

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

.....

```

NN      NN      CCCCCCCC PPPPPPPP SSSSSSSS TTTTTTTTTT AAAAAA CCCCCCCC LL      PPPPPPPP
NN      NN      CCCCCCCC PPPPPPPP PP      PP SS      SS TTTTTTTTTT AAAAAA CCCCCCCC LL      PPPPPPPP
NN      NN      CC      PP      PP SS      SS TT      TT AA      AA CC      LL      PP      PP
NN      NN      CC      PP      PP SS      SS TT      TT AA      AA CC      LL      PP      PP
NN      NN      CC      PP      PP SS      SS TT      TT AA      AA CC      LL      PP      PP
NN      NN      CC      PPPPPPPP SSSSSS TT      TT AA      AA CC      LL      PPPPPPPP
NN      NN      CC      PPPPPPPP SSSSSS TT      TT AA      AA CC      LL      PPPPPPPP
NN      NN      CC      PP      SS      TT      TT AAAAAAAAAA CC      LL      PP
NN      NN      CC      PP      SS      TT      TT AAAAAAAAAA CC      LL      PP
NN      NN      CC      PP      SS      TT      TT AA      AA CC      LL      PP
NN      NN      CC      PP      SS      TT      TT AA      AA CC      LL      PP
NN      NN      CCCCCCCC PP      SSSSSSSS TT      TT AA      AA CCCCCCCC LLLLLLLLLL PP
NN      NN      CCCCCCCC PP      SSSSSSSS TT      TT AA      AA CCCCCCCC LLLLLLLLLL PP

```

```

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
LL      IIIIII SSSSSSSS
LL      IIIIII SSSSSSSS

```

```
0001 0 XTITLE 'Clear/Purge Parse States and Data'
0002 0 MODULE NCPSTACLP (IDENT = 'V04-000', LIST(NOOBJECT)) =
0003 1 BEGIN
0004 1
0005 1
0006 1 *****
0007 1 *
0008 1 *   COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0009 1 *   DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0010 1 *   ALL RIGHTS RESERVED.
0011 1 *
0012 1 *   THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0013 1 *   ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0014 1 *   INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0015 1 *   COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0016 1 *   OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0017 1 *   TRANSFERRED.
0018 1 *
0019 1 *   THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0020 1 *   AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0021 1 *   CORPORATION.
0022 1 *
0023 1 *   DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0024 1 *   SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0025 1 *
0026 1 *****
0027 1
0028 1
0029 1
0030 1 ++
0031 1 FACILITY:      Network Control Program (NCP)
0032 1
0033 1 ABSTRACT:
0034 1
0035 1     States and data for the parsing of NCP clear and purge commands
0036 1
0037 1 ENVIRONMENT:  VAX/VMS Operating System
0038 1
0039 1 AUTHOR:      Darrell Duffy   , CREATION DATE: 14-September-79
0040 1
0041 1 MODIFIED BY:
0042 1
0043 1     V03-019 PRD0053      Paul R. DeStefano      05-Feb-1984
0044 1             Allow OBJECT parameter to accept both name and number.
0045 1
0046 1     V03-018 PRD0052      Paul R. DeStefano      05-Feb-1984
0047 1             Complete addition of and enable X25-Access parsing.
0048 1
0049 1     V03-017 RPG0017      Bob Grosso              01-Oct-1982
0050 1             Clear module Configurator.
0051 1
0052 1     V03-016 RPG0016      Bob Grosso              21-Sep-1982
0053 1             Add clear DTE and Clear Group and Clear Tracepoint
0054 1
0055 1     V03-015 RPG0015      Bob Grosso              09-Aug-1982
0056 1             Add CLEAR LINE MODE/MICROCODE.
0057 1
```


58	0058	1	V03-014	RPG0014	Bob Grosso	15-Jul-1982
59	0059	1		Add CLEAR/PURGE X25-Trace, Module Configurator, Console		
60	0060	1		loader, loopers.		
61	0061	1				
62	0062	1	V013	RPG0013	Bob Grosso	09-Jul-1982
63	0063	1		Add NI node parameters.		
64	0064	1		Remove CLEAR/PURGE NODE/EXECUTOR since parse tables have		
65	0065	1		have exceeded keyword limit.		
66	0066	1				
67	0067	1	V012	RPG0012	Bob Grosso	10-May-1982
68	0068	1		Add support for circuit MRT and RPR for NI support.		
69	0069	1		Change MAX BLOCK to MAX DATA.		
70	0070	1				
71	0071	1	V011	TMH0011	Tim Halvorsen	08-Mar-1982
72	0072	1		Remove prompting for CLEAR LINE, CIRCUIT and NODE		
73	0073	1		parameters, since it is somewhat confusing, and		
74	0074	1		these commands are normally used only to clear the		
75	0075	1		entire database entry, rather than a single parameter.		
76	0076	1		It also makes it easier to hide latent support of		
77	0077	1		parameters.		
78	0078	1				
79	0079	1	V010	TMH0010	Tim Halvorsen	20-Jan-1982
80	0080	1		Add CIRCUIT TRANSPORT TYPE, VERIFICATION parameters.		
81	0081	1		Remove prompting of X25 specific parameters.		
82	0082	1				
83	0083	1	V009	TMH0009	Tim Halvorsen	08-Jan-1982
84	0084	1		Remove TMH0004, thus restoring RETRANSMIT TIMER		
85	0085	1		to a line parameter, which is what NM V3.0 finally		
86	0086	1		came up with.		
87	0087	1		Add additional prompting for CLEAR LINE.		
88	0088	1				
89	0089	1	V008	TMH0008	Tim Halvorsen	18-Dec-1981
90	0090	1		Add CLEAR OBJECT PROXY, EXECUTOR DEFAULT PROXY & NODE PROXY.		
91	0091	1				
92	0092	1	V007	TMH0007	Tim Halvorsen	11-Nov-1981
93	0093	1		Add circuit source type to clear logging.		
94	0094	1				
95	0095	1	V006	TMH0006	Tim Halvorsen	22-Oct-1981
96	0096	1		Fix CLEAR CIRCUIT TRIBUTARY to send the right		
97	0097	1		NICE parameter code. Add CLEAR EXEC DEFAULT ACCESS		
98	0098	1		and CLEAR NODE ACCESS.		
99	0099	1				
100	0100	1	V005	TMH0005	Tim Halvorsen	18-Aug-1981
101	0101	1		Change NODE MAXIMUM LINES to MAXIMUM CIRCUITS.		
102	0102	1		Change NODE SERVICE LINE to SERVICE CIRCUIT.		
103	0103	1		Change NODE LINE (loop line) to CIRCUIT.		
104	0104	1		Add CLEAR EXECUTOR PIPELINE QUOTA.		
105	0105	1				
106	0106	1	V004	TMH0004	Tim Halvorsen	05-Aug-1981
107	0107	1		Change RETRANSMIT TIMER to a circuit parameter		
108	0108	1		from a line parameter.		
109	0109	1				
110	0110	1	V003	TMH0003	Tim Halvorsen	07-Jul-1981
111	0111	1		Rename circuit maximum blocks to maximum transmits.		
112	0112	1		Add line clock parameter.		
113	0113	1		Add dispatching for CLEAR X25-xxx.		
114	0114	1		Add node subaddresses parameter.		

