

```
// LOG PRINTER,NOEJECT
// DATE 090510
// NOHALT
*
*** MOVE '$@SPV1' SOURCE DECK TO R1.
*
// LOAD $MAINT,F1
// RUN
// COPY FROM-READER,TO-R1,LIBRARY-S,RETAIN-R,NAME-$@SPV1
// END
*
*** RUN MACRO PROCESSOR.
*
// LOAD $MPXDV,F1
// COMPILE SOURCE-$@SPV1,UNIT-R1
// FILE NAME-$SOURCE,PACK-F2F2F2,UNIT-F2,RETAIN-T,TRACKS-25
// RUN
*
*** ASSEMBLE THE '$@SPV1' SOURCE.
*
// LOAD $ASSEM,F2
// SWITCH 1000000
// COMPILE OBJECT-R1,SOURCE-DUMMY,UNIT-R1
// FILE NAME-$SOURCE,PACK-F2F2F2,UNIT-F2,RETAIN-S,TRACKS-25
// FILE NAME-$WORK,PACK-F2F2F2,UNIT-F2,RETAIN-S,TRACKS-25
// FILE NAME-$WORK1,PACK-F2F2F2,UNIT-F2,RETAIN-S,TRACKS-25
// RUN
```

OPTIONS NODECK,LIST,NOXREF,REL,OBJ(P)

THE LIST OF OPTIONS USED DURING THIS ASSEMBLY IS-- NODECK,LIST,NOXREF,REL,OBJ

EXTERNAL SYMBOL LIST

VER 15, MOD 00 09/05/10 PAGE 1

SYMBOL	TYPE
\$@SPV1	MODULE
SKDIOS	ENTRY
JUHALT	ENTRY
NENTRY	ENTRY
NXACTV	ENTRY
NLOADR	ENTRY
DIODSP	EXTRN
DIODWT	EXTRN
DIODQ1	EXTRN
\$@COIH	EXTRN
DSKNOP	EXTRN
DIODQ2	EXTRN
DIODQ3	EXTRN
\$#SUPC	EXTRN
SUPEND	EXTRN

\$@SPV1 - SUPERVISOR FOR A DEDICATED SYSTEM

```

ERR LOC  OBJECT CODE      ADDR STMT SOURCE STATEMENT                VER 15, MOD 00  09/05/10  PAGE  2
0000          2  @$SPV1 START 0000                START ADDRESS
          3  *          LEVEL 14
          4  *****
          5  *
          6  *TITLE:  'LOWCR'  THIS IS THE FIRST 256 BYTES OF THE NUCLEUS
          7  *
          8  *STATUS: CHANGE LEVEL 14.          DATED 11/03/76
          9  *
         10  *FUNCTION/OPERATION: THE FIRST 256 BYTES CONTAIN SYSIN HISTORY, SOME
         11  *          COMMON DC'S, THE SYSTEM READ ROUTINE, SYSTEM COMMUNICATION
         12  *          REGION AND THE ENTRY POINTS FOR SYSTEM DUMP, GENERAL ENTRY,
         13  *          IOS AND IOS WAIT, THE DUMP PROGRAM WRITES OUT THE XNST AREA
         14  *          AND READS IN THE BOOT STRAP FOR THE MAIN DUMP PROGRAMS.
         15  *
         16  *ENTRY POINTS:  NCENTR  FOR GENERAL ENTRY
         17  *          NECIOS  FOR IOS READ/WRITE
         18  *          NECIOW  FOR IOS WAIT
         19  *          CDUMPD  FOR CORE DUMP
         20  *          NREAD  XR1 POINTS TO IOB
         21  *          HLT1   LSR DISPLAY ROUTINE
         22  *
         23  *INPUT-  NCENTER- XR2 POINTS TO A PARAMETER TABLE.
         24  *          NREAD-  XR1 POINTS TO THE IOB.
         25  *
         26  *OUTPUT-  FOR THE LSR DISPLAY ROUTINE, THE CONTAINS OF REGISTERS.
         27  *          DUMPPG: THE PROGRAM WRITES OUT THE XNST AREA
         28  *
         29  *EXTERNAL ROUTINES:  NONE
         30  *
         31  *EXITS:   NREAD- RETURNS TO CALLER IF READ SUCCESSFUL.
         32  *
         33  *TABLES/WORK AREAS:  CONTAINS THE SYSTEM COMMUNICATION REGION.
         34  *          SYSIN HISTORY TABLE.
         35  *
         36  *CHARACTER CODE DEPENDENCY: A
         37  *
         38  *NOTES: N/A.
         39  *
         40  *ATTRIBUTES: REUSABLE
         41  *
         42  *CHANGE ACTIVITY - @$SPV1
         43  *          RELEASE 10
         44  *          @01-          - SAVE TRANSIENT AREA ON SYSTEM PACK NOT ON F1.
         45  *          RELEASE 14
         46  *          @02-INCR/SI401 - ALLOW 5 OPTION FOR CEFE SWA DUMP
         47  *          @03-INCR/SI404 - ALLOW 7 OPTION FOR REAL 5448 DUMP
         48  *****

          0001  50          EXTRN DIODSP                DISK IOS READ/WRITE
          0002  51          EXTRN DIODWT                DISK IOS WAIT
          0003  52          EXTRN DIODQ1                ADDRESS OF DISK QUEUE 1
          0004  53          EXTRN @$COIH(3)            CONSOLE INTERRUPT HANDLER
          0005  54          EXTRN DSKNOP(3)            @ OF 5445 TIO OVERLAP NOOP
          0006  55          EXTRN DIODQ2(3)            @ OF 5445 DIS QUEUE
          0007  56          EXTRN DIODQ3(3)            @ OF TAPE QUEUE
          0008  57          EXTRN $#SUPC(3)            @ OF SEC COMM AREA                @03
          0066  58          ENTRY SKDIOS                SCHEDULER BYTES RESOLVES
          0796  59          ENTRY JUHALT                EXTERNS DURING SYS GEN
    
```

\$@SPV1 - SUPERVISOR FOR A DEDICATED SYSTEM

ERR	LOC	OBJECT CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00 09/05/10 PAGE 3
			0009	60		EXTRN SUPEND	ADDRESS- END OF SUPERVISOR
				61	*		
0000	35 10	0013		62	SPBEG L	DMP,IAR	ENTRY FOR DUMP PROGRAM
0004	35 10	0015		63		L ENT,IAR	ENTRY FOR GENERAL ENTRY
0008	35 10	0030		64		L IOS,IAR	FOR IOS
000C	35 10	0034		65		L IOSW,IAR	FOR IOS WAIT
0010	0050		0011	66		DC AL2(SYS)	ADDR OF SYSTEM COMMUNICATION
0012	0100		0013	67	DMP	DC AL2(CDUMPD)	ADDRESS FOR DUMP LINKAGE
0014	04D6		0015	68	ENT	DC AL2(NENTRY)	ADDRESS OF GENERAL ENTRY
0016	0C		0016	69	NCX0C	DC XL1'0C'	CONSTANT OF 0C
0017	0020		0018	70	NCX20	DC XL2'0020'	CONSTANT OF 0020
0019	0E		0019	71	SPVRID	DC XL1'0E'	SUPERVISOR ID - REL 14
001A	0000000000000000		0028	72	NSYSIN	DC XL15'00'	SYSIN HISTORY
0022	0000000000000000			72			
				74	*****		
				75	*	COMMON EQUATES, DC'S AND DS'S	*
				76	*****		
			0020	77	NSRQ	EQU X'20'	SYSTEM PACK USED INDICATOR
			0010	78	NFCHRQ	EQU X'10'	FETCH REQUEST INDICATOR
			001C	79	NRPGRQ	EQU X'1C'	LOAD BY RELATIVE C/S INDICATOR
			0000	80	##	EQU 0	MODIFYABLE CODE INDICATOR
			0000	81	#	EQU 0	MODIFYABLE CODE INDICATOR
			0001	82	XR1	EQU 1	INDEX REGISTER NUMBER 1
			0002	83	XR2	EQU 2	INDEX REGISTER NUMBER 2
			0004	84	PSR	EQU 4	PROGRAM STATUS REGISTER
			0008	85	ARR	EQU 8	ADDRESS RECALL REGISTER
			0010	86	IAR	EQU 16	INSTRUCTION ADDRESS REGISTER
			00FF	87	SAVEDA	EQU X'00FF'	SAVE AREA DISK DATA ADD REG
			00FD	88	SAVEDC	EQU X'00FD'	SAVE AREA DIS CONTROL REG
			037C	89	SAVERG	EQU X'037C'	REGISTER SAVE AREA FOR DUMP
			037F	90	SAVXR1	EQU X'037F'	INDEX REGISTER 1 SAVE AREA
			0399	91	REGEND	EQU X'0399'	END OF REGISTER SAVE AREA
			0385	92	PRTBUF	EQU X'0385'	DUMP PRINT BUFFER & WORK AREA
			0400	93	CHAIN	EQU X'0400'	CHAIN IMAGE AREA ADDRESS
			03FF	94	PBUF	EQU X'03FF'	END OF TRANSIENT AREA
			0391	95	LNO1	EQU X'0391'	REG 2 SAVE AREA
			03A3	96	LNO2	EQU X'03A3'	START OF PRINT BUFFER
			03BF	97	LNO3	EQU X'03BF'	REG 1 SAVE AREA
				99	*****		
				100	*	SUPERVISOR DISK IOS READ, WRITE AND WAIT ROUTINE.	*
				101	*****		
0029	34 08	003E		102	NREAD	ST NREADE+3,ARR	SAVE ARR FOR RETURN BRANCH
002D	C0 87	0001		103		B DIODSP	BRANCH-DISK IOS-READ OR WRITE
			0030	104	IOS	EQU *-1	
0031	C0 87	0002		105		B DIODWT	BRANCH-DISK IOS-WAIT
			0034	106	IOSW	EQU *-1	
0035	7D 40	02		107		CLI 2(,XR1),X'40'	CHECK FOR GOOD COMPLETION
0038	F2 01	04		108		JNE NRDHPL	NO-HALT
003B	C0 87	0000		109	NREADE	B #	RETURN IF GOOD COMPLETION
003F	F0 00	03		110	NRDHPL	HPL X'03',X'00'	'1' HALT-TERMINAL DISK ERROR
0042	0F 03	05BC 05BC		111	HARDAB	SLC NXBUSY(4),NXBUSY	CLEAR BUSY AND ACTIVE BYTES
0048	7A 02	4A		112		SBN NPSCH2(,XR1),X'02'	SET FOR HARD ABORT
004B	C0 87	0004		113		B NCENTR	BRANCH TO GENERAL ENTRY
004F	84		004F	114		DC XL1'84'	RIB FOR END OF JOB TRANSIENT

			116	*****		
			117	*		*
			118	*	SYSTEM COMMUNICATION AREA (SCA).	*
			119	*		*
			120	*****		*****
		0050	121	SYS EQU *		SYSTEM COMMON AREA.
0050	0478	0051	122	NIADOR DC	AL2(N1COMN)	ADDRESS OF PGM. LEVEL 1 COMMON.
0052	0000	0053	123		DC XL2'0000'	NO PROGRAM LEVEL 2 COMMON.
0054	05BB	0055	124		DC AL2(NXACTV)	ADDRESS OF XNST SCHEDULER QUEUE.
0056	00	0056	125	FECHTR DC	XL1'00'	FETCH TRACE /SYSGEN/LIB MAINT.
0057	00	0057	126	PRISZ DC	XL1'00'	PRINTER SIZE
0058	00	0058	127		DC XL1'00'	PAGE SIZE LEFT TRACTOR
0059	00	0059	128		DC XL1'00'	PAGE SIZE RIGHT TRACTOR
005A	F1	005A	129		DC XL1'F1'	SYST LIST IO.
005B	000000	005D	130	SYSLOG DC	XL3'000000'	C/S AND DEV. INFO FOR SYSLOG.
005E	0005	005F	131		DC AL2(DSKNOP)	@ OF SCHEDULER WORK AREA
0060	00	0060	132	N1SYSQ DC	XL1'00'	SYSTEM Q BYTE (IPL'0 PACK).
0061	0000	0062	133		DC XL2'0000'	C/S OF SYSTEM OBJECT LIBRARY.
0063	80000000	0066	134	SKDIOS DC	XL4'80000000'	SCHEDULER SWITCHES.
0067	0003	0068	135	DISKQ DC	AL2(DIODQ1)	5444 DISK QUEUE
0069	00	0069	136		DC XL1'00'	PARTITION VALUE FOR LEVEL TWO.
006A	00	006A	137		DC XL1'00'	MODEL '6' BYTE.
006B	00	006B	138	XTA DC	XL1'00'	X'80' - 5448 SUPPORTED REL 14
		006C	139	IPLPRM EQU *		PART OF TRANSIENT STORAGE.
006C	D65B5BE3D4C9C6	0072	140		DC CL7'O\$STMIF'	\$\$TMIF LOAD PARAMETER LIST.
0073	00	0073	141		DC XL1'00'	AREA USED BY IPL.
0074	0000	0075	142		DC XL2'0000'	LOAD ADDRESS FOR \$\$TMIF.
0076	000000000000000000	007F	143		DC XL10'00'	TRANSIENT STORAGE AREA.
007E	0000		143			
			145	*	OPTIONAL ENTRIES.	
0080	0004	0081	147	IPLCON DC	AL2(\$@COIH)	CONSOLE INTERRUPT HANDLER.
0082	000000	0084	148		DC XL3'000000'	C/S/# OF ROLL-OUT AREA.
0085	00	0085	149		DC XL1'00'	TAPE AND 5445 COMM
0086	0008	0087	150		DC AL2(\$#SUPC)	@ OF ADDITIONAL COMM AREA
0088	0000	0089	151		DC XL2'0000'	HALT/SYSLOG AND SYSIN ARR SAVE
008A	0000	008B	152		DC XL2'0000'	SCHEDULER BYTES.
008C	00	008C	153		DC XL1'00'	TAPE/5445 MVF BYTE.
008D	0006	008E	154		DC AL2(DIODQ2)	5445 DISK QUEUE
008F	0007	0090	155		DC AL2(DIODQ3)	TAPE QUEUE
			157	*****		*****
			158	*	LSR DISPLAY ROUTINE.	*
			159	*	TO USE THIS ROUTINE-	*
			160	*	1. SET IAR TO ADDRESS X'00E8', PRESS START.	*
			161	*	2. PRESS START AGAIN TO GET TO SECOND HPL	*
			162	*	3. AT SECOND HPL, SET AREA SWITCHES FOR OPERATION	*
			163	*	A. FOR A STORE-34XX, WHERE XX=REGISTER NUMBER.	*
			164	*	B. FOR A SENSE-30XX, WHERE XX=REGISTER NUMBER.	*
			165	*	4. PRESS START, TURN ROLLER TO POSITION #3 TO DISPLAY HPL	*
			166	*	5. RECORD CONTENTS OF ROLLER #3, THE QUE CODE DISPLAYED IS	*
			167	*	THE HIGH ORDER BYTE OF THE REGISTER.	*
			168	*	6. PRESS START AGAIN TO GET TO SECOND HPL	*
			169	*	7. RECORD CONTENTS OF ROLLER #3, THE Q CODE DISPLAYED IS	*
			170	*	THE LOW ORDER BYTE OF THE REGISTER.	*
			171	*	8. REPEAT 1-7 UNTIL YOU HAVE FINISHED DISPLAYING DESIRED	*

