASE KEY
		0040		9087	DFKSPC	EQU	X'40'			SPACE BAR
		0011		9088	DFKRKY	EQU	X'11'			IQUIRY REQUEST KEY
		0002		9089	DFKEMS	EQU	X'02'			ENTER MINUS KEY
		0010		9090	DFKACK	EQU	X'10'			BACK SPACE CTRL
		0011		9091	DFKKIX	EQU	X'11'			BACKSPACE &INDX CTRL
		001D		9092	DFKEUD	EQU	X'1D'			EXIT, UNLOCK, DISABLE CTRL
		0018		9093	DFKLOK	EQU	X'18'			LOCK KEYBOARD CTRL
		0012		9094	DFKENB	EQU	X'12'			ENABLE INTERRUPTS CTRL
		001C		9095	DFKULK	EQU	X'1C'			UNLOCK KEYBOARD CTRL
		0019		9096	DFKEXL	EQU	X'19'			EXIT LEVEL, LOCK KEYBOARD CTRL
		0040		9097	DFKDTK	EQU	X'40'			DATA KEY FUNCTION BIT
				9098		*****				

## DFKEYN - VIRTUAL MEMORY KEYBOARD ROUTINE

ERR LOC OBJECT CODE ADDR STMT SOURCE STATEMENT VER 15, MOD 00 31/05/21 PAGE 133

```
          9100 *****
263D 75 C0 19          9101 DFK180 L      DFKIET(,@BR),@I1IAR      RESTORE INTERRUPT ADDR TO NUC
2640 F3 10 18          9102          SIO    DFKLOK,@KEYBD          LOCK KEYBOARD
2643 5E 01 30 38      9103          ALC    DFKRET(@CADDR,@BR),DFKXDP(,@BR)  DON'T DO HALT
2647 F2 87 03          9104          J      DFKNAB          DON'T UNLOCK KEYBOARD
          264A 9105 DFKXIT EQU    *          ENTRY TO EXIT DEPRES
264A F3 10 1C          9106          SIO    DFKULK,@KEYBD          UNLOCK KEYBOARD
          264D 9107 DFKNAB EQU    *          ENTRY TO ENABLE
264D F3 10 12          9108          SIO    DFKENB,@KEYBD          ENABLE INTERRUPTS
2650 75 20 30          9109          L      DFKRET(,@BR),@P1IAR      RETURN TO INTERRUPTED PROGRAM
          9110 *
          2653 9111 DFKENT EQU    *          ENTRY TO PROCESS INTERRUPT DATA
2653 D0 FF 96          9112          BC     DFKDLP(,@BR),X'FF'      UPDATE LINE POSITION
2656 78 80 1D          9113          TBN    DFKNSK(,@BR),@PRITY      TEST FOR PARITY ERROR
2659 E0 10 BB          9114          BT     DFKROR(,@XR)          JUMP IF PARITY ERROR
265C BC 87 BC          9115          MVI    DFK520+@Q(,@XR),@UCB      SET PARITY INDR OFF
265F 78 10 1D          9116          TBN    DFKNSK(,@BR),@KFUNK      FUNCTION KEY ?
2662 E0 10 00          9117          BT     DFK350(,@XR)          JUMP IF YES
2665 78 40 1D          9118          TBN    DFKNSK(,@BR),DFKDTK      DATA KEY ?
2668 D0 90 4A          9119          BF     DFKXIT(,@BR)          NO -- GO EXIT
266B D0 87 DD          9120          B      DFKTST(,@BR)          GO CHK CMND KEY ONLY, RI MRGN
266E BC 80 51          9121          MVI    DFK380+@Q-DFKBS3(,@XR),@NOP  SET BACKSPACE INDEX OFF
2671 5C 00 7C 1C      9122 DFK200 MVC    DFK220+@OPD2(1,@BR),DFKATA(,@BR)  SET DATA TBL DISP
2675 75 02 17          9123          L      DFKBLE(,@BR),@XR          *** LOAD XR WITH TABLE ADDR
2678 2C 00 0000 00    9124 DFK220 MVC    *-*(1),*-(,@XR)          MOVE DATA CHAR TO LINE BUFFER
267D D0 87 83          9125          B      DFKRT1(,@BR)          PRINT AND UPDATE POSITION
2680 D0 87 4A          9126          B      DFKXIT(,@BR)          GO EXIT
          9127 *****
```

## DFKEYN - VIRTUAL MEMORY KEYBOARD ROUTINE

ERR LOC OBJECT CODE ADDR STMT SOURCE STATEMENT VER 15, MOD 00 31/05/21 PAGE 134

```
9129 *****
9130 * THIS ROUTINE UPDATES ALL LINE BUFFER ADDRESSES IN DFKEYN BY THE *
9131 * VALUE PLACED IN 'DFKNPS'. IT CHECKS FOR MARGIN REQUIREMENTS. IF *
9132 * THE RIGHT MARGIN IS HIT, A CARRIAGE RETURN AND EOS ARE GENERATED. *
9133 * IF LEFT MARGIN IS HIT, NOTHING IS UPDATED. *
9134 * TWO ENTRY POINTS ARE PROVIDED: *
9135 * B DFKRT1(,@BR) PRINTS 1 CHAR AND UPDATES POSITION *
9136 * B DFKDLP(,@BR) UPDATES POSITION AND TEST RT MARGIN *
9137 *****
2683 9138 DFKRT1 EQU *
2683 7C 01 1F 9139 MVI DFKNPS(,@BR),DFK001 SET CHARACTER COUNT TO 1
2686 74 08 AB 9140 ST DFK260+@OP1(,@BR),@ARR SAVE RETURN ADDRESS
2689 5C 00 24 1F 9141 MVC DFKCNT(1,@BR),DFKNPS(,@BR) SET PRINT COUNT
268D D2 02 23 9142 LA DFKPPL(,@BR),@XR XR = PPL ADDRESS
2690 D0 87 AC 9143 B DFKPRT(,@BR) GO PRINT CHARACTER ON SYS PRINT
2693 F2 87 03 9144 J DFK240 GO UPDATE POSITION
9145 *
2696 9146 DFKDLP EQU * ENTRY TO UPDATE POSITION
2696 74 08 AB 9147 ST DFK260+@OP1(,@BR),@ARR SAVE RETURN ADDRESS
2699 5E 01 26 1F 9148 DFK240 ALC DFKPPL+@PDATA(@CADDR,@BR),DFKNPS(,@BR) UPDATE DATA ADDR
269D 5C 01 7B 26 9149 MVC DFK220+@OP1(@CADDR,@BR),DFKSTN(,@BR) UPDATE POS ADDR
26A1 9C 01 81 26 9150 MVC DFK480-DFKBS3+@OP1(@CADDR,@XR),DFKSTN(,@BR)
26A5 7C 00 1F 9151 MVI DFKNPS(,@BR),@ZERO ZERO LINE POSITION INCREMENT
26A8 C0 87 0000 9152 DFK260 B *-* RETURN
9153 *****
```

## DFKEYN - VIRTUAL MEMORY KEYBOARD ROUTINE

```
ERR LOC  OBJECT CODE      ADDR STMT SOURCE STATEMENT      VER 15, MOD 00  31/05/21  PAGE 135

          9155 *****
          9156 *   THIS ROUTINE DETERMINES WHICH DEVICE(S) IS TO BE USED FOR OUTPUT.  *
          9157 *   IT THEN CALL THE CORRECT IOCS.  INPUT IS THE ADDRESS OF THE PPL    *
          9158 *   STORED IN XR.  UPON EXIT XR IS RESTORED TO PAGE 3 BASE ADDRESS.    *
          9159 *****
          26AC 9160 DFKPRT EQU  *                               ENTRY TO INTERFACE
26AC 74 08 DC          9161          ST   DFK320+@OP1(,@BR),@ARR  SAVE RETURN ADDRESS
26AF 74 02 BF          9162          ST   DFKP20(,@BR),@XR      SET PPL ADDRESS FOR DSPLYN
26B2 1D 00 044A BC     9163          CLC   $PRDEV-1(1),DFKP10-1(,@BR)  TEST FOR CRT USE
26B7 F2 01 0E          9164          JNE   DFK280                SKIP CRT IF NOT IN USE
26BA C0 87 2004        9165          B     $$PLYN                GO TO CRT IOCS
          26BD 9166 DFKP10 EQU  *-1                            ADDR OF DSPLYN ENTRY
26BE 0000          26BF 9167 DFKP20 DC   AL2(*-*)                PPL ADDRESS
26C0 1D 01 044B BD     9168          CLC   $PRDEV(@CADDR),DFKP10(,@BR) IS PRINTER USED TOO ?
26C5 F2 81 0E          9169          JE    DFK300                SKIP PRINTER OP IF NOT
26C8 3A 1E 03E4        9170 DFK280 SBN  $LPRP3,@KENAB            FORCE MATRIX PRINT MODE      1-3
26CC C0 87 12B1        9171          B     I$CALL                GO TO DFPRNT
26D0 2800          26D1 9172          DC   AL2(V$SPRT)            VADDR OF DFPRNT
26D2 3B 1E 03E4        9173          SBF   $LPRP3,@KENAB            RESET MATRIX PTR. FLAGS      1-3
26D6 75 02 36          9174 DFK300 L     DFKXRS(,@BR),@XR        RESTORE PAGE 3 ADDRESS
26D9 C0 87 0000        9175 DFK320 B     *-*                    RETURN TO CALLING ROUTINE
          9176 *****
```

DFKEYN - VIRTUAL MEMORY KEYBOARD ROUTINE

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	31/05/21	PAGE 136
					9178	*****	*****			
				26DD	9179	DFKTST EQU *	ENTRY TO TEST RIGHT MARGIN			
26DD	74	08	ED		9180	ST DFK340+@OP1(,@BR),@ARR	SAVE RETURN ADDRESS			
26E0	5D	01	26 2A		9181	CLC DFKPPL+@PDATA(@CADDR,@BR),DFKRMG(,@BR)	AT RIGHT MARGIN ?			
26E4	E0	02	72		9182	BNL DFK440(,@XR)	DO CARRIER RETURN IF YES			
26E7	F3	10	1C		9183	SIO DFKULK,@KEYBD	UNLOCK KEYBOARD			
26EA	C0	87	0000		9184	DFK340 B *-*	RETURN TO CALLING ROUTINE			
					9185	*****	*****			

## DFKEYN - VIRTUAL MEMORY KEYBOARD ROUTINE

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	31/05/21	PAGE 137
					9187	*****				
					9188	* PAGE 3				*
					9189	*				*
					9190	* THIS ROUTINE CHOOSES THE DESIRED ROUTINE PER REQUEST.				*
					9191	*****				
2700					9192	ORG DFKBS2+256	PLACE PAGE 3			
				2700	9193	DFKBS3 EQU *	BASE ADDRESS FOR PAGE 3			
				2700	9194	DFK350 EQU *	ENTRY FOR FNCT KEY PROCESSING			
2700	7D	11	1C		9195	CLI DFKNSK-1(,@BR),DFKRKY	INQUIRY REQUEST ?			
2703	D0	81	3D		9196	BE DFK180(,@BR)	GO EXIT			
2706	7D	16	1C		9197	CLI DFKNSK-1(,@BR),DFKBSP	BACKSPACE KEY ?			
2709	F2	81	41		9198	JE DFKSPB	JUMP YES			
270C	7D	15	1C		9199	CLI DFKNSK-1(,@BR),DFKRTN	RETURN KEY ?			
270F	F2	81	66		9200	JE DFK460	JUMP YES			
2712	7D	03	1C		9201	CLI DFKNSK-1(,@BR),DFKERS	ERASE KEY ?			
2715	F2	81	71		9202	JE DFKERA	JUMP YES			
2718	D0	87	DD		9203	B DFKTST(,@BR)	CHECK FOR RIGHT MARGIN			
271B	7D	40	1C		9204	CLI DFKNSK-1(,@BR),DFKSPC	SPACE BAR ?			
271E	F2	81	7C		9205	JE DFKSPA	JUMP YES			
2721	7D	02	1C		9206	CLI DFKNSK-1(,@BR),DFKEMS	ENTER MINUS KEY ?			
2724	F2	81	8B		9207	JE DFK500	DO FORMS INDEV IF YES			
2727	7D	05	1C		9208	CLI DFKNSK-1(,@BR),DFKTAB	TAB KEY ?			
272A	D0	01	4A		9209	BNE DFKXIT(,@BR)	EXIT IF NO			
					9210	* CONTINUE				

## DFKEYN - VIRTUAL MEMORY KEYBOARD ROUTINE

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00 31/05/21 PAGE 138
				9212		*****		
				9213	*		ENTRY FOR TAB OPERATIONS	
272D	BC	80	51	9214		MVI DFK380+@Q(,@XR),@NOP	SET BACK SPACE INDR OFF	
2730	D0	87	83	9215		B DFKRT1(,@BR)	GO PRINT ONE CHARACTER	
				9216	*		CONTINUE TO TEST TYPO.	
				9217		*****		
				2733 9218	DFKATC	EQU *	ENTRY TO TEST TYPAMATIC	
2733	79	02	1D	9219		TBF DFKNSK(,@BR),@TYPAM	TYPAMATIC MODE ?	
2736	D0	10	4A	9220		BT DFKXIT(,@BR)	EXIT IF NO	
2739	F3	10	18	9221		SIO DFKLOK,@KEYBD	RESET BAIL FOR TYPO	
273C	5C	01	2C 2E	9222		MVC DFKIME(2,@BR),DFKMCT(,@BR)	INITIALIZE TIMING LOOP	
2740	5F	01	2C 21	9223	DFK360	SLC DFKIME(2,@BR),DFKC01(,@BR)	DECREMENT COUNTER	
2744	E0	84	40	9224		BH DFK360(,@XR)	LOOP FOR 100 MS	
2747	70	10	1D	9225		SNS DFKNSK(,@BR),@KEYBD	SENSE DATA	
274A	E0	87	00	9226		B DFK350(,@XR)	RETURN FOR CONTINUED TYPO	
				9227		*****		



DFKEYN - MATRIX PRINTER ROUTINE

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00 31/05/21 PAGE 139
					9229	*****	*****	
				274D	9230	DFKSPB EQU *	ENTRY TO HANDLE BACKSPACE	
274D	BC	10	70		9231	MVI DFKPL1+@PCTRL(,@XR),DFKACK	SET BACKSPACE CTRL	
2750	F2	80	06		9232	DFK380 JC DFK400,@NOP	JUMP IF NOT FIRST BACKSPACE	
2753	BC	11	70		9233	MVI DFKPL1+@PCTRL(,@XR),DFKKIX	SET BACKSPACE ANC INDE	
2756	BC	87	51		9234	MVI DFK380+@Q(,@XR),@UCB	SET INDEX INDR OFF	
2759	5D	01	26 28		9235	DFK400 CLC DFKSTN(@CADDR,@BR),DFKLMG(,@BR)	TEST LEFT MARGIN	
275D	F2	81	0D		9236	JE DFK420	JUMP TO NOT BACKSPACE	
2760	E2	02	70		9237	LA DFKPL1(,@XR),@XR	XR = PPL ADDRESS	
2763	D0	87	AC		9238	B DFKPRT(,@BR)	GO DO BACKSPACE	
2766	5F	01	26 21		9239	SLC DFKSTN(@CADDR,@BR),DFKC01(,@BR)	SET NEW POSITION	
276A	D0	87	96		9240	B DFKDLP(,@BR)	GO UPDATE LINE POSITION	
276D	E0	87	33		9241	DFK420 B DFKATC(,@XR)	GO TEST TYPAMATIC	
					9242	*****	*****	
				2770	9243	DFKPL1 EQU *	1ST BYTE OF BACKSPACE PPL	
2770				2770	9244	DS CL1	CONTROL BYTE	
2771	00			2771	9245	DC XL1'00'	COUNT BYTE	
					9246	*****	*****	

## DFKEYN - MATRIX PRINTER ROUTINE

ERR LOC OBJECT CODE ADDR STMT SOURCE STATEMENT VER 15, MOD 00 31/05/21 PAGE 140

```

          9248 *****
2772 78 02 1D          9249 DFK440 TBN   DFKNSK( ,@BR) ,@TYPAM      TYPO BIT ON
2775 E0 10 33          9250          BT    DFKATC( ,@XR)          YES, GO SENSE AGAIN
2778 E2 02 A3          9251 DFK460 LA    DFKPL2( ,@XR) ,@XR        XR = PPL ADDRESS
277B D0 87 AC          9252          B     DFKPRT( ,@BR)          RETURN CARRIAGE
277E 3C 1E 0000        9253 DFK480 MVI   *-*,@EOS                MOVE EOS TO CURRENT LOCATION
2782 5C 01 26 28        9254          MVC   DFKSTN-DFKBS2(@CADDR,@BR),DFKLMG( ,@BR)  SET NEW POSITION
2786 D0 87 3D          9255          B     DFK180( ,@BR)          GO EXIT LEVEL - LOCK KEYBOARD
          9256 *****
          2789 9257 DFKERA EQU    *                                ENTRY FOR ERASE DEY
2789 B4 02 A8          9258          ST    DFKPL3+@PDATA( ,@XR) ,@XR  SET PAGE ADDR IN PPL
278C AE 01 A8 B1        9259          ALC   DFKPL3+@PDATA(@CADDR,@XR),DFKMSD( ,@XR)  CALC DATA ADDR
2790 E2 02 A5          9260          LA    DFKPL3( ,@XR) ,@XR        XR = PPL ADDRESS
2793 D0 87 AC          9261          B     DFKPRT( ,@BR)          PRINT ERASED MESSAGE & RETURN
2796 5C 01 26 28        9262          MVC   DFKSTN-DFKBS2(@CADDR,@BR),DFKLMG( ,@BR)  SET NEW POSITION
279A D0 87 4A          9263          B     DFKXIT( ,@BR)          GO EXIT LEVEL
          9264 *****
          279D 9265 DFKSPA EQU    *                                ENTRY FOR SPACE BAR KEY
279D 7C 39 1C          9266          MVI   DFKATA-DFKBS2( ,@BR),DFKLNK  MOVE IN DISP OF BLANK
27A0 D0 87 71          9267          B     DFK200( ,@BR)          BRANCH TO HANDLE DATA KEYS
          9268 *****
          27A3 9269 DFKPL2 EQU    *                                ADDR OF RETURN PPL
27A3 8080          27A4 9270          DC    XL2'8080'              RETURN CARRIAGE PPL
          27A5 9271 DFKPL3 EQU    *                                FIRST BYTE 'ERASE' PPL
27A5 C0          27A5 9272          DC    XL1'C0'                  PRINT & RETURN CTRL
27A6 07          27A6 9273          DC    AL1(DFKSGL)              COUNT BYTE
27A7 0000          27A8 9274          DC    AL2(*-*)              ADDR OF MESSAGE 'ERASE'
          27A9 9275 DFKSG1 EQU    *                                START OF MESSAGE
27A9 40C5D9C1E2C5C4    27AF 9276          DC    CL7' ERASED'          MESSAGE
          0007 9277 DFKSGL EQU    *-DFKSG1                        LENGTH OF MESSAGE
27B0 00A9          27B1 9278 DFKMSD DC    AL2(DFKSG1-DFKBS3)        DISP TO ERASE MESSAGE
          9279 *****
27B2 D2 02 21          9280 DFK500 LA    DFKC01( ,@BR) ,@XR        POINT XR TO INDEX PPL
27B5 D0 87 AC          9281          B     DFKPRT( ,@BR)          INDEX A LINE
27B8 D0 87 4A          9282          B     DFKXIT( ,@BR)          GO EXIT
          9283 *****
```

DFKEYN - ERP SECTION

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	31/05/21	PAGE 141
					9285		*****			
				27BB	9286	DFKROR EQU	*			ENTRY TO ERP
27BB	F2	87	07		9287	DFK520 JC	DFK540,@UCB			JUMP IF 1ST ERROR
27BE	3A	20	03D2		9288	SBN	\$IOIND,\$HRDER			SET HARD ERROR INDR
27C2	E0	87	7E		9289	B	DFK480(,@XR)			GO EXIT - HARD ERROR
					9290	*				
27C5	1C	07	0435 21		9291	DFK540 MVC	\$HIST1(#HISLN),DFKIST(,@BR)			SET UP HISTORY ENTRY
27CA	BC	80	BC		9292	MVI	DFK520+@Q(,@XR),@NOP			SET PARITY INDR
27CD	F0	00	00		9293	HPL	*-*,*-*			WAIT ON FIRST ERROR
27CE					9294	ORG	*-2			PLACE ERROR CODE
27CE	2040			27CF	9295	DC	AL2(@HKBER)			WAIT CODE
27D0	3A	04	03D5		9296	SBN	\$INDR2,\$ERPND			SET ERROR PENDING INDR
27D4	D0	87	4A		9297	B	DFKXIT(,@BR)			GO RETRY CHARACTER
					9298		*****			

## DFPRNT - MATRIX PRINTER ROUTINE

ERR LOC OBJECT CODE ADDR STMT SOURCE STATEMENT VER 15, MOD 00 31/05/21 PAGE 142

```

9300 *****
9301 * THIS IOCR IS USED FOR ALL MATRIX PRINTER FUNCTIONS. *
9302 * IT IS ALSO USED BY DLFPRT 'LINE PRINTER ROUTINE' FOR IOCR OPERATION *
9303 * AVAILABLE FUNCTIONS INCLUDE... *
9304 * PRINT ONLY *
9305 * PRINT AND RETURN CARRIAGE *
9306 * RETURN CARRIAGE ONLY *
9307 * BACKSPACE *
9308 * INDEX AND BACKSPACE *
9309 * CHANGES TO DFPRNT MAY DIRECTLY AFFECT IT'S INTERFACE WITH DLFPRT *
9310 *****
2800 9311 ORG *,256,0
2800 9312 USING DFPA SE,@BR SET BASE REG
2800 9313 DFP100 EQU * ENTRY TO PRINTER IOCR
2800 1C 01 144A FD 9314 MVC I$VADR,DFPPCH(@CADDR,@BR) VM PATCH PAGE ENTRY ADDR 1-5
2805 C0 87 1358 9315 DFP100 B I$CVAD LOAD PATCH PAGE 1-5
2809 4C 01 11 144C 9316 MVC DFP101+@OP1(@CADDR,@BR),I$CADR MOVE CADDR TO BRANCH 1-5
280E C0 87 0000 9317 DFP101 B *-* BRANCH TO PATCH PAGE 1-5
9318 * 1-5
2812 1C 01 144A FD 9319 DFP102 MVC I$VADR,DFPPCH(@CADDR,@BR) VM PATCH PAGE ENTRY ADDR 1-5
2817 3C 39 144A 9320 MVI I$VADR,DFPX39 ADD DISP X'39' 1-5
281B D0 87 05 9321 B DFP100(,@BR) BRANCH TO LOAD PAGE 1-5
281E 4D00 281F 9322 DFP105 DC AL(@VADDR)(V$LPRT) LINE PRINTER PAGE
2820 E0 87 00 9323 B 0(,@XR) BRANCH TO LINE PRINTER ROUTINE
2823 F1 E2 00 2823 9324 DFP115 EQU * MATRIX PRINTER ROUTINE
2826 78 40 F5 9325 APL @PBUSY WAIT FOR PRINTER NOT BUSY 1-4
2829 F2 10 11 9326 TBN DFPIST+@PCTRL(,@BR),@PRINT DOE THIS OP PRINT
282C 7C 00 F6 9327 JT DFP120 JUMP IF YES
282F 78 10 DE 9328 MVI DFPIST+@PRCNT(,@BR),@ZERO SET PPL CNTR BYTE TO ZERO
2832 F2 90 3D 9329 TBN DFPPCF+@PCTRL(,@BR),@TBLEF TAB LEFT OPERATION ?
2835 1F 00 03C2 E7 9330 JF DFP180 GO DO OP IF NOT
283A F2 87 55 9331 SLC $PRPOS(1),DFP001(,@BR) SET NEW CURRENT POSITION
9332 J DFP240 GO DO OP
9333 *
9334 * PRINTING IS REQUIRED - SET UP PRINT PCF
9335 *
283D 71 E4 F8 9336 DFP120 LIO DFPIST+@PDATA(,@BR),@PDAR LOAD DATA LSR WITH DATA ADDR
2840 4E 00 F6 03C2 9337 ALC DFPIST+@PRCNT(1,@BR),$PRPOS ADD CURRENT POSITION
2845 4F 00 F6 03C0 9338 SLC DFPIST+@PRCNT(1,@BR),$RMGRN SUBTRACT RIGHT MARGIN VALUE
284A F2 84 06 9339 JH DFP140 JUMP IF RIGHT MARGIN HIT
284D 7C 00 F6 9340 MVI DFPIST+@PRCNT(,@BR),@ZERO SET COUNT BYTE TO ZERO
2850 F2 87 0F 9341 J DFP160 GO SET NEW PRINT POSITION
2853 5F 00 DF F6 9342 DFP140 SLC DFPPCF+@PRCNT(1,@BR),DFPIST+@PRCNT(,@BR) SET CNT TO HIT
9343 * * MARGIN
2857 7A 80 DE 9344 SBN DFPPCF+@PCTRL(,@BR),@RETRN SET CARRIAGE TO RETURN
285A 5C 00 E5 DF 9345 MVC DFPORK(1,@BR),DFPPCF+@PRCNT(,@BR) RIGHT JUSTIFY CNT
285E 5E 01 F8 E5 9346 ALC DFPIST+@PDATA(@CADDR,@BR),DFPORK(,@BR) ADD CNT TO DATA
9347 * * ADDRESS IN LIST
2862 1E 00 03C2 DF 9348 DFP160 ALC $PRPOS(1),DFPPCF+@PRCNT(,@BR) UPDATE HEAD POSITION
2867 5F 00 DF E7 9349 SLC DFPPCF+@PRCNT(1,@BR),DFP001(,@BR) SET PCF CNT = CNT-1...
9350 * * THIS IS HARDWARE REQUIREMENT
286B F2 02 04 9351 JNL DFP180 JUMP IF SOMETHING TO PRINT
286E 5C 01 DF E9 9352 MVC DFPPCF+@PRCNT(2,@BR),DFPETN(,@BR) SET CARRIER RTRN ONLY
2872 78 80 DE 9353 DFP180 TBN DFPPCF+@PCTRL(,@BR),@RETRN OP FOR CARRIAGE RETURN
2875 F2 90 1A 9354 JF DFP240 JUMP IF NO
2878 4C 00 E1 03C2 9355 DFP200 MVC DFPPCF+@RTCNT(1,@BR),$PRPOS SET CURRENT POS IN

```

DFPRNT - MATRIX PRINTER ROUTINE

ERR LOC		OBJECT CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00		31/05/21	PAGE 143
				9356	*					
						* CARRIAGE RETURN CNT				
287D	4F	00 E1 03C1		9357	SLC	DFPPCF+@RTCNT(1,@BR), \$LMRGN				SUBTRACT LEFT MARGIN VALUE
2882	F2	84 03		9358	JH	DFP220				JUMP IF NO
2885	7C	01 DE		9359	MVI	DFPPCF+@PCTRL(,@BR),@INDEX				SET OP TO INDEY ONLY
2888	0C	00 03C2 03C1		9360	DFP220 MVC	\$PRPOS(1), \$LMRGN				SET CURRENT POS TO LEFT MARGIN
288E	5F	00 E1 E7		9361	SLC	DFPPCF+@RTCNT(1,@BR), DFP001(,@BR)				SET HARDWARE COUNT
2892	74	01 DD		9362	DFP240 ST	DFPAPC(,@BR),@BR				SET PAGE ADDR IN PCF ADDR BYTE
2895	5E	01 DD EB		9363	ALC	DFPAPC(@CADDR,@BR), DFPCFD(,@BR)				ADD DISP TO GET TRUE ADDR
			2899	9364	DFP250 EQU	*				LINE PRINTER I/O ENTRY 1-4
2899	71	E6 DD		9365	LIO	DFPAPC(,@BR),@PCAR				LOAD CONTROL LSR WITH NORMAL PCF
289C	F3	E0 00		9366	DFP260 SIO	@PSIOR,@PSIOQ				START THE PRINT OPERATION
289F	E0	00 B3		9367	DFP270 BC	RETURN-DLFPRT(,@XR), *-*				RETURN TO LINE PRINTER RTN. 1-4
28A0				9368	ORG	DFP270+@Q				* INITIALIZE 1-4
28A0	80		28A0	9369	DC	AL1(@NOP)				* TO NOT BRANCH 1-4
28A2				9370	ORG	DFP270+@INST3				* TO LINE PRINTER RTN. 1-4
28A2	F2	80 07		9371	DFP280 JC	DFP320,@NOP				JUMP TO ERP IF ERP IN PROCESS
				9372	*					
				9373	*****					

## DFPRNT - MATRIX PRINTER ROUTINE

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	31/05/21	PAGE 144
					9375	*****	*****			
					9376	*	THIS ROUTINE WAITS FOR THE OPERATION TO COMPLETE AND CHECKS			*
					9377	*	FOR ERRORS. FORMS CHECKS WILL CAUSE A SOFT HALT.			*
					9378	*	UNIT CHECKS WILL CAUSE ENTRY TO THE ERP.			*
					9379	*****	*****			
				28A5	9380	DFPRCK EQU	*			ENTRY TO CHECK FOR ERRORS
28A5	5C	01	ED	EE	9381	MVC	DFPRCT(DFPRCL,@BR),DFPERC(@BR)			INITILIZE RETRY COUNTER
28A9	7C	87	A3		9382	MVI	DFP280+@Q-DFPASE(@BR),@UCB			SET ERP IN PROCESS INDR
28AC	F1	E2	00		9383	DFP320 APL	@PBUSY			WAIT FOR NOT BUSY
28AF	7C	00	9E		9384	MVI	DFP260+2(@BR),@ZERO			SET MATRIX PRINT
28B2	D1	E1	CD		9385	DFP340 TIO	DFP360(@BR),@PFORM			TEST FOR END OF FORMS
28B5	71	E2	E3		9386	LIO	DFPOFF(@BR),@PLITE			TURN END OF FORMS LAMP OFF
28B8	D1	E0	00		9387	DFP335 TIO	*-*(@BR),@PERR			BRANCH TO ERP IF UNIT CHECK 1-4
28B8					9388	ORG	DFP335			* INITIALIZE DFP335 1-4
28B8	E1	E0	CD		9389	TIO	DLFRPE-DLFPRT(@XR),@PERR			* TO BRANCH TO 1-4
28B8					9390	ORG	DFP335			* DFP335 1-4
28B8	D1	E0	D3		9391	TIO	DFPRPE(@BR),@PERR			* ENTRY TO LOAD ERP SECTION 1-4
				28BA	9392	DFP333 EQU	*-1			LAST BYTE OF TIO INST. 1-4
28BB	E0	00	00		9393	DFP330 BC	*-*(@XR),*-*			BRANCH TO LINE PRINTER RTN. 1-4
28BC					9394	ORG	DFP330+@Q			* INITIALIZE 1-4
28BC	80			28BC	9395	DC	AL1(@NOP)			* TO NOT BRANCH 1-4
28BD					9396	ORG	DFP330+@D1			* INITIALIZE FOR 1-4
28BD	25			28BD	9397	DC	AL1(DLF100-DLFPRT)			* RETURN TO DLFPRT ENTRY 1-4
28BE					9398	ORG	DFP330+@INST3			* TO LINE PRINTER ROUTINE 1-4
28BE	1C	01	144A	FD	9399	MVC	I\$VADR,DFPPCH(@VADDR,@BR)			VM PATCH PAGE 1-5
28C3	3C	00	144A		9400	MVI	I\$VADR,@ZERO			SET DISP = 0 1-5
28C7	D0	87	05		9401	B	DFP100(@BR)			BRANCH TO LOAD PAGE 1-5
28CA	D0	87	12		9402	DFP300 B	DFP102(@BR)			BRANCH TO LOAD PATCH PAGE 1-5
					9403	*				
					9404	*****	*****			
					9405	*				
28CD	71	E2	E7		9406	DFP360 LIO	DFPITE(@BR),@PLITE			TURN ON FORMS INDR LAMP
28D0	D0	87	B2		9407	B	DFP340(@BR)			GO TEST FORMS AGAIN
					9408	*				
				28D3	9409	DFPRPE EQU	*			ENTRY TO LOAD ERP SECTUIN
28D3	C0	87	1330		9410	B	I\$LDXR			LOAD ERP PAGE USING XR
28D7	2900			28D8	9411	DC	AL2(V\$SPRT+DFPNDX-DFPRNT)			PRINTER ERROR IOCR VADDR
28D9	E0	87	00		9412	B	0(@XR)			EXECUTE ERP
					9413	*****	*****			

## DFPRNT - MATRIX PRINTER ROUTINE

ERR LOC OBJECT CODE ADDR STMT SOURCE STATEMENT VER 15, MOD 00 31/05/21 PAGE 145

```

          9415 *****
          9416 * CONSTANT AND EQUATE AREA FOR DPRINT *
          9417 *****
28DC      2800 9418 DFPASE EQU DFPRNT BASE VALUE FOR CALL SECTION
          0002 9419 DFPRCL EQU 2 NUMBER OF RETRY COUNTERS
          28DD 9420 DFPAPC DS CL(@CADDR) ADDRESS OF NRML PCF
          28DE 9421 DFPPCF EQU * LEFT BYTE OF PCF
28DE      28DF 9422 DS CL2 CTRL AND CNT BYTES
28E0 11 28E0 9423 DC XL1'11' RETURN CARRIAGE INDEX CMND
28E1      28E2 9424 DS CL2 COUNT & INDEX
          28E2 9425 DFPPCO EQU *-1 LAST BYTE OF CCF
28E3 00 28E3 9426 DFPOFF DC XL1'00' TURN OFF INDR LAMP CTRL
28E4 0000 28E5 9427 DFPORK DC XL2'0000' WORK AREA
28E6 0001 28E7 9428 DFP001 DC XL2'0001' CONSTANT OF ONE
28E8 8080 28E9 9429 DFPETN DC 2AL1(@RETRN) CARRIER RETURN CTRL
28EA 00DE 28EB 9430 DFPCFD DC AL2(DFPPCF-DFPASE) DISPLACEMENT OF PCF IN PAGE
          9431 *
28EC      28ED 9432 DFPRCT DS CL(DFPRCL) ERROR COUNT
28EE 03 28EE 9433 DFPERC DC XL1'03' RETRY COUNT
28EF 00F9 28F0 9434 DFPYCD DC AL2(DFPSYC-DFPASE) DISPLACEMENT OF SYNC PCF IN PAGE
28F1 00000000 28F4 9435 DFPDSV DC XL4'00' SAVE AREA FOR CNT AND DATA ADDR
          28F5 9436 DFPIST EQU *
28F5      28F8 9437 DS CL4 PRINT PARAMETER LIST (PPL)
28F5      9438 ORG DFPIST RESET INSTR CNTR
28F5 00000000 28F8 9439 DC XL4'00' SET INITIAL LIST TO ZERO
          28F9 9440 DFPSYC EQU * LEFT BYTE OF SYNC CHECK PCF
28F9 0520 28FA 9441 DC XL2'0520' RETURN AND INDEX, TAB RIGHT
28FB      28FB 9442 DS CL1
28FC 5309 28FD 9443 DFPPCH DC AL2(V$PCH2+DFP100-DFPASE+@DOP2) PATCH PAGE 2 1-5
          0039 9444 DFPX39 EQU X'39' DISP = X'39' 1-5
          28E7 9445 DFPITE EQU DFP001 FORMS INDR LIGHT CTRL
          0001 9446 DFPYCT EQU 1 DISPLACEMENT CYNK CK CNTR
          9447 *
          9448 * THE FOLLOWING EQUATES ARE FOR THE LINE PRINTER MODULE (DLFPRT)
          9449 *
          28F5 9450 DLFIST EQU DFPIST
          28E5 9451 DLFORK EQU DFPORK
          28F4 9452 DLFDSV EQU DFPDSV
          28E7 9453 DLF001 EQU DFP001
          28DE 9454 DLFPCF EQU DFPPCF

```



## DFPRNT - MATRIX PRINTER ROUTINE

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	31/05/21	PAGE 146
					9456		*****			
					9457		* THIS ROUTINE DETERMINES THE ERROR AND BRANCHES TO THE PROPER ERP			*
					9458		*****			
2900					9459		ORG *,256,0			
				2800	9460		USING DFPASE,@BR			SET BASE REGS
				2900	9461		USING DFPNDX,@XR			
				2900	9462	DFPNDX	EQU *			ENTRY TO ERP SECTION
2900	38	01	03E4		9463		TBN \$LPRP3,@INDEX			TEST DUMMY PRINT POS. USED 1-3
2904	F2	90	0A		9464		JF DFP378			JUMP NO
2907	0C	00	03C2 03E5		9465		MVC \$PRPOS(1),\$LPROS			RESTORE CORRECT POSITION
290D	3B	01	03E4		9466		SBF \$LPRP3,@INDEX			RESET DUMMY POS. FLAG 1-3
				2911	9467	DFP378	EQU *			ENTRY SENSE ERROR
2911	B0	E2	D9		9468		SNS DFPRSN(,@XR),@PSNSQ			SENSE ERROR BYTES
2914	38	04	03D5		9469		TBN \$INDR2,\$ERPND			HAS LOG ENTRY BEEN SET UP
2918	F2	10	0C		9470		JT DFP380			JUMP IF YES
291B	2C	07	0435 DD		9471		MVC \$HIST1(#HISLN),DFPOGE(,@XR)			MOVE LOG TO NUCLEUS
2920	3A	04	03D5		9472		SBN \$INDR2,\$ERPND			SET ENTRY PENDING INDR
2924	F0	00	00		9473		HPL *-*,*-*			SOFT HALT ON INITIAL ERROR
2925					9474		ORG *-2			PLACE HALT CODE
2925	0070			2926	9475		DC AL2(@HPRER)			DISPLAY CODE '123'
2927	1E	00	0434 E7		9476	DFP380	ALC \$HISTE+@HSTPE(1),DFP001(,@BR)			ADD ONE TO RETRY COUNTER
292C	B8	20	D9		9477		TBN DFPRSN(,@XR),@PMGCK			MARGIN CHECK
292F	F2	10	07		9478		JT DFPMCK			JUMP IF YES
				2932	9479	DFPSCK	EQU *			ENTRY FOR SYNC CHK.
					9480		*			
					9481		* LINE PRINTER MODE ONLY			
					9482		*			
2932	38	40	03E4		9483		TBN \$LPRP3,@PRINT			LINE PRINTER ERROR 1-3
2936	F2	90	0F		9484		JF DFPSCK			JUMP IF NOT PRINT OP
				2939	9485	DFPMCK	EQU *			ENTRY FOR MARGIN CHECK
2939	5F	00	ED E7		9486		SLC DFPRCT-DFPGCT(1,@BR),DFP001(,@BR)			DECREMENT RETRY CNT
293D	F2	81	72		9487		JZ DFP400			JUMP IF NO MORE RETRIES
2940	4C	00	FB 03C1		9488		MVC DFPSYC+@SYCNT(1,@BR),\$LMRGN			SET CNT TO HARD LEFT MARGIN
2945	F2	87	0B		9489		J DFP420			GO DO FIRST PART OF SYNC CHK
				2948	9490	DFPSC2	EQU *			
2948	5F	00	EC E7		9491		SLC DFPRCT-DFPYCT(1,@BR),DFP001(,@BR)			DECREMENT CYNC CNT
294C	F2	81	63		9492		JZ DFP400			JUMP IF NO MORE TRYs
294F	5C	02	F8 F4		9493		MVC DFPIST+@PDATA(@CADDR+1,@BR),DFPDSV(,@BR)			RESTORE ORIGINAL
					9494		*			* COUNT AND DATA ADDR
2953	B4	01	D5		9495	DFP420	ST DFPASY(,@XR),@BR			SET PAGE ADDR IN PCF ADDR
2956	9E	01	D5 F0		9496		ALC DFPASY(@CADDR,@XR),DFPYCD(,@BR)			CALC PCF ADDR
295A	B1	E6	D5		9497		LIO DFPASY(,@XR),@PCAR			LOAD CONTROL LSR WITH SYNC SCF
295D	7A	80	F9		9498		SBN DFPSYC+@PCTRL(,@BR),@RETRN			SET CHAIN BIT ON
2960	1C	00	03C2 FB		9499		MVC \$PRPOS(1),DFPSYC+@SYCNT(,@BR)			SET UP NEW HEAD POSITION
2965	5F	00	FB E7		9500		SLC DFPSYC+@SYCNT(1,@BR),DFP001(,@BR)			SUBTRACT 1
2969	F2	02	03		9501		JNL DFP440			JUMP IF NOT NEG
296C	7B	80	F9		9502		SBF DFPSYC+@PCTRL(,@BR),@RETRN			SET CHAIN BIT OFF
296F	38	40	03E4		9503	DFP440	TBN \$LPRP3,@PRINT			CHECK IF ENTRY FROM LINE PTR 1-3
2973	F2	90	39		9504		JF DLF450			JUMP NOT
2976	3A	01	03E4		9505		SBN \$LPRP3,@INDEX			SET DUMMY PRINT POS. FLAG 1-3
297A	0C	00	03E5 03C2		9506		MVC \$LPROS(1),\$PRPOS			SET LINE PRINTER PRINT POSITION
2980	6C	00	BD D3		9507		MVC DFP330+@D1(1,@BR),DFPEXT(,@XR)			SET DLRPRT ERROR ENTRY 1-4
2984	2C	01	144A D2		9508		MVC ISVADR,DFPLBU(2,@XR)			GET LINE PRINTER BUFFER ADDR 1-4
2989	C0	87	1354		9509		B ISLOCK			GET LINE PRINTER BUFFER 1-4
298D	4C	01	E5 144C		9510		MVC DLFORK(2,@BR),ISCADR			SAVE BUFFER CADDR ADDR 1-4
2992	C0	87	1330		9511		B ISLDXR			

## DFPRNT - MATRIX PRINTER ROUTINE

ERR LOC OBJECT CODE ADDR STMT SOURCE STATEMENT VER 15, MOD 00 31/05/21 PAGE 147

```

2996 4D00          2997 9512      DC      AL(@VADDR)(V$LPRT)      LINE PRINTER PAGE
2998 B9 04 D9      9513      TBF      DFPRSN(,@XR),DFPVCK      TEST VERTICLE CYCLE CHECK      1-4
299B F2 90 11      9514      JF       DLF450      IF VERTICAL CYCLE CHK      1-4
299E 9C 01 DE E5      9515      MVC      BUFRWK-DLFPRT(2,@XR),DLFORK(,@BR)  GET BUFFER ADDR      1-4
29A2 8C 01 DF 03EA      9516      MVC      DLFBPT-DLFPRT(2,@XR),$LPRIO  RESTORE BUF PTR & PDAR      1-4
29A7 2C 00 03E3 DF      9517      MVC      $BUFPT(1),DLFBPT-DLFPRT(,@XR)  RESTORE BUFFER POINTER      1-4
29AC BC 80 89      9518      MVI      DLF350-DLFPRT+@Q(,@XR),@NOP    FORCE ERROR CHECK
29AF D0 87 9C      29AF 9519 DLF450 EQU      *
29AF D0 87 9C      9520      B       DFP260(,@BR)      GO TO MATRIX PRINTER

9522 *****
9523 * MATRIX PRINTER HARD FAILURE ROUTINE *
9524 *****
29B2 3A 21 03D2      9525 DFP400 SBN      $IOIND,$MPDWN+$HRDER      SET MAT4IX PRINTER DOWN INDR
29B6 3C 00 0434      9526      MVI      $HISTE+@HSTPE,@ZERO      SET HARD ERROR INDR
29BA 38 40 03E4      9527      TBN      $LPRP3,@PRINT      ENTRY FROM LINE PTR.      1-3
29BE F2 90 0D      9528      JF       DFP480      JUMP IF NOT
29C1 C0 87 1330      9529      B       I$LDXR      LOAD PAGE
29C5 4D00          29C6 9530      DC      AL2(V$LPRT)      LINE PRINTER PAGE
29C7 3C 00 03E3      9531      MVI      $BUFPT,@ZERO      RESET LINE PTR. BUFFER PTR.      1-3
29CB E0 87 B3      9532      B       RETURN-DLFPRT(,@XR)      GO TO LINE PRINTER PAGE
29CE D0 87 CA      29CE 9533 DFP480 EQU      *
29CE D0 87 CA      9534      B       DFP300(,@BR)      RETURN TO MATRIX PRINTER
29CE D0 87 CA      9535 *****
29D1 4F00          29D2 9536 DFPLBU DC      AL2(V$LPRB)      LINE PRINTER BUFFER VADDR      1-4
29D3 88          29D3 9537 DFPEXT DC      AL1(DLF350-DLFPRT)      DISPLACEMENT TO DLFPRT ERROR      1-4
29D4          29D5 9538 DFPASY DS      CL(@CADDR)      ADDR OF ERP PCF
29D6 E0          29D6 9539      DC      AL1(@PSIOQ)      HISTORY LOG SIO Q BYTE
29D7 00          29D7 9540 DFPIOR DC      AL1(@PSIOR)      HISTORY LOG SIO R BYTE
29D8          29D9 9541 DFPRSN DS      CL2      ERROR SENSE BYTES
29DA 00000001      29DD 9542 DFPERR DC      XL4'00000001'      ERROR INFO
29DA 00000001      29DD 9543 DFPOGE EQU      *-1      LAST BYTE OF HISTORY LOG
29DA 00000001      0000 9544 DFPGCT EQU      0      DISPLACEMENT MARGIN CK CNT
29DA 00000001      0004 9545 DFPVCK EQU      X'04'      PRINTER VERTICAL CYCLE CK.      1-4
29DA 00000001      9546 *****
29DA 00000001      9547 *##### X'29FF' IMG_0188 #####
29DA 00000001      9548 * NOT YET SCANNED OR OBJ CHECKED !!
29DA 00000001      9549 *          96 COLUMN CARD READER / PUNCHER
29DA 00000001      9550 *##### X'2AFF' #####
2AFD          9551      ORG      X'2AFD'
2AFD          9552 *****
2AFD          9553 * 5703-XM1 COPYRIGHT IBM CORP. 1970 *
2AFD          9554 *          REFER TO INSTRUCTIONS ON COPY RIGHT NOTICE, 120-2083 *
2AFD          9555 * *
2AFD          9556 *****
2AFD          9557 *STATUS *
2AFD          9558 *   VERSION 1 MODIFICATION 0 *
2AFD          9559 * *
2AFD          9560 *FUNCTION *
2AFD          9561 *   * FZXINP EXECUTION CAUSES KEYBOARD DATA ENTRY TO BE ENABLED *
2AFD          9562 *   DURING PROGRAM OPERATION. ENTERED DATA ARE SYNTAX CHECKED WITH *
2AFD          9563 *   RESPECT TO FORM AND TYPE, AND VALID ELEMENTS ARE CONVERTED TO *
2AFD          9564 *   INTERNAL FORMAT AND PLACED IN THE RUN-TIME STACK ON AN INDI- *
2AFD          9565 *   VIDUAL BASIS. *
2AFD          9566 *   * THIS ROUTINE PERFORMS THE PRIMARY FUNCTION OF SUPPORTING THE *
2AFD          9567 *   EXECUTION OF BASIC PROGRAM 'INPUT' STATEMENTS. ON A SECONDARY *

```

## DFPRNT - MATRIX PRINTER ROUTINE

ERR LOC OBJECT CODE ADDR STMT SOURCE STATEMENT VER 15, MOD 00 31/05/21 PAGE 148

```

9568 * LEVEL, THE MESSAGE PRINTING, SYNTAX CHECKING, AND DATA CONVER- *
9569 * SION FACILITIES REQUIRED FOR 'INPUT' ARE ORGANIZED FOR ALTER- *
9570 * NATE USE DURING THE EXECUTION OF 'MAT INPUT' AND 'GET' (CARD) *
9571 * BASIC STATEMENTS. *
9572 * * TWO ENTRY POINTS ARE PROVIDED FOR 'INPUT' MODE OPERATIONS. THE *
9573 * FIRST (FZXIP1) OPERATES IN CONJUNCTION WITH STACKED DATA TYPE *
9574 * CODES AND A COUNT PARAMETER IN I$PARM TO ALLOW KEYBOARD DATA *
9575 * INPUT AND DATA LINE VALIDITY CHECKING. THE SECOND ENTRY POINT *
9576 * (FZXIP2) OPERATES ON THE VALIDITY-CHECKED DATA LINE TO CONVERT *
9577 * AND STACK SEQUENTIALLY OCCURRING DATA ELEMENTS. *
9578 * * SIX ALTERNATE ENTRY POINTS ARE PROVIDED FOR USE WITH 'MAT INPUT' *
9579 * AND 'GET' (CARD) OPERATIONS. *
9580 * * ENTRY POINTS FZXPQ1, FZXPQ2, AND FZXPEM ARE USED TO PRINT *
9581 * QUESTION MARK(S) OR ERROR MESSAGES ON THE SYSTEM PRINT *
9582 * DEVICE. *
9583 * * ENTRY POINT FZXGCS IS USED TO SYNTAX CHECK AN ENTIRE 'GET' *
9584 * (CARD) INPUT LINE (INTO WHICH COMMA DELIMITERS HAVE BEEN *
9585 * INSERTED WERE NOT ORIGINALLY EXISTENT). *
9586 * * ENTRY POINT FZXMIS IS USED TO VALIDITY CHECK A PARTIAL OR *
9587 * ENTIRE ARRAY ROW. *
9588 * * ENTRY POINT FZXCNV IS USED TO CONVERT AND STACK INDIVIDUAL *
9589 * INPUT LINE ELEMENTS AFTER THE LINE HAS BEEN SYNTAX OR *
9590 * VALIDITY CHECKED. *
9591 * *
9592 * ENTRY POINTS *
9593 * * ENTRY FZXIP1 - FOR ENABLING 'INPUT' KEYBOARD DATA ENTRY AND *
9594 * SYNTAX CHECKING THE RESULTING DARA LINE. CALLING SEQUENCE IS *
9595 * B I$CALL *
9596 * DC AL2(V$XKIN) *
9597 * WHERE THE ADDRESS CONSTANT PARAMETER DEFINES THE VIRTUAL *
9598 * ADDRESS OF ENTRY POINT FZXIP1. *
9599 * * ENTRY FZXIP2 - FOR CONVERTING AND STACKING A SINGLE DATA ELE- *
9600 * MENT FROM THE 'INPUT' DATA LINE. CALLING SEQUENCE IS *
9601 * B I$CALL *
9602 * DC AL2(V$XSIN) *
9603 * WHERE THE ADDRESS CONSTANT PARAMETER DEFINES THE VIRTUAL *
9604 * ADDRESS OF ENTRY POINT FZXIP2. *
9605 * * ENTRY FZXPQ1 - FOR PRINTING A SINGLE QUESTION MARK (?) ON THE *
9606 * SYSTEM PRINT DEVICE, CALLING SEQUENCE IS *
9607 * B I$CALL *
9608 * DC AL2(FZXPQ1) *
9609 * WHERE THE ADDRESS CONSTANT PARAMETER DEFINES THE VIRTUAL *
9610 * ADDRESS OF ENTRY POINT FZXPQ1. *
9611 * * ENTRY FZXPQ1 - FOR PRINTING A DOUBLE QUESTION MARK (??) ON THE *
9612 * SYSTEM PRINT DEVICE, CALLING SEQUENCE IS *
9613 * B I$CALL *
9614 * DC AL2(FZXPQ2) *
9615 * WHERE THE ADDRESS CONSTANT PARAMETER DEFINES THE VIRTUAL *
9616 * ADDRESS OF ENTRY POINT FZXPQ2. *
9617 * * ENTRY FZXPEM - FOR PRINTING A DATA INPUT ERROR MESSAGE ON THE *
9618 * SYSTEM PRINT DEVICE. CALLING SEQUENCE IS *
9619 * B I$CALL *
9620 * DC AL2(FZXPEM) *
9621 * WHERE THE ADDRESS CONSTANT PARAMETER DEFINES THE VIRTAL *
9622 * ADDRESS OF ENTRY POINT FZXPEM. *
9623 * * ENTRY FZXGCS - FOR SYNTAX CHECKING A DATA LINE RESULTING FROM *

```

## DFPRNT - MATRIX PRINTER ROUTINE

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 31/05/21 PAGE 149
		9624	*	A 'GET' USING A CARD INPUT FILE. CALLING SEQUENCE IS	*
		9625	*	B I\$CALL	*
		9626	*	DC AL2(V\$CDSY)	*
		9627	*	WHERE THE ADDRESS CONSTANT PARAMETER DEFINES THE VIRTUAL	*
		9628	*	ADDRESS OF ENTRY POINT FZXGCS.	*
		9629	*	* ENTRY FZXMIS - FOR SYNTAX CHECKING A DATA LINE RESULTING FROM	*
		9630	*	A 'MAT INPUT' ARRAY ROW INPUT. CALLING SEQUENCE IS	*
		9631	*	B I\$CALL	*
		9632	*	DC AL2(FZXMIS)	*
		9633	*	WHERE THE ADDRESS CONSTANT PARAMETER DEFINES THE VIRTUAL	*
		9634	*	ADDRESS OF ENTRY POINT FZXMIS.	*
		9635	*	* ENTRY FZXCNV - FOR CONVERTING AND STACKING A SINGLE DATA ELE-	*
		9636	*	MENT FROM A 'GET' (CARD) OR 'MAT INPUT' DATA LINE.	*
		9637	*	CALLING SEQUENCE IS	*
		9638	*	B I\$CALL	*
		9639	*	DC AL2(V\$CDCV)	*
		9640	*	WHERE THE ADDRESS CONSTANT PARAMETER DEFINES THE VIRTUAL	*
		9641	*	ADDRESS OF ENTRY POINT FZXCNV.	*
		9642	*	* ENTRY POINT EXECUTION FOR THESE OPERATIONS IS SUBJECT TO INPUT	*
		9643	*	CONDITIONS DESCRIBED BELOW.	*
		9644	*		*
		9645	*	*INPUT	*
		9646	*	* I\$STAK - 2 BYTES, FOR THE RUN-TIME STACK POINTER.	*
		9647	*	* FOR ENTRY FZXIPI, THIS CONTAINS THE CORE ADDRESS OF THE	*
		9648	*	LEFTMOST BYTE OF THE STACKED NATA TYPE SPECIFICATION CODES.	*
		9649	*	* FOR ENTRY FZXIP2, THIS CONTAINS THE CORE ADDRESS OF THE	*
		9650	*	FIRST AVAILABLE STACK LOCATION.	*
		9651	*	* FOR ENTRY FZXMIS, THIS CONTAINS THE CORE ADDRESS OF THE	*
		9652	*	SINGLE STACKED DATA TYPE SPECIFICATION CODE.	*
		9653	*	* FOR ENTRY FZXCNV, THIS CONTAINS THE CORE ADDRESS OF THE	*
		9654	*	STACK LOCATION INTO WHICH THE CONVERTED DATA ELEMENT IS	*
		9655	*	TO BE STACKED.	*
		9656	*	* I\$PARM - 2 BYTES, FOR THE INTERPRETER COMMUNICATIONS PARAMETER.	*
		9657	*	* FOR ENTRY FZXIPI, THE RIGHT BYTE IN I\$PARM CONTAINS A	*
		9658	*	COUNT OF THE STACKED DATA TYPE SPECIFICATION CODES.	*
		9659	*	* FOR ENTRY FZXMIS, THE RIGHT BYTE IN I\$PARM CONTAINS A	*
		9660	*	VALUE OF '1' FOR THE SINGLE STACKED DATA TYPE SPEC CODE.	*
		9661	*	* I\$PUB1 - 2 BYTES, FOR THE DATA BUFFER CORE ADDRESS. FOR ENTRY	*
		9662	*	FZXIP2 ONLY, THIS CONTAINS THE CORE ADDRESS OF THE 'INPUT' DATA	*
		9663	*	BUFFER LEFTMOST BYTE.	*
		9664	*	* I\$ERRC - 1 BYTE, FOR THE INTERPRETER ERROR CONDITION CODE, FOR	*
		9665	*	ENTRY FZXPEM ONLY, THIS CONTAINS AN ERROR CODE IN THE DECIMAL	*
		9666	*	NUMBER RANGE X'F0' THROUGH X'F4', INDICATING OUTPUT OF AN ERROR	*
		9667	*	MESSAGE IN THE RANGE 800 THROUGH 804 (SYSTEM ERROR MESSAGE	*
		9668	*	NUMBERS) RESPECTIVELY.	*
		9669	*	* REGISTER @XR - FOR ENTRIES FZXGCS, FZXMIS AND FZXCNV, THIS	*
		9670	*	CONTAINS THE CORE ADDRESS OF THE DATA BUFFER LEFTMOST BYTE.	*
		9671	*	* RUN-TIME STACK - FOR ENTRIES FZXIPI AND FZXMIS, THIS CONTAINS	*
		9672	*	DATA TYPE SPECIFICATION CODES BEGINNING AT THE CORE ADDRESS	*
		9673	*	REFERENCED BY I\$STAK.	*
		9674	*	* KEYBOARD DATA ENTRY - FOR ENTRY FZXIPI ONLY, NUMERIC, SIGNED	*
		9675	*	INTERNAL (-&PI, ETC.), AND CHARACTER CONSTANTS ENCLOSED IN	*
		9676	*	QUOTES ARE ENTERED FROM THE SYSTEM CONSOLE KEYBOARD IN COMPLI-	*
		9677	*	ANCE WITH THE CURRENT 'INPUT' STATEMENT ASSIGNMENT LIST.	*
		9678	*	* INPUT DATA BUFFER - 256 BYTES, FOR DATA ELEMENT STORAGE DURING	*
		9679	*	SYNTAX CHECKING AND CONVERSION.	*



## DFPRNT - MATRIX PRINTER ROUTINE

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00	31/05/21	PAGE 150
9680	*		*	FOR ENTRIES FZXGCS AND FZXMIS, THIS BUFFER CONTAINS THE	*		
9681	*			DATA INPUT LINE (AS ENTERED) BEGINNING AT THE LEFTMOST	*		
9682	*			BUFFER BYTE.	*		
9683	*		*	FOR ENTRIES FZXIP2 AND FZXCNV, THIS BUFFER CONTAINS, IN	*		
9684	*			ADDITION TO THE ORIGINAL DATA, A 2-BYTE FIELD IN THE LAST	*		
9685	*			TWO BUFFER BYTES. THIS FIELD CONTAINS THE CORE ADDRESS OF	*		
9686	*			THE CHARACTER PRECEDING THE FIRST CHARACTER OF THE NEXT	*		
9687	*			DATA ELEMENT TO BE CONVERTED AND STACKED.	*		
9688	*				*		
9689	*		OUTPUT		*		
9690	*		*	INPUT DATA BUFFER - 256 BYTES, FOR DATA ELEMENT STORAGE DURING	*		
9691	*			SYNTAX CHECKING AND CONVERSION.	*		
9692	*		*	AFTER ENTRY FZXIP1, THIS BUFFER IS LOCKED INTO CORE, DATA	*		
9693	*			ARE LOADED INTO THE BUFFER FROM THE SYSTEM KEYBOARD, AND	*		
9694	*			THESE DATA ARE COMPLETELY SYNTAX/TYPE CHECKED (EXCEPT WHEN	*		
9695	*			CONSOLE INTERRUPT IS INVOKED).	*		
9696	*		*	AFTER ENTRY FZXIP2, THIS BUFFER IS UNLOCKED FROM CORE	*		
9697	*			FOLLOWING CONVERSION OF THE FINAL DATA ELEMENT (DEFINED BY	*		
9698	*			AN EOS CHARACTER DELIMITER).	*		
9699	*		*	AFTER ENTRY FZXMIS, THE DATA CONTAINED IN THIS BUFFER ARE	*		
9700	*			COMPLETELY SYNTAX/TYPE CHECKED (UP TO THE POINT WOCRE AN	*		
9701	*			ERROR CONDITION, IF ONE EXISTS. IS ENCOUNTERED).	*		
9702	*		*	AFTER ENTRY FZXGCS, THE DATA CONTAINED IN THIS BUFFER ARE	*		
9703	*			COMPLETELY SYNTAX CHECKED (UP TO THE POINT WHERE AN ERROR	*		
9704	*			CONDITION, IF ONE EXISTS. IS ENCOUNTERED).	*		
9705	*		*	DATA BUFFER POINTER - 2 BYTES, FOR THE CORE ADDRESS OF THE	*		
9706	*			CURRENTLY REFERENCED BUFFER CHARACTER. THIS POINTER IS CON-	*		
9707	*			TAINED IN THE TWO RIGHTMOST BYTES OF THE INPUT DATA BUFFER.	*		
9708	*		*	AFTER ENTRIES FZXIP1, FIXMIS, AND FIXGCS, THIS IS SET TO	*		
9709	*			CONTAIN THE CORE ADDRESS OF THE BYTE PRECEDING THE FIRST	*		
9710	*			BUFFER BYTE,	*		
9711	*		*	AFTER ENTRIES FZXIP2 AND FZXCNV, THIS IS SET TO CONTAIN	*		
9712	*			THE CORE ADDRESS OF THE DELIMITER FOLLOWING THE CONVERTED	*		
9713	*			DATA ELEMENT.	*		
9714	*		*	I\$PUB1 - 2 BYTES, FOR THE INPUT DATA BUFFER CORE ADDRESS.	*		
9715	*			AFTER ENTRY FZXIP1 ONLY, THIS IS SET TO CONTAIN THE CORE	*		
9716	*			ADDRESS OF THE INPUT DATA BUFFER LEFTMOST BYTE.	*		
9717	*		*	REGISTER @XR - AFTER ENTRIES FZXGCS, FZXMIS AND FZXCNV, THIS	*		
9718	*			CONTAINS THE CORE ADDRESS OF THE LAST REFERENCED INPUT DATA	*		
9719	*			BUFFER CHARACTER,	*		
9720	*		*	I\$PARM - 2 BYTES, FOR THE INTERPRETER COMMUNICATIONS PARAMETER,	*		
9721	*			AFTER ENTRIES FZXMIS AND FZXGCS, THE LEFT BYTE IN I\$PARM IS SET	*		
9722	*			TO CONTAIN A COUNT OF THE DATA ELEMENTS SYNTAX CHECKED IN THE	*		
9723	*			BUFFER.	*		
9724	*		*	I\$ERRC - 1 BYTE, FOR THE INTERPRETER ERROR CONDITION CODE.	*		
9725	*		*	AFTER ENTRY FZXPEM, THIS IS RESET TO CONTAIN NULL ERROR	*		
9726	*			CODE I@NERR.	*		
9727	*		*	AFTER ENTRY FZXMIS, THIS IS SET TO CONTAIN ERROR CODE	*		
9728	*			X'F0', X'F2', OR X'F4' (CORRESPONDING TO ERROR MESSAGES	*		
9729	*			800, 802, OR 804 RESPECTIVELY) WHEN A SYNTAX OR DATA TYPE	*		
9730	*			ERROR IS ENCOUNTERED.	*		
9731	*		*	AFTER ENTRY FZXGCS, THIS IS SET TO CONTAIN ERROR CODE	*		
9732	*			X'FF' WHEN A SYNTAX ERROR IS ENCOUNTERED.	*		
9733	*		*	RUN-TIME STACK - AFTER ENTRIES FZXIP2 AND FZXCNV, THIS CONTAINS	*		
9734	*			THE CONVERTED INPUT DATA ELEMENT BEGINNING AT THE CORE ADDRESS	*		
9735	*			SPECIFIED IN I\$STAK. ARITHMETIC ELEMENTS ARE STACKED IN PACKED	*		

## DFPRNT - MATRIX PRINTER ROUTINE

ERR LOC OBJECT CODE ADDR STMT SOURCE STATEMENT VER 15, MOD 00 31/05/21 PAGE 151

```

9736 *      FLOATING POINT FORMAT OF CURRENT PRECISION.  CHARACTER ELEMENTS *
9737 *      ARE STACKED IN 19-BYTE FORMAT. *
9738 *      * PRINTER/CRT MESSAGES - *
9739 *          * FOR ENTRY FZXIP1, FZXPQ1 IS EXECUTED EACH TIME THAT KEY- *
9740 *          BOARD DATA INPUT IS REQUESTED. *
9741 *          * FOR ENTRY FZXIP1, AN ERROR MESSAGE (800, 801, 802, OR 803) *
9742 *          IS PRINTED WHENEVER A SYNTAX/TYPE ERROR IS ENCOUNTERED, *
9743 *          AND KEYBOARD DATA INPUT RE-ENTRY IS REQUESTED. *
9744 *          * FOR ENTRY FZXPQ1, THE PRINT DEVICE CARRIER IS RETURNED, *
9745 *          A SINGLE QUESTION MARK IS PRINTED, AND THE CARRIER IS *
9746 *          RETURNED AGAIN TO START A NEW LINE. *
9747 *          * FOR ENTRY FZXPQ2, THE PRINT DEVICE CARRIER IS RETURNED, *
9748 *          A DOUBLE QUESTION MARK IS PRINTED, AND THE CARRIER IS *
9749 *          RETURNED AGAIN TO START A NEW LINE. *
9750 *          * FOR ENTRY FZXPQM, THE PRINT DEVICE CARRIER IS RETURNED, *
9751 *          A MESSAGE OF THE FORM *
9752 *          ERROR NNN AT LINE LULL MESSAGE... *
9753 *          IS PRINTED, AND THE CARRIER IS RETURNED TO START A NEW *
9754 *          LINE.  MESSAGES ARE - *
9755 *          * 800 - INVALID INPUT DATA - NUMERIC CONSTANT *
9756 *          * 801 - INVALID INPUT DATA - CHARACTER DATA *
9757 *          * 802 - TOO MANY INPUT DATA ELEMENTS *
9758 *          * 803 - NOT ENOUGH DATA ELEMENTS ENTERED *
9759 *          * 804 - NOT ENOUGH ARRAY ROW ELEMENTS ENTERED *
9760 *      * I$STAK - 2 BYTES, FOR THE RUN-TIME STACK POINTER.  FOR ENTRY *
9761 *      FZXIP1 ONLY, THIS POINTER IS DECREMENTED BY THE LENGTH OF A *
9762 *      VIRTUAL ADDRESS (2 BYTES) WHENEVER CONSOLE INTERRUPT IS *
9763 *      INVOKED DURING KEYBOARD INPUT. *
9764 *      * I$WRK1 - 2 BYTES, FOR INTERPRETER COMMON WORK AREA 1.  AFTER *
9765 *      ENTRY FZXIP1 ONLY, THIS AREA IS SET TO CONTAIN THE VIRTUAL *
9766 *      ADDRESS OF THE CURRENT 'INPUT' STATEMENT 'STH' PSEUDO INSTRU- *
9767 *      CTION WHENEVER CONSOLE INTERRUPT IS INVOKED DURING KEYBOARD I/P. *
9768 *      * ISRESW - 1 BYTE, FOR THE RECURSIVE STATEMENT ERROR SWITCH. *
9769 *      THIS SWITCH IS SET TO CODE @NOP (DISABLES THE ERROR CONDITION) *
9770 *      WHENEVER CONSOLE INTERRUPT IS INVOKED DURING KEYBOARD INPUT. *
9771 *      *
9772 *EXTERNAL REFERENCES *
9773 *      * V$SKEY - VIRTUAL ENTRY ADDRESS FOR DFKEYN, V.M. KEYBOARD IOCS. *
9774 *      * V$SPRT - VIRTUAL ENTRY ADDRESS FOR DFPRNT, V.M. MATRIX PRT IOCS. *
9775 *      * DSPLIN - ENTRY POINT FOR THE SYSTEM CRT IOCS (LABEL DSPLIN IS *
9776 *      REFERENCED INDIRECTLY USING I$CSXA TO BUILD A CORE ADDRESS). *
9777 *      * I$CALL - ENTRY POINT FOR PAGING MODULE V.M. PROGRAM CALL RTN. *
9778 *      * I$RTRN - ENTRY POINT FOR PAGING MODULE V.M. RETURN CONTROL RTN. *
9779 *      * I$LOCK - ENTRY POINT FOR PAGING MODULE V.M. PAGE LOCKING RTN. *
9780 *      * I$UNLK - ENTRY POINT FOR PAGING MODULE V.M. PAGE UNLOCKING RTN. *
9781 *      * I$STCK - ENTRY POINT FOR INTERPRETER ELEMENT STACKING ROUTINE. *
9782 *      * I$CUPF - ENTRY POINT FOR FLOATING POINT VALUE PACKING ROUTINE. *
9783 *      * I$BSET - ENTRY POINT FOR ICBAN EXECUTION CONTROL BRANCH RTN. *
9784 *      * I$STAK - 2 BYTES, FOR THE RUN-TIME STACK POINTER. *
9785 *      * I$PARM - 2 BYTES, FOR THE INTERPRETER COMMUNICATION PARAMETER. *
9786 *      * I$WRK1 - 2 BYTES, FOR INTERPRETER COMMON WORK AREA 1. *
9787 *      * I$ERRC - 1 BYTE, FOR THE INTERPRETER EXECUTION ERROR CODE. *
9788 *      * I$STHA - 2 BYTES, FOR VIRTUAL ADDRESS OF CURRENT STMT 'STH'. *
9789 *      * I$RESW - 1 BYTE, FOR INTERPRETER STATEMENT RECURSION ERROR SW. *
9790 *      * I$PUB1 - 2 BYTES, FOR THE DATA BUFFER SAVED CORE ADDRESS. *
9791 *      * I$VADR - 2 BYTES, FOR PAGING MODULE VIRTUAL ADDRESS PARAMETER. *

```

## DFPRNT - MATRIX PRINTER ROUTINE

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 31/05/21 PAGE 152
		9792	*	* I\$CADR - 2 BYTES, FOR PAGING MODULE CORE ADDRESS OUTPUT PARAM.	*
		9793	*	* \$INLNO - 2 BYTES, FOR THE SYSTEM EXECUTION LINE NUMBER AREA.	*
		9794	*	* \$PRDEV - 2 BYTES, FOR THE SYSTEM PRINT DEVICE INDICATOR.	*
		9795	*	* \$EXFTR - 1 BYTE, FOR THE SYSTEM CORE EXTENSION FACTOR.	*
		9796	*	* \$CISUS - 1 BYTE FOR THE SYSTEM CONSOLE INTERRUPT INDICATOR.	*
		9797	*	* V\$BFR1 - VIRTUAL ADDRESS FOR GENERAL V.M. BUFFER 1.	*
		9798	*	* I\$BASE - CORE ADDRESS FOR INTERP BASE ADDRESSABILITY.	*
		9799	*	* I\$CSXA - CORE ADDRESS OF 1ST BYTE IN CORE EXTENSION PAST 8K.	*
		9800	*		*
		9801	*	*EXITS, NORMAL	*
		9802	*	CONTROL IS NORMALLY PASSED TO THE PAGING ROUTINE AT ENTRY POINT	*
		9803	*	I\$RTRN (IPGRTN) FOR A RETURN TO THE CALLING PROGRAM. THE SINGLE	*
		9804	*	EXCEPTION OCCURS WHEN A CONSOLE INTERRUPT IS INVOKED WHILE DATA	*
		9805	*	ARE BEING ENTERED THROUGH THE KEYBOARD DURING FZXIP1 EXECUTION.	*
		9806	*	IN THIS EVENT, CONTROL IS PASSED TO THE CORE-RESIDENT INTERPRETER	*
		9807	*	AT ENTRY POINT I\$BSET (ICBSET) TO FORCE RE-EXECUTION OF THE LAST	*
		9808	*	EXECUTED 'STH' PSEUDO INSTRUCTION (I.E, THE STATEMENT HEADER	*
		9809	*	ASSOCIATED WITH THE CURRENT 'INPUT' STATEMENT).	*
		9810	*		*
		9811	*	*EXITS, ERROR	*
		9812	*	* FOR ENTRY FZXMIS, CONTROL IS PASSED TO THE PAGING ROUTINE AT	*
		9813	*	ENTRY POINT I\$RTRN (IPGRTN) WHENEVER A SYNTAX OR DATA TYPE	*
		9814	*	ERROR IS ENCOUNTERED. ERROR PARAMETER I\$ERRC IS SET TO ONE OF	*
		9815	*	THE CODES X'F0', X'F2', OR X'F4' WHEN THIS SITUATION OCCURS	*
		9816	*	(SEE OUTPUT ABOVE).	*
		9817	*	* FOR ENTRY FZXSCS, CONTROL IS PASSED TO THE PAGING ROUTINE AT	*
		9818	*	ENTRY POINT I\$RTRN (IPGRTN) WHENEVER A SYNTAX ERROR IS	*
		9819	*	ENCOUNTERED. ERROR PARAMETER I\$ERRC IS SET TO CODE X'FF'.	*
		9820	*	* FOR ALL OTHER ENTRY POINTS, ENCOUNTERED ERROR CONDITIONS CAUSE	*
		9821	*	SPECIFIC ERROR-CORRECTION ACTIONS TO BE EXECUTED INTERNALLY.	*
		9822	*		*
		9823	*	*TABLES/WORK AREAS	*
		9824	*	* INPUT DATA BUFFER - SINGLE VIRTUAL PAGE USED TO CONTAIN	*
		9825	*	BOARD ENTERED DATA CONSTANTS AND INTERNAL CONSTANT SYMBOLS.	*
		9826	*	BYTES 254, 255 IN THIS PAGE ARE USED TO CONTAIN THE DATA BUFFER	*
		9827	*	CHARACTER POINTER.	*
		9828	*	* ERROR MESSAGE PAGE - THE 3RD VIRTUAL PAGE OR FZXINP CODING CON-	*
		9829	*	TAINS FIVE ERROR MESSAGE PARAMETER LISTS AND MESSAGE TEXTS FOR	*
		9830	*	SYSTEM ERROR MESSAGES 800, 801, 802, 803, AND 804. THIS PAGE	*
		9831	*	IS CREATED BY THE MTEXT MESSAGE GENERATOR, AND INCLUDES PATCH_	*
		9832	*	AREA SUFFICIENT FOR REASONABLE MESSAGE TEXT EXPANSION.	*
		9833	*	* INTERNAL CONSTANT SYNTAX CHECKING TABLE - THIS CONTAINS A LIST	*
		9834	*	OF INTERNAL CONSTANT SYMBOLS FOR VALIDITY CHECKING.	*
		9835	*	* NUMBER OF TABLE ENTRIES - 3	*
		9836	*	* TABLE ENTRY LENGTH - 4 BYTES	*
		9837	*	* ENTRY FORMAT -	*
		9838	*	* FILLED FROM LEFT TO RIGHT WITH THE LETTER PORTION	*
		9839	*	OF INTERNAL CONSTANT SYMBOLS (REVERSE SPELLING).	*
		9840	*	* PADDED ON RIGHT WITH BINARY ZEROS.	*
		9841	*	* INTERNAL CONSTANT CONVERSION TABLE - THIS CONTAINS A LIST OF	*
		9842	*	INTERNAL CONSTANT IDENTIFIERS AND THE VIRTUAL ADDRESS ASSOCI-	*
		9843	*	ATED WITH EACH (SIGNED) INTERNAL CONSTANT.	*
		9844	*	* NUMBER OF TABLE ENTRIES - 6	*
		9845	*	* TABLE ENTRY LENGTH - 4 BYTES	*
		9846	*	* ENTRY FORMAT -	*
		9847	*	* BYTE 0 - CONTAINS A SIGN CHARACTER (+ OR -)	*