:	115	0115	1	:				
:	116	0116	1	:		V002	TMH0002	Tim Halvorsen
:	117	0117	1	:			Add CLEAR CIRCUIT	11-Jun-1981
:	118	0118	1	:				
:	119	0119	1	:		V001	LMK0001	Len Kawell
:	120	0120	1	:			Remove HOST ID prompt from CLEAR EXECUTOR.	5-Jan-1981
:	121	0121	1	:	--			

```
123 0122 1 %SBTTL 'Definitions'
124 0123 1
125 0124 1
126 0125 1 INCLUDE FILES:
127 0126 1
128 0127 1
129 0128 1 LIBRARY 'LIB$NMALIBRY';
130 0129 1 LIBRARY 'LIB$NCPLIBRY';
131 0130 1 LIBRARY 'SYSS$LIBRARY:TPAMAC';
132 0131 1
133 0132 1
134 0133 1 EXTERNAL REFERENCES:
135 0134 1
136 0135 1
137 0136 1 EXTERNAL
138 0137 1 NCP$G_STTBL_CNO, ! Clear NODE
139 0138 1 NCP$G_KYTBL_CNO,
140 0139 1 NCP$G_STTBL_CEX, ! Clear EXECUTOR
141 0140 1 NCP$G_KYTBL_CEX,
142 0141 1 NCP$G_STTBL_CCF, ! Clear Configurator
143 0142 1 NCP$G_KYTBL_CCF,
144 0143 1 NCP$G_STTBL_CCS, ! Clear Console
145 0144 1 NCP$G_KYTBL_CCS,
146 0145 1 NCP$G_STTBL_CLD, ! Clear Loader
147 0146 1 NCP$G_KYTBL_CLD,
148 0147 1 NCP$G_STTBL_CLP, ! Clear Looper
149 0148 1 NCP$G_KYTBL_CLP,
150 0149 1 NCP$G_STTBL_CAC, ! Clear X25-ACCESS
151 0150 1 NCP$G_KYTBL_CAC,
152 0151 1 NCP$G_STTBL_CPR, ! Clear X25-PROTOCOL
153 0152 1 NCP$G_KYTBL_CPR,
154 0153 1 NCP$G_STTBL_CPRDTE, ! Clear X25-PROTOCOL DTE
155 0154 1 NCP$G_KYTBL_CPRDTE,
156 0155 1 NCP$G_STTBL_CPRGRP, ! Clear X25-PROTOCOL Group
157 0156 1 NCP$G_KYTBL_CPRGRP,
158 0157 1 NCP$G_STTBL_CSE, ! Clear X25-SERVER
159 0158 1 NCP$G_KYTBL_CSE,
160 0159 1 NCP$G_STTBL_CTR, ! Clear X25-TRACE
161 0160 1 NCP$G_KYTBL_CTR,
162 0161 1 NCP$G_STTBL_CTRTPT, ! Clear X25-TRACE Tracepoint
163 0162 1 NCP$G_KYTBL_CTRTPT,
164 0163 1 NCP$G_STTBL_C9S, ! Clear X29-SERVER
165 0164 1 NCP$G_KYTBL_C9S,
166 0165 1 PBK$G_CNO_ENT;
167 0166 1
168 0167 1 ACT_DFN ! External symbols for action routines
169 0168 1
170 0169 1 EXTERNAL ROUTINE
171 0170 1 ACT$NXT_STATE; ! Set vector to next state table
```



```

: 173      0171 1 %SBTTL 'Parameter blocks'
: 174      0172 1
: 175      0173 1
: 176      0174 1 BIND DATA:
: 177      0175 1
: 178      0176 1
: 179      0177 1
: 180      0178 1 Parameter Blocks
: 181      0179 1
: 182      0180 1
: 183      0181 1
: 184      0182 1 General blocks
: 185      0183 1
: 186      0184 1
: 187      P 0185 1 BUILD_PBK
: 188      P 0186 1
: 189      P 0187 1 (CLPU,
: 190      P 0188 1
: 191      P 0189 1 ALL, LITB, 0, VRB ALL,
: 192      P 0190 1 KWN, LITB, NMASC_ENT_KNO, VRB_ENT,
: 193      P 0191 1
: 194      0192 1 )

