

NNN		NNN	CCCCCCCCCCCC	PPPPPPPPPPPP	
NNN		NNN	CCCCCCCCCCCC	PPPPPPPPPPPP	
NNN		NNN	CCCCCCCCCCCC	PPPPPPPPPPPP	
NNN		NNN	CCC	PPP	PPP
NNN		NNN	CCC	PPP	PPP
NNN		NNN	CCC	PPP	PPP
NNNNNN		NNN	CCC	PPP	PPP
NNNNNN		NNN	CCC	PPP	PPP
NNNNNN		NNN	CCC	PPP	PPP
NNN	NNN	NNN	CCC	PPPPPPPPPPPP	
NNN	NNN	NNN	CCC	PPPPPPPPPPPP	
NNN	NNN	NNN	CCC	PPPPPPPPPPPP	
NNN		NNNNNN	CCC	PPP	
NNN		NNNNNN	CCC	PPP	
NNN		NNNNNN	CCC	PPP	
NNN		NNN	CCC	PPP	
NNN		NNN	CCC	PPP	
NNN		NNN	CCC	PPP	
NNN		NNN	CCC	PPP	
NNN		NNN	CCCCCCCCCCCC	PPP	
NNN		NNN	CCCCCCCCCCCC	PPP	
NNN		NNN	CCCCCCCCCCCC	PPP	

NCI
V04

:

```

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

```

```

LL      I I I I I  SSSSSSSS
LL      I I I I I  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
LLLLLLLLLLLL  I I I I I  SSSSSSSS
LLLLLLLLLLLL  I I I I I  SSSSSSSS

```



```
0001 0 %TITLE 'Main Entry'
0002 0 MODULE NCPMAIN (IDENT = 'V04-000', MAIN = NCP$MAIN,
0003 0 ADDRESSING_MODE(EXTERNAL=GENERAL),
0004 0 ADDRESSING_MODE(NONEXTERNAL=GENERAL)) =
0005 1 BEGIN
0006 1
0007 1
0008 1 *****
0009 1 *
0010 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0011 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0012 1 * ALL RIGHTS RESERVED.
0013 1 *
0014 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0015 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0016 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0017 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0018 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0019 1 * TRANSFERRED.
0020 1 *
0021 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0022 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0023 1 * CORPORATION.
0024 1 *
0025 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0026 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0027 1 *
0028 1 *
0029 1 *****
0030 1
0031 1
0032 1 ++
0033 1 FACILITY: NCP Network Control Program
0034 1
0035 1 ABSTRACT:
0036 1
0037 1 NCP main entry and command line acceptance
0038 1
0039 1 ENVIRONMENT: VAX/VMS Operating System
0040 1
0041 1 AUTHOR: Darrell Duffy , CREATION DATE: 17-August-1979
0042 1
0043 1 MODIFIED BY:
0044 1
0045 1 V03-003 PRD0057 Paul R. DeStefano 05-Feb-1984
0046 1 No longer need to save CLI verify state.
0047 1
0048 1 V03-002 RPG0002 Bob Grosso 03-Aug-1982
0049 1 Stop NCP from automatically starting the NETACT.
0050 1
0051 1 V03-001 RPG47232 Bob Grosso 21-Jul-1982
0052 1 Increase size of command input buffer, sysin_buf_siz.
0053 1
0054 1 V002 TMH0002 Tim Halvorsen 13-Jul-1981
0055 1 Change all non-local references to general addressing.
0056 1
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 'OBJ$:NCPLIBRY.L32';
79 0078 1 LIBRARY 'SYSSLIBRARY:STARLET.L32';
80 0079 1 LIBRARY 'SYSSLIBRARY:TPAMAC.L32';
81 0080 1
82 0081 1
83 0082 1 : MACROS:
84 0083 1
85 0084 1
86 0085 1
87 0086 1 : EQUATED SYMBOLS:
88 0087 1
89 0088 1
90 0089 1
91 0090 1
92 0091 1 : OWN STORAGE:
93 0092 1
94 0093 1
95 0094 1
96 0095 1
97 0096 1 : EXTERNAL REFERENCES:
98 0097 1
99 0098 1
100 0099 1 EXTERNAL ROUTINE
101 0100 1 NCP$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
```