## DFPRNT - MATRIX PRINTER ROUTINE

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00	31/05/21	PAGE 153
		9848	*	* BYTE 1 - CONTAINS SYMBOL 1ST LETTER CHARACTER	*		
		9849	*	* BYTES 2, 3 - CONTAIN INTERNAL CONSTANT VIRTUAL ADDR	*		
		9850	*	ASSOCIATED WITH THE SIGNED CONSTANT SYMBOL.	*		
		9851	*	* ARITHMETIC CONSTANT CONVERSION BUCKET - 16 BYTES, FOR ACCUMU-	*		
		9852	*	LATING CHARACTERS USED TO GENERATE AN INTERNAL FORM ARITHMETIC	*		
		9853	*	FLOATING POINT VALUE.	*		
		9854	*	* CHARACTER CONSTANT CONVERSION BUCKET - 19 BYTES, FOR ACCUMU-	*		
		9855	*	LATING CHARACTERS USED TO GENERATE AN INTERNAL FORM CHARACTER	*		
		9856	*	ELEMENT.	*		
		9857	*		*		
		9858	*	*ATTRIBUTES	*		
		9859	*	REUSABLE, NATURALLY RELOCATABLE	*		
		9860	*		*		
		9861	*	*CHARACTER CODE DEPENDENCY	*		
		9862	*	THE OPERATION OF THIS MODULE DEPENDS UPON THE FOLLOWING PROPER-	*		
		9863	*	TIES OF THE INTERNAL REPRESENTATION OF THE EXTERNAL CHAR SET,	*		
		9864	*	* MOST CODING HAS BEEN ARRANGED SO THAT REDEFINITION OF CHAR-	*		
		9865	*	ACTER CONSTANTS, BY REASSEMBLY, WILL RESULT IN A CORRECT	*		
		9866	*	MODULE FOR THE NEW DEFINITION.	*		
		9867	*	* NUMERIC CHARACTERS 0 THROUGH 9 ARE PRESUMED TO BE CODED IN	*		
		9868	*	INCREASING COLLATING SEQUENCE, AND THE RANGE OF CHARACTER	*		
		9869	*	CONSTANTS FOR THIS SERIES IS EXPECTED TO COLLATE HIGHER THAN	*		
		9870	*	THAT FOR ANY OTHER CHARACTER IN THE EXTERNAL CHARACTER SET.	*		
		9871	*	* ALPHABETIC LETTERS A THROUGH Z ARE PRESUMED TO EF CODED IN	*		
		9872	*	INCREASING COLLATING SEQUENCE, AND THE RANGE OF CKARACTER	*		
		9873	*	CONSTANTS FOR THIS SERIES IS EXPECTED TO COLLATE HIGHER THAN	*		
		9874	*	THAT FOR ANY OTHER NON-NUMERIC CHARACTER IN THE EXTERNAL	*		
		9875	*	CHARACTER SET.	*		
		9876	*	* DECIMAL NUMBERS MUST BE CODED SO TWAT THE LOW ORDER FOUR	*		
		9877	*	BITS, WHEN CONSIDERED AS A BINARY INTEGER, IDENTIFY THE	*		
		9878	*	VALUE OF THE DIGIT.	*		
		9879	*	THE SPECIFIC INSTRUCTIONS (INSTRUCTION SEQUENCES) WHICH REQUIRE	*		
		9880	*	MODIFICATION IF THESE PROPERTIES OF THE CHARACTER SET ARE CHANGED	*		
		9881	*	MAY BE IDENTIFIED EY -	*		
		9882	*	* THE 2 INSTRUCTIONS BEGINNING AT LABEL FZX575.	*		
		9883	*	* THE 8 INSTRUCTIONS BEGINNING AT LABEL FZX750.	*		
		9884	*	* THE 2 INSTRUCTIONS BEGINNING AT LABEL FZX845.	*		
		9885	*	* THE 2 INSTRUCTIONS BEGINNING AT LABEL FZX904.	*		
		9886	*	* THE 2 INSTRUCTIONS BEGINNING AT LABEL FZX940.	*		
		9887	*	* THE SINGLE INSTRUCTION IDENTIFIED BY LABEL F7X996.	*		
		9888	*		*		
		9889	*	*NOTES	*		
		9890	*	ERROR PROCEDURES	*		
		9891	*	* ERROR 1 - AN ARITHMETIC DATA ELEMENT IS EXPECTED DURING	*		
		9892	*	'INPUT' SYNTAX CHECKING, AND THE PROCESSED ELEMENT IS EITHER	*		
		9893	*	NON-ARITHMETIC OR OTHERWISE INVALID. THE ERROR MESSAGE	*		
		9894	*	'INVALID INPUT DATA - NUMERIC CONSTANT' IS DISPLAYED.	*		
		9895	*	* ERROR 2 - A CHARACTER DATA ELEMENT IS EXPECTED DURING 'INPUT'	*		
		9896	*	SYNTAX CHECKING, AND THE PROCESSED ELEMENT IS EITHER NON-	*		
		9897	*	CHARACTER OR OTHERWISE INVALID. THE ERROR MESSAGE 'INVALID	*		
		9898	*	INPUT DATA - CHARACTER DATA' IS DISPLAYED.	*		
		9899	*	* ERROR 3 - MORE THAN THE EXPECTED NUMBER OF DATA ELEMENTS ARE	*		
		9900	*	ENCOUNTERED DURING 'INPUT' SYNTAX' CHECKING. THE ERROR	*		
		9901	*	MESSAGE 'TOO MANY INPUT DATA ELEMENTS' IS DISPLAYED.	*		
		9902	*	* ERROR 4 - LESS THAN THE EXPECTED NUMBER OF DATA ELEMENTS ARE	*		
		9903	*	ENCOUNTERED DURING 'INPUT' SYNTAX CHECKING. THE ERROR	*		

## DFPRNT - MATRIX PRINTER ROUTINE

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 31/05/21 PAGE 154
		9904	*	MESSAGE 'NOT ENOUGH DATA ELEMENTS ENTERED' IS DISPLAYED.	*
		9905	*	* IN EACH OF THE ABOVE CASES, THE DISPLAY IS PERFORMED ON THE	*
		9906	*	CURRENT SYSTEM PRINT DEVICE(S), AND THE ERROR MESSAGE IS	*
		9907	*	FOLLOWED (ON A SEPARATE LINE) WITH A SINGLE QUESTION MARK	*
		9908	*	REQUESTING CORRECT RE-ENTRY OF THE 'INPUT' DATA LINE.	*
		9909	*	* ERROR CONDITIONS ENCOUNTERED DURING 'MAT INPUT' AND 'GET'	*
		9910	*	(CARD) DATA SYNTAX CHECKING ARE NOT CONSIDERED TO BE ASSOCI-	*
		9911	*	ATED WITH FZXINP, BUT EACH IS REGARDED (AS A CODE IN I\$ERRC)	*
		9912	*	TO BE OUTPUT TO THE 'MAT INPUT' OR 'GET' (CARD) PROCESSING	*
		9913	*	ROUTINES.	*
		9914	*	* KEYBOARD DATA INPUT CAN BE ABORTED AT ANY TIME, DURING DATA	*
		9915	*	ENTRY, THROUGH CONSOLE INTERRUPT. WHEN THIS OCCURS, THE	*
		9916	*	'STH' PSEUDO INSTRUCTION FOR THE CURRENT 'INPUT' STATEMENT	*
		9917	*	IS RE-EXECUTED AND THE PROGRAM IS HALTED IN 'PAUSE' MODE.	*
		9918	*	* FZXINP OTHERWISE UTILIZES INPUT IOCS ROUTINE DFKEYN (FOR THE	*
		9919	*	KEYBOARD) AND OUTPUT IOCS ROUTINES DFPRNT (MATRIX PRINTER)	*
		9920	*	AND DSPLYN (CRT), AND IS STASKCT TO THE ERP'S INHERENT IS	*
		9921	*	THESE CONTROL PROGRAMS.	*
		9922	*		*
		9923	*	REGISTER USAGE	*
		9924	*	* REGISTER @BR IS TO CONTAIN THE CORE PAGE BASE ADDRESS	*
		9925	*	ESTABLISHED THROUGH PAGING MODULE CONTROL FOR THE PAGE WHICH	*
		9926	*	INCLUDES FZXINP, AND IS RESTORED TWROUGH NE PAGING MODULE.	*
		9927	*	* REGISTER @XR IS NOT SAVED. IT IS USED IN FZXINP FOR GENERAL	*
		9928	*	PURPOSE INDEXING OPERATIONS, AND IN CERTAIN CASES IS ALSO	*
		9929	*	USED AS AN INPUT OR OUTPUT PARAMETER FOR THIS ROUTINE.	*
		9930	*		*
		9931	*	SAVED/RESTORED AREAS	*
		9932	*	NONE	*
		9933	*		*
		9934	*	MODIFICATION CONSIDERATIONS	*
		9935	*	NONE	*
		9936	*		*
		9937	*	REQUIRED MODULES	*
		9938	*	* @SYSEQ - COMMON SYSTEM EQUATES.	*
		9939	*	* @FXDEQ - SYSTEM NUCLE4S ADDRESSES AND INDICATOR EQUATES.	*
		9940	*	* \$V\$EQU - VIRTUAL MEMORY FIXED ADDRESS EQUATES.	*
		9941	*	* \$B@EQU - COMPILER PARAMETER AND CONSTANT EQUATES.	*
		9942	*	* \$I\$EQU - INTERPRETER FI\ED LOCATION ADDRESS ELATES.	*
		9943	*	* \$I@SEQ - INTERPRETER PARAMETER EQUATES (FOR STD. PREC, ONLY)	*
		9944	*	* \$I@LEQ - INTERPRETER PARAMETER EQUATES (FOR LONG PREC, ONLY)	*
		9945	*	* MTEXT - SYSTEM ERROR MESSAGE GENERATOR.	*
		9946	*		*
		9947	*	OTHER	*
		9948	*	NONE	*
		9949	*	*****	*

DFPRNT - MATRIX PRINTER ROUTINE

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	31/05/21	PAGE 155
					9951	*****	*****			
					9952	*	START OF INPUT STATEMENT EXECUTION MODULE			*
					9953	*****	*****			
					9954	*				
					9955	*	ESTABLISH ADDRESSABILITY FOR INPUT ROUTINE 1ST VM PAGE			
					9956	*				
					9957	*FZXPI8	VPAGE 0	SET 1ST PAGE ADDRESSABILITY		
2B00					9958	ORG	*,256,0	SET STARTING ADDRESS		
				2B00	9959	FZXP1B	EQU *	START OF PROGRAM CODING		
2A01					9960	ORG	*-255	RESET IAR TO PAGE		
2B00					9961	ORG	*,256,0	* BOUNDARY ADDRESS		
				2B00	9962	USING	*,@BR	SET PAGE RASE ADDRESS		
2B00					9963	ORG	FZXP1B	RESET STARTING ADDRESS		
					9964	***	END OF EXPANSION ***			
					9965	*				
					9966	*****	*****			

## DFPRNT - MATRIX PRINTER ROUTINE

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	31/05/21	PAGE 156
					9968		*****			
					9969	*	INPUT ENTRY 1 - PERMIT DATA INPUT VIA KEYBOARD AND CHECK SYNTAX			*
					9970		*****			
					9971	*				
				2B00	9972	MIN EQU *	FZXINP INPUT/CHECKING ENTRY PT			
					9973	*				
					9974	*	LOCK THE KEYBOARD INPUT BUFFER INTO CORE VIRTUAL MEMORY			
					9975	*				
	2B00	1C	01	144A	91	9976	MVC I\$VADR,FZXBVA(@VADDR,@BR)	SET PAGING VADDR PARAM FOR BFR		
	2B05	C0	87	1354		9977	B I\$LOCK	LINK TO CORELOAD AND LOCK BUFFER		
	2B09	0C	01	0DC8	144C	9978	MVC FZXBCA,I\$CADDR(@CADDR)	SAVE BUFFER CADDR IN FIXED AREA		
	2B0F	C0	87	0E24		9979	B I\$I700	CHECK LINE PRINTER BUFFER		
					9980	*				
					9981	*	INITIALIZE THE INPUT BUFFER TO BLANKS			
					9982	*				
	2B13	35	02	0DC8		9983	L FZXBCA,@XR	LOAD THE INPUT BUFFER CADDR		
	2B17	BC	40	FF		9984	MVI B@LVPG-1(,@XR),B@BLNK	INITIALIZE THE KEYBOARD INPUT		
	2B1A	AC	FE	FE	FF	9985	MVC B@LVPG-2(,@XR),B@LVPG-1(B@LVPG-1,@XR)	* BUFFER TO BLANKS		
					9986	*				
					9987	*	PRINT '?' TO REQUEST DATA INPUT VIA SYSTEM KEYBOARD			
					9988	*				
	2B1E	C0	87	12B1		9989	FZX010 B I\$CALL	LINK TO RETURN CARRIER AND		
	2B22	2C00			2B23	9990	DC AL(@VADDR)(FZX050)	* PRINT SINGLE QUESTION MARK		
					9991	*				
	2B24	3B	1E	03E4		9992	SBF \$LPRP3,@KENAB	RESET MATRIX PRINT IND.	1-3	
					9993	*				
					9994	*	EXECUTE THE KEYBOARD INPUT ROUTINE			
					9995	*				
	2B28	35	02	0DC8		9996	L FZXBCA,@XR	LOAD THE INPUT BUFFER CADDR		
	2B2C	C0	87	12B1		9997	B I\$CALL	LINK TO EXECUTE KEYBOARD INPUT		
	2B30	2500			2B31	9998	DC AL(@VADDR)(V\$SKEY)	VADDR OF KEYBOARD INPUT IOCR		
					9999	*				
						*	TEST FOR CONSOLE INTERRUPT DURING KEYBOARD INPUT - RE-EXECUTE			
					1	*	CURRENT STATEMENT 'STH' INSTRUCTION WHEN INTERRUPT HAS BEEN INVOKED			
					2	*				
	2B32	3D	80	0496		3	CLI \$CISUS,@NOP	IF NO PENDING CONSOLE INTERRUPT		
	2B36	F2	01	1E		4	JNE FZX020	* BRANCH TO SYNTAX CHECK INPUT		
					5	*				
	2B39	1F	00	0D4E	8F	6	SLC I\$STAK,FZXLVA(@CADDR-1,@BR)	DECR STACK PT PAST BR VADDR		
	2B3E	0C	01	0D59	0D51	7	MVC I\$WRK1,I\$STHA(@VADDR)	SET BRANCH VADDR FOR LAST 'STH'		
	2B44	3C	80	0CE9		8	MVI I\$RESW,@NOP	DISABLE STMT NO. RECURSION ERR		
	2B48	1C	01	130B	56	9	MVC I\$MOD4+@OP1,FZXPSA(@CADDR,@BR)	RE-INITLZ PAGE STCK PT 1-4		
	2B4D	C2	01	0C60		10	LA I\$BASE,@BR	LOAD THE INTERPRETER BASE CADDR		
	2B51	C0	87	119D		11	B I\$BSET	EXIT TO RE-EXECUTE 'STH' PMC		
	2B55	15CB			2B56	12	FZXPSA DC AL(@CADDR)(I\$PSTK+1)	PASS LINK STACK PT INITLZN		
					13	*				
					14	*	SYNTAX CHECK THE ENTIRE INPUT LINE - CHECK INPUT LINE DATA ELEMENTS			
					15	*	FOR CORRESPONDENCE WITH PROGRAM INPUT STATEMENT VARIABLE DATA LIST			
					16	*				
	2B57	C0	87	12B1		17	FZX020 B I\$CALL	LINK TO SYNTAX CHECK INPUT LINE		
	2B5B	2E20			2B5C	18	DC AL(@VADDR)(FZX270)	VADDR OF 'INPUT' SYNTAX CHECKER		
					19	*				
					20	*	TEST FOR A DISCOVERED SYNTAX OR DATA TYPE ERROR - ON ERROR CONDITION			
					21	*	PRINT APPROPRIATE MESSAGE AND REPEAT KEYBOARD DATA INPUT ROUTINE			
					22	*				
	2B5D	3D	00	0CBC		23	CLI I\$ERRC,I@NERR	IF NO DATA SYNTAX/TYPE ERROR		

DFPRNT - MATRIX PRINTER ROUTINE

ERR LOC		OBJECT CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	31/05/21	PAGE 157
2B61	C0	81 12D3		24	BE	I\$RTRN			* EXIT TO THE CALLING PROGRAM
				25	*				
2B65	C0	87 12B1		26	B	I\$CALL			LINK TO RETURN CARRIER AND
2B69	2C18		2B6A	27	DC	AL(@VADDR)(FZX080)			* PRINT SPECIFIED ERROR MESSAGE
				28	*				
2B6B	D0	87 1E		29	B	FZX010(,@BR)			GO REPEAT DATA LINE INPUT
				30	*				* AND SYNTAX/TYPE ERROR CHECK
				31	*				
				32	*****				

## DFPRNT - MATRIX PRINTER ROUTINE

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	31/05/21	PAGE 158
					34		*****			
					35		* INPUT ENTRY 2 - CONVERT AND STACK SINGLE INPUT LINE DATA ELEMENT *			
					36		*****			
					37		*			
				2B6E	38	FZXIP2 EQU *	FZXINP CONVERSION ENTRY POINT			
					39		*			
					40		* CONVERT NEXT AVAILABLE INPUT LINE ELEMENT AND MOVE TO RUN-TIME STACK			
					41		*			
	2B6E	35	02	0DC8	42	L FZXBCA,@XR	LOAD THE INPUT BUFFER CADDR			
					43		*			
	2B72	C0	87	12B1	44	B I\$CALL	LINK TO CONVERT AND STACK NEXT			
	2B76	3100		2B77	45	DC AL(@VADDR)(FZX860)	* DATA ELEMENT IN INPUT BUFFER			
					46		*			
					47		* TEST FOR FINAL ELEMENT IN INPUT BUFFER - RELEASE BUFFER PAGE			
					48		* FROM CORE WHEN END OF INPUT LINE IS ENCOUNTERED			
					49		*			
	2B78	BD	1E	00	50	CLI B@CHAR(,@XR),B@EOST	IF ELEMENT DELIMITER NOT EOS			
	2B7B	F2	01	0D	51	JNE FZX030	* GO EXIT LEAVING BUFFER LOCKED			
					52		*			
	2B7E	1C	01	144A 91	53	MVC I\$VADR,FZXBVA(@VADDR,@BR)	SET PAGING VADDR PARAM FOR BFR			
	2B83	C0	87	1350	54	B I\$UNLK	LINK TO UNLOCK BUFFER FROM CORE			
					55		*			
	2B87	3B	1E	03E4	56	SBF \$LPRP3,@KENAB	RESET MATRIX PRINT IND. 1-3			
	2B8B	C0	87	12D3	57	FZX030 B I\$RTRN	EXIT TO THE CALLING PROGRAM			
					59		*****			
					60		* INPUT EXECUTION ROUTINE CONSTANTS (1ST VM PAGE)			
					61		*****			
					62		*			
	2B8F	02		2B8F	63	FZXLVA DC AL(@CADDR-1)(@VADDR)	LENGTH OF A VIRTUAL ADDRESS			
	2B90	5400		2B91	64	FZXBVA DC AL(@VADDR)(V\$BFR1)	KEYBOARD INPUT BUFFER VADDR			
					65		*			
					66		*****			

## DFPRNT - MATRIX PRINTER ROUTINE

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 31/05/21 PAGE 159
			68	*****	
			69	* VIRTUAL MEMORY INPUT EXECUTION ROUTINE 2ND VM PAGE -	*
			70	* PERFORMS FOLLOWING ACTIVITIES DEPENDING ON ENTRY POINT	*
			71	*	*
			72	* SINGLE QUESTION MARK ENTRY -	*
			73	* * RETURNS CARRIER ON SYSTEM PRINT DEVICE	*
			74	* * PRINTS '?' AND RETURNS THE CARRIER	*
			75	*	*
			76	* DOUBLE QUESTION MARK ENTRY -	*
			77	* * RETURNS CARRIER ON SYSTEM PRINT DEVICE	*
			78	* * PRINTS '??' AND RETURNS THE CARRIER	*
			79	*	*
			80	* ERROR MESSAGE ENTRY -	*
			81	* * RETURNS CARRIER ON SYSTEM PRINT DEVICE	*
			82	* * PRINTS ERROR MESSAGE INDICATED BY INTEGER VALUE IN I\$ERRC	*
			83	* (I\$ERRC = DECIMAL 0,... 4 TO SPECIFY MESSAGE 800,... 804)	*
			84	* * RETURNS PRINT DEVICE CARRIER AFTER ERROR MESSAGE OUTPUT	*
			85	* * RESTORES ERROR CODE I\$ERRC TO NULL STATUS	*
			86	*****	
			87	*	
			88	* ESTABLISH ADDRESSABILITY FOR INPUT ROUTINE 2ND VM PAGE	
			89	*	
2C00			90	*FZXP2B VPAGE 0	SET 2ND PAGE ADDRESSASILIFI
			91	ORG *,256,0	SET STARTING ADDRESS
	2C00		92	FZXP2B EQU *	START OF PROGRAM CODING
2B01			93	ORG *-255	RESET IAR TO PAGE
2C00			94	ORG *,256,0	* BOUNDARY ADDRESS
	2C00		95	USING *,@BR	SET PAGE BASE ADDRESS
2C00			96	ORG FZXP2B	RESET STARTING ADDRESS
			97	*** END OF EXPANSION ***	
		2D00	99	USING FZXP3B,@XR	SET ERROR MESSAGE ADDRESSIBILITY
			100	*	
			101	*****	



## DFPRNT - MATRIX PRINTER ROUTINE

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 31/05/21 PAGE 160
		103		*****	
		104		* ENTRY FOR SINGLE QUESTION MARK OUTPUT TO PRINTER	*
		105		*****	
		106		*	
		2C00	107	FZXPQ1 EQU *	PRINT (?) ENTRY POINT
		108		*	
2C00 D2 02 E2		109	FZX050 LA	FZXQM1(,@BR),@XR	LOAD SINGLE QUE TION MARK CADDR
2C03 F2 87 03		110	J	FZX070	BRANCH TO PRINT QUESTION MARK
		112		*****	
		113		* ENTRY FOR DOUBLE QUESTION MARK OUTPUT TO PRINTER	*
		114		*****	
		115		*	
		2C06	116	FZXPQ2 EQU *	PRINT (??) ENTRY POINT
		117		*	
2C06 D2 02 E1		118	FZX060 LA	FZXQM2(,@BR),@XR	LOAD DOUBLE QUESTION MARK CADDR
		119		*	
2C09 74 02 F2		120	FZX070 ST	FZXPRP+@PDATA(,@BR),@XR	STORE OUTPUT FIELD CADDR IN PPL
		121		*	
		122		* RETURN PRINTER CARRIER AND PRINT SPECIFIED QUESTION MARK(S)	
		123		*	
2C0C D0 87 A1		124	B	FZX150(,@BR)	LINK TO RETURN PRINTER CARRIER
		125		*	
2C0F D2 02 EF		126	LA	FZXPRP(,@BR),@XR	LOAD PRINT & RETURN PPL CADDR
2C12 D0 87 A8		127	B	FZX170(,@BR)	LINK TO PRINT QUESTION MARK(S)
		128		*	
2C15 F2 87 85		129	J	FZX140	BRANCH TO EXIT PRINT ROUTINE
		130		*	
		131		*****	

## DFPRNT - MATRIX PRINTER ROUTINE

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00 31/05/21 PAGE 161
					133		*****	
					134		* ENTRY FOR ERROR MESSAGE OUTPUT TO PRINTER	*
					135		*****	
					136		*	
				2C18	137	FZXPEM EQU	* PRINT MESSAGE ENTRY POINT	
					138		*	
					139		* CONVERT BINARY LINE NUMBER TO ZONED DECIMAL FOR OUTPUT	
					140		*	
	2C18	4C	01 E5 03CF		141	FZX080 MVC	FZXBLN(, @BR), \$INLNO(B@LSNO) MOVE BINARY LINE NO. TO WORK	
	2C1D	54	30 ED DB		142	ZAZ	FZXDLN(B@LDSN, @BR), FZX2D0(1, @BR) INITLZ DECIMAL LINE NO.	
	2C21	54	30 E9 DC		143	ZAZ	FZXDAC(B@LDSN, @BR), FZX2D1(1, @BR) SET DECML ACCUM EQUAL 1	
	2C25	7C	01 29		144	MVI	FZX090+@Q(, @BR), @B1 SET BINARY MASK FOR 2**0 BIT	
	2C28	78	00 E5		145	FZX090 TBN	FZXBLN(, @BR), *- * TEST BINARY LINE NUMBER BIT	
	2C2B	F2	90 04		146	JF	FZX100 * AND BRANCH IF BIT IS ZERO	
	2C2E	56	03 ED E9		147	AZ	FZXDLN(B@LDSN, @BR), FZXDAC(B@LDSN, @BR) INCR DECIMAL NUMBER	
	2C32	5E	00 29 29		148	FZX100 ALC	FZX090+@Q(, @BR), FZX090+@Q(1, @BR) SHIFT BINARY MASK LEFT	
	2C36	F2	20 07		149	JNOL	FZX110 BRANCH UNLESS MASK EXCEEDS 2**7	
	2C39	5C	00 E5 E4		150	MVC	FZXBLN(, @BR), FZXBLN-1(1, @BR) MOVE HIGH ORDER BYTE TO LOW	
	2C3D	7C	01 29		151	MVI	FZX090+@Q(, @BR), @B1 SET BINARY MASK FOR 2**8 BIT	
	2C40	56	03 E9 E9		152	FZX110 AZ	FZXDAC(B@LDSN, @BR), FZXDAC(B@LDSN, @BR) DOUBLE DECML ACCUM	
	2C44	D0	08 28		153	BNOZ	FZX090(, @BR) REPEAT LOOP UNTIL ACCLM > 8192	
					154		*	
					155		* RETURN PRINTER CARRIER FOR ERROR MESSAGE OUTPUT	
					156		*	
	2C47	D0	87 A1		157	B	FZX150(, @BR) LINK TO RETURN PRINTER CARRIER	
					158		*	
					159		* ACCESS THE INPUT ROUTINE ERROR MESSAGES	
					160		*	
	2C4A	1C	01 144A E0		161	MVC	I\$VADR, FZXEVA(@VADDR, @BR) SET PAGING VADDR PARAMETER	
	2C4F	C0	87 1354		162	B	I\$LOCK LINK TO READ AND LOCK MESSAGES	
	2C53	35	02 144C		163	L	I\$CADR, @XR LOAD MESSAGE PAGE BASE CADDR	
	2C57	74	02 69		164	ST	FZX120+@OP1(, @BR), @XR SAVE MESSAGE PAGE BASE CADDR	
					165		*	
					166		* PRINT ERROR NUMBER SEGMENT OF THE ERROR MESSAGE	
					167		*	
	2C5A	E2	02 00		168	LA	@M250(, @XR), @XR LOAD ERROR NO. SEGMENT PPL CADDR	
	2C5D	D0	87 A4		169	B	FZX160(, @BR) LINK TO PRINT ERROR NO. SEGMENT	
					170		*	
	2C60	D2	02 F7		171	LA	FZXPNP(, @BR), @XR LOAD ERROR NO. DIGIT PPL CADDR	
	2C63	D0	87 A8		172	B	FZX170(, @BR) LINK TO PRINT DIGIT IN I@ERRC	
					173		*	
					174		* PRINT LINE NUMBER SEGMENT OF THE ERROR MESSAGE	
					175		*	
	2C66	C2	02 0000		176	FZX120 LA	*-*, @XR LOAD MESSAGE PAGE BASE CADDR	
	2C6A	E2	02 04		177	LA	@M251(, @XR), @XR LOAD LINE NO. SEGMENT CADDR	
	2C6D	D0	87 A4		178	B	FZX160(, @BR) LINK TO PRINT LINE NO. SEGMENT	
					179		*	
	2C70	D2	02 EA		180	LA	FZXELN(, @BR), @XR LOAD DECIMAL LINE NO. CADDR	
	2C73	74	02 F6		181	ST	FZXPPSP+@PDATA(, @BR), @XR STORE LINE NO. CORE ADDR IN PPL	
	2C76	D2	02 F3		182	LA	FZXPPSP(, @BR), @XR LOAD PRINT & STOP PPL CORE ADDR	
	2C79	D0	87 A8		183	B	FZX170(, @BR) LINK TO PRINT ERROR LINE NUMBER	
					184		*	
					185		* PRINT ERROR MESSAGE SPECIFIED BY DIGIT IN ERROR CODE I\$ERRC -	
					186		* RETURN CARRIER TOLLOWING ERROR MESSAGE OUTPUT	
					187		*	
	2C7C	75	02 69		188	L	FZX120+@OP1(, @BR), @XR LOAD MESSAGE PAGE BASE ADDR	

DFPRNT - MATRIX PRINTER ROUTINE

ERR LOC		OBJECT CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00 31/05/21 PAGE 162
2C7F	E2	02 04		189	LA	@@M256-@PPLNG(,@XR),@XR	INITLZ ERROR MESSAGE PPL POINTER
2C82	E2	02 04		190	FZX130 LA	@PPLNG(,@XR),@XR	INCREMENT ERROR MESSAGE PPL PT
2C85	17	00 0CBC DC		191	SZ	I\$ERRC(1),FZX2D1(1,@BR)	DECK ERROR MESSAGE DIGIT CODE
2C8A	D0	02 82		192	BNM	FZX130(,@BR)	REPEAT LOOP UNTIL CGG_JS <
2C8D	D0	87 A4		193	B	FZX160(,@BR)	LINK TO PRINT ERROR MESSAGE
				194	*		
2C90	3C	00 0CBC		195	MVI	I\$ERRC,I@NERR	RESET ERROR CODE TO_ LL STATUS
				196	*		
				197	*	RELEASE THE ERROR MESSAGE PAGE FROM CORE VIRTUAL MEMORY	
				198	*		
2C94	1C	01 144A E0		199	MVC	I\$VADR,FZXEVA(@VADDR,@BR)	SET PAGING VADDR PARAMETER
2C99	C0	87 1350		200	B	I\$UNLK	LINK TO UNLOCK MESSAGE PAGE
				201	*		
				202	*	ERROR MESSAGE ROUTINE EXIT - RETURN TO CALLING ROUTINE	
				203	*		
2C9D	C0	87 12D3		204	FZX140 B	I\$RTRN	RETURN TO CALLING ROUTINE
				205	*		
				206	*****		

## DFPRNT - MATRIX PRINTER ROUTINE

ERR LOC OBJECT CODE ADDR STMT SOURCE STATEMENT VER 15, MOD 00 31/05/21 PAGE 163

```

208 *****
209 * PRINTER/CRT MESSAGE OUTPUT ROUTINE *
210 * * PRINTS SPECIFIED MESSAGE AND/OR CONTROLS CARRIER FOR MATRIX *
211 * PRINTER AND/OR CRT AS DEFINED AT SYSTEM DEVICE PARAMETER *
212 * *
213 * INPUT - *
214 * * REGISTER @XR - CONTAINS CORE ADDRESS OF PRINT PARAMETER LIST *
215 * *
216 * OUTPUT - *
217 * * PRINTED LINE AND/OR CARRIER CONTROL AS SPECIFIED IN PARAM LIST *
218 *****
219 *
220 * SPECIAL ENTRY 1 - SET MESSAGE OUTPUT ROUTINE TO PERFORM CARR RETURN
221 *
2CA1 D2 02 FB 222 FZX150 LA FZXCRP(, @BR), @XR LOAD CARRIER RETURN PPL CADDR
223 *
224 * SPECIAL ENTRY 2 - MODIFY PRINT FIELD CADDR IN PPL FOR RELOCATION
225 *
2CA4 9C 00 02 68 226 FZX160 MVC @PDATA-1(, @XR), FZXECA(1, @BR) ADJUST ERROR MESSAGE CADDR
227 * * IN PPL FOR CURR CORE LOCATION
228 *
229 * NORMAL PRINT ROUTINE ENTRY - SAVE THE RETURN ADDRESS
230 *
2CA8 74 08 DA 231 FZX170 ST FZX210+@OP1(, @BR), @ARR SAVE RETURN BRANCH ADDRESS
232 *
233 * DETERMINE POSSIBLE CORE ENTRY ADDRESS FOR THE CRT IOCR
234 *
2CAB 5C 01 C2 DE 235 MVC FZX180+@OP1(, @BR), FZXPPA(@CADDR, @BR) SET UP BASE CADDR
2CAF 4E 00 C1 043B 236 ALC FZX180+@OP1-1(, @BR), $EXFTR(1) * AND ADD EXTENSION FACTOR
237 *
238 * TEST FOR TYPE OF PRINT DEVICE ACTIVE ON SYSTEM
239 *
2CB4 1D 00 044A C1 240 CLC $PRDEV-1, FZX180+@OP1-1(, @BR) TEST PRINT DEVICE PARAMETER
2CB9 F2 82 11 241 JL FZX200 * AND BRANCH IF PRINTER ONLY
242 *
243 * CRT (AND POSSIBLY PRINTER) ACTIVE - OUTPUT AREA AREA ON THE CRT
244 *
2CBC 74 02 C4 245 ST FZX190(, @BR), @XR STORE PRINT PARAM LIST CADDR
246 *
2CBF C0 87 0000 247 FZX180 B *- * LINK TO EXECUTE THE CRT IOCR
2CC3 2CC4 248 FZX190 DS CL(@CADDR) PRINT PARAMETER LIST CADDR
249 *
250 * TEST FOR MATRIX PRINTER ACTIVE ON THE SYSTEM
251 *
2CC5 1D 01 044B C2 252 CLC $PRDEV, FZX180+@OP1(@CADDR, @BR) TEST PRINT DEVICE PARAM
2CCA F2 02 0A 253 JNL FZX210 * AND BRANCH IF CRT ONLY
254 *
255 * MATRIX PRINTER ACTIVE - OUTPUT PRINT AREA ON THE MATRIX PRINTER
256 *
2CCD 3A 1E 03E4 257 FZX200 SBN $LPRP3, @KENAB SET MATRIX PRINT IND. 1-3
2CD1 C0 87 12B1 258 B I$CALL GO TO MATRIX PRINTER ROUTINE
2CD5 2800 2CD6 259 DC AL(@VADDR) (V$SPRT) MATRIX PRINTER IOCR VADDR
260 *
261 * RETURN CONTROL TO THE MESSAGE OUTPUT ROUTINE
262 *
2CD7 C0 87 0000 263 FZX210 B *- * RETURN TO CALLING ROUTINE

```

[illegible]

DFPRNT - MATRIX PRINTER ROUTINE

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 31/05/21 PAGE 165
			267	*****	
			268	* INPUT EXECUTION ROUTINE CONSTANTS (2ND VM PAGE)	*
			269	*****	
			270	*	
2CDB F0		2CDB	271	FZX2D0 DC DL1'0'	DECIMAL INTEGER 0
2CDC F1		2CDC	272	FZX2D1 DC DL1'1'	DECIMAL INTEGER +1
			273	*	
2CDD 2004		2CDE	274	FZXPDA DC AL(@CADDR)(I\$CSXA+4)	CRT CORE ENTRY BASE ADDRESS
			275	*	
2CDF 2D00		2CE0	276	FZXEVA DC AL(@VADDR)(FZXP3B)	ERROR MESSAGE PAGE BASE VADDR
		2C68	277	FZXECA EQU FZX120+@OP1-1	ERROR MESSAGE PAGE BASE CADDR
			278	*	
2CE1 6F		2CE1	279	FZXQM2 EQU *	DOUBLE QUESTION MARK FIELD CADDR
		2CE1	280	DC CL1'?'	QUESTION MARK FIELD CONSTANT
		2CE2	281	FZXQM1 EQU *	SINGLE QUESTION MARK FIELD CADDR
2CE2 6F40		2CE3	282	DC CL2'?'	QUESTION MARK FIELD CONSTANT
		0002	283	FZXQML EQU 2	LENGTH OF QUESTION MARK FIELD
			284	*	
			285	*****	

## DFPRNT - MATRIX PRINTER ROUTINE

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	31/05/21	PAGE 166
					287	*****	*****			
					288	* INPUT EXECUTION ROUTINE WORK AREAS (2ND VM PAGE)	*			
					289	*****	*****			
					290	*				
2CE4				2CE5	291	FZXBLN DS	CL(B@LSNO)			BINARY LINE NUMBER WORK AREA
2CE6				2CE9	292	FZXDAC DS	CL(B@LDSN)			DECIMAL CONVERSION ACCUMULATOR
					293	*				
				2CEA	294	FZXELN EQU	*			LINE NUMBER OUTPUT FIELD CADDR
2CEA				2CED	295	FZXDLN DS	CL(B@LDSN)			LINE NUMBER OUTPUT WORK AREA
2CEE	40			2CEE	296		DC CL1 ' '			OUTPUT FIELD BLANK CHARACTER
					297	*				
					298	*FZXPRP PPL	FUNC-@PRETR,CNT-FZXQML			PRINT '? (?)' AND RETURN PPL
				2CEF	299	FZXPRP EQU	*			PPL ADDRESS
2CEF	C0			2CEF	300		DC AL1(@PRETR)			FUNCTION REQUESTED
2CF0	02			2CF0	301		DC AL1(FZXQML)			PRINT COUNT
2CF1	0000			2CF2	302		DC AL2(*-*)			DATA ADDRESS
					303	***	END OF EXPANSION ***			
					305	*FZXPPSP PPL	FUNC-@PRINT,CNT-B@LDSN+1			PRINT LINE NO. AND STOP PPL
				2CF3	306	FZXPPSP EQU	*			PPL ADDRESS
2CF3	40			2CF3	307		DC AL1(@PRINT)			FUNCTION REQUESTED
2CF4	05			2CF4	308		DC AL1(B@LDSN+1)			PRINT COUNT
2CF5	0000			2CF6	309		DC AL2(*-*)			DATA ADDRESS
					310	***	END OF EXPANSION ***			
					312	*FZXPNP PPL	FUNC-@PRINT,CNT-1,CADDR-I\$ERRC			PRINT ERROR DIGIT PPL
				2CF7	313	FZXPNP EQU	*			PPL ADDRESS
2CF7	40			2CF7	314		DC AL1(@PRINT)			FUNCTION REQUESTED
2CF8	01			2CF8	315		DC AL1(1)			PRINT COUNT
2CF9	0CBC			2CFA	316		DC AL2(I\$ERRC)			DATA ADDRESS
					317	***	END OF EXPANSION ***			
					319	*FZXCRP PPL	FUNC-@RETRN,CNT-@RTRNC			CARRIER RETURN PARA4ETER LIST
				2CFB	320	FZXCRP EQU	*			PPL ADDRESS
2CFB	80			2CFB	321		DC AL1(@RETRN)			FUNCTION REQUESTED
2CFC	80			2CFC	322		DC AL1(@RTRNC)			PRINT COUNT
2CFD	0000			2CFE	323		DC AL2(*-*)			DATA ADDRESS
					324	***	END OF EXPANSION ***			
					325	*				
					326	*****	*****			



## DFPRNT - MATRIX PRINTER ROUTINE

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00 31/05/21 PAGE 167
					328		*****	
					329		* VIRTUAL MEMORY INPUT EXECUTION ROUTINE (3RD VM PAGE)	*
					330		* CONTAINS ERROR MESSAGE TEXT AND PRINT PARAMETER LISTS	*
					331		*****	
					332		*	
					333		* ESTABLISH ADDRESSABILITY FOR INPUT ROUTINE 3RD VM PACE	
					334		*	
					335	*FZXP3B	VPAGE 0 SET 3RD PAGE ADDRESSABILITY	
	2D00				336	ORG	*,256,0 SET STARTING ADDRESS	
				2D00	337	FZXP3B EQU	* START OF PROGRAM CODING	
	2C01				338	ORG	*-255 RESET IAR TO PAGE	
	2D00				339	ORG	*,256,0 * BOUNDARY ADDRESS	
				2D00	340	USING	*,@BR SET PAGE BASE ADDRESS	
	2D00				341	ORG	FZXP3B RESET STARTING ADDRESS	
					342	***	END OF EXPANSION ***	
					344		*****	
					345		* INPUT EXECUTION ERROR MESSAGE PARAMETERS	*
					346		*****	
					347		*	
					348		MTEXT @@M250-@PRINT,	
					349		@@M251-@PRINT,	
					350		@@M256-@PRETR,	
					351		@@M257-@PRETR,	
					352		@@M258-@PRETR,	
					353		@@M259-@PRETR,	
					354		@@M260-@PRETR,	
					355		PATCH-040	
					356		*****	
					357		* PPL'S AND TEXT FOR MESSAGE	*
					358		*****	
	2D00 40			2D00	359	@M250 DC	AL1(@PRINT) PRINT CONTROL FUNCTION	
	2D01 08			2D01	360	DC	IL1'08' LENGTH OF MESSAGE	
	2D02 2D1C			2D03	361	DC	AL(@CADDR)(@T250) ADDR OF MESSAGE	
					362	*		
	2D04 40			2D04	363	@M251 DC	AL1(@PRINT) PRINT CONTROL FUNCTION	
	2D05 09			2D05	364	DC	IL1'09' LENGTH OF MESSAGE	
	2D06 2D24			2D07	365	DC	AL(@CADDR)(@T251) ADDR OF MESSAGE	
					366	*		
	2D08 C0			2D08	367	@M256 DC	AL1(@PRETR) PRINT CONTROL FUNCTION	
	2D09 25			2D09	368	DC	IL1'37' LENGTH OF MESSAGE	
	2D0A 2D2D			2D0B	369	DC	AL(@CADDR)(@T256) ADDR OF MESSAGE	
					370	*		
	2D0C C0			2D0C	371	@M257 DC	AL1(@PRETR) PRINT CONTROL FUNCTION	
	2D0D 23			2D0D	372	DC	IL1'35' LENGTH OF MESSAGE	
	2D0E 2D52			2D0F	373	DC	AL(@CADDR)(@T257) ADDR OF MESSAGE	
					374	*		
	2D10 C0			2D10	375	@M258 DC	AL1(@PRETR) PRINT CONTROL FUNCTION	
	2D11 17			2D11	376	DC	IL1'23' LENGTH OF MESSAGE	
	2D12 2D75			2D13	377	DC	AL(@CADDR)(@T258) ADDR OF MESSAGE	
					378	*		
	2D14 C0			2D14	379	@M259 DC	AL1(@PRETR) PRINT CONTROL FUNCTION	
	2D15 20			2D15	380	DC	IL1'32' LENGTH OF MESSAGE	
	2D16 2D8C			2D17	381	DC	AL(@CADDR)(@T259) ADDR OF MESSAGE	
					382	*		
	2D18 C0			2D18	383	@M260 DC	AL1(@PRETR) PRINT CONTROL FUNCTION	

DFPRNT - MATRIX PRINTER ROUTINE

ERR LOC		OBJECT CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	31/05/21	PAGE 168
2D19	25		2D19	384	DC	IL1'37'			LENGTH OF MESSAGE
2D1A	2DAC		2D1B	385	DC	AL(@CADDR)(@@T260)			ADDR OF MESSAGE
				386	*				
			2D1C	387	@@T250 EQU	*			LEFT BYTE OF MESSAGE
2D1C	C5D9D9D6D940F8F0		2D23	388	DC	CL008'ERROR 80'			
			2D24	389	@@T251 EQU	*			LEFT BYTE OF MESSAGE
2D24	40C1E340D3C9D5C5		2D2C	390	DC	CL009' AT LINE '			
			2D2D	391	@@T256 EQU	*			LEFT BYTE OF MESSAGE
2D2D	C9D5E5C1D3C9C440		2D51	392	DC	CL037'INVALID INPUT DATA - NUMERIC CONSTANT'			
			2D52	393	@@T257 EQU	*			LEFT BYTE OF MESSAGE
2D52	C9D5E5C1D3C9C440		2D74	394	DC	CL035'INVALID INPUT DATA - CHARACTER DATA'			
			2D75	395	@@T258 EQU	*			LEFT BYTE OF MESSAGE
2D75	E3D6D640D4C1D5E8		2D8B	396	DC	CL023'TOO MANY INPUT ELEMENTS'			
			2D8C	397	@@T259 EQU	*			LEFT BYTE OF MESSAGE
2D8C	D5D6E340C5D5D6E4		2DAB	398	DC	CL032'NOT ENOUGH DATA ELEMENTS ENTERED'			
			2DAC	399	@@T260 EQU	*			LEFT BYTE OF MESSAGE
2DAC	D5D6E340C5D5D6E4		2DD0	400	DC	CL037'NOT ENOUGH ARRAY ROW ELEMENTS ENTERED'			
				401	*				
				402	* PATCH AREA FOR MESSAGES				
				403	*				
2DD1			2DF8	404	\$\$\$001 DS	CL040			MSG EXPANSION PATCH AREA

## DFPRNT - MATRIX PRINTER ROUTINE

ERR	LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 31/05/21 PAGE 169
				406	*****	
				407	* VIRTUAL MEMORY INPUT EXECUTION ROUTINE (4TH VM PAGE)	*
				408	* SYNTAX CHECKING ROUTINE FOR THE FOLLOWING DATA INPUT MODES -	*
				409	* * INPUT - CHECKS SYNTAX AND SEQUENCE OF KEYBOARD INPUT LINE DATA	*
				410	* * MAT INPUT - CHECKS SYNTAX AND TYPE OF ARRAY ROW INPUT LINE DATA	*
				411	* * GET(CARD) - CHECKS SYNTAX OF DATA READ FROM A CARD DATA FILE	*
				412	*	*
				413	* INPUT -	*
				414	* * INPUT, MAT INPUT -	*
				415	* * I\$STAK - REFERENCES LEFTMOST BYTE OF STACKED DATA TYPE	*
				416	* SPECIFICATION CODES	*
				417	* * I\$PARM - CONTAINS NUMBER OF STACKED SPECIFICATION CODES	*
				418	* * INPUT, MAT INPUT, GET(CARD) -	*
				419	* * REGISTER @XR - REFERENCES LEFTMOST BYTE OF INPUT LINE BFR	*
				420	*	*
				421	* OUTPUT -	*
				422	* * I\$PARM-1 - CONTAINS INDEX NO OF LAST PROCESSED DATA ELEMENT	*
				423	* * I\$ERRC - CONTAINS CODE I@NERR IF NO SYNTAX ERROR ENCOUNTERED	*
				424	* - CONTAINS DECIMAL DIGIT 0,1,... 4 (CORRESPONDING TO ERROR	*
				425	* MESSAGE 800,801,... 804) FOR INPUT, MAT INPUT ERROR	*
				426	* - CONTAINS CODE X'FF' FOR GET(CARD) SYNTAX ERROR	*
				427	* * REGISTER @XR - REFERENCES LAST SCANNED INPUT BUFFER CHARACTER	*
				428	* * INPUT BUFFER - LAST 2 BYTES IN BUFFER PAGE INITIALIZED TO	*
				429	* (CADDR-1) WHERE CADDR IS ADDRESS OF INPUT BUFFER LEFT BYTE	*
				430	*****	
				431	*	
				432	* ESTABLISH ADDRESSABILITY FOR INPUT ROUTINE 4TH VM PAGE	
				433	*	
2E00				434	*FZXP4B VPAGE 0 SET 3RD PAGE ADDRESSABILITY	
				435	ORG *,256,0 SET STARTING ADDRESS	
	2E00			436	FZXP4B EQU * START OF PROGRAM CODING	
2D01				437	ORG *-255 RESET IAR TO PAGE	
2E00				438	ORG *,256,0 * BOUNDARY ADDRESS	
		2E00		439	USING *,@BR SET PAGE BASE ADDRESS	
2E00				440	ORG FZXP4B RESET STARTING ADDRESS	
				441	*** END OF EXPANSION ***	
				442	*	
				443	*****	

DFPRNT - MATRIX PRINTER ROUTINE

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 31/05/21 PAGE 170
		445		*****	
		446		* ENTRY FOR GET(CARD) DATA SYNTAX CHECKING	*
		447		*****	
		448		*	
	2E00	449	FZXGCS EQU	*	GET(CARD) SYNTAX CHECKER ENTRY
		450		*	
		451		* INITIALIZE SYNTAX CHECKER FOR GET(CARD) MODE	
		452		*	
2E00	3C 80 0D58	453	FZX250 MVI	FZXETS,@NOP	DISABLE ELEMENT TYPE CHECKING
2E04	3C FF 0CBC	454		MVI ISERRC,FZXSEC	SET CONTINGENCY ERROR CODE
2E08	7C 87 6E	455		MVI FZX320+@Q(,@BR),@UCB	DISALLOW BUFFER TERMINAL COMMA
2E0B	7C 87 94	456		MVI FZX340+@Q(,@BR),@UCB	DISABLE ELEMENT COUNT CHECKING
2E0E	7C 87 B3	457		MVI FZX390+@Q(,@BR),@UCB	DISABLE ELEMENT TYPE PROCESSING
2E11	7C 80 9B	458		MVI FZX370+@Q(,@BR),@NOP	FORCE GET(CARD) ERROR EXIT 1-3
2E14	F2 87 19	459		J FZX290	BRANCH TO CONTINUE INITIAL1ZING
		460		*	
		461		*****	

DFPRNT - MATRIX PRINTER ROUTINE

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	31/05/21	PAGE 171
					463	*****				
					464	* ENTRY FOR MAT INPUT DATA SYNTAX, TYPE, COUNT CHECKING	*			
					465	*****				
					466	*				
				2E17	467	FZXMIS EQU *	MAT INPUT SYNTAX CHECKER ENTRY			
					468	*				
					469	* INITIALIZE SYNTAX CHECKER FOR MAT INPUT MODE				
					470	*				
	2E17	7C	80	6E	471	FZX260 MVI FZX320+@Q(,@BR),@NOP	ALLOW BUFFER TERMINAL COMMA			
	2E1A	7C	F4	97	472	MVI FZX350+@Q(,@BR),FZXER4	SET ERROR 804 FOR TOO FEW DATA			
	2E1D	F2	87	06	473	J FZX280	SKIP TO CONTINUE INITIALIZATION			
					474	*				
					475	*****				

## DFPRNT - MATRIX PRINTER ROUTINE

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00 31/05/21 PAGE 172
					477	*****	*****	
					478	*	ENTRY FOR INPUT MODE DATA SYNTAY, TYPE, COUNT CHECKING	*
					479	*****	*****	
					480	*		
					481	*	INITIALIZE SYNTAX CHECKER FOR INPUT MODE	
					482	*		
2E20	7C	87	6E		483	FZX270 MVI	FZX320+@Q(, @BR), @UCB	DISALLOW BUFFER TERMINAL COMMA
2E23	7C	F3	97		484	MVI	FZX350+@Q(, @BR), FZXER3	SET ERROR 803 FOR TOO FEW DATA
					485	*		
2E26	3C	01	0D58		486	FZX280 MVI	FZXETS, @BNE	ENABLE ELEMENT TYPE CHECKING
2E2A	7C	81	94		487	MVI	FZX340+@Q(, @BR), @BE	ENABLE ELEMENT COUNT CHECKING
2E2D	7C	80	B3		488	MVI	FZX390+@Q(, @BR), @NOP	ENABLE ELEMENT TTPE PROCESSING
					489	*		
2E30	4C	01	C0 0D4E		490	FZX290 MVC	FZXSTP(, @BR), I\$STAK(@CADDR)	INITLZ DATA SPEC CODE POINTER
2E35	4C	00	F8 0D57		491	MVC	FZXSTC(, @BR), I\$PARM(B@LCNN)	MOVE SPEC CODE COUNT TO 4RK
2E3A	7C	00	F9		492	MVI	FZXDT(, @BR), @ZERO	SET DATA TYPE COUNTER EQUAL ZERO
2E3D	3C	00	0D59		493	MVI	FZXSER, I@NERR	SET ELEMENT ERROR CODE TO NULL
2E41	3C	00	0D56		494	MVI	FZXCNT, @ZERO	SET BUFFER ELEMENT COUNT = ZERO
					495	*		
					496	*	ESTABLISH DATA STARTING ADDRESS IN LAST 2 BYTES OF BUFFER	
					497	*	REGISTER @XR CONTAINS CORE ADDRESS OR LEFTMOST BYTE IN BUFFER	
					498	*		
2E45	B4	02	FF		499	ST	FZXBPT(, @XR), @XR	STORE BUFFER LEFT BYTE CADDR
2E48	9F	01	FF F7		500	SLC	FZXBPT(, @XR), FZX4B1(@CADDR, @BR)	DECR BUFFER POWER TO
2E4C	B5	02	FF		501	L	FZXBPT(, @XR), @XR	* BYTE PRECEDING BFR - LOAD PT
					502	*		
					503	*	IF ACCESS 1ST BYTE OF NEXT ELEMENT IN BUFFER - SET ELEMENT PARAMETERS	
					504	*		
2E4F	D0	87	AF		505	FZX300 B	FZX380(, @BR)	LINK TO SET EXPECTED DATA TYPE
2E52	D0	87	E6		506	B	FZX450(, @BR)	LINK TO GET NEXT DATA CHARACTER
2E55	1E	00	0D56 F7		507	FZX310 ALC	FZXCNT, FZX4B1(1, @BR)	INCREMENT COUNT FOR CURR ELEMENT
					508	*		
					509	*	SYNTAX/TYPE CHECK CURRENTLY REFERENCED ELEMENT	
					510	*		
2E5A	C0	87	12B1		511	B	I\$CALL	LINK TO PERFORM ELEMENT SYNTAX
2E5E	2F00			2E5F	512	DC	AL(@VADDR)(FZX500)	* AND TYPE CHECKING
					513	*		
2E60	3D	00	0D59		514	CLI	FZXSER, I@NERR	IF SYNTAX/TYPE ERROR LOAD FOUND
2E64	F2	01	33		515	JNE	FZX370	* GO RETURN TO CALLING ROUTINE
					516	*		
					517	*	DELIMITER PROCESSING - CHECK FOR COMMA FOLLOWING VALID ELEMENT	
					518	*		
2E67	BD	6B	00		519	CLI	B@CHAR(, @XR), B@CMA	IF ELEMENT DELIMITER NOT A COMMA
2E6A	F2	01	19		520	JNE	FZX330	* GO CHECK FOR END-OF-DATA CHAR
					521	*		FOR INPUT AND GET(CARD) MODES.
2E6D	D0	00	4F		522	FZX320 BC	FZX300(, @BR), *-*	* GO PROCESS ELEMENT AFTER COMMA
					523	*		
					524	*	MAT INPUT COMMA DELIMETER PROCESSING - CHECK FOR TERMINAL COMMA	
					525	*		
2E70	D0	87	AF		526	B	FZX380(, @BR)	LINK TO SET EXPECTED DATA TYPE
2E73	D0	87	E6		527	B	FZX450(, @BR)	LINK TO GET NEXT DATA CHARACTER
2E76	BD	1E	00		528	CLI	B@CHAR(, @XR), B@EOST	IF CHAR FOLLOWING COMMA NOT EOS
2E79	D0	01	55		529	BNE	FZX310(, @BR)	* GO PROCESS ELEMENT AFTER COMMA
					530	*		CURRENT DATA LINE ENDS W/ COMMA-
2E7C	3D	F0	0CBC		531	CLI	I\$ERRC, FZXER0	* IF MORE ELEMENTS EXPECTED FOR
2E80	F2	81	21		532	JE	FZX360	* CURR ARRAY ROW, GO RETURN FOR

## DFPRNT - MATRIX PRINTER ROUTINE

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	31/05/21	PAGE 173
2E83	F2 87 14	533	*					
		534	J	FZX370	* NEXT LINE, ELSE EXIT ON ERROR			
		535	*					
		536	*	DELIMITER PROCESSING - CHECK FOR EOS FOLLOWING LAST DATA ELEMENT				
		537	*					
2E86	BD 1E 00	538	FZX330	CLI	B@CHAR(,@XR),B@EOST	IF ELEMENT DELIMITER NOT AN EOS		
2E89	F2 01 0E	539		JNE	FZX370	* GO EXIT ON DATA SYNTAX ERROR		
		540	*					
		541	*	EOS DELIMITER ENCOUNTERED - CHECK FOR VALID ELEMENT COUNT				
		542	*					
2E8C	D0 87 AF	543		B	FZX380(,@BR)	LINK TO SET EXPECTED DATA TYPE		
2E8F	3D F2 0CBC	544		CLI	I\$ERRC,FZXER2	IF NO MORE ELEMENTS EXPECTED		
2E93	F2 00 0E	545	FZX340	JC	FZX360,*-*	* GO EXIT ON VALID DATA INPUT -		
		546	*			* BRANCH UNCONDITIONALLY FOR GET		
		547	*					
2E96	3C 00 0CBC	548	FZX350	MVI	I\$ERRC,*-*	PREMATURE EOS - SET ERROR CODE		
		549	*			* SPECIFYING MORE DATA NEEDED		
		550	*			* AND EXIT ON DATA CNT ERROR 1-3		
2E9A	F2 87 0B	551	FZX370	JC	FZX375,@UCB	JUMP IF NOT GET(CARD) ENTRY 1-3		
2E9D	3C BD 0CBC	552		MVI	I\$ERRC,@@E718	SET ERROR CODE 1-3		
2EA1	F2 87 04	553		J	FZX375	JUMP TO RETURN 1-3		
		554	*					
		555	*	SYNTAX CHECKER EXIT - SUPPRESS ERROR CODE FOR VALID DATA LINE				
		556	*					
2EA4	3C 00 0CBC	557	FZX360	MVI	I\$ERRC,I@NERR	SET ERROR CODE TO NULL STATUS		
		558	*			* WHEN NO SYNTAX/TYPE/COUNT ERR		
2EA8	7C 87 9B	559	FZX375	MVI	FZX370+@Q(,@BR),@UCB	FORCE NO GET(CARD) ENTRY 1-3		
2EAB	C0 87 12D3	560		B	I\$RTRN	RETURN TO CALLING ROUTINE 1-3		
		561	*					
		562	*****					



## DFPRNT - MATRIX PRINTER ROUTINE

ERR LOC OBJECT CODE ADDR STMT SOURCE STATEMENT VER 15, MOD 00 31/05/21 PAGE 174

```

564 *****
565 * INPUT ELEMENT TYPE SPECIFICATION ROUTINE *
566 * * OPERATES ON STACKED ELEMENT SPECIFICATION CODES TO DEFINE DATA *
567 * ELEMENT TYPE (ARITHMETIC OR CHARACTER) OR END OF DATA LIST *
568 * *
569 * INPUT - *
570 * * RUN-TIME STACK - CONTAINS DATA SPECIFICATION CODES WITH LEFT *
571 * BYTE OF STACKED CODES REFERENCED BY POINTER I$STAK. *
572 * * FZXSTC - STACKED CODE COUNTER, INITIALLY SET TO VALUE IN *
573 * I$PARM AND USED TO DETERMINE END-OF-DATA CONDITION. *
574 * *
575 * OUTPUT - *
576 * * I$ERRC - SET TO ONE OF 3 CODES (EXCEPT WHEN ROUTINE IS DISABLED) *
577 * * CODE X'F0' - INDICATES NEXT ELEMENT MUST BE ARITHMETIC *
578 * * CODE X'F1' - INDICATES NEXT ELEMENT MUST SE CHARACTER *
579 * * CODE X'F2' - INDICATES NO MORE ELEMENTS TO BE ACCEPTED *
580 *****
581 *
582 * ENTRY - SAVE RETURN ADDRESS AND CHECK ENABLED STATUS
583 *
2EAF 74 08 E5 584 FZX380 ST FZX440+@OP1(,@BR),@ARR SET RETURN BRANCH ADDRESS
2EB2 F2 00 2D 585 FZX390 JC FZX440,*-* GO RETURN IF SPEC RTN DISABLED
586 *
587 * CHECK CURRENT SPECIFICATION CODE FOR INDICATION OR ADDITIONAL
588 * REQUIRED DATA ELEMENT(S) OBEYING SPECIFIED DATA TYPE - SPEC CODE
589 * FORMAT IS THAT OF A 1-BYTE COUNT WITH BIT 0 INDICATING THE TYPE
590 * OF ELEMENT BEING SPECIFIED: 0 - ARITHMETIC,
591 * 1 - CHARACTER.
592 *
2EB5 5F 00 F9 F7 593 FZX400 SLC FZXDTC(,@BR),FZX4B1(B@LCXX,@BR) DECR CODE ELEMENT COUNT
2EB9 F2 02 22 594 JNM FZX430 GO SET TYPE INDICATOR FOR CURR
595 * * SPEC IF COUNT NOT LESS THAN 0
596 *
597 * CURRENT SPECIFICATION CODE DEPLETED - GET NEXT CODE TO WORK STACK AND
598 * RESET TYPE INDICATOR ACCORDINGLY
599 *
2EBC 4C 00 F9 0000 600 FZX410 MVC FZXDTC(,@BR),*-(B@LCXX) MOVE STACKED CODE TO WORK AREA
2EC1 5E 00 C0 F7 601 ALC FZXSTP(,@BR),FZX4B1(@CADDR-1,@BR) INCREMENT STACK POINTER
602 *
2EC5 7C F0 DF 603 MVI FZX430+@Q(,@BR),FZXER0 SET TYPE INDR ASSUMING ARITH
2EC8 78 80 F9 604 TBN FZXDTC(,@BR),FZXDTM TEST SPEC CODE TYPE BIT AND
2ECB F2 90 06 605 JF FZX420 * BRANCH IF ARITHMETIC
2ECE 7C F1 DF 606 MVI FZX430+@Q(,@BR),FZXER1 SET TYPE INDR FOR CHAR SPEC
2ED1 7B 80 F9 607 SBF FZXDTC(,@BR),FZXDTM SUPPRESS TYPE BIT IN SPEC CODE
608 *
2ED4 5F 00 F8 F7 609 FZX420 SLC FZXSTC(,@BR),FZX4B1(B@LCNN,@BR) DECR THE SPEC CODE COUNT
2ED8 D0 02 B5 610 BNM FZX400(,@BR) GO PROCESS NEW SPEC IF VALID
611 * IF NO MORE SPECIFICATION CODES
2EDB 7C F2 DF 612 MVI FZX430+@Q(,@BR),FZXER2 * SET TYPE INDR FOR END-OF-DATA
613 *
614 * TYPE SPECIFICATION ROUTINE EXIT - SET TYPE SPEC AND RETURN
615 *
2EDE 3C 00 0CBC 616 FZX430 MVI I$ERRC,*-* SET DATA TYPE SPECIFICATION
617 *
2EE2 C0 87 0000 618 FZX440 B *-* RETURN TO CALLING ROUTINE
619 *

```

[illegible]

DFPRNT - MATRIX PRINTER ROUTINE

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 31/05/21 PAGE 176
		622		*****	
		623		* DATA CHARACTER 'SET' ROUTINE -	*
		624		* * ADVANCES DATA BUFFER POINTER (REG @XR) TO NEYT NON-BLANK CHAR	*
		625		*****	
		626		*	
2EE6	74 08 F5	627	FZX450 ST	FZX470+@OP1(,@BR),@ARR	SET RETURN BRANCH ADDRESS
		628		*	
2EE9	E2 02 01	629	FZX460 LA	@B1(,@XR),@XR	INCR DATA CHARACTER POINTER
2EEC	BD 40 00	630		CLI B@CHAR(,@XR),B@BLNK	TEST FOR A BLANK CHARACTER
2EEF	D0 81 E9	631		BE FZX460(,@BR)	REPEAT LOOP UNTIL NON-BLANK
		632		*	
2EF2	C0 87 0000	633	FZX470 B	*-*	RETURN TO CALLING ROUTINE
		634		*	
		635		*****	

DFPRNT - MATRIX PRINTER ROUTINE

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 31/05/21 PAGE 177
			637	*****	
			638	* INPUT EXECUTION ROUTINE CONSTANTS (4TH VM PAGE)	*
			639	*****	
2EF6 0001		2EF7	640	*	
			641	FZX4B1 DC IL2'1' BINARY INTEGER +1	
			643	*****	
			644	* INPUT EXECUTION ROUTINE WORK AREAS (4TH VM PAGE)	*
			645	*****	
			646	*	
2EF8		2EC0	647	FZXSTP EQU FZX410+@DOP2	TEMPORARY STACK POINTER
		2EF8	648	FZXSTC DS CL(B@LCNN)	STACKED SPEC CODE COUNTER
			649	*	
2EF9		2EF9	650	FZXDTC DS CL(B@LCXX)	DATA TYPE CODE WORK AREA
		0080	651	FZXDTM EQU X'80'	SPEC CODE DATA TYPE MASK
			652	*	
		0D58	653	FZXETS EQU I\$WRK1-1	ELEMENT TYPE CHECKING SWITCH
			654	*	* CHECK - @BNE, NOCHECK - @NOP
		0D59	655	FZXSER EQU I\$WRK1	SYNTAX CHECKER ERROR CODE BYTE
		00FF	656	FZXSEC EQU X'FF'	SYNTAX CHECKER SPECIAL ERR CODE
			657	*	
		0D56	658	FZXCNT EQU I\$PARM-1	PROCESSED BUFFER ELEMENT COUNT
			659	*	
			660	*****	

## DFPRNT - MATRIX PRINTER ROUTINE

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	31/05/21	PAGE 178
					662	*****				
					663	*	VIRTUAL MEMORY INPUT EXECUTION ROUTINE (5TH VM PAGE) -			*
					664	*	ELEMENT SYNTAX CHECKING ROUTINE FOR FOLLOWING DATA TYPES -			*
					665	*	* ARITHMETIC CONSTANTS (I, F, E FORMATS)			*
					666	*	* BASIC INTERNAL CONSTANTS			*
					667	*	* CHARACTER CONSTANTS			*
					668	*				*
					669	*	INPUT -			*
					670	*	* I\$ERRC - SET TO X'F0' FOR EXPECTED ARITHMETIC CONSTANT			*
					671	*	- SET TO X'F1' FOR EXPECTED CHARACTER CONSTANT			*
					672	*	* FZXETS - SET TO @BNE WHEN CONSTANT TYPE IS TO BE CHECKED			*
					673	*	- SET TO @NOP WHEN CONSTANT TYPE NOT TO BE CHECKED			*
					674	*	* FZXSER - INITIALIZED TO CODE I@NERR (X'00')			*
					675	*	* REGISTER @XR - REFERENCES 1ST CHARACTER OF CONSTANT			*
					676	*				*
					677	*	OUTPUT -			*
					678	*	* FZXSER - SET TO CODE X'FF' WHEN SYNTAX/TYPE ERROR OCCURS			*
					679	*	* REGISTER @XR - REFERENCES DELIMITER IMMEDIATELY FOLLOWING			*
					680	*	CONSTANT WHEN NO SYNTAX/TYPE ERROR HAS BEEN ENCOUNTERED			*
					681	*****				*
					682	*				*
					683	*	ESTABLISH ADDRESSABILITY FOR INPUT ROUTINE (5TH VM PAGE)			*
					684	*				*
					685	*FZXP5B	VPAGE 0	SET 5TH PAGE ADDRESSABILITY		
2F00					686	ORG	*,256,0	SET STARTING ADDRESS		
				2F00	687	FZXP5B	EQU *	START OF PROGRAM CODING		
2E01					688	ORG	*-255	RESET IAR TO PAGE		
2F00					689	ORG	*,256,0	* BOUNDARY ADDRESS		
				2F00	690	USING	*,@BR	SET PAGE BASE ADDRESS		
2F00					691	ORG	FZXP5B	RESET STARTING ADDRESS		
					692	***	END OF EXPANSION ***			
					693	*				
					694	*	ELEMENT SYNTAX CHECKER ENTRY - INITIALIZE DATA TYPE CHECKING			
					695	*				
2F00	4C	00	15	0D58	696	FZX500	MVC FZX510+@Q(,@BR),FZXETS(1)	SET ARITHMETIC TYPE CHECKING		
2F05	4C	00	3A	0D58	697		MVC FZX540+@Q(,@BR),FZXETS(1)	SET CHARACTER TYPE CHECKING '		
					698	*				
					699	*	TEST FOR CHARACTER CONSTANT PROCESSING			
					700	*				
2F0A	BD	7D	00		701		CLI B@CHAR(,@XR),B@SQUO	IF 1ST CHARACTER IS SINGLE QUOTE		
2F0D	F2	81	25		702		JE FZX530	* GO PERFORM CHAR CONSTANT CHECK		
					703	*				
					704	*	ARITHMETIC CONSTANT ASSUMED - PERFORM ARITHMETIC TYPE CHECK			
					705	*				
2F10	3D	F0	0CBC		706		CLI I\$ERRC,FZXER0	IF ARITH CONSTANT NOT EXPECTED		
2F14	F2	00	79		707	FZX510	JC FZX590,*-*	* GO EXIT ON DATA TYPE ERROR -		
					708	*		* TEST IS DISABLED FOR GET(CARD)		
					709	*				
					710	*	ARITHMETIC CONSTANT ALLOWABLE - TEST FOR NUMERIC OR INTERNAL			
					711	*				
2F17	BD	4E	00		712		CLI B@CHAR(,@XR),B@PLUS	IF 1ST CHARACTER IS PLUS SIGN		
2F1A	F2	81	03		713		JE FZX520	* BRANCH TO GET NEXT CHARACTER		
2F1D	BD	60	00		714		CLI B@CHAR(,@XR),B@MINS	IF 1ST CHARACTER IS MINUS SIGN		
2F20	D0	81	98		715	FZX520	BE FZX610(,@BR)	* LINK TO GET NEXT DATA CHAR		
2F23	BD	50	00		716		CLI B@CHAR(,@XR),B@ICON	IF INTERNAL CONSTANT IDENTIFIER		
2F26	F2	81	34		717		JE FZX560	* FOUND, GO TEST REST OF SYMBOL		

718	*
719	*****

DFPRNT - MATRIX PRINTER ROUTINE

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	31/05/21	PAGE 180
					721		*****			
					722	*	NUMERIC CONSTANT SYNTAX CHECKING			*
					723		*****			
					724	*				
					725	*	CALL SYNTAX CHECKER FOR ASSUMED NUMERIC CONSTANT			
					726	*				
	2F29	76	02	A9	727	A	FZX5M1(,@BR),@XR			DECREMENT THE DATA POINTER
					728	*				
	2F2C	C0	87	12B1	729	B	I\$CALL			LINK TO SYNTAX CHECK ASSUMED
	2F30	3000			730	DC	AL(@VADDR)(FZX650)			* NUMERIC ARITHMETIC CONSTANT
				2F31	731	*				
	2F32	F2	87	5F	732	J	FZX600			GO RETURN TO CALLING ROUTINE
					733	*				
					734		*****			



## DFPRNT - MATRIX PRINTER ROUTINE

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	31/05/21	PAGE 181
				736		*****				
				737	*	CHARACTER CONSTANT SYNTAX CHECKING				*
				738		*****				
				739	*					
				740	*	PERFORM CHARACTER TYPE CHECK				
				741	*					
2F35	3D	F1	0CBC	742	FZX530	CLI	I\$ERRC,FZXER1			IF CHAR CONSTANT NOT EXPECTED
2F39	F2	00	54	743	FZX540	JC	FZX590,*-*			* GO EXIT ON DATA TYPE ERROR -
				744	*					* TEST IS DISABLED FOR GET(CARD)
				745	*					
				746	*	CHARACTER CONSTANT ALLOWABLE - CHECK SYNTAX VALIDITY				
				747	*					
2F3C	E2	02	01	748	FZX550	LA	@B1(,@XR),@XR			INCREMENT THE DATA POINTER
2F3F	BD	1E	00	749		CLI	B@CHAR(,@XR),B@EOST			IF PREMATURE DATA TERMINATION
2F42	F2	81	4B	750		JE	FZX590			* GO SET SYNTAX ERROR AND RETURN
2F45	BD	7D	00	751		CLI	B@CHAR(,@XR),B@SQUO			IF CHARACTER IS NOT A QUOTE
2F48	D0	01	3C	752		BNE	FZX550(,@BR)			* BRANCH TO CONTINUE SCAN LOOP
2F4B	E2	02	01	753		LA	@B1(,@XR),@XR			INCREMENT THE DATA POINTER
2F4E	BD	7D	00	754		CLI	B@CHAR(,@XR),B@SQUO			IF QUOTE PAIR IS INDICATED
2F51	D0	81	3C	755		BE	FZX550(,@BR)			* BRANCH TO CONTINUE SCAN LOOP
				756	*					* ELSE LAST QUOTE ENDS CONSTANT
2F54	BD	40	00	757		CLI	B@CHAR(,@XR),B@BLNK			IF BLANK CHAR AFTER CORSTANT
2F57	D0	81	98	758		BE	FZX610(,@BR)			* LINK TO GET ELEMENT DELIMITER
2F5A	F2	87	37	759		J	FZX600			GO RETURN TO CALLING ROUTINE
				760	*					
				761		*****				

## DFPRNT - MATRIX PRINTER ROUTINE

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	31/05/21	PAGE 182
				763		*****				
				764	*	INTERNAL CONSTANT SYNTAX CHECKING				*
				765		*****				
				766	*					
				767	*	MOVE INTERNAL, CONSTANT SYMBOL TO WORK AREA				
				768	*					
	2F5D	5F	03 B9 B9	769	FZX560	SLC	FZXICW(,@BR),FZXICW(FZXICL,@BR) CLEAR AREA TO BINARY 0'S			
				770	*					
	2F61	D0	87 98	771	FZX570	B	FZX610(,@BR) LINK TO GET NEXT DATA CHARACTER			
	2F64	BD	C1 00	772		CLI	B@CHAR(,@XR),B@LETA IF CHAR NOT A LETTER OR DIGIT			
	2F67	F2	82 11	773		JL	FZX580 * GO TEST FOR SYMBOL VALIDITY			
	2F6A	5C	02 B9 B8	774		MVC	FZXICW(,@BR),FZXICW-1(FZXICL-1,@BR) SHIFT SYMBOL RIGHT			
	2F6E	6C	00 B6 00	775		MVC	FZXICC(,@BR),B@CHAR(1,@XR) MOVE NEW CHAR INTO WORK AREA			
	2F72	7D	00 B9	776		CLI	FZXICR(,@BR),@ZERO IF SYMBOL WORK AREA NOT FILLED			
	2F75	D0	81 61	777		BE	FZX570(,@BR) * LOOP TO GET NEXT SYMBOL CHAR			
	2F78	D0	87 98	778		B	FZX610(,@BR) * ELSE LINK TO GET DELIMITER			
				779	*					
				780	*	CHECK INTERNAL CONSTANT SYMBOL FOR VALID SYNTAX				
				781	*					
	2F7B	5D	03 B9 AD	782	FZX580	CLC	FZXICW(,@BR),FZXIEX(FZXICL,@BR) IF SYMBOL IS &E			
	2F7F	F2	81 12	783		JE	FZX600 * GO RETURN TO CALLER			
	2F82	5D	03 B9 B1	784		CLC	FZXICW(,@BR),FZXIPI(FZXICL,@BR) IF SYMBOL IS &PI			
	2F86	F2	81 0B	785		JE	FZX600 * GO RETURN TO CALLER			
	2F89	5D	03 B9 B5	786		CLC	FZXICW(,@BR),FZXIS2(FZXICL,@BR) IF SYMBOL IS &SQR2			
	2F8D	F2	81 04	787		JE	FZX600 * GO RETURN TO CALLER			
				788	*					
				789	*	EXIT - SET ERROR CONDITION CODE IF INVALID SYNTAX OR TYPE ENCOUNTERED				
				790	*					
	2F90	3C	FF 0D59	791	FZX590	MVI	FZXSER,FZXSEC SET ERROR CODE FOR INVALIDITY			
				792	*					
	2F94	C0	87 12D3	793	FZX600	B	I\$RTRN RETURN TO CALLING ROUTINE			
				794	*					
				795		*****				