```

```
196 0193 1
197 0194 1
198 0195 1
199 0196 1
200 0197 1
201 P 0198 1
202 P 0199 1
203 P 0200 1
204 P 0201 1
205 P 0202 1
206 P 0203 1
207 P 0204 1
208 P 0205 1
209 P 0206 1
210 P 0207 1
211 P 0208 1
212 P 0209 1
213 P 0210 1
214 P 0211 1
215 P 0212 1
216 P 0213 1
217 P 0214 1
218 P 0215 1
219 P 0216 1
220 P 0217 1
221 P 0218 1
222 P 0219 1
223 P 0220 1
224 P 0221 1
225 P 0222 1
226 P 0223 1
227 P 0224 1
228 P 0225 1
229 P 0226 1
230 P 0227 1
231 P 0228 1
232 P 0229 1
233 P 0230 1
234 P 0231 1
235 P 0232 1
236 P 0233 1
237 P 0234 1
238 P 0235 1
239 P 0236 1
240 P 0237 1
241 P 0238 1
242 P 0239 1
243 P 0240 1
244 P 0241 1
245 0242 1
246 0243 1
247 P 0244 1
248 P 0245 1
249 P 0246 1
250 P 0247 1
251 P 0248 1
252 P 0249 1

Circuits

BUILD_PCL
(CCI,

STA, LITB, PCCI_STA,
SER, LITB, PCCI_SER,
CTM, LITB, PCCI_LCT,
COS, LITB, PCCI_COS,
MRT, LITB, PCCI_MRT,
RPR, LITB, PCCI_RPR,
HET, LITB, PCCI_HET,
LIT, LITB, PCCI_LIT,
BLK, LITB, PCCI_BLK,
MRC, LITB, PCCI_MRC,
RCT, LITB, PCCI_RCT,
NUM, LITB, PCCI_NUM,
POL, LITB, PCCI_POL,
OWN, LITB, PCCI_OWN,
LIN, LITB, PCCI_LIN,
USE, LITB, PCCI_USE,
TYP, LITB, PCCI_TYP,
DTE, LITB, PCCI_DTE,
CHN, LITB, PCCI_CHN,
MBL, LITB, PCCI_MBL,
MWI, LITB, PCCI_MWI,
TRI, LITB, PCCI_TRI,
BBT, LITB, PCCI_BBT,
TRT, LITB, PCCI_TRT,
MRB, LITB, PCCI_MRB,
MTR, LITB, PCCI_MTR,
ACB, LITB, PCCI_ACB,
ACI, LITB, PCCI_ACI,
IAB, LITB, PCCI_IAB,
IAI, LITB, PCCI_IAI,
IAT, LITB, PCCI_IAT,
DYB, LITB, PCCI_DYB,
DYI, LITB, PCCI_DYI,
DYT, LITB, PCCI_DYT,
DTH, LITB, PCCI_DTH,

VER, LITB, PCCI_VER,
XPT, LITB, PCCI_XPT,

, END, , ,
)

BUILD_PBK
(CCI,

ALL, LITB, 0, VRB ALL,
ENT, TKN, , VRB_ENT,
```



```

: 253 P 0250 1
: 254 P 0251 1 STA, LITB, 0, .
: 255 P 0252 1 SER, LITB, 0, .
: 256 P 0253 1 CTM, LITB, 0, .
: 257 P 0254 1 COS, LITB, 0, .
: 258 P 0255 1 MRT, LITB, 0, .
: 259 P 0256 1 RPR, LITB, 0, .
: 260 P 0257 1 HET, LITB, 0, .
: 261 P 0258 1 LIT, LITB, 0, .
: 262 P 0259 1 BLK, LITB, 0, .
: 263 P 0260 1 MRC, LITB, 0, .
: 264 P 0261 1 RCT, LITB, 0, .
: 265 P 0262 1 NUM, LITB, 0, .
: 266 P 0263 1 POL, LITB, 0, .
: 267 P 0264 1 OWN, LITB, 0, .
: 268 P 0265 1 LIN, LITB, 0, .
: 269 P 0266 1 USE, LITB, 0, .
: 270 P 0267 1 TYP, LITB, 0, .
: 271 P 0268 1 DTE, LITB, 0, .
: 272 P 0269 1 CHN, LITB, 0, .
: 273 P 0270 1 MBL, LITB, 0, .
: 274 P 0271 1 MWI, LITB, 0, .
: 275 P 0272 1 TRI, LITB, 0, .
: 276 P 0273 1 BBT, LITB, 0, .
: 277 P 0274 1 TRT, LITB, 0, .
: 278 P 0275 1 MRB, LITB, 0, .
: 279 P 0276 1 MTR, LITB, 0, .
: 280 P 0277 1 ACB, LITB, 0, .
: 281 P 0278 1 ACI, LITB, 0, .
: 282 P 0279 1 IAB, LITB, 0, .
: 283 P 0280 1 IAI, LITB, 0, .
: 284 P 0281 1 IAT, LITB, 0, .
: 285 P 0282 1 DYB, LITB, 0, .
: 286 P 0283 1 DYI, LITB, 0, .
: 287 P 0284 1 DYT, LITB, 0, .
: 288 P 0285 1 DTH, LITB, 0, .
: 289 P 0286 1
: 290 P 0287 1 VER, LITB, 0, .
: 291 P 0288 1 XPT, LITB, 0, .
: 292 P 0289 1
: 293 0290 1 )
: 294 0291 1
: 295 P 0292 1 BUILD_SDB
: 296 P 0293 1
: 297 0294 1 (CCI, NMASC_ENT_CIR, VRB_ENT, CCI)