\$@SPV1 - SUPERVISOR FOR A DEDICATED SYSTEM

			172	*		REGISTERS, THEN IPL OR DUMP STORAGE IF DESIRED.	*	
			173			*****		
00E8			174		ORG	X'00E8'	START AT HEX '00E8'	
00E8	F0	00	00		HLT1	HPL	X'00',X'00'	THE LSR DISPLAY ROUTINE
00EB	F0	50	6F		HLT2	HPL	X'6F',X'50'	SET OP AND Q CODES
00EE	30	00	00F3			SNS	STORE+1,X'00'	SENSE DATA SWITCHES-OP&Q CODE
00F2	34	01	00EA		STORE	ST	HLT1+2,X'01'	STORE OR SENSE REGISTER INFO
00F6	0C	00	00EC	00EA		MVC	HLT2+1(1),HLT1+2	MOVE SECOND HALF OF REG INFO
00FC	C0	87	00E8			B	HLT1	REPEAT ABOVE LOOP

\$@SPV1 - SUPERVISOR FOR A DEDICATED SYSTEM

ERR	LOC	OBJECT CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00 09/05/10 PAGE 6
				182		*****	
				183	*	THE FOLLOWING IS THE DUMP LINKAGE TO WRITE THE	*
				184	*	TRANSIENT AREA OUT ONTO CYL 0 SECTOR B0-BB, THEN LOADS	*
				185	*	THE DUMP PROGRAM INTO CORE AT X'0100' FROM CYL 0,	*
				186	*	SECTORS D4-D8 OF THE SYSTEM PACK.	*
				187		*****	
0100	3C	B0	0149	188	CDUMPD	MVI DPDFCR+2,X'B0'	SET SECTOR TO WRITE TRANS AREA
0104	30	A4	00FF	189		SNS SAVEDA,X'A4'	SAVE DISK DATA ADDRESS REGISTER
0108	30	A6	00FD	190		SNS SAVEDC,X'A6'	SAVE DISK CONTROL REGISTER
010C	31	A4	0146	191		LIO DPDFDR,X'A4'	LOAD DISK DATA ADDRESS REGISTER
0110	31	A6	0147	192		LIO DPDFCR,X'A6'	LOAD DISK CONTROL REGISTER
0114	F3	AB	00	193		SIO X'00',X'AB'	RECAL; SET HEAD TO TRACK 0,F1
0117	3C	00	014A	194		MVI DPDFCR+3,X'00'	SET REGISTER TO 1 SECTOR
011B	F3	AB	00	195		SIO X'00',X'AB'	SET HEAD TO F1,C0,T1
011E	3C	02	014A	196		MVI DPDFCR+3,X'02'	SET REGISTER TO 3 SECTORS
0122	F3	AA	00	197		SIO X'00',X'AA'	WRITE OUT TRANSIENT AREA
0125	F1	A2	00	198		APL X'00',X'A2'	WAIT ON WRITE
0128	3C	D0	0149	199		MVI DPDFCR+2,X'D0'	SET SECTOR TO DO
012C	31	A4	0146	200		LIO DPDFDR,X'A4'	LOAD DISK DATA ADDRESS REGISTER
0130	3C	00	014A	201		MVI DPDFCR+3,X'00'	SET REGISTER TO 1 SECTOR
0134	F3	AB	00	202		SIO X'00',X'AB'	SET HEADS TO SYSTEM PACK @01
0137	3C	02	014A	203		MVI DPDFCR+3,X'02'	SET REGISTER TO 3 SECTORS
013B	F3	A0	00	204		SIO X'00',X'A0'	READ OM DUMP PROGRAM @01
013E	F1	A2	00	205		APL X'00',X'A2'	WAIT ON READ
0141	C0	87	0100	206		B DUMPPG	PASS CONTROL TO DUMP PROGRAM
0145	0100		0146	207	DPDFDR	DC XL2'0100'	ADDRESS OF TRANSIENT AREA
			0147	208	DPDFCR	EQU *	DISK CONTROL FIELD
0147	0000B0F0		014A	209		DC XL4'0000B0F0'	
014B	0147		014C	210	DDFCR@	DC AL2(DPDFCR)	ADDRESS OF DISK CONTROL FIELD
				212		*****	
				213	*	TRANSIENT AREA - DEDICATED SUPERVISOR, 96 COLUMN DUMP	*
				214		*****	
0100				215		ORG X'0100'	STARTS AT HEX 0100 TRAN AREA
0100	30	00	00E9	216	DUMPPG	SNS FESW,X'00'	START OF MAIN DUMP-SENSE SW
0104	34	02	037F	217		ST SAVXR1,XR2	STORE XR1 IN REG SAVE AREA
0108	C2	02	037C	218		LA SAVERG,XR2	POINT XR2 AT REG SAVE AREA-300
010C	B4	01	01	219		ST 1(,XR2),XR1	STORE XR1 IN REG SAVE AREA
010F	B4	04	05	220		ST 5(,XR2),PSR	STORE PSR IN REG SAVE AREA
0112	B0	E4	07	221		SNS 7(,XR2),X'E4'	STORE LPIAR IN REG SAVE AREA
0115	B0	E6	09	222		SNS 9(,XR2),X'E6'	STORE LPDAR IN REG SAVE AREA
				223		*****	
				224	*	THE FOLLOWING THREE INSTRUCTIONS MUST START AT X'0118'	*
				225	*	BECAUSE OF MFCU-LESS SYSTEM.	*
				226		*****	
0118	B0	F4	0B	227		SNS 11(,XR2),X'F4'	STORE MPTAR IN REG SAVE AREA
011B	B0	F5	0D	228		SNS 13(,XR2),X'F5'	STORE MRDAR IN REG SAVE AREA
011E	B0	F6	0F	229		SNS 15(,XR2),X'F6'	STORE MPCAR IN REG SAVE AREA
0121	0C	03	05BC 0370	231		MVC NXBUSY(4),DPCSN	SET DUMP RTN @ ON TRANS QUEUE
0127	35	02	048E	232		L NP1BEG,XR2	POINT XR2 AT END OF SUPERVISOR
012B	36	02	036A	233		A M1,XR2	SUBTRACT 1 FROM XR2
012F	8C	1D	00 0399	234		MVC 0(30,XR2),REGEND	MOVE SAVED REGS INTO SAVE AREA
0134	F3	E0	02	235		SIO X'02',X'E0'	SPACE PRINTER
0137	C1	E6	0137	236		TIO *,X'E6'	WAIT ON PRINTER
013B	30	E3	00F9	237	CHKPRT	SNS TMP1,X'E3'	SENSE FOR 48 CHARACTER CHAIN
013F	38	04	00F9	238		TBN TMP1,X'04'	48 CHARACTER CHAIN
0143	F2	90	04	239		JF DSNS	JUMP IF NO



\$@SPV1 - SUPERVISOR FOR A DEDICATED SYSTEM

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER	MOD	00	09/05/10	PAGE	7	
0146	3C	31	01A5		240		MVI DIMEND+1,X'31'							IF YES, RESET COMPARE END
				014A	241	DSNS	EQU *							
014A	35	01	0011		242		L NCSYS@,XR1							POINT XR1 AT SYST COMN AREA
014E	7D	60	07		243		CLI NCPRTZ(,XR1),X'60'							CHECK FOR 96 POSITION PRINTER
0151	C0	01	03B1		244		BNE MODPRT							GO SETUP FOR A 120 POSITION
0155	0D	01	00E9	0376	245	NOTMOD	CLC FESW(2),CEFE							IS 'CEFE' IN ADDRESS SWITCHES
015B	C0	81	0284		246		BE DMPFND							JUMP IF CEFE DUMP REQUEST
				015F	247	DUMPBG	EQU *							OTHERWISE GIVE CORE DUMP
015F	C2	01	0000		248		LA SPBEG,XR1							POINT XR1 TO DUMP START

\$@SPV1 - SUPERVISOR FOR A DEDICATED SYSTEM

ERR	LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 09/05/10 PAGE 8
				250	*****	
				251	* START OF MAIN CORE DUMP LOOP *	
				252	*****	
0163	3C	40 03FF		253	LOOPB MVI PBUF,C' ' CLEAR THE PRINT BUFFER	
0167	0C	82 03FE 03FF		254	MVC PBUF-1(131),PBUF *	
016D	C2	02 0385		255	LOOPC LA PRTBUF,XR2 POINT XR2 AT PRINT BUFFER	
0171	3C	80 021D		256	MVI DUPLN1+1,X'80' RESTORE TO AN UNCONDITIONAL BR	
0175	9C	17 22 17		257	MOD1 MVC 34(24,XR2),23(,XR1) MOVE IN 24 BYTES OF CORE	
0179	BC	5C 23		258	MOD2 MVI 35(,XR2),C'*' INSERT * ON RIGHT	
017C	B4	01 0A		259	MOD6 ST 10(,XR2),XR1 STORE CORE ADD ON LEFT OF 24	
017F	34	01 0203		260	LOOPN ST CORE@,XR1 STORE XR1-CORE ADDRESS	
0183	E2	01 22		261	MOD3 LA 34(,XR2),XR1 USE XR1 TO POINT TO CORE	
				262	* CORE IMAGE IN BUFFER	
0186	98	03 00 00		263	MOVE MNN 0(,XR2),0(,XR1) MOVE NUM TO HEX PRINT BUFFER	
018A	C0	87 033E		264	B CHALPH BRANCH TO CHECK IF ALPHA	
018E	98	02 00 00		265	MNZ 0(,XR2),0(,XR1) MOVE ZONE TO HEX PRINT BUFFER	
0192	C0	87 033E		266	B CHALPH BRANCH TO CHECK IF ALPHA	
0196	4D	00 00 0400		267	DCMP CLC 0(1,XR1),CHAIN COMPARE AGAINST CHAIN IMAGE	
019B	F2	81 11		268	JE RTURN JUMP IF CHAR LOCATED IN IMAGE	
019E	0E	00 019A 05B4		269	ALC DCMP+4(1),NCX01 INCREMENT POINTER BY 1	
01A4	3D	78 019A		270	DIMEND CLI DCMP+4,X'78' END OF IMAGE	
01A8	C0	82 0196		271	BL DCMP REPEAT LOOP IF CHAR NOT FOUND	
01AC	7C	4B 00		272	MVI 0(,XR1),C'.' REPLACE UNFOUND CHAR WITH '.'	
01AF	3C	00 019A		273	RTURN MVI DCMP+4,X'00' RESET POINTER	
01B3	36	01 036A		274	A M1,XR1 SUBTRACT 1 FROM CORE POINTER	
01B7	34	01 00F9		275	ST TMP1,XR1 CHECK TO SEE IF 4 BYTES HAVE	
01BB	38	03 00F9		276	TBN TMP1,X'03' BEEN CONVERTED	
01BF	F2	90 04		277	JF SAVE2 IF NOT JUMP	
01C2	36	02 036A		278	A M1,XR2 OTHERWISE SUB 1 FROM HEX BUFFER	
				279	* TO FORMAT IN GROUPS OF 4	
01C6	34	02 00F9		280	SAVE2 ST TMP1,XR2 CHECK IF ALL BYTES HAVE BEEN	
01CA	3D	78 00F9		281	CLI TMP1,X'78' CONVERTED	
01CE	C0	84 0186		282	BH MOVE IF NO CONTINUE	
01D2	4C	01 02 036C		283	MVC 2(2,XR1),ASKTR SET UP LEFT DELIMITER	