DFPRNT - MATRIX PRINTER ROUTINE

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	31/05/21	PAGE 183
				797		*****				
				798	*	DATA CHARACTER 'GET' ROUTINE -				*
				799	*	* ADVANCES DATA BUFFER POINTER (REG @XR TO NEXT NON-BLANK CHAR				*
				800		*****				
				801	*					
2F98	74	08	A7	802	FZX610	ST	FZX630+@OP1(,@BR),@ARR	SET RETURN BRANCH ADDRESS		
				803	*					
2F9B	E2	02	01	804	FZX620	LA	@B1(,@XR),@XR	INCR DATA CHARACTER POINTER		
2F9E	BD	40	00	805		CLI	B@CHAR(,@XR),B@BLNK	TEST FOR A BLANK CHARACTER		
2FA1	D0	81	9B	806		BE	FZX620(,@BR)	REPEAT LOOP UNTIL NON-BLANK		
				807	*					
2FA4	C0	87	0000	808	FZX630	B	*-*	RETURN TO CALLING ROUTINE		
				809	*					
				810		*****				

DFPRNT - MATRIX PRINTER ROUTINE

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	31/05/21	PAGE 184
					812	*****	*****			
					813	* INPUT EXECUTION ROUTINE CONSTANTS (5TH VM PAGE)	*			
					814	*****	*****			
					815	*				
	2FA8	FFFF		2FA9	816	FZX5M1 DC	IL2'-1'			BINARY INTEGER -1
					817	*				
				0004	818	FZXICL EQU	4			LENGTH OF INT CON TEST SYMBOL
					819	*				
	2FAA	C5		2FAA	820		DC CL1'E'			TEST SYMBOL FOR &E
	2FAB	000000		2FAD	821	FZXIEX DC	XL(FZXICL-1)'00'			SYMBOL CONSTANT FILLER
					822	*				
	2FAE	C9D7		2FAF	823		DC CL2'IP'			TEST SYMBOL FOR &PI
	2FB0	0000		2FB1	824	FZXIPI DC	XL(FZXICL-2)'00'			SYMBOL CONSTANT FILLER
					825	*				
	2FB2	F2D9D8E2		2FB5	826	FZXIS2 DC	CL4'2RQS'			TEST SYMBOL FOR &SQR2
					828	*****	*****			
					829	* INPUT EXECUTION ROUTINE WORK AREAS (5TH VM PAGE)	*			
					830	*****	*****			
					831	*				
				2FB6	832	FZXICC EQU	*			CURR INTERNAL CON SYMBOL CHAR
	2FB6			2FB9	833	FZXICW DS	CL(FZXICL)			INTERNAL SYMBOL WORK AREA
				2FB9	834	FZXICR EQU	FZXICW			SYMBOL WORK AREA RIGHT BYTE ADDR
					835	*				
					836	*****	*****			

## DFPRNT - MATRIX PRINTER ROUTINE

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 31/05/21 PAGE 185
		838		*****	
		839		* VIRTUAL MEMORY INPUT EXECUTION ROUTINE (6TH VM PAGE)	*
		840		* ELEMENT SYNTAX CHECKING ROUTINE FOR NUMERIC ARITHMETIC CONSTANTS	*
		841		*	*
		842		* INPUT -	*
		843		* * FZXSER - INITIALIZED TO CODE I@NERR (X'00')	*
		844		* * REGISTER @XR - REFERENCES CHARACTER PRECEDING 1ST NON-SIGN CHAR	*
		845		*	*
		846		* OUTPUT -	*
		847		* * FZXSER - SET TO CODE X'FF' WHEN SYNTAX ERROR IS FOUND	*
		848		* * REGISTER @XR - REFERENCES DELIMITER IMMEDIATELY FOLLOWING	*
		849		* CONSTANT WHEN NO SYNTAX ERROR HAS BEEN ENCOUNTERED	*
		850		*****	
		851		*	
		852		* ESTABLISH ADDRESSABILITY FOR INPUT ROUTINE (6TH VM PAGE)	
		853		*	
3000		854		*FZXP6B VPAGE 0 SET 6TH PAGE ADDRESSABILITY	
		855		ORG *,256,0 SET STARTING ADDRESS	
2F01		856		FZXP6B EQU * START OF PROGRAM CODING	
3000		857		ORG *-255 RESET IAR TO PAGE	
		858		ORG *,256,0 * BOUNDARY ADDRESS	
		859		USING *,@BR SET PAGE BASE ADDRESS	
3000		860		ORG FZXP6B RESET STARTING ADDRESS	
		861		*** END OF EXPANSION ***	
		862		*	
		863		* ENTRY - INITIALIZE FOR NUMERIC CONSTANT SYNTAX CHECKING	
		864		*	
3000 5C 01 EA E4		865		FZX650 MVC FZXXCT(,@BR),FZXXZR(FZXXCL,@BR) SET 'ZERO' EXPONENT COUNT	
3004 7C 80 30		866		MVI FZX690+@Q(,@BR),@NOP SET CONSTANT FRACTION SWITCH ON	
3007 7C 87 49		867		MVI FZX720+@Q(,@BR),@UCB SET CONSTANT DIGIT SWITCH OFF	
300A 7C 87 B0		868		MVI FZX800+@Q(,@BR),@UCB SET MANTISSA ZERO SWITCH ON	
		869		*	
		870		*****	

## DFPRNT - MATRIX PRINTER ROUTINE

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00 31/05/21 PAGE 186
				872		*****		
				873		* NUMERIC CONSTANT MANTISSA SYNTAX CHECKING		*
				874		*****		
				875		*		
				876		* INCREMENT PAST INSIGNIFICANT LEADING ZEROS		
				877		*		
300D	D0	87	C8	878	FZX660	B	FZX830(,@BR)	LINK TO GET NEXT DATA CHARACTER
3010	D0	81	0D	879		BE	FZX660(,@BR)	LOOP IF CHARACTER IS DECML ZERO
3013	F2	82	10	880		JL	FZX680	BRANCH IF CHARACTER NOT DECIMAL
				881		*		
				882		* PROCESS INTEGER DIGITS - INCREMENT EXPONENT COUNT FOR EACH DIGIT		
				883		*		
3016	7C	87	30	884	FZX670	MVI	FZX690+@Q(,@BR),@UCB	SET CONSTANT FRACTION SWITCH OFF
3019	7C	80	B0	885		MVI	FZX800+@Q(,@BR),@NOP	SET MANTISSA ZERO SWITCH OFF
				886		*		
301C	5E	01	EA E2	887	FZX675	ALC	FZXXCT(,@BR),FZX6B1(FZXXCL,@BR)	INCREMENT EXPONENT COUNT
3020	D0	87	C8	888		B	FZX830(,@BR)	LINK TO GET NEXT DATA CHARACTER
3023	D0	02	1C	889		BNL	FZX675(,@BR)	LOOP IF CHARACTER IS A DECIMAL
				890		*		
				891		* TEST FOR A DECIMAL POINT OR MANTISSA DELIMITER		
				892		*		
3026	BD	4B	00	893	FZX680	CLI	B@CHAR(,@XR),B@DPNT	IF CHARACTER NOT A DECIMAL POINT
3029	F2	01	1C	894		JNE	FZX720	* GO CHECK FOR AT LEAST 1 DIGIT
302C	F2	87	07	895		J	FZX700	* ELSE SKIP TO TEST FRACTIONALS
				896		*		
				897		* PROCESS LEADING FRACTIONAL ZEROS - DECREMENT EXPONENT COUNT FOR		
				898		* EACH ZERO WHEN CONSTANT CONTAINS NO SIGNIFICANT INTEGER DIGITS		
				899		*		
302F	F2	00	04	900	FZX690	JC	FZX700,*-*	BRANCH IF ANY INTEGER COMPONENT
3032	5F	01	EA E2	901		SLC	FZXXCT(,@BR),FZX6B1(FZXXCL,@BR)	DECREMENT EXPONENT COUNT
3036	D0	87	C8	902	FZX700	B	FZX830(,@BR)	LINK TO SET NEXT DATA CHARACTER
3039	D0	81	2F	903		BE	FZX690(,@BR)	LOOP IF CHARACTER IS DECML ZERO
303C	F2	82	09	904		JL	FZX720	BRANCH IF CHARACTER NOT DECIMAL
				905		*		
				906		* INCREMENT PAST TRAILING FRACTIONAL DIGITS		
				907		*		
303F	7C	80	B0	908		MVI	FZX800+@Q(,@BR),@NOP	SET MANTISSA ZERO SWITCH OFF
				909		*		
3042	D0	87	C8	910	FZX710	B	FZX830(,@BR)	LINK TO GET NEXT DATA CHARACTER
3045	D0	02	42	911		BNL	FZX710(,@BR)	LOOP IF CHARACTER IS A DECIMAL
				912		*		
				913		* TEST FOR NO MANTISSA DIGITS (AN ERROR CONDITION)		
				914		*		
3048	F2	00	75	915	FZX720	JC	FZX810,*-*	BRANCH IF NO 401\$\$A DIGITS
				916		*		
				917		*****		

## DFPRNT - MATRIX PRINTER ROUTINE

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00 31/05/21 PAGE 187
				919			*****	
				920			* NUMERIC CONSTANT EXPONENT SYNTAX CHECKING	*
				921			*****	
				922			*	
				923			* TEST FOR E-FORMAT SPECIFICATION	
				924			*	
304B	BD	C5	00	925		CLI	B@CHAR(,@XR),B@EXPC	IF NO E-FORMAT SPECIFICATION
304E	F2	01	5E	926		JNE	FZX800	* BRANCH TO TEST VALUE MAGNITUDE
				927			*	
				928			* E-FORMAT - ESTABLISH THE EXPONENT SIGN	
				929			*	
3051	7C	80	A2	930		MVI	FZX780+@Q(,@BR),@NOP	SET POSITIVE EXPONENT STATUS
3054	D0	87	C8	931		B	FZX830(,@BR)	LINK TO GET NEXT DATA CHARACTER
3057	BD	4E	00	932		CLI	B@CHAR(,@XR),B@PLUS	IF EXPONENT PLUS SIGN SPECIFIED
305A	F2	81	09	933		JE	FZX730	* SKIP TO GET THE NEXT CHARACTER
305D	BD	60	00	934		CLI	B@CHAR(,@XR),B@MINS	IF EXP MINUS SIGN NOT SPECIFIED
3060	F2	01	06	935		JNE	FZX740	* SO PROCESS EXPONENT DIGITS
3063	7C	87	A2	936		MVI	FZX780+@Q(,@BR),@UCB	* ELSE SET NEGATIVE EXP STATUS
3066	D0	87	C8	937	FZX730	B	FZX830(,@BR)	LINK TO GET CHARACTER AFTER SIGN
				938			*	
				939			* TEST FOR NO EXPONENT DIGITS (AN ERROR CONDITION)	
				940			*	
3069	BD	F0	00	941	FZX740	CLI	B@CHAR(,@XR),B@DEC0	IF NO EXPONENT DIGIT FOUND
306C	F2	82	51	942		JL	FZX810	* BRANCH TO SET SYNTAX ERROR
				943			*	
				944			* ESTABLISH EXPONENT DIGIT(S) FOR CONVERSION TO BINARY	
				945			*	
306F	64	10	EC 00	946		ZAZ	FZX6DX(FZDXL,@BR),B@CHAR(1,@XR)	SAVE 1ST EXPONENT DIGIT
3073	D0	87	C8	947		B	FZX830(,@BR)	LINK TO GET NEXT DATA CHARACTER
3076	F2	82	0B	948		JL	FZX750	BRANCH IF CHAR NOT A DECIMAL
3079	5C	00	EB EC	949		MVC	FZX6DX-1(,@BR),FZX6DX(1,@BR)	SHIFT 1ST EXPONENT DIGIT AND
307D	6C	00	EC 00	950		MVC	FZX6DX(,@BR),B@CHAR(1,@XR)	* STORE NEW DIGIT IN UNITS POS
3081	D0	87	C8	951		B	FZX830(,@BR)	LINK TO GET CHAR AFTER EXPONENT
				952			*	
				953			* CONVERT EXPONENT DECIMAL DIGITS TO BINARY	
				954			*	
3084	7B	F0	EC	955	FZX750	SBF	FZX6DX(,@BR),B@ZPOS	SUPPRESS UNITS DIGIT ZONE BITS
3087	7B	F0	EB	956		SBF	FZX6DX-1(,@BR),B@ZPOS	SUPPRESS TENS DIGIT ZONE BITS
308A	5E	00	EB EB	957		ALC	FZX6DX-1(,@BR),FZX6DX-1(1,@BR)	DOUBLE TENS DIGIT FOR 2X
308E	5E	00	EC EB	958		ALC	FZX6DX(,@BR),FZX6DX-1(1,@BR)	ADD 2 X TM TO 1.NITS PCS
3092	5E	00	EB EB	959		ALC	FZX6DX-1(,@BR),FZX6DX-1(1,@BR)	DOUBLE TENS DIGIT FOR 4X
3096	5E	00	EB EB	960		ALC	FZX6DX-1(,@BR),FZX6DX-1(1,@BR)	DOUBLE TENS DIGIT FOR 8X
309A	5E	00	EC EB	961		ALC	FZX6DX(,@BR),FZX6DX-1(1,@BR)	ADD I 1 ?T\S 101441T\$ P05
309E	7C	00	EB	962		MVI	FZX6DX-1(,@BR),@ZERO	ZERO TO FORM 2-BYTE BINARY NO.
				963			*	
				964			* MODIFY EXPONENT COUNT WITH SPECIFIED EXPONENT	
				965			*	
30A1	F2	00	07	966	FZX780	JC	FZX790,*-*	BRANCH IF SPEC EXP IS NEGATIVE
30A4	5E	01	02 EC	967		ALC	FZXXCL(,@BR),FZX6BX(FZXXCL,@BR)	ADD SPECIFIED EXPONENT
30A8	F2	87	04	968		J	FZX800	* TO COUNT AND GO TEST RANGE
30AB	5F	01	EA EC	969	FZX790	SLC	FZXXCT(,@BR),FZX6BX(FZXXCL,@BR)	SUBTRACT SPEC EXPONENT
				970			*	
				971			* TEST FOR ZERO MANTISSA - BYPASS CONSTANT RANGE ERROR CHECK WHEN	
				972			* MANTISSA IS ZERO, SINCE EXPONENT WILL ALWAYS BE SET TO E-98	
				973			*	
30AF	F2	00	12	974	FZX800	JC	FZX820,*-*	BRANCH IF MANTISSA IS ZERO



DFPRNT - MATRIX PRINTER ROUTINE

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 31/05/21 PAGE 188
			975	*	
			976	* TEST FOR CONSTANT RANGE ERROR - COMBINED EXPONENT MUST FALL WITHIN	
			977	* RANGE (128-98) TO (128+99) ... 'ZERO' EXPONENT IS 128	
			978	*	
30B2	5D 01 EA E6		979	CLC FZXXCT(,@BR),FZXXHI(FZXXCL,@BR) IF EXPONENT EXCEEDS HIGH	
30B6	F2 84 07		980	JH FZX810 * BOUNDARY, GO EXIT ON ERROR	
30B9	5D 01 EA E8		981	CLC FZXXCT(,@BR),FZXXLO(FZXXCL,@BR) IF EXPONENT NOT LESS THAN	
30BD	F2 02 04		982	JNL FZX820 * LOW BOUNDARY, RETURN TO CALLER	
			983	*	
			984	* EXIT - SET ERROR CODE IF INVALID SYNTAX ENCOUNTERED	
			985	*	
30C0	3C FF 0D59		986	FZX810 MVI FZXSER,FZXSEC SET ERROR CODE FOR INVALIDITY	
			987	*	
30C4	C0 87 12D3		988	FZX820 B I\$RTRN RETURN TO CALLING ROUTINE	
			989	*	
			990	*****	

## DFPRNT - MATRIX PRINTER ROUTINE

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 31/05/21 PAGE 189
		992		*****	
		993	*	DATA CHARACTER 'GET' ROUTINE -	*
		994	*	* ADVANCES DATA BUFFER POINTER (REG @XR) TO NEXT NON-BLANK CHAR	*
		995	*	* SETS PSR DEPENDING ON CHARACTER TYPE -	*
		996	*	* LOW - NON-DECIMAL CHARACTER	*
		997	*	* EQUAL - DECIMAL ZERO CHARACTER	*
		998	*	* HIGH - DECIMAL DIGIT OTHER THAN ZERO	*
		999	*	* SETS CONSTANT DIGIT SWITCH ON WHEN DIGIT ENCOUNTERED	*
		1000		*****	
		1001	*		
30C8	74 08 E0	1002	FZX830 ST	FZX850+@OP1(,@BR),@ARR	SET RETURN BRANCH ADDRESS
		1003	*		
30CB	E2 02 01	1004	FZX840 LA	@B1(,@XR),@XR	INCR DATA CHARACTER POINTER
30CE	BD 40 00	1005	CLI	B@CHAR(,@XR),B@BLNK	TEST FOR A BLANK CHARACTER
30D1	D0 81 CB	1006	BE	FZX840(,@BR)	REPEAT LOOP UNTIL NON-BLANK
		1007	*		
30D4	BD F0 00	1008	CLI	B@CHAR(,@XR),B@DEC0	IF CHARACTER NOT A DECIMAL DIGIT
30D7	F2 82 03	1009	JL	FZX850	* GO RETURN TO CALLING ROUTINE
30DA	7C 80 49	1010	MVI	FZX720+@Q(,@BR),@NOP	* ELSE SET CONSTANT DIGIT SW ON
		1011	*		
30DD	C0 87 0000	1012	FZX850 B	*-*	RETURN TO CALLMS MANE
		1013	*		
		1014		*****	

DFPRNT - MATRIX PRINTER ROUTINE

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	31/05/21	PAGE 190
					1016	*****	*****			
					1017	* INPUT EXECUTION ROUTINE CONSTANTS (6TH VM PAGE)	*			
					1018	*****	*****			
					1019	*				
30E1	0001			30E2	1020	FZX6B1 DC	IL2'1' BINARY INTEGER +1			
					1021	*				
				0002	1022	FZXXCL EQU	2 LENGTH OF EXPONENT TEST COUNT			
30E3	0180			30E4	1023	FZXZR DC	AL(FZXXCL)(256+B@NXZR) TEST NORM EXPONENT - ZERO			
30E5	01E3			30E6	1024	FZXHI DC	AL(FZXXCL)(256+B@NXHI) TEST NORM EXPONENT - MAXIMUM			
30E7	011E			30E8	1025	FZXLO DC	AL(FZXXCL)(256+B@NXLO) TEST NORM EXPONENT - MINIMUM			
					1027	*****	*****			
					1028	* INPUT EXECUTION ROUTINE WOTK AREAS (6TH VM PAGE)	*			
					1029	*****	*****			
					1030	*				
30E9				30EA	1031	FZXCT DS	CL(FZXXCL) EXPONENT TEST COUNT AREA			
					1032	*				
				0002	1033	FZDXL EQU	2 LENGTH OF MAX DECIMAL EXPONENT			
30EB				30EC	1034	FZX6DX DS	CL(FZDXL) DECIMAL EXPONENT WORK AREA			
				30EC	1035	FZX6BX EQU	FZX6DX BINARY SPECIFIED EXPONENT			
					1036	*				
					1037	*****	*****			

## DFPRNT - MATRIX PRINTER ROUTINE

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 31/05/21 PAGE 191
		1039		*****	
		1040	*	VIRTUAL MEMORY INPUT EXECUTION ROUTINE (7TH VM PAGE) -	*
		1041	*	ELEMENT CONVERSION ROUTINE FOR FOLLOWING DATA TYPES -	*
		1042	*	* ARITHMETIC CONSTANTS (I, F, E FORMATS)	*
		1043	*	* BASIC INTERNAL CONSTANTS	*
		1044	*	* CHARACTER CONSTANTS	*
		1045	*		*
		1046	*	INPUT -	*
		1047	*	* REGISTER @XR - REFERENCES LEFTMOST BYTE OF FULL PAGE BUFFER	*
		1048	*	* DATA POINTER LOCATED IN LAST 2 BYTES OF BUFFER - POINTER	*
		1049	*	REFERENCES CHARACTER PRECEDING 1ST CHARACTER OF CONSTANT.	*
		1050	*		*
		1051	*	OUTPUT -	*
		1052	*	* CONVERTED ELEMENT STACKED BEGINNING AT I\$STAK - STACKED DATA	*
		1053	*	FORMAT IS PACKED FLOATING POINT (CURRENT PRECISION) OR	*
		1054	*	19-BYTE CHARACTER ELEMENT.	*
		1055	*	* REGISTER @XR - REFERENCES DELIMITER FOLLOWING CONVERTED CON.	*
		1056	*	* DATA POINTER IN BUFFER UPDATED TO REFERENCE THE CON DELIMITER	*
		1057		*****	
		1058	*		
		1059	*	ESTABLISH ADDRESSABILITY FOR INPUT ROUTINE 7TH VM PAGE	
		1060	*		
3100		1061	*FZXP7B	VPAGE 0	SET 7TH PAGE ADDRESSABILITY
		1062	ORG	*,256,0	SET STARTING ADDRESS
	3100	1063	FZXP7B	EQU *	START OF PROGRAM CODING
3001		1064	ORG	*-255	RESET IAR TO PAGE
3100		1065	ORG	*,256,0	* BOUNDARY ADDRESS
	3100	1066	USING	*,@BR	SET PAGE BASE ADDRESS
3100		1067	ORG	FZXP7B	RESET STARTING ADDRESS
		1068	***	END OF EXPANSION ***	
		1069	*		
		1070		*****	

## DFPRNT - MATRIX PRINTER ROUTINE

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	31/05/21	PAGE 192
					1072	*****				
					1073	* ELEMENT CONVERSION ENTRY - ACCESS 1ST CHARACTER OF CONSTANT				*
					1074	*****				
					1075	*				
				3100	1076	FZXCNV EQU *	INPUT DATA CONVERSION ENTRT PT			
					1077	*				
3100	74	02	B7		1078	FZX860 ST FZX906+@OP1(,@BR),@XR	SAVE OUFF1R_LEFT BYTE CORE CDR			
3103	B5	02	FF		1079	L FZXBPT(,@XR),@XR	LOAD THE BUFFER DATA POINTER			
3106	D0	87	BF		1080	B FZX910(,@BR)	LINK TO GET NUT DATA CHARACTER			
					1081	*				
					1082	* TEST FOR CHARACTER CONSTANT PROCESSING				
					1083	*				
3109	BD	7D	00		1084	CLI B@CHAR(,@XR),B@SQUO	IF 1ST CHARACTER IS SINGLE COTE			
310C	F2	81	2B		1085	JE FZX873	* GO PERFORM CHAR CON CONVERSION			
					1086	*				
					1087	* TEST FOR NUMERIC OR INTERNAL ARITHMETIC CONSTANT - PERFORM SIGN				
					1088	* PROCESSING FOR THE POSSIBLE INTERNAL CONSTANT SYMBOL				
					1089	*				
310F	74	02	30		1090	ST FZX870+@OP1(,@BR),@XR	SAVE CON 1ST CHIM IN CASE OF NUM			
3112	7C	4E	CF		1091	MVI FZXICB-1(,@BR),B@PLUS	SET POSITIVE INTERNAL CON INDR			
3115	BD	4E	00		1092	CLI B@CHAR(,@XR),B@PLUS	IF 1ST CHARACTER IS PLUS SIGN			
3118	F2	81	09		1093	JE FZX863	* BRANCH TO GET NEXT CHARACTER			
311B	BD	60	00		1094	CLI B@CHAR(,@XR),B@MINS	IF 1ST CHARACTER NOT MINUS			
311E	F2	01	06		1095	JNE FZX866	* BRANCH TO TEST CONSTANT TYPE			
3121	7C	60	CF		1096	MVI FZXICB-1(,@BR),B@MINS	* ELSE SET NEG INTERNAL CON INDR			
					1097	*				
3124	D0	87	BF		1098	FZX863 B FZX910(,@BR)	LINK TO GET NEXT DATA CHARACTER			
3127	BD	50	00		1099	FZX866 CLI B@CHAR(,@XR),B@ICON	IF INTERNAL CONSTANT IDENTIFIER			
312A	F2	81	56		1100	JE FZX893	* FOUND, GO TEST REST OF SYMBOL			
					1101	*				
					1102	*****				

DFPRNT - MATRIX PRINTER ROUTINE

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	31/05/21	PAGE 193
					1104	*****				
					1105	* NUMERIC ARITHMETIC CONSTANT CONVERSION				*
					1106	*****				
					1107	*				
					1108	* CALL FLOATING POINT CONVERSION/STACKINS ROUTINE				
					1109	*				
	312D	C2	01	0000	1110	FZX870 LA	*-*,@BR			LOAD CADDR OF CONSTANT 1ST CHAR
					1111	*				
	3131	C0	87	12B1	1112	B	I\$CALL			LINK TO CONVERT AND STACK THE
	3135	3200			1113	DC	AL(@VADDR)(FZX920)			* CONSTANT AS PACKED FLOATING PT
				3136	1114	*				
	3137	F2	87	7A	1115	J	FZX906			GO EXIT CONVERSION ROUTINE
					1116	*				
					1117	*****				

## DFPRNT - MATRIX PRINTER ROUTINE

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00 31/05/21 PAGE 194
				1119		*****		
				1120		* CHARACTER CONSTANT CONVERSION AND STACKING		*
				1121		*****		
				1122		*		
				1123		* INITIALIZE THE CONSTANT BUCKET FOR CHARACTER PROCESSING.		
				1124		*		
313A	7C	40	CF	1125	FZX873	MVI	FZXSTS(, @BR), B@DTYP	SET BUCKET STATUS FOR CHAR CON
313D	5C	11	E1 E2	1126		MVC	FZXCRR(, @BR), FZXBLK(I@LCRF, @BR)	BLANK THE CHAR DATA FIELD
3141	7C	D0	62	1127		MVI	FZX883+@D1(, @BR), FZXCRI-FZXP7B	INITLZ CHAR DATA FLD DISP
				1128		*		
				1129		* ACCESS NEXT CHARACTER - TEST FOR PAIRED QUOTES OR END OF CONSTANT		
				1130		*		
3144	E2	02	01	1131	FZX876	LA	@B1(, @XR), @XR	INCREMENT THE DATA POINTER
3147	BD	7D	00	1132		CLI	B@CHAR(, @XR), B@SQUO	IF CNACharacter IS NOT A QUOTE
314A	F2	01	09	1133		JNE	FZX880	* BRANCH TO PROCESS CHARACTER
314D	E2	02	01	1134		LA	@B1(, @XR), @XR	INCREMENT THE DATA POINTER
3150	BD	7D	00	1135		CLI	B@CHAR(, @XR), B@SQUO	IF QUOTE PAIR NOT SPECIFIED GO
3153	F2	01	15	1136		JNE	FZX886	* EXECUTE END OF CONSTANT RTN
				1137		*		
				1138		* PROCESS THE DATA CHARACTER - STORE IN CONSTANT BUCKET WHEN BUCKET		
				1139		* IS NOT FILLED, OTHERWISE DISREGARD THE CHARACTER.		
				1140		*		
3156	7D	E1	62	1141	FZX880	CLI	FZX883+@D1(, @BR), FZXCRR-FZXP7B	IF BUCKET ALREADY FILLED
3159	D0	84	44	1142		BH	FZX876(, @BR)	IT BYPASS CHAR AND GO PROC NEXT
				1143		*		
315C	5E	00	CF C4	1144		ALC	FZXSTS(, @BR), FZX7B1(1, @BR)	INCREMENT STATUS BYTE COUNTER
3160	6C	00	00 00	1145	FZX883	MVC	*-(, @BR), B@CHAR(1, @XR)	MOVE CURRENT CHARACTER TO BUCKET
3164	5E	00	62 C4	1146		ALC	FZX883+@D1(, @BR), FZX7B1(1, @BR)	INCREMENT BUCKET POINTER
3168	D0	87	44	1147		B	FZX876(, @BR)	BRANCH TO PROCESS NEXT CHARACTER
				1148		*		
				1149		* END OR CONSTANT - ACCESS DELIMITER FOLLOWING CONSTANT		
				1150		*		
316B	BD	40	00	1151	FZX886	CLI	B@CHAR(, @XR), B@BLNK	IF BLANK AFTER CONSTANT
316E	D0	81	BF	1152		BE	FZX910(, @BR)	* LINK TO GET ELEMENT DELIMETER
				1153		*		
				1154		* MOVE CONVERTED CONSTANT TO THE RUN-TIME STACK		
				1155		*		
3171	74	02	7F	1156		ST	FZX890+@OP1(, @BR), @XR	SAVE THE DELIMETER CORE ADDRESS
3174	35	02	0D4E	1157		L	I\$STAK, @XR	LOAD THE RUN-TIME STACK POINTER
3178	9C	12	12 E1	1158		MVC	I@LCRV-1(, @XR), FZXCRR(I@LCRV, @BR)	STACK THE CHAR CONSTANT
317C	C2	02	0000	1159	FZX890	LA	*-*, @XR	RELOAD THE DELIMETER CORE ADDR
3180	F2	87	31	1160		J	FZX906	GO EXIT THE CONVERSION ROUTINE
				1161		*		
				1162		*****		



## DFPRNT - MATRIX PRINTER ROUTINE

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	31/05/21	PAGE 195
				1164		*****				
				1165		* INTERNAL ARITHMETIC CONSTANT STACKING				
				1166		*****				
				1167		*				
				1168		* INTERNAL CONSTANT BUCKET CONSISTS OF 2 BYTES, THE 1ST OF WHICH				
				1169		* CONTAINS THE CONSTANT SIGN CHARACTER - MOVE 1ST LETTER OF CONSTANT				
				1170		* SYMBOL TO THE 2ND BUCKET BYTE				
				1171		*				
3183	D0	87	BF	1172	FZX893	B	FZX910(,@BR)		LINK TO GET 1ST SYMBOL LETTER	
3186	6C	00	D0 00	1173		MVC	FZXICB(,@BR),B@CHAR(1,@XR)		MOVE 1ST LETTER TO THE BUCKET	
318A	74	02	AA	1174		ST	FZX900+@OP1(,@BR),@XR		SAVE CURRENT CHARACTER CORE ADDR	
				1175		*				
				1176		* SEARCH THE INTERNAL CONSTANT SYMBOL TABLE FOR MATCH WITH BUCKET				
				1177		*				
318D	D2	02	DF	1178		LA	FZXICT-FZXITL(,@BR),@XR		LOAD SYMBOL TABLE BASE ADDRESS	
3190	E2	02	04	1179	FZX896	LA	FZXITL(,@XR),@XR		INCR TABLE POINTER TO NEXT ENTRY	
3193	6D	01	D0 01	1180		CLC	FZXICB(,@BR),FZXICN(2,@XR)		COMPARE BUCKET WITH TBL ENTRY	
3197	D0	01	90	1181		BNE	FZX896(,@BR)		CONTINUE SEARCH IF NO ID MATCH	
				1182		*				
				1183		* MATCHNINA ENTRY FOUND - STACK THE INTERNAL CONSTANT LOCATED IN				
				1184		* VIRTUAL MEMORY AT THE ADDRESS SPECIFIED IN THE SYMBOL TABLE				
				1185		*				
319A	2C	01	144A 03	1186		MVC	I\$VADR,FZXICA(@VADDR,@XR)		SET PASING VIRTUAL ADDR PARAM	
319F	35	02	0D4E	1187		L	I\$STAK,@XR		LOAD THE RUN-TIME STACK POINTER	
31A3	C0	87	0B50	1188		B	I\$STCK		LINK TO STACK THE INTERNAL CON.	
				1189		*				
				1190		* ACCESS DELIMITER FOLLOWING INTERNAL CONSTANT SYMBOL				
				1191		*				
31A7	C2	02	0000	1192	FZX900	LA	*-*,@XR		LOAD THE DATA BUFFER POINTER	
31AB	D0	87	BF	1193	FZX903	B	FZX910(,@BR)		LINK TO SET NEXT DATA CHARACTER	
31AE	BD	C1	00	1194		CLI	B@CHAR(,@XR),B@LETA		IF CHAR IS A LETTER OR DIGIT	
31B1	D0	02	AB	1195		BNL	FZX903(,@BR)		* LOOP TO SET NEXT CHARACTER	
				1196		*				
				1197		* EXIT - SAVE DATA POINTER AND RETURN TO CALLING ROUTINE				
				1198		*				
31B4	C2	01	0000	1199	FZX906	LA	*-*,@BR		LOAD DATA BUFFER BASE CORE ADDR	
31B8	74	02	FF	1200		ST	FZXBPT(,@BR),@XR		STORE DATA POINTER IN BUFFER	
				1201		*				
31BB	C0	87	12D3	1202		B	I\$RTRN		RETURN TO CALLING ROUTINE	
				1203		*				
				1204		*****				

DFPRNT - MATRIX PRINTER ROUTINE

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	31/05/21	PAGE 196
				1206		*****				
				1207	*	DATA CHARACTER 'GET' ROUTINE -				*
				1208	*	* ADVANCES DATA BUFFER POINTER (REG @XR) TO NEXT NON-BLANK CHAR				*
				1209		*****				
				1210	*					
31BF	74	08	CE	1211	FZX910	ST	FZX916+@OP1(,@BR),@ARR	SET RETURN BRANCH ADDRESS		
				1212	*					
31C2	E2	02	01	1213	FZX913	LA	@B1(,@XR),@XR	INCR DATA CHARACTER POINTER		
31C5	BD	40	00	1214		CLI	B@CHAR(,@XR),B@BLNK	TEST FOR A BLANK CHARACTER		
31C8	D0	81	C2	1215		BE	FZX913(,@BR)	REPEAT LOOP UNTIL NON-BLANK		
				1216	*					
31CB	C0	87	0000	1217	FZX916	B	*-*	RETURN TO CALLING ROUTINE		
				1218	*					
				1219		*****				

DFPRNT - MATRIX PRINTER ROUTINE

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	31/05/21	PAGE 197
					1221	*****				
					1222	* INPUT EXECUTION ROUTINE CONSTANTS (7TH VM PAGE)				*
					1223	*****				
					1224	*				
				31C4	1225	FZX7B1 EQU FZX913+@D1	BINARY INTEGER +1			
					1226	*				
					1227	*****				
					1228	* INPUT EXECUTION ROUTINE WORK AREAS (7TH VM PAGE)				*
					1229	*****				
					1230	*				
				31CF	1231	FZXBKT EQU *	CONVERSION BUCKET BASE ADDRESS			
31CF				31E1	1232	DS CL(I@LCRV)	CONVERSION BUCKET AREA			
31E2	40			31E2	1233	FZXBLK DC AL1(B@BLNK)	BLANK INITIALIZATION CHAR			
					1234	*				
				31CF	1235	FZXSTS EQU FZXBKT+I@STAT	BUCKET STATUS BYTE ADDRESS			
				31D0	1236	FZXCR1 EQU FZXBKT+I@STAT+1	BUCKET 1ST CHAR FIELD BYTE ADDR			
				31E1	1237	FZXCRR EQU FZXBKT+I@LCRF	BUCKET LAST CHAR FIELD BYTE ADDR			
					1238	*				
				31D0	1239	FZXICB EQU FZXBKT+1	INTERNAL CONSTANT BUCKET ADDR			
					1240	*				
					1241	*****				

## DFPRNT - MATRIX PRINTER ROUTINE

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	31/05/21	PAGE 198
				1243		*****				
				1244		* INTERNAL CONSTANT SYMBOL TABLE				*
				1245		*****				
				1246		*				
				0004	1247	FZXITL EQU	4			LENGTH OF A SYMBOL TABLE ENTRY
				0001	1248	FZXICN EQU	1			DISP FOR TABLE ENTRY CON SYMBOL
				0003	1249	FZXICA EQU	3			DISP FOR TABLE ENTRY CON VADDR
				1250		*				
				31E3	1251	FZXICT EQU	*			SYMBOL TABLE STARTING ADDRESS
				1252		*				
31E3	4E			31E3	1253		DC	AL1(B@PLUS)		SIGN IDENTIFIER FOR +&SQR2
31E4	E2			31E4	1254		DC	AL1(B@CIS2)		LETTER IDENTIFIER FOR &SQR2
31E5	F500			31E6	1255		DC	AL(@VADDR)(I@ICBA+0*I@LPFV)		VIRTUAL ADDR OF +&SQR2
				1256		*				
31E7	4E			31E7	1257		DC	AL1(B@PLUS)		SIGN IDENTIFIER FOR +&PI
31E8	D7			31E8	1258		DC	AL1(B@CIPI)		LETTER IDENTIFIER FOR &PI
31E9	F505			31EA	1259		DC	AL(@VADDR)(I@ICBA+1*I@LPFV)		VIRTUAL ADDR OF +&PI
				1260		*				
31EB	4E			31EB	1261		DC	AL1(B@PLUS)		SIGN IDENTIFIER FOR +&E
31EC	C5			31EC	1262		DC	AL1(B@CIEX)		LETTER IDENTIFIER FOR &E
31ED	F50A			31EE	1263		DC	AL(@VADDR)(I@ICBA+2*I@LPFV)		VIRTUAL ADDR OF +&E
				1264		*				
31EF	60			31EF	1265		DC	AL1(B@MINS)		SIGN IDENTIFIER FOR -&SQR2
31F0	E2			31F0	1266		DC	AL1(B@CIS2)		LETTER IDENTIFIER FOR &SQR2
31F1	F50F			31F2	1267		DC	AL(@VADDR)(I@ICBA+3*I@LPFV)		VIRTUAL ADDR OF -&SQR2
				1268		*				
31F3	60			31F3	1269		DC	AL1(B@MINS)		SIGN IDENTIFIER FOR -&PI
31F4	D7			31F4	1270		DC	AL1(B@CIPI)		LETTER IDEV/F/ER FOR &PI
31F5	F514			31F6	1271		DC	AL(@VADDR)(I@ICBA+4*I@LPFV)		VIRTUAL ADDR OF -&PI
				1272		*				
31F7	60			31F7	1273		DC	AL1(B@MINS)		SIGN IDENTIFIER FOR -&E
31F8	C5			31F8	1274		DC	AL1(B@CIEX)		LETTER IDENTIFIER FOR &E
31F9	F519			31FA	1275		DC	AL(@VADDR)(I@ICBA+5*I@LPFV)		VIRTUAL ADDR OF -&E
				1276		*				
				1277		*****				

## DFPRNT - MATRIX PRINTER ROUTINE

ERR	LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 31/05/21 PAGE 199
				1279	*****	
				1280	* VIRTUAL MEMORY INPUT EXECUTION ROUTINE (8TH VM PAGE)	*
				1281	* ELEMENT CONVERSION ROUTINE FOR NUMERIC ARITHMETIC CONSTANTS	*
				1282	*	*
				1283	* INPUT -	*
				1284	* * REGISTER @XR - REFECENCES 1ST CHARACTER OF CONSTANT	*
				1285	*	*
				1286	* OUTPUT -	*
				1287	* * CONVERTED VALUE STACKED BEGINNING AT I\$STAK - STACKED DATA	*
				1288	* * FORMAT IS PACKED FLOATING POINT (CURRENT PRECISION)	*
				1289	* * REGISTER @XR - REFERENCES DELIMITER FOLLOWING CONVERTED CON.	*
				1290	*****	
				1291	*	
				1292	* ESTABLISH ADDRESSABILITY FOR INPUT ROUTINE (8TH VM PAGE)	
				1293	*	
				1294	*FZXP8B VPAGE 0 SET 8TH PAGE ADDRESSABILITY	
3200				1295	ORG *,256,0 SET STARTING ADDRESS	
			3200	1296	FZXP8B EQU * START OF PROGRAM CODING	
3101				1297	ORG *-255 RESET IAR TO PAGE	
3200				1298	ORG *,256,0 * BOUNDARY ADDRESS	
			3200	1299	USING *,@BR SET PAGE BASE ADDRESS	
3200				1300	ORG FZXP8B RESET STARTING ADDRESS	
				1301	*** END OF EXPANSION ***	
				1302	*	
				1303	* ENTRY - INITIALIZE FOR NUMERIC CONSTANR CONVERSION	
				1304	*	
3200	7C	80	EA	1305	FZX920 MVI FZXEXP(,@BR),B@NXZR SET ZERO FLT PT NORMALIZED EXP	
3203	54	60	F1 E7	1306	ZAZ FZXMNR(I@PREC,@BR),FZX8D0(1,@BR) ZERO THE VALUE MANTISSA	
3207	7C	EB	41	1307	MVI FZX950+@D1(,@BR),FZXMN1-FZXP8B INITLZ MANTISSA CHAR D1SP	
320A	D0	87	BE	1308	B FZX990(,@BR) LINK TO ESTABLISH MANTISSA SIGN	
				1309	*	* AND ACCESS 1ST NON-SIGN CHAR
				1310	*	
				1311	*****	

## DFPRNT - MATRIX PRINTER ROUTINE

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00 31/05/21 PAGE 200
				1313			*****	
				1314			* NUMERIC CONSTANT MANTISSA CONVERSION	*
				1315			*****	
				1316			*	
				1317			* INCREMENT PAST INSIGNIFICANT LEADING ZEROS	
				1318			*	
320D	D0	81	D3	1319	FZX923	BE	FZX992(,@BR)	LINK TO GET NEXT DATA CHARACTER
3210	D0	81	0D	1320		BE	FZX923(,@BR)	* UNTIL NON-ZERO CHAR IS FOUND
				1321			*	
				1322			* ASSUME AN INTEGER COMPONENT TEST FOR FRACTIONAL COMPONENT ONLY	
				1323			*	
3213	7C	80	33	1324	FZX926	MVI	FZX943+@Q(,@BR),@NOP	SET FRACTION COMPONENT SW OFF
3216	BD	4B	00	1325		CLI	B@CHAR(,@XR),B@DPNT	IF CHARACTER NOT A DECIMAL POINT
3219	F2	01	10	1326		JNE	FZX940	* GO TEST FOR INTEGER DIGITS
				1327			*	
				1328			* FRACTIONAL COMPONENT ONLY - ADJUST EXPONENT FOR ZEROS AFTER POINT	
				1329			*	
321C	F2	87	04	1330		J	FZX933	SKIP TO SET CEAR AFTER POINT
321F	5F	00	EA D8	1331	FZX930	SLC	FZXEXP(,@BR),FZX8B1(1,@BR)	DECREMENT NORMALIZED E*PONENT
3223	D0	87	D3	1332	FZX933	B	FZX992(,@BR)	LINK TO GET NEXT DATA CN&RACTER
3226	D0	81	1F	1333		BE	FZX930(,@BR)	LOOP IF CHAR IS A DECIMAL ZERO
				1334			*	
				1335			* TEST FOR FRACTIONAL DIGITS WHICH DO NOT AFFECT EXPONENT	
				1336			*	
3229	7C	87	33	1337	FZX936	MVI	FZX943+@Q(,@BR),@UCB	SET FRACTION COMPONENT SW ON
322C	BD	F0	00	1338	FZX940	CLI	B@CHAR(,@XR),B@DEC0	IF NO MORE FRACTIONAL DIGITS
322F	F2	82	27	1339		JL	FZX956	* GC WRAP-UP MANTISSA PROCESSING
				1340			*	
				1341			* PROCESS MANTISSA DIGITS - STORE DIGITS IN SUCCESSIVE MANTISSA	
				1342			* POSITIONS UNTIL MANTISSA BUCKET IS FILLED - ADJUST NORMALATED	
				1343			* EXPONENT FOR SIGNIFICANT INTEGER DIGITS ONLY.	
				1344			*	
3232	F2	00	04	1345	FZX943	JC	FZX946,*-*	BRANCH IF DIGIT IS FRACTIONAL
3235	5E	00	EA D8	1346		ALC	FZXEXP(,@BR),FZX8B1(1,@BR)	INCREMENT NORMALIZED EXPONENT
				1347			*	
3239	7D	F1	41	1348	FZX946	CLI	FZX950+@D1(,@BR),FZXMNR-FZXP8B	IF BUCKET ALREADY FILLED
323C	F2	84	08	1349		JH	FZX953	* BYPASS DIGIT AND GO PROC NEXT
323F	6C	00	00 00	1350	FZX950	MVC	*-*(,@BR),B@CHAR(1,@XR)	* ELSE MOVE DIGIT TO MANTISSA
3243	5E	00	41 D8	1351		ALC	FZX950+@D1(,@BR),FZX8B1(1,@BR)	AND BUMP BUCKET POINTER
				1352			*	
3247	D0	87	D3	1353	FZX953	B	FZX992(,@BR)	LINK TO GET NEXT DATA CHARACTER
324A	D0	02	32	1354		BNL	FZX943(,@BR)	LOOP IF CHAR IS A DECIMAL DIGIT
				1355			*	
				1356			* TEST FOR A DECIMAL POINT FOLLOWING SIGNIFICANT DIGITS	
				1357			*	
324D	BD	4B	00	1358		CLI	B@CHAR(,@XR),B@DPNT	IF CHARACTER NOT A DECIMAL POINT
3250	F2	01	06	1359		JNE	FZX956	* GO WRAP-UP MANTISSA PROCESSING
3253	D0	87	D3	1360		B	FZX992(,@BR)	LINK TO GET NEXT DATA CHARACTER
3256	D0	87	29	1361		B	FZX936(,@BR)	BRANCH TO PRECESS FRACTIONALS
				1362			*	
				1363			* TEST MANTISSA SIGN - MODIFY SIGN ZONE FOR NEGATIVE VALUE	
				1364			*	
3259	7D	80	8E	1365	FZX956	CLI	FZX973+@Q(,@BR),@NOP	IF MANTISSA SIGN IS POSITIVE
325C	F2	81	03	1366		JE	FZX960	* GO BEGIN EXPONENT PROCESSING
325F	7B	20	F1	1367		SBF	FZXSGN(,@BR),B@ZPOS-B@ZNEG	* ELSE SET NEG MANTISSA ZONE
				1368			*	

[illegible]



## DFPRNT - MATRIX PRINTER ROUTINE

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	31/05/21	PAGE 202
					1371		*****			
					1372	*	NUMERIC CONSTANT EXPONENT CONVERSION			*
					1373		*****			
					1374	*				
					1375	*	TEST FOR E-FORMAT SPECIFICATION			
					1376	*				
3262	BD	C5	00		1377	FZX960	CLI B@CHAR(,@XR),B@EXPC			IF NO E-FORMAT SPECIFICATION
3265	F2	01	33		1378		JNE FZX980			* SKIP PAST EXPONENT PROCESSING
					1379	*				
3268	D0	87	D3		1380		B FZX992(,@BR)			LINK TO GET NEXT DATA CHARACTER
326B	D0	87	BE		1381		B FZX990(,@BR)			LINK TO ESTABLISH SPEC EXPONENT
					1382	*				* SIGN AND GET 1ST EXP DIGIT
					1383	*				
					1384	*	ESTABLISH EXPONENT DIGIT(S) FOR CONVERSION TO BINARY			
					1385	*				
326E	7C	00	E9		1386		MVI FZX8DX(,@BR),@ZERO			SET EXP AREA RH BYTE = BINARY 0
3271	5C	00	E8 E9		1387	FZX963	MVC FZX8DX-1(,@BR),FZX8DX(1,@BR)			SHIFT UP ABEA CONTENTS LEFT
3275	68	03	E9 00		1388		MNN FZX8DX(,@BR),B@CHAR(,@XR)			MOVE EXP DIGIT NUMERIC TO AREA
3279	D0	87	D3		1389		B FZX992(,@BR)			* RH BYTE, LINK TO GET NEXT CHAR
327C	D0	02	71		1390		BNL FZX963(,@BR)			LOOP IF CHAR IS EXPONENT DIGIT
327F	F2	87	04		1391		J FZX970			* ELSE GO CONV EXPONENT TO BIN.
					1392	*				
					1393	*	CONVERT EXPONENT TO A BINARY NUMBER			
					1394	*				
3282	5E	00	E9 E6		1395	FZX966	ALC FZX8DX(,@BR),FZXB10(1,@BR)			ADD BINARY 10 TO EXP UNITS
3286	5F	00	E8 D8		1396	FZX970	SLC FZX8DX-1(,@BR),FZX8B1(1,@BR)			DECR EXPONENT TENS DINT
328A	D0	02	82		1397		BNM FZX966(,@BR)			LOOP UNTIL TENS DIGIT < ZERO
					1398	*				
					1399	*	MODIFY NORMALIZED EXPONENT WITH SPECIFIED EXPONENT			
					1400	*				
328D	F2	00	07		1401	FZX973	JC FZX976,*-*			BRANCH IF SPEC EXP IS NEGATIVE
3290	5E	00	EA E9		1402		ALC FZXEXP(,@BR),FZX8BX(1,@BR)			ADD SPEC EXPONENT MAGNITUDE
3294	F2	87	04		1403		J FZX980			* TO NORM EXP, GO TEST MANTISSA
3297	5F	00	EA E9		1404	FZX976	SLC FZXEXP(,@BR),FZX8BX(1,@BR)			SUBTRACT SPEC EYP MAGNITUDE
					1405	*				* FROM NORMALIZED EXPONENT
					1406	*				
					1407	*	TEST FOR SIGNIFICANT UNITS IN MANTISSA			
					1408	*				
329B	7D	F0	EB		1409	FZX980	CLI FZXMN1(,@BR),B@DEC0			IF LEADING MANTISSA DIGIT NOT
329E	F2	01	06		1410		JNE FZX984			* ZERO, GO STACK THE ELEMENT
32A1	7C	1E	EA		1411		MVI FZXEXP(,@BR),B@NXLO			* ELSE SET MINIMUM EXPONENT
32A4	7A	F0	F1		1412		SBN FZXSGN(,@BR),B@ZPOS			* AND FORCE MANTISSA POSITIVE
					1413	*				
					1414	*	MOVE CONVERTED FLOATING POINT VALUE TO RUN-TIME STACK			
					1415	*				
32A7	74	02	B9		1416	FZX984	ST FZX986+@OP1(,@BR),@XR			SAVE ELEMENT DELIMETER CADDR
32AA	35	02	0D4E		1417		L I\$STAK,@XR			LOAD THE RUN-TIME STACK POINTER
32AE	9C	07	07 F1		1418		MVC I@LUFV-1(,@XR),FZXMNR(I@LUFV,@BR)			MOVE VALUE TO STACK
32B2	C0	87	0A85		1419		B I\$CUPF			LINK TO PACK THE STACKED VALUE
					1420	*				
					1421	*	EXIT - RESTORE BUFFER POINTER AND RETURN TO CALLING ROUTINE			
					1422	*				
32B6	C2	02	0000		1423	FZX986	LA *-*,@XR			RELOAD ELEMENT DELIMITER CADDR
					1424	*				
32BA	C0	87	12D3		1425		B I\$RTRN			RETURN TO CALLING ROUTINE
					1426	*				

[illegible]

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	31/05/21	PAGE 204
					1429	*****	*****			
					1430	*	ARITHMETIC CONSTANT SIGN PROCESSING ROUTINE			*
					1431	*	* SETS SIGN SWITCH (FZX973+@Q) DEPENDING ON CHARACTER REFFRENCED			*
					1432	*	BY REGISTER @XR -			*
					1433	*	* SWITCH SET TO CODE @NOP WHEN CHARACTER NOT A SIGN			*
					1434	*	* SWITCH SET TO CODE @NOP WHEN CHARACTER IS PLUS SIGN			*
					1435	*	* SWITCH SET TO CODE @UCB WHEN CHARACTER IS MINUS SIGN			*
					1436	*	* ADVANCES DATA BUFFER POINTER (REG @XR) TO NEXT NON-BLANK			*
					1437	*	CHARACTER WHEN REFERENCED CHARACTER IS A SIGN			*
					1438	*	* SETS PSR DEPENDING ON TYPE OF FINALLY REFERENCED CHARACTER -			*
					1439	*	* LOW - NON-DECIMAL CHARACTER			*
					1440	*	* EQUAL - DECIMAL ZERO CHARACTER			*
					1441	*	* HIGH - DECIMAL DIGIT OTHER THAN ZERO			*
					1442	*****	*****			
					1443	*				
	32BE	74	08	E5	1444	FZX990	ST FZX998+@OP1( ,@BR) ,@ARR SET RETURN BRANCH ADDRESS.			
					1445	*				
	32C1	7C	80	8E	1446		MVI FZX973+@Q( ,@BR) ,@NOP SET SIGN SWITCH FOR POSITIVE			
	32C4	BD	4E	00	1447		CLI B@CHAR( ,@XR) ,B@PLUS IF CHARACTER IS A PLUS SIGN			
	32C7	F2	81	0C	1448		JE FZX994 * SKIP TO GET THE NENT CHARACTER			
	32CA	BD	60	00	1449		CLI B@CHAR( ,@XR) ,B@MINS IF CHARACTER NOT A MINUS SIGN			
	32CD	F2	01	0F	1450		JNE FZX996 * SKIP TO SET PSR AND RETURN			
	32D0	7C	87	8E	1451		MVI FZX973+@Q( ,@BR) ,@UCB SET SIGN SWITCH FOR NEGATIVE			
					1452	*				
					1453	*****	*****			

DFPRNT - MATRIX PRINTER ROUTINE

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 31/05/21 PAGE 205
		1455		*****	
		1456	*	DATA CHARACTER 'GET' ROUTINE -	*
		1457	*	* ADVANCES DATA BUFFER POINTER (REG @XR) TO NEXT NON-BLANK CHAR	*
		1458	*	* SETS PSR DEPENDING ON TYPE OF FINALLY REFERENCED CHARACTER	*
		1459	*	* LOW - NON-DECIMAL CHARACTER	*
		1460	*	* EQUAL - DECIMAL ZERO CHARACTER	*
		1461	*	* HIGH - DECIMAL DIGIT OTHER THAN ZERO	*
		1462		*****	
		1463	*		
32D3	74 08 E5	1464	FZX992 ST	FZX998+@OP1(,@BR),@ARR	SET RETURN BRANCH ADDRESS
		1465	*		
32D6	E2 02 01	1466	FZX994 LA	@B1(,@XR),@XR	INCR DATA CHARACTER POINTER
32D9	BD 40 00	1467	CLI	B@CHAR(,@XR),B@BLNK	TEST FOR A BLANK CHARACTER
32DC	D0 81 D6	1468	BE	FZX994(,@BR)	REPEAT LOOP UNTIL NON-BLANK
		1469	*		
32DF	BD F0 00	1470	FZX996 CLI	B@CHAR(,@XR),B@DEC0	SET PSR FOR NEW CHARACTER
32E2	C0 87 0000	1471	FZX998 B	*-*	RETURN TO CALLING ROUTINE
		1472	*		
		1473		*****	

## DFPRNT - MATRIX PRINTER ROUTINE

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	31/05/21	PAGE 206
					1475	*****				
					1476	* INPUT EXECUTION ROUTINE CONSTANTS (8TH VM PAGE)				*
					1477	*****				
					1478	*				
				32D8	1479	FZX8B1 EQU	FZX994+@D1			BINARY INTEGER +1
32E6	0A			32E6	1480	FZXB10 DC	IL1'10'			BINARY INTEGER +10
32E7	F0			32E7	1481	FZX8D0 DC	DL1'0'			DECIMAL INTEGER 0
					1482	*				
					1483	*****				
					1484	* INPUT EXECUTION ROUTINE WORK AREAS (8TH VM PAGE)				*
					1485	*****				
					1486	*				
32E8				32E9	1487	FZX8DX DS	CL(FZXDXL)			DECIMAL EXPONENT WORK AREA
				32E9	1488	FZX8BX EQU	FZX8DX			BINARY SPECIFID EXPONENT CADDR
					1489	*				
				32EA	1490	FZX8BK EQU	*			CONVERSION BUCKET BASE ADDRESS
32EA				32F1	1491	DS	CL(I@LUFV)			CONVERSION BUCKET AREA
					1492	*				
				32EA	1493	FZXEXP EQU	FZX8BK+I@UEXP			BUCKET EXPONENT BYTE ADDRESS
				32EB	1494	FZXMN1 EQU	FZX8BK+I@UMN1			BUCKET IST MANTISSA BYTE ADDR
				32F1	1495	FZXMNR EQU	FZX8BK+I@UMNR			BUCKET RIGHT MANTISSA BYTE ADDR
				32F1	1496	FZXSGN EQU	FZX8BK+I@SIGN			BUCKET MANTISSA SIGN BYTE ADDR
					1497	*				
					1498	*****				

DFPRNT - MATRIX PRINTER ROUTINE

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 31/05/21 PAGE 207
			1500	*****	
			1501	* INPUT EXECUTION ROUTINE COMMON EQUATES	*
			1502	*****	
			1503	*	
		0DC8	1504	FZXBCA EQU I\$PUB1	INPUT BUFFER CADDR SAVE AREA
			1505	*	
		00FF	1506	FZXBPT EQU 255	DISP FOR DATA BUFFER POINTER
			1507	*	
		00F0	1508	FZXER0 EQU C'0'	CODE SPECIFYING ERROR 800
		00F1	1509	FZXER1 EQU C'1'	CODE SPECIFYING ERROR 801
		00F2	1510	FZXER2 EQU C'2'	CODE SPECIFYING ERROR 802
		00F3	1511	FZXER3 EQU C'3'	CODE SPECIFYING ERROR 803
		00F4	1512	FZXER4 EQU C'4'	CODE SPECIFYING ERROR 804
			1513	*	
			1514	*****	
			1515	*	
			1516	*** END OF VIRTUAL MEMORY INPUT EXECUTION ROUTINE CODING ***	

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 31/05/21 PAGE 208
		1518		*****	
		1519	*	5703-XM1 COPYRIGHT IBM CORP. 1970	*
		1520	*	REFER TO INSTRUCTIONS ON COPY RIGHT NOTICE, 120-2083	*
		1521	*		*
		1522		*****	
		1523	*	*STATUS	*
		1524	*	VERSION 1 MODIFICATION 0	*
		1525	*		*
		1526	*	*FUNCTION	*
		1527	*	* FZREAD EXECUTION CAUSES THE CURRENTLY REFERENCED ELEMENT IN THE	*
		1528	*	PROGRAM 'DATA' FILE TO BE MOVED TO THE RUN-TIME STACK. THE	*
		1529	*	'DATA' FILE POINTER IS ADVANCED TO REFREVE THE NEXT 'DATA'	*
		1530	*	FILE ELEMENT.	*
		1531	*	* THIS ROUTINE OPERATES ON THE FOLLOWING PSEUDO INSTRUCTIONS TO	*
		1532	*	ACCESS AND STACK THE CURRENTY REFERENCED PROGRAM 'DATA' FILE	*
		1533	*	ELEMENT.	*
		1534	*	* 'DCA' - DEFINE CONSTANT ADDRESS (FORMAT - OP VADR)	*
		1535	*	THE DATA ELEMENT AT VIRTUAL ADDRESS VADR IS DEFINED AS AN	*
		1536	*	ELEMENT IN THE 'DATA' FILE. THE POSITION OF THE ELEMENT	*
		1537	*	IN THE FILE IS DIRECTLY RELATED TO THE POSITION OF THE	*
		1538	*	'DCA' INSTRUCTION WITH RESPECT TO OTHER 'DCA' INSTRUCTIONS	*
		1539	*	IN THE PROGRAM.	*
		1540	*	* 'DDL' - DEFINE 'DATA' LINKAGE (FORMAT - OP VADR)	*
		1541	*	'DDL' ALWAYS FOLLOWS A STRING OF 'DCA' INSTRUCTIONS.	*
		1542	*	THE 'DCA' INSTRUCTION BEGINNING AT VADR IS THE NEXT	*
		1543	*	SEQUENTIAL 'DCA' IN THE PROGRAM. WHEN VADR = X'0000',	*
		1544	*	'DDL' MARKS THE END OF THE 'DATA' FILE.	*
		1545	*	* 'EOP' - END OF PMC PAGE (FORMAT - OP)	*
		1546	*	EACH PSEUDO MACHINE CODE VIRTUAL PAGE IS TERMINATED WITH	*
		1547	*	AT LEAST ONE 'EOP' INSTRUCTION. 'EOP' EXECUTION RESULTS	*
		1548	*	IN CONTROL BEING PASSED TO THE FIRST PSUEDO INSTRUCTION	*
		1549	*	WHICH APPEARS IN THE NEXT SEQUENTIAL VIRTUAL PAGE.	*
		1550	*	* 'DATA' FILE POINTER I\$DATA CONTAINS EITHER THE VIRTUAL ADDRESS	*
		1551	*	OF A 'DCA' INSTRUCTION OR THAT OF A 'DDL' OR 'EOP' FOLLOWING A	*
		1552	*	STRING OF 'DCA' INSTRUCTIONS. IN THE LATTER CASE, THE CURRENT	*
		1553	*	'DCA' INSTRUCTION IS THAT INDICATED BY THE 'DDL' OR 'EOP'.	*
		1554	*	THE ELEMENT REFERENCED BY THE OPERAND OF THE CURRENT 'DCA'	*
		1555	*	INSTRUCTION IS STACKED, AND I\$DATA IS INCREMENTED TO REFERENCE	*
		1556	*	THE NEXT 'DCA' INSTRUCTION.	*
		1557	*		*
		1558	*	*ENTRY POINTS	*
		1559	*	THIS ROUTINE HAS A SINGLE ENTRY POINT - FZREAD - WHOSE FUNCTION	*
		1560	*	IS DEFINED ABOVE. CALLING SEQUENCE IS	*
		1561	*	B I\$CALL	*
		1562	*	DC AL2(V\$XS?O)	*
		1563	*	WHERE THE ADDRESS CONSTANT PARAMETER DEFINES THE VIRTUAL ADDRESS	*
		1564	*	OF ENTRY POINT FXREAD. EXECUTION IS SUBJECT TO INPUT CONDITIONS	*
		1565	*	DESCRIBED BELOW.	*
		1566	*		*
		1567	*	*INPUT	*
		1568	*	* I\$STAK - 2 BYTES, FOR THE RUN-TIME STACK POINTER. THIS IS TO	*
		1569	*	CONTAIN THE CORE ADDRESS OF THE FIRST AVAILABLE STACK LOCATION.	*
		1570	*	* I\$DATA - 2 BYTES, FOR THE 'DATA' FILE POINTER. THIS IS TO	*
		1571	*	CONTAIN THE VIRTUAL ADDRESS OF THE CURRENT 'DCA' INSTRUCTION	*
		1572	*	OR THAT OF A 'DDL' OR 'EOP' INDICATING THE 'DCA' INSTRUCTION.	*
		1573	*	* PMC 'DATA' FILE - 'DATA' FILE ELEMENT-REFERENCING PSEUDO	*



ERR	LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00	31/05/21	PAGE 209
			1574	*	INSTRUCTIONS GENERATED FOR EACH 'DATA' FILE STATEMENT AND			*
			1575	*	ACCESSED USING FILE POINTER I\$DATA.			*
			1576	*				*
			1577	*	OUTPUT			*
			1578	*	* I\$STAK - 2 BYTES, FOR THE RUN-TIME STACK POINTER, WHEN NO			*
			1579	*	ERROR OCCURS, THIS CONTAINS THE CORE ADDRESS OF THE LEFTMOST			*
			1580	*	BYTE OF THE 'DATA' FILE ELEMENT STACKED DURING FZREAD EXECUTION.			*
			1581	*	* I\$DATA - 2 BYTES, FOR THE 'DATA' FILE POINTER. WHEN NO ERROR			*
			1582	*	OCCURS, THIS CONTAINS THE VIRTUAL ADDRESS OF THE NEXT SEQUEN-			*
			1583	*	TIAL 'DCA' INSTRUCTION OR THAT OF A 'DDL' OR 'EOP' INDICATING			*
			1584	*	THE NEXT 'DCA' INSTRUCTION.			*
			1585	*	* I\$ERRC - 1 BYTE, FOR THE ERROR CONDITION CODE. THIS CONTAINS			*
			1586	*	A NULL CODE (I@NERR) WHEN NO ERROR CONDITION EXISTS OR AN			*
			1587	*	ERROR CODE SPECIFYING THE PARTICULAR ERROR CONDITION DISCOVERED.			*
			1588	*	* RUN-TIME STACK - WHEN NO ERROR CONDITION OCCURS, THIS CONTAINS			*
			1589	*	THE CURRENT 'DATA' FILE ELEMENT AT THE TOP STACK POSITION.			*
			1590	*				*
			1591	*	EXTERNAL REFERENCES			*
			1592	*	* I\$STCK - ENTRY POINT FOR INTERPRETER ELEMENT STACKING ROUTINE.			*
			1593	*	* ISLDYR - ENTRY POINT FOR PAGING MODLLE CONVERT AND LOAD @XR RTN.			*
			1594	*	* I\$QTRN - ENTRY POINT FOR PAGING MODLLE V.M. TETURN CONTROL RTN.			*
			1595	*	* I\$STAK - 2 BYTES, FOR THE STACK POINTER.			*
			1596	*	* I\$DATA - 2 BYTES, FOR THE PROGRAM 'DATA' FILE POINTER.			*
			1597	*	* I\$VADR - 2 BYTES, FOR PAGING MODULE VIRTUAL ADDRESS PARAMETER.			*
			1598	*	* I\$SLN5 - 1 BYTE, FOR ELEMENT STACKING LENGTH PARAM TO ISTACK.			*
			1599	*	* I\$ERRC - 1 BYTE, FOR THE INTERPRETER EXECUTION ERROR CODE.			*
			1600	*				*
			1601	*	EXITS, NORMAL			*
			1602	*	CONTROL IS ALWAYS PASSED TO THE PAGING ROUTINE AT ENTRY POINT			*
			1603	*	I\$RTRN (IPGRTN) FOR A RETURN TO CALLING PROGRAM.			*
			1604	*				*
			1605	*	EXITS, ERROR			*
			1606	*	CONTROL IS PASSED TO THE PAGING ROUTINE AT ENTRY POINT I\$RTRN			*
			1607	*	(IPFRTN) WITH THE PARAMETER I\$ERRC CONTAINING THE APPROPRIATE			*
			1608	*	ERROR MESSAGE CODE.			*
			1609	*				*
			1610	*	TABLES/WORK AREAS			*
			1611	*	FZREAD PMC EXECUTION BRANCH ADDRESS TABLE - 6 BYTES, FOR 'DATA'			*
			1612	*	FILE DEFINITION PMC OPCODE TRANSLATION TO AN FZREAD ENTRY POINT			*
			1613	*	ADDRESS. THIS TABLE CONSITS OF THREE 2 BYTE ENTRIES CONTAINING			*
			1614	*	THE FOLLOWING INFORMATION -			*
			1615	*	* BYTE 0 - DUMMY SPACER.			*
			1616	*	* BYTE 1 - PAGE DISPLACEMENT WITHIN FZREAD FOR THE INTERNAL			*
			1617	*	ENTRY POINT ASSOCIATED WITH A 'DCA', 'DDL' OR 'EOP' PSEUDO			*
			1618	*	INSTRUCTION.			*
			1619	*				*
			1620	*	ATTRIBUTES			*
			1621	*	* REUSABLE			*
			1622	*	* NATURALLY RELOCATBLE			*
			1623	*				*
			1624	*	CHARACTER CODE DEPENDENCY			*
			1625	*	THE OPERATION OF THIS MODULE DOES NOT DEPEND UPON A PARTICULAR			*
			1626	*	INTERNAL REPRESENTATION OF THE EXTERNAL CHARACTER SET.			*
			1627	*				*
			1628	*	NOTES			*
			1629	*	ERROR PROCEDURES			*

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00	31/05/21	PAGE 210
		1630	*	* ERROR 1 - FILE POINTER CONTAINS AN INVALID 'DATA'			*
		1631	*	FILE VIRTUAL ADDRESS. AN ERROR CODE FOR THE MESSAGE			*
		1632	*	'NO DATA STATEMENT SPECIFIED IS ESTABLISHED IN INTERPRETER			*
		1633	*	PARAMETER I\$ERRC.			*
		1634	*	* ERROR 2 - A 'DDL' INSTRUCTION WITH OPERAND X'0000' IS EN-			*
		1635	*	COUNTERED WHILE ATTEMPTING TO ACCESS THE NEXT 'DCA' INSTRU-			*
		1636	*	TION. AN ERROR CODE FOR THE MESSAGE 'INSUFFICIENT DATA FOR			*
		1637	*	READ' IS ESTABLISHED IN INTERPRETER PARAMETER I\$ERRC.			*
		1638	*	* IN EACH OF THESE CASES, CONTROL IS PASSED IMMEDIATELY TO			*
		1639	*	PAGING MODULE ENTRY POINT I\$RTRN (IPGRTN).			*
		1640	*				*
		1641	*	REGISTER USAGE			*
		1642	*	* REGISTER @BR IS TO CONTAIN THE CORE PAGE BASE ADDRESS			*
		1643	*	ESTABLISHED THROUGH PAGING MODULE CONTROL FOR THE PAGE WHICH			*
		1644	*	INCLUDES FZREAD, AND IS RESTORED THROUGH THE PAGING MODULE.			*
		1645	*	* RESISTER @XR IS NOT SAVED. IT IS USED IN FZREAD FOR GENERAL			*
		1646	*	PURPOSE INDEXING OPERATIONS.			*
		1647	*				*
		1648	*	SAVED/RESTORED AREAS			*
		1649	*	NONE			*
		1650	*				*
		1651	*	MODIFICATION CONSIDERATIONS			*
		1652	*	'DATA' FILE ELEMENT REFERENCING PMC OPERATION IS BASED UPON			*
		1653	*	THE SEQUENCE AND LENGTH OF THE ENTRIES IN THE FZREAD PSEUDO			*
		1654	*	INSTRUCTION BRANCH ADDRESS TABLE. TABLE ENTRIES ARE SELECTED			*
		1655	*	USING THE NUMERIC REPRESENTATION OF OPCODE 'EOP' AS A BASE			*
		1656	*	DISPLACEMENT, AND ANY CHANGES TO THE RELATIONSHIP BETWEEN THE			*
		1657	*	CONSTANTS FOR ALL OPCODES OPERATED ON BY THIS ROUTINE MUST			*
		1658	*	TAKE FULL CONSIDERATIONS OF THIS TABLE USAGE AND ORGANIZATION.			*
		1659	*				*
		1660	*	REQUIRED MODULES			*
		1661	*	* @SYSEQ - COMMON SYSTEM EQUATES.			*
		1662	*	* @ERMEQ - SYSTEM ERROR MESSAGE CODE EQUATES			*
		1663	*	* \$B@EQU - COMPILER PARAMETER AND CONSTANT EQUATES.			*
		1664	*	* \$I\$EQU - INTERPRETER FIXED LOCATION ADDRESS EQUATES.			*
		1665	*	* \$I@SEQ - INTERPRETER PARAMETER EQUATES (FOR STD. PREC, ONLY)			*
		1666	*	* \$I@LEQ - INTERPRETER PARAMETER EQUATES (FOR LONG PREC, ONLY)			*
		1667	*				*
		1668	*	OTHER			*
		1669	*	NONE			*
		1670	*	*****			*

## FZREAD - S/3 BASIC INTERPRETER STATEMENT EXEC RTN

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00 31/05/21 PAGE 211
					1672	*****	*****	
					1673	* START OF READ STATEMENT EXECUTION MODULE	*	
					1674	*****	*****	
					1675	*		
					1676	* ESTABLISH ADDRESSABILITY FOR THE READ STATEMENT ROUTINE		
					1677	*		
3300					1678	ORG *,B@LVPG,0	BEGIN AT PAGE BOUNDARY	
				3300	1679	USING *,@BR	DEFINE READ RTN BASE ADDRESS	
					1680	*		
					1681	* ENTER FZREAD - TEST FOR A DATA STATEMENT SPECIFICATION.		
					1682	*		
				3300	1683	FZREAD EQU *	FZREAD ENTRY POINT	
					1684	*		
3300	3D	56	0D52		1685	CLI I\$DATA-1,@VENTA	IF DATA POINTER IS DEFINED	
3304	F2	02	08		1686	JNL FZR020	* GO CONTINUE 'READ' EXECUTION	
					1687	*		
					1688	* NO DATA STATEMENT - SET 'NO DATA STATEMENT SPECIFIED' ERROR MESSAGE		
					1689	*		
3307	3C	BE	0CBC		1690	FZR010 MVI I\$ERRC,@E720	SET INTERPRETER ERROR CODE	
330B	C0	87	12D3		1691	B I\$RTRN	RETURN TO TERMINATE EXECUTION	
					1692	*		
					1693	* LOAD THE DATA PMC PAGE INTO CORE VIRTUAL MEMORY - THIS PAGE CONTAINS		
					1694	* (IN GENERAL) A SERIES OF 'DCA' INSTRUCTIONS WHICH DEFINE THE VADDR'S		
					1695	* OF THE CONSTANTS WHICH COMPRISE THE PROGRAM DATA FILE.		
					1696	*		
330F	4C	01	19 0D53		1697	FZR020 MVC FZR030(,@BR),I\$DATA(@VADDR)	SET PAGING PARAMETER TO LOAD	
					1698	*	* CURRENT DATA FILE OP CODE	
3314	C0	87	1330		1699	B I\$LDXR	LINK TO LOAD CURR DATA FILE PMC	
3318				3319	1700	FZR030 DS CL(@VADDR)	VADDR OF CURR DATA FILE OP CODE	
					1701	*		
					1702	* ESTABLISH BRANCH ADDRESS FROM OP CODE DISPLACEMENT TABLE		
					1703	*		
331A	74	02	2B		1704	FZR040 ST FZR060+@OP1(,@BR),@XR	SAVE THE DATA FILE OP CODE CADDR	
331D	6C	00	27 00		1705	MVC FZR050+@DD2(,@BR),I@XOPC(B@LCOP,@XR)	MOVE OP CODE TO DISP	
3321	D2	02	06		1706	LA FZRBAT-B@CEOP+1(,@BR),@XR	LOAD BRANCH TABLE BASE ADDR	
3324	6C	00	2E 00		1707	FZR050 MVC FZR070+@D1(,@BR),*-(1,@XR)	MOVE TABLE ENTRY TO BR INST	
3328	C2	02	0000		1708	FZR060 LA *-*,@XR	RESTORE DATA FILE OP CODE CADDR	
					1709	*		
					1710	* BRANCH TO EXECUTION ROUTINE SPECIFIED BY THE DATA FILE OP CODE		
					1711	*		
332C	D0	87	00		1712	FZR070 B *-*(,@BR)	GO EXECUTE CURR DATA FILE PMC	
					1713	*		
					1714	*****	*****	

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	31/05/21	PAGE 212
				1716		*****				
				1717	*	F7RDCA	- DEFINE VIRTUAL ADDRESS OR CURRENT DATA FILE ELEMENT			*
				1718		*****				
				1719	*					
		332F		1720	FZRDCA EQU	*	BEGIN FZRDCA EXECUTION			
				1721	*					
				1722	*	STACK THE DATA ELEMENT SPECIFIED BY THE 'DCA' VIRTUAL ADDRESS OPERAND				
				1723	*					
	332F	2C	01	144A	02	1724	FZR080 MVC I\$VADR,I@XVAD(B@LCVA,@XR)	SET PAGING PARAM FOR DATA VADDR		
	3334	3C	12	0BA2		1725	MVI I\$SLNG,I@LCRV-1	SET STACKING ROUTINE TO STACK		
				1726	*			* MAXIMUM LENGTH DATA ELEMENT		
	3338	35	02	0D4E		1727	L I\$STAK,@XR	LOAD THE STACK POINTER		
	333C	C0	87	0B50		1728	B I\$STCK	LINK TO STACK THE DATA ELEMENT		
				1729	*					
				1730	*	ADVANCE DATA FILE POINTER TO REFERENCE NEXT DATA FILE PMC				
				1731	*					
	3340	1E	00	0D53	6C	1732	FZR090 ALC I\$DATA,FZRLDA(@VADDR-1,@BR)	INCREMENT DATA FILE POINTER		
	3345	C0	87	12D3		1733	B I\$RTRN	RETURN TO THE INTERPRETER		
				1734	*					
				1735		*****				

