


```

NN      NN      CCCCCCCC  PPPPPPPP  MM      MM      AAAAAA  IIIIII  NN      NN
NN      NN      CCCCCCCC  PPPPPPPP  MM      MM      AAAAAA  IIIIII  NN      NN
NN      NN      CC          PP          PP  MMMM  MMMM  AA      AA  II      II  NN      NN
NN      NN      CC          PP          PP  MMMM  MMMM  AA      AA  II      II  NN      NN
NNNN    NN      CC          PP          PP  MM  MM  MM  AA      AA  II      II  NNNN    NN
NNNN    NN      CC          PP          PP  MM  MM  MM  AA      AA  II      II  NNNN    NN
NN  NN  NN      CC          PPPPPPPP  MM      MM  AA      AA  II      II  NN  NN  NN
NN  NN  NN      CC          PPPPPPPP  MM      MM  AA      AA  II      II  NN  NN  NN
NN      NNNN    CC          PP          MM      MM  AAAAAAAAAA  II      II  NN      NNNN
NN      NNNN    CC          PP          MM      MM  AAAAAAAAAA  II      II  NN      NNNN
NN      NN      CC          PP          MM      MM  AA      AA  II      II  NN      NN
NN      NN      CC          PP          MM      MM  AA      AA  II      II  NN      NN
NN      NN      CCCCCCCC  PP          MM      MM  AA      AA  IIIIII  NN      NN
NN      NN      CCCCCCCC  PP          MM      MM  AA      AA  IIIIII  NN      NN

```

```

LL      IIIIII  SSSSSSSS
LL      IIIIII  SSSSSSSS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SSSSSS
LL      II      SSSSSS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SS
LLLLLLLLLLL  IIIIII  SSSSSSSS
LLLLLLLLLLL  IIIIII  SSSSSSSS

```

```

1 0001 0 %TITLE 'Main Entry'
2 0002 0 MODULE NCPMAIN (IDENT = 'V04-000', MAIN = NCP$MAIN,
3 0003 0 ADDRESSING_MODE(EXTERNAL=GENERAL),
4 0004 0 ADDRESSING_MODE(NONEXTERNAL=GENERAL)) =
5 0005 1 BEGIN
6 0006 1
7 0007 1
8 0008 1 *****
9 0009 1 *
10 0010 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY *
11 0011 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS. *
12 0012 1 * ALL RIGHTS RESERVED. *
13 0013 1 *
14 0014 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED *
15 0015 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE *
16 0016 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER *
17 0017 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY *
18 0018 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY *
19 0019 1 * TRANSFERRED. *
20 0020 1 *
21 0021 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE *
22 0022 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT *
23 0023 1 * CORPORATION. *
24 0024 1 *
25 0025 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS *
26 0026 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL. *
27 0027 1 *
28 0028 1 *
29 0029 1 *****
30 0030 1
31 0031 1
32 0032 1 **
33 0033 1 FACILITY: NCP Network Control Program
34 0034 1
35 0035 1 ABSTRACT:
36 0036 1
37 0037 1 NCP main entry and command line acceptance
38 0038 1
39 0039 1 ENVIRONMENT: VAX/VMS Operating System
40 0040 1
41 0041 1 AUTHOR: Darrell Duffy , CREATION DATE: 17-August-1979
42 0042 1
43 0043 1 MODIFIED BY:
44 0044 1
45 0045 1 V03-003 PRD0057 Paul R. DeStefano 05-Feb-1984
46 0046 1 No longer need to save CLI verify state.
47 0047 1
48 0048 1 V03-002 RPG0002 Bob Grosso 03-Aug-1982
49 0049 1 Stop NCP from automatically starting the NETACT.
50 0050 1
51 0051 1 V03-001 RPG47232 Bob Grosso 21-Jul-1982
52 0052 1 Increase size of command input buffer, sysin_buf_siz.
53 0053 1
54 0054 1 V002 TMH0002 Tim Halvorsen 13-Jul-1981
55 0055 1 Change all non-local references to general addressing.
56 0056 1
57 0057 1 V001 TMH0001 Tim Halvorsen 13-Jun-1981