ERR	LOC	OBJECT CODE	ADDR	STMT	SOURCE	STATEMENT	VER	MOD	00	09/05/10	PAGE	9
				285		*****						
				286	*	COMMON PRINT ROUTINE						*
				287		*****						
01D7	31	E4	0374	288	PRSION	LIO PTCHNN,X'E4'						LOAD PRINTER DATA ADDRESS REG
01DB	31	E6	0372	289		LIO PTBUFN,X'E6'						LOAD PRINTER IMAGE ADDRESS REG
01DF	F3	E2	02	290		SIO 2,X'E2'						PRINT AND SPACE TWO LINES
01E2	C1	E2	01E2	291		TIO *,X'E2'						WAIT ON CARRIAGE BUSY
01E6	B0	E3	80	292		SNS 128(,XR2),X'E3'						SENSE PRINTER CHECK INFO
01E9	B8	01	7F	293		TBN 127(,XR2),X'01'						TEST FOR PRINTER CHECKS
01EC	F2	90	07	294		JF PRTOKN						JUMP IF NO PRINTER CHECKS
01EF	F0	3E	6C	295		HPL X'6C',X'3E'						'PC' HALT
01F2	C0	87	01D7	296		B PRSION						RETRY PRINT OPERATION IF ERROR
01F6	3C	40	03FF	297	PRTOKN	MVI PBUF,C'						BLANK OUT PRINT BUFFER
01FA	0C	82	03FE 03FF	298		MVC PBUF-1(131),PBUF						
0200	C2	01	0000	299	DUPCHK	LA #,XR1						RESTORE XR1
0204	0D	01	037B 0203	300		CLC MXO(2),CORE@						CHECK FOR END OF DUMP
020A	F2	04	1B	301		JNH LASTLN						YES, GO PRINT LAST LINE
020D	5D	17	17 2F	302	MOD4	CLC 23(24,XR1),47(,XR1)						CHECK FOR A DUPLICATED LINE
0211	D2	01	18	303	MOD5	LA 24(,XR1),XR1						POINT XR1 AT NEXT 24 BYTES
0214	34	01	0203	304		ST CORE@,XR1						SAVE XR1
0218	C0	01	016D	305		BNE LOOPE						NOT DUP LINES, PRINT NEXT LINE
021C	C0	80	0200	306	DUPLN1	BC DUPCHK,X'80'						IF NOT 1ST DUP LINE DON'T PRINT
0220	3C	87	021D	307		MVI DUPLN1+1,X'87'						SET BRANCH TO UNCONDITIONAL
0224	C0	87	01D7	308		B PRSION						BRANCH TO PRINT BLANK LINE
				310		*****						
				311	*	ROUTINE TO PRINT LAST LINE						*
				312		*****						
0228	F2	80	4A	313	LASTLN	JC FORMAT,X'80'						FIRST TIME INDICATOR
022B	3C	87	0229	314		MVI LASTLN+1,X'87'						SET 1ST TIME BRANCH TO UNCOND
022F	C2	02	0385	315		LA PRTBUF,XR2						POINT XR2 TO PRINT BUFFER
0233	D2	01	18	316		LA 24(,XR1),XR1						UPDATE XR1
0236	0C	00	0377 0203	317		MVC TEMP(1),CORE@						MOVE LAST DIGIT OF CORE ADD
023C	0F	00	0377 037B	318		SLC TEMP(1),MXO						SUBTRACT D8 FROM LAST DIGIT
0242	F2	81	19	319		JE DO16						RESULT 0 - 16 BYTES LEFT
0245	38	10	0377	320		TBN TEMP,X'10'						CHECK IF ALL BYTES DONE
0249	F2	10	29	321		JT FORMAT						ALL BYTES DONE, GET OUT
024C	9C	07	12 07	322		MVC 18(8,XR2),7(,XR1)						MOVE 8 BYTES TO PRINT BUF
0250	BC	5C	13	323		MVI 19(,XR2),C'*'						PUT IN * OK THE RIGHT
0253	3C	36	0185	324		MVI MOD3+2,X'36'						MODIFY LOAD ADDRESS INSTR
0257	C2	02	0391	325		LA LNO1,XR2						POINT XR2 AT PRINT BUFFER
025B	F2	87	0F	326		J EXIT						JUMP TO EXIT ROUTINE
025E	9C	0F	1A 0F	327	DO16	MVC 26(16,XR2),15(,XR1)						MOVE 16 BYTES TO PRINT BUF
0262	BC	5C	1B	328		MVI 27(,XR2),C'*'						PUT IN * ON THE RIGHT
0265	3C	2C	0185	329		MVI MOD3+2,X'2C'						MODIFY LOAD ADDRESS INST
0269	C2	02	03A3	330		LA LNO2,XR2						POINT XR2 AT PRINT BUFFER
026D	34	01	03BF	331	EXIT	ST LNO3,XR1						SAVE XR1
0271	C0	87	017F	332		B LOOPN						BRANCH TO CONVERT LAST BYTES
				334		*****						
				335	*	LOAD INTO CORE THE FORMAT DUMP TRANSIENT						*
				336		*****						
0275	C2	02	006C	337	FORMAT	LA IPLPRM,XR2						POINT XR2 AT LOAD PARM LIST
0279	8C	07	07 02DB	338		MVC 7(8,XR2),NAME+7						MOVE IN MODULE NAME-\$\$\$SPD_
027E	BC	F1	06	339		MVI 6(,XR2),C'1'						SET FOR \$\$\$SPD1-CORE DUMP
0281	F2	87	58	340		J DP01						JUMP TO LOAD \$\$\$SPD1
				342		*****						

ERR	LOC	OBJECT CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00 09/05/10 PAGE 10
				343	*	ROUTINE DETERMINES THE REQUESTED DUMP AND TAKES THE	*
				344	*	NECESSARY ACTION	*
				345	*****	*****	*****
0284	F0	50 6F		346	DMPFND HPL	X'6F',X'50'	'50' HALT-START OF CEFE DUMP
0287	C2	02 006C		347	LA	IPLPRM,XR2	ADD OF DUMP MODULE LOAD PARM
028B	B0	00 0B		348	SNS	11(,XR2),X'00'	SENSE DATA ADDRESS REG-OPTION
028E	BA	F0 0B		349	SBN	11(,XR2),X'F0'	SET HIGH ORDER BITS ON-'FOX'
0291	BD	F1 0B		350	CLI	11(,XR2),C'1'	1,DTF OR 0,CORE DUMP REQUEST @02
0294	F2	82 92		351	JL	SETCBD	0,GO SET CORE DUMP LIMITS @02
0297	8C	07 07 02DB		352	MVC	7(8,XR2),NAME+7	MOVE IN MODULE NAME-\$\$\$SPD_ @02
029C	F2	81 3D		353	JE	DP01	1,DTF DUMP REQUESTED ? @02
029F	AC	00 06 0B		354	MVC	6(1,XR2),11(,XR2)	CHANGE NAME TO PROPER FOR @02
02A3	8E	00 06 0313		355	ALC	6(1,XR2),TWO	THIS REQUEST. @02
02A8	BD	F4 0B		356	CLI	11(,XR2),C'4'	4,TAPE DUMP REQUESTED ? @02
02AB	F2	81 6A		357	JE	DP03	YES-DUMP FOR TAPE DUMP @02
02AE	BD	F6 0B		358	CLI	11(,XR2),C'6'	6,OPTION @03
02B1	F2	81 11		359	JE	DPEOJ	YES, INVALID EXCEPT FOR 5404 @03
02B4	38	80 006B		360	TBN	XTA,D5448	5448 SUPPORTED ? @03
02B8	F2	90 04		361	JF	MODHI	NO, CONTINUE @03
02BB	3C	F7 02C0		362	MVI	MODHI+1,C'7'	YES, PERMIT 7 OPTION @03
02BF	BD	F5 0B		363	MODHI CLI	11(,XR2),C'5'+#	5 OR 7 OPTION HIGH OPTION # @03
02C2	F2	04 1B		364	JNH	DP04	2,5444; 3,5445; 4,TAPE @02
				365	*		5,SWA; 7,5448 @03
02C5	3D	FE 00E9		366	DPEOJ CLI	FESW,X'FE'	CONTINUE WITH OPTIONS ? @02
02C9	C0	81 013B		367	BE	CHKPRT	YES, RE-DO INITIALIZATION @02
02CD	F0	7C 63		368	HPL	X'63',X'7C'	NO, GIVE 'EJ' HALT @02
02D0	C0	87 02C5		369	B	DPEOJ	REPEAT ABOVE LOOP @02
			02D4	370	NAME EQU	*	@02
02D4	D65B5BE2D7C4C6E2	02DB		371	DC	CL8'O\$\$\$PDFS'	NAME FOR LOAD PARM LIST @02
02DC	3C	00 02F9		372	DP01 MVI	DP05+2,X'00'	SET JUMP TO NO-OP @02
02E0	C2	01 0003		373	DP04 LA	DIODQ1,XR1	POINT XR1 TO DISK QUEUE @02
02E4	7C	00 0E		374	MVI	LASTSK(,XR1),X'00'	SET LAST SEEK TO CYL 0 @02
02E7	5C	03 03 05		375	DP06 MVC	QLAST(4,XR1),QSELF(,XR1)	RESET QUEUE POINTERS @02
02EB	4D	01 0D 0068		376	CLC	NXTQUE(2,XR1),DISKQ	CHECK FOR LAST QUE MEMBER @02
02F0	75	01 0D		377	L	NXTQUE(,XR1),XR1	POINT TO NEXT QUE MEMBER @02
02F3	C0	01 02E7		378	BNE	DP06	IF NOT LAST, REPEAT LOOP @02
02F7	F2	87 1E		379	DP05 J	DP03	GO TO GET THE MODULE @02
02FA	3A	80 05BC		380	SBN	NXBUSY,X'80'	SET TRANSIENT AREA ACTIVE @02
02FE	3C	81 048C		381	MVI	NP1RIB,X'81'	SET TO DO A FAKE FIND @02
0302	C0	87 0004		382	B	NCENTR	BRANCH TO GENERAL ENTRY
0306	C1		0306	383	DC	XL1'C1'	RIB TO FIND SELECTED DUMP PROG
0307	2C	01 0312 01		384	MVC	DP02(2),1(,XR2)	MOVE IN C/S ADDRESS
030C	C0	87 0004		385	B	NCENTR	BRANCH TO GENERAL ENTRY
0310	C0		0310	386	DC	XL1'C0'	RIB FOR TRANS BY C/S/#
0311	0000		0312	387	DP02 DC	XL2'0000'	C/S ADDRESS OF TRANSIENT
0313	02		0313	388	TWO DC	XL1'02'	READ 3 SECTORS @02
0314	C0	87 02C5		389	B	DPEOJ	REPEAT ABOVE LOOP
0318	8C	01 09 048E		390	DP03 MVC	9(2,XR2),NP1BEG	PROGRAM BEGINNING ADDRESS @02
031D	BC	00 07		391	MVI	7(,XR2),X'00'	INITIALIZE PARAMETER
0320	3B	C0 05BC		392	SBF	NXBUSY,X'C0'	SET TRANSIENT LEVEL INACTIVE
0324	C0	87 0004		393	B	NCENTR	BRANCH TO GENERAL ENTRY
0328	75		0328	394	DC	XL1'75'	SYSTEM FETCH/FIND
				396	*****	*****	*****
				397	*	THE FOLLOWING DETERMINES THE LIMITS OF CORE TO BE DUMPED	*
				398	*****	*****	*****
0329	F0	50 7C		399	SETCBD HPL	X'7C',X'50'	'5E' HALT-SET CORE BANDS
032C	B0	00 0B		400	SNS	11(,XR2),X'00'	SENSE DATA SWITCHES FOR ADDRESS

\$@SPV1 - SUPERVISOR FOR A DEDICATED SYSTEM

ERR	LOC	OBJECT CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00 09/05/10 PAGE 11
032F	2C 00	037A 0B	401	MVC	MXO-1(1),11(,XR2)	SET END OF CORE BAND TO DUMP	
0334	BC 00	0B	402	MVI	11(,XR2),X'00'	ZERO OUT HIGH ORDERS BITS	
0337	B5 01	0B	403	L	11(,XR2),XR1	POINT XR1 AT CORE DUMP START	
033A	C0 87	0163	404	B	LOOPB	GO DUMP MAIN CORE	
			405	*****			
			406	*	ROUTINE TO CONVERT EBCDIC CHARACTER TO HEX DUMP	*	
			407	*****			
033E	BA F0	00	408	CHALPH SBN	0(,XR2),X'F0'	ASSUME NUMERIC	
0341	34 08	0357	409	ST	ENDRTR+3,ARR	SAVE CALLER RETURN ADDRESS	
0345	BD FA	00	410	CLI	0(,XR2),X'FA'	IT IS NUMERIC	
0348	F2 82	05	411	JL	SUBSTN	JUMP IF YES	
034B	8F 00	00 0368	412	SLC	0(1,XR2),HX39	SUB X'39' TO MAKE EBDIC ALPHA	
0350	36 02	036A	413	SUBSTN A	M1,XR2	SUB 1 FROM HEX BUFF POINTER	
0354	C0 87	0000	414	ENDRTR B	#	RETURN TO CALLER	
			416	*****			
			417	*	CONSTANTS AND EQUATES USED BY THE DUMP PROGRAM	*	
			418	*****			
0368			419	ORG	X'0368'		
0368	39		0368 420	HX39 DC	XL1'39'	HEX 39-FOR HEX TO EBCDIC CONVRT	
0369	FFFF		036A 421	M1 DC	XL2'FFFF'	USED TO SUBTRACT ONE BY ADDING	
036B	5C40		036C 422	ASKTR DC	CL2'*'	CHARACTER *	
			0203 423	CORE@ EQU	DUPCHK+3	SAVE AREA-LAST CORE AREA ADDRESS	
			00F9 424	TMP1 EQU	X'00F9'		
			0080 425	D5448 EQU	X'80'	TEST FOR 5448 SUPPORT @03	
			00E9 426	FESW EQU	X'00E9'	FE SWITCH	
036D	00D00280		0370 427	DPCSN DC	XL4'00D00280'	DISK ADDRESS OF DUMP ROUTINE	
0371	037C		0372 428	PTBUFN DC	XL2'037C'	ADDRESS OF PRINT BUFFER	
0373	0400		0374 429	PTCHNN DC	XL2'0400'	ADDRESS OF CHAIN IMAGE AREA	
0375	CEFE		0376 430	CEFE DC	XL2'CEFE'	USED IN CHECK FOR CEFE	
0377	00		0377 431	TEMP DC	XL1'00'	TEMP STORAGE FOR LAST LINE ROUT	
0378	03C7		0379 432	PRT120 DC	XL2'03C7'	120 PRINT BUFFER ADDRESS	
037A			433	ORG	X'037A'	MXO MUST BE AT X'037A'	
037A	00D8		037B 434	MXO DC	XL2'00D8'	MAXIMUM CORE TO DUMP	