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 31/05/21 PAGE 213
		1737		*****	
		1738	*	FZRDDL - DEFINE LINKAGE ADDRESS FOR NEXT DATA FILE PSEUDO INSTR.	*
		1739		*****	
		1740	*		
		3349	1741	FZRDDL EQU *	BEGIN FZRDDL EXECUTION
		1742	*		
		1743	*	TEST FOR END OF THE PROGRAM DATA FILE	
		1744	*		
3349	BD 56 01	1745	FZR100	CLI I@XVAD-1(,@XR),@VENTA	IF 'DDL' OPERAND IS VALID VADDR
334C	F2 02 08	1746		JNL FZR120	* GO PERFORM LINKAGE OPERATION
		1747	*		
		1748	*	END OF DATA FILE - SET 'INSUFFICIENT DATA FOR READ' ERROR MESSAGE	
		1749	*		
334F	3C BF 0CBC	1750	FZR110	MVI I\$ERRC,@E721	SET INTERPRETER ERROR CODE
3353	C0 87 12D3	1751		B I\$RTRN	RETURN TO TERMINATE EXECUTION
		1752	*		
		1753	*	DATA FILE CONTINUED - LINK TO NEXT DATA FILE PMC SEQUENCE	
		1754	*		
3357	2C 01 0D53 02	1755	FZR120	MVC I\$DATA,I@XVAD(B@LCVA,@XR)	SET DATA FILE PT - LINKAGE ADDR
335C	D0 87 0F	1756		B FZR020(,@BR)	GO PROCESS NEXT DATA FILE PMC
		1757	*		
		1758		*****	

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	31/05/21	PAGE 214
				1760		*****				
				1761	*	FZREOP - CONTINLE DATA FILE PMC ON NEXT VIRTUAL PAGE				*
				1762		*****				
				1763	*					
				335F 1764	FZREOP EQU	*	BEGIN FZREOP EXECUTION			
				1765	*					
				1766	*	ADVANCE DATA FILE POINTER TO REFERENCE 1ST PSUEDO INSTRUCTION ON				
				1767	*	NEXT SEQUENTIAL VIRTUAL PAGE.				
				1768	*					
335F	1E	00	0D52 6B	1769	FZR130	ALC	I\$DATA-1,FZRBNI(1,@BR)			INCREMENT POINTER PAGE NUMBER
3364	3C	00	0D53	1770		MVI	I\$DATA,@ZERO			SET POINTER PAGE DISP TO ZERO
3368	D0	87	0F	1771		B	FZR020(,@BR)			GO PROCESS NEXT DATA FILE PMC
				1772	*					
				1773		*****				

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	31/05/21	PAGE 215
			1775	*****	*****			
			1776	* READ STATEMENT EXECUTION ROUTINE CONSTANTS	*			
			1777	*****	*****			
			1778	*				
336B 01		336B	1779	FZRBN1 DC	IL1'1' BINARY INTEGER +1			
336C 03		336C	1780	FZRLDA DC	AL1(B@LDCA) LENGTH OF 'DCA' PSEUDO INST			
			1781	*				
			1782	*****	*****			
			1783	* READ STMT RTN PSEUDO OPCODE EXECUTION BRANCH ADDRESS TABLE				
			1784	*****	*****			
			1785	*				
		336D	1786	FZRBAT EQU *	BRACH TABLE STARTING ADDRESS			
			1787	*				
336D 005F		336E	1788	DC	AL(@CADDR)(FZREOP-FZREAD) EOP (X'68') END OF PMC PAGE			
336F 002F		3370	1789	DC	AL(@CADDR)(FZRDCA-FZREAD) DCA (X'6A') DEFINE CON VADDR			
3371 0049		3372	1790	DC	AL(@CADDR)(FZRDDL-FZREAD) DDL (X'6C') DEFINE DATA LINK			
			1791	*				
			1792	*****	*****			
			1793	*				
			1794	* END OF READ STATEMENT EXECUTION ROUTINE CODING				



ERR LOC OBJECT CODE ADDR STMT SOURCE STATEMENT VER 15, MOD 00 31/05/21 PAGE 216

```

1796 *****
1797 * 5703-XM1 COPYRIGHT IBM CORP. 1970 *
1798 * REFER TO INSTRUCTIONS ON COPY RIGHT NOTICE, 120-2083 *
1799 * *
1800 *****
1801 *STATUS *
1802 * VERSION 1 MODIFICATION 0 *
1803 * *
1804 *FUNCTION - *
1805 * * FZSPRT EXECUTION CAUSES DATA OUTPUT AND/OR CARRIER/CURSOR *
1806 * POSITIONING ON THE SYSTEM PRINT DEVICE UNDER CONTROL OF CODES *
1807 * DEVELOPED FROM THE FORMAT SPECIFIED IN A BASIC PROGRAM 'PRINT' *
1808 * STATEMENT. *
1809 * * THE FOLLOWING ACTIONS ARE PERFORMED, DEPENDING ON THE CODE *
1810 * STORED IN INTERPRETER PARAMETER I$PARM - *
1811 * * CODE X'01' - PRINT AND NO SPACE. *
1812 * THE DATA ELEMENT AT THE TOP OF THE RUN?TIME STACK IS CON- *
1813 * VERTED TO OUTPUT FORMAT AND PRINTED. IF THE ELEMENT IS *
1814 * ARITHMETIC, THE CARRIER/CURSOR IS RETURNED TO THE START OF *
1815 * THE NEXT LINE (BEFORE PRINTING) WHEN THE CURRENT LINE CAN- *
1816 * NOT CONTAIN THE FORMATTED VALUE. THE CARRIER/CURSOR IS *
1817 * LEFT POSITIONED AT THE END OF THE PRINTED VALUE. *
1818 * * CODE X'02' - PRINT AND SPACE FULL ZONE. *
1819 * THE DATA ELEMENT AT THE TOP OF THE RUN-TIME STACK IS CON- *
1820 * VERTED TO OUTPUT FORMAT AND PRINTED. IF THE ELEMENT IS *
1821 * ARITHMETIC, THE CARRIER/CURSOR IS RETURNED TO THE START OF *
1822 * THE NEXT LINE (BEFORE PRINTING) WHEN THE CURRENT LINE CAN- *
1823 * NOT CONTAIN THE FORMATTED VALUE. IF THE ELEMENT IS A *
1824 * CHARACTER REFERENCE, THE CARRIER/CURSOR IS RETURNED TO THE *
1825 * START OF THE NEXT LINE (BEFORE PRINTING) WHEN THE CURRENT *
1826 * LINE DOES NOT CONTAIN A FULL PRINT ZONE (18 SPACES). AT *
1827 * THE END OF PRINTING, THE CARRIER/CURSOR IS SPACED TO THE *
1828 * END OF THE FULL PRINT ZONE. *
1829 * * CODE X'03' - PRINT AND SPACE PACKED ZONE. *
1830 * THE DATA ELEMENT AT THE TOP OF THE RUN-TIME STACK IS CON- *
1831 * VERTED TO OUTPUT FORMAT AND PRINTED. IF THE ELEMENT IS *
1832 * ARITHMETIC, THE CARRIER/CURSOR IS RETURNED TO THE START OF *
1833 * THE NEXT LINE (BEFORE PRINTING) WHEN THE CURRENT LINE CAN *
1834 * NOT CONTAIN THE FORMATTED VALUE. AFTER AN ARITHMETIC ELE- *
1835 * MENT IS PRINTED, THE CARRIER/CURSOR IS SPACED TO THE END *
1836 * OF THE PACKED PRINT ZONE DEFINED IN FUNCTIONAL SPECIF1- *
1837 * CATIONS. AFTER A CHARACTER ELEMENT IS PRINTED, THE *
1838 * CARRIER/CURSOR IS LEFT POSITIONED AT THE END OF THE *
1839 * PRINTED ELEMENT. *
1840 * * CODE X'04' - PRINT AND RETURN CARRIER/CURSOR. *
1841 * THE DATA ELEMENT AT THE TOP OF THE RUN-TIME STACK IS CON- *
1842 * VERTED TO OUTPUT FORMAT AND PRINTED. IF THE ELEMENT IS *
1843 * ARITHMETIC, THE CARRIER/CURSOR IS RETURNED TO THE START OF *
1844 * THE NEXT LINE (BEFORE PRINTING) WHEN THE CURRENT LINE CAN- *
1845 * NOT CONTAIN THE FORMATTED VALUE. AFTER THE ELEMENT IS *
1846 * PRINTED, THE CARRIER/CURSOR IS RETURNED TO THE START OF *
1847 * THE NEXT LINE. *
1848 * * CODE X'05' - SPACE FULL ZONE. *
1849 * THE CARRIER/CURSOR IS SPACED 18 CHARACTERS. IF NO MORE *
1850 * THAN 18 CHARACTERS REMAIN IN THE CURRENT LINE, THE *
1851 * CARRIER/CURSOR IS RETURNED TO THE START OF THE NEXT LINE. *

```

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 31/05/21 PAGE 217
		1852	*	* CODE X'06' - SPACE PACKED ZONE.	*
		1853	*	THE CARRIER/CURSOR IS SPACED 3 CHARACTERS, IF NO MORE	*
		1854	*	THAN 3 CHARACTERS REMAIN IN THE CURRENT LINE, THE	*
		1855	*	CARRIER/CURSOR IS RETURNED TO THE START OF THE NEXT LINE.	*
		1856	*	* CODE X'07' - RETURN CARRIER/CURSOR,	*
		1857	*	THE CARRIER/CURSOR IS RETURNED. TO THE START OF THE NEXT	*
		1858	*	LINE.	*
		1859	*	* CODE X'08' - RETURN CARRIER/CURSOR ON CONDITION.	*
		1860	*	WHEN THE CURRENT LINE DOES NOT CONTAIN MORE THAN 18 CHAR-	*
		1861	*	ACTERS, THE CARRIER/CURSOR IS RETURNED TO THE START OF THE	*
		1862	*	NEXT LINE.	*
		1863	*	* WHEN REQUIRED, ELEMENT CONVERSION AND OUTPUT ARE PERFORMED IN	*
		1864	*	THE RUN-TIME STACK, SO TWAT THE STACKED ELEMENT IS NOT RECOVER-	*
		1865	*	ABLE. AFTER PRINTING, ARITHMETIC ELEMENT OUTPUT FORMAT DEPENDS	*
		1866	*	ON THE MAGNITUDE AND FRACTIONAL CHARACTERISTICS OF THE VALUE.	*
		1867	*	CHARACTER REFERENCE FORMATTING INVOLVES TRUNCATION OF TRAILING	*
		1868	*	BLANKS. CHARACTER CONSTANTS (LITERALS) ARE PRINTED AS SPECI-	*
		1869	*	FIED IN THE 'PRINT' STATEMENT.	*
		1870	*	* EITHER THE MATRIX PRINTER OR THE CRT (OR BOTH) MAY BE USED FOR	*
		1871	*	OUTPUT, DEPENDING ON THE CURRENT DEFINITION OF THE SYSTEM PRINT	*
		1872	*	DEVICE. CRT OUTPUT IS BASED ON A FIXED DISPLAY WIDTH OF 64	*
		1873	*	CHARACTERS, WHILE PRINTER LINE WIDTH IS BASED ON THAT ASSIGNED	*
		1874	*	THROUGH THE 'WIDTH' SYSTEM COMMAND.	*
		1875	*		*
		1876	*	*ENTRY POINTS	*
		1877	*	THIS ROUTINE HAS A SINGLE ENTRY POINT - FZSPRT - WHOSE FUNCTION	*
		1878	*	IS DEFINED ABOVE. CALLING SEQUENCE IS -	*
		1879	*	B I\$CALL	*
		1880	*	DC AL2(V\$XSPR)	*
		1881	*	WHERE THE ADDRESS CONSTANT PARAMETER DEFINES THE VIRTUAL ADDRESS	*
		1882	*	OF ENTRY POINT FZSPRT. EXECUTION IS SUBJECT TO INPUT CONDITIONS	*
		1883	*	DESCRIBED BELOW.	*
		1884	*		*
		1885	*	*INPUT	*
		1886	*	* #ISPARM - 2 BYTES, FOR THE PRINT CONTROL PARAMETER. THIS CON-	*
		1887	*	TAINS A CONTROL CODE, AS INDICATED UNDER 'FUNCTION', IN THE	*
		1888	*	RIGHTMOST BYTE.	*
		1889	*	* I\$STAK - 2 BYTES, FOR THE RUN-TIME STACK POINTER. FOR THOSE	*
		1890	*	CONTROL CODES SPECIFYING A DATA ELEMENT (SEE 'FUNCTION') THIS	*
		1891	*	CONTAINS, THE CORE ADDR OF THE FIRST AVAILABLE STACK LOCATION.	*
		1892	*	* RUN-TIME STACK - THIS CONTAINS AN UNPACKED FLOATING POINT VALUE	*
		1893	*	OR CHARACTER ELEMENT IN THE TOP STACK POSITION FOR CONTROL	*
		1894	*	CODES SPECIFYING DATA OUTPUT (SEE 'FUNCTION').	*
		1895	*	* I\$SLLC - 1 BYTE, FOR THE LENGTH CODE DEFINING THE LAST STACKED	*
		1896	*	DATA ELEMENT. WHEN DATA OUTPUT IS SPECIFIED, THIS IS USED TO	*
		1897	*	DETERMINE THE TYPE OF DATA ITEM (ARITHMETIC OR CHARACTER) CON-	*
		1898	*	TAINED IN THE TOP STACK POSITION.	*
		1899	*	* \$PRPOS - 1 BYTE, FOR THE MATRIX PRINTER CARRIER POSITION	*
		1900	*	INDICATORS. THIS CONTAINS THE CARRIER POSITION, RELATIVE TO	*
		1901	*	THE HARDWARE LEFT MARGIN AS 0, OF THE MATRIX PRINTER CARRIER.	*
		1902	*	* \$RMRGN - 1 BYTE, FOR THE MATRIX PRINTER SOFTWARE RIGHT MARGIN	*
		1903	*	INDICATOR.	*
		1904	*	* \$CRPOS - 1 BYTE, FOR THE CRT CURSOR POSITION INDICATOR. THIS	*
		1905	*	CONTAINS THE CURSOR POSITION, RELATIVE TO THE LEFT CRT MARGIN	*
		1906	*	AS 0, OF THE CRT CURSOR.	*
		1907	*	* \$PRDEV - 2 BYTES, FOR THE SYSTEM PRINT DEVICE INDICATOR.	*

```

1908 * * $EXFTR - 1 BYTE, FOR THE SYSTEM CORE EXTENSION FACTOR. *
1909 * * * * *
1910 *OUTPUT *
1911 * * PRINTED OUTPUT AND/OR CARRIER/CURSOR CONTROL - AS SPECIFIED BY *
1912 * * THE CODE IN I$PARM, THE TYPE OF DATA ELEMENT IN THE STACK, AND *
1913 * * THE CURRENTLY DEFINED SYSTEM PRINT DEVICE(S). *
1914 * * I$PARM - 2 BYTES, FOR THE PRINT CONTROL PARAMETER, THIS INPUT *
1915 * * CONTROL CODE IS DESTROYED DURING EXECUTION. *
1916 * * RUN-TIME STACK - WHEN A DATA ELEMENT HAS BEEN PRINTED, THE *
1917 * * STACKED ELEMENT HAS BEEN CONVERTED IN PLACE TO OUTPUT FORMAT. *
1918 * * $PRPOS - 1 BYTE, FOR THE MATRIX PRINTER CARRIER POSITION *
1919 * * INDICATOR. THIS HAS BEEN MODIFIED TO INDICATE THE CURRENT *
1920 * * CARRIER POSITION AFTER PRINTED OUTPUT WHEN THE MATRIX PRINTER *
1921 * * IS A SYSTEM PRINT DEVICE. *
1922 * * $CRPOS - 1 BYTE, FOR THE CRT CURSOR POSITION INDICATOR. THIS *
1923 * * HAS BEEN MODIFIED TO INDICATE CURRENT CURSOR POSITION AFTER *
1924 * * DISPLAYED OUTPUT WHEN THE CRT IS A SYSTEM PRINT DEVICE. *
1925 * * * * *
1926 *EXTERNAL REFERENCES *
1927 * * VSSPRT - VIRTUAL ENTRY ADDRESS FOR DFPRNT, V.M. MATRIX PRT IOCS. *
1928 * * DSPLYN - ENTRY POINT FOR THE SYSTEM CRT IOCS (LABEL DSPLYN IS *
1929 * * REFERENCED INDIRECTLY USING I$CSXA TO BUILD A CODE ADDRESS). *
1930 * * I$CALL - ENTRY POINT FOR PAGING MODULE V.M. PROGRAM CALL RTN. *
1931 * * I$RTRN - ENTRY POINT FOR PAGING MODULE V.M. RETURN CONTROL RTN. *
1932 * * I$CSXA - CORE ADDRESS OF 1ST BYTE IN CORE EXTENSION PAST 8K. *
1933 * * I$PARM - 2 BYTES, FOR THE INTERPRETER COMMUNICATIONS PARAMETER. *
1934 * * I$STAK - 2 BYTES, FOR THE RUN-TIME STACK POINTER. *
1935 * * I$SLLC - 1 BYTE, FOR LENGTH CODE (L-1) OF LAST STACKED ELEMENT. *
1936 * * I$WRK1 - 2 BYTES, FOR INTERPRETER COMMON WORK AREA 1. *
1937 * * I$WRK2 - 2 BYTES, FOR INTERPRETER COMMON WORK AREA 2. *
1938 * * $PRPOS - 1 BYTE, FOR MATRIX PRINTER CARRIER POSITION INDICATOR. *
1939 * * $RMRGN - 1 BYTE, FOR POSITION OF SOFTWARE RIGHT PRINTER MARGIN. *
1940 * * $CRPOS - 1 BYTE, FOR CRT CURSOR POSITION INDICATOR. *
1941 * * $PRDEV - 2 BYTES, FOR THE SYSTEM PRINT DEVICE INDICATOR. *
1942 * * $EXFTR - 1 BYTE, FOR THE SYSTEM CORE EXTENSION FACTOR. *
1943 * * * * *
1944 *EXITS, NORMAL *
1945 * * CONTROL IS ALWAYS PASSED TO THE PAGING ROUTINE AT ENTRY POINT *
1946 * * I$RTRN (IPGRTN) FOR A RETURN TO THE CALLING PROGRAM. *
1947 * * * * *
1948 *EXITS, ERROR *
1949 * * N/A *
1950 * * * * *
1951 *TABLES/WORKAREAS *
1952 * * FZSPRT BRANCH DISPLACEMENT TABLE - USED TO DIRECT OUTPUT OPERA- *
1953 * * TIONS FOR SPECIFIC ELEMENT TYPE - CONTROL CODE COMBINATIONS. *
1954 * * * NUMBER OF TABLE ENTRIES - 16 *
1955 * * * TABLE ENTRY LENGTH - 1 BYTE *
1956 * * * ENTRY FORMAT - SINGLE BYTE DISPLACEMENT WITHIN AN FZSPRT *
1957 * * * VIRTUAL PAGE FOR THE INTERNAL ENTRY POINT ASSOCIATED WITH *
1958 * * * EACH ELEMENT-CONTROL COMBINATION. *
1959 * * * RUN-TIME STACK - THE FIRST 20 AVAILABLE STACK LOCATIONS *
1960 * * * (INCLUDING LOCATIONS CONTAINING AN ELEMENT TO BE CONVERTED) ARE *
1961 * * * USED AS THE 'PRINT' OUTPUT BUFFER. *
1962 * * * * *
1963 *ATTRIBLIES *

```

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 31/05/21 PAGE 219
		1964	*	* REUSABLE	*
		1965	*	* NATURALLY RELOCATABLE	*
		1966	*		*
		1967	*	*CHARACTER CODE DEPENDENCY	*
		1968	*	OPERATION OR THIS MODULE DEPENDS UPON THE FOLLOWING PROPER-	*
		1969	*	TIES QF THE INTERNAL REPRESENTATION OF THE EXTERNAL CHARACTER SET.	*
		1970	*	* MOST CODING HAS BEEN ARRANGED SO THAT REDEFINITION OF CHAR-	*
		1971	*	ACTER CONSTANTS, BY REASSEMBLY, WILL RESULT IN A CORRECT	*
		1972	*	MODULE FOR THE NEW DEFINITION.	*
		1973	*	* NUMERIC CHARACTERS 0 THROUGH 9 ARE PRESUMED TO BE CODED SUCH	*
		1974	*	THAT THE HIGH ORDER FOUR BITS CONTAIN A SIGN ZONE WITH X'F'	*
		1975	*	DEFINING A POSITIVE DIGIT.	*
		1976	*	THE SPECIFIC INSTRUCTIONS (INSTRUCTION SEQUENCES) WHICH REQUIRE	*
		1977	*	MODIFICATION IF THESE PROPERTIES OF THE CHARACTER SET ARE CHANGED	*
		1978	*	MAY OF IDENTIFIED BY -	*
		1979	*	* THE 4 INSTRUCTIONS BEGINNING AT LABEL FZS035.	*
		1980	*	* THE SINGLE INSTRUCTION IDENTIFIED BY LABEL FZS410.	*
		1981	*	* THE SINGLE INSTRUCTION IDENTIFIED BY LABEL FZS435.	*
		1982	*		*
		1983	*	*NOTES	*
		1984	*	ERROR PROCEDURES	*
		1985	*	FZSPRT UTILIZES OUTPUT IOCS ROUTINES DFPRNT (MATRIX PRINTER)	*
		1986	*	AND DSPLYN (CRT), AND IS SUBJECT TO THE ERP'S INHERENT IN	*
		1987	*	THESE PROGRAMS. FZSPRT OTHERWISE CONTAINS NO ERROR CONDITION	*
		1988	*	TESTS.	*
		1989	*		*
		1990	*	REGISTER USAGE	*
		1991	*	* REGISTER @BR IS TO CONTAIN THE CORE PAGE BASE ADDRESS	*
		1992	*	ESTABLISHED THROUGH PAGING MODULE CONTROL FOR THE PAGE WHICH	*
		1993	*	INCLUDES FZSPRT, AND IS RESTORED THROUGH THE PAGING MODULE.	*
		1994	*	* REGISTER @XR IS NOT SAVED, IT IS USED IN FZSPRT FOR GENERAL	*
		1995	*	PURPOSE INDEXING OPERATIONS.	*
		1996	*		*
		1997	*	SAVED/RESTORED AREAS	*
		1998	*	NONE	*
		1999	*		*
		2000	*	MODIFICATION CONSIDERATIONS	*
		2001	*	NONE	*
		2002	*		*
		2003	*	REQUIRED MODULES	*
		2004	*	* @SYSEQ - COMMON SYSTEM EQUATES.	*
		2005	*	* @FXDEQ - SYSTEM NUCLEUS ADDRESSES AND INDICATOR EQUATES.	*
		2006	*	* \$V\$EQU - VIRTUAL MEMORY FIXED ADDRESS EQUATES.	*
		2007	*	* \$B@EQU - COMPILER PARAMETER AND CONSTANT EQUATES.	*
		2008	*	* \$I@EQU - INTERPRETER FIXED LOCATION ADDRESS EQUATES.	*
		2009	*	* \$I@SEQ - INTERPRETER PARAMETER EQUATES (FOR STD. PREC. ONLY).	*
		2010	*	* \$I@LEQ - INTERPRETER PARAMETER EQUATES (FOR LONG PREC. ONLY).	*
		2011	*		*
		2012	*	OTHER	*
		2013	*	NONE	*
		2014	*	*****	*

## FZSPRT - S/3 BASIC INTERPRETER PRINT STATEMENT EXEC RTN

ERR	LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 31/05/21 PAGE 220
			2016	*****		
			2017	* START OF PRINT STATEMENT EXECUTION MODULE		*
			2018	*****		
			2019	*		
			2020	* ESTABLISH ADDRESSABILITY FOR PRINT ROUTINE 1ST VM PAGE		
			2021	*		
3400			2022	*FZSP1B VPAGE 0		
			2023	ORG *,256,0	SET STARTING ADDRESS	
	3400		2024	FZSP1B EQU *	START OF PROGRAM CODING	
3301			2025	ORG *-255	RESET IAR TO PAGE	
3400			2026	ORG *,256,0	* BOUNDARY ADDRESS	
	3400		2027	USING *,@BR	SET PAGE BASE ADDRESS	
3400			2028	ORG FZSP1B	RESET STARTING ADDRESS	
			2029	*** END OF EXPANSION ***		
			2030	*		
			2031	* ENTER FZSPRT - ACCESS THE STACKED DATA ELEMENT		
			2032	*		
	3400		2033	FZSPRT EQU *	FZSPRT ENTRY POINT	
3400 35 02 0D4E			2034	L I\$STAK,@XR	LOAD THE STACK POINTER	
			2035	*		
			2036	* INITIALIZE AND TEST FOR CARRIER CONTROL (ONLY) PARAMETER		
			2037	*		
3404 7C 00 C7			2038	FZS010 MVI FZSCNT(,@BR),@ZERO	CLEAR DATA CHARACTER COUNTER	
			2039	*		
3407 3D 05 0D57			2040	CLI I\$PARM,B@PRSL	IF CARRIER CONTROL ONLY,	
340B D0 02 A4			2041	BNL FZS180(,@BR)	* GO PERFORM THE OPERATION	
			2042	*		
			2043	* TEST FOR CHARACTER ELEMENT PROCESSING		
			2044	*		
340E 3D 12 0BA1			2045	FZS020 CLI I\$SLLC,I@LCRV-1	IF STACK CONTAINS CHAR ELEMENT	
3412 D0 81 73			2046	BE FZS130(,@BR)	* GO ESTABLISH CHARACTER OUTPUT	
			2047	*		
			2048	*****		



ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	31/05/21	PAGE 221
				2050			*****			
				2051			* ARITHMETIC ELEMENT CONVERSION TO OUTPUT FORMAT			*
				2052			*****			
				2053			*			
				2054			* PROCESS THE SIGN OF THE STACKED ARITHMETIC VALUE			
				2055			*			
3415	7C	40	6E	2056	FZS030	MVI	FZS120+@Q(,@BR),B@BLNK			SET SIGN CHARACTER TO BLANK
3418	B8	F0	07	2057	FZS035	TBN	I@SIGN(,@XR),B@ZPOS			IF STACKED VALUE IS POSITIVE
341B	F2	10	06	2058		JT	FZS040			* SKIP PAST MINUS PROCESSING
341E	7C	60	6E	2059		MVI	FZS120+@Q(,@BR),B@MINS			SET SIGN CHARACTER TO MINUS
3421	BA	F0	07	2060		SBN	I@SIGN(,@XR),B@ZPOS			MAKE STACKED VALUE POSITIVE
3424	7C	01	C7	2061	FZS040	MVI	FZSCNT(,@BR),@B1			SET CHARACTER COUNT FOR SIGN
				2062			*			
				2063			* TEST FOR A ZERO VALUE (CATEGORIZED AS AN INTEGER) - A ZERO VALUE IS			
				2064			* LEFT IN THE STACK IN THE FORM 'S0', WHERE 'S' IS THE SIGN POSITION			
				2065			*			
3427	BD	F0	01	2066	FZS050	CLI	I@MANL(,@XR),B@DEC0			IF MOST SIGNIFICANT DIGIT NOT
342A	F2	01	07	2067		JNE	FZS060			* ZERO, GO ESTABLISH FORMAT
342D	5E	00	C7 DF	2068		ALC	FZSCNT(,@BR),FZSBN1(1,@BR)			INCR CHAR COUNT FOR ZERO DIGIT
3431	F2	87	39	2069		J	FZS120			* AND GO SET FOR ARITH OUTPUT
				2070			*			
				2071			* VALUE NOT ZERO - TEST MAGNITUDE FOR OUTPUT IN E- OR F-FORMAT			
				2072			*			
3434	BD	81	00	2073	FZS060	CLI	I@DEXP(,@XR),B@NXZR+1			IF VALUE LESS THAN 1E+0, OR
3437	F2	82	28	2074		JL	FZS110			* GREATER THAN OR EQUAL TO
343A	BD	86	00	2075		CLI	I@DEXP(,@XR),B@NXZR+I@APRC			* 1E+6 (1E+11 FOR LONG PREC),
343D	F2	84	22	2076		JH	FZS110			* GO CONVERT TO E OR F FORMAT
				2077			*			
				2078			* POSSIBLE I-FORMAT - TEST FOR A FRACTIONAL COMPONENT			
				2079			*			
3440	6C	00	56 00	2080	FZS070	MVC	FZS090+@Q(,@BR),I@DEXP(1,@XR)			ESTABLISH THE NUMBER OF
3444	5F	00	56 E0	2081		SLC	FZS090+@Q(,@BR),FZSNXZ(1,@BR)			* INTEGER DIGIT POSITIONS
3448	7C	07	4D	2082		MVI	FZS080+@D1(,@BR),I@PREC			SET DISP FOR MANTISSA RH BYTE
				2083			*			
				2084			*			
344B	BD	F0	00	2085	FZS080	CLI	*-(,@XR),B@DEC0			IF FRACTIONAL DIGIT, GO CONVERT
344E	F2	01	11	2086		JNE	FZS110			* THE VALUE FOR E- OR F-FORMAT
3451	5F	00	4D DF	2087		SLC	FZS080+@D1(,@BR),FZSBN1(1,@BR)			DECR THE MANTISSA POINTER
3455	7D	00	4D	2088	FZS090	CLI	FZS080+@D1(,@BR),*-*			IF MORE FRACTIONAL POSITIONS
3458	D0	84	4B	2089		BH	FZS080(,@BR)			* REMAIN, GO REPEAT LOOP
				2090			*			
				2091			* NO FRACTIONAL COMPONENT - VALUE IS LEFT IN THE STACK IN THE FORM			
				2092			* 'S123' (I-FORMAT) WHERE 'S' IS THE SIGN POSITION			
				2093			*			
345B	5E	00	C7 4D	2094	FZS100	ALC	FZSCNT(,@BR),FZS080+@D1(1,@BR)			INCR CHAR COUNT FOR DIGITS
345F	F2	87	0B	2095		J	FZS120			* AND GO SET FOR ARITH OUTPUT
				2096			*			
				2097			* VALUE CANNOT BE HANDLED USING I-FORMAT - ROUND AND CONVERT VALUE,			
				2098			* LEAVING IN STACK IN THE FORM 'S123.45' (F-FORMAT) OR 'S1.239E+9'			
				2099			* (E-FORMAT) WHERE 'S' IS THE SIGN POSITION.			
				2100			*			
3462	C0	87	12B1	2101	FZS110	B	I\$CALL			LINK TO ROUND AND CONVERT THE
3466	3500			2102		DC	AL(@VADDR)(FZS300)			* VALUE TO E- OR F-FORMAT
				2103			*			
3468	4E	00	C7 0D56	2104		ALC	FZSCNT(,@BR),I\$PARM-1(1)			INCR CHAR COUNT FROM CONVERSION
				2105			*			





ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 31/05/21 PAGE 223
		2114		*****	
		2115		* CHARACTER ELEMENT CONVERSION TO OUTPUT FORMAT	*
		2116		*****	
		2117		*	
		2118		* DETERMINE THE TYPE OF CHARACTER ELEMENT IN THE STACK	
		2119		*	
3473	B8 20 00	2120	FZS130 TBN	I@STAT(,@XR),B@CTYP	IF ELEMENT IS A STRING SEGMENT
3476	F2 10 1C	2121	JT	FZS160	* GO ESTABLISH SEGMENT PARAMS
		2122		*	
		2123		* ELEMENT IS FROM A CHARACTER REFERENCE - LEAVE ELEMENT IN STACK IN	
		2124		* THE FORM 'REFERENCE' (NO TRAILING BLANKS)	
		2125		*	
3479	1E 00 0D57 E1	2126	FZS140 ALC	I\$PARM,FZSCAJ(1,@BR)	ADJUST OUTPUT CONTROL PARAMETER
		2127		*	* FOR CHARACTER REFERENCE
347E	7C 13 8A	2128	MVI	FZS155+@D1(,@BR),I@LCRF+1	SET DISP FOR BYTE AFTER ELEMENT
3481	5F 00 8A DF	2129	FZS150 SLC	FZS155+@D1(,@BR),FZSBN1(1,@BR)	DECR THE ELEMENT POINTER
3485	F2 81 29	2130	JE	FZS190	BRANCH IF ALL CHARS ARE BLANKS
3488	BD 40 00	2131	FZS155 CLI	*-(,@XR),B@BLNK	TEST ELEMENT CHAR FOR BLANK
348B	D0 81 81	2132	BE	FZS150(,@BR)	* AND REPEAT LOOP UNTIL RIGHT-
		2133		*	* MOST NON-BLANK CHAR IS FOUND
348E	5C 00 C7 8A	2134	FZS941 MVC	FZSCNT(,@BR),FZS155+@D1(1,@BR)	SET CHAR COUNT FOR NUMBER
		2135		*	* OF SIGNIFICANT ELEMENT CHARS
3492	F2 87 0C	2136	J	FZS170	GO SET FOR CHARACTER OUTPUT
		2137		*	
		2138		* ELEMENT IS A CHARACTER STRING SEGMENT - LEAVE ELEMENT IN STACK IN	
		2139		* THE FORM 'SEGMENT' (TRAILING BLANKS ALLOWED)	
		2140		*	
3495	1E 00 0D57 E2	2141	FZS160 ALC	I\$PARM,FZSSAJ(1,@BR)	ADJUST OUTPUT CONTROL PARAMETER
		2142		*	* FOR CHARACTER STRING SEGMENT
349A	BB E0 00	2143	SBF	I@STAT(,@XR),X'FF'-B@CCNT	SET CHAR COUNT EQUAL TO COUNT
349D	6C 00 C7 00	2144	MVC	FZSCNT(,@BR),I@STAT(1,@XR)	* FIELD IN ELEMENT STATUS BYTE.
		2145		*	
		2146		* ADJUST OUTPUT AREA POINTER FOR THE CHARACTER ELEMENT	
		2147		*	
34A1	E2 02 01	2148	FZS170 LA	@B1(,@XR),@XR	INCR POINTER PAST STATUS BYTE
		2149		*	
		2150		*****	

## FZSPRT - S/3 BASIC INTERPRETER PRINT STATEMENT EXEC RTN

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00 31/05/21 PAGE 224
				2152		*****		
				2153		* OUTPUT OPERATION INTERFACE ROUTINE		*
				2154		*****		
				2155		*		
				2156		* PAD THE CONVERTED DATA FIELD WITH BLANKS TO A FULL PRINT ZONE		
				2157		*		
34A4	7C	11	B2	2158	FZS180	MVI	FZS190+@Q(,@BR),I@LFPZ-1	SET LENGTH OF FIELD TO BE
34A7	5F	00	B2 C7	2159		SLC	FZS190+@Q(,@BR),FZSCNT(1,@BR)	* PADDED - BYPASS PADDING
34AB	F2	82	07	2160		JL	FZS200	* OPERATION IF LENGTH - 0
34AE	BC	40	12	2161		MVI	I@LFPZ(,@XR),B@BLNK	PROPAGATE BLANKS TO FILL
34B1	AC	00	11 12	2162	FZS190	MVC	I@LFPZ-1(,@XR),I@LFPZ(@VQ,@XR)	* THE FIELD TO FULL ZONE
				2163		*		
				2164		* CONVERT THE OUTPUT PARAMETER TO AN ENTRY POINT DISPLACEMENT		
				2165		*		
34B5	34	02	0D59	2166	FZS200	ST	I\$WRK1,@XR	SAVE THE PRINT FIELD POINTER
34B9	D2	02	E4	2167		LA	FZSCAT-1(,@BR),@XR	LOAD CONTROL ADDRESS TABLE BASF
34BC	4C	00	C5 0D57	2168		MVC	FZS210+@OPD2(,@BR),I\$PARM(1)	SET THE TABLE DISPLACEMENT
34C1	2C	00	0D57 00	2169	FZS210	MVC	I\$PARM,*-(1,@XR)	MOVE ENTRY PT DISP TO PARAMETER
				2170		*		
				2171		* ESTABLISH THE DATA FIELD CHARACTER COUNT PARAMETER		
				2172		*		
34C6	3C	00	0D56	2173	FZS230	MVI	I\$PARM-1,*-	MOVE DATA FIELD COUNT TO PARAM
				2174		*		
				2175		* ESTABLISH POSSIBLE CORE ENTRY ADDRESS FOR THE CRT IOCR		
				2176		*		
34CA	1C	01	0D5B E4	2177		MVC	I\$WRK2,FZSPDA(@CADDR,@BR)	SET BASE CRT ENTRY CORE ADDRESS
34CF	0E	00	0D5A 043B	2178		ALC	I\$WRK2-1,\$EXFTR(1)	ADJUST CADDR FOR CORE EYTENSION
				2179		*		
				2180		* OUTPUT THE DATA FIELD AS SPECIFIED BY CONTROL PARAMETER		
				2181		*		
34D5	C0	87	12B1	2182	FZS240	B	I\$CALL	LINK TO OUTPUT THE DATA FIELD
34D9	3600			2183		DC	AL(@VADDR)(FZS600)	OUTPUT RIN VIRTUAL ADDRESS
				2184		*		
				2185		* RETURN CONTROL TO THE INTERPRETER CALLING ROUTINE		
				2186		*		
34DB	C0	87	12D3	2187	FZS260	B	I\$RTRN	RETURN TO INTERPRETER
				2188		*		
				2189		*****		

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00 31/05/21 PAGE 225
			2191	*****		
			2192	* PRINT EXECUTION ROUTINE CONSTANTS (1ST VM PAGE)		
			2193	*****		
			2194	*		
34DF 01		34DF	2195	FZSBN1 DC	IL1 '1'                      BINARY INTEGER+1	
			2196	*		
34E0 80		34E0	2197	FZSNXZ DC	AL1(B@NXZR)                      ZERO NORMALIZED EXPONENT	
34E1 08		34E1	2198	FZSCAJ DC	AL1(B@PRRL)                      CTL PARAM ADJUST - CHAR REF	
34E2 0C		34E2	2199	FZSSAJ DC	AL1(B@PRPR+B@PRRL)              CTL PARAM ADJUST - CHAR STRING	
			2200	*		
34E3 2004		34E4	2201	FZSPDA DC	AL(@CADDR)(I\$CSXA+@INST4) CRT IOCR CORE ENTTY ADDR BASE	
			2202	*		
			2203	*****		

ERR LOC OBJECT CODE ADDR STMT SOURCE STATEMENT VER 15, MOD 00 31/05/21 PAGE 226

```

2205 *****
2206 * OUTPUT CONTROL PARAMETER FUNCTION ADDRESS TABLE *
2207 *****
2208 *
2209 * DISPLACEMENT ENTRIES IN THE FOLLOWING TABLE REFERENCE THE MATRIX
2210 * PRINTER OUTPUT ROUTINE (3RD VM PAGE), BUT ARE USED ALSO IN CON-
2211 * JUNCTION WITH THE CRT OUTPUT ROUTINE (4TH VM PAGE).  THUS, 4TH PAGE
2212 * DISPLACEMENTS MUST BE KEPT IDENTICAL WITH 3RD PAGE DISPLACEMENTS
2213 * WHICH ARE REFERENCED IN THE TABLE (E.G, FOR CODE 9, FZS860-FZS810
2214 * MUST BE KEPT IDENTICAL TO FZS660-FZS610).
2215 *
34E5 2216 FZSCAT EQU * CONTROL ADDR TABLE ADDRESS
2217 *
34E5 00 34E5 2218 DC AL1(FZS610-FZS610) CODE 1 - PRT ARITH, NO SPACE
34E6 18 34E6 2219 DC AL1(FZS620-FZS610) CODE 2 - PRT ARITH, SPACE FULL
34E7 1E 34E7 2220 DC AL1(FZS630-FZS610) CODE 3 - PRT ARITH, SPACE PACK
34E8 4D 34E8 2221 DC AL1(FZS650-FZS610) CODE 4 - PRT ARITH, RTRN CARR
2222 *
34E9 59 34E9 2223 DC AL1(FZS660-FZS610) CODE 5 - SPACE FULL
34EA 5F 34EA 2224 DC AL1(FZS670-FZS610) CODE 6 - SPACE PACKED
34EB 73 34EB 2225 DC AL1(FZS680-FZS610) CODE 7 - RETURN CARRIER
34EC 79 34EC 2226 DC AL1(FZS690-FZS610) CODE 8 - RETURN CARR ON COND
2227 *
34ED 00 34ED 2228 DC AL1(FZS610-FZS610) CODE 9 - PRI CHAR, NO SPACE
34EE 82 34EE 2229 DC AL1(FZS695-FZS610) CODE 10 - PRT CHAR, SPACE FULL
34EF 00 34EF 2230 DC AL1(FZS610-FZS610) CODE 11 - PRT CHAR, SPACE PACK
34F0 4D 34F0 2231 DC AL1(FZS650-FZS610) CODE 12 - PRT CHAR, RTRN CARR
2232 *
34F1 00 34F1 2233 DC AL1(FZS610-FZS610) CODE 13 - PRT STRING, NO SPACE
34F2 88 34F2 2234 DC AL1(FZS700-FZS610) CODE 14 - PRT STRING, SPACE LNG
34F3 00 34F3 2235 DC AL1(FZS610-FZS610) CODE 15 - PRT STRING, SPACE PKD
34F4 4D 34F4 2236 DC AL1(FZS650-FZS610) CODE 16 - PRT STRING, RTRN CARR
2237 *
2238 *****
2239 * PRINT EXECUTION ROUTINE EQUATES (1ST VM PAGE) *
2240 *****
2241 *
0000 2242 FZSPAL EQU 0 DISP FOR OUTPUT AREA LEFT BYTE
2243 *
34C7 2244 FZSCNT EQU FZS230+@Q DATA CHARACTER COUNTER
2245 *
2246 *****

```

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	31/05/21	PAGE 227
					2248	*****	*****			
					2249	*	VIRTUAL MEMORY PRINT E-EXECUTION ROUTINE 2ND VM PAGE -			*
					2250	*	* ROUNDS THE ARITHMETIC VALUE IN THE RUN-TIME STACK			*
					2251	*	* CONVERTS ARITHMETIC VALUE TO E- OR F-FORMAT FOR OUTPUT			*
					2252	*				*
					2253	*	INPUT -			*
					2254	*	* RUN-TIME STACK - CONTAINS ARITHMETIC VALUE TO BE CONVERTED			*
					2255	*	* REGISTER @XR - CONTAINS CORE ADDRESS OF VALUE EXPONENT BYTE			*
					2256	*				*
					2257	*	OUTPUT -			*
					2258	*	* RUN-TIME STACK - CONTAINS CONVERTED ARITHMETIC VALUE			*
					2259	*	* REGISTER @XR - CONTAINS CORE ADDRESS OF VALUE SIGN POSITION			*
					2260	*	* I\$PARM-1 - 1 BYTE, CONTAINS VALUE CHAR COUNT (NOT INCL SIGN)			*
					2261	*****	*****			
					2262	*				
					2263	*	ESTABLISH ADDRESSABILITY FOR PRINT ROUTINE 2ND VM PAGE			
					2264	*				
					2265	*	FZSP2B VPAGE 0			
3500					2266	ORG	*,256,0	SET STARTING ADDRESS		
				3500	2267	FZSP2B EQU	*	START OF PROGRAM CODING		
3401					2268	ORG	*-255	RESET IAR TO PAGE		
3500					2269	ORG	*,256,0	* BOUNDARY ADDRESS		
				3500	2270	USING	*,@BR	SET PAGE BASE ADDRESS		
3500					2271	ORG	FZSP2B	RESET STARTING ADDRESS		
					2272	***	END OF EXPANSION ***			
					2273	*				
					2274	*	CONVERSION ENTRY - ROUND THE ARITHMETIC VALUE FOR E- OR F-FORMAT			
					2275	*				
				3500	2276	FZS300 EQU	*	CONVERSION ROUTINE ENTRY POINT		
3500 96 60 07 CC					2277	AZ	I@APRC+1(I@APRC+1,@XR),FZSDC5(1,@BR)	ROUND THE VALUE UP		
3504 F2 08 07					2278	JNOZ	FZS310	IF NO OVFLOW SKIP TO CONTINUE,		
3507 BC F1 01					2279	MVI	I@MANL(,@XR),B@DEC1	* ELSE SET MOST SIGNIFICANT		
350A 9E 00 00 CA					2280	ALC	I@DEXP(,@XR),FZS2B1(1,@BR)	* DIGIT = 1 AND INCR EXPONENT		
					2281	*				
					2282	*	TEST MAGNITUDE OF VALUE FOR OUTPUT IN E- OR F-FORMAT			
					2283	*				
350E BD 80 00					2284	FZS310 CLI	I@DEXP(,@XR),B@NXZR	IF VALUE LESS THAN 1E-1, OR		
3511 D0 82 4D					2285	BL	FZS400(,@BR)	* GREATER THAN OR EQUAL TO		
3514 BD 86 00					2286	CLI	I@DEXP(,@XR),B@NXZR+I@APRC	* 1E+6 (1E+11 FOR LONG PREC),		
3517 D0 84 4D					2287	BH	FZS400(,@BR)	* GO CONVERT VALUE TO E-FORMAT		
					2289	*****	*****			
					2290	*	F-FORMAT OUTPUT CONVERSION ROUTINE			*
					2291	*****	*****			
					2292	*				
					2293	*	SHIFT FRACTIONAL-COMPONENT RIGHT TO INSERT DECIMAL POINT			
					2294	*				
351A 7C 85 25					2295	FZS320 MVI	FZS330+@Q(,@BR),B@NXZR+I@APRC-1	ESTABLISH LENGTH CODE FOR		
351D 6F 00 25 00					2296	SLC	FZS330+@Q(,@BR),I@DEXP(1,@XR)	* FRACTIONAL COMPONENT		
3521 F2 82 04					2297	JL	FZS340	BRANCH IF NO FRACTION		
3524 AC 00 07 06					2298	FZS330 MVC	I@APRC+1(,@XR),I@APRC(@VQ,@XR)	SHIFT FRACTION RIGHT BY 1		
					2299	*				
					2300	*	ESTABLISH F-FORMAT DECIMAL POINT - VALUE IS LEFT IN STACK IN FORM			
					2301	*	'S.123456', S123.456', OR 'S123456.' WHERE 'S' IS THE SIGN POSITION			
					2302	*				
3528 6C 00 36 00					2303	FZS340 MVC	FZS350+@D1(,@BR),I@DEXP(1,@XR)	CALCULATE DISPLACEMENT		

ERR LOC		OBJECT CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00 31/05/21		PAGE 228
352C	5E 00 36	CA	2304		ALC	FZS350+@D1(,@BR),FZS2B1(1,@BR) * FOR THE DECIMAL POINT			
3530	5F 00 36	CD	2305		SLC	FZS350+@D1(,@BR),FZS2XZ(1,@BR) * IN F-FORMAT FIELD			
3534	BC 4B 00		2306	FZS350	MVI	*-*(,@XR),B@DPNT INSERT THE DECIMAL POINT			
			2307	*					
			2308	*	TRUNCATE	INSIGNIFICANT ZEROS FROM THE ROUNDED VALUE			
			2309	*					
3537	7C 08 40		2310	FZS360	MVI	FZS380+@D1(,@BR),I@APRC+2 SET DISP FOR BYTE AFTER VALUE			
353A	5F 00 40	CA	2311	FZS370	SLC	FZS380+@D1(,@BR),FZS2B1(1,@BR) DECR VALUE CHAR POINTER			
353E	BD F0 00		2312	FZS380	CLI	*-*(,@XR),B@DEC0 TEST VALUE CHARACTER FOR ZERO			
3541	D0 81 3A		2313		BE	FZS370(,@BR) * AND REPEAT UNTIL NON-ZERO			
			2314	*					
			2315	*	SET COUNT	PARAMETER AND RETURN TO CALLING PAGE			
			2316	*					
3544	1C 00 0D56	40	2317	FZS390	MVC	I\$PARM-1,FZS380+@D1(1,@BR) MOVE DATA CHARACTER COUNT			
			2318	*		* TO THE OUTPUT PARAMETER			
3549	C0 87 12D3		2319		B	I\$RTRN RETURN TO CALLING PAGE			
			2320	*					
			2321	*****					

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	31/05/21	PAGE 229
				2323		*****				
				2324	*	E-FORMAT OUTPUT CONVERSION ROUTINE				*
				2325		*****				
				2326	*					
				2327	*	SHIFT MANTISSA (EXCEPT MOST SIGNIFICANT DIGIT) RIGHT TO INSERT				
				2328	*	DECIMAL POINT - ESTABLISH E-FORMAT DECIMAL POINT, LEAVING VALUE				
				2329	*	IN STACK IN FORM 'S1.23496' WHERE 'S' IS THE SIGN POSITION				
				2330	*					
354D	AC	04	07 06	2331	FZS400	MVC	I@APRC+1(,@XR),I@APRC(I@APRC-1,@XR) SHIFT MANTISSA RIGHT			
3551	BC	4B	02	2332		MVI	FZSPAL+2(,@XR),B@DPNT INSERT E-FORMAT DECIMAL POINT			
3554	9F	00	00 CA	2333		SLC	I@DEXP(,@XR),FZS2B1(1,@BR) ADJUST EXPONENT TO COMPENSATE			
				2334	*					
				2335	*	TRUNCATE INSIGNIFICANT ZEROS FROM ROUNDED VALUE - KEEP AT LEAST ONE				
				2336	*	DIGIT TO RIGHT OF DECIMAL POINT				
				2337	*					
3558	BB	F0	03	2338	FZS410	SBF	FZSPAL+3(,@XR),B@ZPOS FLAG DIGIT AFTER DECIMAL POINT			
355B	7C	08	64	2339		MVI	FZS430+@D1(,@BR),I@APRC+2 SET DISP FOR BYTE AFTER VALUE			
355E	5F	00	64 CA	2340	FZS420	SLC	FZS430+@D1(,@BR),FZS2B1(1,@BR) DECR VALUE CHAR POINTER			
3562	BD	F0	00	2341	FZS430	CLI	*-(,@XR),B@DEC0 TEST VALUE CHARACTER FOR ZERO			
3565	D0	81	5E	2342		BE	FZS420(,@BR) * AND REPEAT UNTIL NON-ZERO			
3568	BA	F0	03	2343	FZS435	SBN	FZSPAL+3(,@XR),B@ZPOS RESTORE DIGIT AFTER DEC POINT			
				2344	*					
				2345	*	SET COUNT PARAMETER FOR FORMATTED MANTISSA PLUS 4 BYTE EXPONENT				
				2346	*					
356B	3C	04	0D56	2347	FZS440	MVI	I\$PARM-1,FZSLXB SET DATA CHAR CNT FOR EXPONENT			
356F	1E	00	0D56 64	2348		ALC	I\$PARM-1,FZS430+@D1(1,@BR) INCR DATA CHAR COUNT FOR VALUE			
				2349	*					
				2350	*	INITIALIZE OUTPUT FORM OF EXPONENT - TEST FOR EXPONENT SIGN				
				2351	*					
3574	5C	03	D6 D1	2352	FZS450	MVC	FZSXWK(,@BR),FZSEXB(FZSLXB,@BR) MOVE EXPONENT IMAGE TO			
				2353	*		* EXPONENT WORK AREA			
3578	6C	00	D2 00	2354		MVC	FZS2BX(,@BR),I@DEXP(1,@XR) DETERMINE BINARY MAGNITUDE			
357C	5F	00	D2 CD	2355		SLC	FZS2BX(,@BR),FZS2XZ(1,@BR) * ASSUMING POSITIVE EXPONENT			
3580	F2	81	29	2356		JE	FZS480 BRANCH IF EXPONENT IS ZERO			
3583	F2	84	0A	2357		JH	FZS470 BRANCH IF EXPONENT IF POSITIVE			
				2358	*					
				2359	*	NEGATIVE EXPONENT - MODIFY SIGN AND RECOMPUTE BINARY EXPONENT				
				2360	*					
3586	7C	60	D4	2361	FZS460	MVI	FZSXWK-FZSLXM(,@BR),B@MINS MAKE EXPONENT SIGN NEGATIVE			
3589	7C	80	D2	2362		MVI	FZS2BX(,@BR),B@NXZR DETERMINE BINARY MAGNITUDE			
358C	6F	00	D2 00	2363		SLC	FZS2BX(,@BR),I@DEXP(1,@XR) * FOR NEGATIVE EXPONENT			
				2364	*					
				2365	*	CONVERT BINARY EXPONENT MAGNITUDE TO ZONED DECIMAL				
				2366	*					
3590	54	10	D8 CB	2367	FZS470	ZAZ	FZSDAC(FZSLXM,@BR),FZSDC1(1,@BR) SET DEC ACCUMULATOR = 1			
3594	7C	01	98	2368		MVI	FZS472+@Q(,@BR),@B1 SET BINARY MASK FOR 2**0 BIT			
3597	78	00	D2	2369	FZS472	TBN	FZS2BX(,@BR),*- TEST BINARY EXP MAGNITUDE BIT			
359A	F2	90	04	2370		JF	FZS474 * AND BRANCH IF BIT IS ZERO			
359D	56	01	D6 D8	2371		AZ	FZSXWK(FZSLXM,@BR),FZSDAC(FZSLXM,@BR) INCR DECIMAL EXP			
35A1	5E	00	98 98	2372	FZS474	ALC	FZS472+@Q(,@BR),FZS472+@Q(1,@BR) SHIFT BINARY MASK LEFT			
35A5	56	01	D8 D8	2373		AZ	FZSDAC(FZSLXM,@BR),FZSDAC(FZSLXM,@BR) DOUBLE DEC ACCUM			
35A9	D0	08	97	2374		BNOZ	FZS472(,@BR) REPEAT LOOP UNTIL ACCUM > 644			
				2375	*					
				2376	*	TEST FOR AND DELETE ANY INSIGNIFICANT ZERO IN THE DECIMAL EXPONENT				
				2377	*					
35AC	7D	F0	D5	2378	FZS480	CLI	FZSXWK-1(,@BR),B@DEC0 TEST FOR EXPONENT LEFTMOST ZERO			



FZSPRT - S/3 BASIC INTERPRETER PRINT STATEMENT EXEC RTN						
ERR LOC	OBJECT CODE	ADDR STMT	SOURCE STATEMENT	VER 15, MOD 00 31/05/21 PAGE 230		
35AF	F2 01 09	2379	JNE FZS490	BRANCH IF NO INSIGNIFICANT ZERO		
35B2	5C 00 D5 D6	2380	MVC FZSXWK-1(,@BR),FZSXWK(1,@BR)	SHIFT SIGNIFICANT DIGIT		
35B6	1F 00 0D56 CA	2381	SLC I\$PARM-1,FZS2B1(1,@BR)	DECREMENT DATA CHARACTER COUNT		
		2382	*			
		2383	* MOVE OUTPUT FORM OF EXPONENT TO THE DATA PRINT FIELD			
		2384	*			
35BB	7C 04 C4	2385	FZS490 MVI FZS500+@D1(,@BR),FZSLXB	SET DIP TO ESTABLISH		
35BE	5E 00 C4 64	2386	ALC FZS500+@D1(,@BR),FZS430+@D1(1,@BR)	* EXPONENT POSITION		
35C2	9C 03 00 D6	2387	FZS500 MVC *-*(,@XR),FZSXWK(FZSLXB,@BR)	MOVE EXPONENT TO PRINT FIELD		
		2388	*			
		2389	* RETURN CONTROL TO THE CALLING PAGE			
		2390	*			
35C6	C0 87 12D3	2391	FZS510 B I\$RTRN	RETURN TO CALLING PAGE		
		2392	*			
		2393	*****			

[illegible][illegible][illegible]

FZSPRT - S/3 BASIC INTERPRETER PRINT STATEMENT EXEC RTN									
ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15,	MOD 00 31/05/21 PAGE 230
	35AF	F2	01 09		2379	JNE	FZS490	BRANCH IF NO INSIGNIFICANT ZERO	
	35B2	5C	00 D5 D6		2380	MVC	FZSXWK-1(,@BR),FZSXWK(1,@BR)	SHIFT SIGNIFICANT DIGIT	
	35B6	1F	00 0D56 CA		2381	SLC	I\$PARM-1,FZS2B1(1,@BR)	DECREMENT DATA CHARACTER COUNT	
					2382	*			
					2383	*	MOVE OUTPUT FORM OF EXPONENT TO THE DATA PRINT FIELD		
					2384	*			
	35BB	7C	04 C4		2385	FZS490 MVI	FZS500+@D1(,@BR),FZSLXB	SET DIP TO ESTABLISH	
	35BE	5E	00 C4 64		2386	ALC	FZS500+@D1(,@BR),FZS430+@D1(1,@BR)	* EXPONENT POSITION	
	35C2	9C	03 00 D6		2387	FZS500 MVC	*-*(,@XR),FZSXWK(FZSLXB,@BR)	MOVE EXPONENT TO PRINT FIELD	
					2388	*			
					2389	*	RETURN CONTROL TO THE CALLING PAGE		
					2390	*			
	35C6	C0	87 12D3		2391	FZS510 B	I\$RTRN	RETURN TO CALLING PAGE	
					2392	*			
					2393	*****			

[illegible]

FZSPRT - S/3 BASIC INTERPRETER PRINT STATEMENT EXEC RTN							
ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00 31/05/21 PAGE 230	
	35AF F2 01 09		2379	JNE	FZS490	BRANCH IF NO INSIGNIFICANT ZERO	
	35B2 5C 00 D5 D6		2380	MVC	FZSXWK-1(,@BR),FZSXWK(1,@BR)	SHIFT SIGNIFICANT DIGIT	
	35B6 1F 00 0D56 CA		2381	SLC	I\$PARM-1,FZS2B1(1,@BR)	DECREMENT DATA CHARACTER COUNT	
			2382	*			
			2383	*	MOVE OUTPUT FORM OF EXPONENT TO THE DATA PRINT FIELD		
			2384	*			
	35BB 7C 04 C4		2385	FZS490 MVI	FZS500+@D1(,@BR),FZSLXB	SET DIP TO ESTABLISH	
	35BE 5E 00 C4 64		2386	ALC	FZS500+@D1(,@BR),FZS430+@D1(1,@BR)	* EXPONENT POSITION	
	35C2 9C 03 00 D6		2387	FZS500 MVC	*-(,@XR),FZSXWK(FZSLXB,@BR)	MOVE EXPONENT TO PRINT FIELD	
			2388	*			
			2389	*	RETURN CONTROL TO THE CALLING PAGE		
			2390	*			
	35C6 C0 87 12D3		2391	FZS510 B	I\$RTRN	RETURN TO CALLING PAGE	
			2392	*			
			2393	*****			

FZSPRT - S/3 BASIC INTERPRETER PRINT STATEMENT EXEC RTN									
ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15,	MOD 00 31/05/21 PAGE 230
	35AF	F2	01 09		2379	JNE	FZS490	BRANCH IF NO INSIGNIFICANT ZERO	
	35B2	5C	00 D5 D6		2380	MVC	FZSXWK-1(,@BR),FZSXWK(1,@BR)	SHIFT SIGNIFICANT DIGIT	
	35B6	1F	00 0D56 CA		2381	SLC	I\$PARM-1,FZS2B1(1,@BR)	DECREMENT DATA CHARACTER COUNT	
					2382	*			
					2383	*	MOVE OUTPUT FORM OF EXPONENT TO THE DATA PRINT FIELD		
					2384	*			
	35BB	7C	04 C4		2385	FZS490 MVI	FZS500+@D1(,@BR),FZSLXB	SET DIP TO ESTABLISH	
	35BE	5E	00 C4 64		2386	ALC	FZS500+@D1(,@BR),FZS430+@D1(1,@BR)	* EXPONENT POSITION	
	35C2	9C	03 00 D6		2387	FZS500 MVC	*-*(,@XR),FZSXWK(FZSLXB,@BR)	MOVE EXPONENT TO PRINT FIELD	
					2388	*			
					2389	*	RETURN CONTROL TO THE CALLING PAGE		
					2390	*			
	35C6	C0	87 12D3		2391	FZS510 B	I\$RTRN	RETURN TO CALLING PAGE	
					2392	*			
					2393	*****			

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00 31/05/21 PAGE 231
			2395	*****	*****	
			2396	* PRINT EXECUTION ROUTINE CONSTANTS (2ND VM PAGE)	*	
			2397	*****	*****	
			2398	*		
35CA	01	35CA	2399	FZS2B1 DC	IL1'1' BINARY INTEGER +1	
35CB	F1	35CB	2400	FZSDC1 DC	DL1'1' DECIMAL INTEGER +1	
35CC	F5	35CC	2401	FZSDC5 DC	DL1'5' DECIMAL INTEGER +5	
			2402	*		
35CD	80	35CD	2403	FZS2XZ DC	AL1(B@NXZR) ZERO NORMALIZED EXPONENT	
			2404	*		
		0004	2405	FZSLXB EQU	4 LENGTH OF EXPONENT IMAGE	
35CE	C54EF0F0	35D1	2406	FZSEXB DC	CL(FZSLXB)'E+00' EXPONENT IMAGE FOR OUTPUT	
			2407	*		
			2408	*****	*****	
			2409	* PRINT EXECUTION ROUTINE WORK AREAS (2ND VM PAGE)	*	
			2410	*****	*****	
			2411	*		
35D2		35D2	2412	FZS2BX DS	CL1 BINARY EXPONENT MAGNITUDE	
35D3		35D6	2413	FZSXWK DS	CL(FZSLXB) EXPONENT CONSTRUCT AREA	
			2414	*		
		0002	2415	FZSLXM EQU	2 LENGTH OF DECMAL EXP MAGNITUDE	
35D7		35D8	2416	FZSDAC DS	CL(FZSLXM) B TO D DECIMAL ACCUMULATOR	
			2417	*		
			2418	*****	*****	

ERR	LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 31/05/21 PAGE 232
				2420	*****	
				2421	* VIRTUAL MEMORY PRINT EXECUTION ROUTINE (3RD VM PAGE)	*
				2422	* * OUTPUTS FORMATTED DATA ELEMENT TO MATRIX PRINTER	*
				2423	* * CONTROLS PRINTER CARRIER DEPENDING ON SPECIFIED CONTROL CODE	*
				2424	* INPUT -	*
				2425	* * RUN-TIME STACK - CONTAINS FORMATTED ELEMENT, IF PRESENT	*
				2426	* * I\$PARM - 1 BYTE, CONTAINS CONTROL CODE BRANCH DISPLACEMENT	*
				2427	* * I\$PARM-1 - 1 BYTE, CONTAINS FORMATTED ELEMENT CHARACTER COUNT	*
				2428	* * I\$WRK1 - 2 BYTES, CONTAINS CORE ADDR OF PRINT AREA LEFT BYTE	*
				2429	* * I\$WRK2 - 2 BYTES, CONTAINS VALUE FOR \$PRDEV 'CRT ONLY' COND	*
				2430	* * I\$SLLC - 1 BYTE, CONTAINS OUTPUT ELEMENT LENGTH CODE (LNG - 1)	*
				2431	*	*
				2432	* OUTPUT -	*
				2433	* * PRINTED ELEMENT AND/OR CARRIER CONTROL ON MATRIX PRINTER	*
				2434	*****	
				2435	*	
				2436	* ESTABLISH ADDRESSABILITY FOR PRINT ROUTINE (3RD VM PAGE)	
				2437	*	
				2438	*FZSP3B VPAGE 0	
3600				2439	ORG *,256,0	SET STARTING ADDRESS
	3600			2440	FZSP3B EQU *	START OF PROGRAM CODING
3501				2441	ORG *-255	RESET IAR TO PAGE
3600				2442	ORG *,256,0	* BOUNDARY ADDRESS
	3600			2443	USING *,@BR	SET PAGE BASE ADDRESS
3600				2444	ORG FZSP3B	RESET STARTING ADDRESS
				2445	*** END OF EXPANSION ***	
				2446	*	
				2447	* PAGE ENTRY - TEST FOR MATRIX PRINTER ACTIVE ON SYSTEM	
				2448	*	
3600 0D 01 044B 0D5B				2449	FZS600 CLC \$PRDEV,I\$WRK2(@CADDR)	IF PRINTER NOT A SYSTEM PRINT ?
3606 F2 02 BF				2450	JNL FZS740	* DEVICE, GO OUTPUT TO THE CRT
				2451	*	
				2452	* INITIALIZE FOR OUTPUT TO THE MATRIX PRINTER	
				2453	*	
3609 4C 00 6A 03C0				2454	MVC FZS3RM(,@BR),\$RMRGN(1)	SET MP RIGHT MARGIN PARAMETER
				2455	*	
				2456	* INITIALIZE THE ELEMENT PRINT PARAMETER LIST	
				2457	*	
360E 7C 40 F2				2458	MVI FZS3PF(,@BR),@PRINT	SET FUNCTION FOR PRINT ONLY
3611 4C 00 F3 0D56				2459	MVC FZS3PC(,@BR),I\$PARM-1(1)	SET COUNT = ELEMENT CHAR COUNT
3616 4C 01 F5 0D59				2460	MVC FZS3PA(,@BR),I\$WRK1(@CADDR)	SET PRINT AREA CORE ADDRESS
				2461	*	
				2462	* TEST FOR AN ARITHMETIC ELEMENT - RETURN CARRIER IF ARITHMETIC	
				2463	* ELEMENT LENGTH EXCEEDS OUTPUT LINE MARGIN	
				2464	*	
361B 5C 00 DB F3				2465	MVC FZS3CC(,@BR),FZS3PC(1,@BR)	SET PARAM = ELEMENT CHAR CNT
361F 3D 12 0BA1				2466	CLI I\$SLLC,I@LCRV-1	IF CURR ELEMENT IS ARITHMETIC ?
3623 D0 01 D2				2467	BNE FZS760(,@BR)	* LINK TO RETURN CARR ON COND
				2468	*	
				2469	* BRANCH TO APPROPRIATE ROUTINE DEPENDING ON CONTROL CODE	
				2470	*	
3626 4C 00 2D 0D57				2471	MVC FZS605+@D1(,@BR),I\$PARM(1)	MOVE CONTROL DISP TO JUMP INST
362B F2 87 00				2472	FZS605 J *-*	GO EXECUTE CONTROL CODE ROUTINE
				2473	*	
				2474	*****	

## FZSPRT - S/3 BASIC INTERPRETER PRINT STATEMENT EXEC RTN

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	31/05/21	PAGE 233
				2476		*****				
				2477	*	OUTPUT ROUTINE FOR PRINT CONTROL CODES 1, 9, 11, 13, 15				*
				2478		*****				
				2479	*					
				2480	*	PRINT THE FORMATTED ELEMENT ONLY (WHEN SIGNIFICANT)				
				2481	*					
362E	7D	00	F3	2482	FZS610	CLI	FZS3PC(,@BR),@ZERO	IF ELEMENT CHAR COUNT NOT ZERO		
3631	F2	81	B8	2483		JE	FZS790	EXIT ROUTINE W/O PRINTING	1-5	
3634	1C	01	144A	2484		MVC	I\$VADR,FZSPCH(@VADDR,@BR)	VM PATCH PAGE ENTRY ADDR	1-5	
3639	C0	87	1358	2485		B	I\$CVAD	LOAD PATCH PAGE	1-5	
363D	4C	01	45 144C	2486		MVC	FZS615+@OP1(@CADDR,@BR),I\$CADR	MOVE CADDR TO BRANCH	1-5	
3642	C0	87	0000	2487	FZS615	B	*-*	BRANCH TO PATCH PAGE	1-5	
				2488	*					
				2489		*****				
				2490	*	OUTPUT ROUTINE FOR PRINT CONTROL CODE 2				*
				2491		*****				
				2492	*					
				2493	*	ESTABLISH FULL PRINT ZONE OUTPUT FORMAT (ARITHMETIC ELEMENT)				
				2494	*					
3646	7C	12	DB	2495	FZS620	MVI	FZS3CC(,@BR),I@LFPZ	SET PARAM - FULL PRINT ZONE		
3649	F2	87	18	2496		J	FZS636	BRANCH TO TEST LINE CAPACITY		
				2497	*					
				2498		*****				

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	31/05/21	PAGE 234
				2500		*****				
				2501	*	OUTPUT ROUTINE FOR PRINT CONTROL CODE 3				*
				2502		*****				
				2503	*					
				2504	*	ESTABLISH PACKED PRINT ZONE OUTPUT FORMAT (ARITHMETIC ELEMENT) -				
				2505	*	THIS ZONE WILL BE 6, 9, 12, 15, OR 18 CHARACTERS LONG DEPENDING ON				
				2506	*	THE LENGTH OF THE ARITHMETIC ELEMENT TO BE PRINTED.				
				2507	*					
364C	7C	04	DB	2508	FZS630	MVI	FZS3CC(,@BR),2*I@LPPZ-2	SET LENGTH ACCUM TO MINIMUM		
				2509	*			* ELEMENT LENGTH LIMIT (4)		
364F	5D	00	F3 DB	2510	FZS632	CLC	FZS3PC(,@BR),FZS3CC(1,@BR)	IF ELEMENT LENGTH WITHIN LIMIT		
3653	F2	04	0A	2511		JNH	FZS634	* BRANCH TO EXIT THIS LOOP		
3656	5E	00	DB F1	2512		ALC	FZS3CC(,@BR),FZS3PZ(1,@BR)	ADD PACKED ZONE INCR TO ACCUM		
365A	7D	10	DB	2513	FZS633	CLI	FZS3CC(,@BR),I@LFPZ-2	IF LENGTH ACCUM NOT MAXIMUM		
365D	D0	82	4F	2514		BL	FZS632(,@BR)	* GO REPEAT ELEMENT LENGTH TEST		
				2515	*					
3660	5E	00	DB F0	2516	FZS634	ALC	FZS3CC(,@BR),FZS3B2(1,@BR)	ADJUST ACCLM TO MAKE PACKED		
				2517	*			* PRINT ZONE FIELD LENGTH		
				2518	*					
				2519	*	TEST LINE CAPACITY TO CONTAIN CURRENT PRINT ZONE FIELD - WHEN RIGHT				
				2520	*	MARGIN IS EXCEEDED, LINE HAS CAPACITY FOR THE DATA ELEMENT BUT NOT				
				2521	*	FOR THE ENTIRE PRINT ZONE ... IN THIS CASE, PRINT ELEMENT ONLY AND				
				2522	*	RETURN THE CARRIER				
				2523	*					
3664	4E	00	DB 03C2	2524	FZS636	ALC	FZS3CC(,@BR),\$PRPOS(1)	ADD PRINT ZONE LNG TO CURRENT		
3669	7D	00	DB	2525	FZS638	CLI	FZS3CC(,@BR),*-*	* CARRIER POSITION - BRANCH		
366C	F2	84	12	2526		JH	FZS655	* IF RIGHT MARGIN IS EXCEEDED		
				2527	*					
				2528	*	LINE HAS CAPACITY FOR ENTIRE PRINT ZONE - PRINT ELEMENT AND SPACE				
				2529	*	TO THE SPECIFIED ZONE POSITION				
				2530	*					
366F	4F	00	DB 03C2	2531	FZS640	SLC	FZS3CC(,@BR),\$PRPOS(1)	RESTORE CURRENT PRINT ZONE LNG		
3674	5C	00	F3 DB	2532		MVC	FZS3PC(,@BR),FZS3CC(1,@BR)	SET COUNT - CAR PRT ZONE LNG		
3678	F2	87	3E	2533		J	FZS710	GO PRINT ELEMENT AND SPACE CARR		
				2534	*					
				2535		*****				

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 31/05/21 PAGE 235
		2537		*****	
		2538	*	OUTPUT ROUTINE FOR PRINT CONTROL CODES 4, 12, 16	*
		2539		*****	
		2540	*		
		2541	*	TEST ELEMENT SIGNIFICANCE - RETURN CARRIER ONLY WHEN NOT SIGNIFICANT	
		2542	*		
367B	7D 00 F3	2543	FZS650 CLI	FZS3PC(,@BR),@ZERO	ELEMENT CHAR COUNT IS ZERO ?
367E	F2 81 20	2544		JE FZS680	* GO RETURN THE CARRIER ONLY
		2545	*		
		2546	*	ELEMENT IS SIGNIFICANT - PRINT ELEMENT AND RETURN CARRIER	
		2547	*		
3681	7C C0 F2	2548	FZS655 MVI	FZS3PF(,@BR),@PRETR	SET PRINT & CARR RETURN FUNC
3684	F2 87 32	2549		J FZS710	GO PRINT ELEMENT AND RTRN CARR
		2551		*****	
		2552	*	OUTPUT ROUTINE FOR PRINT CONTROL CODE 5	*
		2553		*****	
		2554	*		
		2555	*	ESTABLISH FULL PRINT ZONE SPACING ONLY	
		2556	*		
3687	7C 12 F3	2557	FZS660 MVI	FZS3PC(,@BR),I@LFPZ	SET COUNT FOR FULL PRINT ZONE
368A	F2 87 03	2558		J FZS675	BRANCH TO EXECUTE SPACING
		2560		*****	
		2561	*	OUTPUT ROUTINE FOR PRINT CONTROL CODE 6	*
		2562		*****	
		2563	*		
		2564	*	ESTABLISH PACKED PRINT ZONE INCREMENT SPACING ONLY	
		2565	*		
368D	7C 03 F3	2566	FZS670 MVI	FZS3PC(,@BR),I@LPPZ	SET COUNT FOR PACKED ZONE INCR
		2567	*		
		2568	*	PRINT CURRENT ZONE SPACE, OR RETURN CARRIER IF END OF LINE IS HIT	
		2569	*		
3690	5C 00 DB F3	2570	FZS675 MVC	FZS3CC(,@BR),FZS3PC(1,@BR)	SET PARAM FOR CURRENT ZONE LNG
3694	D0 87 D2	2571		B FZS760(,@BR)	LINK TO RETURN CARRIER ON COND
3697	5D 00 DB 6A	2572		CLC FZS3CC(,@BR),FZS3RM(1,@BR)	IF CARRIER WAS NOT RETURNED
369B	F2 04 1B	2573		JNH FZS710	* GO PRINT CURRENT ZONE SPACE,
369E	F2 87 2D	2574		J FZS750	* ELSE EXIT RTN W/0 PRINTING
		2575	*		
		2576		*****	

## FZSPRT - S/3 BASIC INTERPRETER PRINT STATEMENT EXEC RTN

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 31/05/21 PAGE 236
		2578		*****	
		2579		* OUTPUT ROUTINE FOR PRINT CONTROL CODE 7	*
		2580		*****	
		2581		*	
		2582		* ESTABLISH CARRIER RETURN ONLY	
		2583		*	
36A1 D2 02 F6		2584	FZS680 LA	FZS3CR(,@BR),@XR	LOAD CARRIER RETURN PPL CADDR
36A4 F2 87 15		2585	J	FZS720	GO EXECUTE CARRIER RETURN
		2587		*****	
		2588		* OUTPUT ROUTINE FOR PRINT CONTROL CODE 8	*
		2589		*****	
		2590		*	
		2591		* RETURN CARRIER IF FULL PRINT ZONE EXCEEDS LINE CAPACITY	
		2592		*	
36A7 7C 12 DB		2593	FZS690 MVI	FZS3CC(,@BR),I@LFPZ	SET PARAM FOR PRINT ZONE
36AA D0 87 D2		2594	B	FZS760(,@BR)	LINK TO RETURN CARRIER ON COND
36AD F2 87 0F		2595	J	FZS730	GO TEST FOR CRT ACTIVE ON SYSTEM
		2597		*****	
		2598		* OUTPUT ROUTINE FOR PRINT CONTROL CODE 10	*
		2599		*****	
		2600		*	
		2601		* RETURN CARRIER IF FULL PRINT ZONE EXCEEDS LINE CAPACITY	
		2602		*	
36B0 7C 12 DB		2603	FZS695 MVI	FZS3CC(,@BR),I@LFPZ	SET PARAM FOR FULL PRINT ZONE
36B3 D0 87 D2		2604	B	FZS760(,@BR)	LINK TO RETURN CARRIER ON COND
		2605		*	
		2606		*****	