```

NCPMAIN
V04-000

Main Entry

I 14
15-Sep-1984 23:46:03
14-Sep-1984 12:48:14

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[NCP.SRC]NCPMAIN.B32;1 Page 2 (1)

:	58	0058	1	:
:	59	0059	1	:
:	60	0060	1	:
:	61	0061	1	--

If invoked via DCL foreign command and there is
text following the "verb", then execute the text
as a single NCP command and exit.

```

: 63 0062 1 %SBTTL 'Definitions'
: 64 0063 1
: 65 0064 1
: 66 0065 1 : TABLE OF CONTENTS:
: 67 0066 1
: 68 0067 1
: 69 0068 1 FORWARD ROUTINE
: 70 0069 1     NCP$COND_HDLR,
: 71 0070 1     NCP$READ_CMD
: 72 0071 1     ;
: 73 0072 1
: 74 0073 1
: 75 0074 1 : INCLUDE FILES:
: 76 0075 1
: 77 0076 1
: 78 0077 1 LIBRARY 'OBJS:NCPLIBRY.L32';
: 79 0078 1 LIBRARY 'SYSSL:BRARY:STARLET.L32';
: 80 0079 1 LIBRARY 'SYSSL:BRARY:TPAMAC.L32';
: 81 0080 1
: 82 0081 1
: 83 0082 1 : MACROS:
: 84 0083 1
: 85 0084 1
: 86 0085 1
: 87 0086 1 : EQUATED SYMBOLS:
: 88 0087 1
: 89 0088 1
: 90 0089 1
: 91 0090 1
: 92 0091 1 : OWN STORAGE:
: 93 0092 1
: 94 0093 1
: 95 0094 1
: 96 0095 1
: 97 0096 1 : EXTERNAL REFERENCES:
: 98 0097 1
: 99 0098 1
: 100 0099 1 EXTERNAL ROUTINE
: 101 0100 1     NCP$INIT_TRMIO,           ! Initialize terminal I/O
: 102 0101 1     NCP$STARTNETACP : NOVALUE, ! Start the network if not
: 103 0102 1     NCP$READ_LINE,         ! Read a line from SYSS$INPUT
: 104 0103 1     NCP$WRITE_LINE,        ! Write a line to SYSS$OUTPUT
: 105 0104 1     NCP$PARSE_CMD,         ! Parse a command line
: 106 0105 1     NCP$UNTELC,           ! Remove the tell executor
: 107 0106 1     ACT$VRB_CL$EXEC,       ! Remove the current executor and
: 108 0107 1     ! Establish the local node as exec
: 109 0108 1     LIB$GET_FOREIGN: ADDRESSING_MODE(GENERAL); ! Get foreign command line

```

```

: 111 0109 1 %SBTTL 'NCP$MAIN Main Entry'
: 112 0110 1 ROUTINE NCP$MAIN =
: 113 0111 1
: 114 0112 1 !++
: 115 0113 1 FUNCTIONAL DESCRIPTION:
: 116 0114 1
: 117 0115 1 Main entry for NCP
: 118 0116 1
: 119 0117 1 FORMAL PARAMETERS:
: 120 0118 1
: 121 0119 1 NONE
: 122 0120 1
: 123 0121 1 IMPLICIT INPUTS:
: 124 0122 1
: 125 0123 1 NONE
: 126 0124 1
: 127 0125 1 ROUTINE VALUE:
: 128 0126 1 COMPLETION CODES:
: 129 0127 1
: 130 0128 1 Exit code for NCP
: 131 0129 1
: 132 0130 1 SIDE EFFECTS:
: 133 0131 1
: 134 0132 1 LOTS
: 135 0133 1
: 136 0134 1 --
: 137 0135 1
: 138 0136 2 BEGIN
: 139 0137 2
: 140 0138 2 BUILTIN
: 141 0139 2 FP, ! Frame pointer
: 142 0140 2 AP, ! Argument pointer
: 143 0141 2 ;
: 144 0142 2
: 145 0143 2 LITERAL
: 146 0144 2 SYSIN_BUF_SIZ = 1024 ! Size of input buffer
: 147 0145 2 ;
: 148 0146 2
: 149 0147 2 OWN
: 150 0148 2 CMD_DSC : VECTOR [2], ! Command line buffer descriptor
: 151 0149 2 SYSIN_BUF : ! Sys$input buffer
: 152 0150 2 VECTOR [SYSIN_BUF_SIZ, BYTE]
: 153 0151 2 ;
: 154 0152 2
: 155 0153 2 BIND
: 156 0154 2 NCP_PMT_DSC = ! Descriptor of NCP prompt
: 157 0155 2 -ASCID ('NCP>'),
: 158 0156 2 CLIBLK = AP: REF BBLOCK ! Arg block from calling cli (DCL)
: 159 0157 2 ;
: 160 0158 2
: 161 0159 2 GLOBAL BIND
: 162 0160 2 NCPS_CMDBUF_DSC = ! Descriptor for sysinbuf
: 163 0161 2 UPLIT (SYSIN_BUF_SIZ, SYSIN_BUF)
: 164 0162 2 ;
: 165 0163 2
: 166 0164 2 LOCAL
: 167 0165 2 RTN_DSC : VECTOR [2] ! Returned descriptor