\$@SPV1 - SUPERVISOR FOR A DEDICATED SYSTEM

ERR	LOC	OBJECT CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00 09/05/10 PAGE 12
				436		*****	
				437	*	NUCLEUS INIALIZATION LOGIC DURING IPL, CLEAR CORE.	*
				438		*****	
037C				439	ORG	X'037C'	MUST REMAIN AT X'037C'
037C 0000			037D	440	CORADD DC	XL2'0000'	SAVE AREA CORE SIZE SET BY NIP
0380				441	ORG	X'0380'	MUST REMAIN AT X'0380'
0380 0100			0381	442	DC	AL2(CDUMPD)	@ OF START OF RESIDENT DUMP @01
				443		*****	
				444	*	CLEAR CORE FROM TOP TO END OF SUPERVISOR	*
				445		*****	
0382 35 01 037D				446	L	CORADD,XR1	LOAD XR1 WITH HIGH CORE ADDRESS
0386 7C FF 01				447	IPLFF MVI	1(,XR1),X'FF'	SET STARTING BYTE TO HEX 'FF'
0389 5C FE 00 01				448	MVC	0(255,XR1),1(,XR1)	CLEAR 256 BYTES TO HEX 'FF'
038D 36 01 03AE				449	A	M256,XR1	SUBTRACT 256 FROM XR1
0391 34 01 03B0				450	ST	TEND,XR1	SAVE XR1-CORE ADDRESS FOR CMPAR
0395 0D 01 03B0 048E				451	CLC	TEND(2),NP1BEG	CHECK FOR END OF CLEAR CORE
039B C0 84 0386				452	BH	IPLFF	NO, REPEAT LOOP-CLEAR 256 MORE
039F C2 02 006C				453	LA	IPLPRM,XR2	YES, POINT XR1 AT 010B
03A3 8C 01 09 048E				454	MVC	9(2,XR2),NP1BEG	SET SCHEDULER LOAD ADDRESS
03A8 C0 87 0004				455	B	NCENTR	BRANCH TO GENERAL ENTRY
03AC 75			03AC	456	DC	XL1'75'	RIB FOR FETCH/FIND (\$\$TMIP)
03AD FF00			03AE	457	M256 DC	XL2'FF00'	MINUS 256-USED TO SUBTRACT
03AF 0000			03B0	458	TEND DC	XL2'00'	SAVE AREA-CORE ADDRESS

				460	*****		
				461	*	ROUTINE MODIFIES INSTRUCTIONS FOR 120 POSITION PRINTER.	*
				462	*****		
03B1	3C 1F	0176		463	MODPRT MVI	MOD1+1,X'1F'	
03B5	3C 28	0177		464	MVI	MOD1+2,X'28'	
03B9	3C 1F	0178		465	MVI	MOD1+3,X'1F'	
03BD	3C 29	017B		466	MVI	MOD2+2,X'29'	
03C1	3C 28	0185		467	MVI	MOD3+2,X'28'	
03C5	3C 1F	020E		468	MVI	MOD4+1,X'1F'	
03C9	3C 1F	020F		469	MVI	MOD4+2,X'1F'	
03CD	3C 3F	0210		470	MVI	MOD4+3,X'3F'	
03D1	3C 20	0213		471	MVI	MOD5+2,X'20'	
03D5	3C E0	037B		472	MVI	MXO,X'E0'	
03D9	0C 01	0170	0379	473	MVC	LOOPC+3(2),PRT120	
03DF	3C 87	0229		474	MVI	LASTLN+1,X'87'	
03E3	3C 08	017E		475	MVI	MOD6+2,X'08'	
03E7	C0 87	0155		476	B	NOTMOD	
				477	*****		
				478	*	DEFAULT PRINTER CHAIN IMAGE.	*
				479	*****		
0400				480	ORG	X'0400'	
0400	F1F2F3F4F5F6F7F8	0417		481	DC	CL24'1234567890#@/STUVWXYZ&,( '	
0408	F9F07B7C61E2E3E4			481			
0410	E5E6E7E8E9506B4D			481			
0418	D1D2D3D4D5D6D7D8	042F		482	DC	CL24'JKLMNOPQR-\$*ABCDEFGHI+.)'	
0420	D9605B5CC1C2C3C4			482			
0428	C5C6C7C8C94E4B5D			482			
				483	*		
0430	4040404040404040	0477		484	DC	72XL1'40'	FOR 120 CHAR UCS CHAIN.
0438	4040404040404040			484			
0440	4040404040404040			484			
0448	4040404040404040			484			
0450	4040404040404040			484			
0458	4040404040404040			484			
0460	4040404040404040			484			
0468	4040404040404040			484			
0470	4040404040404040			484			
				486	*****		
				487	*		*
				488	*	PROGRAM LEVEL ONE COMMUNICATION AREA (PLCA).	*
				489	*		*
				490	*****		
		0478		491	NP1COMN EQU	*	PROGRAM LEVEL ONE.
		0478		492	NP1IOB EQU	*	PROGRAM LEVEL IOB.
0478	0000	0479		493	DC	XL2'0000'	IOS CHAIN POINTER.
047A	00	047A		494	DC	XL1'00'	COMPLETION CODE.
047B	0000	047C		495	DC	XL2'0000'	IOB Q AND R BYTES.
047D	000000	047F		496	NP1NB DC	XL3'000000'	IOB CYL/SECT ADDRESS NUMBER.
0480	0000	0481		497	DC	XL2'0000'	DATA BUFFER ADDRESS.
0482	00000028	0485		498	DC	XL4'00000028'	SENSE, ERROR COUNT, FLAG BYTES.
0486	00000000	0489		499	DC	XL4'00000000'	OBR-SDR TRANSIENT AREA.
048A	FFFF	048B		500	NP1DTF DC	XL2'FFFF'	ADDRESS OF FIRST DIF.
048C	00	048C		501	NP1RIB DC	XL1'00'	PROGRAM RIB BYTE.
048D	0009	048E		502	NP1BEG DC	AL2(SUPEND)	PROGRAM LEVEL BEGINNING ADDRESS.
048F	4000	0490		503	NP1END DC	XL2'4000'	PROGRAM LEVEL ENDING ADDRESS.
0491	00	0491		504	NP1Q DC	XL1'00'	PROGRAM LEVEL Q BYTE.
0492	0000	0493		505	NP1RLF DC	XL2'0000'	PROGRAM RELOCATION FACTOR.

\$@SPV1 - SUPERVISOR FOR A DEDICATED SYSTEM

ERR	LOC	OBJECT CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00 09/05/10 PAGE 14
0494	0000		0495	506	NP1CYL DC	XL2'0000'	C/S OF FIRST LOAD FOR THIS LEVEL
0496	0000		0497	507	NP1OLB DC	XL2'0000'	C/S OF PROGRAM OBJECT LIBRARY.
0498	0000		0499	508	NP1XR1 DC	XL2'0000'	REGISTER ONE SAVE AREA.
049A	0000		049B	509	NP1XR2 DC	XL2'0000'	REGISTER TWO SAVE AREA.
049C	0000		049D	510	NP1NSI DC	XL2'0000'	RETURN ADDRESS.
049E	0000		049F	511	NP1ORL DC	XL2'0000'	PROGRAM OVERLAY RELOCATION FACT.
04A0	0000		04A1	512	NP1TXT DC	XL2'0000'	OVERLAY TEXT ADDRESS.
04A2	0000		04A3	513	NP1TMP DC	XL2'0000'	TEMPORARY STORAGE FOR LOADER.
04A4	000000000000000000		04AD	514	NP1PAR DC	XL10'00'	PGM. LEVEL PARM LIST FOR LOADER.
04AC	0000			514			
04AE	00		04AE	516	NP1EOJ DC	XL1'00'	END OF JOB.
04AF	00		04AF	517	NP1UPS DC	XL1'00'	UPSI SWITCH.
04B0	0000000000000000		04B5	518	NP1NAM DC	XL6'00000000000000'	PROGRAM NAME.
04B6	00		04B6	519	NP1REL DC	XL1'00'	RELEASE LEVEL.
04B7	0000000000000000		04BC	520	NP1DAT DC	XL6'00000000000000'	PROGRAM DATE.
04BD	00000000		04C0	521	NP1SIN DC	XL4'00000000'	SYSIN INDICATOR.
04C1	0000000000000000		04C7	522	NP1SCH DC	XL7'0000000000000000'	SCHEDULER BYTES.
04C8	0000		04C9	523	DC	XL2'0000'	MODEL 6 HALT SAVE AREA.
04CA	0000		04CB	524	DC	XL2'0000'	RESERVED.
04CC	0000		04CD	525	NP1WKB DC	XL2'0000'	DISK IOS WORK AREA.
04CE	00		04CE	526	NP1LOG DC	XL1'00'	DISK LOG UNIT ID FOR EOJ.
04CF	0000		04D0	527	NP1LEV DC	XL2'0000'	SCHEDULER BYTES.
04D1	0000		04D2	528	DC	XL2'0000'	2 SAVE AREA FOR 2ND LEVEL HALT
04D3	0000		04D4	529	DC	XL2'0000'	2 EOJ RETURN ADDRESS
04D5	00		04D5	530	DC	XL1'00'	1 SYSPUNCH ID



ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00 09/05/10 PAGE 15
			532	*****	*****	
			533	*	*	
			534	*TITLE:	'NENTRY'	
			535	*	*	
			536	*STATUS:	CHANGE LEVEL 8.	
			537	*	*	
			538	*FUCNTION/OPERATION:	DETERMINES CALLERS REQUEST THEN GOES TO PROPER	
			539	*	ROUTINE. IF TRANSIENT REQUEST THE TRANSIENT SCHEDULER WILL	
			540	*	BRING IN THE PROPER TRANSIENT AND GIVE IT CONTROL. AFTER	
			541	*	THE TRANSIENT EXITS, A CHECK WILL BE MADE FOR ANY TRANSIENTS	
			542	*	THAT REMAIN ON THE QUEUE AND IF ANY EXITS, THEY WILL BE	
			543	*	REFRESHED. WHEN THE QUEUE IS EMPTY THE SCHEDULER WILL TEST	
			544	*	IF THE LOADER IS ALSO NEEDED, IF IT IS THE LOADER WILL GET	
			545	*	CONTROL OTHER WISE GENERAL EXIT WILL GET CONTROL.	
			546	*	*	
			547	*ENTRY POINTS:	NENTRY THE ARR POINTS TO THE RIB.	
			548	*	IF TRANSIENT REQUEST BY CYLINDER/SECTOR THE C/S/#	
			549	*	WILL BE THE NEXT 3 BYTES AFTER THE RIB.	
			550	*	IF LOADER REQUEST XR2 WILL POINT TO THE PARAMETER	
			551	*	LIST FOR THE LOADER (EITHER 7 OR 10 BYTES).	
			552	*	*	
			553	*INPUT-	THE REQUEST INDICATOR BYTE(RIB) AND IF REQUIRED EITHER	
			554	*	3 BYTES C/S/# FOR TRANSIENT SCHEDULER OR PARAMETER LIST	
			555	*	LIST LOADER.	
			556	*	*	
			557	*OUTPUT-	NONE	
			558	*	*	
			559	*EXTERNAL ROUTINES:	NREAD INTEFACE WITH IOS DETECTS ALL DISK ERRORS	
			560	*	AND CALLS PROPER HALT OR ABORT	
			561	*	*	
			562	*EXITS-NORMAL:	IF EVERY OK RETURN TO CALLER OTHERWISE BRANCH TO DUMP	
			563	*	AND ABORT.	
			564	*	A RIB OF X'01' MEANS A RIB FOR CCP WAS RECEIVED	
			565	*	SO A J- HALT IS ISSUED AND JOB IS CANCELED.	
			566	*	*	
			567	*TABLES/WORK AREAS:	PROGRAM LEVEL AND SYSTEM COMMUNICATION AREA	
			568	*	TRANSIENT SCHEDULER C/S ADDR TABLE	
			569	*	TRANSIENT QUEUE	
			570	*	TRANSIENT BUSY BYTE AND QUEUE COUNTER	
			571	*	TEMPORY STORAGE FOR RIB FOR TRANSIENT SCHEDULER	
			572	*	*	
			573	*CHARACTER CODE DEPENDENCY:	A	
			574	*	*	
			575	*NOTES:	N/A.	
			576	*	*	
			577	*ATTRIBUTES:	REUSABLE	
			578	*	*	
			579	*****	*****	
		04D6	580	ENTRY	NENTRY	ENTRY POINT FOR GENERAL ENTRY
		05BB	581	ENTRY	NXACTV	ACTIVE TRANSIENT ID BYTE.
		04EB	582	USING	NXNSI, XR2	ESTABLISH XR2 AS BASE REGISTER
		04D6	583	NENTRY	EQU *	
04D6	34 02 0627		584	NE01	ST NXXR2, XR2	SAVE XR2-POINTS AT CALLERS PARM
04DA	C2 02 04EB		585	LA	NXNSI, XR2	USE NE10 AS A BASE-XR2
04DE	B4 08 00		586	ST	NXNSI(, XR2), ARR	SAVE ARR AT NE10+4 (RIB POINTE)
04E1	0C 01 05E9 005C		587	MVC	NXTABL+9, SYSLOG-1(2)	SET C/S TABLE FOR CURRENT SLOG
04E7	8C 00 CD 0000		588	NE10	MVC NXRIB(1, XR2), ##	STORE RIB IN WORK AREA
04EC	AE 01 00 C9		589	ALC	NXNSI(2, XR2), NCX01(, XR2)	UPDATE NSI TO ONE BYTE PAST RIB