```

```
299 0295 1  
300 0296 1  
301 0297 1  
302 0298 1  
303 0299 1  
304 P 0300 1  
305 P 0301 1  
306 P 0302 1  
307 P 0303 1  
308 P 0304 1  
309 P 0305 1  
310 P 0306 1  
311 P 0307 1  
312 P 0308 1  
313 P 0309 1  
314 P 0310 1  
315 P 0311 1  
316 P 0312 1  
317 P 0313 1  
318 P 0314 1  
319 P 0315 1  
320 P 0316 1  
321 P 0317 1  
322 P 0318 1  
323 P 0319 1  
324 P 0320 1  
325 P 0321 1  
326 P 0322 1  
327 P 0323 1  
328 P 0324 1  
329 P 0325 1  
330 P 0326 1  
331 P 0327 1  
332 P 0328 1  
333 P 0329 1  
334 P 0330 1  
335 P 0331 1  
336 P 0332 1  
337 P 0333 1  
338 P 0334 1  
339 P 0335 1  
340 P 0336 1  
341 P 0337 1  
342 P 0338 1  
343 P 0339 1  
344 P 0340 1  
345 P 0341 1  
346 P 0342 1  
347 P 0343 1  
348 P 0344 1  
349 P 0345 1  
350 P 0346 1  
351 P 0347 1  
352 P 0348 1  
353 P 0349 1  
354 P 0350 1  
355 P 0351 1
```

Lines

BUILD_PCL

(CLI,

STA, LITB, PCLI_STA, ,
SVM, LITB, PCLI_SER, ,
CTM, LITB, PCLI_LCT, ,
COS, LITB, PCLI_COS, ,
DEV, LITB, PCLI_DEV, ,
PRO, LITB, PCLI_PRO, ,
DPX, LITB, PCLI_DUP, ,
CON, LITB, PCLI_CON, ,
CLO, LITB, PCLI_CLO, ,
TYP, LITB, PCLI_LTY, ,
STM, LITB, PCLI_STI, ,
NTM, LITB, PCLI_NTI, ,
HTI, LITB, PCLI_HTI, ,
MBL, LITB, PCLI_MBL, ,
MRT, LITB, PCLI_MRT, ,
MWI, LITB, PCLI_MWI, ,
TRB, LITB, PCLI_TRI, ,
SLT, LITB, PCLI_SLT, ,
DDT, LITB, PCLI_DDT, ,
DLT, LITB, PCLI_DLT, ,
SRT, LITB, PCLI_SRT, ,
BFN, LITB, PCLI_BFN, ,

MCD, LITB, PCLI_MCD, ,
XMD, LITB, PCLI_XMD, ,

, END, , ,
)