```

```

: 168 0166 2
: 169 0167 2
: 170 0168 2
: 171 0169 2
: 172 0170 2
: 173 0171 2
: 174 0172 2
: 175 0173 2
: 176 0174 2
: 177 0175 2
: 178 0176 2
: 179 0177 2
: 180 0178 2
: 181 0179 2
: 182 0180 2
: 183 0181 2
: 184 0182 2
: 185 0183 2
: 186 0184 2
: 187 0185 2
: 188 0186 2
: 189 0187 2
: 190 0188 2
: 191 0189 2
: 192 0190 2
: 193 0191 2
: 194 0192 2
: 195 0193 2
: 196 0194 2
: 197 0195 2
: 198 0196 2
: 199 0197 2
: 200 0198 2
: 201 0199 2
: 202 0200 2
: 203 0201 2
: 204 0202 2
: 205 0203 2
: 206 0204 2
: 207 0205 2
: 208 0206 2
: 209 0207 2
: 210 0208 2
: 211 0209 2
: 212 0210 2
: 213 0211 2
: 214 0212 2
: 215 0213 2
: 216 0214 2
: 217 0215 2
: 218 0216 2
: 219 0217 2
: 220 0218 2
: 221 0219 2
: 222 0220 2
: 223 0221 2
: 224 0222 2

;
EXTERNAL
  NCP$G_BEGIN_ZERO,      ! Parameter data blocks to clear
  NCP$G_END_ZERO,        ! End of the blocks
  NCP$G_STATE_TBL,      ! State table for parsing
  NCP$G_KEY_TBL
;

NCP$INIT_TRMIO ();      ! Initialize terminal I/O

! NCP$STARTNETACP ();    ! Start the network, if not started
.FP = NCP$COND_HDLR;    ! Enable the handler

ACT$VRB_CLEXEC ();      ! Establish link to local node

IF LIB$GET_FOREIGN(NCP$ CMDBUF_DSC,0,CMD_DSC) ! If a foreign command,
AND .CMD_DSC [0] NEQ 0 ! and there is a command line,
THEN
  BEGIN
  CMD_DSC [1] = SYSIN_BUF;      ! Construct descriptor of line
  CH$FILL(0,NCP$G_END_ZERO-NCP$G_BEGIN_ZERO, ! Zero parameters
  NCP$G_BEGIN_ZERO);
  NCP$PARSE_CMD(CMD_DSC,NCP$G_STATE_TBL,NCP$G_KEY_TBL,RTN_DSC);
  RETURN TRUE;
  END;

WHILE
  BEGIN
  LOCAL
  READ_STATUS;

  READ_STATUS = NCP$READ_CMD      ! Read line into buffer
  ?
  NCP$ CMDBUF_DSC,                ! Buffer descriptor
  NCP_PMT_DSC,                    ! Prompt descriptor
  CMD_DSC                          ! Return dsc of line
  );
  IF NOT .READ_STATUS
  THEN
  BEGIN
  IF (.READ_STATUS NEQ RMSS_EOF)
  THEN
  RETURN .READ_STATUS
  ELSE FALSE                      ! EOF terminates without error
  END
  ELSE TRUE
  END

DO                                ! Do so until error returned
  BEGIN
  CH$FILL                          ! Zero the parameter block area
  (
  0,                                ! Fill character
  NCP$G_END_ZERO - NCP$G_BEGIN_ZERO, ! Count
  NCP$G_BEGIN_ZERO                 ! Beginning address
  );