NCPMAIN
V04-000

Main Entry
NCP\$MAIN Main Entry

K 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 4
(3)

```

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 Main entry for NCP
117 0115 1
118 0116 1 FORMAL PARAMETERS:
119 0117 1
120 0118 1 NONE
121 0119 1
122 0120 1 IMPLICIT INPUTS:
123 0121 1
124 0122 1 NONE
125 0123 1
126 0124 1 ROUTINE VALUE:
127 0125 1 COMPLETION CODES:
128 0126 1
129 0127 1 Exit code for NCP
130 0128 1
131 0129 1 SIDE EFFECTS:
132 0130 1
133 0131 1 LOTS
134 0132 1
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$input buffer
152 0150 2 VECTOR [SYSIN_BUF_SIZ, BYTE]
153 0151 2 ;
154 0152 2
155 0153 2 BIND
156 0154 2 NCP_PMT_DSC = ! Descriptor of NCP prompt
157 0155 2 ASCID ('NCP>'),
158 0156 2 CLIBLK = AP: REF BBLOCK ! Arg block from calling cli (DCL)
159 0157 2 ;
160 0158 2
161 0159 2 GLOBAL BIND
162 0160 2 NCP$_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 EXTERNAL
172 0170 2 NCP$G_BEGIN_ZERO, ! Parameter data blocks to clear
173 0171 2 NCP$G_END_ZERO, ! End of the blocks
174 0172 2 NCP$G_STATE_TBL, ! State table for parsing
175 0173 2 NCP$G_KEY_TBL
176 0174 2 ;
177 0175 2 NCP$INIT_TRMIO (); ! Initialize terminal I/O
178 0176 2
179 0177 2 ! NCP$STARTNETACP (); ! Start the network, if not started
180 0178 2 .FP = NCP$COND_HDLR; ! Enable the handler
181 0179 2
182 0180 2 ACT$VRB_CLEEXEC (); ! Establish link to local node
183 0181 2
184 0182 2 IF LIB$GET_FOREIGN(NCP$ CMDBUF_DSC,0,CMD_DSC) ! If a foreign command,
185 0183 2 AND .CMD_DSC [0] NEQ 0 ! and there is a command line,
186 0184 2 THEN
187 0185 2 BEGIN
188 0186 2 CMD_DSC [1] = SYSIN_BUF; ! Construct descriptor of line
189 0187 2 CH$FILL(0,NCP$G_END_ZERO-NCP$G_BEGIN_ZERO, ! Zero parameters
190 0188 2 NCP$G_BEGIN_ZERO);
191 0189 2 NCP$PARSE_CMD(CMD_DSC,NCP$G_STATE_TBL,NCP$G_KEY_TBL,RTN_DSC);
192 0190 2 RETURN TRUE;
193 0191 2 END;
194 0192 2
195 0193 2 WHILE
196 0194 2 BEGIN
197 0195 2 LOCAL
198 0196 2 READ_STATUS;
199 0197 2
200 0198 2 READ_STATUS = NCP$READ_CMD ! Read line into buffer
201 0199 2 ?
202 0200 2 NCP$ CMDBUF_DSC, ! Buffer descriptor
203 0201 2 NCP$ PMT_DSC, ! Prompt descriptor
204 0202 2 CMD_DSC ! Return dsc of line
205 0203 2 );
206 0204 2 IF NOT .READ_STATUS
207 0205 2 THEN
208 0206 2 BEGIN
209 0207 2 IF (.READ_STATUS NEQ RMSS_EOF)
210 0208 2 THEN
211 0209 2 RETURN .READ_STATUS
212 0210 2 ELSE FALSE ! EOF terminates without error
213 0211 2 END
214 0212 2 ELSE TRUE
215 0213 2 END
216 0214 2 DO ! Do so until error returned
217 0215 2 BEGIN
218 0216 2
219 0217 2 CH$FILL ! Zero the parameter block area
220 0218 2 (
221 0219 2 0, ! Fill character
222 0220 2 NCP$G_END_ZERO - NCP$G_BEGIN_ZERO, ! Count
223 0221 2 NCP$G_BEGIN_ZERO ! Beginning address
224 0222 2 );
```