BUILD_PBK

(CLI,

ALL, LITB, 0, VRB_ALL,
ENT, TKN, , VRB_ENT,

STA, LITB, 0, ,
SVM, LITB, 0, ,
CTM, LITB, 0, ,
COS, LITB, 0, ,
DEV, LITB, 0, ,
PRO, LITB, 0, ,
DPX, LITB, 0, ,
CON, LITB, 0, ,
CLO, LITB, 0, ,
TYP, LITB, 0, ,
STM, LITB, 0, ,


```
.. 356 P 0352 1 NTM, LITB, 0, .  
.. 357 P 0353 1 HTI, LITB, 0, .  
.. 358 P 0354 1 MBL, LITB, 0, .  
.. 359 P 0355 1 MRT, LITB, 0, .  
.. 360 P 0356 1 MWI, LITB, 0, .  
.. 361 P 0357 1 TRB, LITB, 0, .  
.. 362 P 0358 1 SLT, LITB, 0, .  
.. 363 P 0359 1 DDT, LITB, 0, .  
.. 364 P 0360 1 DLT, LITB, 0, .  
.. 365 P 0361 1 SRT, LITB, 0, .  
.. 366 P 0362 1 BFN, LITB, 0, .  
.. 367 P 0363 1  
.. 368 P 0364 1 MCD, LITB, 0, .  
.. 369 P 0365 1 XMD, LITB, 0, .  
.. 370 P 0366 1  
.. 371 P 0367 1 )  
.. 372 P 0368 1  
.. 373 P 0369 1 BUILD_SDB  
.. 374 P 0370 1  
.. 375 P 0371 1 (CLI, NMASC_ENT_LIN, VRB_ENT, CLI)
```

```
.. 377      0372      1
.. 378      0373      1
.. 379      0374      1
.. 380      0375      1
.. 381      0376      1
.. 382      P 0377      1
.. 383      P 0378      1
.. 384      P 0379      1
.. 385      P 0380      1
.. 386      P 0381      1
.. 387      P 0382      1
.. 388      P 0383      1
.. 389      P 0384      1
.. 390      P 0385      1
.. 391      P 0386      1
.. 392      P 0387      1
.. 393      P 0388      1
.. 394      P 0389      1
.. 395      P 0390      1
.. 396      P 0391      1
.. 397      P 0392      1
.. 398      P 0393      1
.. 399      P 0394      1
.. 400      P 0395      1
.. 401      P 0396      1
.. 402      P 0397      1
.. 403      P 0398      1
.. 404      P 0399      1
.. 405      P 0400      1
.. 406      P 0401      1
.. 407      P 0402      1
.. 408      P 0403      1
.. 409      P 0404      1
.. 410      P 0405      1
.. 411      P 0406      1
.. 412      P 0407      1
.. 413      0408      1

Logging

BUILD_PCL
(CLO,
NAM, LITB, PCLO_LNA, ,
SNO, NADR, PCLO_SIN, ,
EVL, ESET, PCLO_EVE, VRB_EVE,
, END, , ,
)

BUILD_PBK
(CLO,
ALL, LITB, 0, VRB_ALL,
EVK, LITB, 0, CLO_EVL,
KNE, LITB, 0, CLO_EVL,
EVL, LITB, 0,
SEX, LITL, 0, CLO_SNO,
SNO, NADR, 0,
NAM, LITB, 0,
CIR, ESCI, , VRB_EVE,
LIN, ESLI, , VRB_EVE,
NOD, ESNO, , VRB_EVE,
)

BUILD_SDB
(CLO, NMASC_ENT_LOG, VRB_ENT, CLO)
```



```

: 415      0409 1
: 416      0410 1
: 417      0411 1
: 418      0412 1
: 419      0413 1
: 420      P 0414 1
: 421      P 0415 1
: 422      P 0416 1
: 423      P 0417 1
: 424      P 0418 1
: 425      P 0419 1
: 426      P 0420 1
: 427      P 0421 1
: 428      P 0422 1
: 429      P 0423 1
: 430      P 0424 1
: 431      P 0425 1
: 432      P 0426 1
: 433      P 0427 1
: 434      P 0428 1
: 435      P 0429 1
: 436      P 0430 1
: 437      P 0431 1
: 438      P 0432 1
: 439      P 0433 1
: 440      P 0434 1
: 441      P 0435 1
: 442      P 0436 1
: 443      P 0437 1
: 444      P 0438 1
: 445      P 0439 1
: 446      P 0440 1
: 447      P 0441 1
: 448      P 0442 1
: 449      P 0443 1
: 450      P 0444 1
: 451      P 0445 1
: 452      P 0446 1
: 453      P 0447 1
: 454      P 0448 1
: 455      P 0449 1

Objects

BUILD_PCL
(COB,
  NUM, LITB,      PCOB_NUM,
  FIL, LITB,      PCOB_FID,
  PRV, LITB,      PCOB_PRV,
  USR, LITB,      PCOB_USR,
  ACC, LITB,      PCOB_ACC,
  PSW, LITB,      PCOB_PSW,
  PRX, LITB,      PCOB_PRX,
  , END, , ,
)

BUILD_PBK
(COB,
  ENT, TKN, , VRB_ENT,
  ALL, LITB, 0, VRB_ALL,
  NUM, LITB, 0,
  FIL, LITB, 0,
  PRV, LITB, 0,
  USR, LITB, 0,
  ACC, LITB, 0,
  PSW, LITB, 0,
  PRX, LITB, 0,
)

BUILD_SDB
(COB, -NMA$C_SENT_OBJ, VRB_ENT, COB)
```

```
457 0450 1 %SBTTL 'Prompt strings'
458 0451 1
459 0452 1
460 0453 1 Build prompt strings
461 0454 1
462 0455 1
463 0456 1 BIND
464 0457 1
465 P 0458 1 PROMPT_STRINGS
466 P 0459 1 (CLPU,
467 P 0460 1
468 L 0461 1 ENT, %STRING(' (CIRCUIT, EXECUTOR, KNOWN, LINE, ', CRLF,
469 P 0462 1 ' LOGGING, MODULE, NODE, OBJECT): ',
470 P 0463 1 ' MOD, %STRING(' ( X25-ACCESS, X25-PROTOCOL, X25-SERVER, ', CRLF,
471 P 0464 1 ' X25-TRACE, X29-SERVER): ',
472 L 0465 1 MOD, %STRING('Module (CONFIGURATOR, CONSOLE, LOADER, ', CRLF,
473 L 0466 1 ' LOOPER, X25-ACCESS, X25-PROTOCOL, ', CRLF,
474 P 0467 1 ' X25-SERVER, X25-TRACE, X29-SERVER): ',
475 P 0468 1 KWN, ' (LINES, LOGGING, NODES, OBJECTS): ',
476 P 0469 1
477 0470 2 )
478 0471 1 ,
479 0472 1
480 0473 1
481 P 0474 1 PROMPT_STRINGS
482 P 0475 1 (CNQ,
483 P 0476 1
484 P 0477 1 ENT, 'Node ID (node-name, node-address): ',