```

```

: 225 0223 3
: 226 0224 3
: 227 0225 3
: 228 0226 3
: 229 0227 3
: 230 0228 3
: 231 0229 3
: 232 0230 3
: 233 0231 3
: 234 0232 3
: 235 0233 3
: 236 0234 3
: 237 0235 3
: 238 0236 3
: 239 0237 1

```

```

NCP$PARSE_CMD      ! Parse the command
(
  CMD_DSC
  NCP$G_STATE_TBL,
  NCP$G_KEY_TBL,
  RTN_DSC
);

NCP$UNTELL ()      ! Remove the tell executor

END

RETURN TRUE      ! Return with success
END;

```

```

.TITLE NCPMAIN Main Entry
.IDENT \V04-000\

.PSECT $PLITS,NOWRT,NOEXE,2

3E 50 43 4E 0000 P.AAB: .ASCII \NCP>\
      00000004 00004 P.AAA: .LONG 4
      00000000' 00008 .ADDRESS P.AAB
      00000400 0000C P.AAC: .LONG 1024
      00000000' 00010 .ADDRESS SYSIN_BUF

```

```

.PSECT $OWNS,NOEXE,2

00000 CMD_DSC: .BLKB 8
00008 SYSIN_BUF:
      .BLKB 1024

```

```

NCP PMT DSC= P.AAA
NCP$_CMDBUF DSC== P.AAC
.EXTRN NCP$INIT TRMIO, NCP$READ LINE
.EXTRN NCP$WRITE LINE, NCP$PARSE_CMD
.EXTRN NCP$UNTELL, ACT$VRB_CLEXEC
.EXTRN LIB$GET FOREIGN
.EXTRN NCP$G_BEGIN_ZERO
.EXTRN NCP$G_END_ZERO, NCP$G_STATE_TBL
.EXTRN NCP$G_KEY_TBL

```

```

.PSECT $CODE$,NOWRT,2

```

OFFC 00000 NCP\$MAIN:

```

      5B 00000000G 00 9E 00002 .WORD Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11 : 0110
      5A 00000000G 00 9E 00009 MOVAB NCP$PARSE_CMD, R11
      59 00000000G 00 9E 00010 MOVAB NCP$G_STATE_TBL, R10
      58 00000000G 00 9E 00017 MOVAB NCP$G_KEY_TBL, R9
      57 00000000' 00 9E 0001E MOVAB NCP$G_BEGIN_ZERO, R8
      56 00000000' 00 9E 00025 MOVAB NCP$_CMDBUF_DSC, R7
      5E 00000000' 08 C2 0002C MOVAB CMD_DSC, R6
      00 00000000G 00 FB 0002F SUBL2 #8, SP
      6D 00000000V 00 9E 00036 CALLS #0, NCP$INIT TRMIO : 0175
      MOVAB NCP$COND_HDLR, (FP) : 0178

```


NCPMAIN
V04-000

Main Entry
NCP\$MAIN Main Entry

N 14
15-Sep-1984 23:46:03
14-Sep-1984 12:48:14

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[NCP.SRC]NCPMAIN.B32;1

Page 7
(3)