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER	15,	MOD	00	09/05/10	PAGE	16
04F0	BD	00	CD	590		CLI	NXRIB(,XR2),X'00'							REQ FOR PROG LEVEL COMN AREA ?
04F3	F2	01	08	591		JNE	NOTPOT							JUMP IF NOT
04F6	C2	02	0478	592		LA	N1COMN,XR2							LOAD ADDR OF PGM COMN AREA
04FA	35	10	04EB	593		L	NXNSI,IAR							LOAD IAR WITH RETURN ADDRESS
				594			*****							*****
				595	*		HALT J- IF CCP RIB 01 RECEIVED							*
				596			*****							*****
04FE	BD	01	CD	597	NOTPOT	CLI	NXRIB(,XR2),CCPRIB							RIB = 01 ?
0501	F2	01	0B	598		JNE	VLDTRIB							NO-CONTINUE
0504	F0	63	10	599		HPL	HD,HJ							HALT J-
0507	C2	01	0478	600		LA	N1COMN,XR1							SET PTR TO PLCOMM
050B	C0	87	0042	601		B	HARDAB							IMMEDIATE CANCEL
050F	B8	80	D1	602	VLDTRIB	TBN	NXBUSY(,XR2),X'80'							IS TRANSIENT AREA ACTIVE ?
0512	F2	10	1E	603		JT	NXBUZ							JUMP IF IT IS ACTIVE
				604	*									IS THIS A REFRESH REQUEST WITH
0515	B8	C0	CD	605		TBN	NXRIB(,XR2),X'C0'							OUT AN ACTIVE TRANSIENT AREA
0518	F2	90	03	606		JF	NOOBR							JUMP IF NOT A REDRESH REQUEST
051B	BB	40	CD	607		SBF	NXRIB(,XR2),X'40'							SET OFF REFRESH BIT IF IT IS ON
051E	34	01	0499	608	NOOBR	ST	NP1XR1,XR1							STORE XR1 IN PGM COMN AREA
0522	C2	01	0478	609		LA	N1COMN,XR1							POINT XR1 TO PGM COMN AREA
0526	4C	01	23 062A	610		MVC	NPXR2(2,XR1),NXXR2+3							SAVE XR2 IN PGM COMN AREA
052B	6C	00	14 CD	611		MVC	NPRIB(1,XR1),NXRIB(,XR2)							STORE RIB IN PGM COMN AREA
052F	6C	01	25 00	612		MVC	NPNSI(2,XR1),NXNSI(,XR2)							STORE CALL NSI IN PGM COMN AREA
0533	B8	80	CD	613	NXBUZ	TBN	NXRIB(,XR2),X'80'							IS THIS A TRANSIENT REQUEST ?
0536	F2	10	17	614		JT	NXSCHD							JUMP IF IT IS TRANSIENT REQ
0539	35	01	062A	615		L	NXXR2+3,XR1							POINT XR1 TO CALLERS PARM LIST
053D	1C	09	04AD 09	616		MVC	NP1PAR(10),9(,XR1)							STORE CALLERS PARMS IN COMN AREA
0542	C2	01	0478	617		LA	N1COMN,XR1							POINT XR1 TO PGM COMN AREA.
0546	B8	41	CD	618		TBN	NXRIB(,XR2),X'41'							LOAD WITH A FIND REQUEST ?
0549	C0	81	066B	619		BE	NXEND							JUMP TO LOADER IF NOT WITH FIND
054D	BC	81	CD	620		MVI	NXRIB(,XR2),X'81'							SET TEMPORARY RIB FOR A FIND
0550	2C	01	065A CA	621	NXSCHD	MVC	NXBRT0+3(2),NXSNT(,XR2)							SET BR TO TRAN AREA-X'0100'
0555	B8	3F	CD	622		TBN	NXRIB(,XR2),X'3F'							REQUEST BY CYL/SECT/NUM ?
0558	F2	90	0E	623		JF	NX010							JUMP IF NOT BY C/S/#
055B	AE	01	00 CB	624		ALC	NXNSI(2,XR2),NCX03(,XR2)							UPDATE NSI BY 3 (CS# IN LINE)
055F	B8	80	D1	625		TBN	NXBUSY(,XR2),X'80'							IS TRANSIENT AREA ACTIVE ?
0562	F2	10	04	626		JT	NX010							JUMP IF IT IS ACTIVE
0565	6C	01	25 00	627		MVC	NPNSI(2,XR1),NXNSI(,XR2)							UPDATE NSI IN PGM COMN AREA
0569	BA	80	D1	628	NX010	SBN	NXBUSY(,XR2),X'80'							SET TRANSIENT AREA ACTIVE
056C	B8	40	CD	629		TBN	NXRIB(,XR2),X'40'							REFRESH REQUEST ?
056F	F2	90	26	630		JF	NXCS#							SKIP QUEING IF NOT REFRESH
0572	AE	00	D1 C9	631		ALC	NXCONT(1,XR2),NCX01(,XR2)							ADD 1 TO NR ON TRANSIENT QUEUE
0576	B8	06	D1	632	NX015	TBN	NXCONT(,XR2),X'06'							MORE THEN 5 TRANSIENT ON QUEUE ?
				633	*									USE Q BYTE FOR CONSTANT X'06'
0579	C0	10	003F	634		BT	NRDHPL							IF MORE THEN 5 HALT ' 1'
				635	*									TOO MANY TRANSIENTS ON QUEUE
				636			*****							*****
				637	*		ROUTINE PUSHES DOWN QUEUE & ADDS NEW REQUEST TO QUEUE							*
				638			*****							*****
057D	AC	06	D8 DF	639	NXPUCH	MVC	NXQUE-28(7,XR2),NXQUE-21(,XR2)							OK, PUSH QUEUE DOWN
0581	AC	06	DF E6	640		MVC	NXQUE-21(7,XR2),NXQUE-14(,XR2)							PUSH QUEUE DOWN
0585	AC	06	E6 ED	641		MVC	NXQUE-14(7,XR2),NXQUE-7(,XR2)							PUSH QUEUE DOWN
0589	AC	06	ED F4	642		MVC	NXQUE-7(7,XR2),NXQUE-0(,XR2)							PUSH QUEUE DOWN
058D	AC	02	F0 D0	643		MVC	NXQUE-4(3,XR2),NXACTV(,XR2)							PUT ACTIVE TRANSIENT ON QUE
0591	B4	01	F2	644		ST	NXQUE-2(,XR2),XR1							STORE XR1 IN QUEUE
0594	AC	01	F4 00	645		MVC	NXQUE(2,XR2),NXNSI(,XR2)							STORE CALLER NSI IN QUEUE
				646	*									NOT A REFRESH REQUEST, QUEING
				647	*									HAS BEEN SKIPPED

\$@SPV1 - SUPERVISOR FOR A DEDICATED SYSTEM

ERR	LOC	OBJECT CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00 09/05/10 PAGE 17
0598	C2	01 0478	648	NXCS#	LA	N1COMN,XR1	POINT XR1 TO COMN AREA
059C	1C	01 05EB 46	649		MVC	NXTABL+11,NPSYSI-2(2,XR1)	SET SYSIN C/S IN TABLE
05A1	B9	3F CD	650		TBF	NXRIB(,XR2),X'3F'	REQUEST BY CYL/SECT/NUM ?
05A4	F2	90 5B	651		JF	NX030	GO TO TABLE LOOK UP IF NOT
05A7	AC	01 C4 00	652		MVC	NX020+4(2,XR2),NXNSI(,XR2)	SET UP ADDRESS OF CS# IN MVC IN
05AB	4C	03 08 0000	653	NX020	MVC	NIONB+1(4,XR1),##	MOVE CYL/SEC/NUM & 1 BYTE GRAB
05B0	F2	87 65	654		J	NX050	SKIP TABLE LOOK UP

```

656 *****
657 * THESE ARE SOME OF THE CONSTANTS USED BY THE SUPERVISOR *
658 * THEY ARE HERE SO THE TRANSIENT SCHEDULER CAN USE A BASE *
659 * REGISTER TO REFERENCE THIS AREA-MUST BE IN THIS ORDER *
660 *****
05B3 0001 05B4 661 NCX01 DC XL2'0001' HEX CONSTANT FOR '0001'
05B5 00 05B5 662 NXSNT DC XL1'00' USED WITH '01' TO FORM '0100'
05B6 03 05B6 663 NCX03 DC XL1'03' HEX CONSTANT '03'
05B7 04 05B7 664 NCX04 DC XL1'04' HEX CONSTANT '04'
05B8 00 05B8 665 NXRIB DC XL1'00' TEMP STORAGE FOR THE RIB
05B9 000000 05BB 666 NRACTV DC XL3'000000' ACTIVE TRANSIENT ID BYTE
667 * CYL/SECT/NUM-TRANSIENT IN CORE
05BC 00 05BC 668 NXBUSY DC XL1'00' TRANSIENT ACTIVE INDICATOR
669 * BIT 0 ON TRANSIENT ACTIVE
670 * BIT 0 OFF TRANSIENT INACTIVE
05BD 0000000000000000 05BC 671 NXCONT EQU NXBUSY NUMBER OF TRANSIENTS ON QUEUE
05C5 0000000000000000 05DF 672 NXQUE DC XL35'00' TRANSIENT QUEUE-IN FOTM OF
05CD 0000000000000000 672
05D5 0000000000000000 672
05DD 000000 672
673 * C/S/#, XR1, NSI(ARR+1)
674 * THE QUEUE HAS THE FOLLOWING FORMAT
675 * NUMBER OF BYTES
676 * 1 1 1 2 2
677 *
678 * | CYL | SECT | NUMBER | XR1 | RETURN | NUMBER 5
679 * | | | OF SECT | | ADDRESS | ON QUEUE
680 *
681 * | CYL | SECT | NUMBER | XR1 | RETURN | NUMBER 4
682 * | | | OF SECT | | ADDRESS | ON QUEUE
683 *
684 * | CYL | SECT | NUMBER | XR1 | RETURN | NUMBER 3
685 * | | | OF SECT | | ADDRESS | ON QUEUE
686 *
687 * | CYL | SECT | NUMBER | XR1 | RETURN | NUMBER 2
688 * | | | OF SECT | | ADDRESS | ON QUEUE
689 *
690 * | CYL | SECT | NUMBER | XR1 | RETURN | NUMBER 1
691 * | | | OF SECT | | ADDRESS | ON QUEUE
692 *
693 *
04EB 694 NXNSI EQU NE10+4 SAVE AREA FOR RIB POINTER
    
```

			696	*****		
			697	*	THIS IS THE TABLE OF CYL/SECT OF ALL SYSTEM TRANSIENTS THAT	*
			698	*	CAN BE CALLED BY A RIB.	*
			699	*****		
		05E0	700	NXTABL EQU *		
05E0	0000	05E1	701	DC	XL2'0000'	FIND
05E2	0000	05E3	702	DC	XL2'0000'	OPEN
05E4	0000	05E5	703	DC	XL2'0000'	CLOSE
05E6	0000	05E7	704	DC	XL2'0000'	EOJ
05E8	0000	05E9	705	DC	XL2'0000'	HALT SYSLOG
05EA	0000	05EB	706	DC	XL2'0000'	SYSIN
05EC	0000	05ED	707	DC	XL2'0000'	SWA GET
05EE	0000	05EF	708	DC	XL2'0000'	SWA PUT
05F0	0000	05F1	709	DC	XL2'0000'	SWA READ/WRITE
05F2	0000	05F3	710	DC	XL2'0000'	VTOC READ/WRITE
05F4	0000	05F5	711	DC	XL2'0000'	ALLOCATION INITIATOR
05F6	0000	05F7	712	DC	XL2'0000'	PROGRAM/PROTECT
05F8	0000	05F9	713	DC	XL2'0000'	ROLL/IN
05FA	0000	05FB	714	DC	XL2'0000'	ROLL/OUT
05FC	0000	05FD	715	DC	XL2'0000'	RPG HALT PROCESSOR
05FE	0000	05FF	716	DC	XL2'0000'	OBR/SDR
0600	0000	0601	717	DC	XL2'0000'	5445 VTOC READ/WRITE
			718	*****		
			719	*	THIS IS THE REST OF THE TRANSIENT SCHEDULER. IT IS SPLIT	*
			720	*	THIS WAY BECAUSE THE OTHER HALF USES A BASE REGISTER	*
			721	*	REQUEST FOR SYSTEM TRANSIENT, DO TABLE LOOKUP.	*
			722	*****		
0602	BB C0 CD	723	NX030	SBF	NXRIB(,XR2),X'C0'	TURN OFF TWO HIGH ORDER BITS
0605	AE 00 CD CD	724		ALC	NXRIB(1,XR2),NXRIB(,XR2)	DOUBLE RIB-TABLE DISPLACEMENT
0609	2C 00 0614 CD	725		MVC	NX040+3(1),NXRIB(,XR2)	PUT IN DISPLCE IN MVC INSTRUCT
060E	E2 02 F4	726		LA	NXTABL-1(,XR2),XR2	POINT XR2 TO TRANSIENT TABLE
0611	6C 01 06 00	727	NX040	MVC	NIOSB(2,XR1),##(,XR2)	MOVE C/S INTO READ IOB
0615	7C 02 07	728		MVI	NIONB(,XR1),X'02'	SET IOB TO READ 3 SECTORS

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00 09/05/10 PAGE 20
			730		*****	
			731	*	REQUEST BY CYL/SECT/NUM, SKIP TABLE LOOK UP	*
			732		*****	
0618	4C 00 03 0060		733	NX050 MVC	NIOQB(1,XR1),N1SYSQ	SET IOB TO READ TRANSIENT IN
061D	4C 01 09 05B5		734	MVC	NIODAT(2,XR1),NXSNT	TRAN AREA ADD INTO DATA BUFF ADD
0622	1D 01 05BA 06		735	CLC	NXACTV-1(2),NIOB(,XR1)	IS TRANSIENT ALREADY IN CORE ?
0627	C2 02 0000		736	NXXR2 LA	##,XR2	RESTORE XR2
062B	C0 01 0029		737	BNE	NREAD	GO READ TRANSIENT INTO CORE
062F	F2 87 1F		738	J	NXSTID	GO BRANCH TO THE TRANSIENT
0632	0F 00 05BC 05B4		739	NXUNWD SLC	NXCONT(1),NCX01	SUBTRACT 1 FROM NUMBER ON QUE
0638	4C 02 07 05DB		740	MVC	NIONB(3,XR1),NXQUE-4	SET IOB TO REFRESH TRANSIENT
063D	C0 87 0029		741	B	NREAD	READ REFRESHED TRANSIENT
0641	35 01 05DD		742	NRD2 L	NXQUE-2,XR1	RESTORE XR1
0645	0C 01 065A 05DF		743	MVC	NXBRTO+3(2),NXQUE	SET BR TO TRANSIENT INTRUCTION
064B	0C 1B 05DF 05D8		744	MVC	NXQUE(28),NXQUE-7	POP UP THE QUEUE
0651	0C 02 05BB 047F		745	NXSTID MVC	NXACTV(3),NP1NB	SET ACTIVE TRANSIENT ID
0657	C0 87 0000		746	NXBRTO B	#	BRANCH TO TRANSIENT
065B	3B 04 006A		747	SBF	SYS+NCMBSV,X'04'	SET OFF CCP IOS INHIBIT BIT
065F	39 07 05BC		748	TBF	NXCONT,X'07'	ANY REQUESTS LEFT ON QUEUE ?
0663	C0 90 0632		749	BF	NXUNWD	GO TO UNWIND QUEUE IF YES
0667	3B 80 05BC		750	SBF	NXBUSY,X'80'	SET TRANSIENT AREA INACTIVE
066B	78 40 14		751	NXEND TBN	NPRIB(,XR1),X'40'	IS THIS A LOADER REQUEST ?
066E	C0 90 078F		752	BF	NEXIT	NO-RETURN TO CALLER
			753	*		YES-LOAD LEVEL 1 TRANSIENT