## FZSPRT - S/3 BASIC INTERPRETER PRINT STATEMENT EXEC RTN

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 31/05/21 PAGE 237
		2608		*****	
		2609		* OUTPUT ROUTINE FOR PRINT CONTROL CONTROL CODE 14	
		2610		*****	
		2611		*	
		2612		* ESTABLISH FULL PRINT ZONE OUTPUT FORMAT (CHARACTER ELEMENT)	
		2613		*	
36B6	7C 12 F3	2614	FZS700 MVI	FZS3PC(,@BR),I@LFPZ SET COUNT FOR ZONE	
		2615		*	
		2616		* EXECUTE ELEMENT OUTPUT TO THE MATRIX PRINTER	
		2617		*	
36B9	D2 02 F2	2618	FZS710 LA	FZS3PL(,@BR),@XR LOAD DATA OLTOLT CORE ADOR	
36BC	D0 87 E3	2619	FZS720 B	FZS780(,@BR) LINK TO EXECUTE PRINTER OUTPUT	
		2620		*	
		2621		* TEST FOR THE CRT ACTIVE AS A SISTEM PRINT DEVICE	
		2622		*	
36BF	0D 00 044A 0D5A	2623	FZS730 CLC	\$PRDEV-1,I\$WRK2-1(1) IF CRT IS NOT A SYSTEM PRINT	
36C5	F2 82 06	2624	JL	FZS750 * DEVICE, GO EXIT THIS ROUTINE	
		2625		*	
		2626		* CRT ACTIVE - SET UP AND OUTPUT TO CRT USINS CRT LINE WIDTH	
		2627		*	
36C8	C0 87 12B1	2628	FZS740 B	I\$CALL LINK TO EXECUTE PRINT ON CRT	
36CC	3700	2629	DC	AL(@VADDR)(FZS800) PRINT CRT RTN VIRTUAL ADDRESS	
		2630		*	
		2631		* RETURN TO PTINT ROUTINE 1ST VM PAGE	
		2632		*	
36CE	C0 87 12D3	2633	FZS750 B	I\$RTRN RETURN TO 1ST PRINT RTN PAGE	
		2634		*	
		2635		*****	

## FZSPRT - S/3 BASIC INTERPRETER PRINT STATEMENT EXEC RTN

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 31/05/21 PAGE 238
		2637		*****	
		2638	*	PRINTER CARRIER RETURN ROUTINE -	*
		2639	*	* RETURNS PRINTER CARRIER WHEN SPECIFIED LENGTH PARAMETER	*
		2640	*	(FZS3CC) EXCEEDS THE CURRENT PRINT LINE CAPACITY.	*
		2641		*****	
		2642	*		
36D2	74 08 EF	2643	FZS760 ST	FZS790+@OP1(,@BR),@ARR STORE RETURN BRANCH ADDRESS	
		2644	*		
		2645	*	TEST LINE CAPACITY TO CONTAIN CURRENT PRINT REGION LENGTH	
		2646	*		
36D5	4E 00 DB 03C2	2647	ALC	FZS3CC(,@BR),\$PRPOS(1) ADD PRINT REGION LENGTH TO CURR	
36DA	7D 00 6A	2648	FZS770 CLI	FZS3RM(,@BR),*-* * CARRIER POSITION - BRANCH IF	
36DD	F2 02 0C	2649	JNL	FZS790 * RIGHT MARGIN NOT EXCEEDED	
		2650	*		
		2651	*	RIGHT MARGIN EXCEEDED - RETURN MATRIX PRINTER CARRIER	
		2652	*		
36E0	D2 02 F6	2653	LA	FZS3CR(,@BR),@XR LOAD CARRIER RETURN PPL CADDR	
		2655		*****	
		2656	*	PRINTER OUTPUT INTERFACE -	*
		2657	*	* EXECUTES MATRIX PRINTER OUTPUT AS SPECIFIED IN PRINT PARAM-	*
		2658	*	ETER LIST REFERENCED BY REGISTER @XR.	*
		2659		*****	
36E3	74 08 EF	2660	FZS780 ST	FZS790+@OP1(,@BR),@ARR STORE RETURN BRANCH ADDRESS	
36E6	C0 87 12B1	2661	B	I\$CALL LINK TO EXECUTE PRINTER IOCR	
36EA	2800	2662	DC	AL(@VADDR)(V\$SPRT) MATRIX PRINTER IOCR VADDR	
		2663	*		
		2664	*	RETURN TO CALLING ROUTINE	
		2665	*		
36EC	C0 87 0000	2666	FZS790 B	*-* RETURN BRANCH	
		2667		*****	

## FZSPRT - S/3 BASIC INTERPRETER PRINT STATEMENT EXEC RTN

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 31/05/21 PAGE 239
		2669		*****	
		2670	*	PRINT EXECUTION ROUTINE CONSTANTS (3RD VM PAGE)	*
		2671		*****	
		2672	*		
36F0 02		36F0 2673	FZS3B2 DC	IL1'2'	BINARY INTEGER +2
		2674	*		
36F1 03		36F1 2675	FZS3PZ DC	AL1(I@LPPZ)	LENGTH OF PACKED ZONE INCR
		2677		*****	
		2678	*	PRINT EXECUTION ROUTINE WORK AREAS (3RD VM PAGE)	*
		2679		*****	
		366A 2680	FZS3RM EQU	FZS638+@Q	MATRIX PRINTER RIGHT MARGIN
		36DB 2681	FZS3CC EQU	FZS770+@Q	PRINT AREA CHARACTER COUNT
		2682	*		
		2683	*FZS3PL PPL		
		36F2 2684	FZS3PL EQU	*	PPL ADDRESS
36F2 00		36F2 2685		DC AL1(*-*)	FUNCTION REQUESTED
36F3 00		36F3 2686		DC AL1(*-*)	PRINT COUNT
36F4 0000		36F5 2687		DC AL2(*-*)	DATA ADDRESS
		2688	***	END OF EXPANSION ***	
		2689	*		
		36F2 2690	FZS3PF EQU	FZS3PL+@PCTRL	PRINT FUNCTION PARAMETER
		36F3 2691	FZS3PC EQU	FZS3PL+@PRCNT	PRINT AREA COUNT PARAMETER
		36F5 2692	FZS3PA EQU	FZS3PL+@PDATA	PRINT AREA COUNT PARAMETER
		2693	*		
		2694	*FZS3CR PPL	FUNC-@RETRN,CNT-@RTRNC	
		36F6 2695	FZS3CR EQU	*	PPL ADDRESS
36F6 80		36F6 2696		DC AL1(@RETRN)	FUNCTION REQUESTED
36F7 80		36F7 2697		DC AL1(@RTRNC)	PRINT COUNT
36F8 0000		36F9 2698		DC AL2(*-*)	DATA ADDRESS
		2699	***	END OF EXPANSION ***	
		2700	*		
36FA 5359		36FB 2701	FZSPCH DC	AL2(V\$PCH2+FZS633-@Q-FZSP3B)	PATCH PAGE ENTRY ADDR 1-3
		2702		*****	

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 31/05/21 PAGE 240
		2704		*****	
		2705	*	VIRTUAL MEMORY PRINT EXECUTION ROUTINE 4TH VM PAGE	*
		2706	*	* OUTPUTS FORMATTED DATA ELEMENT TO CRT DISPLAY UNIT	*
		2707	*	* CONTROLS CRT CURSOR DEPENDING ON SPECIFIED CONTROL CODE	*
		2708	*		*
		2709	*	INPUT -	*
		2710	*	* RUN-TIME STACK - CONTAINS FORMATTED ELEMENT, IF PRESENT	*
		2711	*	* I\$PARM - 1 BYTE, CONTAINS CONTROL CODE BRANCH DISPLACEMENT	*
		2712	*	* I\$PARM-1 - 1 BYTE, CONTAINS FORMATTED ELEMENT CHARACTER COUNT	*
		2713	*	* I\$WRK1 - 2 BYTES, CONTAINS CORE ADDR OF PRINT AREA LEFT BYTE	*
		2714	*	* I@WRK2 - 2 BYTES, CONTAINS VALUE FOR \$PRDEV 'CRT ONLY' COND	*
		2715	*	* ISSLLC - 1 BYTE, CONTAINS OUTPUT ELEMENT LENGTH CODE (LNG - 1)	*
		2716	*		*
		2717	*	OUTPUT -	*
		2718	*	* DISPLAYED ELEMENT AND/OR CURSOR CONTROL ON CRT DISPLAY UNIT	*
		2719		*****	
		2720	*		
		2721	*	ESTABLISH ADDRESSABILITY FOR PRINT ROUTINE (4TH VM PAGE)	
		2722	*		
		2723		*FZSP4B VPAGE 0	
3700		2724		ORG *,256,0 SET STARTING ADDRESS	
		2725	FZSP4B EQU *	START OF PROGRAM CODING	
3601		2726		ORG *-255 RESET IAR TO PAGE	
3700		2727		ORG *,256,0 * BOUNDARY ADDRESS	
		2728		USING *,@BR SET PAGE BASE ADDRESS	
3700		2729		ORG FZSP4B RESET STARTING ADDRESS	
		2730		*** END OF EXPANSION ***	
		2731	*		
		2732	*	PAGE ENTRY - ESTABLISH CRT IOCR EXECUTION CORE ADDRESS	
		2733	*		
3700 4C 01 D7 0D5B		2734	FZS800 MVC	FZS982+@OP1(,@BR),I\$WRK2(@CADDR) SET CRT EXECUTION CADDR	
		2735	*		
		2736	*	INITIALIZE FOR OUTPUT TO THE CRT DISPLAY UNIT	
		2737	*		
3705 7C 40 64		2738		MVI FZS4RM(,@BR),@DLNLG SET CRT RIGHT MARGIN PARAMETER	
		2739	*		
		2740	*	INITIALIZE THE ELEMENT PRINT PARAMETER LIST	
		2741	*		
3708 7C 40 E0		2742		MVI FZS4PF(,@BR),@PRINT SET FUNCTION FOR PRINT ONLY	
370B 4C 00 E1 0D56		2743		MVC FZS4PC(,@BR),I\$PARM-1(1) SET COUNT - ELEMENT CHAR COUNT	
3710 4C 01 E3 0D59		2744		MVC FZS4PA(,@BR),I\$WRK1(@CADDR) SET PRINT AREA CODE ADDRESS	
		2745	*		
		2746	*	TEST FOR AN ARITHMETIC ELEMENT - RETURN CURSOR IF ARITHMETIC	
		2747	*	ELEMENT LENGTH EXCEEDS OUTPUT LINE MARGIN	
		2748	*		
3715 5C 00 C6 E1		2749		MVC FZS4CC(,@BR),FZS4PC(1,@BR) SET PARAM = ELEMENT CHAR CNT	
3719 3D 12 0BA1		2750		CLI I\$SLLC,I@LCRV-1 IF CURR ELEMENT IS ARITHMETIC	
371D D0 01 BD		2751		BNE FZS960(,@BR) * LINK TO RTRN CURSOR ON COND	
		2752	*		
		2753	*	BRANCH TO APPROPRIATE ROUTINE DEPENDING ON CONTROL CODE	
		2754	*		
3720 4C 00 27 0D57		2755		MVC FZS805+@D1(,@BR),I\$PARM(1) MOVE CONTROL DISP TO JUMP INST	
3725 F2 87 00		2756	FZS805 J	*-* GO EXEC CONTROL CODE ROUTINE	
		2757	*		
		2758		*****	

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	31/05/21	PAGE 241
				2760			*****			
				2761			* OUTPUT ROUTINE FOR PRINT CONTROL CODES 1, 9, 11, 13, 15			*
				2762			*****			
				2763			*			
				2764			* DISPLAY THE FORMATTED ELEMENT ONLY (WHEN SIGNIFICANT)			
				2765			*			
3728	7D	00	E1	2766	FZS810	CLI	FZS4PC(,@BR),@ZERO		IF ELEMENT CHAR COUNT NOT ZERO	
372B	F2	01	85	2767		JNE	FZS910		* GO DISPLAY ELEMENT ONLY,	
372E	F2	87	88	2768		J	FZS950		* ELSE EXIT RTN W/O DISPLAYING	
				2769			*			
3731	000000000000000000	373F		2770		DC	XL15'00'		PATCH SPACE	1-5
				2772			*****			
				2773			* OUTPUT ROUTINE FOR PRINT CONTROL CODE 2			*
				2774			*****			
				2775			*			
				2776			* ESTABLISH FULL PRINT ZONE OUTPUT FORMAT (ARITHMETIC ELEMENT)			
				2777			*			
3740	7C	12	C6	2778	FZS820	MVI	FZS4CC(,@BR),I@LFPZ		SET PARAM = FULL PRINT ZONE	
3743	F2	87	18	2779		J	FZS836		BRANCH TO TEST LINE CAPACITY	
				2780			*			
				2781			*****			

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 31/05/21 PAGE 242
		2783		*****	
		2784		* OUTPUT ROUTINE FOR PRINT CONTROL CODE 3	*
		2785		*****	
		2786		*	
		2787		* ESTABLISH PACKED PRINT ZONE OUTPUT FORMAT (ARITHMETIC ELEMENT) -	
		2788		* THIS ZONE WILL BE 6, 9, 12, 15, OR 18 CHARACTERS LONG DEPENDING ON	
		2789		* THE LENGTH OF THE ARITHMETIC ELEMENT TO BE PRINTED	
		2790		*	
3746	7C 04 C6	2791	FZS830 MVI	FZS4CC(,@BR),2*I@LPPZ-2	SET LENGTH ACCUN TO MINIMUM
		2792		*	* ELEMENT LENGTH LIMIT (4)
3749	5D 00 E1 C6	2793	FZS832 CLC	FZS4PC(,@BR),FZS4CC(1,@BR)	IF ELEMENT LENGTH WITHIN LIMIT
374D	F2 04 0A	2794		JNH FZS834	* BRANCH TO EXIT THIS LOOP
3750	5E 00 C6 DF	2795		ALC FZS4CC(,@BR),FZS4PZ(1,@BR)	ADD PACKED ZONE INCR TO ACCUM
3754	7D 10 C6	2796		CLI FZS4CC(,@BR),I@LFPZ-2	IF LENGTH ACCUM NOT MAXIMUM
3757	D0 82 49	2797		BL FZS832(,@BR)	* GO REPEAT ELEMENT LENGTH TEST
		2798		*	
375A	5E 00 C6 DE	2799	FZS834 ALC	FZS4CC(,@BR),FZS4B2(1,@BR)	ADJUST ACCUM TO MAKE PACKED
		2800		*	* PRINT ZONE FIELD LENGTH
		2801		*	
		2802		* TEST LINE CAPACITY TO CONTAIN CURRENT POINT ZONE FIELD - WHEN RIGHT	
		2803		* MARGIN IS EXCEEDED, LINE HAS CAPACITY FOR TED DATA ELEMENT BUT NOT	
		2804		* FOR THE ENTIRE PRINT ZONE ... IN THIS CASE, DISPLAY ELMEMENMT ONLY	
		2805		* AND RETURN THE CURSOR.	
		2806		*	
375E	4E 00 C6 03E2	2807	FZS836 ALC	FZS4CC(,@BR),\$CRPOS(1)	ADD PRINT ZONE LNG TO CURRENT
3763	7D 00 C6	2808	FZS838 CLI	FZS4CC(,@BR),*-*	* CURSOR POSITION - BRANCH
3766	F2 82 12	2809		JM FZS855	* IF RIGHT MARGIN IS EXCEEDED
		2810		*	
		2811		* LINE HAS CAPACITY FOR ENTIRE PRINT ZONE - DISPLAY ELEMENT AND SPACE	
		2812		* TO THE SPECIFIED ZONE POSITION	
		2813		*	
3769	4F 00 C6 03E2	2814	FZS840 SLC	FZS4CC(,@BR),\$CRPOS(1)	RESTORE CURRENT PRINT ZONE LNG
376E	5C 00 E1 C6	2815		MVC FZS4PC(,@BR),FZS4CC(1,@BR)	SET COUNT = CURR PRT ZONE LNG
3772	F2 87 3E	2816		J FZS910	GO DISPLAY ELEM & SPACE CURSOR
		2817		*	
		2818		*****	

## FZSPRT - S/3 BASIC INTERPRETER PRINT STATEMENT EXEC RTN

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00 31/05/21 PAGE 243
				2820			*****	
				2821	*		OUTPUT ROUTINE FOR PRINT CONTROL CODES 4, 12, 16	*
				2822			*****	
				2823	*			
				2824	*		TEST ELEMENT SIGNIFICANCE - RETURN CURSOR NO WHEN NOT SIGNIFICANT	
				2825	*			
3775	7D	00	E1	2826	FZS850	CLI	FZS4PC(,@BR),@ZERO	IF ELEMENT CHAR COUNT IS ZERO
3778	F2	81	20	2827		JE	FZS880	* GO RETURN THE CURSOR ONLY
				2828	*			
				2829	*		ELEMENT IS SIGNIFICANT - DISPLAY ELEMENT AND RETURN CURSOR	
				2830	*			
377B	7C	C0	E0	2831	FZS855	MVI	FZS4PF(,@BR),@PRETR	SET PRINT & CARR RETURN FUNC
377E	F2	87	32	2832		J	FZS910	GO DISPLAY ELEM AND RTRN CURSOR
				2834			*****	
				2835	*		OUTPUT ROUTINE FOR PRINT CONTROL CODE 5	*
				2836			*****	
				2837	*			
				2838	*		ESTABLISH FULL PRINT ZONE SPACING ONLY	
				2839	*			
3781	7C	12	E1	2840	FZS860	MVI	FZS4PC(,@BR),I@LFPZ	SET CO:AT R04 FLU *QM ZONE
3784	F2	87	03	2841		J	FZS875	BRANCH TO EXEC?TE SPACINS
				2843			*****	
				2844	*		OUTPUT ROUTINE FOR PRINT COHT4OL CODE 6	*
				2845			*****	
				2846	*			
				2847	*		ESTABLISH PACKED PRINT ZONE INCREMENT SPACING ONLY	
				2848	*			
3787	7C	03	E1	2849	FZS870	MVI	FZS4PC(,@BR),I@LPPZ	SET COUNT FOR PACKED ZONE INCR
				2850	*			
				2851	*		DISPLAY CURRENT ZONE, OR RETURN CURSOR IF END OF LINE IS HIT	
				2852	*			
378A	5C	00	C6 E1	2853	FZS875	MVC	FZS4CC(,@BR),FZS4PC(1,@BR)	SET PARAM FOR CURRENT ZONE LNG
378E	D0	87	BD	2854		B	FZS960(,@BR)	LINK TO RETURN CURSOR ON COND
3791	5D	00	C6 64	2855		CLC	FZS4CC(,@BR),FZS4RM(1,@BR)	IF CURSOS WAS NOT RETURNED
3795	F2	04	1B	2856		JNH	FZS910	* GO DISPLAY CURR ZONE SPACE
3798	F2	87	1E	2857		J	FZS950	* ELSE EXIT RTN W/O DISPLAYING
				2858	*			
				2859			*****	



ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00	31/05/21	PAGE 244
		2861		*****			
		2862		* OUTPUT ROUTINE FOR PRINT CONTROL CODE 7			*
		2863		*****			
		2864		*			
		2865		* ESTABLISH CURSOR RETURN ONLY			
		2866		*			
379B	D2 02 C6	2867	FZS880	LA FZS4CC(,@BR),@XR		LOAD CURSOR RETURN PPL CADDR	
379E	F2 87 15	2868		J FZS920		GO EXECUTE CURSOR RETURN	
		2870		*****			
		2871		* OUTPUT ROUTINE FOR PRINT CONTROL CODE 8			*
		2872		*****			
		2873		*			
		2874		* RETURN CURSOR IF FULL PRINT ZONE EXCEEDS LINE CAPACITY			
		2875		*			
37A1	7C 12 C6	2876	FZS890	MVI FZS4CC(,@BR),I@LFPZ		SET PARAM FOR FULL PRINT ZONE	
37A4	D0 87 BD	2877		B FZS960(,@BR)		LINK TO RETURN CLRSR ON COND	
37A7	F2 87 0F	2878		J FZS950		GO EXIT DISPLAY ROUTINE	
		2880		*****			
		2881		* OUTPUT ROUTINE FOR PRINT CONTROL CODE 10			*
		2882		*****			
		2883		*			
		2884		* RETURN CURSOR IF FULL PRINT ZONE EXCEEDS LINE CAPACITV			
		2885		*			
37AA	7C 12 C6	2886	FZS895	MVI FZS4CC(,@BR),I@LFPZ		SET PARAM FOR FULL PRINT ZONE	
37AD	D0 87 BD	2887		B FZS960(,@BR)		LINK TO RETURN CURSOS ON COND	
		2888		*			
		2889		*****			

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00	31/05/21	PAGE 245
		2891		*****			
		2892		* OUTPUT ROUTINE FOR PRINT CONTROL CODE 14			*
		2893		*****			
		2894		*			
		2895		* ESTABLISH FULL PRINT ZONE OUTPUT FORMAT (CHARACTER ELEMENT)			
		2896		*			
37B0	7C 12 E1	2897	FZS900	MVI FZS4PC(,@BR),I@LFPZ	SET COUNT FOR FULL PRINT ZONE		
		2898		*			
		2899		* EXECUTE ELEMENT OUTPUT TO THE CRT DISPLAY UNIT			
		2900		*			
37B3	D2 02 E0	2901	FZS910	LA FZS4PL(,@BR),@XR	LOAD DATA OUTPUT PPL CORE ADDR		
		2902		*			
37B6	D0 87 CE	2903	FZS920	B FZS980(,@BR)	LINK TO EXECUTE CRT OUTPUT		
		2904		*			
		2905		* RETURN TO PRINT ROUTINE 3RD VM PAGE			
		2906		*			
37B9	C0 87 12D3	2907	FZS950	B I\$RTRN	RETURN TO 3RD PRINT RTN PAGE		
		2908		*			
		2909		*****			

## FZSPRT - S/3 BASIC INTERPRETER PRINT STATEMENT EXEC RTN

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 31/05/21 PAGE 246
		2911		*****	
		2912	*	DISPLAY UNIT CURSOR RETURN ROUTINE -	*
		2913	*	* RETURNS CURSOR WHEN SPECIFIED LENGTH PARAMETER (FZS4CC)	*
		2914	*	EXCEEDS THE CURRENT CRT DISPLAY LINE CAPACITY.	*
		2915		*****	
		2916	*		
37BD	74 08 DD	2917	FZS960 ST	FZS990+@OP1(,@BR),@ARR	STORE RETURN BRANCH ADDRESS
		2918	*		
		2919	*	TEST LINE CAPACITY TO CONTAIN CURRENT DISPLAY REGION LENGTH	
		2920	*		
37C0	4E 00 C6 03E2	2921	ALC	FZS4CC(,@BR),\$CRPOS(1)	ADD PRINT REGION LENGTH TO CURR
37C5	7D 00 64	2922	FZS970 CLI	FZS4RM(,@BR),*-*	* CURSOR POSITION - BRANCH IF
37C8	F2 02 0F	2923	JNL	FZS990	* RIGHT MARGIN NOT EXCEEDED
		2924	*		
		2925	*	RIGHT MARGIN EXCEEDED - RETURN DISPLAY UNIT CURSOR	
		2926	*		
37CB	D2 02 E4	2927	LA	FZS4CR(,@BR),@XR	LOAD CURSOR RETURN PPL CADDR
		2928	*		
		2929		*****	
		2930	*	DISPLAY UNIT OUTPUT INTERFACE -	*
		2931	*	* EXECUTES CRT DISPLAY OUTPUT AS SPECIFIED IN PRINT PARAMETER	*
		2932	*	* LIST REFERENCED BY REGISTER @XR.	*
		2933		*****	
		2934	*		
37CE	74 08 DD	2935	FZS980 ST	FZS990+@OP1(,@BR),@ARR	STORE RETURN BRANCH ADDRESS
		2936	*		
37D1	74 02 D9	2937	ST	FZS984(,@BR),@XR	STORE PPL CORE ADDRESS
37D4	C0 87 0000	2938	FZS982 B	*-*	LINK TO EXECUTE CRT IOCR
37D8		2939	FZS984 DS	CL(@CADDR)	CRT IOCS PARAMETER LIST CADDR
		2940	*		
		2941	*	RETURN TO CALLING ROUTINE	
		2942	*		
37DA	C0 87 0000	2943	FZS990 B	*-*	RETURN BRANCH
		2944	*		
		2945		*****	

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 31/05/21 PAGE 247
		2947		*****	
		2948	*	PRINT EXECUTION ROUTINE CONSTANTS (4TH VM PAGE)	*
		2949		*****	
		2950	*		
37DE 02		37DE 2951	FZS4B2 DC	IL1'2'	BINARY INTEGER +2
		2952	*		
37DF 03		37DF 2953	FZS4PZ DC	AL1(I@LPPZ)	LENGTH OF PACKED ZONE INCR
		2954	*		
		2955		*****	
		2956	*	PRINT EXECUTION ROUTINE WORK AREAS (4TH VM PAGE)	*
		2957		*****	
		2958	*		
		3764 2959	FZS4RM EQU	FZS838+@Q	CRT DISPLAY RIGHT MARGIN
		37C6 2960	FZS4CC EQU	FZS970+@Q	PRINT AREA CHARACTER COUNT
		2961	*		
		2962	*FZS4PL PPL		
37E0 00		37E0 2963	FZS4PL EQU	*	PPL ADDRESS
37E1 00		37E0 2964	DC	AL1(*-*)	FUNCTION REQUESTED
37E2 0000		37E1 2965	DC	AL1(*-*)	PRINT COUNT
		37E3 2966	DC	AL2(*-*)	DATA ADDRESS
		2967	***	END OF EXPANSION ***	
		37E0 2969	FZS4PF EQU	FZS4PL+@PCTRL	PRINT FUNCTION PARAMETER
		37E1 2970	FZS4PC EQU	FZS4PL+@PRCNT	PRINT AREA COUNT PARAMETER
		37E3 2971	FZS4PA EQU	FZS4PL+@PDATA	PRINT AKEA CADDR PARAMETER
		2972	*		
		2973	*FZS4CR DPL	FUNC-@REYRN,CNT=@RTRNC	
37E4 80		37E4 2974	FZS4CR EQU	*	PPL ADDRESS
37E5 80		37E4 2975	DC	AL1(@RETRN)	FUNCTION REQUESTED
37E6 0000		37E5 2976	DC	AL1(@RTRNC)	PRINT COUNT
		37E7 2977	DC	AL2(*-*)	DATA ADDRESS
		2978	***	END OF EXPANSION ***	
		2979	*		
		2980		*****	
		2981	*		
		2982	***	END OF PRINT EXECUTION ROUTINE CODING ***	
		2983	#####	X'3800' #####	
		2984	*	N O T Y E T S C A N N E D O R O B J C H E C K E D ! !	
		2985	#####	X'4BFF' #####	
4BFF		2986	ORG	X'4BFF'	T E M P ! ! !

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00	31/05/21	PAGE 248
		2988		*****			
		2989	*	5703-XM1 COPYRIGHT IBM CORP. 1970			*
		2990	*	REFER TO INSTRUCTIONS ON COPY RIGHT NOTICE, 120-2083			*
		2991	*				*
		2992		*****			*
		2993	*	STATUS			*
		2994	*	VERSION 1 MODIFICATION 0			*
		2995	*				*
		2996	*	FUNCTION			*
		2997	*	* FZZVMP EXECUTION CAUSES ALL MODIFIED CORE VIRTUAL MEMORY PAGES			*
		2998	*	TO BE WRITTEN BACK TO DISK (PUSHED) OR ALL UNLOCKED CORE			*
		2999	*	VIRTUAL MEMORY PAGES TO BE LOADED INTO CORE (PULLED).			*
		3000	*	* OPERATION OF THIS ROUTINE DEPENDS UPON THE ENTRY POINT SELECTED			*
		3001	*	FOR EXECUTION -			*
		3002	*	* ENTRY POINT FZZVPS - ALL CORE VIRTUAL MEMORY PAGES REFER-			*
		3003	*	ENCED WITH A 'MODIFY' INDICATOR IN THE PAGING MODULE 'LOCK			*
		3004	*	AND READ ONLY' INDICATOR TABLE ARE WRITTEN INTO DISK			*
		3005	*	VIRTUAL MEMORY. THE 'MODIFY' INDICATOR IS UNSET IN THE			*
		3006	*	INDICATOR TABLE. THIS 'PUSH' IS AUTOMATICALLY ADJUSTED			*
		3007	*	TO PROCESS AN EXPANDED TABLE AND CORE PAGE REGION FOR			*
		3008	*	EXTENDED CORE CONFIGURATIONS.			*
		3009	*	* ENTRY POINT FZZVPL - ALL CORE VIRTUAL MEMORY PAGES REFER-			*
		3010	*	ENCED WITH A 'LOCK' INDICATOR IN THE PAGING MODULE 'LOCK			*
		3011	*	AND READ ONLY' INDICATOR TABLE ARE REPLACED WITH THE			*
		3012	*	CORRESPONDING PAGE FROM DISK VIRTUAL MEMORY. THIS 'PULL'			*
		3013	*	IS AUTOMATICALLY ADJUSTED TO PROCESS AN EXPANDED TABLE AND			*
		3014	*	CORE PAGE REGION FOR EXTENDED CORE CONFIGURATIONS.			*
		3015	*				*
		3016	*	ENTRY POINTS			*
		3017	*	* ENTRY FZZVPS - FOR PERFORMING THE 'PUSH' OPERATION.			*
		3018	*	CALLING SEQUENCE IS			*
		3019	*	B IPGCAL			*
		3020	*	DC AL2(V\$VMPS)			*
		3021	*	WHERE THE ADDRESS CONSTANT PARAMETER DEFINES THE VIRTUAL			*
		3022	*	ADDRESS OF ENTRY POINT FZZVPS.			*
		3023	*	* ENTRY FZZVPL - FOR PERFORMING THE 'PULL' OPERATION.			*
		3024	*	CALLING SEQUENCE IS			*
		3025	*	B IPGCAL			*
		3026	*	DC AL2(V\$VMPL)			*
		3027	*	WHERE THE ADDRESS CONSTANT PARAMETER DEFINES THE VIRTUAL			*
		3028	*	ADDRESS OF ENTRY POINT FXXVPL.			*
		3029	*	* IN EACH CASE, EXECUTION IS SUBJECT TO THE INPUT CONDITIONS			*
		3030	*	DESCRIBED BELOW.			*
		3031	*				*
		3032	*	INPUT			*
		3033	*	* \$EXFTR - 1 BYTE, FOR THE SYSTEM CORE EXTENSION FACTOR. THIS			*
		3034	*	CONTAINS THE NUMBER OF CORE PAGES (256-BYTE REGIONS) AVAILABLE			*
		3035	*	FOR GENERAL USE BEYOND THE 8K MINIMUM CONFIGURATION.			*
		3036	*	* PAGE INDICATOR TABLE - 10 BYTES (MINIMUM), FOR THE PAGING			*
		3037	*	MODULE 'LOCK AND READ ONLY' CORE VIRTUAL MEMORY INDICATORS.			*
		3038	*	THIS TABLE, WHICH IS EXPANDED TO (10+\$EXFTE-1) BYTES WHEN			*
		3039	*	\$EXFTR IS NON-ZERO, CONTAINS A SINGLE BYTE ENTRY CORRESPONDING			*
		3040	*	TO EACH CORE PAGE. BIT 6 (MASK X'02') IN EACH ENTRY INDICATES			*
		3041	*	THE MODIFICATION STATUS OF A CORE PAGE (1 = MODIFIED).			*
		3042	*	BIT 7 (MASK X'01') IN EACH ENTRY INDICATES THE LOCKED STATUS			*
		3043	*	OF A CORE PAGE (1 = LOCKED).			*

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00	31/05/21	PAGE 249
		3044	*	* PAGE REFERENCE TABLE - 256 BYTES, FOR THE PAGING MODULE CORE	*		
		3045	*	VIRTUAL MEMORY MAP. EACH BYTE IN THIS TABLE IS ASSOCIATED WITH	*		
		3046	*	A SPECIFIC VIRTUAL MEMORY PAGE, AND CONTAINS EITHER A VALUE OF	*		
		3047	*	ZERO OR THE NUMBER OF THE CORE PAGE CURRENTLY FILLED WITH THAT	*		
		3048	*	VIRTUAL MEMORY PAGE.	*		
		3049	*		*		
		3050	*	*OUTPUT	*		
		3051	*	* DISK VIRTUAL MEMORY - FOR ENTRY POINT FZZVPS ONLY, EACH CORE	*		
		3052	*	VIRTUAL MEMORY PAGE, FOR WHICH A 'PAGE MODIFY' BIT IS SET IS	*		
		3053	*	WRITTEN BACK TO DISK VIRTUAL MEMORY SO THAT DISK V.M. PAGES	*		
		3054	*	REFLECT THE CURRENT PROCESSING STATUS.	*		
		3055	*	* CORE VIRTUAL MEMORY - FOR ENTRY POINT FZZVPL ONLY, EACH CORE	*		
		3056	*	VIRTUAL MEMORY PAGE, FOR WHICH A 'PAGE LOCKED' BIT IS NOT SET,	*		
		3057	*	IS REPLACED WITH THE CORRESPONDING DISK VIRTUAL MEMORY PAGE	*		
		3058	*	SO THAT CORE V.M. PAGES REFLECT CURRENT DISK STATUS.	*		
		3059	*		*		
		3060	*	*EXTERNAL REFERENCES	*		
		3061	*	* \$DISKN - ENTRY POINT FOR THE SYSTEM PHYSICAL DISK IOCS.	*		
		3062	*	* \$WAITF - CORE ADDRESS OF 'WAIT' FUNCTION DISK PARAMETER LIST.	*		
		3063	*	* I\$RTRN - ENTRY POINT FOR PAGING MODULE V.M. RETURN CONTROL RTN.	*		
		3064	*	* \$EXFTR - 1 BYTE, FOR THE SYSTEM CORE EXTENSION FACTOR.	*		
		3065	*	* I\$CSXA - CORE ADDRESS OF 1ST BYTE IN CORE EXTENSION PAST 8K.	*		
		3066	*	* ISPLAT - CORE ADDRESS OF PAGE INDICATOR TABLE BASE ENTRY.	*		
		3067	*	* I\$PSTB - CORE ADDRESS OF PAGE REFERENCE TABLE BASE ENTRY.	*		
		3068	*		*		
		3069	*	*EXITS, NORMAL	*		
		3070	*	CONTROL IS ALWAYS PASSED TO THE PAGING ROUTINE AT ENTRY POINT.	*		
		3071	*	I\$RTRN (IPGRTN) FOR A RETURN TO THE CALLING PROGRAM.	*		
		3072	*		*		
		3073	*	*EXITS, ERROR	*		
		3074	*	N/A	*		
		3075	*		*		
		3076	*	*TABLES/WORK AREAS	*		
		3077	*	* DISK ADDRESS CONVERSION WORK AREAS - TWO 2-BYTE AREAS USED TO	*		
		3078	*	CONVERT LOGICAL DISK ADDRESSES TO PHYSICAL (A LA DL4ICS).	*		
		3079	*	* DISK PARAMETER LIST - 6 BYTES, FOR VIRTUAL PAGE READ/WRITE	*		
		3080	*	OPERATIONS.	*		
		3081	*		*		
		3082	*	*ATTRIBUTES	*		
		3083	*	* REUSABLE	*		
		3084	*	* NATURALLY RELOCATABLE	*		
		3085	*		*		
		3086	*	*CHARACTER CODE DEONENCY	*		
		3087	*	THE OPERATION OR THIS MODULE DOES NOT DEPEND UPON A PARTICULAR	*		
		3088	*	INTERNAL REPRESENTATION OF THE EXTERNAL CHARACTER SET.	*		
		3089	*		*		
		3090	*	*NOTES	*		
		3091	*	ERROR PROCEDURES	*		
		3092	*	NONE	*		
		3093	*		*		
		3094	*	REGISTER USAGE	*		
		3095	*	* REGISTER @BR IS TO CONTAIN THE CORE PAGE BASE ADDRESS	*		
		3096	*	ESTABLISHED THROUGH PAGING MODULE CONTROL FOR THE PAGE WHICH	*		
		3097	*	INCLUDES FZZVMP, AND IS RESTORED THROUGH THE PAGING MODULE.	*		
		3098	*	* REGISTER @XR IS NOT SAVED. IT IS USED IN FZZVMP FPR GENERAL	*		
		3099	*	PURPOSE INDEXING OPERATIONS.	*		

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	31/05/21	PAGE 250	
				3100	*						*
				3101	*	SAVED/RESTORED AREAS					*
				3102	*	NONE					*
				3103	*						*
				3104	*	MODIFICATION CONSIDERATIONS					*
				3105	*	NONE					*
				3106	*						*
				3107	*	REQUIRED MODULES					*
				3108	*	* @SYSEQ - COMMON SYSTEM EQUATES					*
				3109	*	* @FXDEQ - SYSTEM NUCLEUS ADDRESSES AND INDICATOR EQUATES.					*
				3110	*	* \$B@EQU - COMPILER PARAMETER AND CONSTANT EQUATES.					*
				3111	*	* \$I\$EQU - INTERPRETER FIXED LOCATION ADDRESS EQUATES.					*
				3112	*	* \$I@SEQ - INTERPRETER PARAMETER EQUATES (FOR STD PREC. ONLY)					*
				3113	*	* \$I@LEQ - INTERPRETER DARANETER EQUATES (FOR LNG PREC. ONLY)					*
				3114	*						*
				3115	*	OTHER					*
				3116	*	NONE					*
				3117	*	*****					*



ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	31/05/21	PAGE 251
					3119	*****				
					3120	* START OF VIRTUAL MEMORY PUSH/PULL EXECUTION ROUTINE				*
					3121	*****				
					3122	*				
					3123	* ESTABLISH VIRTUAL PAGE ADDRESSABILTY				
					3124	*				
					3125	*FZPGB VPAGE 0				
4C00					3126	ORG *,256,0	SET STARTING ADDRESS			
				4C00	3127	FZZPGB EQU *	START OF PROGRAM CODING			
4B01					3128	ORG *-255	RESET IAR TO PAGE			
4C00					3129	ORG *,256,0	WOMAN ADDRESS			
				4C00	3130	USING *,@BR	SET PAGE EASE ADDRESS			
4C00					3131	ORG FZZPGB	RESET STARTING ADDRESS			
					3132	*** END OF EXPANSION ***				
					3133	*				
					3134	*****				

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	31/05/21	PAGE 252
					3136	*				
					3137	*	ENTRY POINT FZZVPS - SET VIRTUAL PAGE PUSH FUNCTION.			
					3138	*				
				4C00	3139	FZZVPS EQU *	VM PUSH ROUTINE ENTRY POINT			
4C00	7C	02	BD		3140	MVI FZZDPL+@DCTRL(,@BR),@DPUT	SET DISK OUTPUT PARAMETER			
4C03	F2	87	03		3141	J FZZ005	GO PERFORM THE PUCH OPERATION			
					3142	*				
					3143	*	ENTRY POINT FZZVPL - SET VIRTUAL PAGE PULL FUNCTION.			
					3144	*				
				4C06	3145	FZZVPL EQU *	VM PULLH ROUTINE ENTRY POINT			
4C06	7C	01	BD		3146	MVI FZZDPL+@DCTRL(,@BR),@DGET	SET DISK OUTPUT PARAMETER			
					3148	*				
					3149	*	INITIALIZE PUSH/PULL ROUTINE FOR 8K SYSTEM ENVIRONMENT.			
					3150	*				
4C09	7C	0A	2B		3151	FZZ005 MVI FZZ020+@D1(,@BR),I@NCPG	SET MAX CORE PAGE COUNT FOR 8K			
4C0C	5C	01	BA B5		3152	MVC FZZHCA(,@BR),FZZSXA(@CADDR,@BR)	SET HIGH CORE ADDR FOR 8K			
					3153	*				
					3154	*	TEST FOR CORE AVAILABILITY BEYOND 8K - RE-INITIALIZE IF EXTENDED CORE			
					3155	*				
4C10	3D	00	043B		3156	CLI \$EXFTR,@ZERO	TEST FOR NULL CORE EXTENSION			
4C14	F2	81	0E		3157	JE FZZ010	BRANCH IF ONLY 8K SYSTEM CONFIG.			
					3158	*				
4C17	4E	00	2B 043B		3159	ALC FZZ020+@D1(,@BR),\$EXFTR(1)	ADD 1 LESS THAN EXTRA NO. OF			
4C1C	5F	00	2B B3		3160	SLC FZZ020+@D1(,@BR),FZZBN1(1,@BR)	* PAGES TO CORE PAGE COUNT			
4C20	4E	00	B9 043B		3161	ALC FZZHCA-1(,@BR),\$EXFTR(1)	SET EXTENDED SYSTEM HIGH CADDR			
					3162	*				
					3163	*****				

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER	15,	MOD	00	31/05/21	PAGE	253
					3165	*								
					3166	*	ACCESS A CORE PAGE ENTRY IN THE PAGING MODULE 'LOCK AND READ ONLY'							
					3167	*	INDICATOR TABLE							
					3168	*								
4C25	C2	02	15E1		3169	FZZ010	LA I\$PLRT-1,@XR						LOAD CORE PAGE INDR TABLE BASE	
4C29	E2	02	00		3170	FZZ020	LA *-*(,@XR),@XR						INCR POINTER TO CORE PAGE ENTRY	
					3171	*								
					3172	*	TEST FOR PUSH OR PULL FUNCTION EXECUTION							
					3173	*								
4C2C	7D	01	BD		3174		CLI FZZDPL+@DCTRL(@BR),@DGET IF DISK PARAM SET FOR INPUT							
4C2F	F2	81	0C		3175		JE FZZ025						* BRANCH TO EXECUTE PAGE PULL	
					3176	*								
					3177	*	PUSH FUNCTION - TEST THE CURRENTLY REFERENCED CORE PAGE INDICATOR							
					3178	*	FOR MODIFY BIT SET ON, AND PUSH THE CORE PAGE ONLY IF MODIFIED							
					3179	*								
4C32	B8	02	00		3180		TBN FZZLRT(@XR),FZZMDY						IF CORE PAGE IS NOT MODIFIED	
4C35	F2	90	6A		3181		JF FZZ090						* GO DECREMENT CORE PAGE COUNT	
4C38	BB	02	00		3182		SBF FZZLRT(@XR),FZZMDY						PAGE MODIFIED - SET INDICATOR	
4C3B	F2	87	06		3183		J FZZ030						* OFF AND GO PERFORM PAGE PUSH	
					3184	*								
					3185	*	PULL FUNCTION - TEST THE CURRENTLY REFERENCED CORE PAGE INDICATOR							
					3186	*	FOR LOCK BIT SET ON, AND PULL THE CORE PAGE ONLY IF NOT LOCKED							
					3187	*								
4C3E	B8	01	00		3188	FZZ025	TBN FZZLRT(@XR),FZZLOK						IF THE CORE PAGE IS LOCKED	
4C41	F2	10	5E		3189		JT FZZ090						* GO DECREMENT CORE PAGE COUNT	
					3190	*								
					3191	*	PUSH OR PULL CURRENTLY REFERENCED CORE PAGE - SEARCH THE PAGE							
					3192	*	REFERENCE TABLE TO DETERMINE THE ACTUAL VIRTUAL PAGE NUMBER							
					3193	*								
4C44	7C	FF	51		3194	FZZ030	MVI FZZ040+@D1(@BR),FZZBM1						SET VIRTUAL PAGE NO. = MINUS 1	
4C47	C2	02	14CA		3195		LA I\$PGTB,@XR						LOAD PAGE REFERENCE TABLE BASE	
4C4B	5E	00	51 B3		3196	FZZ035	ALC FZZ040+@D1(@BR),FZZBN1(1,@BR)						INCREMENT VIRTUAL PAGE NO.	
4C4F	5D	00	00 2B		3197	FZZ040	CLC *-*(,@BR),FZZ020+@D1(1,@BR)						COMPARE REF TBL ENTRY W/ CORE	
4C53	D0	01	4B		3198		BNE FZZ035(@BR)						* PAGE NO. AND LOOP IF NO MATCH	
					3199	*								
					3200	*	*****							

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER	MOD	DATE	PAGE	
					3202		*****		15	00	31/05/21	254
					3203	*	CONVERT VIRTUAL PAGE NUMBER TO A PHYSICAL DISK ADDRESS					
					3204		*****					
					3205	*						
					3206	*	ESTABLISH LOGICAL DISK ADDRESS IN THE DISK PARAMETER LIST					
					3207	*						
	4C56	7C	07	BE	3208	MVI	FZZDPL+@DCYL(,@BR),B@DVCY SET VIRTUAL MEMORY BASE CYL NO.					
	4C59	5C	00	BF 51	3209	MVC	FZZDPL+@DSAD(,@BR),FZZ040+@D1(1,@BR) SET RELATIVE SECTOR					
					3210	*	* ADDRESS EQUAL VIRT PAGE NO.					
					3211	*						
					3212	*	DETERMINE THE TRACK SECTOR COUNT (= LOGICAL SECTOR ADDRESS, MOD 24).					
					3213	*	INCREMENT THE CYLINDER/DISK/TRACK INDICATOR DURING EACH PASS THROUGH					
					3214	*	THE SUBTRACTION (DIVISION) LOOP.					
					3215	*						
	4C5D	5C	01	BC B8	3216	MVC	FZZCNT(,@BR),FZZCDT(@DADDR,@BR) INITLZ CYL/DISK/TRACK CNT					
	4C61	5F	01	BC B8	3217	FZZ050 SLC	FZZCNT(,@BR),FZZCDT(@DADDR,@BR) INCR CYL/DISK/TRACK COUNT					
	4C65	5F	00	BF B6	3218	SLC	FZZDPL+@DSAD(,@BR),FZZNST(1,@BR) DECR LOGICAL SECTOR ADDR					
	4C69	D0	02	61	3219	BNM	FZZ050(,@BR) REPEAT UNTIL SADDR IS NEGATIVE					
	4C6C	5E	00	BF B6	3220	ALC	FZZDPL+@DSAD(,@BR),FZZNST(1,@BR) RESTORE POSITIVE SADDR					
					3221	*						
					3222	*	THE DISK PARAMETER LIST NOW CONTAINS THE PHYSICAL SECTOR COUNT -					
					3223	*	THE CYLINDER CORRECTION COUNT CONTAINS THE INCREMENT WITH WHICH TO					
					3224	*	ADJUST THE LOGICAL CYLINDER ADDRESS, AND BITS 0 AND 1 OF THE DISK/					
					3225	*	TRACK INDICATOR BYTE ARE SET RESPECTIVELY TO THE CORRECT PHYSICAL					
					3226	*	DISK AND TRACK STATUS CONDITIONS.					
					3227	*						
					3228	*	CONVERT THE LOGICAL (BASE) CYLINDER ADDRESS TO A PHYSICAL ADDRESS					
					3229	*						
	4C70	5E	00	BE BB	3230	ALC	FZZDPL+@DCYL(,@BR),FZZCNT-1(1,@BR) ADJUST THE CYL ADDR					
					3231	*						
					3232	*	SHIFT SECTOR COUNT 2 BITS LEFT (MULTIPLY BY 4)					
					3233	*						
	4C74	5E	00	BF BF	3234	ALC	FZZDPL+@DSAD(,@BR),FZZDPL+@DSAD(1,@BR) SHIFT COUNT (2X)					
	4C78	5E	00	BF BF	3235	ALC	FZZDPL+@DSAD(,@BR),FZZDPL+@DSAD(1,@BR) SHIFT COUNT (4X)					
					3236	*						
					3237	*	SET THE SECTOR ADDRESS DISK (REMOVABLE OR FIXED) INDICATOR BIT					
					3238	*						
	4C7C	78	80	BC	3239	TBN	FZZCNT(,@BR),FZZIDM TEST INDICATOR DISK BIT					
	4C7F	F2	90	03	3240	JF	FZZ060 * AND BRANCH IF NOT EQUAL 1					
	4C82	7A	01	BF	3241	SBN	FZZDPL+@DSAD(,@BR),FZZSDM SET SADDR FOR FIXED DISK					
					3242	*						
					3243	*	SET THE SECTOR ADDRESS TRACK (UPPER OR LOWER) INDICATOR BIT					
					3244	*						
	4C85	78	40	BC	3245	FZZ060 TBN	FZZCNT(,@BR),FZZITM TEST INDICATOR TRACK BIT					
	4C88	F2	90	03	3246	JF	FZZ070 * AND BRANCH IF NOT EQUAL 1					
	4C8B	7A	80	BF	3247	SBN	FZZDPL+@DSAD(,@BR),FZZSTM SET SADDR FOR LOWER TRACK					
					3248	*						
					3249		*****					

FZZVMP - S/3 BASIC INTERPRETER V.M. PUSH/PULL EXEC RTN									
ERR	LOC	OBJECT CODE		ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00 31/05/21 PAGE 255	
					3251	*****			
					3252	* PERFORM READ/WRITE BETWEEN CORE PAGE AND DISK VIRTUAL MEMORY			*
					3253	*****			
					3254	*			
					3255	* CALCULATE THE AFFECTED CORE PAGE ACTUAL CORE ADDRESS			
					3256	*			
4C8E	5C	01	C2	BA	3257	FZZ070	MVC	FZZDPL+@DBFR2(,@BR),FZZHCA(@CADDR,@BR)	SET HIGH CORE ADDR
4C92	5F	00	C1	2B	3258		SLC	FZZDPL+@DBFR1(,@BR),FZZ020+@D1(1,@BR)	SUB CORE PAGE NO.
					3259	*			
					3260	* PERFORM THE CORE PAGE - VIRTUAL MEMORY DISK OPERATION			
					3261	*			
4C96	D2	02	BD		3262		LA	FZZDPL(,@BR),@XR	LOAD PARAMETER LIST CORE ADDR
4C99	74	02	A1		3263		ST	FZZ080(,@BR),@XR	STORE DPL CORE ADOR FOR CALL
4C9C	C0	87	0025		3264		B	\$DISKN	LINK TO READ/WRITE THE CORE PAGE
4CA0				4CA1	3265	FZZ080	DS	CL(@CADDR)	PARAMETER LIST CORE ADDRESS
					3267	*			
					3268	* SET NEXT CORE PAGE PROCESSING - EXIT IF NO MORE CORE PAGES			
					3269	*			
4CA2	5F	00	2B	B3	3270	FZZ090	SLC	FZZ020+@D1(,@BR),FZZBN1(1,@BR)	DECR THE CORE PAGE NUMBER
4CA6	D0	84	25		3271		BP	FZZ010(,@BR)	GO PROCESS NEW PAGE UNLESS ZERO
					3272	*			
					3273	* EXIT - RETURN TO THE CALLING ROUTINE			
					3274	*			
4CA9	C0	87	0025		3275		B	\$DISKN	LINK TO WAIT I/O COMPLETED
4CAD	057F			4CAE	3276		DC	AL(@CADDR)(\$WAITF)	'WAIT' FUNCTION PARAM CADDR
					3277	*			
4CAF	C0	87	12D3		3278		B	I\$RTRN	RETURN TO CALLING ROUTINE
					3279	*			
					3280	*****			

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15,	MOD 00	31/05/21	PAGE 255
					3251	*****	*****				
					3252	*	PERFORM READ/WRITE BETWEEN CORE PAGE AND DISK VIRTUAL MEMORY				*
					3253	*****	*****				
					3254	*					
					3255	*	CALCULATE THE AFFECTED CORE PAGE ACTUAL CORE ADDRESS				
					3256	*					
4C8E	5C	01	C2	BA	3257	FZZ070	MVC FZZDPL+@DBFR2(,@BR),FZZHCA(@CADDR,@BR)	SET	HIGH	CORE	ADDR
4C92	5F	00	C1	2B	3258		SLC FZZDPL+@DBFR1(,@BR),FZZ020+@D1(1,@BR)	SUB	CORE	PAGE	NO.
					3259	*					
					3260	*	PERFORM THE CORE PAGE - VIRTUAL MEMORY DISK OPERATION				
					3261	*					
4C96	D2	02	BD		3262		LA FZZDPL(,@BR),@XR	LOAD	PARAMETER	LIST	CORE
4C99	74	02	A1		3263		ST FZZ080(,@BR),@XR	STORE	DPL	CORE	ADOR
4C9C	C0	87	0025		3264		B \$DISKN	LINK	TO	READ/WRITE	THE
4CA0				4CA1	3265	FZZ080	DS CL(@CADDR)	PARAMETER	LIST	CORE	ADDRESS
					3267	*					
					3268	*	SET NEXT CORE PAGE PROCESSING - EXIT IF NO MORE CORE PAGES				
					3269	*					
4CA2	5F	00	2B	B3	3270	FZZ090	SLC FZZ020+@D1(,@BR),FZZBN1(1,@BR)	DECR	THE	CORE	PAGE
4CA6	D0	84	25		3271		BP FZZ010(,@BR)	GO	PROCESS	NEW	PAGE
					3272	*					
					3273	*	EXIT - RETURN TO THE CALLING ROUTINE				
					3274	*					
4CA9	C0	87	0025		3275		B \$DISKN	LINK	TO	WAIT	I/O
4CAD	057F			4CAE	3276		DC AL(@CADDR)(\$WAITF)	'WAIT'	FUNCTION	PARAM	CADDR
					3277	*					
4CAF	C0	87	12D3		3278		B I\$RTRN	RETURN	TO	CALLING	ROUTINE
					3279	*					
					3280	*****	*****				

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00 31/05/21 PAGE 256
					3282	*****		
					3283	* VIRTUAL MEMORY PUSH/PULL ROUTINE CONSTANTS		*
					3284	*****		
					3285	*		
4CB3	01			4CB3	3286	FZZBN1 DC	IL1 '1'	BINARY INTEGER +1
					3287	*		
4CB4	2000			4CB5	3288	FZZSXA DC	AL (@CADDR) (I\$CSXA)	CORE EXTENSION STARTING ADDRESS
					3289	*		
4CB6	18			4CB6	3290	FZZNST DC	AL1 (@DTRSZ)	NO. OF SECTORS PER DISK TRACK
4CB7	FFC0			4CB8	3291	FZZCDT DC	XL (@DADDR) 'FFC0'	CYLINDER/DISK/TRACK DECREMENT
					3293	*****		
					3294	* VIRTUAL MEMORY PUSH/PULL ROUTINE WORK AREAS		*
					3295	*****		
					3296	*		
4CB9				4CBA	3297	FZZHCA DS	CL (@CADDR)	HIGHEST AVAILABLE CADDR + 1
					3298	*		
4CBB				4CBC	3299	FZZCNT DS	CL (@DADDR)	CYLINDER/DISK/TRACK COUNTER
					3300	*		
					3301	*FZZDPL DPL	CNT-1	VM I/O DISK PARAMETER LIST
				4CBD	3302	FZZDPL EQU	*	DISK PARAMETER LIST
4CBD	00			4CBD	3303		DC AL1 (*-*)	REQUESTED FUNCTION
4CBE	00			4CBE	3304		DC AL1 (*-*)	CYLINDER ADDRESS
4CBF	00			4CBF	3305		DC AL1 (*-*)	HEAD/SECTOR/DRIVE/DISK SPEC
4CC0	01			4CC0	3306		DC AL1 (1)	SECTOR COUNT
4CC1	0000			4CC2	3307		DC AL2 (*-*)	BUFFER ADDRESS
					3308	***	END OF EXPANSION ***	
					3310	*****		
					3311	* VIRTUAL MEMORY PUSH/PULL ROUTINE EQUATES REFERENCING CONSTANTS		*
					3312	*****		
					3313	*		
				00FF	3314	FZZBM1 EQU	X 'FF'	BINARY INTEGER -1
					3315	*		
				0000	3316	FZZLRT EQU	0	DISP FOR PAGE INDR TABLE ENTRY
				0001	3317	FZZLOK EQU	X '01'	CORE PAGE INDICATOR LOCK MASK
				0002	3318	FZZMDY EQU	X '02'	CORE PAGE INDICATOR MODIFY MASK
					3319	*		
				0080	3320	FZZIDM EQU	X '80'	INDICATOR DISK BIT MASK
				0040	3321	FZZITM EQU	X '40'	INDICATOR TRACE BIT MASK
				0001	3322	FZZSDM EQU	X '01'	SECTOR ADDR DISK BIT MASK
				0080	3323	FZZSTM EQU	X '80'	SECTOR ADDR TRACK BIT MASK
					3324	*		
					3325	*	END OF VIRTUAL MEMORY PUSH/PULL ROUTINE CODING	*****
					3326	*		

## DLFPRT - LINE PRINTER ROUTINE

ERR	LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00	31/05/21	PAGE 257
	3328				*****			
	3329	*			5703-XM1 COPYRIGHT IBM CORP. 1970			*
	3330	*			REFER TO INSTRUCTIONS ON COPY RIGHT NOTICE, 120-2083			*
	3331	*						*
	3332				*****			*
	3333	*			*STATUS			*
	3334	*			VERSION 1 MODIFICATION 0			*
	3335	*						*
	3336	*			*FUNCTION			*
	3337	*			* DLFprt EXECUTION CAUSES DATA OUTPUT AND/OR CARRIER POSITIONING			*
	3338	*			ON THE SYSTEM PRINT DEVICE UNDER CONTROL OF CODES RECEIVED FROM			*
	3339	*			THE CALLING ROUTINE, PRINTING IS DONE BIDIRECTIONALLY			*
	3340	*			* THE FOLLOWING ACTIONS ARE PERFORMED DEPENDING ON THE CODE AND			*
	3341	*			CARRIER POSITION:			*
	3342	*			* INDEX, PRINT AND INDEX & TAB, PRINT AND INDEX			*
	3343	*			* INPUT CODES			*
	3344	*			* PRINT X'40' WILL CAUSE THE DATA TO BE PRINTED TO			*
	3345	*			BE MOVED INTO THE LINE PRINTER BUFFER			*
	3346	*			* PRINT & RETRN X'C0' WILL CAUSE THE DATA TO BE MOVED INTO			*
	3347	*			THE BUFFER, AND THE CONTENTS PRINTED			*
	3348	*			* CARRAGE RETRN X'80' WILL CAUSE AN INDEX IF THE BUFFER IS			*
	3349	*			EMPTY OR THE BUFFER PRINTED IF NOT			*
	3350	*						*
	3351	*			*ENTRY POINTS			*
	3352	*			THIS ROUTINE HAS A SINGLE CALLING ENTRY POINT - DLFprt - WHOSE			*
	3353	*			FUNCTION IS DEFINED ABOVE. THE CALLING SEQUENCE IS:			*
	3354	*			B I\$LDXR			*
	3355	*			DC AL2(V\$LPRT)			*
	3356	*			WHERE THE ADDRESS CONSTANT PARAMETER DEFINES THE VIRTUAL ADDRESS			*
	3357	*			OF ENTRY POINT DLFprt.			*
	3358	*						*
	3359	*			*INPUT			*
	3360	*			* \$PRPOS - 1 BYTE CARRIER POSITION RELATIVE TO HARDWARE LEFTMGN			*
	3361	*			* \$LMRGN - 1 BYTE SOFTWARE LEFT MARGIN INDICATOR			*
	3362	*						*
	3363	*			*OUTPUT			*
	3364	*			* PRINTED OUTPUT AND CARRIER POSITIONING			*
	3365	*			* \$PRPOS - 1 BYTE 'DUMMY' CARRIER POSITION INDICATING WHERE THE			*
	3366	*			CARRIER SHOULD BE. SET EQUAL TO \$LMRGN AFTER PRINTING.			*
	3367	*			* \$BUFPT - 1 BYTE POINTS AT NEXT AVAIL BYTE IN LINE PRINT BUFFER			*
	3368	*			* \$LPRP3 - 1 BYTE LINE PRINTER INDICATORS			*
	3369	*			* 3LPRI0 - 2 BYTES ONE FOR BUFFER INCREMENT ONE FOR PDAR DISP.			*
	3370	*						*
	3371	*			*EXTERNAL REFERENCES			*
	3372	*			* V\$LPRT2 - VIRTUAL ENTRY SECOND PAGE OF LINE PRINTER ROUTINE			*
	3373	*			* V\$LPRTB - VIRTUAL ADDRESS OF THE LINE PRINTER BUFFER			*
	3374	*			* I\$LDXR - ENTRY POINT FOR PAGING MODULE V.M. LOAD XR ROUTINE			*
	3375	*			* \$LPRI0 - ENTRY POINT FOR PAGING MODULE V.M. CONVERT ADDRESS			*
	3376	*						*
	3377	*			*EXITS, NORMAL			*
	3378	*			EXIT IS TO THE CALLING ROUTINE VIA A BRANCH TO THE V.M. PAGING			*
	3379	*			ROUTINE.			*
	3380	*						*
	3381	*			*EXITS, ERROR			*
	3382	*			NONE			*
	3383	*						*



## DLFPRT - LINE PRINTER ROUTINE

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00	31/05/21	PAGE 258
		3384	*	TABLES/WORKAREAS			*
		3385	*	N/A			*
		3386	*				*
		3387	*	ATTRIBLTES			*
		3388	*	NATURALLY RELOCATABLE AND REUSABLE			*
		3389	*				*
		3390	*	CHARACTLR CODE DEPENDENCY			*
		3391	*	THE OPERATION OF THIS MODULE DEPENDS UPON AN INTERNAL REPRESENTATION OF THE EXTERNAL CHARACTER SET WHICH IS EQUIVALENT TO THE			*
		3392	*	ONE USED AT ASSEMBLY TIME.			*
		3393	*				*
		3394	*				*
		3395	*	NOTES			*
		3396	*	ERROR PROCEDLRES			*
		3397	*	IF A PRINTER UNIT CHECK OCCURES. THE LINE IN WHICH THE CHECK			*
		3398	*	OCCURED WILL BE REPRINTED			*
		3399	*				*
		3400	*	REGISTER USAGE			*
		3401	*	REGISTER 1 (@BR) IS USED AS A BASE REGISTER FOR DFPRNT			*
		3402	*	REGISTER 2 (@XR) IS USED AS A BASE REGISTER FOR: THE FIRST			*
		3403	*	PAGE OF DLFPRT, LINE PRINTER BUFFER, OR IN THE CASE OF A UNIT			*
		3404	*	CHECK, THE PRINTER ERROR HANDELING ROUTINE 'DFPNDX'.			*
		3405	*				*
		3406	*	SAVED/RESTORED AREAS			*
		3407	*	NONE			*
		3408	*				*
		3409	*	MODIFICATION CONSIDERATIONS			*
		3410	*	CHANGES TO EITHER DLFPRT OR DFPRNT MAY DIRECTLY AFFECT THE			*
		3411	*	INTERFACE BETWEEN THE TWO MODULES.			*
		3412	*				*
		3413	*	REQUIRED MODULES			*
		3414	*	@SYSEQ			*
		3415	*	@FXDEQ			*
		3416	*	@HDWEQ			*
		3417	*	\$V\$EQU			*
		3418	*	\$I\$EQU			*
		3419	*	DFPRNT			*
		3420	*				*
		3421	*	OTHER			*
		3422	*	NONE			*
		3423	*	*****			*

## DLFPRT - LINE PRINTER ROUTINE

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	31/05/21	PAGE 259
					3425	*****				
	4D00				3426	ORG	*,256,0			SET STARTING ADDRESS
				2800	3427	USING	DFPASE,@BR			SET PAGE BASE ADDRESS - DFPRNT
				4D00	3428	USING	DLFPRT,@XR			SET PAGE BASE ADDRESS
					3429	*				
				4D00	3430	DLFPRT EQU	*			ENTRY BIDIR PRINT
	4D00	7C	87	BC	3431	MVI	DFP330+@Q(,@BR),@UCB			SET BRANCH TO LINE PRINTER PAGE
	4D03	B4	02	66	3432	ST	DLF155+@OP1(,@XR),@XR			SAVE XR
	4D06	3A	40	03E4	3433	SBN	\$LPRP3,@PRINT			SET LINE PRINTER FLAG
	4D0A	2C	01	144A D7	3434	MVC	I\$VADR,DLFVD1(@VADDR,@XR)			GET PRINTER BUFFER VADDR
	4D0F	C0	87	1349	3435	B	I\$MDFY			LOAD BUFFER & SET PAGE MDFY BIT
	4D13	8C	01	D9 144C	3436	MVC	BUFADR(2,@XR),I\$CADR			SAVE BUFFER ADDR
				4D18	3437	DLF050 EQU	*			PROCESS PRINTER UNIT CHECK
	4D18	7C	25	BD	3438	MVI	DFP330+@D1(,@BR),DENTRY			SET ENTRY DISPLACEMENT
	4D1B	BC	87	A9	3439	MVI	DLF360+@Q(,@XR),@UCB			FORCE RETURN ENTRY
	4D1E	6C	02	BA F6	3440	MVC	DFP333(3,@BR),DLFEOR(,@XR)			SET DLFPRT ERROR ENTRY
					3441	*				
	4D22	D0	87	A2	3442	B	DFP280(,@BR)			GO CHECK FOR PREV. ERROR
					3444	*****				
					3445	*				
					3446	*	FIND FUNCTION			
					3447	*				
					3448	*****				
				4D25	3449	DLF100 EQU	*			RETURN FROM ERROR CHECK
	4D25	BC	80	A9	3450	MVI	DLF360+@Q(,@XR),@NOP			RESET ENTRY INDICATOR
	4D28	78	40	F5	3451	TBN	DLFIST+@PCTRL(,@BR),@PRINT			IS OP A PRINT ?
	4D2B	F2	90	4A	3452	JF	DLF170			CHECK IF BUFFER FULL
					3453	*****				
					3454	*				
					3455	*	ENTRY TO FILL BUFFER			
					3456	*				
					3457	*****				
	4D2E	39	01	03E4	3458	TBF	\$LPRP3,@INDEX			TEST DUMMY PRINT
	4D32	F2	90	0A	3459	JF	DLF140			SKIP IF IN USE
	4D35	3A	01	03E4	3460	SBN	\$LPRP3,@INDEX			SET DUMMY PRINT POS. USED
	4D39	0C	00	03E5 03C2	3461	MVC	\$LPROS(1),\$PRPOS			SAVE TRUE POSITION
				4D3F	3462	DLF140 EQU	*			UPDATE BUFFER POINTER
					3463	*				
					3464	*****				
					3465	*				
	4D3F	1E	00	03E3 F6	3466	ALC	\$BUFPT,DLFIST+@PRCNT(1,@BR)			ADD NEXT COUNT TO BUFFER PTR
	4D44	1E	00	03C2 F6	3467	ALC	\$PRPOS(1),DLFIST+@PRCNT(,@BR)			UPDATE HEAD POSITION
					3468	*				
					3469	*	INCREMENT BUFFER POINTER			
					3470	*				
	4D49	2C	01	144A ED	3471	MVC	I\$VADR,DLFPCH(@VADDR,@XR)			V.M. PATCH PAGE ENTRY ADDR 1-5
	4D4E	C0	87	1358	3472	DLF143 B	I\$CVAD			LOAD PATCH PAGE 1-5
	4D52	8C	01	5A 144C	3473	MVC	DLF145+@OP1(@CADDR,@XR),I\$CADR			MOVE CADDR TO BRANCH 1-5
	4D57	C0	87	0000	3474	DLF145 B	*-*			1-5
					3475	*				
					3476	*	MOVE DATA TO BUFFER			
					3477	*				
	4D5B	B5	02	D9	3478	DLF146 L	BUFADR(,@XR),@XR			XR - BUFFER CADDR
	4D5E	8C	00	00 0000	3479	DLF150 MVC	*-*(@VQ,@XR),*-*			MOVE DATA INTO BUFFER
					3480	*				

## DLFPRT - LINE PRINTER ROUTINE

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00 31/05/21 PAGE 260
	4D63	C2	02	0000	3481	DLF155	LA *-*,@XR	RESTORE DLFPRT BASE ADDR
					3482	*		
					3483	*	TEST FOR CARRAGE RETURN	
					3484	*		
	4D67	7D	C0	F5	3485		CLI DLFIST+@PCTRL(,@BR),@PRETR	TEST CARRAGE RETURN ON
	4D6A	F2	01	4C	3486		JNE DLF175	JUMP TO RETURN IF NO C.R.
				4D6D	3487	DLF160	EQU *	LOAD PAGE2 LINE PRINTER
	4D6D	7C	88	BD	3488		MVI DFP330+@D1(,@BR),DERROR	SET ERROR ENTRY DISP.
	4D70	2C	01	144A EB	3489	DLF165	MVC I\$VADR,DLFVD2(@VADDR,@XR)	VADDR VLPRT2
	4D75	E0	87	93	3490		B DLF400(,@XR)	LOAD BASE
				4D78	3492	DLF170	EQU *	CHECK IF BUFFER EMPTY
	4D78	3D	00	03E3	3493		CLI \$BUFPT,@ZERO	IS BUFFER EMPTY ?
	4D7C	E0	01	6D	3494		BNE DLF160(,@XR)	GO TO PRINT EXIT
	4D7F	7C	01	DE	3495		MVI DLFPCF(,@BR),@INDEX	SET INDEX ONLY
	4D82	7C	87	A0	3496		MVI DFP270+@Q(,@BR),@UCB	FORCE RETURN
	4D85	D0	87	92	3497		B DFP240(,@BR)	GO DO I/O
					3499	*		
					3500	*	NO ERROR, CHECK FOR PREVIOUS ERROR	
					3501	*		
	4D88	F2	00	1D	3502	DLF350	JC DLF360,*-*	JUMP NO PREVIOUS ERROR
	4D89				3503		ORG DLF350+@Q	* INITIALIZE
	4D89	87			3504		DC AL1(@UCB)	* TO INDICATE
	4D8B				3505		ORG DLF350+@INST3	* NO PREVIOUS PRINTER ERROR
	4D8B	BC	87	89	3506		MVI DLF350+@Q(,@XR),@UCB	RESET ERROR INDICATOR
	4D8E	2C	01	144A E3	3507	DLF355	MVC I\$VADR,DLFRTY(@VADDR,@XR)	VADDR RETRY ENTRY VLPRT2
				4D93	3508	DLF400	EQU *	PREPARE TO EXIT LINE PTR PAGE1
	4D93	3C	80	12B6	3509		MVI I\$LBFR,@NOP	FORCE LINE PRINTER UNLOCK
	4D97	C0	87	1358	3510		B I\$CVAD	LOAD LINE PRINTER PAGE2
	4D9B	8C	01	A7 144C	3511		MVC DLF425+@OP1(@CADDR,@XR),I\$CADR	MOVE CADDR TO BR
	4DA0	C0	87	1354	3512		B I\$LOCK	LOCK PAGE VLPRT2 1-5
	4DA4	C0	87	0000	3513	DLF425	B *-*	BRANCH TO PAGE2
	4DA8	E0	00	25	3515	DLF360	BC DLF100(,@XR),*-*	FORMAT NEXT LINE / GO TO ENTRY
	4DA9				3516		ORG DLF360+@Q	* INITIALIZE
	4DA9	80			3517		DC AL1(@NOP)	* TO FORMAT
	4DAB				3518		ORG DLF360+@INST3	* NEXT LINE TO BE PRINTED
	4DAB	2C	01	144A EF	3519		MVC I\$VADR,DLFPC1(@VADDR,@XR)	V.M. PATCH PAGE ENTRY ADDR 1-5
	4DB0	E0	87	4E	3520	DLF375	B DLF143(,@XR)	BRANCH TO MV CADDR TO BRANCH 1-5
					3522	*****		
					3523	***** RETURN TO CALLER *****		
					3524	*****		
				4DB3	3525	RETURN	EQU *	LINE PRINTER RETURN AREA
	4DB3	0C	00	03C2 03C1	3526		MVC \$PRPOS(1),\$LMRGN	SET DUMMY POSITION LEFT MGN
				4DB9	3527	DLF175	EQU *	RETURN FROM DLFPRT
	4DB9	7C	80	BC	3528		MVI DFP330+@Q(,@BR),@NOP	RESET BRANCH TO LINR PRINTER
	4DBC	7C	80	A0	3529		MVI DFP270+@Q(,@BR),@NOP	RESET DFPRNT EXIT
	4DBF	6C	02	BA F3	3530		MVC DFP333(3,@BR),DFPEOR(,@XR)	RESTORE DFPRNT ERROR TEST
	4DC3	7C	11	E0	3531		MVI DLFPCF+2(,@BR),@TBLIX	RESTORE MATRIX PRINTER END
	4DC6	3B	40	03E4	3532		SBF \$LPRP3,@PRINT	RESET LINE PRINTER FLAG
	4DCA	D0	87	CA	3533		B DFP300(,@BR)	RETURN TO CALLER
					3534	*		
					3535	*****		
	4DCD				3536	DLFRPE	EQU *	PRINTER UNIT CHECK ENTRY

## DLFPRT - LINE PRINTER ROUTINE

ERR LOC OBJECT CODE ADDR STMT SOURCE STATEMENT VER 15, MOD 00 31/05/21 PAGE 261

4DCD	C0 87 1330		3537	B	I\$LDXR	BR TO FORCE DLFPRT TO BE MOST	
			3538	*		* RECENTLY USED PAGE	
4DD1	4D00	4DD2	3539	DC	AL2(V\$LPRT)	DLFPRT VADDR	
4DD3	D0 87 D3		3540	B	DFPRPE-DFPRNT(,@BR)	GO PROCESS LOAD ERP SECTION	
			3541	*			
			3542	*	*****		
4DD6	4F00	4DD7	3543	DLFVD1 DC	AL(@VADDR)(V\$LPRB)	LINE PRINTER BUFFER PAGE	
4DD8	0000	4DD9	3544	BUFADR DC	XL2'00'	SAVED BUFFER ADDR	
			3545	*			
4DDA	0000	4ddb	3546	DFPWTB DC	XL2'00'	LINE WIDTH	
4DDC	00	4DDC	3547	DFPRES DC	XL1'00'	LINE COUNT	
4DDD	0000	4DDE	3548	BUFRWK DC	XL2'00'	BUFFER POINTER	
4DDF	00	4DDF	3549	DLFBPT DC	XL1'00'	BUFFER INCREMENT	
			3550	*			
4DE0	0025	4DE1	3551	DLFMAR DC	AL2(DLF500-VLPRT2)	DISPLACENENT TO FORMAT LINE	
4DE2	4E49	4DE3	3552	DLFRTY DC	AL2(V\$LPRT2+DLF700-VLPRT2)	RETRY ENTRY POINT	
			3553	*			
4DE4	00	4DE4	3554	DFPPOS DC	XL1'00'	CHARACTER POSITION ON LINE	
4DE5	8080C00001	4DE9	3555	LPRCMD DC	XL5'8080C00001'	LINE PRINTER CMDS.	
4DEA	4E00	4DEB	3556	DLFVD2 DC	AL2(V\$LPRT2)	LINE PRINTER PAGE2	
		004E	3557	DLFX4E EQU	X'4E'	VLPRT2 LOCK BIT	1-5
		0053	3558	DLFX53 EQU	X'53'	VLPRT3 LOCK BIT	1-5
		0090	3559	DLTABL EQU	X'90'	TAB LEFT AND CHAIN	
4DEC	5391	4DED	3560	DLFPCH DC	AL2(V\$PCH2+DLF400-@D1-DLFPRT)	PATCH PAGE ENTRY ADDR	1-5
4DEE	53B6	4DEF	3561	DLFPC1 DC	AL2(V\$PCH2+DLF175-@DD2-DLFPRT)	PATCH PAGE ENTRY ADDR	1-5
4DF0	00	4DF0	3562	DLFSWC DC	XL1'00'	RETURN CARRIAGE SWITCH	1-5
		00A0	3563	DLTABR EQU	X'A0'	TAB RIGHT AND CHAIN	
		0088	3564	DERROR EQU	DLF350-DLFPRT	ERROR CHECK ENTRY DISP.	
		0025	3565	DENTRY EQU	DLF100-DLFPRT	ENTRY RETURN DISP.	
		0001	3566	DLFRTN EQU	X'01'	RETURN CARRIAGE INDICATOR	1-5
			3567	*			
			3568	*	INSTRUCTION MODIFICATION TP DFPRNT AT DFP335		
			3569	*			
4DF1	D1 E0 D3		3570	TIO	DFPRPE-DFPRNT(,@BR),@PERR	FORCE BRANCH TO DFPRNT ERROR	
		4DF3	3571	DFPEOR EQU	*-1	LAST BYTE OF FORCE DFPRNT ERROR	
4DF4	E1 E0 CD		3572	TIO	DLFRPE(,@XR),@PERR	FORCE BRANCH TO DLFPRT ERROR	
		4DF6	3573	DLFEOR EQU	*-1	LAST BYTE DLFPRT FORCE ERROR	
			3574	*	*****		
			3575	*	END V\$LPRT	*****	
			3576	*	*****		