		00000000G	00		00	FB	0003D	CALLS	#0, ACT\$VRB_CLEXEC	:	0180
					56	DD	00044	PUSHL	R6	:	0182
					7E	D4	00046	CLRL	-(SP)	:	
		00000000G	00		57	DD	00048	PUSHL	R7	:	
			1E		03	FB	0004A	CALLS	#3, LIB\$GET_FOREIGN	:	
					50	E9	00051	BLBC	R0, 1\$:	
					66	D5	00054	TSTL	CMD_DSC	:	0183
					1A	13	00056	BEQL	1\$:	
0000*	8F		04	A6	08	A6	9E	MOVAB	SYNBUF, CMD_DSC+4	:	0186
				6E		00	2C	MOVCS	#0, (SP), #0, #<NCP\$G_END_ZERO--	:	0187
						68			NCP\$G_BEGIN_ZERO>, NCP\$G_BEGIN_ZERO	:	
					4200	8F	BB	PUSHR	#*M<R9,SP>	:	0189
					0440	8F	BB	PUSHR	#*M<R6,R10>	:	
			6B		04	FB	0006D	CALLS	#4, NCP\$PARSE_CMD	:	
					37	11	00070	BRB	3\$:	0190
					56	DD	00072	1\$: PUSHL	R6	:	0199
					F8	A7	9F	PUSHAB	NCP_PMT_DSC	:	
					57	DD	00077	PUSHL	R7	:	
		00000000V	00		03	FB	00079	CALLS	#3, NCP\$READ_CMD	:	
			0A		50	E8	00080	BLBS	READ_STATUS, 2\$:	0204
		0001827A	8F		50	D1	00083	CMPL	READ_STATUS, #98938	:	0207
					1D	13	0008A	BEQL	3\$:	
					04		0008C	RET		:	0209
0000*	8F		00	6E		00	2C	2\$: MOVL	#0, (SP), #0, #<NCP\$G_END_ZERO--	:	0218
					68		00094		NCP\$G_BEGIN_ZERO>, NCP\$G_BEGIN_ZERO	:	
					4200	8F	BB	PUSHR	#*M<R9,SP>	:	0225
					0440	8F	BB	PUSHR	#*M<R6,R10>	:	
			6B		04	FB	0009D	CALLS	#4, NCP\$PARSE_CMD	:	
		00000000G	00		00	FB	000A0	CALLS	#0, NCP\$UNTELC	:	0232
					C9	11	000A7	BRB	1\$:	
			50		01	D0	000A9	3\$: MOVL	#1, R0	:	0236
					04		000AC	RET		:	0237

; Routine Size: 173 bytes, Routine Base: \$CODE\$ + 0000

```

241 0238 1 %SBTTL 'NCP$READ_CMD Read and build a command'
242 0239 1 GLOBAL ROUTINE NCP$READ_CMD (INP_DSC, PMT_DSC, RTN_DSC) = !
243 0240 1
244 0241 1 !++
245 0242 1 ! FUNCTIONAL DESCRIPTION:
246 0243 1 !
247 0244 1 !     Read a command with comments (!) and continuation lines (-)
248 0245 1 !     and return a descriptor to the line within the input buffer.
249 0246 1 !
250 0247 1 ! FORMAL PARAMETERS:
251 0248 1 !
252 0249 1 !     INP_DSC      Address of the input buffer descriptor
253 0250 1 !     PMT_DSC      Address of a descriptor of a prompt string
254 0251 1 !     RTN_DSC      Address of the descriptor to return
255 0252 1 !
256 0253 1 ! IMPLICIT INPUTS:
257 0254 1 !
258 0255 1 !     NONE
259 0256 1 !
260 0257 1 ! IMPLICIT OUTPUTS:
261 0258 1 !
262 0259 1 !     NONE
263 0260 1 !
264 0261 1 ! ROUTINE VALUE:
265 0262 1 ! COMPLETION CODES:
266 0263 1 !
267 0264 1 !     Return code from NCP$READ_LINE
268 0265 1 !
269 0266 1 ! SIDE EFFECTS:
270 0267 1 !
271 0268 1 !     NONE
272 0269 1 !
273 0270 1 ! --
274 0271 1 !
275 0272 2 ! BEGIN
276 0273 2 !
277 0274 2 ! LITERAL
278 0275 2 !     SPACE = 32, ! Character code for a space
279 0276 2 !     TAB   = 9  ! Character code for a tab
280 0277 2 !
281 0278 2 !
282 0279 2 ! MAP
283 0280 2 !     INP_DSC : REF VECTOR [2], ! Input buffer descriptor
284 0281 2 !     RTN_DSC : REF VECTOR [2], ! Return command descriptor
285 0282 2 !     PMT_DSC : REF VECTOR [2], ! Prompt descriptor
286 0283 2 !
287 0284 2 !
288 0285 2 ! LOCAL
289 0286 2 !     CMD_DSC : VECTOR [2], ! Temp descriptor for command
290 0287 2 !     STATUS, ! Hold status value here
291 0288 2 !     CHAR, ! Temp for character
292 0289 2 !     LAST_ADR ! Temp for address of last byte of
293 0290 2 !     command
294 0291 2 !
295 0292 2 !
296 0293 2 !     CMD_DSC [0] = .INP_DSC [0]; ! Initialize temp dsc
297 0294 2 !     CMD_DSC [1] = .INP_DSC [1];