ERR	LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 09/05/10 PAGE 21
				755	*****	
				756	*	
				757	*TITLE: 'NLOADR' RESIDENT LOADER AND RELOCATION PROGRAM	
				758	*	
				759	*STATUS: CHANGE LEVEL 6.	
				760	*	
				761	*FUCNTION/OPERATION: TO LOAD IN PROGRAMS REQUESTED BY THE CALLER.	
				762	* THEN EITHER GIVE CONTROL BACK TO THE CALLER (LOAD) OR TO	
				763	* THE PROGRAM LOADED (FETCH).	
				764	*	
				765	*ENTRY POINTS: NLOADR XR1 POINTS TO PROGRAM LEVEL COMMUNICATION	
				766	* AREA (PLCA) AND LOADER PARM LIST IS IN THE AREA.	
				767	*	
				768	*INPUT- XR1 POINTS TO THE PROGRAM LEVEL COMMUNICATION AREA	
				769	*	
				770	*OUTPUT- NONE	
				771	*	
				772	*EXTERNAL ROUTINES: NREAD INTERFACE WITH IOS DETECTS ALL DISK ERRORS	
				773	* AND CALLS PROPER HALT	
				774	*	
				775	*EXITS-NORMAL: BR TO GENERAL EXIT	
				776	*	
				777	*ATTRIBUTES: REUSABLE	
				778	*	
				779	*****	
			0672	780	ENTRY NLOADR ENTRY POINT FOR LOADER	
0672	79	0C 14		781	NLOADR TBF NPRIB(,XR1),X'0C' IS THIS A FETCH REQUEST ?	
0675	F2	90 0E		782	JF NL005 IF NOT SKIP FETCH INSTR GROUP	
0678	5C	01 35 30		783	MVC NPLOD(2,XR1),NPLNK(,XR1) MOVE LINK ADDRESS TO LOAD ADDR	
067C	7D	40 14		784	CLI NPRIB(,XR1),X'40' LOAD MODULE BY RELATIVE C/S ?	
067F	F2	81 04		785	JE NL005 JUMP IF YES	
0682	5E	01 35 1B		786	ALC NPLOD(2,XR1),NPRLF(,XR1) ADD RELOCATION FACTOR TO LOAD @	
0686	78	04 14		787	NL005 TBN NPRIB(,XR1),X'04' IS THIS A SYST FETCH TO ADDR ?	
0689	F2	90 0C		788	JF NOFTCH JUMP IF NOT	
068C	5C	01 1D 2D		789	MVC NPCYL(2,XR1),NPCS(,XR1) SAVE C/S ADDRESS OF ROOT PHASE	
0690	5C	01 1B 35		790	MVC NPRLF(2,XR1),NPLOD(,XR1) FIND RELOCATION FACTOR BY SUBTR	
0694	5F	01 1B 30		791	SLC NPRLF(2,XR1),NPLNK(,XR1) LINK EDIT ADDR FROM PROG ADDR	
				792	*****	
				793	* ROUTINE TO CALCULATE ABSOLUTE CYLINDER/SECTOR OF	
				794	* OVERLAY MODULE FROM RELATIVE CYLINDER/SECTOR.	
				795	*****	
0698	79	1C 14		796	NOFTCH TBF NPRIB(,XR1),NRPGRQ LOAD MODULE BY RELATIVE C/S ?	
069B	F2	90 2B		797	JF NOTRPG SKIP CYCL CALCULATION IF NOT	
069E	5F	01 30 1B		798	SLC NPLNK(2,XR1),NPRLF(,XR1) ADJUST LOAD ADDR FOR OVERLAY MOD	
06A2	5C	01 2B 1D		799	MVC NPTEMP(2,XR1),NPCYL(,XR1) MOVE IN ROOT ABSOLUTE C/S	
06A6	79	7F 2D		800	NL010 TBF NPCS(,XR1),X'7F' IS RELATIVE C/S SECTOR COUNT 0 ?	
06A9	F2	10 19		801	JT NL030 YES-EXIT DECREMENT LOOP	
06AC	4F	00 2D 05B4		802	SLC NPCS(1,XR1),NCX01 DECREMENT BY ONE.	
06B1	4E	00 2B 05B7		803	ALC NPTEMP(1,XR1),NCX04 ADD 1 SECTOR '04' TO ABS SECTOR	
06B6	78	60 2B		804	TBN NPTEMP(,XR1),X'60' LAST SECTOR OF TRACK ?	
06B9	F2	90 05		805	JF NL020 NO-BYPASS TRACK INCREMENT	
06BC	4E	01 2B 0018		806	ALC NPTEMP(2,XR1),NCX20 INCREMENT TRACK BY ONE	
06C1	C0	87 06A6		807	NL020 B NL010 LOOP...	
06C5	5E	01 2D 2B		808	NL030 ALC NPCS(2,XR1),NPTEMP(,XR1) ADD ABST C/S TO RELATIVE CYL	
				809	*****	
				810	* CHECK FOR ENOUGH CORE MEMORY AVAILAIBLE	
				811	*****	
06C9	5D	01 35 16		812	NOTRPG CLC NPLOD(2,XR1),NPBEG(,XR1) ABOVE LOW CORE BOUNDARY ?	

ERR	LOC	OBJECT CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00 09/05/10 PAGE 22	
06CD	C0	82 003F	813	BL	NRDHPL	HALT IF BELOW SUPERVISOR		
			814	*****				
			815	*	SET UP IOB TO READ REQUESTED MODULE IN	*		
			816	*****				
06D1	4C	00 03 0060	817	NLODER	MVC NIOQB(1,XR1),N1SYSQ	SET IOB Q BYTE TO SYSTEM PACK		
06D6	78	20 14	818	TBN	NPRIB(,XR1),NSRQ	USE SYSTEM OR PROGRAM IOB ?		
06D9	F2	10 04	819	JT	NSS	JUMP IF SYSTEM IOB IS TO BE USED		
06DC	5C	00 03 19	820	MVC	NIOQB(1,XR1),NPQ(,XR1)	SET IOB TO PROGRAM PACK		
06E0	5C	01 09 35	821	NSS	MVC NIODAT(2,XR1),NPLOD(,XR1)	PUT LOAD ADDRESS IN IOB		
06E4	5C	02 07 2E	822	MVC	NIONB(3,XR1),NP#S(,XR1)	PUT C/S/# IN IOB		
06E8	4F	00 07 05B4	823	SLC	NIONB(1,XR1),NCX01	SUBTRACT ONE FROM NUM OF SECTOR		
06ED	C0	87 0029	824	B	NREAD	READ REQUESTED MODULE IN		
06F1	5C	01 27 35	825	NRD3	MVC NPORLF(2,XR1),NPLOD(,XR1)	SET OVERLAY RELOCATION FACTOR		
06F5	5F	01 27 30	826	SLC	NPORLF(2,XR1),NPLNK(,XR1)	SUBTRACT LINK EDIT ADDRESS		
			827	*				
06F9	F2	81 79	828	JE	NLEND2	SKIP AROUND RELOCATION IF EQUAL		
			829	*****				
			830	*	THIS IS THE RELOCATION PROGRAM	*		
			831	*****				
06FC	5C	01 29 35	832	LARLV1	MVC NPTXT(2,XR1),NPLOD(,XR1)	SET TEXT POINTER		
0700	5C	00 2B 31	833	MVC	NPTEMP(1,XR1),NPRLD(,XR1)	STORE DISPLACEMENT OF RLD'S		
0704	7C	00 2A	834	MVI	NPTEMP-1(,XR1),X'00'	RESET HIGH ORDER BYTE = 0		
0707	7D	00 31	835	CLI	NPRLD(,XR1),X'00'	DISPLACEMENT OF RLD'S = 0 ?		
070A	F2	01 03	836	JNE	NRE010	NO-DON'T INCREMENT		
070D	7C	01 2A	837	MVI	NPTEMP-1(,XR1),X'01'	YES, SET HIGH ORDER BYTE TO 1		
0710	5C	01 06 01	838	NRE010	MVC NIOSB(2,XR1),NPCHAN(,XR1)	SET CYLINDER/SECTOR		
0714	7C	02 07	839	MVI	NIONB(,XR1),X'02'	SET READ TO 3 SECTORS		
0717	3C	00 05B9	840	MVI	NXACTV-2,X'00'	SET ACTIVE CYLINDER TO ZERO		
071B	4C	01 09 05B5	841	NREAD4	MVC NIODAT(2,XR1),NXSNT	SET ADDR FOR TRANSIENT AREA		
0720	C0	87 0029	842	B	NREAD	READ RLD'S INTO TRANSIENT AREA		
0724	5E	01 09 2B	843	ALC	NIODAT(2,XR1),NPTEMP(,XR1)	UPDATE TO FIRST RLD		
0728	7C	00 2A	844	MVI	NPTEMP-1(,XR1),X'00'	SET HIGH ORDER BYTE TO ZERO		
072B	75	02 09	845	NMORLD	L NIODAT(,XR1),XR2	LOAD ADDR OF RLD		
072E	6C	00 2B 00	846	MVC	NPTEMP(1,XR1),0(,XR2)	MOVE RLD INTO WORK AREA		
0732	75	02 29	847	L	NPTXT(,XR1),XR2	LOAD ADDR OF CURRENT TXT DISPL.		
0735	78	FF 2B	848	TBN	NPTEMP(,XR1),X'FF'	END OF RLD'S ?		
0738	F2	10 36	849	JT	NLEND	JUMP IF YES		
073B	76	02 2B	850	A	NPTEMP(,XR1),XR2	UPDATE CURRENT @ TO NEXT TXT @		
073E	7D	80 2B	851	CLI	NPTEMP(,XR1),X'80'	RLD IS:		
0741	F2	84 0A	852	JH	NUPDAT	HIGH-IGNORE RLD, DON'T UPD TXT @		
0744	F2	81 04	853	JE	NUPTXT	EQUAL-UPDATE TXT @, NOT TXT		
0747	9E	01 00 27	854	NRLADD	ALC 0(2,XR2),NPORLF(,XR1)	LOW-ADD RELOC. FACTOR TO TXT		
074B	74	02 29	855	NUPTXT	ST NPTXT(,XR1),XR2	SAVE THE UPDATE TXT @		
074E	4E	01 09 05B4	856	NUPDAT	ALC NIODAT(2,XR1),NCX01	UPDATE PTR TO NEXT RLD		
0753	78	04 08	857	TBN	NIODAT-1(,XR1),X'04'	END OF RLD BUFFER ?		
0756	C0	90 072B	858	BF	NMORLD	NO-GO PROCESS NEXT RLD		
075A	7C	00 2B	859	MVI	NPTEMP(,XR1),X'00'	SET TEMP TO 0 FOR NEXT READ		
075D	4E	00 06 0016	860	ALC	NIOSB(1,XR1),NCX0C	ADD 3 TO SECTOR ADDRESS		
0762	78	60 06	861	TBN	NIOSB(,XR1),X'60'	TRACK OVERFLOW ?		
0765	F2	90 05	862	JF	NOOVER	NO-BYPASS TRACK INCREMENT		
0768	4E	01 06 0018	863	ALC	NIOSB(2,XR1),NCX20	YES-INCREMENT TRACK + 1		
076D	C0	87 071B	864	NOOVER	B NREAD4	REPEAT READ LOOP		
0771	5E	01 33 27	865	NLEND	ALC NPENT(2,XR1),NPORLF(,XR1)	ADD RELOCATION TO ENTRY POINT		
0775	75	02 23	866	NLEND2	L NPXR2(,XR1),XR2	POINT XR2 TO CALLERS BUFFER		
0778	78	10 14	867	TBN	NPRIB(,XR1),NFCHRQ	IS THIS A FETCH REQUEST ?		
077B	F2	90 07	868	JF	NLD010	JUMP IF NOT A FETCH REQUEST		
077E	5C	01 25 33	869	MVC	NPNSI(2,XR1),NPENT(,XR1)	CHANGE RETURN ADDR IF A FETCH		
0782	F2	87 0A	870	J	NLDONE	BYPASS CALLER PARM LIST UPDATE		



\$@SPV1 - SUPERVISOR FOR A DEDICATED SYSTEM

ERR	LOC	OBJECT CODE	ADDR	STMT	SOURCE	STATEMENT	VER	MOD	00	09/05/10	PAGE	23	
			0785	871	NLD010	EQU *							
	0785	7D 40 14		872		CLI NPRIB(,XR1),X'40'							
	0788	F2 81 04		873		JE NLDONE							
	078B	9C 09 09 35		874		MVC 9(10,XR2),NPPAR(,XR1)							
				875	*****								
				876	*	GENERAL EXIT ROUTINE						*	
				877	*****								
			078F	878	NLDONE	EQU *							
	078F	75 01 21		879	NEXIT	L NPXR1(,XR1),XR1							
	0792	35 10 049D		880		L NP1NSI,IAR							