## DLFPRT - LINE PRINTER ROUTINE

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	31/05/21	PAGE 262
					3578	*****				
					3579	*				
					3580	*	ENTRY TO FORMAT PRINT LINE			
					3581	*				
					3582	*****				
4E00					3583	ORG	*,256,0			SET STARTING ADDRESS
				2800	3584	USING	DFPASE,@BR			SET PAGE BASE ADDRESS - DFPRNT
				4D00	3585	USING	DLFPRT,@XR			SET PAGE BASE ADDRESS
				4E00	3586	VLPRT2 EQU	*			
4E00	2C	01	144A	D7	3587	MVC	I\$VADR,DLFVD1(@VADDR,@XR)			GET BUFFER ADDR
4E05	C0	87	1354		3588	B	I\$LOCK			LOCK PRINT BUFFER
4E09	8C	01	D9	144C	3589	MVC	BUFADR(2,@XR),I\$CADR			SAVE LINE PRINTER BUFFER CADDR
4E0E	8C	01	DE	144C	3590	MVC	BUFRWK(2,@XR),I\$CADR			SAVE BUFFER ADDRESS
					3591	*****				
					3592	*				
					3593	*	DETERMINE ANY MARGIN COMPUTATION REQUIRED			
					3594	*				
					3595	*****				
4E13	8C	00	DC	03E3	3596	MVC	DFPRES(1,@XR),\$BUFPT			SAVE COUNT
4E18	8C	00	DB	03C0	3597	MVC	DFPWT(1,@XR),\$RMGRN			SET RIGHT MARGIN VALUE
4E1D	8F	00	DB	03C1	3598	SLC	DFPWT(1,@XR),\$LMGRN			CALCULATE WIDTH
4E22	F2	87	04		3599	J	DLF525			CONTINUE
					3600	*				
				4E25	3601	DLF500 EQU	*			FORMAT LINE
4E25	AE	01	DE	DB	3602	ALC	BUFRWK(2,@XR),DFPWT(,@XR)			GET NEXT PDAR ADDR
				4E29	3603	DLF525 EQU	*			
4E29	AD	00	DB	DC	3604	CLC	DFPWT(1,@XR),DFPRES(,@XR)			COMPARE WIDTH TO LINE LNTH
4E2D	F2	02	0C		3605	JNL	DLF550			JUMP LENGTH < WIDTH
					3606	*****				
					3607	*				
					3608	*	COMPUTE MARGIN AND FORMAT DATA			
					3609	*				
					3610	*****				
4E30	AF	00	DC	DB	3611	SLC	DFPRES(1,@XR),DFPWT(,@XR)			NEXT LINE = RESIDUAL
4E34	2C	00	03E3	DB	3612	MVC	\$BUFPT(1),DFPWT(,@XR)			SET NEW LINE - WIDTH
4E39	F2	87	08		3613	J	DLF600			GO TO FORMAT NEXT LINE
					3614	*				
					3615	*	COUNT < WIDTH			
					3616	*				
				4E3C	3617	DLF550 EQU	*			
4E3C	2C	00	03E3	DC	3618	MVC	\$BUFPT(1),DFPRES(,@XR)			\$BUFPT RESIDUAL
4E41	7C	87	A0		3619	MVI	DFP270+@Q(,@BR),@UCB			FORCE LINE PRINT EXIT
					3620	*				
				4E44	3621	DLF600 EQU	*			FORMAT LINE
4E44	8C	00	DF	03E3	3622	MVC	DLFBPT(1,@XR),\$BUFPT			SAVE BUFFER POINTER
				4E49	3623	DLF700 EQU	*			PRINT RETRY ENTRY POINT
4E49	B1	E4	DE		3624	LIO	BUFRWK(,@XR),@PDAR			SET DATA ADDR
4E4C	6C	04	E2	E9	3625	MVC	DFPPCO(5,@BR),LPRCMD(,@XR)			SET LINE PRINTER CMDS.
					3626	*				
					3627	*	COMMON MARGIN ENTRY			
					3628	*				
4E50	7C	00	9E		3629	MVI	DFP260-DFPRNT+@D1(,@BR),@ZERO			SET TO PRINT RIGHT
4E53	8C	00	E4	03E5	3630	MVC	DFPPOS(1,@XR),\$LPROS			GET ACTUAL POSITION
4E58	0C	00	03E5	03C1	3631	MVC	\$LPROS(1),\$LMGRN			SET REFERENCE
4E5E	0E	00	03E5	03E3	3632	ALC	\$LPROS(1),\$BUFPT			UPDATE PRINT POSITION
					3633	*				

## DLFPRT - LINE PRINTER ROUTINE

ERR LOC OBJECT CODE ADDR STMT SOURCE STATEMENT VER 15, MOD 00 31/05/21 PAGE 263

4E64	1F	00	03E3 E7	3634	SLC	\$BUFPT(1),DLF001(,@BR)	COUNT LESS ONE
4E69	4C	00	E1 03E3	3635	MVC	DLFPCF+3(1,@BR),\$BUFPT	MOVE DATA COUNT TO PCF
4E6E	2D	00	03C1 E4	3636	CLC	\$LMRGN(1),DFPPOS(,@XR)	AT LEFT MARGIN ?
4E73	F2	81	61	3637	JE	DLF950	JUM IF AT LEFT MARGIN
				3639	*****		
				3640	*		
				3641	*		
				3642	***	IS PRINT POSITION < HALF OF DATA COUNT ?	
				3643	*	TAKE ONE-HALF OF COUNT ROUTINE (DIVIDE)	
				3644	*		
				3645	*****		
4E76	7C	00	E4	3646	MVI	DLFORK-1(,@BR),@ZERO	
4E79	4C	00	E5 03E3	3647	MVC	DLFORK(1,@BR),\$BUFPT	MOVE COUNT TO WORK AREA
4E7E	5E	01	E5 E5	3648	ALC	DLFORK(2,@BR),DLFORK(,@BR)	ADD THREE TIMES
4E82	5E	01	E5 E5	3649	ALC	DLFORK(2,@BR),DLFORK(,@BR)	
4E86	5E	01	E5 E5	3650	ALC	DLFORK(2,@BR),DLFORK(,@BR)	
4E8A	58	01	E4 E4	3651	MZN	DLFORK-1(,@BR),DLFORK-1(,@BR)	MOVE ZONE NUM
4E8E	58	02	E4 E5	3652	MNZ	DLFORK-1(,@BR),DLFORK(,@BR)	DLFORK-1=1/2 NEXT LINE CNT
				3653	*		
				3654	*		
				3655	MOVE CARRAGE TO LEFT MARGIN OR TAB		
4E92	8F	00	E4 03C1	3656	SLC	DFPPOS(1,@XR),\$LMRGN	PRPOS WITH IN WIDTH
4E97	9D	00	E4 E4	3657	CLC	DFPPOS(1,@XR),DLFORK-1(,@BR)	IS PRPOS > 1/2 NEXT LINE
4E9B	F2	82	2E	3658	JL	DLF900	SET TO GO TO LEFT MARGIN
				3660	*****		
				3661	*		
				3662	DETERMINE TAB DIRECTION		
				3663	*****		
4E9E	1E	00	03E3 E7	3663	ALC	\$BUFPT(1),DLF001(,@BR)	COUNT PLUS ONE
4EA3	0C	00	03E5 03C1	3664	MVC	\$LPROS(1),\$LMRGN	SET POSITION TO LEFT MARGIN
4EA9	7C	01	9E	3665	MVI	DFP260-DFPRNT+2(,@BR),@B1	SET TO PRINT LEFT
4EAC	8D	00	E4 03E3	3666	CLC	DFPPOS(1,@XR),\$BUFPT	COMPARE PRINT POS. TO LINE LNG
4EB1	F2	81	23	3667	JE	DLF950	JUMP EQUAL LINE & POSITION
4EB4	F2	84	10	3668	JH	DLF800	JUMP TO TAB LEFT

DLPRT - LINE PRINTER ROUTINE

ERR LOC    OBJECT CODE            ADDR STMT SOURCE STATEMENT            VER 15, MOD 00    31/05/21    PAGE 264

				3670	*					
				3671	*	COMPUTE	TAB	RIGHT		
				3672	*					
4EB7	2F	00	03E3	E4	3673	SLC	\$BUFPT(1),DFPPOS(,@XR)	GET	TAB	DISTANCE
4EBC	8C	00	E4	03E3	3674	MVC	DFPPOS(1,@XR),\$BUFPT	SAVE	BUFFER	POINTER
4EC1	7C	A0	DE		3675	MVI	DLFPCF(,@BR),DLTABR	SET	TAB	RIGHT OP
4EC4	F2	87	08		3676	J	DLF920	JUMP	TO	SET TAB COUNT



ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	31/05/21	PAGE 265
					3678	*				
					3679	*	COMPUTE LEFT TAB			
					3680	*				
	4EC7	8F 00 E4 03E3		4EC7	3681	DLF800 EQU	*	FIND TAB LEFT COUNT		
					3682	SLC	DFPPOS(1,@XR), \$BUFPT	GET TAB DISTANCE		
				4ECC	3683	DLF900 EQU	*	SET TAB LEFT		
	4ECC	7C 90 DE			3684	MVI	DLFPCF(, @BR), DLTABL	SET TAB LEFT OP		
				4ECF	3685	DLF920 EQU	*	HARDWARE REQUIREMENT		
	4ECF	9F 00 E4 E7			3686	SLC	DFPPOS(1,@XR), DLF001(, @BR)	ONE LESS		
	4ED3	6C 00 DF E4			3687	MVC	DLFPCF+1(, @BR), DFPPOS(, @XR)	SET TAB COUNT		
				4ED7	3688	DLF950 EQU	*	SET AT LEFT MARGIN INDICATION		
	4ED7	2C 01 03EA DF			3689	MVC	\$LPRIO, DLFBPT(2, @XR)	SAVE PDAR ADDR & BUFR. INCR.		
	4EDC	74 02 E5			3690	ST	DLFORK(, @BR), @XR	SAVE XR		
	4EDF	B5 02 D9			3691	L	BUFADR(, @XR), @XR	XR = CADDR LINE PRINTER BUFFER		
	4EE2	74 02 DD			3692	ST	DFPAPC(, @BR), @XR	SAVE BUFFER ADDR		
	4EE5	7C FB DD			3693	MVI	DFPAPC(, @BR), DLFCAR	GET DISP. TO COMMANDS		
	4EE8	9C 04 FF E2			3694	MVC	BFPCRO-LPBUFR(5, @XR), DFPPCO(, @BR)	MOVE COMMANDS TO PCAR		
	4EEC	75 02 E5			3695	L	DLFORK(, @BR), @XR	RESTORE XR TO VLPRT2		
	4EEF	3C 00 03E3			3696	MVI	\$BUFPT, @ZERO	SET BUFFER PTR = 0		
	4EF3	D0 87 99			3697	B	DFP250(, @BR)	GO TO DFPRNT TO DO I/O		
					3698	*				
					3699	*****	*****	*****		
					3700	*****	END V\$LPRT2	*****		
					3701	*****	*****	*****		

DLFPRT - LINE PRINTER ROUTINE

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	31/05/21	PAGE 266
					3703	*****	*****			
					3704	*	LINE DRINTER BUFFER AREA			
					3705	*****	*****			
4F00					3706	ORG	*,256,0			
				4F00	3707	USING	LPBUFR,@XR			SET BASE FOR BUFFER AREA
				4F00	3708	LPBUFR EQU	*			LINE PRINTER BUFFER AREA
4F00				4FFA	3709	DS	CL251			LINE PRINTER BUFFER AREA
					3711	*****	LINE PRINTER COMMANDS PCAR			*****
				4FFB	3712	BFPCAR EQU	*			LINE PRINTER COMMANDS
4FFB	0000000000			4FFF	3713	DC	XL5'00'			LINE PRINTER COMMANDS
				4FFF	3714	BFPCRO EQU	*-1			LAST BYTE OF COMMANDS
				00FB	3715	DLFCAR EQU	BFPCAR-LPBUFR			DISPLACEMENT TO PCAR
					3716	*****	*****			

## VLPRT3 - BI-DIRECTIONAL PRINT ROUTINE CORRECTION PAGE

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00 31/05/21	PAGE 267
5300				3718		ORG	X'5300'	PATCH AREA	1-5
				3719			*****		
				3720		*			*
				3721		*	THIS PAGE 15 USED BY THE BI-DIRECTIONAL PRINT ROUTINES TO CORRECT		*
				3722		*	PROBLEMS CONNECTED WITS APAR NUMBERS 968 AND 972. THE ROUTINES		*
				3723		*	USING THIS PAGE AND THEIR ENTRY POINTS ARE:		*
				3724		*	DFPRNT - VLPRT3, DFPENT		*
				3725		*	FZSPRT - VLPRT4		*
				3726		*	DLFPRT - VLPRT5, VLPRT6		*
				3727		*			*
				3728			*****		
				5300 3729	VLPRT3	EQU	*	DFPRNT INTERFACE	1-5
				5300 3730	DFPCHK	EQU	*		1-5
				2800 3731			USING DFPASE,@BR		1-5
				4D00 3732			USING DLFPRT,@XR		1-5
5300	7D	00	F6	3733		CLI	DFPIST+@PRCNT(,@BR),@ZERO	ANOTHER LINE TO PRINT	1-5
5303	F2	01	0B	3734		JNE	DFPENT	CONTINUE PROCESSING LINE	1-5
5306	F2	87	30	3735		J	DFPULK	GO TO UNLOCK ROUTINE	1-5
5309	C0	87	1354	3736		B	I\$LOCK	LOCK PAGE VLPRT3	1-5
530D	6C	03	F8 03	3737		MVC	DFPIST+@PLNGH-1(@PLNGH,@BR),@PLNGH-1(,@XR)	MOVE THE PRT	1-5
				3738		*		* PARAMETER LIST TO WRK AREA	1-5
5311	5C	02	F4 F8	3739	DFPENT	MVC	DFPDSV(@CADDR+1,@BR),DFPIST+@PDATA(,@BR)	MOVE THE PRT	1-5
				3740		*		* CNT AND DATA ADDRESS	1-5
5315	4C	00	FB 03C2	3741		MVC	DFPSYC+@SYCNT(1,@BR),\$PRPOS	SAVE HD POSITION FOR SYNC	1-5
531A	5C	01	DF F6	3742		MVC	DFPPCF+@PRCNT(2,@BR),DFPIST+@PRCNT(,@BR)	SET CTRL+CNT	1-5
531E	39	1E	03E4	3743		TBF	\$LPRP3,@KENAB	TEST FOP MATRIX PRINT MODE	1-5
5322	D0	90	23	3744		BF	DFP115(,@BR)	BRANCH IF MATRIX PRINT	1-5
5325	38	80	03D2	3745		TBN	\$IOIND,\$LNPTR	IS LINE PRINTER REQUESTED ?	1-5
5329	D0	90	23	3746		BF	DFP115(,@BR)	BRANCH IF NOT	1-5
532C	C0	87	1330	3747		B	I\$LDXR	BRANCH TO LOAD PAGE	1-5
5330	4D00			5331 3748		DC	AL(@VADDR)(V\$LPRT)	LINE PRINTER PAGE	1-5
5332	C0	87	1354	3749		B	I\$LOCK	GO LOCK PAGE	1-5
5336	E0	87	00	3750		B	@ZERO(,@XR)	BRANCH TO LINE PRINTER LINK	1-5
				3751		*			1-5
				5339 3752	DFPULK	EQU	*	UNLOCK ALL LINE PRINTER	1-5
				3753		*		* ROUTINE PAGES	1-5
5339	7C	80	A3	3754		MVI	DFP280+@Q-DFPASE(,@BR),@NOP	SET ERP INDR OFF	1-5
533C	1C	01	144A 1F	3755		MVC	I\$VADR,DFP105(2,@BR)	DLFPRT VM ADDR	1-5
5341	C0	87	1350	3756		B	I\$UNLK	UNLOCK PAGE	1-5
5345	3C	4E	1449	3757		MVI	I\$VADR-1,DLFX4E	VLPRT2 VM ADDR	1-5
5349	C0	87	1350	3758		B	I\$UNLK	UNLOCK PAGE	1-5
534D	3C	53	1449	3759		MVI	I\$VADR-1,DLFX53	VLPRT3 VM ADDR	1-5
5351	C0	87	1350	3760		B	I\$UNLK	UNLOCK PAGE	1-5
5355	C0	87	12D3	3761		B	I\$RTRN	BRANCH TO CALLING PGM-FZPRNT	1-5
				3762		*			1-5
				5359 3763	VLPRT4	EQU	*	FZSPRT INTERFACE	1-5
				3600 3764		USING	FZSP3B,@BR		1-5
5359	4E	00	DB 03C2	3765	FZS991	ALC	FZS3CC(,@BR),\$PRPOS(1)	ADD PRT ZONE LNG TO CURRENT	1-5
535E	5D	00	DB 6A	3766		CLC	FZS3CC(,@BR),FZS3RM(1,@BR)	* CARRIER POSITION - BRANCH	1-5
5362	F2	84	03	3767		JH	FZS992	* IF RIGHT MGN IS EXCEEDED	1-5
5365	D0	87	B9	3768		B	FZS710(,@BR)	BRANCH BACK IF NOT	1-5
5368	38	80	03D2	3770	FZS992	TBN	\$IOIND,\$LNPTR	IS LINE PRINTER REQUESTED ?	1-5
536C	F2	90	03	3771		JF	FZS993	NO, DON'T SET CARRIAGE RTN	1-5
536F	7C	C0	F2	3772		MVI	FZS3PF(,@BR),@PRETR	SET CARRIAGE RETURN INDR	1-5
5372	D2	02	F2	3773	FZS993	LA	FZS3PL(,@BR),@XR	LOAD DATA OUTDUT PPL CADDR	1-5

## VLPRT3 - BI-DIRECTIONAL PRINT ROUTINE CORRECTION PAGE

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	31/05/21	PAGE 268
5375	C0	87	12B1		3774	B	I\$CALL			LINK TO EXECUTE PRINTER IOCR 1-5
5379	2800			537A	3775	DC	AL(@VADDR)(V\$SPRT)			MATRIX PRINTER IOCR VADDR 1-5
537B	7C	40	F2		3776	MVI	FZS3PF(, @BR), @PRINT			SET INDR TO PRINT ONLY 1-5
537E	0D	00	044A	0D5A	3777	CLC	\$PRDEV-1, I\$WRK2-1(1)			IF CRT IS NOT A SYSTEM PRINT 1-5
5384	F2	82	06		3778	JL	FZS994			* DEVICE, EXIT ROUTINE 1-5
5387	C0	87	12B1		3779	B	I\$CALL			LINK TO EXCUTE PRINT ON CRT 1-5
538B	3700			538C	3780	DC	AL(@VADDR)(FZS800)			PRINT CRT RTN VADDR 1-5
538D	C0	87	12D3		3781	FZS994 B	I\$RTRN			RETURN TO 1ST PRINT RTN PAGE 1-5
				5391	3783	VLPRT5 EQU	*			DLFPRT INTERFACE NO. 1 1-5
				2800	3784	USING	DFPASE, @BR			1-5
				4D00	3785	USING	DLFPRT, @XR			1-5
5391	5F	01	F2	E7	3786	SLC	DLFDSV-2(2, @BR), DLF001(, @BR)			COUNT LESS ONE 1-5
5395	BD	01	F0		3787	CLI	DLFSWC(, @XR), DLFRTN			IS SWITCH SET FOR RTN CARRAGE 1-5
5398	F2	81	04		3788	JE	DLF960			YES, DO NOT INCR DATA PTR 1-5
539B	5E	01	F8	F2	3789	ALC	DLFIST+@PDATA(2, @BR), DLFDSV-2(, @BR)			GET DATA ADDR PTR 1-5
539F	9C	01	62	F8	3790	DLF960 MVC	DLF150+@DOP2(2, @XR), DLFIST+@PDATA(, @BR)			SET DATA ADDR 1-5
53A3	9C	00	5F	F2	3791	MVC	DLF150+@VQ(1, @XR), DLFDSV-2(, @BR)			GET COUNT FOR MVC 1-5
53A7	8C	00	60	03E3	3792	MVC	DLF150+@D1(1, @XR), \$BUFPT			MOVE BUFFER DISP. INTO INST. 1-5
53AC	9F	00	60	E7	3793	SLC	DLF150+@D1(1, @XR), DLF001(, @BR)			DISP. LESS ONE 1-5
53B0	BC	00	F0		3794	MVI	DLFSWC(, @XR), X'00'			SET CARRAGE RETURN SW OFF 1-5
53B3	E0	87	5B		3795	B	DLF146(, @XR)			CONTINUE 1-5
					3796	*				
				53B6	3797	VLPRT6 EQU	*			DLFPRT INTERFACE NO. 2 1-5
53B6	7C	40	F5		3798	MVI	DLFIST+@PCTRL(, @BR), @PRINT			SET PRINT ONLY 1-5
53B9	6C	00	F6	DC	3799	MVC	DLFIST+@PRCNT(, @BR), DFPRES(1, @XR)			BUF PTR - RESIDUAL 1-5
53BD	6C	00	F2	DC	3800	MVC	DLFDSV-2(, @BR), DFPRES(1, @XR)			DATA COUNT - RESIDUAL 1-5
53C1	0C	00	03C2	03C1	3801	MVC	\$PRPOS(1), \$LMRGN			SET DUMMY POSITION-LEFT MGN. 1-5
53C7	BC	01	F0		3802	MVI	DLFSWC(, @XR), DLFRTN			SET SWITCH FOR RTN CARRIAGE 1-5
53CA	E0	87	25		3803	B	DLF100(, @XR)			CONTINUE PROCESSING 1-5
					3804	*##### X'5400' #####				
					3805	* N O T S C A N N E D (GENERAL PURPOSE BUFFERS 1 & 2.)				
					3806	*##### X'55FF' #####				
				FFFF	3807	END				

TOTAL STATEMENTS IN ERROR IN THIS ASSEMBLY = 0

## CROSS REFERENCE

VER 15, MOD 00 31/05/21 PAGE 269

SYMBOL	LEN	VALUE	DEFN	REFERENCES
\$\$\$CMD	001	0020	0660	
\$\$\$DAT	001	0040	0659	
\$\$\$EPL	001	0091	0656	
\$\$\$ERN	001	0080	0710	
\$\$\$FUN	001	0010	0661	
\$\$\$NLN	001	00A0	0706	
\$\$\$STD	001	0081	0655	
\$\$\$001	040	2DF8	0404	
\$\$BNLN	001	0605	0636	0638
\$\$CDBS	001	08C0	0686	
\$\$CDND	001	0666	0645	
\$\$CDRD	001	0890	0684	0686
\$\$CKEY	001	0603	0634	
\$\$CKFF	001	0B3D	0666	
\$\$COFF	001	0B44	0665	
\$\$CSNS	001	209C	0695	
\$\$DATB	001	0BBF	0667	
\$\$EOSA	001	0AFE	0664	
\$\$ERSK	001	1C00	0705	
\$\$FITS	001	1D00	0713	
\$\$FLIB	001	06FF	0712	
\$\$ILEN	001	0601	0630	0632 0636
\$\$ILHD	001	0600	0628	0630
\$\$INLN	001	0607	0643	0645 0647
\$\$INND	001	06FA	0647	
\$\$KBDT	001	09E1	0654	0658
\$\$KBSN	001	09E2	0658	0663
\$\$KLD1	001	0600	0718	7877 8607
\$\$KLD2	001	0700	0720	
\$\$KLD3	001	0C00	0722	
\$\$LPOS	001	09EB	0663	
\$\$PCNT	001	07E9	0679	
\$\$PLYN	001	2004	0693	7676 9165
\$\$PRES	001	0890	0652	0654 0664 0665 0666 0667 0684
\$\$PRFL	001	2143	0697	
\$\$PRNT	001	0707	0673	0674 0678 0679 7675
\$\$PRTN	001	0782	0674	
\$\$PSIO	001	07CE	0678	
\$\$PYCD	001	2200	0699	
\$\$PYMP	001	2000	0691	0693 0695 0697 0699
\$\$SLIB	001	1C00	0708	
\$\$TPCD	001	0606	0638	0643
\$\$UPAR	001	0602	0632	0634
\$\$WSPB	001	1E00	0711	
\$\$XIND	001	06FF	0709	0712
\$\$ZERO	001	0000	0224	0225 0227 0228 0229 0233 0691
\$ABORT	001	0010	0337	
\$BASIC	001	0080	0395	
\$BIGCD	001	0080	0471	7945
\$BLDPL	001	0579	0604	0606
\$BLNOE	001	0569	0594	
\$BLOAD	001	0522	0585	0587 0590 0603 0604 7825 8473
\$BLRTN	001	0550	0593	0594
\$BRSAV	001	03C5	0282	0283 8914* 8934
\$BSADR	001	0587	0609	0611
\$BUFPT	001	03E3	0490	0491 8729 9517* 9531* 3466* 3493 3596 3612* 3618* 3622 3632 3634*

CROSS REFERENCE															
S Y M B O L	L E N	V A L U E	D E F N	R E F E R E N C E S								VER 15, MOD 00    31/05/21    PAGE 270			
\$CABLD	001	04B4	0563	3635	3647	3663*	3666	3673*	3674	3682	3696*	3792			
\$CAERK	001	0469	0540	0564											
\$CAERR	001	03CD	0288	0543											
\$CAIPL	001	049D	0559	0290											
\$CALLI	001	0008	0480	0561											
\$CARDI	001	0001	0251												
\$CARPL	001	04A1	0561	0563											
\$CIENT	001	0483	0550	0551	9041	9050									
\$CIEXT	001	0480	0549	0550											
\$CIMSK	001	0476	0546	0549											
\$CISUS	001	0496	0554	0559	0003										
\$CLBFR	001	0010	0438												
\$CMDKY	001	0008	0350												
\$CMODE	001	0002	0400												
\$CONFIG	001	03DD	0463	0473	7945										
\$CRPOS	001	03E2	0489	0490	2807	2814	2921								
\$CRTAD	001	044D	0528	0529											
\$CRTAV	001	0002	0344												
\$CRTDN	001	0002	0368												
\$CRTIN	001	03D3	0365	0372											
\$CRTNO	001	0004	0347												
\$CRTPU	001	0004	0369												
\$CRTSP	001	0008	0370												
\$CRTUP	001	0001	0367												
\$CRUSH	001	0080	0476												
\$CSDPL	001	050E	0575	0576											
\$C0001	001	0464	0532	0538											
\$DATE	001	043A	0513	0514											
\$DBGUF	001	03E0	0475	0484	7411										
\$DBLOK	001	0001	0425												
\$DFDET	001	03E8	0496	0497											
\$DISKN	001	0025	0227	7816	7836	8462	8487	3264	3275						
\$DKERR	001	0008	0406												
\$DKSIZ	001	03D7	0450	0458	0499										
\$DK100	001	0001	0452												
\$DK200	001	0002	0453												
\$DK400	001	0004	0454												
\$DK600	001	0008	0455												
\$DK800	001	0010	0456												
\$DPLSV	001	0449	0524	0526	7832	8481									
\$DTNMB	001	0040	0271												
\$DTRDR	001	0040	0359												
\$ENDNU	001	0600	0618	0628	0652	0673	0709	0718	0720	0722	2752				
\$ERDPL	001	046F	0543	05											

## CROSS REFERENCE

VER 15, MOD 00 31/05/21 PAGE 271

SYMBOL	LEN	VALUE	DEFN	REFERENCES
\$ER1N2	001	0050	0300	
\$EXADR	001	0517	0578	0580
\$EXCMD	001	0001	0332	
\$EXFTR	001	043B	0514	0519 7592 8922 0236 2178 3156 3159 3161
\$FCIND	001	0010	0410	
\$FDIND	001	0040	0417	
\$FEARR	001	0004	0225	
\$FEMAP	001	0588	0611	0612
\$FILIB	001	03DA	0461	0462
\$FITIN	001	0010	0386	
\$FUIND	001	0020	0415	
\$GUFIO	001	0583	0608	0609
\$GUFIR	001	0008	0260	
\$HISTE	001	042E	0511	0512 9476* 9526*
\$HIST1	001	0435	0512	0513 9291* 9471*
\$HRDER	001	0020	0356	9288 9525
\$INDR1	001	03D4	0372	0398
\$INDR2	001	03D5	0398	0423 9296* 9469 9472*
\$INDR3	001	03D6	0423	0450
\$INLNO	001	03CF	0290	0292 0304 0311 0141
\$INRPT	001	0020	0268	
\$IOIND	001	03D2	0339	0365 9288* 9525* 3745 3770
\$IOPGS	001	0010	0479	7411
\$IOYES	001	0002	0254	
\$IPLDV	001	05FF	0615	0618
\$IRKEY	001	0020	0478	
\$KEYBD	001	03E1	0484	0489
\$KEYCD	001	03C3	0248	0282
\$KEYDT	001	0040	0392	
\$KE090	001	00DE	0228	
\$KE130	001	01D5	0229	
\$KYBSY	001	0010	0265	
\$LDRTN	001	0571	0603	
\$LEVEL	001	03DF	0473	0475
\$LIST	001	0002	0427	
\$LMRGN	001	03C1	0243	0245 8927 9357 9360 9488 3526 3598 3631 3636 3656 3664 3801
\$LNPTR	001	0080	0362	3745 3770
\$LOADB	001	054A	0587	
\$LOADR	001	051A	0580	0583
\$LPRIO	001	03EA	0497	9516 3689*
\$LPROS	001	03E5	0492	0494 8739 9465 9506* 3461* 3630 3631* 3632* 3664*
\$LPRP3	001	03E4	0491	0492 8628* 8705* 8737 8740* 9170* 9173* 9463 9466* 9483 9503 9505* 9527 9992* 0056* 0257* 3433* 3458 3460* 3532* 3743
\$MOUNT	001	0020	0441	
\$MPDWN	001	0001	0341	9525
\$NEXTB	001	03E6	0494	0495
\$NEXTL	001	03E7	0495	0496
\$NOENB	001	0008	0433	
\$NOLST	001	0004	0257	
\$NUCBS	001	03C0	0240	0241
\$NWRKF	001	0080	0446	
\$NWRKR	001	0040	0443	
\$PASWD	001	042D	0510	0511
\$PAUSD	001	04BA	0564	0566
\$PAUSE	001	0002	0334	
\$PGMDT	001	0020	0389	



## CROSS REFERENCE

VER 15, MOD 00 31/05/21 PAGE 272

SYMBOL	LEN	VALUE	DEFN	REFERENCES
\$PGMST	001	0010	0353	
\$PKERT	001	0419	0508	0510
\$PLST1	001	0454	0529	0530
\$PLST2	001	045B	0530	0531
\$PLST3	001	0462	0531	0532
\$PRDEV	001	044B	0526	0528 7584 7591* 7592* 7607* 9163 9168 0240 0252 2449 2623 3777
\$PRESN	001	0002	0377	
\$PROCI	001	0001	0374	
\$PRPOS	001	03C2	0245	0248 8739* 9331* 9337 9348* 9355 9360* 9465* 9499* 9506 2524 2531 2647 3461 3467* 3526* 3741 3765 3801*
\$PSDBR	001	04FA	0569	
\$PSDXR	001	04F2	0568	0569
\$PSTEP	001	0004	0335	
\$PSTMT	001	0008	0336	
\$PTCH1	001	03F5	0499	0503
\$READY	001	0080	0419	
\$REORD	001	0040	0477	
\$RLOAD	001	051E	0583	0585
\$RMGRN	001	03C0	0241	0243 8926 9338 2454 3597
\$RSTR	001	04D6	0566	0568 0570 0575
\$RUNIT	001	0001	0313	
\$SFAID	001	050D	0571	
\$SPRNT	001	0465	0538	0540
\$SRTRN	001	04FE	0570	0571
\$STEPT	001	0002	0314	
\$SWPCR	001	0511	0576	0578
\$TABLN	001	03CB	0285	0288
\$TFLOW	001	0008	0320	
\$TRACE	001	0004	0315	
\$TRALL	001	0010	0321	
\$TROVR	001	054E	0590	0593
\$TRUNK	001	0080	0273	
\$TRVAR	001	0020	0322	
\$UNMSK	001	048D	0551	0554
\$USRDR	001	03DC	0462	0463
\$VMDEF	001	0080	0326	
\$VOLF1	001	03FE	0505	0506
\$VOLF2	001	040E	0507	
\$VOLID	001	03F6	0503	0504 0508
\$VOLR1	001	03F6	0504	0505
\$VOLR2	001	0406	0506	0507
\$WAITF	001	057F	0606	0608 7837 8488 3276
\$WFDEF	001	0040	0520	
\$WFLOK	001	0008	0383	
\$WFNME	001	0443	0519	0524
\$WSIND	001	0004	0380	
\$XIND1	001	03D0	0311	0330
\$XIND2	001	03D1	0330	0339
\$XIND3	001	03D8	0458	0461
\$XPREC	001	0040	0323	
\$XRSAB	001	03C7	0283	0285
\$ZTRAD	001	05A2	0612	
\$12K	001	0004	0467	
\$16CKY	001	0008	0469	
\$16K	001	0002	0466	
\$22IMP	001	0001	0464	

## CROSS REFERENCE

VER 15, MOD 00 31/05/21 PAGE 273

SYMBOL	LEN	VALUE	DEFN	REFERENCES
###BL	001	0000	1437	
###CK	001	0000	1565	
###CN	001	0000	1533	
###CO	001	0000	1325	
###CS	001	0000	1385	
###DR	001	0000	1129	
###ER	001	0000	1329	
###FS	001	0000	1425	
###IN	001	0000	1569	
###PW	001	0000	1573	
###RS	001	0000	1405	
###SA	001	0000	1393	
###SS	001	0000	1389	
###VU	001	0600	1349	
###0T	001	0700	1121	
###1T	001	0000	1125	
###BCO	001	0600	1137	
###BOV	001	0800	1409	
###DPR	001	0700	1145	
###DRE	001	0889	1161	
###DSP	001	2800	1181	
###ECM	001	0C00	1441	
###EFK	001	0C00	1461	
###ERR	001	0C00	1433	
###EXM	001	0C00	1321	
###FIL	001	0E00	1401	
###FIS	001	0E00	1397	
###FML	001	0200	1529	
###FMS	001	0200	1369	3964
###GRA	001	0889	1293	
###GUF	001	0C00	1429	
###INL	001	0600	1509	
###INS	001	0600	1133	7857 8583
###KAL	001	0C00	1297	
###KCA	001	0C00	1513	
###KCH	001	0C00	1265	
###KCN	001	0C00	1381	
###KCT	001	0C00	1233	
###KDE	001	0C00	1229	
###KDI	001	0D00	1309	
###KDN	001	0C00	1217	
###KDO	001	0E00	1313	
###KED	001	0C00	1153	
###KEN	001	0C00	1157	
###KEX	001	0C00	1177	
###KGO	001	0C00	1149	
###KHE	001	0C00	1333	
###KKE	001	0C00	1561	
###KLI	001	0C00	1237	
###KLL	001	0920	1537	
###KLO	001	0C00	1241	
###KME	001	0D00	1221	
###KMO	001	0C00	1165	
###KNA	001	0C00	1277	
###KOV	001	0E00	1197	
###KPA	001	0C00	1173	

## CROSS REFERENCE

SYMBOL LEN VALUE DEFN REFERENCES VER 15, MOD 00 31/05/21 PAGE 274

###KPO 001 0C00 1261  
###KPR 001 0C00 1285  
###KRE 001 0C00 1205  
###KRL 001 0700 1301  
###KRM 001 0C00 1169  
###KRN 001 0700 1189  
###KRO 001 0D00 1193  
###KRS 001 0C00 1517  
###KRU 001 0C00 1213  
###KRV 001 0800 1305  
###KSA 001 0C00 1249  
###KSE 001 0E00 1289  
###KSO 001 0C20 1341  
###KSS 001 0C00 1273  
###KSV 001 0980 1269  
###KSY 001 0C00 1281  
###KWI 001 0C00 1209  
###KWR 001 0C00 1201  
###LOA 001 0600 1141  
###MIP 001 0C00 1337  
###SDS 001 0C00 1449  
###SFF 001 0E00 1453  
###SFL 001 0F00 1445  
###SFO 001 1500 1417  
###SFS 001 0C00 1413  
###SPA 001 0C00 1253  
###SPO 001 0806 1257  
###SPS 001 0C00 1245  
###STR 001 1600 1421  
###TDC 001 1000 1225  
###TSY 001 1000 1185  
###TVK 001 0FC0 1361  
###UAL 001 0C00 1377  
###UAT 001 0900 1473  
###UCD 001 0900 1481  
###UCN 001 0C00 1465  
###UCP 001 0700 1469  
###UDE 001 0C00 1485  
###UDI 001 0C00 1489  
###UEX 001 0C00 1373  
###UIN 001 0C00 1477  
###UPA 001 0C00 1457  
###UPO 001 0C00 1525  
###UPT 001 0C00 1521  
###VCR 001 2000 1317  
###VLO 001 0600 1353  
###VOD 001 0600 1357  
###VVM 001 0000 1365  
###VXI 001 0600 1345  
###ZDU 001 1100 1497  
###ZLB 001 1100 1541  
###ZLO 001 1100 1501  
###ZLV 001 0F00 1557  
###ZL1 001 0F00 1545  
###ZL2 001 0F00 1549  
###ZL3 001 0C00 1553

7867 8595

## CROSS REFERENCE

VER 15, MOD 00 31/05/21 PAGE 275

SYMBOL	LEN	VALUE	DEFN	REFERENCES
\$\$\$ZTR	001	1000	1493	
\$\$\$ZUT	001	0C00	1505	
\$\$#BLN	001	18D4	1436	
\$\$#CKT	001	2118	1564	
\$\$#CNF	001	2000	1532	
\$\$#COR	001	0800	1324	
\$\$#CSA	001	1000	1384	
\$\$#DRT	001	0000	1128	
\$\$#ERM	001	0928	1328	
\$\$#FSP	001	1880	1424	
\$\$#INV	001	212C	1568	
\$\$#PWR	001	2300	1572	
\$\$#RSP	001	1780	1404	
\$\$#SAV	001	1180	1392	
\$\$#SSA	001	1128	1388	
\$\$#VUF	001	0B08	1348	
\$\$#0TR	001	0000	1120	
\$\$#1TR	001	0080	1124	
\$\$@#BL	001	0001	1438	
\$\$@#CK	001	0004	1566	
\$\$@#CN	001	0001	1534	
\$\$@#CO	001	003A	1326	
\$\$@#CS	001	003A	1386	
\$\$@#DR	001	0008	1130	
\$\$@#ER	001	0032	1330	
\$\$@#FS	001	0030	1426	
\$\$@#IN	001	003A	1570	
\$\$@#PW	001	00C0	1574	
\$\$@#RS	001	0030	1406	
\$\$@#SA	001	0108	1394	
\$\$@#SS	001	0001	1390	
\$\$@#VU	001	0002	1350	
\$\$@#0T	001	0018	1122	
\$\$@#1T	001	0018	1126	
\$\$@BCO	001	0018	1138	
\$\$@BOV	001	0018	1410	
\$\$@DPR	001	0005	1146	
\$\$@DRE	001	0001	1162	
\$\$@DSP	001	0004	1182	
\$\$@ECM	001	0006	1442	
\$\$@EFK	001	0002	1462	
\$\$@ERR	001	0003	1434	
\$\$@EXM	001	0003	1322	
\$\$@FIL	001	0009	1402	
\$\$@FIS	001	0009	1398	
\$\$@FML	001	0052	1530	
\$\$@FMS	001	0052	1370	
\$\$@GRA	001	0003	1294	
\$\$@GUF	001	0010	1430	
\$\$@INL	001	0010	1510	
\$\$@INS	001	0010	1134	
\$\$@KAL	001	000F	1298	
\$\$@KCA	001	000C	1514	
\$\$@KCH	001	000C	1266	
\$\$@KCN	001	0010	1382	
\$\$@KCT	001	0009	1234	

## CROSS REFERENCE

VER 15, MOD 00 31/05/21 PAGE 276

SYMBOL	LEN	VALUE	DEFN	REFERENCES
#\$@KDE	001	0010	1230	
#\$@KDI	001	0005	1310	
#\$@KDN	001	0010	1218	
#\$@KDO	001	000C	1314	
#\$@KED	001	000E	1154	
#\$@KEN	001	0006	1158	
#\$@KEX	001	0003	1178	
#\$@KGO	001	0002	1150	
#\$@KHE	001	000C	1334	
#\$@KKE	001	0006	1562	
#\$@KLI	001	0011	1238	
#\$@KLL	001	0001	1538	
#\$@KLO	001	0008	1242	
#\$@KME	001	0003	1222	
#\$@KMO	001	0004	1166	
#\$@KNA	001	0008	1278	
#\$@KOV	001	0009	1198	
#\$@KPA	001	0005	1174	
#\$@KPO	001	000D	1262	
#\$@KPR	001	0009	1286	
#\$@KRE	001	0002	1206	
#\$@KRL	001	0004	1302	
#\$@KRM	001	0003	1170	
#\$@KRN	001	0003	1190	
#\$@KRO	001	000A	1194	
#\$@KRS	001	000A	1518	
#\$@KRU	001	0003	1214	
#\$@KRV	001	000D	1306	
#\$@KSA	001	0011	1250	
#\$@KSE	001	0004	1290	
#\$@KSO	001	000D	1342	
#\$@KSS	001	000B	1274	
#\$@KSV	001	0002	1270	
#\$@KSY	001	000F	1282	
#\$@KWI	001	0002	1210	
#\$@KWR	001	0002	1202	
#\$@LOA	001	0013	1142	
#\$@MIP	001	000D	1338	
#\$@SDS	001	0004	1450	
#\$@SFF	001	0008	1454	
#\$@SFL	001	0005	1446	
#\$@SFO	001	0003	1418	
#\$@SFS	001	0011	1414	
#\$@SPA	001	0004	1254	
#\$@SPO	001	0003	1258	
#\$@SPS	001	0001	1246	
#\$@STR	001	0002	1422	
#\$@TDC	001	0003	1226	
#\$@TSY	001	0003	1186	
#\$@TVK	001	0001	1362	
#\$@UAL	001	0011	1378	
#\$@UAT	001	000C	1474	
#\$@UCD	001	000B	1482	
#\$@UCN	001	0009	1466	
#\$@UCP	001	000F	1470	
#\$@UDE	001	000E	1486	

## CROSS REFERENCE

VER 15, MOD 00 31/05/21 PAGE 277

SYMBOL	LEN	VALUE	DEFN	REFERENCES
#\$@UDI	001	0008	1490	
#\$@UEX	001	000E	1374	
#\$@UIN	001	000F	1478	
#\$@UPA	001	0004	1458	
#\$@UPO	001	0005	1526	
#\$@UPT	001	0012	1522	
#\$@VCR	001	0008	1318	
#\$@VLO	001	0002	1354	
#\$@VOD	001	0016	1358	
#\$@VVM	001	0030	1366	
#\$@VXI	001	0002	1346	
#\$@ZDU	001	0008	1498	
#\$@ZLB	001	0002	1542	
#\$@ZLO	001	000C	1502	
#\$@ZLV	001	0006	1558	
#\$@ZL1	001	0007	1546	
#\$@ZL2	001	000D	1550	
#\$@ZL3	001	000A	1554	
#\$@ZTR	001	0001	1494	
#\$@ZUT	001	0014	1506	
#\$BCOM	001	0080	1136	
#\$BOLV	001	1780	1408	
#\$DPRI	001	014C	1144	
#\$DREA	001	0200	1160	
#\$DSPL	001	0240	1180	
#\$ECMA	001	1900	1440	
#\$EFKE	001	1990	1460	
#\$ERRP	001	18C0	1432	
#\$EXMS	001	07D4	1320	
#\$FILN	001	1724	1400	
#\$FIST	001	1700	1396	
#\$FMLN	001	1E00	1528	
#\$FMST	001	0D00	1368	
#\$GRAP	001	0690	1292	
#\$GUFU	001	1880	1428	
#\$INLN	001	1C84	1508	
#\$INST	001	0020	1132	
#\$KALL	001	06A4	1296	
#\$KCAL	001	1CC4	1512	
#\$KCHA	001	053C	1264	
#\$KCND	001	0F80	1380	
#\$KCTL	001	03BC	1232	
#\$KDEL	001	035C	1228	
#\$KDIS	001	0744	1308	
#\$KDNT	001	0300	1216	
#\$KDOV	001	0780	1312	
#\$KEDI	001	0188	1152	
#\$KENA	001	01C4	1156	
#\$KEXT	001	0234	1176	
#\$KGOS	001	0180	1148	
#\$KHEL	001	0A30	1332	
#\$KKEY	001	2100	1560	
#\$KLIS	001	0400	1236	
#\$KLLA	001	2004	1536	
#\$KLOG	001	0444	1240	
#\$KMER	001	030C	1220	

## CROSS REFERENCE

VER 15, MOD 00 31/05/21 PAGE 278

SYMBOL	LEN	VALUE	DEFN	REFERENCES
#\$KMOU	001	0204	1164	
#\$KNAM	001	05C0	1276	
#\$KOVN	001	0290	1196	
#\$KPAS	001	0220	1172	
#\$KPOO	001	0508	1260	
#\$KPRT	001	063C	1284	
#\$KREA	001	02BC	1204	
#\$KRLA	001	0700	1300	
#\$KRMO	001	0214	1168	
#\$KRNU	001	0280	1188	
#\$KROV	001	028C	1192	
#\$KRSU	001	1D24	1516	
#\$KRUN	001	02CC	1212	
#\$KRVL	001	0710	1304	
#\$KSAV	001	0488	1248	
#\$KSET	001	0680	1288	
#\$KSOV	001	0AC8	1340	
#\$KSSP	001	0594	1272	
#\$KSVL	001	058C	1268	
#\$KSYM	001	0600	1280	
#\$KWID	001	02C4	1208	
#\$KWRI	001	02B4	1200	
#\$LOAD	001	0100	1140	
#\$MIPP	001	0A80	1336	
#\$SDSY	001	192C	1448	
#\$SFFI	001	193C	1452	
#\$SFLO	001	1918	1444	
#\$SFOV	001	1844	1416	
#\$SFSY	001	1800	1412	
#\$SPAC	001	04CC	1252	
#\$SPOV	001	04DC	1256	
#\$SPSY	001	0484	1244	
#\$STRO	001	1850	1420	
#\$TDCK	001	0350	1224	
#\$TSYK	001	0250	1184	
#\$TVKB	001	0BAC	1360	
#\$UALL	001	0F00	1376	
#\$UATR	001	1A38	1472	
#\$UCDI	001	1AD8	1480	
#\$UCNF	001	19B8	1464	
#\$UCPL	001	19DC	1468	
#\$UDEL	001	1B24	1484	
#\$UDIS	001	1B5C	1488	
#\$UEXL	001	0EA8	1372	
#\$UINI	001	1A88	1476	
#\$UPAC	001	1980	1456	
#\$UPOV	001	1D24	1524	
#\$UPTF	001	1D5C	1520	
#\$VCRT	001	07B4	1316	
#\$VLOA	001	0B80	1352	
#\$VODK	001	0B88	1356	
#\$VVMR	001	0C00	1364	
#\$VXIT	001	0B00	1344	
#\$ZDUM	001	1BA4	1496	
#\$ZLBM	001	2008	1540	
#\$ZLOA	001	1BC4	1500	



## CROSS REFERENCE

VER 15, MOD 00 31/05/21 PAGE 279

SYMBOL	LEN	VALUE	DEFN	REFERENCES
#\$ZLVR	001	20B0	1556	
#\$ZL1M	001	2010	1544	
#\$ZL2M	001	2030	1548	
#\$ZL3M	001	2088	1552	
#\$ZTRA	001	1B9C	1492	
#\$ZUTM	001	1C14	1504	
#@#BAD	001	0455	0880	
#@#IO1	001	0459	0888	
#@#IO2	001	045D	0889	
#@#TAT	001	0941	0916	
#@#TBA	001	09A1	0920	
#@#TFS	001	0941	0914	
#@#TSY	001	0941	0918	
#@#VFP	001	0700	0906	
#@#VLP	001	093D	0909	
#@#WDB	001	050C	0901	
#@#WFT	001	0500	0899	
###BA	001	0001	0881	
###IO	001	0001	0893	
###SC	001	0002	0890	
###TA	001	0010	0917	
###TB	001	0010	0921	
###TS	001	0005	0919	
###TW	001	0020	0915	
###VM	001	0100	0910	
###WD	001	00BD	0902	
###WF	001	0003	0900	
###04	001	0004	0892	
###08	001	0008	0891	
###BOV	001	0018	0869	
###ECM	001	0006	0883	
###ERR	001	0003	0877	
###GUF	001	0010	0873	
###LDS	001	0002	0879	
###SDS	001	0004	0875	
###SFF	001	0008	0887	
###SFL	001	0005	0885	7866 8594
###SFO	001	0005	0895	
###SFS	001	0011	0871	
###VSF	001	0010	0923	
###VSL	001	000F	0924	7856 8582
###VTR	001	0001	0908	
#@BOVL	001	0400	0868	
#@CORS	001	0005	0774	
#@ECMA	001	0481	0882	
#@ERRP	001	0441	0876	
#@GUFU	001	0401	0872	
#@LDSV	001	044D	0878	
#@MVSD	001	0001	0782	
#@NERO	001	0003	0776	
#@OBRA	001	0002	0778	
#@PTFL	001	0006	0797	
#@PTFS	001	0001	0796	
#@SDSY	001	04AD	0874	
#@SFFI	001	04BD	0886	
#@SFLO	001	0499	0884	7865 8593

## CROSS REFERENCE

VER 15, MOD 00 31/05/21 PAGE 280

SYMBOL	LEN	VALUE	DEFN	REFERENCES
#@SFOV	001	04C4	0894	
#@SFSY	001	0480	0870	
#@VCNT	001	0002	0794	
#@VLAB	001	0001	0789	
#@VLSD	001	0001	0780	
#@VSFI	001	09A1	0922	7855 8581
#@VTRL	001	0708	0907	
#@WAF1	001	0401	0867	
#@WAR1	001	0400	0866	
#CNDIS	001	0001	0749	
#CNFIG	001	0005	0785	
#CORSV	001	0010	0773	
#DKEXT	001	0002	0756	
#FIGSC	001	0001	0786	
#FMSTD	001	0000	0002	
#HISCT	001	0006	0763	
#HISDX	001	0003	0758	
#HISLN	001	0008	0755	0756 9291 9471
#HISN1	001	0003	0761	
#HISN2	001	0005	0762	
#HISTC	001	0007	0765	
#HISTN	001	0009	0767	
#HISTQ	001	0000	0759	
#HISTR	001	0001	0760	
#HISTS	001	0008	0766	
#HISTV	001	000F	0768	
#HSEND	001	0007	0764	
#HSENT	001	0001	0757	
#IOSDR	001	0019	0784	
#MVSDR	001	000D	0781	
#NEROV	001	009C	0775	
#OBRAD	001	001D	0777	
#PKCNT	001	0002	0742	
#PKMRW	001	002B	0743	
#PKRDD	001	0003	0740	
#PKRTD	001	0003	0739	
#PKRTL	001	0004	0746	
#PKVRD	001	000B	0744	
#PKVWD	001	0007	0745	
#PKWTD	001	0001	0741	
#PTFDA	001	00DC	0795	
#RDWTL	001	0004	0747	
#SDRDK	001	0011	0783	
#VLSDR	001	000C	0779	
#VLTBE	001	0008	0734	
#VOLF1	001	0009	0787	
#VOLNG	001	0006	0732	0734 0756
#VOLOC	001	0005	0733	
#VOLR1	001	0008	0788	
#VTCF1	001	0025	0791	
#VTCF2	001	0027	0793	
#VTCR1	001	0024	0790	
#VTCR2	001	0026	0792	
@\$D1BF	001	0008	2230	7400 7404
@\$D1DC	001	0000	2229	
@\$D1DF	001	001E	2234	

CROSS REFERENCE															
SYMBOL	LEN	VALUE	DEFN	REFERENCES	VER 15, MOD 00 31/05/21 PAGE 281										
@SD1DP	001	0016	2233												
@SD1DV	001	000E	2232												
@SD1E1	001	0000	2223												
@SD1FS	001	000A	2231												
@SD1SW	001	001F	2236	7399											
@SD2AS	001	0002	2241												
@SD2BS	001	0003	2248	7688	7796	8453									
@SD2CB	001	0005	2251	7697	8009*	8010*	8011*	8159	8162*	8449	8496*	8500*	8527*	8540*	
@SD2CF	001	0001	2240	7426*	7427	7571	8141	8648*	8650	8699*					
@SD2CP	001	0005	2249	7693	7755*	7840	7886	8317	8355*	8370	8505	8536*	8677	8677*	
@SD2CS	001	0004	2250	7688	7739	7796	8451	8453	8514*	8688					
@SD2CY	001	0006	2252												
@SD2DA	001	0007	2253												
@SD2DC	001	0000	2245	7428	7585	7589	7617	8154	8654	8685					
@SD2DD	001	0009	2254	7737	8678	8678*									
@SD2EE	001	000F	2257	8688	8688*										
@SD2E1	001	0040	2244	7445	7456										
@SD2FS	001	000B	2255	7736											
@SD2IO	001	0001	2246	7572	7574	7576	7580*	7596	7626	7698	7700	7798	7889	7977	8142
				8144	8146	8150*	8311	8494	8656	8658	8660*	8668*	8684*	8687	8687*
@SD2LC	001	000D	2256	8299	8315	8356*	8369	8497*	8542*	8679	8679*				
@SD2PN	001	000A	2242												
@SD2SF	001	000B	2243												
@SD2VB	001	0002	2247	7694	7841	7887	8163	8318	8371	8506					
@\$L1BF	001	0008	2263												
@\$L1DC	001	0001	2262												
@\$L1DF	001	0008	2265	7378	7404										
@\$L1DP	001	0008	2266												
@\$L1DV	001	0006	2267												
@\$L1E	001	0020	2261	7409											
@\$L1FS	001	0002	2264												
@\$L2AS	001	0001	2273												
@\$L2BS	001	0001	2280												
@\$L2CB	001	0001	2283	7697	7894	7944	7963	8009	8010	8011	8527	8540	8604	8605	
@\$L2CF	001	0002	2272												
@\$L2CP	001	0002	2281	7693	7755	7840	7886	8317	8355	8370	8414	8536	8677		
@\$L2CS	001	0001	2282	7739	8453	8514	8603								
@\$L2DA	001	0002	2284												
@\$L2DC	001	0001	2277	8687											
@\$L2DD	001	0002	2285	77											

## CROSS REFERENCE

SYMBOL LEN VALUE DEFN REFERENCES VER 15, MOD 00 31/05/21 PAGE 282

@\$M2CI	001	0008	2318	7574	8142	8150	8656	8684			
@\$M2CO	001	0004	2319	7572	7580	8144	8658	8684			
@\$M2EF	001	0002	2293	7596	7626	7700	7798	7889	7977	8660	8668
@\$M2FI	001	0080	2307	8146	8494						
@\$M2FO	001	0040	2308	7576							
@\$M2FP	001	0020	2309	7711	8311						
@\$M2FT	001	0010	2312								
@\$M2NS	001	00FF	2292								
@@E001	001	0000	2111	2113							
@@E003	001	0001	2113	2115							
@@E004	001	0002	2115	2117							
@@E005	001	0003	2117	2119							
@@E006	001	0004	2119	2121							
@@E007	001	0005	2121	2123							
@@E008	001	0006	2123	2125							
@@E009	001	0007	2125	2127							
@@E010	001	0008	2127	2129							
@@E011	001	0009	2129	2131							
@@E012	001	000A	2131	2133							
@@E013	001	000B	2133	2135							
@@E014	001	000C	2135	2137							
@@E015	001	000D	2137	2139							
@@E016	001	000E	2139	2141							
@@E017	001	000F	2141	2143							
@@E018	001	0010	2143	2145							
@@E019	001	0011	2145	2147							
@@E020	001	0012	2147	2149							
@@E021	001	0013	2149	2151							
@@E023	001	0014	2151	2153							
@@E024	001	0015	2153	2155							
@@E025	001	0016	2155	2157							
@@E026	001	0017	2157	2159							
@@E027	001	0018	2159	2161							
@@E028	001	0019	2161	2163							
@@E029	001	001A	2163	2165							
@@E030	001	001B	2165	2167							
@@E031	001	001C	2167	2169							
@@E032	001	001D	2169	2171							
@@E035	001	001E	2171	2173							
@@E036	001	001F	2173	2175							
@@E037	001	0020	2175	2177							
@@E038	001	0021	2177	2179							
@@E039	001	0022	2179	2181							
@@E040	001	0023	2181	2183							
@@E041	001	0024	2183	2185							
@@E042	001	0025	2185	2187							
@@E043	001	0026	2187	2189							
@@E044	001	0027	2189	2191							
@@E045	001	0028	2191	2193							
@@E046	001	0029	2193	2195							
@@E060	001	002A	2195	2197							
@@E080	001	002B	2197								
@@E100	001	0000	1583	1585							
@@E101	001	0001	1585	1587							
@@E102	001	0002	1587	1589							
@@E103	001	0003	1589	1591							

## CROSS REFERENCE

VER 15, MOD 00 31/05/21 PAGE 283

SYMBOL	LEN	VALUE	DEFN	REFERENCES
@@E110	001	0004	1591	1593
@@E112	001	0005	1593	1595
@@E113	001	0006	1595	1597
@@E114	001	0007	1597	1599
@@E115	001	0008	1599	1601
@@E116	001	0009	1601	1603
@@E117	001	000A	1603	1605
@@E120	001	000B	1605	1607
@@E122	001	000C	1607	1609
@@E123	001	000D	1609	1611
@@E124	001	000E	1611	1613
@@E129	001	000F	1613	1615
@@E130	001	0010	1615	1617
@@E131	001	0011	1617	1619
@@E133	001	0012	1619	1621
@@E134	001	0013	1621	1623
@@E135	001	0014	1623	1625
@@E136	001	0015	1625	1627
@@E137	001	0016	1627	1629
@@E138	001	0017	1629	1631
@@E139	001	0018	1631	1633
@@E142	001	0019	1633	1635
@@E143	001	001A	1635	1637
@@E150	001	001B	1637	1639
@@E151	001	001C	1639	1641
@@E160	001	001D	1641	1643
@@E162	001	001E	1643	1645
@@E163	001	001F	1645	1647
@@E164	001	0020	1647	1649
@@E200	001	0021	1649	1651
@@E205	001	0022	1651	1653
@@E210	001	0023	1653	1655
@@E211	001	0024	1655	1657
@@E212	001	0025	1657	1659
@@E213	001	0026	1659	1661
@@E215	001	0027	1661	1663
@@E216	001	0028	1663	1665
@@E217	001	0029	1665	1667
@@E220	001	002A	1667	1669
@@E221	001	002B	1669	1671
@@E222	001	002C	1671	1673
@@E223	001	002D	1673	1675
@@E225	001	002E	1675	1677
@@E226	001	002F	1677	1679
@@E227	001	0030	1679	1681
@@E228	001	0031	1681	1683
@@E229	001	0032	1683	1685
@@E230	001	0033	1685	1687
@@E232	001	0034	1687	1689
@@E234	001	0035	1689	1691
@@E237	001	0036	1691	1693
@@E240	001	0037	1693	1695
@@E241	001	0038	1695	1697 2708
@@E242	001	0039	1697	1699
@@E248	001	003A	1699	1701
@@E249	001	003B	1701	1703

## CROSS REFERENCE

VER 15, MOD 00 31/05/21 PAGE 284

SYMBOL	LEN	VALUE	DEFN	REFERENCES
@@E250	001	003C	1703	1705
@@E251	001	003D	1705	1707
@@E252	001	003E	1707	1709
@@E253	001	003F	1709	1711
@@E254	001	0040	1711	1713
@@E255	001	0041	1713	1715
@@E256	001	0042	1715	1717
@@E300	001	0043	1717	1719
@@E301	001	0044	1719	1721
@@E302	001	0045	1721	1723
@@E303	001	0046	1723	1725
@@E304	001	0047	1725	1727
@@E305	001	0048	1727	1729
@@E308	001	0049	1729	1731
@@E310	001	004A	1731	1733
@@E315	001	004B	1733	1735
@@E316	001	004C	1735	1737
@@E320	001	004D	1737	1739
@@E325	001	004E	1739	1741
@@E330	001	004F	1741	1743
@@E335	001	0050	1743	1745
@@E338	001	0051	1745	1747
@@E340	001	0052	1747	1749
@@E350	001	0053	1749	1751
@@E351	001	0054	1751	1753
@@E352	001	0055	1753	1755
@@E360	001	0056	1755	1757
@@E361	001	0057	1757	1759
@@E362	001	0058	1759	1761
@@E371	001	0059	1761	1763
@@E380	001	005A	1763	1765
@@E390	001	005B	1765	1767
@@E400	001	005C	1767	1769
@@E410	001	005D	1769	1771
@@E415	001	005E	1771	1773
@@E417	001	005F	1773	1775
@@E420	001	0060	1775	1777
@@E430	001	0061	1777	1779
@@E432	001	0062	1779	1781
@@E433	001	0063	1781	1783
@@E450	001	0064	1783	1785
@@E451	001	0065	1785	1787
@@E460	001	0066	1787	1789
@@E461	001	0067	1789	1791
@@E464	001	0068	1791	1793
@@E465	001	0069	1793	1795
@@E466	001	006A	1795	1797
@@E467	001	006B	1797	1799
@@E469	001	006C	1799	1801
@@E470	001	006D	1801	1803
@@E471	001	006E	1803	1805
@@E473	001	006F	1805	1807
@@E474	001	0070	1807	1809
@@E475	001	0071	1809	1811
@@E476	001	0072	1811	1813
@@E477	001	0073	1813	1815

## CROSS REFERENCE

VER 15, MOD 00 31/05/21 PAGE 285

SYMBOL	LEN	VALUE	DEFN	REFERENCES
@@E478	001	0074	1815	1817
@@E479	001	0075	1817	1819
@@E480	001	0076	1819	1821
@@E481	001	0077	1821	1823
@@E482	001	0078	1823	1825
@@E483	001	0079	1825	1827
@@E484	001	007A	1827	1829
@@E485	001	007B	1829	1831
@@E486	001	007C	1831	1833
@@E487	001	007D	1833	1835
@@E488	001	007E	1835	1837
@@E489	001	007F	1837	1839
@@E490	001	0080	1839	1841
@@E491	001	0081	1841	1843
@@E492	001	0082	1843	1845
@@E493	001	0083	1845	1847
@@E494	001	0084	1847	1849
@@E495	001	0085	1849	1851
@@E496	001	0086	1851	1853
@@E497	001	0087	1853	1855
@@E498	001	0088	1855	1857
@@E500	001	0089	1857	1859
@@E501	001	008A	1859	1861
@@E530	001	008B	1861	1863
@@E531	001	008C	1863	1865
@@E535	001	008D	1865	1867
@@E540	001	008E	1867	1869
@@E541	001	008F	1869	1871
@@E542	001	0090	1871	1873
@@E543	001	0091	1873	1875
@@E544	001	0092	1875	1877
@@E545	001	0093	1877	1879
@@E546	001	0094	1879	1881
@@E547	001	0095	1881	1883
@@E548	001	FFFF	2087	
@@E549	001	0096	1883	1885
@@E550	001	0097	1885	1887
@@E551	001	0098	1887	1889
@@E552	001	0099	1889	1891
@@E553	001	009A	1891	1893
@@E554	001	009B	1893	1895
@@E555	001	009C	1895	1897
@@E556	001	009D	1897	1899
@@E558	001	009E	1899	1901
@@E570	001	009F	1901	1903
@@E571	001	00A0	1903	1905
@@E572	001	00A1	1905	1907
@@E573	001	00A2	1907	1909
@@E574	001	00A3	1909	1911
@@E575	001	FFFF	2089	
@@E578	001	00A4	1911	1913
@@E579	001	FFFF	2091	
@@E580	001	FFFF	2093	
@@E585	001	00A5	1913	1915
@@E595	001	FFFF	2095	
@@E597	001	FFFF	2097	



## CROSS REFERENCE

VER 15, MOD 00 31/05/21 PAGE 286

SYMBOL	LEN	VALUE	DEFN	REFERENCES
@@E598	001	FFFF	2099	
@@E600	001	00A6	1915	1917
@@E601	001	00A7	1917	1919
@@E602	001	00A8	1919	1921
@@E603	001	00A9	1921	1923
@@E604	001	00AA	1923	1925
@@E606	001	00AB	1925	1927
@@E607	001	00AC	1927	1929
@@E608	001	00AD	1929	1931
@@E609	001	00AE	1931	1933
@@E610	001	00AF	1933	1935
@@E611	001	00B0	1935	1937
@@E612	001	00B1	1937	1939
@@E613	001	00B2	1939	1941
@@E614	001	00B3	1941	1943
@@E700	001	00B4	1943	1945
@@E701	001	00B5	1945	1947 7291
@@E710	001	00B6	1947	1949 7380 7384 7402
@@E712	001	00B7	1949	1951 7578 8148
@@E713	001	00B8	1951	1953
@@E714	001	00B9	1953	1955 8404
@@E715	001	00BA	1955	1957 7742
@@E716	001	00BB	1957	1959
@@E717	001	00BC	1959	1961
@@E718	001	00BD	1961	1963 8256 0552
@@E720	001	00BE	1963	1965 1690
@@E721	001	00BF	1965	1967 1750
@@E723	001	00C0	1967	1969
@@E724	001	00C1	1969	1971
@@E725	001	00C2	1971	1973
@@E726	001	00C3	1973	1975
@@E727	001	00C4	1975	1977
@@E728	001	00C5	1977	1979 7257
@@E729	001	00C6	1979	1981
@@E730	001	00C7	1981	1983
@@E732	001	00C8	1983	1985 7267
@@E752	001	00C9	1985	1987
@@E753	001	00CA	1987	1989
@@E754	001	00CB	1989	1991
@@E755	001	00CC	1991	1993
@@E756	001	00CD	1993	1995
@@E757	001	00CE	1995	1997
@@E758	001	00CF	1997	1999
@@E759	001	00D0	1999	2001
@@E760	001	00D1	2001	2003
@@E761	001	00D2	2003	2005
@@E762	001	00D3	2005	2007
@@E763	001	00D4	2007	2009
@@E764	001	00D5	2009	2011
@@E765	001	00D6	2011	2013
@@E766	001	00D7	2013	2015
@@E767	001	00D8	2015	2017
@@E768	001	00D9	2017	2019
@@E769	001	00DA	2019	2021
@@E770	001	00DB	2021	2023
@@E771	001	00DC	2023	2025

## CROSS REFERENCE

VER 15, MOD 00 31/05/21 PAGE 287

SYMBOL	LEN	VALUE	DEFN	REFERENCES
@@E772	001	00DD	2025	2027
@@E773	001	00DE	2027	2029
@@E774	001	00DF	2029	2031 5159
@@E775	001	00E0	2031	2033 5363
@@E776	001	00E1	2033	2035 4805
@@E777	001	00E2	2035	2037 4127
@@E778	001	00E3	2037	2039 4107
@@E779	001	00E4	2039	2041 4118
@@E780	001	00E5	2041	2043 6192
@@E781	001	00E6	2043	2045
@@E782	001	00E7	2045	2047
@@E783	001	00E8	2047	2049
@@E784	001	00E9	2049	2051
@@E785	001	00EA	2051	2053
@@E786	001	00EB	2053	2055
@@E790	001	00EC	2055	2057 5123
@@E791	001	00ED	2057	2059 5234
@@E792	001	00EE	2059	2061
@@E793	001	00EF	2061	2063 6204
@@E794	001	00F0	2063	2065 6195
@@E795	001	00F1	2065	2067 5612
@@E796	001	00F2	2067	2069 5598
@@E797	001	00F3	2069	2071
@@E798	001	00F4	2071	2073
@@E800	001	FFFF	2101	
@@E801	001	FFFF	2103	
@@E802	001	FFFF	2105	
@@E803	001	FFFF	2107	
@@E804	001	FFFF	2109	
@@E900	001	00F5	2073	2075 2704
@@E901	001	00F6	2075	2077 2706
@@E902	001	00F7	2077	2079 2705
@@E903	001	00F8	2079	2081 2707
@@E905	001	00F9	2081	2083
@@E906	001	00FA	2083	2085
@@E910	001	00FB	2085	2703
@@M250	001	2D00	0359	0168
@@M251	001	2D04	0363	0177
@@M256	001	2D08	0367	0189
@@M257	001	2D0C	0371	
@@M258	001	2D10	0375	
@@M259	001	2D14	0379	
@@M260	001	2D18	0383	
@@T250	001	2D1C	0387	0361
@@T251	001	2D24	0389	0365
@@T256	001	2D2D	0391	0369
@@T257	001	2D52	0393	0373
@@T258	001	2D75	0395	0377
@@T259	001	2D8C	0397	0381
@@T260	001	2DAC	0399	0385
@ALTFL	001	0001	0963	
@ARR	001	0008	0017	6799 7764 9140 9147 9161 9180 0231 0584 0627 0802 1002 1211 1444 1464 2643 2660 2917 2935
@ASIGN	001	007C	0072	
@ASTER	001	005C	0070	
@BCRDL	001	0050	0089	7947

CROSS REFERENCE																		
SYMBOL	LEN	VALUE	DEFN	REFERENCES										VER 15, MOD 00			31/05/21	PAGE 288
@BE	001	0081	0044	0487														
@BF	001	0090	0053															
@BH	001	0084	0042															
@BKSPC	001	0010	1060															
@BL	001	0082	0043															
@BLANK	001	0040	0066	7370	8186													
@BM	001	0082	0055															
@BNE	001	0001	0047	8172	8173	0486												
@BNH	001	0004	0045															
@BNL	001	0002	0046															
@BNM	001	0002	0058															
@BNOL	001	0020	0051															
@BNOZ	001	0008	0050															
@BNP	001	0004	0057															
@BNZ	001	0001	0059															
@BOL	001	00A0	0049															
@BOZ	001	0088	0048															
@BP	001	0084	0054															
@BR	001	0001	0014	4098	4106	4106	4107	4108	4117	4117	4118	4119	4120	4126	4127			
				4133	4135	4137	4174	4175	4176	4177	4179	4180	4180	4185	4186			
				4211	4212	4243	4252	4253	4257	4266	4266	4267	4272	4272	4278			
				4279	4296	4300	4300	4304	4305	4309	4320	4326	4326	4331	4331			
				4332	4332	4333	4478	4486	4487	4490	4503	4510	4619	4626	4628			
				4642	4644	4644	4651	4652	4655	4666	4667	4789	4804	4806	4820			
				4826	4827	4831	4832	4833	4834	4835	4835	4836	4845	4846	4849			
				4853	4854	4855	4856	4856	4857	4883	4883	4884	4894	4904	4905			
				4906	4908	4908	4915	4915	4916	4925	4927	4935	4935	4938	4980			
				4983	4985	4986	5098	5112	5117	5122	5124	5129	5131	5132	5141			
				5143	5144	5144	5145	5146	5148	5150	5151	5152	5158	5160	5167			
				5169	5174	5174	5175	5179	5180	5189	5191	5192	5200	5201	5202			
				5207	5208	5209	5219	5225	5233	5235	5239	5356	5362	5364	5366			
				5374	5380	5384	5397	5406	5407	5407	5408	5409	5410	5411	5412			
				5421	5422	5423	5423	5424	5425	5430	5431	5438	5439	5439	5440			
				5440	5442	5583	5590	5595	5598	5599	5610	5611	5612	5614	5615			
				5622	5624	5628	5634	5646	5650	5651	5652	5653	5654	5655	5655			
				5656	5657	5668	5684	5685	5686	5687	5688	5689	5692	5693	5698			
				5699	5700	5729	5739	5740	5743	5744	5744	5745	5745	5750	5750			
				5751	5752	5758	5760	5761	5761	5762	5763	5891	5909	5918	5919			
				5919	5920	6034	6041	6048	6050	6057	6057	6058	6185	6191	6192			
				6193	6195	6196	6197	6198	6204	6206	6208	6209	6213	6222	6336			
				6342	6343	6350	6351	6352	6356	6357	6362	6387	6388	6389	6395			
				6421	6426	6426	6427	6428	6435	6435	6436	6438	6445	6446	6447			
				6448	6449	6451	6452											