```

```

298 0295 2
299 0296
300 0297 DO ! Read and skip comments
301 0298 IF NOT (STATUS = ! Read a line
302 0299 NCP$READ_LINE
303 0300 ( ! Into here
304 0301 CMD_DSC, ! With this prompt
305 0302 .PMT_DSC, ! Return descriptor here
306 0303 .RTN_DSC
307 0304 )
308 0305
309 0306 THEN RETURN .STATUS ! Return failure
310 0307 WHILE CH$RCHAR (CH$PTR(.RTN_DSC [1])) !
311 0308 EQLU '!' ! Comments begin with !
312 0309
313 0310
314 0311 DO
315 0312 BEGIN
316 0313 LAST_ADR = .RTN_DSC [1] + ! Address of last char + 1
317 0314 .RTN_DSC [0];
318 0315 DO
319 0316 BEGIN
320 0317 LAST_ADR = .LAST_ADR - 1; ! Backup last address
321 0318 CHAR = CH$RCHAR ? ! and fetch the character there
322 0319 CH$PTR (.LAST_ADR) )
323 0320 END
324 0321 WHILE (.CHAR EQLU SPACE) OR ! Scan back until not blank
325 0322 (.CHAR EQLU TAB)
326 0323
327 0324 IF .CHAR NEQU '-' ! If this is not a continuation line
328 0325 THEN EXITLOOP ! nope, we are done
329 0326
330 0327 .CMD_DSC [1] = .LAST_ADR; ! Read starting here
331 0328 .CMD_DSC [0] = .CMD_DSC [0] ! For the remaining space
332 0329 - .RTN_DSC [0]
333 0330 END
334 0331 WHILE
335 0332 (
336 0333 STATUS =
337 0334 NCP$READ_LINE ! Read another line
338 0335 (
339 0336 CMD_DSC,
340 0337 ASCID (' '), ! Prompt for continuation lines
341 0338 .RTN_DSC
342 0339 )
343 0340 )
344 0341
345 0342 .RTN_DSC [0] = .LAST_ADR ! Return appropriate length
346 0343 - .INP_DSC [1] + 1 ;
347 0344 .RTN_DSC [1] = .INP_DSC [1]; ! and address
348 0345 RETURN .STATUS ! and the status code
349 0346 END;