\$@SPV1 - SUPERVISOR FOR A DEDICATED SYSTEM

ERR LOC OBJECT CODE ADDR STMT SOURCE STATEMENT VER 15, MOD 00 09/05/10 PAGE 24

882 \* \$SCA EXP-Y  
884+ PRINT ON

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00 09/05/10 PAGE 25
		886+*			-----	
		887+*			SYSTEM COMMUNICATION AREA EQUATES.	
		888+*			-----	
		889+*				
	0001	891+NCPL1	EQU	1	ADDR OF PGM LVL 1 COMM AREA.	
	0003	892+NCPL2	EQU	NCPL1+2	ADDR OF PGM LVL 2 COMM AREA.	
	0005	893+NCXTAB	EQU	NCPL2+2	ADDR OF TRANSIENT SCHED. TABLE.	
	0006	894+NCFCTR	EQU	NCXTAB+1	FETCH TRACE ID:	
		895+*			X'80' - FETCH TRACE IS ACTIVE.	
	0006	896+NCSGEN	EQU	NCXTAB+1	SYSGEN BYTE:	
		897+*			X'80' - USED BY FETCH TRACE.	
		898+*			X'7C' - USED BY \$MAINT.	
		899+*			X'02' - SYSTEM MAINTENANCE.	
		900+*			X'01' - SYSTEM GENERATION.	
	0007	901+NCPRTZ	EQU	NCSGEN+1	PRINTER SIZE.	
	0008	902+NCLPSZ	EQU	NCPRTZ+1	LEFT TRACTOR PAGE SIZE.	
	0009	903+NCRPSZ	EQU	NCLPSZ+1	RIGHT TRACTOR PAGE SIZE.	
	000A	904+NCSYSL	EQU	NCRPSZ+1	SYSTEM LIST ID:	
		905+*			X'F1' - 5408, 5410 OR 5412.	
		906+*			X'F2' - 5404 OR 5406.	
	000D	907+NCSLOG	EQU	NCSYSL+3	SYSLOG INDICATOR: C/S/DEV INFO.	
		908+*			X'00' - PRINTER	
		909+*			X'A0' - PRIMARY	
		910+*			X'B0' - CRT	
		911+*			X'C0' - CONSOLE	
		912+*			X'E0' - SECONDARY	
	000F	913+NCSWRK	EQU	NCSLOG+2	C/S OF SCHEDULER WORK AREA.	
	0010	914+NCSYSQ	EQU	NCSWRK+1	Q BYTE FOR SYSTEM PACK:	
		915+*			X'A1' - R1	
		916+*			X'A9' - F1	
	0012	917+NCOLIB	EQU	NCSYSQ+2	C/S OF SYSTEM OBJECT LIBRARY.	
	0013	918+NCSCH1	EQU	NCOLIB+1	SCHEDULER SWITCHES:	
		919+*			X'80' - SYSTEM LOG ON.	
		920+*			X'40' - SYSTEM DATE RECEIVED.	
		921+*			X'20' - DUAL PROGRAMMING FEATURE	
		922+*			X'10' - SCHED INTERLOCK PL/1.	
		923+*			X'08' - SCHED INTERLOCK PL/2.	
		924+*			X'04' - DATE FORMAT	
		925+*			0 - MMDDYY.	
		926+*			1 - DDMMYY.	
		927+*			X'03' - DISK CONFIG:	
		928+*			00 - R1, F1.	
		929+*			01 - R1, F1, R2.	
		930+*			11 - R1, F1, R2, F2.	
	0014	932+NCSMV1	EQU	NCSCH1+1	D.M. / SCHED SWITCHES PL/1.	
		933+*			X'80' - IPL SUCCESSFUL.	
		934+*			X'08' - OFFLINE MVF ON R1.	
		935+*			X'04' - OFFLINE MVF ON R2.	
		936+*			X'02' - OTHER FILE ON R1.	
		937+*			X'01' - OTHER FILE ON R2.	
	0015	939+NCSMV2	EQU	NCSMV1+1	D.M. / SCHED SWITCHES PL/2.	
		940+*			X'10' - LOCAL DISPLAY ADAPTOR.	
		941+*			X'08' - OFFLINE MVF ON R1.	
		942+*			X'04' - OFFLINE MVF ON R2.	
		943+*			X'02' - OTHER FILE ON R1.	

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00 09/05/10 PAGE 26
			944+*		X'01' - OTHER FILE ON R2.	
	0016	946+NCSCHEQU		NCSMV2+1	SCHEDULER BYTE:	
			947+*		X'80' - PRINTER INTERLOCK BIT.	
			948+*		X'40' - ROLLIN NOW PROCESSING.	
			949+*		X'20' - RESERVED.	
			950+*		X'10' - INPUT FOR I-TYPE PGM.	
			951+*		X'08' - ROLLOUT REQUESTED.	
			952+*		X'04' - PARTITION STMT RECV'D.	
			953+*		X'02' - ROLLOUT NOW PROCESSING.	
			954+*		X'01' - INQUIRY PENDING.	
	0018	956+NCDSKQ	EQU	NCSCH+2	ADDR OF 5444 R1,F1 I/O QUEUE.	
	0019	957+NCSCHEQU		NCDSKQ+1	PARTITION VALUE FOR PGM LVL 2.	
	001A	958+NCMBSV	EQU	NCSCH2+1	MISCELLANEOUS BYTE:	
			959+*		X'80' - MODEL 6.	
			960+*		X'40' - CHAINED PROC PROCESSED.	
			961+*		X'20' - LIST RPG II.	
			962+*		X'10' - SUPP OCL ON CALL STMT.	
			963+*		X'08' - UNIT RECORD IND FOR INQ.	
			964+*		X'04' - CCP/DLOG.	
			965+*		X'02' - LDA SUPPORT FOR ICA.	
			966+*		X'01' - NOEJECT SPECIFIED.	
	001B	968+NCXTA	EQU	NCMBSV+1	MISCELLANEOUS BYTE:	
			969+*		X'80' - 5448 SUPPORTED.	
	002F	970+NCSTOR	EQU	NCXTA+20	TRANSIENT STORAGE AREA.	
	0031	971+NC@CIO	EQU	NCSTOR+2	ADDR OF KEYB INTERRUPT HANDLER.	
	0034	972+NCRCSS	EQU	NC@CIO+3	C/S OF ROLLOUT AREA.	
	0035	973+NCCONF	EQU	NCRCSS+1	3411 AND 3340 CONFIGURATION.	
			974+*		X'80' - HE HALT ISSUED BY DLOG.	
			975+*		X'40' - RESERVED FOR CCP.	
			976+*		X'20' - RESERVED FOR CCP.	
			977+*		X'10' - D1	
			978+*		X'18' - D1, D2	
			979+*		X'07' - T1, T2, T3, T4	
			980+*		X'06' - T1, T2, T3	
			981+*		X'05' - T1, T2	
			982+*		X'04' - T1	
	0037	984+NCADDC	EQU	NCCONF+2	ADDR OF ADDITIONAL SYSTEM	
			985+*		COMMUNICATION AREA (ASCA).	
	0039	986+NCARSV	EQU	NCADDC+2	ADDR OF HALT/SYSLOG SYSIN AREA.	
			987+*	DISK SYSTEM ONLY.		
	003A	988+NCSCHEQU		NCARSV+1	SCHEDULER BYTE.	
	003B	989+NCSCHEQU		NCSCH3+1	SCHEDULER BYTE.	
	003C	990+NCSMV3	EQU	NCSCH4+1	D.M. MULTI VOLUME FILE.	
			991+*		PGM LVL 1:	
			992+*		X'80' - MVF ON D1.	
			993+*		X'40' - MVF ON D2	
			994+*		X'20' - OTHER TYPE OF FILE ON D1	
			995+*		X'10' - OTHER TYPE OF FILE ON D2	
			996+*		PGM LVL 2:	
			997+*		X'08' - MVF ON D1.	
			998+*		X'04' - MVF ON D2	
			999+*		X'02' - OTHER TYPE OF FILE ON D1	
			1000+*		X'01' - OTHER TYPE OF FILE ON D2	

\$@SPV1 - SUPERVISOR FOR A DEDICATED SYSTEM

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00 09/05/10 PAGE 27
		003E	1002+NCDISK5	EQU	NCSMV3+2	ADDR OF 5445 OR 5448 I/O QUEUE.
		0040	1003+NCTAPQ	EQU	NCDISK5+2	ADDR OF 3411 I/O QUEUE.
			1004+*		NEXT 9 BYTE ARE USED ONLY BY A DPF SYSTEM.	
		0042	1005+NCSXR1	EQU	NCTAPQ+2	XR1 SAVE AREA'S
		0044	1006+NCSXR2	EQU	NCSXR1+2	XR2 FOR INTERRUPT
		0046	1007+NCSPSR	EQU	NCSXR2+2	PSR LEVEL 0.
		0047	1008+NCSNS	EQU	NCSPSR+1	READER SELECT SWITCH SENSE INFO.
			1009+*		X'80' - LEVEL 1.	
			1010+*		X'40' - CANCEL.	
			1011+*		X'20' - MFCU.	
			1012+*		X'10' - AUXILIARY READER.	
			1013+*		X'08' - PRINTER KEYBOARD.	
		0049	1015+NCHALT	EQU	NCSNS+2	ADDR OF RESIDENT NPHALT ROUTINE.
		0049	1016+NCEDPF	EQU	NCHALT	END OF DPF SYSTEM COMM AREA.
			1017+		PRINT ON	
			1019 *		\$PLCA EXP-Y	
			1021+		PRINT ON	
			1023+*		-----	
			1024+*		PROGRAM LEVEL COMMUNICATION AREA EQUATES.	
			1025+*		-----	
			1026+*			
		0000	1028+NPIOB	EQU	0	PROGRAM LEVEL IOB.
		0001	1029+NPCHAN	EQU	NPIOB+1	IOS CHAIN POINTER.
		0002	1030+NIOCMP	EQU	NPIOB+2	COMPLETION CODE.
		0003	1031+NIOQB	EQU	NPIOB+3	IOB Q BYTE.
		0004	1032+NIO RB	EQU	NPIOB+4	IOB R BYTE.
		0005	1033+NIOCB	EQU	NPIOB+5	IOB CYLINDER BYTE.
		0006	1034+NIO SB	EQU	NPIOB+6	IOB SECTOR BYTE.
		0007	1035+NIONB	EQU	NPIOB+7	NUM OF SECTORS TO BE READ-1.
		0009	1036+NIODAT	EQU	NPIOB+9	DATA BUFFER ADDRESS.
		000B	1037+NIOSNS	EQU	NPIOB+11	SENSE BYTES.
		000C	1038+NIOERR	EQU	NPIOB+12	IOS ERROR RETRY COUNTER.
		000D	1039+NIOFLG	EQU	NPIOB+13	FLAG BITS.
		0011	1040+NIOOBR	EQU	NPIOB+17	ARR, XR2 SAVE AREA.
		0013	1042+NPDTF@	EQU	NIOOBR+2	FIRST DTF ADDRESS.
		0014	1043+NPRIB	EQU	NPDTF@+1	PROGRAM RIB.
		0016	1044+NPBEG	EQU	NPRIB+2	PROGRAM LEVEL BEGINNING ADDRESS.
		0018	1045+NPEND	EQU	NPBEG+2	PROGRAM LEVEL END ADDRESS.
		0019	1046+NPQ	EQU	NPEND+1	PROGRAM LEVEL Q BYTE.
		001B	1047+NPRLF	EQU	NPQ+2	PROGRAM RELOCATION FACTOR.
		001D	1048+NPCYL	EQU	NPRLF+2	C/S OF FIRST LOAD.
		001F	1049+NPOLIB	EQU	NPCYL+2	C/S OF PROGRAM OBJECT LIBRARY.
		0021	1050+NPXR1	EQU	NPOLIB+2	XR1 SAVE AREA.
		0023	1051+NPXR2	EQU	NPXR1+2	XR2 SAVE AREA.
		0025	1052+NPNSI	EQU	NPXR2+2	RETURN ADDRESS.
		0027	1053+NPORLF	EQU	NPNSI+2	OVERLAY RELOCATION FACTOR.
		0029	1054+NPTXT	EQU	NPORLF+2	OVERLAY TEXT ADDRESS.
		002B	1055+NPTEMP	EQU	NPTXT+2	TEMPORARY STORAGE FOR LOADER.
		002D	1057+NPCS	EQU	NPTEMP+2	C/S ADDRESS OF MODULE.
		002E	1058+NP#S	EQU	NPCS+1	NUMBER OF SECTORS TO READ.
		0030	1059+NPLNK	EQU	NP#S+2	LINK EDITED ADDRESS.

\$@SPV1 - SUPERVISOR FOR A DEDICATED SYSTEM

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00 09/05/10 PAGE 28
		0031	1060+NPRLD	EQU	NPLNK+1	DISPLACEMENT OF RLD.
		0033	1061+NPENT	EQU	NPRLD+2	ADDRESS OF ENTRY POINT.
		0035	1062+NPLOD	EQU	NPENT+2	LOAD ADDRESS.
		0035	1063+NPPAR	EQU	NPLOD	*
		0036	1065+NPEOJ	EQU	NPLOD+1	END OF JOB ID.
			1066+*			X'80' - RJE ACTIVE.
			1067+*			X'40' - SUCCESSFUL IPL.
			1068+*			X'20' - LIBRARY MAINT. BIT.
			1069+*			X'10' - CANCEL PENDING.
			1070+*			X'0E' - RPG SYSLOG BITS FOR EOJ.
			1071+*			X'01' - DO NOT CLOSE DTF AT EOJ.
		0037	1073+NPUPSI	EQU	NPEOJ+1	UPSI SWITCH BITS.
		003D	1074+NPNAME	EQU	NPUPSI+6	PROGRAM NAME.
		003E	1075+NPRELS	EQU	NPNAME+1	RELEASE LEVEL.
		0044	1076+NPPDATE	EQU	NPRELS+6	PROGRAM DATE.

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00 09/05/10 PAGE 29
		0048	1078+	NPSYSI EQU	NPCDATE+4	SYSIN INDICATOR.
			1079+*			C/S/# OF SECTORS/DEV INFO:
			1080+*			X'F0' - MCFU/1.
			1081+*			X'F8' - MCFU/2.
			1082+*			X'F4' - 1442.
			1083+*			X'A0' - KEYBOARD 5406.
			1084+*			X'B0' - KEYBOARD/CRT DISP.
			1085+*			X'C0' - CONSOLE.
			1086+*			X'D0' - DATA96.
			1087+*			X'40' - 3741.
		0049	1089+	NPSCH1 EQU	NPSYSI+1	READER/INTERPRETER SWITCHES.
			1090+*			X'80' - // DATE RECEIVED (INTRA)
			1091+*			X'40' - // COMPILE RECEIVED.
			1092+*			X'20' - // SWITCH RECEIVED.
			1093+*			X'10' - PROCEDURE.
			1094+*			X'08' - OVERRIDE REQUEST.
			1095+*			X'04' - INTRA MODE.
			1096+*			X'02' - INTER MODE.
			1097+*			X'01' - IPL MODE.
		004A	1099+	NPSCH2 EQU	NPSCH1+1	SCHEDULER SWITCHES.
			1100+*			X'80' - CONTINUATION OF OCL STMT
			1101+*			X'40' - UTIL CNTL CARDS IN JSWA.
			1102+*			X'20' - /# READ FROM SYSIN.
			1103+*			X'10' - A FILE STMT RECEIVED.
			1104+*			X'08' - EOJ HALT INDICATOR.
			1105+*			X'04' - FLUSH.
			1106+*			X'02' - IMMEDIATE CANCEL.
			1107+*			X'01' - CONTROLLED CANCEL.
		004B	1109+	NPSCH3 EQU	NPSCH2+1	SCHEDULER SWITCHES.
			1110+*			X'80' - TAG SORT REQUIRED (5444)
			1111+*			X'40' - ALLOCATE TRANSIENT.
			1112+*			X'20' - SOURCE REQUIRED.
			1113+*			X'10' - LOG STATUS:
			1114+*			OFF - LOG ON SPEC.
			1115+*			ON - LOG OFF SPEC.
			1116+*			X'08' - MVF FILE ALLOCATED.
			1117+*			X'04' - ADDITIONAL PROC STMT.
			1118+*			X'02' - 1ST 'LOAD-RUN' JOB READ.
			1119+*			X'01' - PROGRAM LEVEL:
			1120+*			0 - LEVEL 1.
			1121+*			1 - LEVEL 2.
		004B	1123+	NPLEVL EQU	NPSCH3	PROGRAM LEVEL.
		004C	1124+	NPOBJQ EQU	NPSCH3+1	OBJECT DECK OUTPUT Q CODE.
		004D	1126+	NPBPSD EQU	NPOBJQ+1	AVAIL STATUS OF SYSIN DEVICES.
			1127+*			X'80' - MFCU.
			1128+*			X'40' - CONSOLE I/O.
			1129+*			X'20' - 1442.
			1130+*			X'10' - AUX ON RDR SELECT SWITCH
			1131+*			X'08' - CONSOLE ON RDR SEL SWITC
			1132+*			X'04' - 3741.
			1133+*			X'02' - STORAGE ROLLED OUT BY
			1134+*			ALLOCATE.
			1135+*			X'01' - NESTED PROCEDURE.