## CROSS REFERENCE

SYMBOL LEN VALUE DEFN REFERENCES VER 15, MOD 00 31/05/21 PAGE 289

	7920	7920	7921	7924	7925	7925	7926	7926	7927	7939	7943	7943
	7944	7944	7947	7948	7964	7978	7979	7983	7983	7988	7995	7995
	7999	7999	8000	8000	8008	8009	8010	8011	8135	8137	8161	8163
	8168	8173	8181	8182	8195	8208	8215	8216	8222	8224	8229	8233
	8234	8238	8240	8252	8257	8266	8287	8299	8309	8310	8313	8314
	8314	8315	8316	8317	8318	8326	8329	8331	8331	8332	8332	8334
	8334	8336	8337	8337	8338	8338	8340	8342	8344	8344	8355	8356
	8357	8357	8369	8370	8371	8374	8374	8375	8408	8435	8459	8459
	8460	8467	8467	8468	8470	8470	8471	8482*	8497	8500	8504	8505
	8506	8513	8514	8515	8519	8520	8521	8527	8528	8528	8533	8535
	8536	8537	8539	8540	8541	8541	8542	8550	8555	8561	8562	8567
	8570	8620	8626	8627	8633	8643	8661	8686	8694	8694	8696	8699
	8700	8704	8706	8708	8709	8727	8731	8732	8742	8910	8914	8915
	8915	8916	8920	8921	8922	8923	8924	8925	8926	8927	8928	8928
	8929	8930	8931	8932	8933	8936	8938	8942	8943	8947	8949	8951
	9037	9038	9039	9039	9040	9101	9103	9103	9109	9112	9113	9116
	9118	9119	9120	9122	9122	9123	9125	9126	9139	9140	9141	9141
	9142	9143	9147	9148	9148	9149	9149	9150	9151	9161	9162	9163
	9168	9174	9180	9181	9181	9195	9196	9197	9199	9201	9203	9204
	9206	9208	9209	9215	9219	9220	9222	9222	9223	9223	9225	9235
	9235	9238	9239	9239	9240	9249	9252	9254	9254	9255	9261	9262
	9262	9263	9266	9267	9280	9281	9282	9291	9297	9312	9314	9316
	9319	9321	9326	9328	9329	9331	9336	9337	9338	9340	9342	9342
	9344	9345	9345	9346	9346	9348	9349	9349	9352	9352	9353	9355
	9357	9359	9361	9361	9362	9362	9363	9363	9365	9381	9381	9382
	9384	9385	9386	9387	9391	9399	9401	9402	9406	9407	9460	9476
	9486	9486	9488	9491	9491	9493	9493	9495	9496	9498	9499	9500
	9500	9502	9507	9510	9515	9520	9534	9962	9976	0006	0009	0010*
	0029	0053	0095	0109	0118	0120	0124	0126	0127	0141	0142	0142
	0143	0143	0144	0145	0147	0147	0148	0148	0150	0150	0151	0152
	0152	0153	0157	0161	0164	0169	0171	0172	0178	0180	0181	0182
	0183	0188	0191	0192	0193	0199	0222	0226	0231	0235	0235	0236
	0240	0245	0252	0340	0439	0455	0456	0457	0458	0471	0472	0483
	0484	0487	0488	0490	0491	0492	0500	0505	0506	0507	0522	0526
	0527	0529	0543	0559	0584	0593	0593	0600	0601	0601	0603	0604
	0606	0607	0609	0609	0610	0612	0627	0631	0690	0696	0697	0715
	0727	0752	0755	0758	0769	0769	0771	0774	0774	0775	0776	0777
	0778	0782	0782	0784	0784	0786	0786	0802	0806	0859	0865	0865
	0866	0867	0868	0878	0879	0884	0885	0887	0887	0888	0889	0901
	0901	0902	0903	0908	0910	0911	0930	0931	0936	0937	0946	0947
	0949	0949	0950	0951	0955	0956	0957	0957	0958	0958	0959	0959
	0960	0960	0961	0961	0962	0967	0967	0969	0969	0979	0979	0981
	0981	1002	1006	1010	1066	1078	1080	1090	1091	1096	1098	1110*
	1125	1126	1126	1127	1141	1142	1144	1144	1145	1146	1146	1147
	1152	1156	1158	1172	1173	1174	1178	1180	1181	1193	1195	1199*
	1200	1211	1215	1299	1305	1306	1306	1307	1308	1319	1320	1324
	1331	1331	1332	1333	1337	1346	1346	1348	1350	1351	1351	1353
	1354	1360	1361	1365	1367	1380	1381	1386	1387	1387	1388	1389
	1390	1395	1395	1396	1396	1397	1402	1402	1404	1404	1409	1411
	1412	1416	1418	1444	1446	1451	1464	1468	1679	1697	1704	1705
	1706	1707	1712	1732	1756	1769	1771	2027	2038	2041	2046	2056
	2059	2061	2068	2068	2080	2081	2081	2082	2087	2087	2088	2089
	2094	2094	2104	2110	2126	2128	2129	2129	2132	2134	2134	2141
	2144	2158	2159	2159	2167	2168	2177	2270	2277	2280	2285	2287
	2295	2296	2303	2304	2304	2305	2305	2310	2311	2311	2313	2317
	2333	2339	2340	2340	2342	2348	2352	2352	2354	2355	2355	2361

## CROSS REFERENCE

SYMBOL   LEN   VALUE   DEFN   REFERENCES   VER 15, MOD 00   31/05/21   PAGE 290

				2362	2363	2367	2367	2368	2369	2371	2371	2372	2372	2373	2373
				2374	2378	2380	2380	2381	2385	2386	2386	2387	2443	2454	2458
				2459	2460	2465	2465	2467	2471	2482	2484	2486	2495	2508	2510
				2510	2512	2512	2513	2514	2516	2516	2524	2525	2531	2532	2532
				2543	2548	2557	2566	2570	2570	2571	2572	2572	2584	2593	2594
				2603	2604	2614	2618	2619	2643	2647	2648	2653	2660	2728	2734
				2738	2742	2743	2744	2749	2749	2751	2755	2766	2778	2791	2793
				2793	2795	2795	2796	2797	2799	2799	2807	2808	2814	2815	2815
				2826	2831	2840	2849	2853	2853	2854	2855	2855	2867	2876	2877
				2886	2887	2897	2901	2903	2917	2921	2922	2927	2935	2937	3130
				3140	3146	3151	3152	3152	3159	3160	3160	3161	3174	3194	3196
				3196	3197	3197	3198	3208	3209	3209	3216	3216	3217	3217	3218
				3218	3219	3220	3220	3230	3230	3234	3234	3235	3235	3239	3241
				3245	3247	3257	3257	3258	3258	3262	3263	3270	3270	3271	3427
				3431	3438	3440	3442	3451	3466	3467	3485	3488	3495	3496	3497
				3528	3529	3530	3531	3533	3540	3570	3584	3619	3625	3629	3634
				3635	3646	3647	3648	3648	3649	3649	3650	3650	3651	3651	3652
				3652	3657	3663	3665	3675	3684	3686	3687	3690	3692	3693	3694
				3695	3697	3731	3733	3737	3739	3739	3741	3742	3742	3744	3746
				3754	3755	3764	3765	3766	3766	3768	3772	3773	3776	3784	3786
				3786	3789	3789	3790	3791	3793	3798	3799	3800			

@BT   001   0010   0052

@BZ   001   0081   0056

@BZ37B   001   00F2   1073

@B1   001   0001   0064

4237	4353	4678	4953	5256	5471	5777	6048	6492	7399	7640	7646
7717	7723	7734	7748	7749	7750	7756	7758	7921	7994	8202	8210
8228	8390	8500	0144	0151	0629	0748	0753	0804	1004	1131	1134

@CADDR   001   0002   0143

1213	1466	2061	2148	2368	3665						
2657	2684	3501	3502	3503	7038	7039	7040	7041	7042	7043	7044
7045	7046	7247	7262	7301	7310	7335	7362	7584	7587	7591	7607
7654	7670	7746	7767	7793	7811	7832	7847	8025	8274	8326	8422
8424	8425	8426	8428	8481	8562	8928	9072	9103	9148	9149	9150
9168	9181	9235	9239	9254	9259	9262	9314	9316	9319	9346	9363
9420	9493	9496	9538	9978	0006	0009	0012	0063	0235	0248	0252
0274	0361	0365	0369	0373	0377	0381	0385	0490	0500	0601	1788
1789	1790	2177	2201	2449	2460	2486	2734	2744	2939	3152	3257

@CARDL   001   0060   0088

@CC37B   001   0000   1069

@CD37B   001   00F0   1087

@CHARA   001   00C1   0073

@CHARF   001   00C6   0074

@CHARR   001   00D9   0075

@CHARZ   001   00E9   0076

@CKY01   001   0001   1021

@CKY02   001   0002   1022

@CKY03   001   0003   1023

@CKY04   001   0004   1024

@CKY05   001   0005   1025

@CKY06   001   0006   1026

@CKY07   001   0007   1027

@CKY08   001   0008   1028

@CKY09   001   0009   1029

@CKY10   001   000A   1030

@CKY11   001   000B   1031

@CKY12   001   000C   1032

0645	7950	7965*	7966	7966	7966*	8171	8180*
------	------	-------	------	------	-------	------	-------

## CROSS REFERENCE

VER 15, MOD 00 31/05/21 PAGE 291

SYMBOL	LEN	VALUE	DEFN	REFERENCES
@CKY13	001	000D	1033	
@CKY14	001	000E	1034	
@CKY15	001	000F	1035	
@CKY16	001	0010	1036	
@CLOFF	001	0010	0095	
@CLON	001	0011	0094	
@CMLON	001	0001	1039	
@CMOFF	001	0000	1038	
@COMMA	001	006B	0067	8189 8219
@CPLUS	001	004E	0080	
@CP37B	001	0004	1100	
@CRERR	001	0090	1055	
@CRPRY	001	0004	1059	
@CRTDS	001	0092	1052	
@CRTQ	001	0090	1054	
@CURSR	001	0040	1056	
@DADDR	001	0002	0141	3216 3217 3291 3299
@DBFR1	001	0004	0130	3258*
@DBFR2	001	0005	0131	3257*
@DBUSY	001	0002	0957	
@DCALK	001	0001	0082	
@DCBCY	001	0009	0116	3330
@DCBT1	001	0050	0118	3333
@DCFLN	001	0004	0941	
@DCNT	001	0003	0129	
@DCRID	001	0001	0955	
@DCST1	001	0040	0117	3331
@DCTRL	001	0000	0126	3140* 3146* 3174
@DCTRW	001	0000	0954	
@DCWID	001	0001	0951	
@DCYL	001	0001	0127	3208* 3230*
@DCYMV	001	0001	0942	
@DD2	001	0003	0031	5745 5745* 5750* 6426 6426* 6435* 6762* 1705* 3561
@DEFLG	001	0002	0964	
@DERCE	001	0020	0994	
@DERD2	001	0008	0986	
@DEREQ	001	0010	0985	
@DERIN	001	0040	0983	
@DERMA	001	0020	0984	
@DERNR	001	0004	0987	
@DERR	001	0000	0958	
@DERSC	001	0001	0989	
@DERTC	001	0002	0988	
@DFCR	001	0006	0944	
@DFDR	001	0004	0945	
@DGET	001	0001	0135	7864 8469 8592 3146 3174
@DHARD	001	0000	0972	
@DLNCT	001	000F	1058	
@DLNLG	001	0040	1057	2738
@DOLAR	001	005B	0069	
@DOP2	001	0004	0029	8004 9443 0647 3790*
@DPLNG	001	0006	0133	
@DPOS	001	0000	0134	
@DPUT	001	0002	0136	7812 7854 8580 3140
@DREAD	001	0001	0948	
@DSAD	001	0002	0128	3209* 3218* 3220* 3234 3234* 3235 3235* 3241* 3247*

## CROSS REFERENCE

VER 15, MOD 00 31/05/21 PAGE 292

SYMBOL	LEN	VALUE	DEFN	REFERENCES
@DSBCY	001	0004	0107	3268
@DSBSY	001	0092	1053	
@DSCS1	001	0000	0108	3269
@DSEEK	001	0000	0947	
@DSIVF	001	0003	0139	
@DSPIN	001	0002	0132	
@DTRSZ	001	0018	0086	3290
@DUNSF	001	0080	0990	
@DVBCY	001	0007	0109	3327
@DVERY	001	0003	0953	
@DVRFY	001	0031	0137	
@DVST1	001	0002	0959	
@DVST2	001	0003	0960	
@DWAIT	001	00FF	0138	
@DWBCY	001	0005	0104	3324
@DWBIT	001	0002	0949	
@DWSIZ	001	00C0	0106	
@DWTB1	001	0003	0105	3325
@DZERO	001	00F0	0065	4132 4171 4250 4488 4635 4865 4926 5111 5116 5128 5365 5379
				5386 5406 5662 5899 6190 6483
@D1	001	0002	0027	6764* 8002 8181* 8215 8216* 8222* 8233* 9396 9507* 1127* 1141 1146*
				1225 1307* 1348 1351* 1479 1707* 2082* 2087* 2088 2094 2128* 2129*
				2134 2303* 2304* 2305* 2310* 2311* 2317 2339* 2340* 2348 2385* 2386
				2386* 2471* 2755* 3151* 3159* 3160* 3194* 3196* 3197 3209 3258 3270*
				3438* 3488* 3560 3629* 3792* 3793*
@EOF	001	001C	0078	7703 8321 8663
@EOFTC	001	0075	0163	
@EOS	001	001E	0077	3340 8180 8192 8204 9253
@ER37B	001	00F0	1074	
@FDDBC	001	0000	0196	
@FDE1	001	000C	0201	
@FDFNA	001	000B	0199	
@FDHLN	001	0002	0209	
@FDLNC	001	0002	0194	
@FDNSC	001	0003	0211	
@FDSD	001	0000	0207	
@FLACE	001	0009	0198	
@FLDBC	001	0001	0197	
@FLDIN	001	0012	1046	
@FLENT	001	0004	0202	
@FLFNA	001	0002	0200	
@FLHLN	001	0002	0210	
@FLLNC	001	0002	0195	
@FLNSC	001	0001	0212	
@FLSD	001	0001	0208	
@HCEPK	001	003C	0829	
@HCOPS	001	001C	0836	
@HCOPY	001	081C	0831	
@HCRHE	001	7858	0852	
@HDNRY	001	1008	0817	
@HDRHE	001	7854	0850	
@HDRLN	001	0007	0093	0673
@HDRV1	001	7840	0842	
@HDRV2	001	7844	0844	
@HDTRD	001	1040	0813	
@HDTRJ	001	1010	0815	



CROSS REFERENCE																			
SYMBOL	LEN	VALUE	DEFN	REFERENCES													VER 15, MOD 00	31/05/21	PAGE 293
@HERPG	001	087C	0819																
@HFEHT	001	0804	0834																
@HIPLE	001	006C	0826																
@HKBER	001	2040	0809	9295															
@HKBHE	001	7848	0846																
@HLOGE	001	1844	0821																
@HPRER	001	0070	0811	9475															
@HPRHE	001	784C	0848																
@HSTAD	001	0009	0970																
@HSTEN	001	0007	0969																
@HSTPE	001	0006	0968	9476*	9526*														
@HSTQR	001	0001	0966																
@HSTSN	001	0005	0967																
@HSTVI	001	000F	0971																
@HUNSF	001	1850	0824																
@IAR	001	0010	0018																
@ID37B	001	0040	1110																
@INDEX	001	0001	0157	0158	8737	9359	9463	9466	9505	3458	3460	3495							
@INST3	001	0003	0033	7951	9370	9398	3505	3518											
@INST4	001	0004	0034	6426	6435	6468	6476	6488	6491	6814	2201								
@INST5	001	0005	0035																
@INST6	001	0006	0036																
@IP37B	001	00C0	1109																
@I1IAR	001	00C0	0021	8943*	9101*														
@KCMDK	001	0020	1020																
@KELOK	001	001B	1019																
@KENAB	001	001E	1017	8628	8705	8944	9052	9170	9173	9992	0056	0257	3743						
@KEXIT	001	001F	1018																
@KEYBD	001	0010	1037	8944	9034	9038	9051	9102	9106	9108	9183	9221	9225						
@KFUNK	001	0010	1040	9071	9116														
@KHARD	001	0011	1045																
@KLEAR	001	000D	1041																
@LINSZ	001	00F4	0085	0647															
@LO37B	001	00F0	1078																
@MAPEN	001	0005	0090																
@MINCR	001	2000	0084																
@MINUS	001	0060	0081																
@NOP	001	0080	0041	3843	4119	5202	6356	6813	6877	7721	8161	8195	8310	8384	8394				
				8526	8539	8560	8567	8626	8627	8672	8704	9121	9214	9232	9292				
				9369	9371	9395	9518	0003	0008	0453	0458	0471	0488	0866	0885				
				0908	0930	1010	1324	1365	1446	3450	3509	3517	3528	3529	3754				
@NORFL	001	0000	0965																
@NTRDY	001	00A0	1102																
@NUMBR	001	007B	0071																
@OPD2	001	0004	0030	6427*	6468*	6469*	6476*	9122*	2168*										
@OP1	001	0003	0028	4272*	5918*	5919*	5920*	6799*	7391*	7424*	7659	7730	7752	7764*	7795*				
				7822*	7823*	7885*	8008	8224*	8347	8354	8483	8485	8661*	8698	8731*				
				8915*	8916*	9140*	9147*	9149*	9150*	9161*	9180*	9316*	0009*	0164*	0188				
				0231*	0235*	0236*	0240	0252	0277	0584*	0627*	0802*	1002*	1078*	1090*				
				1156*	1174*	1211*	1416*	1444*	1464*	1704*	2486*	2643*	2660*	2734*	291				

## CROSS REFERENCE

VER 15, MOD 00 31/05/21 PAGE 294

SYMBOL	LEN	VALUE	DEFN	REFERENCES
@PCTRL	001	0000	0150	9231* 9233* 9326 9329 9344* 9353 9359* 9498* 9502* 2690 2969 3451 3485 3798*
@PCYL	001	0001	0937	
@PC37B	001	00F2	1094	
@PDAR	001	00E4	1003	9336* 3624*
@PDATA	001	0003	0152	9063 9148* 9181 9258* 9259* 9336 9346* 9493* 0120* 0181* 0226* 2692 2971 3739 3789* 3790
@PD37B	001	0080	1108	
@PERR	001	00E0	1010	8742 9387 9389 9391 3570 3572
@PFLAG	001	0000	0936	
@PFORM	001	00E1	1008	9385
@PGCSZ	001	0020	0083	0084
@PLITE	001	00E2	1009	9386* 9406*
@PLNGH	001	0004	1000	3737 3737 3737*
@PMGCK	001	0020	1011	9477
@PN37B	001	00F0	1093	
@PPLNG	001	0004	0149	0189 0190
@PRCNT	001	0001	0151	9328* 9337* 9338* 9340* 9342 9342* 9345 9348 9349* 9352* 2691 2970 3466 3467 3733 3742 3742* 3799*
@PRETR	001	00C0	0155	0300 0367 0371 0375 0379 0383 2548 2831 3485 3772
@PRINT	001	0040	0153	0155 9326 9483 9503 9527 0307 0314 0359 0363 2458 2742 3433 3451 3532 3776 3798
@PRITY	001	0080	1044	9113
@PSAD	001	0002	0938	
@PSIOQ	001	00E0	1006	9366 9539
@PSIOR	001	0000	1005	9366 9540
@PSNSQ	001	00E2	1012	9468
@PSR	001	0004	0016	
@PWAIT	001	00FF	0159	7417
@P1IAR	001	0020	0019	9037* 9109*
@P2IAR	001	0040	0020	
@Q	001	0001	0025	4107* 4118* 4119* 4126* 4127* 5202* 5208* 5598* 5612* 6048* 6057 6057* 6192* 6195* 6204* 6351* 6356* 6812 6875* 6877* 7417* 7444* 7753 7921* 7925 7925* 7949 8003 8161* 8173* 8182* 8195* 8310* 8313* 8338* 8535* 8539* 8567* 8570* 8626* 8627* 8633* 8704* 8708* 8709* 9115* 9121* 9214* 9234* 9292* 9368 9382* 9394 9518* 0144* 0148 0148* 0151* 0455* 0456* 0457* 0458* 0471* 0472* 0483* 0484* 0487* 0488* 0559* 0603* 0606* 0612* 0696* 0697* 0866* 0867* 0868* 0884* 0885* 0908* 0930* 0936* 1010* 1324* 1337* 1365 1446* 1451* 2056* 2059* 2080* 2081* 2158* 2159* 2244 2295* 2296* 2368* 2372 2372* 2680 2681 2701 2959 2960 3431* 3439* 3450* 3496* 3503 3506* 3516 3528* 3529* 3619* 3754*
@RD37B	001	00F1	1088	
@REGL	001	0002	0013	5255 7329 9039
@RETRN	001	0080	0154	0155 9344 9353 9429 9498 9502 0321 2696 2975
@RLDWN	001	004F	0160	
@RTCNT	001	0003	1002	9355* 9357* 9361*
@RTRNC	001	0080	0162	0322 2697 2976
@RT37B	001	0005	1101	
@SBLN	001	0005	0171	
@SBLNL	001	0002	0185	
@SCTS	001	0100	0101	
@SDFLN	001	0007	0091	
@SDF0	001	0000	0167	8510
@SDF1	001	0001	0168	8528* 8536 8541* 8542
@SDF2	001	0002	0169	8521
@SDF3	001	0003	0170	

## CROSS REFERENCE

VER 15, MOD 00 31/05/21 PAGE 295

SYMBOL	LEN	VALUE	DEFN	REFERENCES
@SECCY	001	0030	0087	
@SIST	001	0001	0182	
@SKCTL	001	0000	0952	
@SLASH	001	0061	0068	
@SLAST	001	0002	0184	8521
@SMIDL	001	0003	0183	
@SNSB0	001	0000	0976	
@SNSB1	001	0001	0977	
@SNSB2	001	0002	0978	
@SNSB3	001	0003	0979	
@SNULL	001	0080	0174	8510
@SN37B	001	00F2	1082	
@SONLY	001	0000	0181	
@SPINA	001	00A0	0961	
@SPINB	001	00B0	0962	
@STEXT	001	0007	0173	8608
@STYPE	001	0006	0172	8533
@SYCNT	001	0002	1001	9488* 9499 9500* 3741*
@SYLVL	001	0005	2739	
@TBCNT	001	0000	0161	
@TBLEF	001	0010	0156	0158 9329
@TBLIX	001	0011	0158	3531
@TJ37B	001	0040	1099	
@TYPAM	001	0002	1043	9219 9249
@TYPO	001	001C	1042	
@UCB	001	0087	0040	3820 4126 5208 6351 6875 7721 8182 8240 8313 8394 8535 8570 8633 8647 8693 8708 8709 9115 9234 9287 9382 0455 0456 0457 0483 0551 0559 0867 0868 0884 0936 1337 1451 3431 3439 3496 3504 3506 3619
@UPARW	001	005A	0079	2722 8974
@VADDR	001	0002	0142	2658 2684 3061 3497 3509 3510 3511 3511 3525 3528 3530 3554 3555 3556 3594 3597 3600 3603 3606 3609 3612 3621 3624 3627 3630 3633 4153 4161 4206 4501 4512 4797 4814 4877 5214 5231 5639 5664 5696 6370 6382 6394 6836 6839 6845 6849 6882 6904 6921 6943 6965 6987 7004 7021 7241 7252 7273 7281 7298 7299 7300 7313 7314 7319 7320 7341 7359 7360 7396 7418 7441 7442 7451 7452 7459 7567 7611 7620 7662 7766 7874 7960 7972 7985 8137 8165 8178 8245 8250 8252 8262 8266 8272 8305 8320 8367 8373 8408 8505 8508 8553 8565 8642 8666 8696 8706 8734 8947 8949 9322 9399 9512 9976 9990 9998 0007 0018 0027 0045 0053 0063 0064 0161 0199 0259 0276 0512 0730 1113 1186 1255 1259 1263 1267 1271 1275 1697 1700 1732 2102 2183 2484 2629 2662 3434 3471 3489 3507 3519 3543 3587 3748 3775 3780
@VENTA	001	0056	0114	3328 3583 7250 7286 1685 1745
@VMDDV	001	00FE	0115	
@VMFD1	001	0000	0110	
@VMFD2	001	0001	0111	
@VMRS3	001	0002	0113	
@VMTRL	001	0001	0112	
@VOLID	001	0006	0092	
@VQ	001	0001	0026	6436 6452 7751 7922 8001 8346 2162 2298 3479 3791*
@WA37B	001	00FF	1107	
@WSFIT	001	0500	0102	
@WSTBL	001	0503	0103	
@XR	001	0002	0015	4128* 4132 4134 4142 4171 4172 4172 4173 4175 4176 4178 4185 4187 4187 4193 4195 4196 4197 4197 4212 4244 4273* 4287 4311*

## CROSS REFERENCE

SYMBOL LEN VALUE DEFN REFERENCES VER 15, MOD 00 31/05/21 PAGE 296

	4327	4327*	4484*	4485	4488	4489	4489	4497	4498	4498	4499	4499
	4502	4510	4625*	4626	4627	4635	4636	4636	4641	4643	4643*	4653*
	4654	4663	4664*	4665	4666	4667	4803	4819	4826	4832	4834	4845
	4847	4848	4848	4854	4864	4893	4907	4909	4909*	4924*	4936	4937
	4938	4981*	4982	4984	4986	5102*	5111	5116	5121	5128	5130	5130
	5131	5142	5145*	5148	5149*	5150	5157	5166	5168	5173	5176	5188
	5190	5209	5219	5240	5360*	5361	5365	5373	5378	5378	5379	5384
	5385	5385	5386	5390	5390	5391	5392	5392	5393	5393	5394	5394
	5395	5395	5396	5396	5397	5408	5409	5421	5424	5438	5441	5441*
	5446*	5589*	5594	5596	5596	5597	5605*	5609	5613	5616	5621	5628
	5633	5646	5650	5652	5653	5657	5662	5668	5684	5689	5690	5694
	5733*	5740	5741	5741	5743	5751	5752	5758	5760	5763	5897*	5898
	5899	5900	5900	5918	5921*	5926	5927	5928*	5929	6040*	6041	6049
	6051	6051	6056	6056	6062	6062	6189*	6190	6194	6197	6203*	6205
	6209	6212	6213	6216	6221	6222	6225	6340*	6341	6343	6344	6349
	6357	6358	6360	6361	6362	6364	6375	6383	6388	6389	6395	6436
	6437	6470	6471	6487	6758*	6762	6763*	6764	6768*	6769	6810*	6836
	6844*	6845	6849	6849*	6850	6861*	6899	6937	6959	6969*	7025*	7240*
	7241	7248*	7249*	7250	7252	7272*	7273	7280*	7286	7298	7299	7300
	7302*	7311*	7312	7313	7314	7369*	7370	7398*	7399	7400	7404	7409
	7409*	7424	7425*	7426	7427	7427*	7428	7433	7434*	7435	7570*	7571
	7571*	7572	7574	7576	7580	7585	7589	7596	7600*	7601	7617	7625
	7626	7632*	7633	7638	7640	7640*	7643	7645	7645	7646	7646*	7647
	7649	7653	7658*	7688	7688	7693	7694	7697	7698	7699	7700	7702*
	7703	7704*	7708*	7709	7717	7717	7718	7722	7723	7723	7724	7724
	7728	7729*	7736	7737	7739	7755	7796	7796	7798	7823	7834*	7840
	7841	7885	7886	7887	7889	7892*	7893	7895	7901	7901	7902	7906
	7907	7914	7915	7916	7922	7924	7931	7932	7934	7935	7964*	7965
	7966	7966	7970*	7976*	7977	7993	7994	7994*	8001	8008*	8009	8010
	8011	8140*	8141	8141*	8142	8144	8146	8150	8154	8159	8162	8163
	8168	8171	8180	8186	8189	8192	8199	8202	8202*	8204	8207	8210
	8224	8228	8228*	8238*	8299	8311	8315	8316	8317	8318	8321	8327
	8340*	8342	8353*	8355	8356	8369	8370	8371	8380*	8381	8389	8390
	8390	8391	8391	8398	8399	8399	8449	8451	8453	8453	8471	8484*
	8494	8496	8497	8504	8505	8506	8510	8513*	8514	8519	8520*	8527
	8536	8540	8542	8550	8555*	8561	8562	8643	8648	8648	8650	8650*
	8654	8656	8658	8660	8661	8662*	8663	8667*	8668	8677	8677	8678
	8678	8679	8679	8684	8685	8687	8687	8688	8688	8697*	8699	8731
	8732*	8735*	8911	8923	8924	8925	8929*	8930	8932*	8933	8934*	8935
	8935*	8936	8937	8937*	8938	8942	8951*	9114	9115	9117	9121	9123*
	9124	9142*	9150	9162	9174*	9182	9214	9224	9226	9231	9233	9234
	9237	9237*	9241	9250	9251	9251*	9258	9258	9259	9259	9260	9260*
	9280*	9289	9292	9323	9367	9389	9393	9412	9461	9468	9471	9477
	9495	9496	9497	9507	9508	9513	9515	9516	9517	9518	9532	9983*
	9984	9985	9985	9996*	0042*	0050	0099	0109*	0118*	0120	0126*	0163*
	0164	0168	0168*	0171*	0176*	0177	0177*	0180*	0181	0182*	0188*	0189
	0189*	0190	0190*	0222*	0226	0245	0499	0499	0500	0501	0501*	0519
	0528	0538	0629	0629*	0630	0701	0712	0714	0716	0727*	0748	0748*
	0749	0751	0753	0753*	0754	0757	0772	0775	0804	0804*	0805	0893
	0925	0932	0934	0941	0946	0950	1004	1004*	1005	1008	1078	1079
	1079*	1084	1090	1092	1094	1099	1131	1131*	1132	1134	1134*	1135
	1145	1151	1156	1157*	1158	1159*	1173	1174	1178*	1179	1179*	1180
	1186	1187*	1192*	1194	1200	1213	1213*	1214	1325	1338	1350	1358
	1377	1388	1416	1417*	1418	1423*	1447	1449	1466	1466*	1467	1470
	1704	1705	1706*	1707	1708*	1724	1727*	1745	1755	2034*	2057	2060
	2066	2073	2075	2080	2085	2108	2120	2131	2143	2144	2148	2148*

[illegible]

CROSS REFERENCE																			
SYMBOL	LEN	VALUE	DEFN	REFERENCES										VER 15, MOD 00			31/05/21	PAGE 298	
B\$CLTM	001	0600	2771																
B\$CMAT	001	0600	2793																
B\$CMGT	001	0665	2794																
B\$CMIN	001	06D3	2795																
B\$CMPR	001	069B	2798																
B\$CMPT	001	069B	2797																
B\$CMPU	001	0600	2799																
B\$CMRD	001	06D0	2796																
B\$CNXT	001	0600	2776																
B\$CPCT	001	0CA8	2858																
B\$CPRT	001	0600	2790																
B\$CPRU	001	0600	2791																
B\$CPSE	001	06E7	2800																
B\$CPUT	001	0600	2784																
B\$CPWA	001	0CA6	2929																
B\$CRAD	001	150D	2899																
B\$CRBS	001	1509	2901																
B\$CREA	001	06CF	2788																
B\$CREM	001	0000	2765																
B\$CRMK	001	0001	2977																
B\$CRSR	001	06E3	2789																
B\$CRST	001	06A6	2785																
B\$CRSW	001	0E42	2976																
B\$CRTN	001	06CF	2782																
B\$CSBF	001	0600	2752	2766	2767	2768	2771	2772	2773	2774	2775	2776	2777	2778	2779				
				2780	2781	2782	2783	2784	2785	2786	2787	2788	2789	2790	2791				
				2792	2793	2794	2795	2796	2797	2798	2799	2800	2801	2802	2805				
				2806	2807	2808	2809												
B\$CSCN	001	14B0	2874																
B\$CSMK	001	0007	2980																
B\$CSSW	001	14BC	2979																
B\$CSTP	001	06D6	2801																
B\$CSTR	001	14CC	2898																
B\$CSXA	001	2000	2758																
B\$CTYP	001	0A5F	2852																
B\$CVPD	001	0C5D	2857																
B\$CVPG	001	0CA5	2856																
B\$CWRK	001	F500	2926																
B\$DIST	001	0700	2818																
B\$DLNK	001	1B37	2924																
B\$DL4T	001	1A6B	2895																
B\$DPWA	001	0E46	2930																
B\$DST2	001	073A	2819																
B\$ERMK	001	0007	2953																
B\$ERSW	001	0993	2952																
B\$FACA	001	0E53	2861																
B\$FAIS	001	15AC	2878																
B\$FAIW	001	15A0	2879																
B\$FCON	001	0A46	2851																
B\$FORT	001	1B0E	2920																
B\$FPWA	001	15AC	2931																
B\$FRMK	001	0007	2971																
B\$FRSW	001	16CC	2970																
B\$FSC1	001	0E4C	2862																
B\$FSC2	001	0E4D	2863																
B\$FSMK	001	0007	2962																



## CROSS REFERENCE

VER 15, MOD 00 31/05/21 PAGE 299

SYMBOL	LEN	VALUE	DEFN	REFERENCES
B\$FSSW	001	0E5C	2961	
B\$FSVA	001	0E4F	2864	
B\$FTND	001	1B0B	2922	
B\$FTPT	001	1B0D	2921	
B\$FVME	001	15A2	2883	
B\$FVMP	001	15A4	2884	
B\$FVMS	001	15A6	2885	
B\$FVPE	001	15A8	2880	
B\$FVPP	001	15AA	2881	
B\$FVPS	001	15AC	2882	
B\$GBSW	001	08AF	2955	
B\$GBWK	001	0001	2956	
B\$GETC	001	0867	2832	
B\$GPTR	001	0878	2834	
B\$GTBF	001	1E00	2756	
B\$IFMK	001	0007	2974	
B\$IFSW	001	16E5	2973	
B\$INVT	001	1B38	2914	
B\$KWMK	001	0001	2968	
B\$KWSW	001	159E	2967	
B\$LBAS	001	185E	2905	
B\$LBSV	001	18E7	2903	
B\$LDRP	001	1A00	2753	
B\$LINE	001	07D0	2820	
B\$LIST	001	1853	2887	
B\$LRTN	001	18EB	2904	
B\$LSTR	001	1862	2902	
B\$LTYP	001	18F2	2888	
B\$MATR	001	18F3	2890	
B\$MBMK	001	0007	2989	
B\$MBSW	001	1903	2988	
B\$MFBK	001	1B8F	2916	
B\$MGMK	001	0007	2986	
B\$MGSW	001	18FF	2985	
B\$MPMK	001	0007	2992	
B\$MPSW	001	1981	2991	
B\$MRMK	001	0007	2983	
B\$MRSW	001	0DDE	2982	
B\$NUMC	001	0873	2833	
B\$NXMK	001	0007	2959	
B\$NXSW	001	071D	2958	
B\$PARP	001	0A41	2841	
B\$PBNL	001	0A01	2847	
B\$PCAD	001	0A40	2842	
B\$PCDL	001	09D3	2846	
B\$PCPG	001	0A35	2845	
B\$PECT	001	0A44	2849	
B\$PERC	001	0A39	2848	
B\$PFAE	001	0033	2839	
B\$PFCL	001	009D	2840	
B\$PFNC	001	094E	2837	
B\$PFWP	001	0015	2838	
B\$PNBY	001	0A41	2843	
B\$PPWA	001	0A35	2928	
B\$PRM1	001	1AF3	2932	
B\$PTBF	001	1F00	2757	



CROSS REFERENCE																	
SYMBOL	LEN	VALUE	DEFN	REFERENCES	VER 15, MOD 00 31/05/21 PAGE 300												
B\$PUTC	001	093A	2836														
B\$PVAD	001	0A43	2844														
B\$RMRK	001	1AE6	2897														
B\$RTRN	001	1AF5	2933														
B\$SABF	001	1C00	2754														
B\$SCAN	001	1514	2876														
B\$SCAT	001	13C8	2871														
B\$SSCON	001	001B	2854														
B\$SCVT	001	12E0	2869														
B\$SDPL	001	07DA	2822														
B\$SFAB	001	0E48	2866														
B\$SFNT	001	143C	2872														
B\$SLDT	001	109C	2868														
B\$SLVT	001	1062	2867														
B\$SNAT	001	131A	2870														
B\$SPAT	001	07E0	2823														
B\$SSTA	001	1BAC	2918														
B\$STAS	001	061B	2807														
B\$STIF	001	0606	2809														
B\$STMA	001	061B	2808														
B\$STML	001	0600	2806														
B\$STRL	001	0600	2805														
B\$SVRB	001	0E46	2865														
B\$SYMB	001	0DBC	2860														
B\$TCD2	001	0001	2938														
B\$TLTH	001	0002	2939	2940													
B\$TOD1	001	0000	2937														
B\$TOTB	001	1AF8	2940														
B\$TTAB	001	1AFA	2936	2940													
B\$TYPE	001	0739	2821														
B\$WORK	001	15A0	2925														
B\$ZDBN	001	19F2	2892														
B@ABAS	001	0007	3525														
B@ACD1	001	0001	3522	3523													
B@ACD2	001	0003	3523	3524													
B@AFLG	001	0000	3517														
B@ALLA	001	005C	3342														
B@AMAX	001	0005	3524	3525													
B@BLNK	001	0040	3351	7931	7965	9984	0630	0757	0805	1005	1151	1214	1233	1467	2056		
				2131	2161												
B@BLSZ	001	0100	3476	3615	3618	3621	3636	3639	7620	7662	7766	8262	8305	8367	8602		
B@BREQ	001	0084	3131														
B@BRHI	001	0088	3132														
B@BRLO	001	0082	3130														
B@BRNE	001	0094	3134														
B@BRNH	001	0098	3135														
B@BRNL	001	0092	3133														
B@CADD	001	0006	3000														
B@CADF	001	0058	3041														
B@CBAS	001	0003	3528														
B@CBNX	001	004A	3034														
B@CBRA	001	0046	3032														
B@CBRC	001	0044	3031														
B@CBRD	001	0048	3033														

## CROSS REFERENCE

VER 15, MOD 00 31/05/21 PAGE 301

SYMBOL	LEN	VALUE	DEFN	REFERENCES
B@CCMC	001	0042	3030	
B@CCMF	001	0040	3029	
B@CCNT	001	001F	3454	2143
B@CCSA	001	003E	3028	
B@CDCA	001	006A	3050	
B@CDDL	001	006C	3051	
B@CDIV	001	000C	3003	
B@CDMN	001	0001	3527	3528
B@CDWA	001	006E	3052	
B@CEOF	001	0070	3053	
B@CEOP	001	0068	3049	1706
B@CFCI	001	0016	3008	
B@CFN0	001	0012	3006	
B@CFN1	001	0014	3007	
B@CFOR	001	004E	3036	
B@CGET	001	0052	3038	6763
B@CHAR	001	0000	3467	0050 0519 0528 0538 0630 0701 0712 0714 0716 0749 0751 0754 0757 0772 0775 0805 0893 0925 0932 0934 0941 0946 0950 1005 1008 1084 1092 1094 1099 1132 1135 1145 1151 1173 1194 1214 1325 1338 1350 1358 1377 1388 1447 1449 1467 1470
B@CHLT	001	0004	2999	
B@CIEX	001	00C5	3427	1262 1274
B@CIMH	001	0066	3048	
B@CINI	001	0056	3040	
B@CIPI	001	00D7	3430	1258 1270
B@CIS2	001	00E2	3433	1254 1266
B@CMF1	001	0018	3009	
B@CMF2	001	001A	3010	
B@CMF3	001	001C	3011	
B@CMA	001	006B	3362	7993 0519
B@CMPY	001	000A	3002	
B@CMSM	001	001E	3012	
B@CNEG	001	0010	3005	
B@CNXT	001	0050	3037	
B@COLN	001	007A	3364	
B@CPMK	001	00FF	3272	3276 3280 3281 3315
B@CPRS	001	0060	3045	
B@CPRU	001	0062	3046	
B@CPUT	001	0054	3039	
B@CPWR	001	000E	3004	
B@CRSR	001	005A	3042	
B@CRST	001	005C	3043	
B@CSA1	001	0036	3024	
B@CSA2	001	0038	3025	
B@CSB1	001	003A	3026	
B@CSC1	001	002A	3018	
B@CSD0	001	002E	3020	
B@CSD1	001	0030	3021	
B@CSD2	001	0032	3022	
B@CSF1	001	0022	3014	
B@CSF2	001	0024	3015	
B@CSTA	001	0034	3023	
B@CSTC	001	0028	3017	
B@CSTF	001	0020	3013	
B@CSTH	001	0064	3047	
B@CSTX	001	003C	3027	

## CROSS REFERENCE

VER 15, MOD 00 31/05/21 PAGE 302

SYMBOL	LEN	VALUE	DEFN	REFERENCES
B@CSUB	001	0008	3001	
B@CSVC	001	0002	2998	
B@CTYP	001	0020	3452	2120
B@CUSC	001	002C	3019	
B@CUSF	001	0026	3016	
B@CVAR	001	005B	3341	
B@DAMK	001	0080	3520	
B@DASA	001	00FF	3281	
B@DASC	001	0040	3285	
B@DASM	001	0038	3283	
B@DCGT	001	0050	3291	
B@DCLS	001	0054	3297	
B@DDAT	001	0024	3277	
B@DDEF	001	0034	3278	
B@DDIM	001	0004	3279	
B@DDUM	001	00FF	3315	
B@DEC0	001	00F0	3410	5611 0941 1008 1338 1409 1470 2066 2085 2312 2341 2378
B@DEC1	001	00F1	3411	4253 4867 4934 5410 6377 2279
B@DEC2	001	00F2	3412	5411 5590
B@DEC3	001	00F3	3413	
B@DEC4	001	00F4	3414	5615
B@DEC5	001	00F5	3415	
B@DEC6	001	00F6	3416	
B@DEC7	001	00F7	3417	
B@DEC8	001	00F8	3418	
B@DEC9	001	00F9	3419	5412 5430
B@DEND	001	0058	3313	3314
B@DEOF	001	0058	3314	
B@DFOR	001	0028	3286	
B@DGET	001	0040	3294	
B@DGSB	001	0020	3292	
B@DGTO	001	0044	3290	
B@DIFA	001	0048	3288	
B@DIFC	001	004C	3289	
B@DIGS	001	007B	3344	
B@DIMG	001	003C	3303	
B@DINP	001	0000	3298	
B@DIVD	001	0061	3361	
B@DLTA	001	00FF	3280	
B@DLTC	001	0040	3284	
B@DLTM	001	0038	3282	
B@DL01	001	0001	3595	3598
B@DL02	001	0003	3598	3601
B@DL03	001	0005	3601	3604
B@DL04	001	0007	3604	3607
B@DL05	001	0009	3607	3610
B@DL06	001	000B	3610	3613
B@DL07	001	0045	3613	3616
B@DL08	001	0145	3616	3619
B@DL09	001	0245	3619	3622
B@DL10	001	0289	3622	3625
B@DL11	001	02C3	3625	3628
B@DL12	001	02FD	3628	3631
B@DL13	001	0337	3631	3634
B@DL14	001	0371	3634	3637
B@DL15	001	0471	3637	3640

## CROSS REFERENCE

VER 15, MOD 00 31/05/21 PAGE 303

SYMBOL	LEN	VALUE	DEFN	REFERENCES
B@DL16	001	0507	3640	
B@DMAT	001	0008	3304	
B@DMGT	001	0044	3305	
B@DMIN	001	0038	3306	
B@DMPR	001	0048	3309	
B@DMPT	001	004C	3308	
B@DMPU	001	0054	3310	
B@DMRD	001	003C	3307	
B@DNXT	001	0044	3287	
B@DPNT	001	004B	3352	7902 0893 1325 1358 2306 2332
B@DPRT	001	002C	3301	
B@DPRU	001	0030	3302	
B@DPSE	001	0050	3311	
B@DPUT	001	0040	3295	
B@DREA	001	000C	3299	
B@DREM	001	00FF	3276	
B@DRSR	001	005C	3300	
B@DRST	001	0050	3296	
B@DRTN	001	005C	3293	
B@DSCY	001	0004	3268	
B@DSIF	001	001C	3317	
B@DSLT	001	0010	3316	
B@DSML	001	0010	3318	
B@DSNS	001	0018	3270	
B@DSS1	001	0000	3269	
B@DSTP	001	0054	3312	
B@DTBN	001	0010	3334	
B@DTB1	001	0050	3333	
B@DTCY	001	0009	3330	
B@DTSN	001	0010	3332	
B@DTS1	001	0040	3331	
B@DTYP	001	0040	3446	6850 7601 7633 7709 8327 8381 1125
B@DURE	001	0020	3164	
B@DVCY	001	0007	3327	3208
B@DVC1	001	0056	3328	
B@DWCY	001	0005	3324	
B@DWT1	001	0003	3325	
B@D1MK	001	0080	3518	
B@D2MK	001	00C0	3519	
B@EOST	001	001E	3340	0050 0528 0538 0749
B@EQUL	001	007E	3366	
B@EXPC	001	00C5	3343	0925 1377
B@FOFL	001	005C	3345	
B@FVAD	001	0001	3530	
B@GETC	001	0001	3469	
B@GETE	001	00FF	3470	
B@GETS	001	0000	3468	
B@GRTR	001	006E	3363	
B@ICON	001	0050	3425	0716 1099
B@LADD	001	0001	3069	
B@LADF	001	0002	3110	
B@LADV	001	0008	3554	3575
B@LBIN	001	0002	3479	3480 3486
B@LBNX	001	0003	3103	
B@LBRA	001	0003	3101	7331
B@LBRC	001	0004	3100	

CROSS REFERENCE															
SYMBOL	LEN	VALUE	DEFN	REFERENCES	VER 15, MOD 00 31/05/21 PAGE 304										
B@LBRD	001	0003	3102												
B@LBRS	001	0001	3104												
B@LCCA	001	0004	3510												
B@LCCC	001	0001	3062	3100											
B@LCDV	001	0004	3555	3576											
B@LCER	001	0001	3060	3124											
B@LCFN	001	0004	3511												
B@LCLN	001	0002	3065	3116	3117	3124									
B@LCLS	001	0001	3113												
B@LCMC	001	0001	3099												
B@LCMF	001	0001	3098												
B@LCNA	001	0006	3509												
B@LCNN	001	0001	3063	3088	3097	3109	3121	0491	0609	0648					
B@LCOP	001	0001	3059	3067	3068	3069	3070	3071	3072	3073	3074	3075	3076	3077	3078
				3079	3080	3081	3082	3083	3084	3085	3086	3087	3088	3089	3090
				3091	3092	3093	3094	3095	3096	3097	3098	3099	3100	3101	3102
				3103	3104	3105	3106	3107	3108	3109	3110	3111	3112	3113	3114
				3115	3116	3117	3118	3119	3120	3121	3122	6762	1705		
B@LCRV	001	0013	3553	3573											
B@LCSA	001	0002	3097												
B@LCVA	001	0002	3061	3075	3076	3077	3078	3079	3080	3081	3082	3083	3084	3086	3087
				3089	3090	3091	3092	3093	3094	3095	3100	3101	3102	3103	3105
				3106	3107	3119	3120	1724	1755						
B@LCXX	001	0001	3064	3096	3108	3110	3114	3115	0593	0600	0650				
B@LDAT	001	0004	3223												
B@LDCA	001	0003	3119	1780											
B@LDDL	001	0003	3120												
B@LDDM	001	0004	3483												
B@LDEF	001	0003	3224												
B@LDIM	001	0003	3225												
B@LDIN	001	0004	3482	3483	3484										
B@LDIV	001	0001	3072												
B@LDMN	001	0002	3480	3509	3510	3522	3523	3524	3527	3554	3555				
B@LDSN	001	0004	3484	0142	0143	0147	0147	0152	0152	0292	0295	0308			
B@LDWA	001	0002	3121	7331											
B@LELP	001	0010	3552												
B@LEND	001	0003	3252												
B@LEOF	001	0001	3122												
B@LEOP	001	0001	3118												
B@LERC	001	0003	3124												
B@LESP	001	0008	3551												
B@LESS	001	004C	3353												
B@LET\$	001	005B	3373												
B@LET#	001	007B	3374												
B@LET@	001	007C	3375												
B@LETA	001	00C1	3377	0772	1194										
B@LETB	001	00C2	3379												
B@LETC	001	00C3	3380												
B@LETD	001	00C4	3381												
B@LETE	001	00C5	3382												
B@LETF	001	00C6	3383												
B@LETG	001	00C7	3384												
B@LETH	001	00C8	3385												
B@LETI	001	00C9	3386												

## CROSS REFERENCE

VER 15, MOD 00 31/05/21 PAGE 305

SYMBOL	LEN	VALUE	DEFN	REFERENCES
B@LETL	001	00D3	3389	
B@LETM	001	00D4	3390	
B@LETN	001	00D5	3391	
B@LETO	001	00D6	3392	
B@LETP	001	00D7	3393	
B@LETQ	001	00D8	3394	
B@LETR	001	00D9	3395	
B@LETS	001	00E2	3396	
B@LETT	001	00E3	3397	
B@LETU	001	00E4	3398	
B@LETV	001	00E5	3399	
B@LETW	001	00E6	3400	
B@LETX	001	00E7	3401	
B@LETY	001	00E8	3402	
B@LETZ	001	00E9	3403	
B@LEXP	001	0008	3442	
B@LFCI	001	0003	3077	7332
B@LFNA	001	0002	3556	3577
B@LFN0	001	0003	3075	
B@LFN1	001	0003	3076	
B@LFOR	001	0003	3105	
B@LFRT	001	0004	3497	3498
B@LGET	001	0003	3107	
B@LGSB	001	0005	3231	
B@LGTO	001	0004	3230	
B@LHLT	001	0001	3068	
B@LIEX	001	0002	3428	
B@LIFN	001	0003	3491	
B@LILP	001	0009	3550	3568 3569 3570
B@LIMG	001	0001	3242	
B@LIMH	001	0003	3117	
B@LINI	001	0002	3109	
B@LINP	001	0005	3237	
B@LIPI	001	0003	3431	
B@LISP	001	0005	3549	3557 3563 3564 3565
B@LIS2	001	0005	3434	
B@LIVT	001	0001	3507	
B@LKCL	001	0005	3236	
B@LKFR	001	0003	3227	
B@LKGT	001	0003	3233	
B@LKIF	001	0002	3229	
B@LKON	001	0002	3262	
B@LKPT	001	0003	3234	
B@LKPU	001	000A	3241	
B@LKRR	001	0007	3239	
B@LKRT	001	0005	3235	
B@LKTO	001	0002	3256	
B@LLET	001	0003	3226	
B@LL01	001	0002	3594	3595
B@LL02	001	0002	3597	3598
B@LL03	001	0002	3600	3601
B@LL04	001	0002	3603	3604
B@LL05	001	0002	3606	3607
B@LL06	001	0002	3609	3610
B@LL07	001	003A	3612	3613
B@LL08	001	0100	3615	3616

## CROSS REFERENCE

VER 15, MOD 00 31/05/21 PAGE 306

SYMBOL	LEN	VALUE	DEFN	REFERENCES
B@LL09	001	0100	3618	3619
B@LL10	001	0044	3621	3622
B@LL11	001	003A	3624	3625
B@LL12	001	003A	3627	3628
B@LL13	001	003A	3630	3631
B@LL14	001	003A	3633	3634
B@LL15	001	0100	3636	3637
B@LL16	001	0096	3639	3640
B@LMAT	001	0003	3243	
B@LMF1	001	0003	3078	
B@LMF2	001	0003	3079	
B@LMF3	001	0003	3080	
B@LMGT	001	0006	3244	
B@LMIN	001	0008	3245	
B@LMPR	001	0008	3248	
B@LMPT	001	0006	3247	
B@LMPU	001	000D	3249	
B@LMPY	001	0001	3071	
B@LMRD	001	0007	3246	
B@LMSM	001	0003	3081	
B@LNEG	001	0001	3074	
B@LNEX	001	0004	3228	
B@LNXT	001	0003	3106	
B@LPAR	001	004D	3354	
B@LPRS	001	0002	3114	
B@LPRT	001	0005	3240	
B@LPRU	001	0002	3115	
B@LPSE	001	0005	3250	
B@LPUT	001	0002	3108	
B@LPWR	001	0001	3073	
B@LREA	001	0004	3238	
B@LREM	001	0003	3222	
B@LRSR	001	0001	3111	
B@LRST	001	0001	3112	
B@LRTN	001	0006	3232	
B@LSA1	001	0003	3093	
B@LSA2	001	0003	3094	
B@LSB1	001	0003	3095	
B@LSC1	001	0003	3087	
B@LSDF	001	0004	3477	
B@LSD0	001	0003	3089	
B@LSD1	001	0003	3090	
B@LSD2	001	0003	3091	
B@LSF1	001	0003	3083	
B@LSF2	001	0003	3084	
B@LSKW	001	0002	3493	
B@LSNO	001	0002	3486	0141 0291
B@LSPT	001	0003	3501	3504
B@LSTA	001	0003	3092	
B@LSTC	001	0003	3086	
B@LSTE	001	0004	3257	
B@LSTF	001	0003	3082	
B@LSTH	001	0003	3116	
B@LSTP	001	0004	3251	
B@LSTX	001	0002	3096	
B@LSUB	001	0001	3070	



CROSS REFERENCE																
SYMBOL	LEN	VALUE	DEFN	REFERENCES	VER 15, MOD 00 31/05/21 PAGE 307											
B@LSVC	001	0001	3067													
B@LTHN	001	0004	3258													
B@LTYP	001	0001	3487													
B@LUFN	001	0002	3494													
B@LUSC	001	0002	3088													
B@LUSF	001	0001	3085													
B@LVPG	001	0100	3581	3584	4097	4242	4376	4788	4979	4991	4992	5097	5355	5582	5728	
				5796	6184	6235	6236	6237	6238	6335	6420	6506	6507	6752	7232	
				7353	7562	7679	7790	7874	7879	8133	8285	8433	8619	9984*	9985	
				9985	9985*	1678										
B@MINS	001	0060	3360	7916	7934	0714	0934	1094	1096	1265	1269	1273	1449	2059	2361	
B@MULT	001	005C	3357													
B@NAAR	001	001D	3545	3575	3627											
B@NCAR	001	001D	3546	3576	3630											
B@NCRV	001	001D	3544	3573	3624											
B@NDGT	001	000A	3537	3543												
B@NEQL	001	007F	3367													
B@NFRT	001	000A	3496	3498												
B@NICN	001	0006	3539	3541												
B@NIEL	001	0007	3541	3557	3563	3568										
B@NIFN	001	0018	3490													
B@NIVR	001	0001	3540	3541												
B@NIVT	001	0057	3506													
B@NLDV	001	0122	3543	3565	3570	3621										
B@NLRV	001	001D	3542	3564	3569	3612										
B@NLTR	001	001D	3536	3542	3543	3544	3545	3546	3547							
B@NSKW	001	0004	3492													
B@NSPT	001	0028	3500													
B@NUFN	001	001D	3547	3577	3633											
B@NVPG	001	0100	3580	3584												
B@NXHI	001	00E3	3461	1024												
B@NXLO	001	001E	3460	4654	4847	5597	1025	1411								
B@NXZR	001	0080	3459	3460	3461	4232	4234	4485	4497	4517	4628	4651	4803	4937	4984	
				5151	5166	5200	5257	5594	5609	5621	5633	5716	5720	5722	5781	
				5783	5786	5788	5791	5908	5938	6194	6205	6341	6349	6414	8023	
				1023	1305	2073	2075	2197	2284	2286	2295	2362	2403			
B@PLUS	001	004E	3355	0712	0932	1091	1092	1253	1257	1261	1447					
B@POWR	001	005A	3356													
B@PREC	001	0020	3448	7718	7722	8389	8398									
B@PROD	001	0023	3557													
B@PRPL	001	0002	3144	7604												
B@PRPN	001	0001	3143													
B@PRPR	001	0004	3146	2199												
B@PRPS	001	0003	3145													
B@PRRC	001	0007	3149	7598												
B@PRRL	001	0008	3150	2198	2199											
B@PRSL	001	0005	3147	6937	2040											
B@PRSS	001	0006	3148													
B@PTAB	001	0000	3502													
B@PTAD	001	0001	3503													
B@PTSA	001	0002	3504													
B@PUD1	001	0006	3160													
B@PUD2	001	0007	3161													
B@PUI0	001	0001	3154													
B@PUI1	001	0004	3155													
B@PUI2	001	0005	3156													

## CROSS REFERENCE

VER 15, MOD 00 31/05/21 PAGE 308

SYMBOL	LEN	VALUE	DEFN	REFERENCES
B@PUNL	001	0002	3158	
B@PUNS	001	0003	3159	
B@PUTM	001	0010	3163	
B@RPAR	001	005D	3358	
B@SADV	001	00E8	3575	3578
B@SAVL	001	0B76	3571	3588
B@SAVS	001	065E	3566	3587
B@SCDV	001	0074	3576	3578
B@SCLN	001	005E	3359	
B@SCRV	001	0227	3573	3587 3588
B@SDMK	001	0080	3488	8533
B@SEXP	001	0004	3441	
B@SFAT	001	0196	3578	3587 3588 3639
B@SFNA	001	003A	3577	3578
B@SFRT	001	0028	3498	
B@SIEL	001	003F	3568	3571
B@SIES	001	0023	3563	3566
B@SIGN	001	0010	3450	
B@SLDL	001	0A32	3570	3571
B@SLDS	001	05AA	3565	3566
B@SLVL	001	0105	3569	3571
B@SLVS	001	0091	3564	3566
B@SQUO	001	007D	3365	7638 7643 7647 7649 7895 8199 8207 8210 0701 0751 0754 1084 1132 1135
B@STAT	001	0000	3440	
B@TASA	001	0012	3175	
B@TASC	001	001E	3181	
B@TASM	001	0018	3177	
B@TASS	001	007B	3182	
B@TCGT	001	0030	3190	
B@TCLS	001	0042	3196	
B@TDAT	001	0006	3171	
B@TDEF	001	0009	3172	
B@TDIM	001	000C	3173	
B@TDUM	001	0078	3214	
B@TEND	001	0072	3212	
B@TEOF	001	0075	3213	
B@TFOR	001	0021	3184	
B@TGET	001	0039	3193	
B@TGSB	001	0033	3191	
B@TGTO	001	002D	3189	
B@TIFA	001	0027	3186	
B@TIFC	001	002A	3187	
B@TIFS	001	007D	3188	
B@TIMG	001	0054	3202	
B@TINP	001	0045	3197	
B@TLTA	001	000F	3174	
B@TLTC	001	001B	3178	
B@TLTM	001	0015	3176	
B@TLTS	001	0079	3179	
B@TMAS	001	007C	3183	
B@TMAT	001	0057	3203	
B@TMGT	001	005A	3204	
B@TMIN	001	005D	3205	
B@TMLS	001	007A	3180	
B@TMPR	001	0066	3208	

## CROSS REFERENCE

VER 15, MOD 00 31/05/21 PAGE 309

SYMBOL	LEN	VALUE	DEFN	REFERENCES
B@TMPT	001	0063	3207	
B@TMPU	001	0069	3209	
B@TMRD	001	0060	3206	
B@TNXT	001	0024	3185	
B@TPRT	001	004E	3200	
B@TPRU	001	0051	3201	
B@TPSE	001	006C	3210	
B@TPUT	001	003C	3194	
B@TRAC	001	0080	3444	
B@TREA	001	0048	3198	
B@TREM	001	0003	3170	
B@TRSR	001	004B	3199	
B@TRST	001	003F	3195	
B@TRTN	001	0036	3192	
B@TSTP	001	006F	3211	
B@VMC1	001	0056	3583	
B@VMLB	001	F0CD	3588	
B@VMSB	001	F5E5	3587	
B@VMSZ	001	0000	3584	3586 3587 3588
B@VMTB	001	0000	3586	
B@ZNEG	001	00D0	3457	1367
B@ZPOS	001	00F0	3456	4134 4627 4819 4982 5157 5190 5361 5616 5927 7932 7935 0955 0956 1367 1412 2057 2060 2338 2343
BFPCAR	001	4FFB	3712	3715
BFPCRO	001	4FFF	3714	3694*
BUFADR	002	4DD9	3544	3436* 3478 3589* 3691
BUFRWK	002	4DDE	3548	9515* 3590* 3602* 3624
CBFAD1	001	0C70	5888	5892
CBFEXP	001	0002	5937	5919 5920 5938
CBFPZD	004	0C70	5897	
CBFSFT	002	0CB1	5938	5919
CBF100	004	0C97	5921	5918* 5919* 5920*
CBF900	004	0CAC	5933	5909
CCZAD1	001	04AD	4616	4620
CCZDC1	001	04FB	4678	4644
CCZDFP	004	04AD	4625	
CCZEXP	001	04FA	4673	4628* 4644* 4651 4667
CCZONE	001	0001	4677	4643
CCZSGN	001	04F9	4672	4626* 4666
CCZ020	003	04C2	4641	4652
CCZ100	005	04DF	4663	4642
CCZ900	004	04F5	4668	4655
CDBACC	001	0004	6067	6051* 6062
CDBADD	001	0003	6068	6051 6051 6056 6056 6062
CDBAD1	001	0CB2	6031	6035
CDBNZD	004	0CB2	6040	
CDBONE	001	0CDA	6072	6041
CDB010	003	0CBD	6049	6048* 6057 6057* 6058
CDB100	004	0CC7	6056	6050
CENAD1	001	0470	4475	4479
CENXZD	004	0470	4484	
CENZRO	001	04AC	4517	4510
CEN100	003	0487	4497	4487
CEN150	003	0498	4502	
CEN200	004	049E	4510	4486
CEN900	004	04A8	4513	4490 4503

## CROSS REFERENCE

VER 15, MOD 00 31/05/21 PAGE 310

SYMBOL	LEN	VALUE	DEFN	REFERENCES
DENTRY	001	0025	3565	3438
DERROR	003	0088	3564	3488
DFKACK	001	0010	9090	9231
DFKATA	001	261C	9053	9122 9266*
DFKATC	001	2733	9218	9241 9250
DFKBLE	002	2617	9049	8938* 9123
DFKBSP	001	0016	9084	9197
DFKBS2	001	2600	9033	8918 8920 8929 8932 9074 9192 9254* 9262* 9266*
DFKBS3	001	2700	9193	8911 8940 9075 9121* 9150* 9278
DFKCNT	001	2624	9061	9141*
DFKC01	002	2621	9056	9058 9223 9239 9280
DFKDIO	001	0065	8945	8916
DFKDLP	001	2696	9146	9112 9240
DFKDTK	001	0040	9097	9118
DFKEMS	001	0002	9089	9206
DFKENB	001	0012	9094	9108
DFKENT	001	2653	9111	8932
DFKERA	001	2789	9257	9202
DFKERS	001	0003	9086	9201
DFKEUD	001	001D	9092	
DFKEXL	001	0019	9096	9034
DFKEYN	001	2500	8913	8910 8918 8935 8940 8945 8955 9032 9074 9075
DFKIAR	002	2615	9048	8930* 8943
DFKIET	002	2619	9050	9101
DFKIME	002	262C	9066	9222* 9223*
DFKIRK	001	2634	9071	9039
DFKIST	002	2621	9058	9291
DFKKIX	001	0011	9091	9233
DFKLLKA	001	25F9	9014	9022
DFKLMG	002	2628	9064	8923* 8951 9235 9254 9262
DFKLNK	001	0039	9022	9266
DFKLOK	001	0018	9093	9102 9221
DFKMCT	002	262E	9067	9222
DFKMSD	002	27B1	9278	9259
DFKNAB	001	264D	9107	9104
DFKNPS	002	261F	9055	8926* 8927* 8928 8931* 9139* 9141 9148 9151*
DFKNSK	001	261D	9054	9038* 9039 9113 9116 9118 9195 9197 9199 9201 9204 9206 9208
				9219 9225* 9249
DFKNTR	001	2603	9036	8929
DFKPG2	002	263A	9074	8949
DFKPG3	002	263C	9075	8947
DFKPL1	001	2770	9243	9231* 9233* 9237
DFKPL2	001	27A3	9269	9251
DFKPL3	001	27A5	9271	9258* 9259* 9260
DFKPPL	001	2623	9059	9063 9142 9148* 9181
DFKPRT	001	26AC	9160	9143 9238 9252 9261 9281
DFKP10	001	26BD	9166	8921* 8922* 9163 9168
DFKP20	002	26BF	9167	9162*
DFKRET	002	2630	9068	8936* 9103* 9109
DFKRKY	001	0011	9088	9070 9195
DFKRMG	002	262A	9065	8925* 8928* 9181
DFKROR	001	27BB	9286	9114
DFKROS	002	2632	9069	8933* 9037
DFKRTN	001	0015	9085	9199
DFKRT1	001	2683	9138	9125 9215
DFKSGL	001	0007	9277	9273

## CROSS REFERENCE

VER 15, MOD 00 31/05/21 PAGE 311

SYMBOL	LEN	VALUE	DEFN	REFERENCES
DFKSG1	001	27A9	9275	9277 9278
DFKSPA	001	279D	9265	9205
DFKSPB	001	274D	9230	9198
DFKSPC	001	0040	9087	9204
DFKSTN	001	2626	9063	8924* 9149 9150 9235 9239* 9254* 9262*
DFKTAB	001	0005	9083	9208
DFKTBL	001	25C0	8956	8937 9022
DFKTST	001	26DD	9179	9120 9203
DFKULK	001	001C	9095	9106 9183
DFKXDP	002	2638	9073	9103
DFKXIT	001	264A	9105	9119 9126 9209 9220 9263 9282 9297
DFKXRS	002	2636	9072	8942* 9174
DFK001	001	0001	9082	9139
DFK100	004	2565	8946	8915* 8916* 8935 8937 9073
DFK120	005	2569	8947	9073
DFK140	004	257E	8952	
DFK160	003	2600	9034	9040
DFK180	003	263D	9101	9196 9255
DFK200	004	2671	9122	9267
DFK220	005	2678	9124	9122* 9149*
DFK240	004	2699	9148	9144
DFK260	004	26A8	9152	9140* 9147*
DFK280	004	26C8	9170	9164
DFK300	003	26D6	9174	9169
DFK320	004	26D9	9175	9161*
DFK340	004	26EA	9184	9180*
DFK350	001	2700	9194	9117 9226
DFK360	004	2740	9223	9224
DFK380	003	2750	9232	9121* 9214* 9234*
DFK400	004	2759	9235	9232
DFK420	003	276D	9241	9236
DFK440	003	2772	9249	9182
DFK460	003	2778	9251	9200
DFK480	004	277E	9253	9150* 9289
DFK500	003	27B2	9280	9207
DFK520	003	27BB	9287	9115* 9292*
DFK540	005	27C5	9291	9287
DFPAPC	002	28DD	9420	9362* 9363* 9365 3692* 3693*
DFPASE	001	2800	9418	9312 9382* 9430 9434 9443 9460 3427 3584 3731 3754* 3784
DFPASY	002	29D5	9538	9495* 9496* 9497
DFPCFD	002	28EB	9430	9363
DFPCHK	001	5300	3730	
DFPDSDV	004	28F4	9435	9452 9493 3739*
DFPENT	004	5311	3739	3734
DFPEOR	001	4DF3	3571	3530
DFPERC	001	28EE	9433	9381
DFPERR	004	29DD	9542	
DFPETN	001	28E9	9429	9352
DFPEXT	001	29D3	9537	9507
DFPGCT	001	0000	9544	9486*
DFPIOR	001	29D7	9540	
DFPIST	001	28F5	9436	9326 9328* 9336 9337* 9338* 9340* 9342 9346* 9438 9450 9493* 3733 3737* 3739 3742
DFPITE	002	28E7	9445	9406
DFPLBU	002	29D2	9536	9508
DFPMCK	001	2939	9485	9478

## CROSS REFERENCE

VER 15, MOD 00 31/05/21 PAGE 312

SYMBOL	LEN	VALUE	DEFN	REFERENCES
DFPNDX	001	2900	9462	9411 9461
DFPOFF	001	28E3	9426	9386
DFPOGE	001	29DD	9543	9471
DFPORK	002	28E5	9427	9345* 9346 9451
DFPPCF	001	28DE	9421	9329 9342* 9344* 9345 9348 9349* 9352* 9353 9355* 9357* 9359* 9361*
				9430 9454 3742*
DFPPCH	002	28FD	9443	9314 9319 9399
DFPPCO	001	28E2	9425	3625* 3694
DFPPOS	001	4DE4	3554	3630* 3636 3656* 3657 3666 3673 3674* 3682* 3686* 3687
DFPRCK	001	28A5	9380	
DFPRCL	001	0002	9419	9381 9432
DFPRCT	002	28ED	9432	9381* 9486* 9491*
DFPRES	001	4DDC	3547	3596* 3604 3611* 3618 3799 3800
DFPRNT	001	2800	9313	9411 9418 3540 3570 3629* 3665*
DFPRPE	001	28D3	9409	9391 3540 3570
DFPRSN	002	29D9	9541	9468* 9477 9513
DFPSCK	001	2932	9479	
DFPSC2	001	2948	9490	9484
DFPSYC	001	28F9	9440	9434 9488* 9498* 9499 9500* 9502* 3741*
DFPULK	001	5339	3752	3735
DFPVCK	001	0004	9545	9513
DFPWITH	002	4DDB	3546	3597* 3598* 3602 3604 3611 3612
DFPX39	001	0039	9444	9320
DFPYCD	002	28F0	9434	9496
DFPYCT	001	0001	9446	9491*
DFP001	002	28E7	9428	9331 9349 9361 9445 9453 9476 9486 9491 9500
DFP100	004	2805	9315	9321 9401 9443
DFP101	004	280E	9317	9316*
DFP102	005	2812	9319	9402
DFP105	002	281F	9322	3755
DFP115	001	2823	9324	3744 3746
DFP120	003	283D	9336	9327
DFP140	004	2853	9342	9339
DFP160	005	2862	9348	9341
DFP180	003	2872	9353	9330 9351
DFP200	005	2878	9355	
DFP220	006	2888	9360	9358
DFP240	003	2892	9362	9332 9354 3497
DFP250	001	2899	9364	3697
DFP260	003	289C	9366	9384* 9520 3629* 3665*
DFP270	003	289F	9367	9368 9370 3496* 3529* 3619*
DFP280	003	28A2	9371	9382* 3442 3754*
DFP300	003	28CA	9402	9534 3533
DFP320	003	28AC	9383	9371
DFP330	003	28BB	9393	9394 9396 9398 9507* 3431* 3438* 3488* 3528*
DFP333	001	28BA	9392	3440* 3530*
DFP335	003	28B8	9387	9388 9390
DFP340	003	28B2	9385	9407
DFP360	003	28CD	9406	9385
DFP378	001	2911	9467	9464
DFP380	005	2927	9476	9470
DFP400	004	29B2	9525	9487 9492
DFP420	003	2953	9495	9489
DFP440	004	296F	9503	9501
DFP480	001	29CE	9533	9528
DLFBPT	001	4DDF	3549	9516* 9517 3622* 3689



## CROSS REFERENCE

VER 15, MOD 00 31/05/21 PAGE 313

SYMBOL	LEN	VALUE	DEFN	REFERENCES
DLFCAR	001	00FB	3715	3693
DLFDSV	004	28F4	9452	3786* 3789 3791 3800*
DLFEOR	001	4DF6	3573	3440
DLFIST	001	28F5	9450	3451 3466 3467 3485 3789* 3790 3798* 3799*
DLFMAR	002	4DE1	3551	
DLFORK	002	28E5	9451	9510* 9515 3646* 3647* 3648 3648* 3649 3649* 3650 3650* 3651 3651*
				3652 3652* 3657 3690* 3695
DLFPCF	001	28DE	9454	3495* 3531* 3635* 3675* 3684* 3687*
DLFPCH	002	4DED	3560	3471
DLFPC1	002	4DEF	3561	3519
DLFPRT	001	4D00	3430	9367 9389 9397 9515* 9516* 9517 9518* 9532 9537 3428 3560 3561
				3564 3565 3585 3732 3785
DLFRPE	001	4DCD	3536	9389 3572
DLFRTN	001	0001	3566	3787 3802
DLFRTY	002	4DE3	3552	3507
DLFSWC	001	4DF0	3562	3787 3794* 3802*
DLFVD1	002	4DD7	3543	3434 3587
DLFVD2	002	4DEB	3556	3489
DLFX4E	001	004E	3557	3757
DLFX53	001	0053	3558	3759
DLF001	002	28E7	9453	3634 3663 3686 3786 3793
DLF050	001	4D18	3437	
DLF100	001	4D25	3449	9397 3515 3565 3803
DLF140	001	4D3F	3462	3459
DLF143	004	4D4E	3472	3520
DLF145	004	4D57	3474	3473*
DLF146	003	4D5B	3478	3795
DLF150	005	4D5E	3479	3790* 3791* 3792* 3793*
DLF155	004	4D63	3481	3432*
DLF160	001	4D6D	3487	3494
DLF165	005	4D70	3489	
DLF170	001	4D78	3492	3452
DLF175	001	4DB9	3527	3486 3561
DLF350	003	4D88	3502	9518* 9537 3503 3505 3506* 3564
DLF355	005	4D8E	3507	
DLF360	003	4DA8	3515	3439* 3450* 3502 3516 3518
DLF375	003	4DB0	3520	
DLF400	001	4D93	3508	3490 3560
DLF425	004	4DA4	3513	3511*
DLF450	001	29AF	9519	9504 9514
DLF500	001	4E25	3601	3551
DLF525	001	4E29	3603	3599
DLF550	001	4E3C	3617	3605
DLF600	001	4E44	3621	3613
DLF700	001	4E49	3623	3552
DLF800	001	4EC7	3681	3668
DLF900	001	4ECC	3683	3658
DLF920	001	4ECF	3685	3676
DLF950	001	4ED7	3688	3637 3667
DLF960	004	539F	3790	3788
DLTABL	001	0090	3559	3684
DLTABR	001	00A0	3563	3675
FBSADA	008	009F	6502	6427
FBSATA	008	129F	6496	6502
FBSATN	004	1100	6340	6502
FBSAT1	008	1188	6413	6388



## CROSS REFERENCE

SYMBOL	LEN	VALUE	DEFN	REFERENCES	VER 15, MOD 00	31/05/21	PAGE 314
FBSBN1	001	1297	6492	6473			
FBSINS	005	128E	6490	6476			
FBSINZ	004	1285	6487	6426			
FBSLNR	001	0009	6404	6375	6383	6447	6452 6490
FBSLNT	001	128D	6489	6469			
FBSLNW	001	000B	6405	6375	6377*	6437	6437 6448 6448 6451 6451 6470 6470 6471 6472
				6481	6482		
FBSLST	001	12B6	6501	6490			
FBSMDS	004	1296	6491	6468			
FBSMDZ	004	128C	6488	6435			
FBSONE	007	1190	6415	6357	6362		
FBSRND	001	1180	6412	6389			
FBSRRR	001	0614	6406	6383	6447*	6452*	6490*
FBSSGN	001	117F	6411	6343*	6395		
FBSWWW	001	061F	6407	6375*	6377*	6437	6448* 6451* 6470
FBSZER	001	0005	6403	6375	6375		
FBSZZZ	011	1278	6481	6436*	6448	6451	6487*
FBS10Y	011	1283	6482	6471*	6472		
FBS100	003	1111	6349				
FBS110	003	111D	6356	6350			
FBS190	004	1148	6369	6352			
FBS200	005	114E	6375				
FBS400	004	120A	6435	6477			
FBS405	004	120E	6436	6426*	6428	6435*	6449
FBS420	005	1212	6437	6453	6475		
FBS425	005	1224	6447	6427*	6469*		
FBS430	005	1229	6448				
FBS440	005	1231	6451	6445			
FBS450	005	1236	6452	6446	6468*	6476*	
FBS600	004	123E	6457	6438			
FBS800	003	1166	6387	6351*	6356*		
FBS810	004	116D	6389	6387			
FBS900	004	117B	6399	6342			
FGSBN1	001	05CF	4953	4835	4856	4908	4935
FGSEVP	004	0500	4796				
FGSFVE	001	05D0	4954	4925			
FGSINL	001	0005	4948	4883	4915	4955	
FGSINS	006	05F5	4964	4883			
FGSITN	001	05FC	4969	4884*	4905	4908*	
FGSMNN	010	05E9	4958	4845			
FGSMOD	005	05D5	4955	4915			
FGSNNL	001	000A	4949	4956	4958	4960	
FGSNNN	010	05DF	4956	4826			
FGSONE	001	0001	4946	4826	4834	4854	4864 4866* 4893 4907 4960
FGSSFZ	002	0619	4989	4986			
FGSTEN	011	05F4	4960	4832	4834	4854	
FGSTHR	001	0003	4947	4826	4832	4834	4845 4854 4949
FGSXM1	001	05FB	4968	4831*	4835*	4853*	4856* 4935* 4938
FGS001	004	0600	4981				
FGS004	004	0614	4987	4983	4985		
FGS005	004	050C	4805	4827			
FGS010	004	0513	4813	4804			
FGS100	004	0529	4832	4836			
FGS110	004	053B	4845	4820			
FGS115	003	054C	4853	4846			
FGS120	004	054F	4854	4857			