485 P 0478 1
486 0479 1 ),
487 0480 1
488 0481 1
489 P 0482 1 PROMPT_STRINGS
490 P 0483 1 (CLI,
491 P 0484 1
492 P 0485 1 ALL, 'All line parameters (Y, N): ',
493 P 0486 1 ENT, 'Line ID (dev-c-u.t): ',
494 P 0487 1 CTM, 'Counter timer (Y, N): ',
495 P 0488 1 DDT, 'Dead timer (Y, N): ',
496 P 0489 1 DLT, 'Delay timer (Y, N): ',
497 P 0490 1 HTI, 'Holdback timer (Y, N): ',
498 P 0491 1 MRT, 'Maximum retransmits (Y, N): ',
499 P 0492 1 SLT, 'Scheduling timer (Y, N): ',
500 P 0493 1 SRT, 'Stream timer (Y, N): ',
501 P 0494 1
502 0495 2 )
503 0496 1 ,
504 0497 1
505 P 0498 1 PROMPT_STRINGS
506 P 0499 1 (CCI,
507 P 0500 1
508 P 0501 1 ALL, 'All circuit parameters (Y, N): ',
509 P 0502 1 ENT, 'Circuit ID (16 characters): ',
510 P 0503 1 BBT, 'Babble timer (Y, N): ',
511 P 0504 1 CTM, 'Counter timer (Y, N): ',
512 P 0505 1 MRB, 'Maximum receive buffers (Y, N): ',
513 P 0506 1 MRC, 'Maximum recalls (Y, N): ',
```



```
514 P 0507 1 !! OWN, 'Owner (Y, N): ',
515 P 0508 1 !!! RCT, 'Recall timer (Y, N): ',
516 P 0509 1 !!! TRT, 'Transmit timer (Y, N): ',
517 P 0510 1 !!! ACB, 'Active base (Y, N): ',
518 P 0511 1 !!! ACI, 'Active increment (Y, N): ',
519 P 0512 1 !!! IAB, 'Inactive base (Y, N): ',
520 P 0513 1 !!! IAI, 'Inactive increment (Y, N): ',
521 P 0514 1 !!! IAT, 'Inactive threshold (Y, N): ',
522 P 0515 1 !!! DYB, 'Dying base (Y, N): ',
523 P 0516 1 !!! DYI, 'Dying increment (Y, N): ',
524 P 0517 1 !!! DYT, 'Dying threshold (Y, N): ',
525 P 0518 1 !!! DTH, 'Dead threshold (Y, N): ',
526 P 0519 1 )
527 P 0520 2 )
528 P 0521 1 ,
529 P 0522 1
530 P 0523 1 PROMPT_STRINGS
531 P 0524 1 (CLO,
532 P 0525 1
533 P 0526 1 ALL, 'All logging of the specified type (Y, N): ',
534 P 0527 1 ENT, 'Type of logging (CONSOLE, FILE, MONITOR): ',
535 P 0528 1 EVK, 'Event list (cls.typ-typ, typ, KNOWN): ',
536 P 0529 1 NAM, 'Name of sink file (Y, N): ',
537 L 0530 1 SRC, %STRING('Source of events (LINE line-id, ', CRLF,
538 P 0531 1 'CIRCUIT circ-id, NODE node-id): '),
539 P 0532 1 SNO, 'Sink node (node-id, EXECUTOR): ',
540 P 0533 1 )
541 P 0534 2 )
542 P 0535 1 ;
```

```

: 544      0536 1
: 545      0537 1 BIND
: 546      0538 1
: 547      P 0539 1
: 548      P 0540 1
: 549      P 0541 1
: 550      P 0542 1
: 551      P 0543 1
: 552      P 0544 1
: 553      P 0545 1
: 554      P 0546 1
: 555      P 0547 1
: 556      P 0548 1
: 557      P 0549 1
: 558      P 0550 1
: 559      P 0551 1
: 560      0552 2
: 561      0553 1

PROMPT_STRINGS
(COB,
ALL, 'All parameters      (Y, N): ',
ENT, 'Object name (object-name): ',
FIL, 'File name          (Y, N): ',
NUM, 'Object number      (Y, N): ',
PRV, 'Privileges         (Y, N): ',
USR, 'User ID            (Y, N): ',
PSW, 'Password           (Y, N): ',
ACC, 'Account            (Y, N): ',
PRX, 'Proxy access       (Y, N): ',
)
;
```



```
.. 563      0554 1 %SBTTL 'State Table Entry'
.. 564      0555 1
.. 565      0556 1 $INIT_STATE (NCP$G_STTBL_CLPU, NCP$G_KYTBL_CLPU);
.. 566      0557 1
.. 567      0558 1
.. 568      0559 1
.. 569      0560 1
.. 570      0561 1
.. 571      0562 1
.. 572      0563 1
.. 573      0564 1
.. 574      0565 1
.. 575      P 0566 1
.. 576      P 0567 1
.. 577      P 0568 1
.. 578      P 0569 1
.. 579      P 0570 1
.. 580      P 0571 1
.. 581      P 0572 1
.. 582      P 0573 1
.. 583      P 0574 1
.. 584      P 0575 1
.. 585      P 0576 1
.. 586      P 0577 1
.. 587      P 0578 1
.. 588      P 0579 1
.. 589      P 0580 1
.. 590      P 0581 1
.. 591      P 0582 1
.. 592      P 0583 1
.. 593      P 0584 1
.. 594      P 0585 1
.. 595      P 0586 1
.. 596      0587 1

      !
      CLEAR/PURGE Commands

      !
      Dispatch to entity types and prompt if end of string

      !
      COMMAND PROMPT
      (CLPU, ENT, NCP$_INVKEY,
      ('CIRCUIT', ST_CCI_ENT, , NMA$C_ENT_CIR, NCP$GL_OPTION, ),
      ('CONFIGURATOR', ST_CLPO_CNF),
      ('CONSOLE', ST_CLPU_CNS),
      ('DTE', ST_CLPU_CPRDTE),
      ('EXECUTOR', TPAS_EXIT, ACT$NXT_STATE,
      NMA$C_ENT_NOD, NCP$GL_OPTION, NEXT_STATE(CEX)),
      ('GROUP', ST_CLPU_CPRGRP),
      ('LINE', ST_CLI_ENT, , NMA$C_ENT_LIN, NCP$GL_OPTION, ),
      ('LOADER', ST_CLPO_LOA),
      ('LOOPER', ST_CLPU_LOO),
      ('LOGGING', ST_CLO_ENT, , NMA$C_ENT_LOG, NCP$GL_OPTION, ),
      ('KNOWN', ST_CLPO_KWN, ACT$SAVPRM, , , PBK$G_CLPU_KWN),
      ('MODULE', ST_CLPU_MOD),