NCPMAIN
V04-000

Main Entry
NCP\$MAIN Main Entry

M 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 6
(3)

```

: 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$PARSE_CMD
      (
      CMD_DSC
      NCP$G_STATE_TBL,
      NCP$G_KEY_TBL,
      RTN_DSC
      );

      NCP$UNTELL ( )

      END

      RETURN TRUE
      END;
```

```

! Parse the command

! Remove the tell executor

! Return with success
```

```

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

.PSECT $PLITS,NOWRT,NOEXE,2
```

```

3E 50 43 4E 00000 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$_CMDBUF_DSC, R7
5E 00000000' 00 9E 0002C MOVAB CMD_DSC, R6
00 00000000G 00 C2 0002C SUBL2 #8, SP
6D 00000000V 00 FB 0002F CALLS #0, NCP$INIT TRMIO : 0175
      00 9E 00036 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	00058	MOVAB	SYN BUF, CMD DSC+4	0186
				6E		00	2C	0005D	MOVCS	#0, (SP), #0, #<NCP\$G_END_ZERO--	0187
						68		00064		NCP\$G_BEGIN_ZERO>, NCP\$G_BEGIN_ZERO	
					4200	8F	BB	00065	PUSHR	#^M<R9,SP>	0189
					0440	8F	BB	00069	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	00074	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	0008D	2\$: MOVCS	#0, (SP), #0, #<NCP\$G_END_ZERO--	0218
						68		00094		NCP\$G_BEGIN_ZERO>, NCP\$G_BEGIN_ZERO	
					4200	8F	BB	00095	PUSHR	#^M<R9,SP>	0225
					0440	8F	BB	00099	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      2
: 300      0297      2
: 301      0298      2
: 302      0299      2
: 303      0300      2
: 304      0301      2
: 305      0302      2
: 306      0303      2
: 307      0304      2
: 308      0305      2
: 309      0306      2
: 310      0307      2
: 311      0308      2
: 312      0309      2
: 313      0310      2
: 314      0311      2
: 315      0312      2
: 316      0313      2
: 317      0314      2
: 318      0315      2
: 319      0316      2
: 320      0317      2
: 321      0318      2
: 322      0319      2
: 323      0320      2
: 324      0321      2
: 325      0322      2
: 326      0323      2
: 327      0324      2
: 328      0325      2
: 329      0326      2
: 330      0327      2
: 331      0328      2
: 332      0329      2
: 333      0330      2
: 334      0331      2
: 335      0332      2
: 336      0333      2
: 337      0334      2
: 338      0335      2
: 339      0336      2
: 340      0337      2
: 341      0338      2
: 342      0339      2
: 343      0340      2
: 344      0341      2
: 345      0342      2
: 346      0343      2
: 347      0344      2
: 348      0345      2
: 349      0346      1

DO
    IF NOT (STATUS =
        NCP$READ_LINE
        (
            CMD_DSC,
            .PMT_DSC,
            .RTN_DSC
        )
    )
    THEN RETURN .STATUS
    WHILE CH$RCHAR (CH$PTR(.RTN_DSC [1]))
        EQLU '!'
    ;
DO
    BEGIN
    LAST_ADR = .RTN_DSC [1] +
        .RTN_DSC [0];
    DO
        BEGIN
        LAST_ADR = .LAST_ADR - 1;
        CHAR = CH$RCHAR (
            CH$PTR (.LAST_ADR) )
        END
    WHILE (.CHAR EQLU SPACE) OR
        (.CHAR EQLU TAB)
    ;
    IF .CHAR NEQU '-'
    THEN EXITLOOP
    ;
    CMD_DSC [1] = .LAST_ADR;
    CMD_DSC [0] = .CMD_DSC [0]
        - .RTN_DSC [0]
    END
    WHILE
    (
        STATUS =
        NCP$READ_LINE
        (
            CMD_DSC,
            ASCII (
                .RTN_DSC
            )
        )
    )
    ;
    RTN_DSC [0] = .LAST_ADR
        - .INP_DSC [1] + 1 ;
    RTN_DSC [1] = .INP_DSC [1];
    RETURN .STATUS
END;