## CROSS REFERENCE

VER 15, MOD 00 31/05/21 PAGE 315

SYMBOL	LEN	VALUE	DEFN	REFERENCES
FGS210	005	055D	4864	4833 4855
FGS220	005	057D	4893	4904 4916
FGS250	006	0585	4900	4883* 4915*
FGS260	003	058E	4905	4894
FGS300	004	05A7	4924	4906
FGS305	005	05BF	4936	4927
FGS900	004	05CB	4942	4806 4849
FKSADD	001	0002	4223	4193
FKSARG	008	037C	4346	4252* 4253* 4257* 4278* 4320* 4331 4332*
FKSCNT	008	0093	4368	4893 4907
FKSCNV	008	02A3	4228	4106* 4117* 4212
FKSCON	008	0393	4358	4287 4368
FKSDCR	001	02BC	4237	4176 4185
FKSINC	001	038A	4353	4300
FKSINS	006	036F	4342	4266
FKSINT	001	0005	4337	4266
FKSITN	001	0384	4348	4267* 4300* 4304
FKSLGT	004	0200	4106	4116 4125
FKSLOG	003	0219	4126	
FKSLTW	004	020B	4117	
FKSMDY	005	0389	4352	4326
FKSMOD	001	0005	4338	4326 4352
FKSONE	001	0001	4222	4172 4172 4175 4178 4180 4185* 4187* 4193* 4195 4196 4197 4250* 4251 4287 4309 4332*
FKSRND	001	038B	4354	4309
FKSSFT	001	0002	4224	4196*
FKSSHT	007	0383	4347	4331* 4332
FKSTEN	007	02AB	4233	4106
FKSTNE	008	02BB	4236	4175 4180
FKSTWO	007	02B3	4235	4117
FKS010	003	0212	4119	4108
FKS020	004	021F	4128	4120
FKS025	004	022F	4136	4107* 4118* 4127* 4133
FKS030	005	0236	4142	4135
FKS090	004	0300	4250	4368
FKS095	004	0321	4273	4272*
FKS100	005	0325	4278	4296 4333
FKS120	006	0332	4295	4266* 4326*
FKS150	004	033B	4300	4279
FKS175	005	0358	4320	4305
FKS205	003	024E	4173	4177
FKS210	003	025F	4178	4174 4186
FKS220	004	0270	4187	4179
FKS600	003	028D	4211	4119* 4126*
FKS700	004	0298	4218	4137 4211
FNBBN1	001	08EC	5256	5144
FNBCNT	001	08E0	5249	5141* 5144* 5150* 5151 5200
FNBDCL	001	08EE	5259	5174
FNBDGT	001	08E1	5250	5148* 5174* 5191
FNBFPL	001	08EE	5258	5131
FNBMK1	001	0002	5245	5131
FNBMN1	002	08EB	5255	5145
FNBPWR	004	0800	5102	
FNBSTRL	008	08E9	5251	5209* 5219
FNB005	003	0810	5121	
FNB010	003	081D	5128	5112

## CROSS REFERENCE

VER 15, MOD 00 31/05/21 PAGE 316

SYMBOL	LEN	VALUE	DEFN	REFERENCES
FNB030	003	082E	5141	5129
FNB200	003	0831	5142	5146
FNB250	004	0841	5148	5143
FNB275	004	0859	5159	5117
FNB300	003	0860	5166	5152
FNB350	004	0871	5174	5180
FNB400	003	088B	5188	5167 5169
FNB500	003	08A6	5208	5158 5189 5192 5201
FNB800	004	08A9	5209	5207
FNB880	003	08D6	5239	5202* 5208* 5233
FNB900	004	08DC	5241	5122 5124 5132 5160 5175 5179 5225 5235 5239
FRBACC	001	0001	5453	5424*
FRBBN1	001	09B4	5471	5384
FRBDC1	001	09B5	5472	5424
FRBEVN	001	0001	5452	5378*
FRBEXP	001	0002	5455	5392 5393 5394
FRBFC1	009	09AC	5461	5411* 5423
FRBFC2	007	09B3	5467	5406* 5407 5407* 5408 5409 5412* 5430 5439 5440 5440*
FRBLNG	001	0001	5456	5421* 5438*
FRBNRM	001	09B6	5473	5397
FRBONE	001	0001	5451	5390*
FRBSQR	004	0900	5360	
FRBSUB	009	09A3	5460	5410* 5421 5423* 5438 5439*
FRBTWO	001	0002	5454	
FRB005	003	0911	5365	5362
FRB010	003	0917	5373	
FRB020	004	0927	5384	5374
FRB030	004	0932	5390	5380
FRB100	004	0969	5421	5425 5442
FRB150	003	097B	5430	5422
FRB400	003	097E	5431	
FRB850	004	0993	5446	5431
FRB900	004	0997	5447	5364 5366
FSSADD	001	0003	5707	5646
FSSCOF	007	0B70	5782	5744
FSSCOS	004	0A00	5589	5583
FSSDCO	001	0B67	5776	5750
FSSEQ8	001	0001	5705	
FSSFP1	007	0AC8	5717	5628
FSSHLF	007	0AD6	5721	5668 5684
FSSINP	008	0B66	5772	5740* 5763
FSSINT	001	0003	5708	5650 5652 5652 5653 5655 5655 5718 5719
FSSLOP	001	0B5E	5771	5739* 5761*
FSSMDY	001	0AD8	5723	5689
FSSMN1	001	0B68	5777	5761
FSSMOD	001	0002	5706	5689
FSSOCT	001	0AC0	5712	5590* 5611* 5615* 5646 5657* 5685 5687 5692 5698
FSSONE	001	0001	5704	5689*
FSSRST	008	0B5D	5770	5744* 5760*
FSSSIN	004	0A1A	5605	
FSSSQD	008	0B55	5769	5743* 5751 5752
FSS008	003	0ACE	5719	5653 5655
FSS050	003	0A14	5598	5595
FSS064	003	0ACB	5718	5650 5652
FSS100	003	0A33	5616	5599
FSS150	003	0A36	5621	5614

## CROSS REFERENCE

VER 15, MOD 00 31/05/21 PAGE 317

SYMBOL	LEN	VALUE	DEFN	REFERENCES
FSS160	004	0A3C	5623	5598* 5612*
FSS200	004	0A43	5628	5622
FSS205	003	0A4B	5633	
FSS225	004	0A5B	5650	
FSS230	004	0A66	5653	5651 5656
FSS260	004	0A74	5657	5654
FSS300	004	0A81	5668	5634
FSS360	003	0A9D	5690	5699
FSS370	004	0AA0	5691	5700
FSS380	003	0AA4	5692	5688
FSS400	004	0AAD	5695	5693
FSS425	004	0AB3	5697	5610 5624
FSS450	003	0AB7	5698	5686
FSS900	004	0B00	5733	5696
FSS905	004	0B17	5744	5745
FSS910	004	0B1F	5750	5762
FSS920	004	0B2F	5758	5745* 5750*
FWSCOT	004	0D00	6189	
FWSLRG	001	0003	6229	6194 6205
FWSPCH	120	0DFB	6230	
FWSSAV	008	0D27	6199	6197* 6209* 6213 6222
FWSTAN	004	0D28	6203	
FWS005	003	0D10	6194	6191
FWS007	003	0D2F	6205	
FWS009	004	0D35	6207	6192* 6193 6195* 6196 6204*
FWS030	004	0D3C	6209	6206
FWS040	004	0D62	6219	6198
FWS900	004	0D80	6227	6208
FZRBAT	001	336D	1786	1706
FZRBN1	001	336B	1779	1769
FZRDCA	001	332F	1720	1789
FZRDDL	001	3349	1741	1790
FZREAD	001	3300	1683	1788 1789 1790
FZREOP	001	335F	1764	1788
FZRLDA	001	336C	1780	1732
FZR010	004	3307	1690	
FZR020	005	330F	1697	1686 1756 1771
FZR030	002	3319	1700	1697*
FZR040	003	331A	1704	
FZR050	004	3324	1707	1705*
FZR060	004	3328	1708	1704*
FZR070	003	332C	1712	1707*
FZR080	005	332F	1724	
FZR090	005	3340	1732	
FZR100	003	3349	1745	
FZR110	004	334F	1750	
FZR120	005	3357	1755	1746
FZR130	005	335F	1769	
FZSBN1	001	34DF	2195	2068 2087 2129
FZSCAJ	001	34E1	2198	2126
FZSCAT	001	34E5	2216	2167
FZSCNT	004	34C7	2244	2038* 2061* 2068* 2094* 2104* 2134* 2144* 2159
FZSDAC	002	35D8	2416	2367* 2371 2373 2373*
FZSDC1	001	35CB	2400	2367
FZSDC5	001	35CC	2401	2277
FZSEXB	004	35D1	2406	2352

## CROSS REFERENCE

SYMBOL LEN VALUE DEFN REFERENCES VER 15, MOD 00 31/05/21 PAGE 318

FZSLXB	001	0004	2405	2347	2352	2385	2387	2406	2413	
FZSLXM	001	0002	2415	2361*	2367	2371	2371	2373	2373	2416
FZSNXZ	001	34E0	2197	2081						
FZSPAL	001	0000	2242	2108*	2332*	2338*	2343*			
FZSPCH	002	36FB	2701	2484						
FZSPDA	002	34E4	2201	2177						
FZSPRT	001	3400	2033							
FZSP1B	001	3400	2024	2028						
FZSP2B	001	3500	2267	2271						
FZSP3B	001	3600	2440	2444	2701	3764				
FZSP4B	001	3700	2725	2729						
FZSSAJ	001	34E2	2199	2141						
FZSXWK	004	35D6	2413	2352*	2361*	2371*	2378	2380	2380*	2387
FZS010	003	3404	2038							
FZS020	004	340E	2045							
FZS030	003	3415	2056							
FZS035	003	3418	2057							
FZS040	003	3424	2061	2058						
FZS050	003	3427	2066							
FZS060	003	3434	2073	2067						
FZS070	004	3440	2080							
FZS080	003	344B	2085	2082*	2087*	2088	2089	2094		
FZS090	003	3455	2088	2080*	2081*					
FZS100	004	345B	2094							
FZS110	004	3462	2101	2074	2076	2086				
FZS120	003	346D	2108	2056*	2059*	2069	2095			
FZS130	003	3473	2120	2046						
FZS140	005	3479	2126							
FZS150	004	3481	2129	2132						
FZS155	003	3488	2131	2128*	2129*	2134				
FZS160	005	3495	2141	2121						
FZS170	003	34A1	2148	2136						
FZS180	003	34A4	2158	2041	2110					
FZS190	004	34B1	2162	2130	2158*	2159*				
FZS2BX	001	35D2	2412	2354*	2355*	2362*	2363*	2369		
FZS2B1	001	35CA	2399	2280	2304	2311	2333	2340	2381	
FZS2XZ	001	35CD	2403	2305	2355					
FZS200	004	34B5	2166	2160						
FZS210	005	34C1	2169	2168*						
FZS230	004	34C6	2173	2244						
FZS240	004	34D5	2182							
FZS260	004	34DB	2187							
FZS3B2	001	36F0	2673	2516						
FZS3CC	003	36DB	2681	2465*	2495*	2508*	2510	2512*	2513	2516*
				2572	2593*	2603*	2647*	3765*	3766	2524*
										2525
										2531*
										2532
										2570*
FZS3CR	001	36F6	2695	2584	2653					
FZS3PA	001	36F5	2692	2460*						
FZS3PC	001	36F3	2691	2459*	2465	2482	2510	2532*	2543	2557*
FZS3PF	001	36F2	2690	2458*	2548*	3772*	3776*			2566*
FZS3PL	001	36F2	2684	2618	2690	2691	2692	3773		2570
FZS3PZ	001	36F1	2675	2512						2614*
FZS3RM	003	366A	2680	2454*	2572	2648	3766			
FZS300	001	3500	2276	2102						
FZS310	003	350E	2284	2278						
FZS320	003	351A	2295							
FZS330	004	3524	2298	2295*	2296*					

## CROSS REFERENCE

VER 15, MOD 00 31/05/21 PAGE 319

SYMBOL	LEN	VALUE	DEFN	REFERENCES
FZS340	004	3528	2303	2297
FZS350	003	3534	2306	2303* 2304* 2305*
FZS360	003	3537	2310	
FZS370	004	353A	2311	2313
FZS380	003	353E	2312	2310* 2311* 2317
FZS390	005	3544	2317	
FZS4B2	001	37DE	2951	2799
FZS4CC	003	37C6	2960	2749* 2778* 2791* 2793 2795* 2796 2799* 2807* 2808 2814* 2815 2853* 2855 2867 2876* 2886* 2921*
FZS4CR	001	37E4	2974	2927
FZS4PA	001	37E3	2971	2744*
FZS4PC	001	37E1	2970	2743* 2749 2766 2793 2815* 2826 2840* 2849* 2853 2897*
FZS4PF	001	37E0	2969	2742* 2831*
FZS4PL	001	37E0	2963	2901 2969 2970 2971
FZS4PZ	001	37DF	2953	2795
FZS4RM	003	3764	2959	2738* 2855 2922
FZS400	004	354D	2331	2285 2287
FZS410	003	3558	2338	
FZS420	004	355E	2340	2342
FZS430	003	3562	2341	2339* 2340* 2348 2386
FZS435	003	3568	2343	
FZS440	004	356B	2347	
FZS450	004	3574	2352	
FZS460	003	3586	2361	
FZS470	004	3590	2367	2357
FZS472	003	3597	2369	2368* 2372 2372* 2374
FZS474	004	35A1	2372	2370
FZS480	003	35AC	2378	2356
FZS490	003	35BB	2385	2379
FZS500	004	35C2	2387	2385* 2386*
FZS510	004	35C6	2391	
FZS600	006	3600	2449	2183
FZS605	003	362B	2472	2471*
FZS610	003	362E	2482	2218 2218 2219 2220 2221 2223 2224 2225 2226 2228 2228 2229 2230 2230 2231 2233 2233 2234 2235 2235 2236
FZS615	004	3642	2487	2486*
FZS620	003	3646	2495	2219
FZS630	003	364C	2508	2220
FZS632	004	364F	2510	2514
FZS633	003	365A	2513	2701
FZS634	004	3660	2516	2511
FZS636	005	3664	2524	2496
FZS638	003	3669	2525	2680
FZS640	005	366F	2531	
FZS650	003	367B	2543	2221 2231 2236
FZS655	003	3681	2548	2526
FZS660	003	3687	2557	2223
FZS670	003	368D	2566	2224
FZS675	004	3690	2570	2558
FZS680	003	36A1	2584	2225 2544
FZS690	003	36A7	2593	2226
FZS695	003	36B0	2603	2229
FZS700	003	36B6	2614	2234
FZS710	003	36B9	2618	2533 2549 2573 3768
FZS720	003	36BC	2619	2585
FZS730	006	36BF	2623	2595



## CROSS REFERENCE

VER 15, MOD 00 31/05/21 PAGE 320

SYMBOL	LEN	VALUE	DEFN	REFERENCES
FZS740	004	36C8	2628	2450
FZS750	004	36CE	2633	2574 2624
FZS760	003	36D2	2643	2467 2571 2594 2604
FZS770	003	36DA	2648	2681
FZS780	003	36E3	2660	2619
FZS790	004	36EC	2666	2483 2643* 2649 2660*
FZS800	005	3700	2734	2629 3780
FZS805	003	3725	2756	2755*
FZS810	003	3728	2766	
FZS820	003	3740	2778	
FZS830	003	3746	2791	
FZS832	004	3749	2793	2797
FZS834	004	375A	2799	2794
FZS836	005	375E	2807	2779
FZS838	003	3763	2808	2959
FZS840	005	3769	2814	
FZS850	003	3775	2826	
FZS855	003	377B	2831	2809
FZS860	003	3781	2840	
FZS870	003	3787	2849	
FZS875	004	378A	2853	2841
FZS880	003	379B	2867	2827
FZS890	003	37A1	2876	
FZS895	003	37AA	2886	
FZS900	003	37B0	2897	
FZS910	003	37B3	2901	2767 2816 2832 2856
FZS920	003	37B6	2903	2868
FZS941	004	348E	2134	
FZS950	004	37B9	2907	2768 2857 2878
FZS960	003	37BD	2917	2751 2854 2877 2887
FZS970	003	37C5	2922	2960
FZS980	003	37CE	2935	2903
FZS982	004	37D4	2938	2734*
FZS984	002	37D9	2939	2937*
FZS990	004	37DA	2943	2917* 2923 2935*
FZS991	005	5359	3765	
FZS992	004	5368	3770	3767
FZS993	003	5372	3773	3771
FZS994	004	538D	3781	3778
FZXBCA	001	0DC8	1504	9978* 9983 9996 0042
FZXBKT	001	31CF	1231	1235 1236 1237 1239
FZXBLK	001	31E2	1233	1126
FZXBLN	002	2CE5	0291	0141* 0145 0150 0150*
FZXBPT	001	00FF	1506	0499* 0500* 0501 1079 1200*
FZXBVA	002	2B91	0064	9976 0053
FZXB10	001	32E6	1480	1395
FZXCNT	001	0D56	0658	0494* 0507*
FZXCNV	001	3100	1076	
FZXCRP	001	2CFB	0320	0222
FZXCRR	001	31E1	1237	1126* 1141 1158
FZXCR1	001	31D0	1236	1127
FZXDAC	004	2CE9	0292	0143* 0147 0152 0152*
FZDXLN	004	2CED	0295	0142* 0147*
FZXDTC	001	2EF9	0650	0492* 0593* 0600* 0604 0607*
FZXDTM	001	0080	0651	0604 0607
FZDXDL	001	0002	1033	0946 1034 1487



## CROSS REFERENCE

VER 15, MOD 00 31/05/21 PAGE 321

SYMBOL	LEN	VALUE	DEFN	REFERENCES
FZXECA	004	2C68	0277	0226
FZXELN	001	2CEA	0294	0180
FZXER0	001	00F0	1508	0531 0603 0706
FZXER1	001	00F1	1509	0606 0742
FZXER2	001	00F2	1510	0544 0612
FZXER3	001	00F3	1511	0484
FZXER4	001	00F4	1512	0472
FZXETS	001	0D58	0653	0453* 0486* 0696 0697
FZXEVA	002	2CE0	0276	0161 0199
FZXEXP	001	32EA	1493	1305* 1331* 1346* 1402* 1404* 1411*
FZXGCS	001	2E00	0449	
FZXICA	001	0003	1249	1186
FZXICB	001	31D0	1239	1091* 1096* 1173* 1180
FZXICC	001	2FB6	0832	0775*
FZXICL	001	0004	0818	0769 0774 0782 0784 0786 0821 0824 0833
FZXICN	001	0001	1248	1180
FZXICR	004	2FB9	0834	0776
FZXICT	001	31E3	1251	1178
FZXICW	004	2FB9	0833	0769 0769* 0774 0774* 0782 0784 0786 0834
FZXIEX	003	2FAD	0821	0782
FZXIPI	002	2FB1	0824	0784
FZXIP2	001	2B6E	0038	
FZXIS2	004	2FB5	0826	0786
FZXITL	001	0004	1247	1178 1179
FZXLVA	001	2B8F	0063	0006
FZXMIS	001	2E17	0467	
FZXMNR	001	32F1	1495	1306* 1348 1418
FZXMN1	001	32EB	1494	1307 1409
FZXPDA	002	2CDE	0274	0235
FZXPEM	001	2C18	0137	
FZXPNP	001	2CF7	0313	0171
FZXPQ1	001	2C00	0107	
FZXPQ2	001	2C06	0116	
FZXPRP	001	2CEF	0299	0120* 0126
FZXPSA	002	2B56	0012	0009
FZXPSP	001	2CF3	0306	0181* 0182
FZXP1B	001	2B00	9959	9963
FZXP2B	001	2C00	0092	0096
FZXP3B	001	2D00	0337	0099 0276 0341
FZXP4B	001	2E00	0436	0440
FZXP5B	001	2F00	0687	0691
FZXP6B	001	3000	0856	0860
FZXP7B	001	3100	1063	1067 1127 1141
FZXP8B	001	3200	1296	1300 1307 1348
FZXQML	001	0002	0283	0301
FZXQM1	001	2CE2	0281	0109
FZXQM2	001	2CE1	0279	0118
FZXSEC	001	00FF	0656	0454 0791 0986
FZXSER	001	0D59	0655	0493* 0514 0791* 0986*
FZXSGN	001	32F1	1496	1367* 1412*
FZXSTC	001	2EF8	0648	0491* 0609*
FZXSTP	005	2EC0	0647	0490* 0601*
FZXSTS	001	31CF	1235	1125* 1144*
FZXXCL	001	0002	1022	0865 0887 0901 0967 0967* 0969 0979 0981 1023 1024 1025 1031
FZXXCT	002	30EA	1031	0865* 0887* 0901* 0969* 0979 0981
FZXXHI	002	30E6	1024	0979

## CROSS REFERENCE

VER 15, MOD 00 31/05/21 PAGE 322

SYMBOL	LEN	VALUE	DEFN	REFERENCES
FZXXLO	002	30E8	1025	0981
FZXXZR	002	30E4	1023	0865
FZX010	004	2B1E	9989	0029
FZX020	004	2B57	0017	0004
FZX030	004	2B8B	0057	0051
FZX050	003	2C00	0109	9990
FZX060	003	2C06	0118	
FZX070	003	2C09	0120	0110
FZX080	005	2C18	0141	0027
FZX090	003	2C28	0145	0144* 0148 0148* 0151* 0153
FZX100	004	2C32	0148	0146
FZX110	004	2C40	0152	0149
FZX120	004	2C66	0176	0164* 0188 0277
FZX130	003	2C82	0190	0192
FZX140	004	2C9D	0204	0129
FZX150	003	2CA1	0222	0124 0157
FZX160	004	2CA4	0226	0169 0178 0193
FZX170	003	2CA8	0231	0127 0172 0183
FZX180	004	2CBF	0247	0235* 0236* 0240 0252
FZX190	002	2CC4	0248	0245*
FZX2D0	001	2CDB	0271	0142
FZX2D1	001	2CDC	0272	0143 0191
FZX200	004	2CCD	0257	0241
FZX210	004	2CD7	0263	0231* 0253
FZX250	004	2E00	0453	
FZX260	003	2E17	0471	
FZX270	003	2E20	0483	0018
FZX280	004	2E26	0486	0473
FZX290	005	2E30	0490	0459
FZX300	003	2E4F	0505	0522
FZX310	005	2E55	0507	0529
FZX320	003	2E6D	0522	0455* 0471* 0483*
FZX330	003	2E86	0538	0520
FZX340	003	2E93	0545	0456* 0487*
FZX350	004	2E96	0548	0472* 0484*
FZX360	004	2EA4	0557	0532 0545
FZX370	003	2E9A	0551	0458* 0515 0534 0539 0559*
FZX375	003	2EA8	0559	0551 0553
FZX380	003	2EAF	0584	0505 0526 0543
FZX390	003	2EB2	0585	0457* 0488*
FZX4B1	002	2EF7	0641	0500 0507 0593 0601 0609
FZX400	004	2EB5	0593	0610
FZX410	005	2EBC	0600	0647
FZX420	004	2ED4	0609	0605
FZX430	004	2EDE	0616	0594 0603* 0606* 0612*
FZX440	004	2EE2	0618	0584* 0585
FZX450	003	2EE6	0627	0506 0527
FZX460	003	2EE9	0629	0631
FZX470	004	2EF2	0633	0627*
FZX5M1	002	2FA9	0816	0727
FZX500	005	2F00	0696	0512
FZX510	003	2F14	0707	0696*
FZX520	003	2F20	0715	0713
FZX530	004	2F35	0742	0702
FZX540	003	2F39	0743	0697*
FZX550	003	2F3C	0748	0752 0755

## CROSS REFERENCE

VER 15, MOD 00 31/05/21 PAGE 323

SYMBOL	LEN	VALUE	DEFN	REFERENCES
FZX560	004	2F5D	0769	0717
FZX570	003	2F61	0771	0777
FZX580	004	2F7B	0782	0773
FZX590	004	2F90	0791	0707 0743 0750
FZX6BX	002	30EC	1035	0967 0969
FZX6B1	002	30E2	1020	0887 0901
FZX6DX	002	30EC	1034	0946* 0949 0949* 0950* 0955* 0956* 0957 0957* 0958 0958* 0959 0959*
				0960 0960* 0961 0961* 0962* 1035
FZX600	004	2F94	0793	0732 0759 0783 0785 0787
FZX610	003	2F98	0802	0715 0758 0771 0778
FZX620	003	2F9B	0804	0806
FZX630	004	2FA4	0808	0802*
FZX650	004	3000	0865	0730
FZX660	003	300D	0878	0879
FZX670	003	3016	0884	
FZX675	004	301C	0887	0889
FZX680	003	3026	0893	0880
FZX690	003	302F	0900	0866* 0884* 0903
FZX7B1	003	31C4	1225	1144 1146
FZX700	003	3036	0902	0895 0900
FZX710	003	3042	0910	0911
FZX720	003	3048	0915	0867* 0894 0904 1010*
FZX730	003	3066	0937	0933
FZX740	003	3069	0941	0935
FZX750	003	3084	0955	0948
FZX780	003	30A1	0966	0930* 0936*
FZX790	004	30AB	0969	0966
FZX8BK	001	32EA	1490	1493 1494 1495 1496
FZX8BX	002	32E9	1488	1402 1404
FZX8B1	003	32D8	1479	1331 1346 1351 1396
FZX8DX	002	32E9	1487	1386* 1387 1387* 1388* 1395* 1396* 1488
FZX8D0	001	32E7	1481	1306
FZX800	003	30AF	0974	0868* 0885* 0908* 0926 0968
FZX810	004	30C0	0986	0915 0942 0980
FZX820	004	30C4	0988	0974 0982
FZX830	003	30C8	1002	0878 0888 0902 0910 0931 0937 0947 0951
FZX840	003	30CB	1004	1006
FZX850	004	30DD	1012	1002* 1009
FZX860	003	3100	1078	0045
FZX863	003	3124	1098	1093
FZX866	003	3127	1099	1095
FZX870	004	312D	1110	1090*
FZX873	003	313A	1125	1085
FZX876	003	3144	1131	1142 1147
FZX880	003	3156	1141	1133
FZX883	004	3160	1145	1127* 1141 1146*
FZX886	003	316B	1151	1136
FZX890	004	317C	1159	1156*
FZX893	003	3183	1172	1100
FZX896	003	3190	1179	1181
FZX900	004	31A7	1192	1174*
FZX903	003	31AB	1193	1195
FZX906	004	31B4	1199	1078* 1115 1160
FZX910	003	31BF	1211	1080 1098 1152 1172 1193
FZX913	003	31C2	1213	1215 1225
FZX916	004	31CB	1217	1211*

## CROSS REFERENCE

VER 15, MOD 00 31/05/21 PAGE 324

SYMBOL	LEN	VALUE	DEFN	REFERENCES
FZX920	003	3200	1305	1113
FZX923	003	320D	1319	1320
FZX926	003	3213	1324	
FZX930	004	321F	1331	1333
FZX933	003	3223	1332	1330
FZX936	003	3229	1337	1361
FZX940	003	322C	1338	1326
FZX943	003	3232	1345	1324* 1337* 1354
FZX946	003	3239	1348	1345
FZX950	004	323F	1350	1307* 1348 1351*
FZX953	003	3247	1353	1349
FZX956	003	3259	1365	1339 1359
FZX960	003	3262	1377	1366
FZX963	004	3271	1387	1390
FZX966	004	3282	1395	1397
FZX970	004	3286	1396	1391
FZX973	003	328D	1401	1365 1446* 1451*
FZX976	004	3297	1404	1401
FZX980	003	329B	1409	1378 1403
FZX984	003	32A7	1416	1410
FZX986	004	32B6	1423	1416*
FZX990	003	32BE	1444	1308 1381
FZX992	003	32D3	1464	1319 1332 1353 1360 1380 1389
FZX994	003	32D6	1466	1448 1468 1479
FZX996	003	32DF	1470	1450
FZX998	004	32E2	1471	1444* 1464*
FZZBM1	001	00FF	3314	3194
FZZBN1	001	4CB3	3286	3160 3196 3270
FZZCDT	002	4CB8	3291	3216 3217
FZZCNT	002	4CBC	3299	3216* 3217* 3230 3239 3245
FZZDPL	001	4CBD	3302	3140* 3146* 3174 3208* 3209* 3218* 3220* 3230* 3234 3234* 3235 3235*
				3241* 3247* 3257* 3258* 3262
FZZHCA	002	4CBA	3297	3152* 3161* 3257
FZZIDM	001	0080	3320	3239
FZZITM	001	0040	3321	3245
FZZLOK	001	0001	3317	3188
FZZLRT	001	0000	3316	3180 3182* 3188
FZZMDY	001	0002	3318	3180 3182
FZZNST	001	4CB6	3290	3218 3220
FZZPGB	001	4C00	3127	3131
FZZSDM	001	0001	3322	3241
FZZSTM	001	0080	3323	3247
FZZSXA	002	4CB5	3288	3152
FZZVPL	001	4C06	3145	
FZZVPS	001	4C00	3139	
FZZ005	003	4C09	3151	3141
FZZ010	004	4C25	3169	3157 3271
FZZ020	003	4C29	3170	3151* 3159* 3160* 3197 3258 3270*
FZZ025	003	4C3E	3188	3175
FZZ030	003	4C44	3194	3183
FZZ035	004	4C4B	3196	3198
FZZ040	004	4C4F	3197	3194* 3196* 3209
FZZ050	004	4C61	3217	3219
FZZ060	003	4C85	3245	3240
FZZ070	004	4C8E	3257	3246
FZZ080	002	4CA1	3265	3263*

CROSS REFERENCE																					
SYMBOL	LEN	VALUE	DEFN	REFERENCES												VER 15, MOD 00		31/05/21		PAGE 325	
FZZ090	004	4CA2	3270	3181	3189																
I\$ADJX	001	0D56	3717																		
I\$ADST	001	0C9D	3672																		
I\$BASE	001	0C60	3674	0010																	
I\$BRCN	001	117B	3726																		
I\$BSET	001	119D	3725	0011																	
I\$B1SW	001	0040	3782																		
I\$B2SW	001	0020	3784																		
I\$CADR	001	144C	3763	7310*	7311	7362	7398	7570	7702	7746	7767*	7795	7808	7809	7847						
I\$CALL	001	12B1	3757	7970	8140	9316	9510	9978	0163	2486	3436	3473	3511	3589	3590						
				4152	4160	4205	4500	4511	4796	4813	5213	5230	5638	5663	5695						
				6210	6214	6219	6223	6369	6381	6393	6838	6881	6903	6920	6942						
				6964	7003	7020	7440	7605	7619	7661	7765	7971	8177	8244	8249						
				8261	8304	8366	8552	8564	8665	8733	9171	9989	9997	0017	0026						
				0044	0258	0511	0729	1112	2101	2182	2628	2661	3774	3779							
I\$CBM1	001	0D43	3693																		
I\$CBN1	001	0D3E	3689																		
I\$CBN2	001	0D3F	3690																		
I\$CBN3	001	0D40	3691																		
I\$CBN4	001	0D41	3692																		
I\$CFBS	001	0AE3	3740																		
I\$CLFA	001	0D4A	3699																		
I\$CLVA	001	0D49	3698	6843	7247																
I\$CL1C	001	0D46	3696	6820	7367																
I\$CL1F	001	0D44	3694	6808	7301																
I\$CL2C	001	0D47	3697																		
I\$CL2F	001	0D45	3695																		
I\$CPG1	001	1600	3654																		
I\$CPUF	001	0A27	3736	7603	7900																
I\$CSCT	001	0D5A	3712																		
I\$CSSW	001	0010	3786																		
I\$CSXA	001	2000	3653	0274	2201	3288															
I\$CUPF	001	0A85	3738	6811	7303	1419															
I\$CVAD	001	1358	3751	7397	7794	9315	2485	3472	3510												
I\$DATA	001	0D53	3680	6987*	1685	1697	1732*	1755*	1769*	1770*											
I\$DAT1	001	0D55	3681	6987																	
I\$DMSW	001	0BC1	3734																		
I\$ECSW	001	0004	3790																		
I\$ERRC	001	0CBC	3679	4136*	4805*	5123*	5159*	5178	5224	5232	5234*	5363*	5623*	6207*	6856						
				7257*	7267*	7291*	7380*	7384*	7402*	7578*	7742*	8148*	8256*	8404*	0023						
				0191*	0195*	0316	0454*	0531	0544	0548*	0552*	0557*	0616*	0706	0742						
				1690*	1750*																
I\$FACT	001	0DD1	3719																		
I\$FADD	001	075D	3742	5691	5759	6359															
I\$FATE	001	0DE6	3720	7335																	
I\$FATP	001	0DE8	3721	7247*	7248	7262	7272														
I\$FDVD	001	0919	3747	6217	6226	6365															
I\$FMPY	001	082A	3745	4213	5177	5223	5629	5669	5742	5753	5764										
I\$FSUB	001	0751	3743	6363																	
I\$FWRK	001	0607	3663	4142*	4193	4195*	4196	4250*	4251	4251*	4252	42									

CROSS REFERENCE																
SYMBOL	LEN	VALUE	DEFN	REFERENCES	VER 15, MOD 00 31/05/21 PAGE 326											
I\$IMPT	001	0DCC	3709													
I\$INDR	001	0DC5	3705													
I\$INIT	001	0607	3662													
I\$INTR	001	0C5C	3666													
I\$IRSW	001	0CDE	3686													
I\$I700	001	0E24	3748	9979												
I\$LBFR	001	12B6	3758	3509*												
I\$LDBR	001	1329	3755	8917												
I\$LDXR	001	1330	3756	4876	7959	7984	8164	8319	8372	8507	8641	8939	9410	9511	9529	
				1699	3537	3747										
I\$LOCK	001	1354	3753	7361	7568	8139	8170	8664	8919	8941	9509	9977	0162	3512	3588	
				3736	3749											
I\$MDFY	001	1349	3752	7569	7695	7842	7987	8138	8169	8707	3435					
I\$MOD4	001	1308	3749	0009*												
I\$NCPG	001	000A	3774													
I\$NDSW	001	0002	3792													
I\$NISW	001	0080	3780													
I\$NPAG	001	0C68	3667													
I\$PARM	001	0D57	3682	6769*	7598*	7604*	0491	0658	2040	2104	2126*	2141*	2168	2169*	2173*	
				2317*	2347*	2348*	2381*	2459	2471	2743	2755					
I\$PGDS	001	144A	3761													
I\$PGNO	001	1449	3760													
I\$PGTB	001	14CA	3764	3195												
I\$PLRT	001	15E2	3765	3169												
I\$PSTK	001	15CA	3766	0012												
I\$PUB1	001	0DC8	3707	1504												
I\$PUB2	001	0DCA	3708													
I\$RESW	001	0CE9	3687	0008*												
I\$RNMK	001	0001	3702													
I\$RNSW	001	0D5C	3701													
I\$RTRN	001	12D3	3759	4218	4316	4513	4668	4942	4987	5241	5447	5697	5765	5933	6063	
				6218	6227	6399	6458	7029	7258	7268	7292	7321	7446	7613	7743	
				7848	8012	8268	8410	8546	8572	8711	8743	8952	0024	0057	0204	
				0560	0793	0988	1202	1425	1691	1733	1751	2187	2319	2391	2633	
				2907	3278	3761	3781									
I\$SDCT	001	0D59	3714													
I\$SDPT	001	0DD0	3711													
I\$SFCT	001	0D5A	3715													
I\$SFFO	001	0D5D	3723													
I\$SICT	001	0D5B	3716													
I\$SLLC	001	0BA1	3730	6803	2045	2466	2750									
I\$SLNG	001	0BA2	3729	7281*	1725*											
I\$SNSW	001	0001	3794													
I\$SSCT	001	0D58	3713													
I\$STAK	001	0D4E	3675	4128	4311	4484	4625	4653	4664	4924	4981	5102	5149	5360	5446	
				5589	5605	5733	5897	5928	6040	6189	6203	6340	6808*	6810	6820*	
				6843*	6844	6899*	7280	7301*	7302	7367*	7368*	7369	7373*	7389*	7390*	
				7391	7392*	7600	7632	7654	7708	7892	8326	8380	8662	0006*	0490	
				1157	1187	1417	1727	2034								
I\$STCK	001	0B50	3728	7282	1188	1728										
I\$STHA	001	0D51	3685	0007												
I\$STKB	001	0639	3664													
I\$STKI	001	0D4F	3676													
I\$STSW	001	0008	3788													
I\$TFSW	001	0D28	3688													
I\$ULNG	001	0C3A	3733	6852*												



## CROSS REFERENCE

SYMBOL	LEN	VALUE	DEFN	REFERENCES	VER 15, MOD 00	31/05/21	PAGE 327
I\$UNLK	001	1350	3754	7419 7443 7612 8253 8267 8409 8710 8948 8950 0054 0200 3756			
I\$USTK	001	0BB0	3732	3758 3760 6857 7304			
I\$VADR	001	144A	3762	6845* 7241* 7252 7273 7300* 7360* 7396* 7418* 7442* 7567* 7611* 7693*			
				7694* 7793* 7840* 7841* 8137* 8252* 8266* 8408* 8696* 8706* 8947* 8949*			
				9314* 9319* 9320* 9399* 9400* 9508* 9976* 0053* 0161* 0199* 1186* 1724*			
I\$WRK1	001	0D59	3683	2484* 3434* 3471* 3489* 3507* 3519* 3587* 3755* 3757* 3759*			
				7320* 7362* 7425 7433* 7653* 7654* 7810* 7812* 7894 8469* 0007* 0653			
				0655 2166* 2460 2744			
I\$WRK2	001	0D5B	3684	7435* 7811* 2177* 2178* 2449 2623 2734 3777			
I\$XAD1	001	0C89	3671	7025			
I\$XAD2	001	0C82	3670	6969			
I\$XAD3	001	0C7B	3669	6861			
I\$XAD4	001	0C74	3668				
I\$XERR	001	0CAB	3673				
I\$XIAR	001	0D4C	3678	6758 6768 7240 7313			
I\$XPAG	001	0C61	3677	7312			
I@APRC	001	0006	3856	2075 2277 2277* 2286 2295 2298 2298* 2310 2331 2331 2331* 2339			
I@APRL	001	000B	3833				
I@APRS	001	0006	3810	3856			
I@ASTA	001	0000	3868				
I@ASTL	001	0020	3844				
I@ASTS	001	0000	3821	3868			
I@CMEQ	001	0004	3925				
I@CMHI	001	0008	3926				
I@CMLO	001	0002	3924				
I@DEXP	001	0000	3903	4485 4498 4499* 4510* 4654* 4667* 4803 4847* 4937* 4938* 4984 5166			
				5373 5390 5391* 5395* 5396* 5397* 5594 5597* 5609 5621 5633 6194			
				6205 6341 6349 2073 2075 2080 2280* 2284 2286 2296 2303 2333*			
				2354 2363			
I@ICBA	001	F500	3870	1255 1259 1263 1267 1271 1275			
I@ICBL	001	F000	3846				
I@ICBS	001	F500	3823	3870			
I@IVBA	001	F531	3871				
I@IVBL	001	F049	3847				
I@IVBS	001	F531	3824	3871			
I@LCRF	001	0012	3885	3886 7639 7645 7645 7645* 1126 1237 2128			
I@LCRV	001	0013	3886	6803 6852 7727 8329 1158 1158* 1232 1725 2045 2466 2750			
I@LFPZ	001	0012	3955	2158 2161* 2162 2162* 2495 2513 2557 2593 2603 2614 2778 2796			
				2840 2876 2886 2897			
I@LPFL	001	0009	3835	3838 3839 7720 7787 8415 8419			
I@LPFS	001	0005	3812	3815 3816 3858 7713 7787 8309 8415 8419			
I@LPFV	001	0005	3858	7333 1255 1259 1263 1267 1271 1275			
I@LPPZ	001	0003	3954	2508 2566 2675 2791 2849 2953			
I@LPSW	001	0080	3843				
I@LSFV	001	0008	3860	5410* 5411* 5412* 5421 5423 5438 5439 5439 5456			
I@LUFL	001	0010	3836	3840			
I@LUFS	001	0008	3813	3817 3859			
I@LUFV	001	0008	3859	3860 3895 3897 3898 3900 3901 4106 4117 4142* 4142* 4172 4175			
				4175 4180 4180 4193 4193 4195 4195* 4196 4196 4212 4228 4236			
				4250* 4251 4251 4251* 4252 4257 4257 4257 4278 4278 4278 4278			
				4287 4287 4287* 4295 4295 4295* 4295* 4309 4309* 4310 4320 4320			
				4320 4320 4327 4342 4342 4342 4342 4342* 4342* 4346 4358 4359			
				4360 4361 4362 4363 4364 4365 4366 4651 4832 4834 4834 4854			
				4854 4864 4864* 4864* 4893 4893 4893* 4893* 4900 4900 4907 4907			
				4907* 4907* 4909 4925 4964 4964 5173 5176 5209 5219 5251 5407			



CROSS REFERENCE																		
SYMBOL	LEN	VALUE	DEFN	REFERENCES												VER 15, MOD 00	31/05/21	PAGE 328
				5407	5407	5409	5421	5423	5438	5440	5460	5461	5628	5668	5684			
				5740	5741	5743	5744	5751	5752	5758	5760	5763	5769	5770	5772			
				5898	5898*	5900	5926	5929	6062	6197	6199	6209	6212	6212*	6212*			
				6213	6216	6216	6216	6221	6221*	6221*	6222	6225	6225	6225	6357			
				6358	6358*	6358*	6360	6360*	6360*	6360*	6361	6361	6361	6362	6364			
				6364	6364	6364	6383	6383	6388	6388	6388	6389	6389	6404	6405			
				6406	6406	6407	6407	6407	6413	6436	6447	6457	6487	6487	6489			
				6496	6497	6498	6499	1418	1418*	1491								
I@LXPT	001	0060	3946															
I@MANL	001	0001	3904	4132	4171*	4175*	4641	4986*	5111	5116	5128	5131*	5365	6190	2066			
				2279*														
I@MANR	001	0007	3905	5121	5142	5148	5157	5188	5190*	5240*								
I@NCPG	001	000A	3948	3151														
I@NERR	001	0000	3957	5178	5224	5232	6856	0023	0195	0493	0514	0557						
I@NXPG	001	0020	3945	3946														
I@NXPT	001	0003	3944	3946														
I@PEXL	001	0008	3839	7717	7723*	7724	7724*	8390*	8391	8391*	8399							
I@PEXP	001	0004	3863															
I@PEXS	001	0004	3816	3863	7717*	7723	8390	8399*										
I@PMNR	001	0003	3862															
I@PMN1	001	0000	3882															
I@PMRL	001	0007	3838															
I@PMRS	001	0003	3815	3862														
I@PRCL	001	000F	3832	3836														
I@PRCS	001	0007	3809	3813	3855													
I@PREC	001	0007	3855	4142	4193	4233	4235	4252	4253*	4257	4304	4310*	4331	4332	4347			
				4489	4628	4636	4663	4663*	4665	4665	4826	4845	4848	4848	4865*			
				4866	4866	4866*	4900*	4905	4925*	4936	4936	4949	4964	4964*	5130			
				5130	5151	5173*	5176	5200	5378	5385	5408	5440	5456	5467	5594			
				5596	5596	5609	5717	5721	5776	5782	5784	5787	5789	5792	5900			
				5908	5926	6041	6415	6490	7901	7901	7901*	7906*	7916*	7924*	7932			
				7935*	7939	1306	2082											
I@PRSW	001	0087	3867	7716	7721	8385	8394											
I@PRTE	001	0000	3942															
I@RSE1	001	0007	3895	4134	4142	4173	4176*	4178	4185*	4187	4488*	4489	4489*	4502*	4663			
				4665*	4666*	4848	4848*	4936*	4982	5130	5130*	5173</						

CROSS REFERENCE																
SYMBOL	LEN	VALUE	DEFN	REFERENCES	VER 15, MOD 00 31/05/21 PAGE 329											
I@UMRL	001	000F	3840	3841												
I@UMRS	001	0007	3817	3818	3864											
I@XBRC	001	0003	3918													
I@XCNT	001	0001	3916	6899												
I@XCOD	001	0001	3917	6769	6937	6959										
I@XLNO	001	0002	3914													
I@XOPC	001	0000	3913	3914	3915	3916	3917	3918	6762	1705						
I@XVAD	001	0002	3915	6836	7241	7312*	7313*	7314*	1724	1745	1755					
I@1SE1	001	0000	3894	3895	3897	4132	4485	4497*	4498	4498*	4499	4499*	4510*	4641	4654*	
				4667*	4803	4847*	4937*	4938*	4984	4986*	5111	5131*	5157	5188	5190*	
				5240*	5361	5365	5373	5384*	5390	5390*	5391*	5392	5392*	5393	5393*	
				5394	5394*	5395	5395*	5396	5396*	5397*	5424*	5594	5597*	5609	5621	
				5633	6049	6051*	6062	6190	6194	6205	6341	6343	6344*	6349	6375	
I@1SE2	001	0008	3897	3898	3900	5116	5121	5128	5142	5148	5150	5166	5386*	5689*		
I@1SE3	001	0010	3900	3901	4171*	4172	4172*	4175*	4180*	4187*	4193*	4195	4197			
IBR810	003	1ACF	7008													
IDFADF	001	1A95	6916	7041												
IDFBAT	001	1AE0	7037	6763												
IDFCLS	001	1AD2	7016	7044												
IDFGET	001	1A40	6832	7038												
IDFILE	001	1A00	6757	7038	7039	7040	7041	7042	7043	7044	7045	7046				
IDFINI	001	1A87	6894	7040												
IDFPRS	001	1A9E	6933	7045												
IDFPRU	001	1AAD	6955	7046												
IDFPUT	001	1A75	6870	7039												
IDFRSR	001	1AC0	6983	7042												
IDFRST	001	1AC9	6999	7043												
IDFSMK	001	000C	6975	6959												
IDF010	004	1A04	6762													
IDF020	004	1A0B	6764	6762*												
IDF030	004	1A0F	6768													
IDF040	003	1A18	6773	6764*												
IDF050	003	1A1B	6799	6876	6938	6960										
IDF055	004	1A1E	6803													
IDF060	006	1A25	6808													
IDF065	004	1A2F	6811	6812	6814	6875*	6877*									
IDF070	006	1A36	6820	6804												
IDF075	004	1A3C	6824	6799*	6816											
IDF100	004	1A40	6836													
IDF110	002	1A49	6839	6836*												
IDF120	006	1A4A	6843													
IDF130	003	1A59	6849													
IDF140	004	1A66	6856	6851												
IDF150	004	1A6E	6861													
IDF200	003	1A75	6875													
IDF220	003	1A84	6886													
IDF300	005	1A87	6899													
IDF310	004	1A8C	6903													
IDF320	003	1A92	6908													
IDF420	004	1A95	6920													
IDF430	003	1A9B	6925													
IDF500	003	1A9E	6937													
IDF510	004	1AA4	6942													
IDF520	003	1AAA	6947													

## CROSS REFERENCE

VER 15, MOD 00 31/05/21 PAGE 330

SYMBOL	LEN	VALUE	DEFN	REFERENCES
IDF620	004	1AB9	6969	6886 6908 6925 6947
IDF700	006	1AC0	6987	
IDF710	003	1AC6	6991	
IDF800	004	1AC9	7003	
IDF900	004	1AD2	7020	
IDF910	004	1AD8	7025	6991 7008
IDF990	004	1ADC	7029	6862 6970
IDIBM2	002	1BA3	7329	7249
IDIFNC	001	1B00	7239	
IDIFTE	002	1BA8	7335	7262
IDIFVA	001	0001	7347	7250 7252 7273*
IDILBI	001	1BA4	7331	7298 7310
IDILFI	001	1BA5	7332	7314
IDILPV	001	1BA6	7333	7319
IDIVAD	002	1BAA	7341	7299* 7319* 7320
IDI010	006	1B09	7247	
IDI020	003	1B13	7249	7253
IDI030	004	1B24	7257	
IDI040	005	1B2C	7262	7251
IDI050	004	1B34	7267	
IDI060	004	1B3C	7272	7263
IDI070	004	1B45	7280	
IDI080	003	1B51	7286	
IDI090	004	1B57	7291	
IDI100	004	1B5F	7298	7287
IDI110	005	1B7E	7310	
IDI130	004	1B95	7319	
IDP210	004	1A7E	6881	
LPBUFR	001	4F00	3708	3694* 3707 3715
LPRCMD	005	4DE9	3555	3625
MIN	001	2B00	9972	
RETURN	001	4DB3	3525	9367 9532
SFACTR	001	1CF6	7458	7366* 7372* 7378 7382 7389
SFADFR	001	1C00	7354	7355
SFAD2D	001	1CF4	7456	7406* 7407 7426
SFAVD1	002	1CEE	7451	7359 7418
SFAVD2	002	1CF0	7452	7360
SFAWK1	002	1CF8	7459	7359* 7396 7399* 7442
SFA0B0	001	00B0	7450	7407 7444
SFA001	001	1CF1	7453	7368 7372 7373
SFA007	001	1CF2	7454	7390
SFA008	001	1CF3	7455	7392
SFA010	004	1C21	7369	7374
SFA020	003	1C37	7378	7371 7445*
SFA030	003	1C44	7382	7379
SFA032	001	1CF5	7457	7406
SFA040	005	1C51	7389	7383
SFA050	005	1C65	7396	7420
SFA060	003	1C76	7400	7410
SFA065	004	1C7C	7402	7413
SFA070	005	1C83	7404	7391* 7401
SFA075	003	1C8F	7407	7417* 7444*
SFA080	004	1C9B	7411	7408
SFA090	003	1CA5	7417	7412
SFA100	003	1CB4	7424	7405
SFA110	004	1CCC	7434	7424*

## CROSS REFERENCE

VER 15, MOD 00 31/05/21 PAGE 331

SYMBOL	LEN	VALUE	DEFN	REFERENCES
SFA115	005	1CDA	7442	7403 7429
SFA120	003	1CE3	7444	7381 7385
SFGBLK	003	003D	8279	8215 8222
SFGBS1	001	2100	8134	8135 8245 8250
SFGBS2	001	2200	8286	8287
SFGBS3	001	2300	8434	8435 8586 8598
SFGBVA	002	214B	8165	8163* 8166 8252
SFGCBA	002	21FC	8274	8168* 8238
SFGCBP	001	00FF	8277	8171
SFGCBV	002	2368	8508	8505* 8506*
SFGCNL	002	22E8	8422	8309* 8314* 8329* 8331 8332 8357* 8423
SFGDEH	001	0006	8609	8519
SFGDLS	001	22E3	8415	8314
SFGDRL	001	00E9	8598	8468
SFGDWL	001	00E3	8586	8460
SFGD2P	004	2276	8354	8316*
SFGELS	001	0004	8419	8391
SFGETR	001	2100	8136	
SFGHDL	001	0007	8608	8519 8609 8612
SFGICR	003	0040	8278	8181 8233
SFGLEH	001	23F4	8611	8521 8528* 8533 8536 8541* 8542
SFGMFA	006	2272	8348	8342*
SFGMLQ	002	22EC	8426	8336* 8337* 8338 8340 8344 8427
SFGMS1	001	00FF	8418	8336
SFGMTA	006	2270	8347	8326* 8344* 8374*
SFGNFM	001	00FF	8276	8159 8162
SFGONE	001	22E4	8416	8374
SFGPAF	001	23F1	8603	8514
SFGPCL	002	22EA	8425	8331* 8334* 8337 8355 8356 8357
SFGPLR	001	23E9	8591	8598
SFGPLW	001	23E3	8579	8586
SFGPSL	001	23F3	8605	8540 8541
SFGRPL	004	2334	8477	8467* 8468*
SFGRST	003	003A	8280	8216
SFGSA0	001	0F00	8607	
SFGSBR	004	233A	8483	8470*
SFGSB2	007	23FA	8613	8550* 8555 8561* 8562
SFGSDF	002	22E6	8421	8315* 8332 8334 8369*
SFGSHD	007	23FA	8612	8519* 8613
SFGSSL	001	23F2	8604	8527 8528
SFGSSZ	002	23F0	8602	8497
SFGSXR	004	233E	8485	8471* 8504* 8513 8520
SFGVCB	002	2234	8320	8317* 8318*
SFGVD2	002	21FA	8272	8137 8266
SFGVNB	002	229D	8373	8370* 8371* 8408
SFGWPL	004	231E	8466	8459* 8460*
SFGXRD	001	00FE	8614	8562*
SFGZRO	002	22E2	8414	8299
SFG120	004	2126	8148	8145
SFG150	003	212D	8150	8147
SFG200	003	2130	8154	8143
SFG205	004	2142	8163	8160
SFG210	003	215A	8172	8161* 8173*
SFG215	004	2160	8177	8240
SFG220	003	216F	8186	8229
SFG225	003	2172	8187	8181* 8215 8216* 8222* 8233*

## CROSS REFERENCE

VER 15, MOD 00 31/05/21 PAGE 332

SYMBOL	LEN	VALUE	DEFN	REFERENCES
SFG227	003	2175	8189	8278 8279 8280
SFG230	003	218A	8202	8208
SFG235	003	219F	8215	8200
SFG240	004	21A8	8219	8224*
SFG245	003	21AF	8222	8280
SFG250	003	21B2	8224	8279
SFG255	003	21B5	8228	8187 8211 8217 8220 8234 8278
SFG260	003	21BB	8233	8190
SFG265	003	21C1	8238	8193
SFG270	003	21C4	8240	8182* 8195*
SFG280	004	21CD	8249	8172
SFG282	005	21D3	8252	8257
SFG285	004	21DF	8256	8205
SFG290	004	21E6	8261	8155
SFG295	005	21EC	8266	8149 8254
SFG450	003	220D	8309	8300
SFG470	004	2220	8315	8312
SFG500	004	2249	8331	8328 8375
SFG520	003	2258	8336	8333
SFG550	006	226D	8346	8338* 8347 8348
SFG555	004	2273	8353	8354
SFG570	004	22A5	8380	8358
SFG575	003	22AF	8384	8310* 8313*
SFG585	003	22C3	8394	8384
SFG690	004	22D0	8404	8322
SFG695	005	22D4	8408	8382 8385 8392 8394 8400
SFG750	003	2300	8449	8537
SFG760	004	230C	8453	8452 8515
SFG780	003	2313	8459	
SFG785	004	2319	8462	8466
SFG790	004	232F	8473	8477
SFG795	004	2337	8482	8483
SFG800	004	233B	8484	8485
SFG810	003	2345	8494	8454
SFG825	003	2355	8500	8495
SFG830	003	2358	8504	8450
SFG840	004	2379	8519	8511
SFG850	003	2386	8526	8535* 8539*
SFG860	003	2394	8533	8522
SFG870	004	239D	8536	8526
SFG880	003	23A4	8539	8534
SFG890	004	23AF	8542	8529
SFG900	004	23B3	8546	8498
SFG920	003	23B7	8550	8245
SFG930	003	23C6	8560	8250 8567* 8570*
SFG935	004	23D0	8564	8560
SFG940	003	23DC	8570	8556
SFG945	004	23DF	8572	8568
SFPBFR	006	1EC8	7752	7746* 7750*
SFPBS1	001	1D00	7563	7564
SFPBS2	001	1E00	7680	7681
SFPBS3	001	1F00	7791	7792 7860 7870
SFPBS4	001	2000	7880	7881
SFPCBP	002	2094	7961	7887* 7983
SFPCBV	002	2095	7960	7961 7962
SFPCFL	005	20DE	8003	7894* 7939* 7944 7999 8000 8010

## CROSS REFERENCE

SYMBOL LEN VALUE DEFN REFERENCES VER 15, MOD 00 31/05/21 PAGE 333

SFPCNL	001	1EF2	7772	7713*	7720*	7727*	7734	7745	7756*	7773	7810
SFPCPT	002	20C0	7986	7943	7964	7979*	7988	7995*	8009		
SFPCPW	002	20FD	8027	7943*	7944*	7948					
SFPCRT	002	1DF0	7676	7591							
SFPCXI	004	1DE0	7659	7625*							
SFPC01	002	1EFC	7782	7740	7758						
SFPDAC	002	20FD	8025	7920*	7924	7926	7926*	8026			
SFPDCA	005	20DF	8002	8000*							
SFPDEV	002	1DEB	7670	7584*	7587*	7607	7672				
SFPDIC	002	1DEB	7672	7639*	7641*						
SFPDLS	001	0004	7787	7724							
SFPDP1	001	1F7E	7853	7860							
SFPDP2	001	1F84	7863	7870							
SFPD1D	001	007E	7860	7814							
SFPD2D	001	0084	7870	7821							
SFPENC	001	0005	8018	7939							
SFPEXI	004	20FA	8022	7906							
SFPEZR	001	20FB	8023	7907							
SFPLEX	001	0004	8016	7906	8018	8022					
SFPLXM	001	0002	8017	7920	7924	7924	7926	7926			
SFPMPT	002	1DEE	7675	7587							
SFPMS1	001	00FF	7788	7747							
SFPMVL	006	1EC6	7753	7747*	7748*	7749	7750				
SFPNGE	002	20FD	8026	7913*	7914*	7915	8027				
SFPONE	001	1DEC	7674	7641							
SFPPRT	002	1EF8	7778	7733*	7745*	7748	7755	7756	7779		
SFPRT2	002	1F8B	7874	7793							
SFPSAO	001	0F00	7877								
SFPSCA	002	1EFA	7781	7767	7847*						
SFPSIO	002	1EF6	7786	7698*	7711						
SFPSTC	003	208E	8004	7893*	7999*						
SFPSTK	006	1ECA	7754	7728*	7749*	7758*	7811				
SFPUTR	001	1D00	7566								
SFPVCA	002	20C0	7985	7886*	7983*	7986					
SFPVD2	002	1DF2	7677	7567	7611						
SFPWK2	002	1EF6	7777	7736*	7738*	7739*	7740*	7786			
SFPWRK	001	1EF4	7776	7696*	7697*	7731	7733	7734*	7737*	7738	
SFPXR1	004	1E7C	7730	7699*	7704						
SFPX01	001	20F5	8020	7995	8011						
SFPZD1	001	20F6	8021	7920							
SFP050	004	1D26	7578	7575							
SFP075	003	1D2D	7580	7577							
SFP100	005	1D30	7584	7573							
SFP120	003	1D43	7589	7586							
SFP130	003	1D54	7596	7588							
SFP133	004	1D61	7600	7597							
SFP135	004	1D6F	7604	7602							
SFP140	004	1D73	7605	7599							
SFP150	005	1D7E	7611	7579	7621	7666					
SFP175	003	1D8B	7617	7590							
SFP200	003	1D9A	7625	7618							
SFP220	004	1DA3	7632								
SFP230	003	1DB3	7640	7644	7648						
SFP250	003	1DD0	7649	7642							
SFP320	004	1DDD	7658	7627	7634	7659					
SFP350	005	1E0D	7693	7689							



## CROSS REFERENCE

VER 15, MOD 00 31/05/21 PAGE 334

SYMBOL	LEN	VALUE	DEFN	REFERENCES
SFP370	003	1E1B	7696	7691
SFP385	004	1E3F	7708	7701
SFP400	003	1E5F	7720	7712
SFP410	003	1E65	7722	
SFP430	003	1E73	7727	7710
SFP450	003	1E76	7728	7716 7719 7721 7725
SFP460	004	1E79	7729	7730
SFP480	004	1EA5	7742	
SFP490	004	1EA9	7743	7706 7757
SFP5	001	0005	7876	7813* 7814* 7820* 7821*
SFP500	004	1EAD	7745	7732 7735 7760
SFP510	005	1EB1	7746	7741
SFP550	006	1EC5	7751	7752 7753 7754
SFP560	004	1ED6	7758	
SFP580	003	1EE0	7764	7690 7705 7759
SFP590	004	1EEE	7768	7764*
SFP610	005	1F1E	7808	7797 7799
SFP625	006	1F28	7810	7808*
SFP630	006	1F2E	7811	7809*
SFP635	004	1F3E	7816	7813* 7814*
SFP640	004	1F50	7825	7820* 7821*
SFP650	004	1F58	7833	7822*
SFP655	004	1F5C	7834	7823*
SFP675	005	1F66	7840	7804
SFP680	006	1F74	7847	7795*
SFP720	004	204A	7920	7909
SFP725	003	2051	7922	7921* 7925 7925* 7927
SFP730	004	205B	7925	7923
SFP750	003	2066	7931	7908
SFP760	003	2075	7939	7933
SFP785	004	2078	7943	7896
SFP790	003	208A	7948	7946 7947* 7949 7951 8004
SFP800	004	2090	7959	7890
SFP830	004	20AA	7976	7885* 8008
SFP850	004	20B7	7983	7952
SFP865	004	20D5	7999	7989
SFP875	005	20DD	8001	8002 8003
SFP950	004	20F1	8012	7978
SFRBS1	001	2400	8621	8620 8727
SFRCAL	001	2400	8625	
SFRCLS	001	240A	8632	
SFRIXR	004	2484	8698	8643*
SFRLPR	003	24B7	8731	8742
SFRNOE	001	24AB	8717	8694* 8718
SFRONE	001	24AA	8715	8694
SFRSET	001	240D	8637	
SFRVD2	002	2412	8642	8696 8706
SFRX10	001	24AC	8720	8699
SFR100	004	240D	8641	
SFR110	003	2416	8647	8627* 8708*
SFR115	003	241C	8650	8647 8700
SFR130	003	241F	8654	
SFR135	004	2448	8667	8661*
SFR140	003	244F	8672	8633* 8657 8704*
SFR200	004	2452	8677	8686
SFR300	003	2461	8684	8672



## CROSS REFERENCE

VER 15, MOD 00 31/05/21 PAGE 335

SYMBOL	LEN	VALUE	DEFN	REFERENCES
SFR900	003	2472	8693	8626* 8655 8659 8680 8709*
SFR950	004	2481	8697	8698
SFR995	003	248C	8704	8693 8695
SFR996	004	24C3	8735	8731*
SFR997	004	24CA	8737	8730
SFR998	006	24D1	8739	8736
SFR999	004	24D7	8740	8738
SF1000	001	24E5	8745	8732
V\$APWR	001	0800	2366	2511
V\$BFR1	001	5400	2429	2619 0064
V\$BFR2	001	5500	2430	2620
V\$CBNZ	001	0CB2	2438	2518 4501 4512
V\$CCON	001	5120	2445	2616
V\$CDCV	001	3100	2442	2571 8565
V\$CDSY	001	2E00	2441	2568 8553
V\$CFPZ	001	0C70	2436	2517 4814 5639 6370
V\$CNXZ	001	0470	2439	2506 4161
V\$CSSR	001	5100	2444	2615
V\$CZFP	001	04AD	2437	2507 4206 5664 6394
V\$DTLN	001	4600	2451	2603
V\$DTVR	001	4700	2452	2604
V\$FABS	001	1761	2337	2535
V\$FACS	001	1400	2353	2527
V\$FASN	001	1413	2352	2528
V\$FATN	001	1100	2351	2524 6382
V\$FCOS	001	0A00	2348	2513 6211 6224
V\$FCOT	001	0D00	2346	2519
V\$FCSC	001	1725	2350	2534
V\$FDEG	001	17DA	2357	2539
V\$FDET	001	4540	2360	2602
V\$FEXP	001	0500	2344	2508 4797 5231
V\$FHCS	001	1500	2356	2529
V\$FHSN	001	1557	2355	2530
V\$FHTN	001	1593	2354	2531
V\$FINT	001	176C	2338	2536
V\$FLGT	001	0200	2342	2501 4153 4877
V\$FLOG	001	0219	2341	2503 5214
V\$FLTW	001	020B	2343	2502
V\$FRAD	001	17CB	2358	2538
V\$FRND	001	1800	2359	2540
V\$FSEC	001	1700	2349	2533
V\$FSGN	001	17A7	2339	2537
V\$FSIN	001	0A1A	2347	2514 6215 6220
V\$FSQR	001	0900	2340	2512
V\$FTAN	001	0D28	2345	2520
V\$IFCI	001	1B00	2329	2544
V\$IFIO	001	1A00	2331	2543
V\$ISDN	001	1900	2330	2541
V\$KBTL	001	1EAC	2473	
V\$KBTS	001	0DAC	2472	
V\$LPRB	001	4F00	2427	2613 9536 3543
V\$LPRT	001	4D00	2425	2611 9322 9512 9530 3539 3748
V\$LPR2	001	4E00	2426	2612 3552 3556
V\$MADD	001	4007	2374	2591
V\$MASN	001	43A0	2372	2598
V\$MCON	001	4324	2379	2596

## CROSS REFERENCE

VER 15, MOD 00 31/05/21 PAGE 336

SYMBOL	LEN	VALUE	DEFN	REFERENCES
V\$MIDN	001	4300	2380	2595
V\$MINV	001	4500	2384	2601
V\$MMPY	001	4100	2376	2592
V\$MSMY	001	4264	2377	2594
V\$MSUB	001	4000	2375	2590
V\$MTRN	001	4400	2383	2600
V\$MZER	001	432B	2381	2597
V\$PCH1	001	5200	2465	2617
V\$PCH2	001	5300	2466	2618 9443 2701 3560 3561
V\$SCDI	001	2A00	2422	2562 8178
V\$SCDO	001	2A96	2423	2563 7972
V\$SFA2	001	5000	2407	2614 7441
V\$SFD1	001	0000	2417	2499 7451
V\$SFD2	001	0100	2418	2500 7452 7677 8272 8642
V\$SKEY	001	2500	2421	2557 8918 8940 9074 9075 9998
V\$SPRT	001	2800	2420	2560 8734 9172 9411 0259 2662 3775
V\$VMPL	001	4C06	2459	2610
V\$VMPS	001	4C00	2458	2609
V\$XKAF	001	1C00	2406	2545 6921
V\$XKCA	001	2400	2410	2553
V\$XKCL	001	240A	2409	2554 7021
V\$XKIN	001	2B00	2405	2564 6904
V\$XKLP	001	24AD	2411	
V\$XKRS	001	240D	2408	2555 7004
V\$XMGT	001	3E06	2399	2585
V\$XMIN	001	3D00	2398	2583
V\$XMPL	001	3F06	2402	2588
V\$XMPS	001	3F00	2401	2587
V\$XMPT	001	3E0C	2400	2586
V\$XMPU	001	3F13	2403	2589
V\$XMRD	001	3E00	2397	2584
V\$XSGT	001	2100	2392	2550 8245 8250 8262 8305 8367
V\$XSIN	001	2B6E	2391	2565
V\$XSPR	001	3400	2394	2574 6943 7606
V\$XSPT	001	1D00	2393	2546 6882 7620 7662 7766 7874 8666
V\$XSPU	001	3800	2395	2578 6965
V\$XSRD	001	3300	2390	2573
V\$00E1	001	0000	2499	
V\$01E1	001	0100	2500	
V\$02E1	001	0200	2501	
V\$02E2	001	020B	2502	
V\$02F3	001	0219	2503	
V\$03CC	001	0300	2504	
V\$04CC	001	0400	2505	
V\$04E1	001	0470	2506	
V\$04E2	001	04AD	2507	
V\$05E1	001	0500	2508	
V\$06CC	001	0600	2509	
V\$07CC	001	0700	2510	
V\$08E1	001	0800	2511	
V\$09E1	001	0900	2512	
V\$10E1	001	0A00	2513	
V\$10E2	001	0A1A	2514	
V\$11CC	001	0B00	2515	
V\$12CC	001	0C00	2516	
V\$12E1	001	0C70	2517	

## CROSS REFERENCE

SYMBOL   LEN   VALUE   DEFN   REFERENCES   VER 15, MOD 00   31/05/21   PAGE 337

V\$12E2	001	0CB2	2518	
V\$13E1	001	0D00	2519	
V\$13E2	001	0D28	2520	
V\$14CC	001	0E00	2521	
V\$15CC	001	0F00	2522	
V\$16CC	001	1000	2523	
V\$17E1	001	1100	2524	
V\$18CC	001	1200	2525	
V\$19CC	001	1300	2526	
V\$20E1	001	1400	2527	
V\$20E2	001	1413	2528	
V\$21E1	001	1500	2529	
V\$21E2	001	1557	2530	
V\$21E3	001	1593	2531	
V\$22CC	001	1600	2532	
V\$23E1	001	1700	2533	
V\$23E2	001	1725	2534	
V\$23E3	001	1761	2535	
V\$23E4	001	176C	2536	
V\$23E5	001	17A7	2537	
V\$23E6	001	17CB	2538	
V\$23E7	001	17DA	2539	
V\$24E1	001	1800	2540	
V\$25E1	001	1900	2541	
V\$26E1	001	1A00	2543	
V\$27E1	001	1B00	2544	
V\$28E1	001	1C00	2545	
V\$29E1	001	1D00	2546	
V\$30CC	001	1E00	2547	
V\$31CC	001	1F00	2548	
V\$32CC	001	2000	2549	
V\$33E1	001	2100	2550	
V\$34CC	001	2200	2551	
V\$35CC	001	2300	2552	
V\$36CC	001	2400	2556	
V\$36E1	001	2400	2553	
V\$36E2	001	240A	2554	
V\$36E3	001	240D	2555	
V\$37E1	001	2500	2557	
V\$38CC	001	2600	2558	
V\$39CC	001	2700	2559	
V\$40E1	001	2800	2560	
V\$41CC	001	2900	2561	
V\$42E1	001	2A00	2562	
V\$42E2	001	2A96	2563	
V\$43E1	001	2B00	2564	
V\$43E2	001	2B6E	2565	
V\$44CC	001	2C00	2566	
V\$45CC	001	2D00	2567	
V\$46E1	001	2E00	2568	
V\$47CC	001	2F00	2569	
V\$48CC	001	3000	2570	
V\$49E1	001	3100	2571	
V\$50CC	001	3200	2572	
V\$51E1	001	3300	2573	
V\$52E1	001	3400	2574	

## CROSS REFERENCE

SYMBOL	LEN	VALUE	DEFN	REFERENCES	VER 15, MOD 00	31/05/21	PAGE 338
V\$53CC	001	3500	2575				
V\$54CC	001	3600	2576				
V\$55CC	001	3700	2577				
V\$56E1	001	3800	2578				
V\$57CC	001	3900	2579				
V\$58CC	001	3A00	2580				
V\$59CC	001	3B00	2581				
V\$60CC	001	3C00	2582				
V\$61E1	001	3D00	2583				
V\$62E1	001	3E00	2584				
V\$62E2	001	3E06	2585				
V\$62E3	001	3E0C	2586				
V\$63E1	001	3F00	2587				
V\$63E2	001	3F06	2588				
V\$63E3	001	3F13	2589				
V\$64E1	001	4000	2590				
V\$64E2	001	4007	2591				
V\$65E1	001	4100	2592				
V\$66CC	001	4200	2593				
V\$66E1	001	4264	2594				
V\$67E1	001	4300	2595				
V\$67E2	001	4324	2596				
V\$67E3	001	432B	2597				
V\$67E4	001	43A0	2598				
V\$68E1	001	4400	2600				
V\$69E1	001	4500	2601				
V\$69E2	001	4540	2602				
V\$70E1	001	4600	2603				
V\$71E1	001	4700	2604				
V\$72CC	001	4800	2605				
V\$73CC	001	4900	2606				
V\$74CC	001	4A00	2607				
V\$75CC	001	4B00	2608				
V\$76E1	001	4C00	2609				
V\$76E2	001	4C06	2610				
V\$77CC	001	4D00	2611				
V\$78CC	001	4E00	2612				
V\$79CC	001	4F00	2613				
V\$80E1	001	5000	2614				
V\$81E2	001	5100	2615				
V\$81E3	001	5120	2616				
V\$82E1	001	5200	2617				
V\$83E2	001	5300	2618				
V\$84E1	001	5400	2619				
V\$85E2	001	5500	2620				
V@CDPT	001	0007	2631				
V@CHGH	001	0008	2736				
V@CMIC	001	0002	2632				
V@CMNI	001	00FF	2629				
V@CMUL	001	0007	2737				
V@CNIX	001	0080	2630				
V@COEX	001	001E	2627				
V@CPLS	001	00F0	2634				
V@CPRC	001	000A	2636				
V@CSQR	001	0003	2734				
V@CSTR	001	0002	2735				

## CROSS REFERENCE

VER 15, MOD 00 31/05/21 PAGE 339

SYMBOL	LEN	VALUE	DEFN	REFERENCES
V@CTTA	001	0027	2637	
V@DCAD	001	0002	2657	2658
V@DEXP	001	0000	2662	
V@DMAN	001	000D	2664	2665
V@DMN1	001	0001	2663	
V@DPDF	001	0002	2652	
V@DSAD	001	0001	2653	
V@DSGN	001	000D	2665	
V@DVAD	001	0004	2658	
V@EART	001	0001	2635	
V@ECRT	001	0038	2708	
V@EFUL	001	00F8	2707	
V@EINV	001	00FB	2703	
V@EIPR	001	00F5	2704	
V@ENSV	001	00F7	2705	
V@ENUL	001	0000	2702	
V@ERPC	001	0020	2633	
V@ESAV	001	00F6	2706	
V@FEHN	001	0002	2732	
V@FEPL	001	0091	2728	
V@FERS	001	0003	2731	
V@FPGS	001	0081	2727	
V@FRET	001	0015	2730	
V@FSPC	001	0040	2729	
V@FTAB	001	0000	2733	
V@KADD	001	004E	2718	
V@KCLE	001	006E	2715	
V@KDIV	001	0061	2721	
V@KEMN	001	006C	2713	
V@KEPL	001	006B	2712	
V@KMUL	001	005C	2720	
V@KPER	001	004B	2723	
V@KPST	001	007B	2717	
V@KPWR	001	005A	2722	
V@KSQR	001	006F	2714	
V@KSTO	001	006D	2716	
V@KSUB	001	0060	2719	
V@LAIP	001	0003	2683	2684
V@LDEX	001	0002	2686	
V@LETE	001	0003	2690	
V@LEXP	001	0001	2680	2682
V@LFKO	001	0006	2685	
V@LINI	001	0200	2689	
V@LLKS	001	0010	2682	
V@LMAN	001	000F	2681	2682
V@LNOP	001	0015	2687	
V@LTBE	001	0007	2684	
V@LVPG	001	0100	2688	2689
V@MCHS	001	00C0	2669	
V@MCRD	001	0010	2645	
V@MDEF	001	0008	2646	
V@MEXC	001	0080	2643	
V@MEXT	001	0004	2672	
V@MICC	001	0010	2628	
V@MIPC	001	0080	2670	
V@MIPL	001	0020	2676	

VER 15, MOD 00 31/05/21 PAGE 340

V@MLST	001	0040	2644		
V@MPND	001	0000	2675		
V@MPOF	001	0080	2673		
V@MPRC	001	0020	2642		
V@MSFU	001	0002	2647		
V@MSTN	001	0004	2641		
V@OALL	001	00F4	2698		
V@ONUL	001	00F0	2694	2695	
V@OPM1	001	00F2	2696	2697	
V@ORTN	001	00F1	2695	2696	
V@OSTK	001	00F3	2697	2698	
V@PEOF	001	0002	2671		
V@PSQ2	001	0014	2674		
VLPRT2	001	4E00	3586	3551	3552
VLPRT3	001	5300	3729		
VLPRT4	001	5359	3763		
VLPRT5	001	5391	3783		
VLPRT6	001	53B6	3797		

```
OL105 I THE CODE LENGTH OF #FMSTD IS 21453 DECIMAL.
OL103 I TOTAL NUMBER OF LIBRARY SECTORS REQUIRED IS 58
      NAME-#FMSTD,PACK-R1R1R1,UNIT-R1,RETAIN-P,LIBRARY-R,CATEGORY-000
```

START ADDRESS	CATEGORY	NAME AND ENTRY	CODE LENGTH HEXADECIMAL	DECIMAL
---------------	----------	----------------	----------------------------	---------

0200	0	#FMSTD	53CD	21453
------	---	--------	------	-------

OL100	I	THE TOTAL CORE USED BY #FMSTD IS 21453 DECIMAL.		
OL101	I	THE START CONTROL ADDRESS OF THIS MODULE IS 0200.		
OL104	I	TOTAL NUMBER OF LIBRARY SECTORS REQUIRED IS    84		
		NAME-#FMSTD,PACK-R1R1R1,UNIT-R1,RETAIN-P,LIBRARY-O		