      ('NODE', ST_CNO_ENT, , NMA$C_ENT_NOD, NCP$GL_OPTION, ),
      ('OBJECT', ST_COB_ENT, , NMA$C_SENT_OBJ, NCP$GL_OPTION, ),
      ('TRACEPOINT', ST_CLPO_CTRTPT),
      ('X25', ST_CLPU_X25),
      ('X29', ST_CLPU_X29)
      )
```



```

: 598      0588 1 %SBTTL 'Clear/Purge Known'
: 599      0589 1
: 600      0590 1
: 601      0591 1
: 602      0592 1
: 603      0593 1
: 604      0594 1
: 605      0595 1
: 606      0596 1
: 607      0597 1
: 608      P 0598 1
: 609      P 0599 1
: 610      P 0600 1
: 611      P 0601 1
: 612      P 0602 1
: 613      P 0603 1
: 614      P 0604 1
: 615      P 0605 1
: 616      P 0606 1
: 617      P 0607 1
: 618      0608 1

Clear purge known

Dispatch for the entity types, prompt if none

COMMAND PROMPT
(CLPU, RWN, NCP$_INVKEY,

('CIRCUITS', ST_CCI_GO, , NMA$C_ENT_CIR, NCP$GL_OPTION, ),
('LINES', ST_CLI_GO, , NMA$C_ENT_LIN, NCP$GL_OPTION, ),
('LOGGING', ST_CLO_GO, , NMA$C_ENT_LOG, NCP$GL_OPTION, ),
('NODES', IPAS_EXIT, ACT$NXT_STATE,
NMA$C_ENT_NOD, NCP$GL_OPTION, NEXT_STATE(CNO)),
('OBJECTS', ST_COB_GO, , NMA$C_SENT_OBJ, NCP$GL_OPTION, )
)
```

```

: 620      0609 1 %SBTTL 'Clear/Purge Module'
: 621      0610 1
: 622      0611 1
: 623      0612 1
: 624      0613 1
: 625      0614 1
: 626      P 0615 1
: 627      P 0616 1
: 628      P 0617 1
: 629      P 0618 1
: 630      P 0619 1
: 631      P 0620 1
: 632      P 0621 1
: 633      P 0622 1
: 634      P 0623 1
: 635      0624 1
: 636      0625 1
: 637      P 0626 1
: 638      0627 1
: 639      P 0628 1
: 640      P 0629 1
: 641      P 0630 1
: 642      P 0631 1
: 643      P 0632 1
: 644      0633 1
: 645      0634 1
: 646      P 0635 1
: 647      0636 1
: 648      P 0637 1
: 649      P 0638 1
: 650      0639 1
: 651      0640 1
: 652      0641 1
: 653      P 0642 1
: 654      P 0643 1
: 655      P 0644 1
: 656      0645 1
: 657      0646 1
: 658      P 0647 1
: 659      P 0648 1
: 660      P 0649 1
: 661      0650 1
: 662      0651 1
: 663      P 0652 1
: 664      P 0653 1
: 665      P 0654 1
: 666      0655 1
: 667      0656 1
: 668      P 0657 1
: 669      P 0658 1
: 670      P 0659 1
: 671      0660 1
: 672      0661 1
: 673      P 0662 1
: 674      P 0663 1
: 675      P 0664 1
: 676      0665 1

      1 %SBTTL 'Clear/Purge Module'

      Modules

      COMMAND PROMPT
      (CLPU, MOD, NCP$INVKEY,

      ('CONFIGURATOR', ST_CLPU_CNF),
      ('CONSOLE', ST_CLPU_CNS),
      ('LOADER', ST_CLPU_LOA),
      ('LOOPER', ST_CLPU_LOO),
      ('X25', ST_CLPU_X25),
      ('X29', ST_CLPU_X29)
      )

      $STATE (ST_CLPU_X25,
      ('-T'));

      $STATE (
      ('ACCESS', ST_CLPU_CAC),
      ('PROTOCOL', ST_CLPU_CPR),
      ('SERVER', ST_CLPU_CSE),
      ('TRACE', ST_CLPU_CTR)
      );

      $STATE (ST_CLPU_X29,
      ('-T'));

      $STATE (
      ('SERVER', ST_CLPU_C9S)
      );

      $STATE (ST_CLPU_CNF,
      (TPAS_LAMBDA, TPAS_EXIT, ACT$NXT_STATE,
      NMASC_ENT_MOD, NCP$GL_OPTION, NEXT_STATE(CCF))
      );
      ! Module Configurator

      $STATE (ST_CLPU_CNS,
      (TPAS_LAMBDA, TPAS_EXIT, ACT$NXT_STATE,
      NMASC_ENT_MOD, NCP$GL_OPTION, NEXT_STATE(CCS))
      );
      ! Module Console

      $STATE (ST_CLPU_LOA,
      (TPAS_LAMBDA, TPAS_EXIT, ACT$NXT_STATE,
      NMASC_ENT_MOD, NCP$GL_OPTION, NEXT_STATE(CLD))
      );
      ! Module Loader

      $STATE (ST_CLPU_LOO,
      (TPAS_LAMBDA, TPAS_EXIT, ACT$NXT_STATE,
      NMASC_ENT_MOD, NCP$GL_OPTION, NEXT_STATE(CLP))
      );
      ! Module looper

      $STATE (ST_CLPU_CAC,
      (TPAS_LAMBDA, TPAS_EXIT, ACT$NXT_STATE,
      NMASC_ENT_MOD, NCP$GL_OPTION, NEXT_STATE(CAC))
      );
      ! X25-Access
```



```

: 677      0666 1
: 678      P 0667 1 $STATE (ST_CLPU CPR,                ! X25-Protocol
: 679      P 0668 1      (TPAS_LAMBDA, TPAS_EXIT, ACT$NXT_STATE,
: 680      P 0669 1      NMAC$ENT_MOD, NCP$GL_OPTION, NEXT_STATE(CPR))
: 681      0670 1      );
: 682      P 0671 1 $STATE (ST_CLPU CPRDTE,                ! X25-Protocol DTE
: 683      P 0672 1      (TPAS_LAMBDA, TPAS_EXIT, ACT$NXT_STATE,
: 684      P 0673 1      NMAC$ENT_MOD, NCP$GL_OPTION, NEXT_STATE(CPRDTE))
: 685      0674 1      );
: 686      P 0675 1 $STATE (ST_CLPU CPRGRP,                ! X25-Protocol Group
: 687      P 0676 1      (TPAS_LAMBDA, TPAS_EXIT, ACT$NXT_STATE,
: 688      P 0677 1      NMAC$ENT_MOD, NCP$GL_OPTION, NEXT_STATE(CPRGRP))