```

.PSECT SPLITS,NOWRT,NOEXE,2

00 00 20 5F 00014 P.AAE: .ASCII \ \<0><0>
00000002 00018 P.AAD: .LONG 2
00000000 0001C .ADDRESS P.AAE

.PSECT \$CODE\$,NOWRT,2

			007C 00000	.ENTRY	NCP\$READ_CMD, Save R2,R3,R4,R5,R6	: 0239
56	00000000G	00	9E 0J002	MOVAB	NCP\$READ_LINE, R6	:
5E		04	C2 00009	SUBL2	#4, SP	:
54		04	AC D0 0000C	MOVL	INP_DSC, R4	: 0293
			64 DD 00010	PUSHL	(R4)	:
04	AE	04	A4 D0 00012	MOVL	4(R4), CMD_DSC+4	: 0294
	53	0C	AC D0 00017	MOVL	RTN_DSC, R3	: 0302
			53 DD 0001B 1\$:	PUSHL	R3	:
		08	AC DD 0001D	PUSHL	PMT_DSC	: 0301
		08	AE 9F 00020	PUSHAB	CMD_DSC	: 0299
66		03	FB 00023	CALLS	#3, NCP\$READ_LINE	:
42		50	E9 00026	BLBC	STATUS, 5\$:
21		04	B3 91 07029	CMPB	@4(R3), #33	: 0308
			EC 13 0C02D	BEQL	1\$:
52	04	A3	63 C1 0002F 2\$:	ADDL3	(R3), 4(R3), LAST_ADR	: 0314
		55	72 9A 00034 3\$:	MOVZBL	-(LAST_ADR), CHAR	: 0319
		20	55 D1 00037	C MPL	CHAR, #32	: 0321
			F8 13 0003A	BEQL	3\$:
		09	55 D1 0003C	C MPL	CHAR, #9	: 0322
			F3 13 0003F	BEQL	3\$:
		2D	55 D1 00041	C MPL	CHAR, #45	: 0324
			18 12 00044	BNEQ	4\$:
04	AE	52	D0 07046	MOVL	LAST_ADR, CMD_DSC+4	: 0327
	6E	63	C2 C004A	SUBL2	(R3), CMD_DSC	: 0329
			53 DD 0004D	PUSHL	R3	: 0338
			00 9F 0004F	PUSHAB	P.AAD	: 0337
		08	AE 9F 00055	PUSHAB	CMD_DSC	: 0335
66		03	FB 00058	CALLS	#3, NCP\$READ_LINE	:
D1		50	E8 0005B	BLBS	STATUS, 2\$:
52		04	A4 C2 0005E 4\$:	SUBL2	4(R4), R2	: 0343
		63	01 A2 9E 00062	MOVAB	1(R2), (R3)	:
04	A3	04	A4 D0 00066	MOVL	4(R4), 4(R3)	: 0344
			C4 0006B 5\$:	RET		: 0346

: Routine Size: 108 bytes, Routine Base: \$CODE\$ + 00AD

```

351 0347 1 %SBTTL 'NCP$COND_HDLR Condition handler'
352 0348 1 ROUTINE NCP$COND_HDLR (SIGARY, MCHARY) =
353 0349 1
354 0350 1
355 0351 1 **
356 0352 1 FUNCTIONAL DESCRIPTION:
357 0353 1 Main condition handler for NCP. It receives a signal which is
358 0354 1 either an internal error or a standard system status. If it is
359 0355 1 an internal error, it prints a message. Otherwise it resignals
360 0356 1 the condition.
361 0357 1
362 0358 1 FORMAL PARAMETERS:
363 0359 1
364 0360 1 Standard condition handler argument list
365 0361 1 SIGARY Address of signal array
366 0362 1 MCHARY Address of mechanism array
367 0363 1
368 0364 1 IMPLICIT INPUTS:
369 0365 1
370 0366 1 NONE
371 0367 1
372 0368 1 IMPLICIT OUTPUTS:
373 0369 1
374 0370 1 NONE
375 0371 1
376 0372 1 ROUTINE VALUE:
377 0373 1 COMPLETION CODES:
378 0374 1
379 0375 1 Condition Resignaled, stack unwound
380 0376 1
381 0377 1 SIDE EFFECTS:
382 0378 1
383 0379 1 NONE
384 0380 1
385 0381 1 --
386 0382 1
387 0383 1 BEGIN
388 0384 1
389 0385 1 MAP
390 0386 1 SIGARY : REF BBLOCK, ! Signal array
391 0387 1 MCHARY : REF BBLOCK ! Mechanism array
392 0388 1 ;
393 0389 1
394 0390 1 LOCAL
395 0391 1 ERRCODE : BBLOCK [4] ! Error code value
396 0392 1 ;
397 0393 1
398 0394 1 EXTERNAL LITERAL
399 0395 1 NCP$_FACILITY ! NCP Facility code
400 0396 1 ;
401 0397 1
402 0398 1 ERRCODE = .SIGARY [CHF$L_SIG_NAME]; ! The value of the status code
403 0399 1
404 0400 1 IF .ERRCODE [ST$V_FAC_NO] ! Is it ours?
405 0401 1 EQL NCP$_FACILITY
406 0402 1 THEN ! yes
407 0403 1 BEGIN