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00 09/05/10 PAGE 30
		004E	1137+	NPSCH4 EQU	NPBPSD+1	SCHEDULER SWITCHES.
			1138+*			X'80' - THIS PROGRAM LEVEL ACTIE
			1139+*			X'40' - // IMAGE RECEIVED.
			1140+*			X'20' - // FORMS RECEIVED.
			1141+*			X'10' - F1 NEEDED FOR ALLOC.
			1142+*			X'08' - R1 NEEDED FOR ALLOC.
			1143+*			X'04' - F2 NEEDED FOR ALLOC.
			1144+*			X'02' - R2 NEEDED FOR ALLOC.
			1145+*			X'01' - INITIAL EOJ PERFORMED.
		004F	1147+	NPSCH8 EQU	NPSCH4+1	SCHEDULER SWITCHES.
			1148+*			X'80' - // READER RECEIVED.
			1149+*			X'40' - CONSOLE IN USE BY D.M.
			1150+*			X'20' - VALID // FILE RECEIVED.
			1151+*			X'10' - SHARED I/O PROGRAM.
			1152+*			X'08' - 'I' OR 'B' TYPE PROGRAM
			1153+*			X'04' - ALLOCATE UNSUCCESSFUL.
			1154+*			X'02' - MAX TRACK REQUEST MET.
			1155+*			X'01' - MIN TRACK REQUEST MET.
		0051	1157+	NPSCH6 EQU	NPSCH8+2	5406 CURRENT HALT INSTRUCTION.
		0052	1158+	NPSCHD EQU	NPSCH6+1	X'80' - NO STOP MODE
		0053	1159+	NPSCHE EQU	NPSCHD+1	RESERVED.
		0055	1160+	NPWKB EQU	NPSCHE+2	DISK IOS WORK AREA ADDRESS.
		0056	1161+	NPSCH9 EQU	NPWKB+1	DISK LOG UNIT FOR EOJ/DEALLOC.
			1162+*			X'80' - F2 THESE
			1163+*			X'40' - R2 UNITS ARE
			1164+*			X'20' - F1 LOGGED ONLY
			1165+*			X'10' - R1 ONCE.
			1166+*			X'08' - F2 ALL THESE
			1167+*			X'04' - R2 UNITS ARE
			1168+*			X'02' - F1 LOGGED AT
			1169+*			X'01' - R1 END OF JOB.
		0057	1171+	NPSCHA EQU	NPSCH9+1	SCHEDULER SWITCHES.
			1172+*			X'20' - CALL TAPE DLOG AT EOJ.
			1173+*			X'08' - ACTIVE FILES ON T1.
			1174+*			X'04' - ACTIVE FILES ON T2.
			1175+*			X'02' - ACTIVE FILES ON T3.
			1176+*			X'01' - ACTIVE FILES ON T4.
		0058	1178+	NPSCHB EQU	NPSCHA+1	SCHEDULER SWITCHES.
			1179+*			X'80' - CHECKPOINT PGM EXECUTING
			1180+*			X'40' - 5445 FILE CARD PRESENT.
			1181+*			X'20' - NEW FILE ALLOCATED ON D1
			1182+*			X'10' - NEW FILE ALLOCATED ON D2
			1183+*			X'08' - DEFERED ALLOCATE REQUEST
			1184+*			X'04' - 2ND LEVEL HALT REQUIRED.
			1185+*			X'02' - TAG SORT REQUIRED (5445)
			1186+*			X'01' - RECOURCE ALLOCATION.
		005A	1188+	NPHLT2 EQU	NPSCHB+2	SAVE AREA FOR SECOND LEVEL HALT.
		005C	1189+	NPEOJ@ EQU	NPHLT2+2	EOJ RETURN ADDRESS.
		005D	1190+	NPUNCH EQU	NPEOJ@+1	SYSPUNCH ID.
		005D	1191+	NPEDED EQU	NPUNCH	END OF DEDICATED PROGRAM LEVEL.

1193+\* THE FOLLOWING EQUATES ARE ONLY FOR A DPF SYSTEM.



\$@SPV1 - SUPERVISOR FOR A DEDICATED SYSTEM

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00 09/05/10 PAGE 31			
			1194+*						
005F		1195+	NPHXR2	EQU	NPUNCH+2	PARAM SAVE AREA.			
0067		1196+	NPHTRN	EQU	NPHXR2+8	TRANSIENT STORAGE AREA.			
008B		1197+	NPHQSV	EQU	NPHTRN+36	TRANSIENT Q SAVE AREA.			
			008C		1199+	NPUTIL	EQU	NPHQSV+1	UTILITY INTERLOCK BYTE.
			1200+*			X'80', X'08' - F2 ALLOCATED			
			1201+*			X'40', X'04' - R2 OR			
			1202+***		UTILITIES ARE: \$ALT, \$BUILD,	X'20', X'02' - F1 IN			
			1203+*		\$COPY, \$INIT.	X'10', X'01' - R1 USE.			
			1204+*						
008E		1205+	NPSAVE	EQU	NPUTIL+2	SAVE AREA FOR IAR.			
0096		1206+	NPSUJV	EQU	NPSAVE+8	SAVE AREA ON A 'JU' HALT.			
			0097		1208+	NPUTL1	EQU	NPSUJV+1	TAPE INTERLOCK BYTE.
			1209+*			X'80' - T1 IN USE.			
			1210+*			X'40' - T2 IN USE.			
			1211+*			X'20' - T3 IN USE.			
			1212+*			X'10' - T4 IN USE.			
			0098		1214+	NPUTL2	EQU	NPUTL1+1	5445 UTILITY INTERLOCK BYTE.
			1215+*			X'80' - LIBRARY USAGE INTERLOCK.			
			1216+*			X'40' - 3881 INTERLOCK.			
			1217+*			X'20' - D2 ALLOCATED.			
			1218+*			X'10' - D1 ALLOCATED.			
			1219+*			X'02' - D2 IN USE.			
			1220+*			X'01' - D1 IN USE.			
			0099		1222+	NPSCH5	EQU	NPUTL2+1	READER SELECT SWITCHES.
			1223+*			X'80' - PROGRAM LEVEL 1.			
			1224+*			X'20' - MFCU POSITION.			
			1225+*			X'10' - AUX POSITION.			
			1226+*			X'08' - PTR/KBD POSITION.			
			1227+*			X'04' - TAPE FILE CARDS REC'D.			
			009B		1229+	NPSTBY	EQU	NPSCH5+2	ADDR OF LEVEL STANDBY ROUTINE.
			009C		1230+	NPSCH7	EQU	NPSTBY+1	SAVE OF LAST SYSIN ASSIGNMENT.
			009D		1231+	NPSCHC	EQU	NPSCH7+1	PGM LEVEL EXPANSION VALUE.
			009E		1232+	NPHALT	EQU	NPSCHC+1	PGM LEVEL HPL.
			009F		1233+	NPHLTQ	EQU	NPHALT+1	Q CODE FOR HPL.
			00A0		1234+	NPHLTR	EQU	NPHLTQ+1	R CODE FOR HPL.
			00A4		1235+	NPHBCH	EQU	NPHLTR+4	BRANCH FOR RESIDENT HALT.
			00A4		1236+	NPEDPF	EQU	NPHBCH	END OF DPF PROGRAM LEVEL.
			1237+			PRINT	ON		

1239 \*\*\*\*\*  
 1240 \* THESE ARE THE EQUATES FOR ENTRY POINTS GENERAL AND IOS \*  
 1241 \*\*\*\*\*

0004 1242 NCENR EQU 4 GENERAL ENTRY  
 0008 1243 NCEIOS EQU 8 IOS  
 000C 1244 NCEIOW EQU 12 I/O WAIT  
 0011 1245 NCSYS@ EQU 17 SYSTEM COMMUNICATION AREA.  
 0400 1246 NCHIMG EQU X'0400' MODEL 10 - PRINT CHAIN IMAGE  
 1247 \* MODEL 6 - KEYBOARD IMAGE  
 001A 1248 NCMSVA EQU X'001A' MODEL 10 - MFCU SIO HISTORY TAB  
 1249 \* MODEL 6 - SYSTEM PRINT AREA

1251 \* BIT  
 1252 \* 0 1 2 3 4 5 6 7  
 1253 \*-----  
 1254 \* USE SYSTEM IOB | 0 | 1 | 1 | | | | |  
 1255 \*-----  
 1256 \* USE PROGRAM IOB | 0 | 1 | 0 | | | | |  
 1257 \*-----  
 1258 \* FETCH | 0 | 1 | | 1 | 0 | 0 | |  
 1259 \*-----  
 1260 \* FETCH TO ADDRESS | 0 | 1 | | 1 | 1 | 0 | |  
 1261 \*-----  
 1262 \* SYSTEM FETCH | 0 | 1 | | 1 | 0 | 1 | |  
 1263 \*-----  
 1264 \* LOAD | 0 | 1 | | 0 | 1 | 0 | |  
 1265 \*-----  
 1266 \* RPG LOAD | 0 | 1 | | 0 | 0 | 0 | |  
 1267 \*-----  
 1268 \* WITH FIND | 0 | 1 | | | | | | 1 |  
 1269 \*-----  
 1270 \* WITH OUT FIND | 0 | 1 | | | | | | 0 |  
 1271 \*-----

1273 \*  
 1274 \* THESE EQU'S CAN BE USED TO CREATE RIB'S  
 1275 \*  
 0060 1276 SYSIOB EQU X'60' USE SYSTEM IOB  
 0040 1277 PGMIOB EQU X'40' USE PROGRAM IOB  
 0010 1278 FETCH EQU X'10' FETCH  
 0018 1279 FETCH@ EQU X'18' FETCH TO AN ADDR  
 0014 1280 SFETCH EQU X'14' SYSTEM FETCH  
 0008 1281 LOAD EQU X'08'  
 0000 1282 RPLGLOD EQU X'00'  
 0001 1283 FIND EQU X'01'  
 1284 \*  
 1285 \* ABOVE EQU'S CAN BE ADDED TOGETHER TO GET THE RIGHT RIB.  
 1286 \*  
 1287 \* EXAMPLE: RIB1 DC AL1(FETCH+PGMIOB+FIND)  
 1288 \* RIB2 DC AL1(LOAD+SYSIOB)  
 1289 \* RIB3 DC AL1(FETCH@+SYSIOB+FIND)

0001 1291 CCPRIB EQU X'01' CCP RIB  
 0010 1292 HD EQU X'10' HALT -  
 0063 1293 HJ EQU X'63' HALT J

1294 \*\*\*\*\*

\$@SPV1 - SUPERVISOR FOR A DEDICATED SYSTEM

```
ERR LOC  OBJECT CODE      ADDR STMT SOURCE STATEMENT          VER 15, MOD 00  09/05/10  PAGE  33
      1295 *          DISK QUEUE EQUATES *
      1296 *****
      1297 *          OFFSETS FOR 5444 QUEUE
0001 1298 QFIRST EQU   1          @ FIRST IOB ON QUEUE
0003 1299 QLAST  EQU   3          @ LAST IOB ON QUEUE
0005 1300 QSELF  EQU   5          @ THIS QUEUE
0006 1301 QFBYTE EQU   6          F BYTE OF DISK OPERATION
0007 1302 QCBYTE EQU   7          C BYTE OF DISK OPERATION
0008 1303 QSBYTE EQU   8          S BYTE OF DISK OPERATION
0009 1304 QNBYTE EQU   9          N BYTE OF DISK OPERATION
000B 1305 QSENSE EQU  11          SENSE INFO SAVE AREA
000D 1306 NXTQUE EQU  13          @ NEXT SPINDLE'S QUEUE
000E 1307 LASTSK EQU  14          SAVE FOR LAST LOGICAL CYL #
0010 1308 PHYSCS EQU  16          PHYSICAL C/S SAVE AREA
0012 1309 SAVEOP EQU  18          SAVE FOR Q/R DURING VERIFY
0013 1310 QSTATS EQU  19          DRIVE STATUS
0796 1311 JUHALT EQU   *          MUST BE IN BECAUSE DISK IOS HAS
      1312 *
      1313 * END CHANGE ACTIVITY - @$SPV1
FFFF 1314          END
```

TOTAL STATEMENTS IN ERROR IN THIS ASSEMBLY = 0

OL105 I THE CODE LENGTH OF @\$SPV1 IS 1942 DECIMAL.  
R@\$SPV1  
EO6EDP 0 3

OL103 I TOTAL NUMBER OF LIBRARY SECTORS REQUIRED IS 12  
NAME-@\$SPV1,PACK-R1R1R1,UNIT-R1,RETAIN-P,LIBRARY-R,CATEGORY-000  
\*  
// READER CONSOLE