```

! Read and skip comments

! Read a line

! Into here

! With this prompt

! Return descriptor here

! Return failure

! Comments begin with !

! Address of last char + 1

! Backup last address

! and fetch the character there

! Scan back until not blank

! If this is not a continuation line

! nope, we are done

! Read starting here

! For the remaining space

! Read another line

! Prompt for continuation lines

! Return appropriate length

! and address

! and the status code

.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

56	00000000G	00	9E	00002	.ENTRY	NCP\$READ_CMD, Save R2,R3,R4,R5,R6	0239
5E		04	C2	00009	MOVAB	NCP\$READ_LINE, R6	
54	04	AC	D0	0000C	SUBL2	#4, SP	
		64	DD	00010	MOVL	INP_DSC, R4	0293
04	AE	04	A4	00012	PUSHL	(R4)	
53	0C	AC	D0	00017	MOVL	4(R4), CMD_DSC+4	0294
		53	DD	0001B	MOVL	RTN_DSC, R3	0302
		08	AC	DD	PUSHL	R3	
		08	AE	9F	PUSHL	PMT_DSC	0301
66		03	FB	00023	PUSHAB	CMD_DSC	0299
42		50	E9	00026	CALLS	#3, NCP\$READ_LINE	
21	04	B3	91	00029	BLBC	STATUS, 5\$	
		EC	13	0002D	CMPB	@4(R3), #33	0308
52	04	A3	63	0002F	BEQL	1\$	
55		72	9A	00034	ADDL3	(R3), 4(R3), LAST_ADR	0314
20		55	D1	00037	MOVZBL	-(LAST_ADR), CHAR	0319
		F8	13	0003A	CMPL	CHAR, #32	0321
09		55	D1	0003C	BEQL	3\$	
		F3	13	0003F	CMPL	CHAR, #9	0322
2D		55	D1	00041	BEQL	3\$	
		18	12	00044	CMPL	CHAR, #45	0324
04	AE	52	D0	00046	BNEQ	4\$	
6E		63	C2	0004A	MOVL	LAST_ADR, CMD_DSC+4	0327
		53	DD	0004D	SUBL2	(R3), CMD_DSC	0329
	00000000'	00	9F	0004F	PUSHL	R3	0338
	08	AE	9F	00055	PUSHAB	P.AAD	0337
66		03	FB	00058	PUSHAB	CMD_DSC	0335
D1		50	E8	0005B	CALLS	#3, NCP\$READ_LINE	
52	04	A4	C2	0005E	BLBS	STATUS, 2\$	
63	01	A2	9E	00062	SUBL2	4(R4), R2	0343
04	A3	04	A4	00066	MOVAB	1(R2), (R3)	
		04	D0	0006B	MOVL	4(R4), 4(R3)	0344
					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 FUNCTIONAL DESCRIPTION:
356 0352 1
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 2 BEGIN
388 0384 2
389 0385 2 MAP
390 0386 2 SIGARY : REF BBLOCK, ! Signal array
391 0387 2 MCHARY : REF BBLOCK ! Mechanism array
392 0388 2 ;
393 0389 2
394 0390 2 LOCAL
395 0391 2 ERRCODE : BBLOCK [4] ! Error code value
396 0392 2 ;
397 0393 2
398 0394 2 EXTERNAL LITERAL
399 0395 2 NCP$_FACILITY ! NCP Facility code
400 0396 2 ;
401 0397 2
402 0398 2 ERRCODE = .SIGARY [CHF$L_SIG_NAME]; ! The value of the status code
403 0399 2
404 0400 2 IF .ERRCODE [ST$V_FAC_NO] ! Is it ours?
405 0401 2 EQL NCP$_FACILITY
406 0402 2 THEN ! yes
407 0403 3 BEGIN
```