```

```

: 408      0404      3      SIGARY [CHF$SIG_ARGS] =      ! Make signal array into message vec
: 409      0405      3      .SIGARY [CHF$SIG_ARGS] - 2      !
: 410      0406      3      $PUTMSG      ! Write the message out
: 411      0407      3      (
: 412      0408      3      MSGVEC = .SIGARY
: 413      0409      3      );
: 414      0410      3      SIGARY [CHF$SIG_ARGS] =      ! Make mess vec into signal array
: 415      0411      3      .SIGARY [CHF$SIG_ARGS] + 2
: 416      0412      3      END
: 417      0413      3      ELSE
: 418      0414      3      RETURN S$$_RESIGNAL      ! Signal the error to the next handler
: 419      0415      3      ;
: 420      0416      3      MCHARY [CHF$MCH_SAVRO] = SUCCESS; ! Set success for status on unwind
: 421      0417      3      IF
: 422      0418      3      .BBLOCK [SIGARY [CHF$SIG_NAME], ST$V_SEVERITY]
: 423      0419      3      EQL
: 424      0420      3      ST$K_SEVERE
: 425      0421      3      THEN      ! If severe error then unwind
: 426      0422      3      BEGIN
: 427      0423      3      BEGIN
: 428      0424      3      Perform cleanup operations here
: 429      0425      3      ;
: 430      0426      3      ;
: 431      0427      3      ;
: 432      0428      3      ;
: 433      0429      3      ;
: 434      0430      3      ;
: 435      0431      3      $UNWIND (DEPADR = MCHARY [CHF$MCH_DEPTH] )
: 436      0432      3      END
: 437      0433      3      ;
: 438      0434      3      ;
: 439      0435      3      RETURN S$$_CONTINUE      ! Otherwise, continue from error
: 440      0436      3      ;
: 441      0437      3      END;

```

```

.EXTRN NCP$ FACILITY, SY$$PUTMSG
.EXTRN SY$$UNWIND

```

				0004 0000 NCP\$COND_HDLR:			
				WORD	Save R2		: 0348
				MOVL	SIGARY, R2		: 0398
000000GOG	8F	50	0C	MOVL	4(R2), ERRCODE		: 0401
				CMPZV	#16, #12, ERRCODE, #NCP\$_FACILITY		: 0405
			62	BNEQ	1\$: 0409
				SUBL2	#2, (R2)		: 0411
				CLRQ	-(SP)		: 0410
				CLRL	-(SP)		: 0414
				PUSHL	R2		: 0417
			0000000G	CALLS	#4, SY\$\$PUTMSG		: 0411
			62	ADDL2	#2, (R2)		: 0410
				BRB	2\$: 0414
			50	MOVZWL	#2328, R0		: 0417
				RET			: 0417
			50	MOVL	MCHARY, R0		: 0417
			0C	MOVL	#1, 12(R0)		: 0417
			A0				: 0417
							: 0417

NCPMAIN
V04-000

Main Entry
NCP\$COND_HDLR Condition handler

H 15
15-Sep-1984 23:46:03
14-Sep-1984 12:48:14

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[NCP.SRC]NCPMAIN.B32;1 Page 14
(6)

: 443 0438 1
: 444 0439 1 END !End of module
: 445 0440 0 ELUDOM

PSECT SUMMARY

Name	Bytes	Attributes
SOWNS	1032	NOVEC, WRT, RD, NOEXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)
SPLITS	32	NOVEC, NOWRT, RD, NOEXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)
\$CODES	361	NOVEC, NOWRT, RD, EXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)

Library Statistics

File	Total	Symbols Loaded	Percent	Pages Mapped	Processing Time
_\$255\$DUA28:[NCP.OBJ]NCPLIBRY.L32;1	373	5	1	52	00:00.1
_\$255\$DUA28:[SYSLIB]STARLET.L32;1	9776	14	0	581	00:01.5
_\$255\$DUA28:[SYSLIB]TPAMAC.L32;1	42	0	0	14	00:00.1

COMMAND QUALIFIERS

BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LISS:NCPMAIN/OBJ=OBJ\$:NCPMAIN MSRCS:NCPMAIN/UPDATE=(ENHS:NCPMAIN)

: Size: 361 code + 1064 data bytes
: Run Time: 00:09.1
: Elapsed Time: 00:31.5
: Lines/CPU Min: 2910
: Lexemes/CPU-Min: 13078
: Memory Used: 74 pages
: Compilation Complete

NCPLIBRY B32	NCPCONCAR LIS	NCPERRMSG LIS	NCPCONMAN LIS	NCPMAIN LIS	NCPNETIO LIS
NMAHEAD B32	NCLIBRY LIS	NMATAIL B32			