```

: 408      0404      3      SIGARY [CHF$SIG_ARGS] =      ! Make signal array into message vec
: 409      0405      3      .SIGARY [CHF$SIG_ARGS] - 2;
: 410      P 0406      3      $PUTMSG      ! Write the message out
: 411      P 0407      3      (
: 412      P 0408      3      MSGVEC = .SIGARY
: 413      0409      3      );
: 414      0410      3      SIGARY [CHF$SIG_ARGS] =      ! Make mess vec into signal array
: 415      0411      3      .SIGARY [CHF$SIG_ARGS] + 2
: 416      0412      3      END
: 417      0413      3      ELSE
: 418      0414      3      RETURN SSS_RESIGNAL      ! Signal the error to the next handler
: 419      0415      3      ;
: 420      0416      3      MCHARY [CHF$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      Perform cleanup operations here
: 428      0424      3      ;
: 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      ! Unwind the stack to the caller
: 437      0433      3      ! of the establisher
: 438      0434      3      END
: 439      0435      3      ;
: 440      0436      3      RETURN SSS_CONTINUE      ! Otherwise, continue from error
: 441      0437      3      END;
```

```

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

                                0004 00000 NCP$COND_HDLR:
                                .WORD Save R2
                                52      04      AC      D0      00002      MOVL      SIGARY, R2      : 0348
                                50      04      A2      D0      00006      MOVL      4(R2), ERRCODE      : 0398
                                0C      10      ED      0000A      CMPZV      #16, #12, ERRCODE, #NCP$_FACILITY      : 0401
                                62      15      12      00013      BNEQ      1$      : 0405
                                7E      02      C2      00015      SUBL2      #2, (R2)      : 0409
                                7E      7C      00018      CLRQ      -(SP)
                                52      D4      0001A      CLRL      -(SP)
                                00      DD      0001C      PUSHL      R2
                                62      04      FB      0001E      CALLS      #4, SYSS$PUTMSG
                                50      02      C0      00025      ADDL2      #2, (R2)      : 0411
                                0C      06      11      00028      BRB      2$      : 0410
                                50      0918      8F      3C      0002A      1$:      MOVZWL      #2328, R0      : 0414
                                0C      50      08      AC      D0      00030      2$:      MOVL      MCHARY, R0      : 0417
                                A0      01      D0      00034      MOVL      #1, 12(R0)
```


NCPMAIN
V04-000

Main Entry
NCP\$COND_HDLR Condition handler

G 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 13
(5)

04	04	A2	03	00	ED	00038	CMPZV	#0, #3, 4(R2), #4	:	0420
				0C	12	0003E	BNEQ	3\$	:	
				7E	D4	00040	CLRL	-(SP)	:	0431
				A0	9F	00042	PUSHAB	8(R0)	:	
		00000000G	00	02	FB	00045	CALLS	#2, SYSSUNWIND	:	
			50	01	D0	0004C	MOVL	#1, R0	:	0435
				04	0004F	3\$:	RET		:	0437

; Routine Size: 80 bytes, Routine Base: \$CODE\$ + 0119

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

```
; 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
```


0267 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

NCPCONCAR
LIS

NCPERRMSG
LIS

NCPCONMAN
LIS

NCPLIBRY
B32

NCPMAIN
LIS

NCPNETIO
LIS

NMAHEAD
B32

NCPLIBRY
LIS

NMATAIL
B32