: 689      0678 1      );
: 690      0679 1
: 691      P 0680 1 $STATE (ST_CLPU CSE,
: 692      P 0681 1      (TPAS_LAMBDA, TPAS_EXIT, ACT$NXT_STATE,
: 693      P 0682 1      NMAC$ENT_MOD, NCP$GL_OPTION, NEXT_STATE(CSE))
: 694      0683 1      );
: 695      0684 1
: 696      P 0685 1 $STATE (ST_CLPU CTR,                ! X25-Trace
: 697      P 0686 1      (TPAS_LAMBDA, TPAS_EXIT, ACT$NXT_STATE,
: 698      P 0687 1      NMAC$ENT_MOD, NCP$GL_OPTION, NEXT_STATE(CTR))
: 699      0688 1      );
: 700      P 0689 1 $STATE (ST_CLPU CTRTPT,                ! X25-Trace Tracepoint
: 701      P 0690 1      (TPAS_LAMBDA, TPAS_EXIT, ACT$NXT_STATE,
: 702      P 0691 1      NMAC$ENT_MOD, NCP$GL_OPTION, NEXT_STATE(CTRTPT))
: 703      0692 1      );
: 704      0693 1
: 705      P 0694 1 $STATE (ST_CLPU C9S,                ! X29-Server
: 706      P 0695 1      (TPAS_LAMBDA, TPAS_EXIT, ACT$NXT_STATE,
: 707      P 0696 1      NMAC$ENT_MOD, NCP$GL_OPTION, NEXT_STATE(C9S))
: 708      0697 1      );
```

```
: 710      0698 1 %SBTTL 'Clear/Purge Node'
: 711      0699 1
: 712      0700 1
: 713      0701 1
: 714      0702 1
: 715      0703 1
: 716      0704 1
: 717      0705 1
: 718      0706 1
: 719      0707 1
: 720      P 0708 1
: 721      P 0709 1
: 722      P 0710 1
: 723      P 0711 1
: 724      P 0712 1
: 725      0713 1
: 726      0714 1
: 727      P 0715 1
: 728      P 0716 1
: 729      P 0717 1
: 730      0718 1
: 731      0719 1

      Clear purge Node

      Collect the node id or prompt

      COMMAND PROMPT
      (CNO, ENT, NCP$_INVVAL,
      ( (SE_NODE_ID), , ACT$SAVPRM, , , PBK$G_CNO_ENT)
      )
      $STATE (
      (TPAS_LAMBDA, TPAS_EXIT, ACT$NXT_STATE,
      NMAC_ENT_NOD, NCP$GL_OPTION, NEXT_STATE(CNO))
      );
```



```
: 733      0720 1 %SBTTL 'Clear/Purge Circuits'
: 734      0721 1
: 735      0722 1
: 736      0723 1
: 737      0724 1
: 738      0725 1
: 739      0726 1
: 740      0727 1
: 741      0728 1
: 742      0729 1
: 743      P 0730 1 COMMAND_PROMPT
: 744      P 0731 1 (CCI, ENT, NCP$_INVVAL,
: 745      P 0732 1
: 746      P 0733 1 ( (SE_CIRC_ID), , ACT$SAVPRM, , , PBK$G_CCI_ENT)
: 747      P 0734 1
: 748      0735 1
: 749      0736 1
: 750      0737 1
: 751      0738 1
: 752      0739 1
: 753      0740 1
: 754      0741 1
: 755      P 0742 1 $STATE (ST_CCI_GO,
: 756      P 0743 1 (TPAS_EOS),
: 757      P 0744 1 (TPAS_LAMBDA, ST_CCI_PRC)
: 758      0745 1 );
: 759      0746 1
: 760      P 0747 1 QUERY_STATES ! Build states to query for params
: 761      P 0748 1 (CCI,
: 762      P 0749 1
: 763      P 0750 1 ALL
: 764      P 0751 1
: 765      0752 1
: 766      0753 1
: 767      0754 1
: 768      0755 1
: 769      0756 1
: 770      0757 1
: 771      P 0758 1 $STATE (ST_CCI_DOIT, ! Set the option mask
: 772      P 0759 1 (TPAS_LAMBDA, , , NMASC_ENT_CIR, NCP$GL_OPTION, )
: 773      0760 1 );
: 774      0761 1
: 775      P 0762 1 $STATE (, ! Perform the function
: 776      P 0763 1 (TPAS_EOS, TPAS_EXIT, ACT$VRB_UTILITY, , , SDB$G_CCI)
: 777      0764 1 );
```

```

: 779      0765 1
: 780      0766 1
: 781      0767 1
: 782      0768 1
: 783      0769 1
: 784      P 0770 1
: 785      P 0771 1
: 786      P 0772 1
: 787      P 0773 1
: 788      P 0774 1
: 789      P 0775 1
: 790      P 0776 1
: 791      P 0777 1
: 792      P 0778 1
: 793      P 0779 1
: 794      P 0780 1
: 795      P 0781 1
: 796      P 0782 1
: 797      P 0783 1
: 798      P 0784 1
: 799      P 0785 1
: 800      P 0786 1
: 801      P 0787 1
: 802      P 0788 1
: 803      P 0789 1
: 804      P 0790 1
: 805      P 0791 1
: 806      P 0792 1
: 807      P 0793 1
: 808      P 0794 1
: 809      P 0795 1
: 810      P 0796 1
: 811      P 0797 1
: 812      P 0798 1
: 813      P 0799 1
: 814      P 0800 1
: 815      P 0801 1
: 816      P 0802 1
: 817      P 0803 1
: 818      P 0804 1
: 819      P 0805 1
: 820      P 0806 1
: 821      0807 1

          :
          :
          : Dispatch for the parameters
          :
          :
          : $STATE (ST_CCI_PRC,
          :
          : DISPATCH_STATES
          : (CCI,
          :
          : ACT, 'ACTIVE',
          : ALL, 'ALL',
          : BBT, 'BABBLE',
          : BLK, 'BLOCKING',
          : CHN, 'CHANNEL',
          : COS, 'COST',
          : CTM, 'COUNTER',
          : DTH, 'DEAD',
          : DTE, 'DTE',
          : DYE, 'DYING',
          : HET, 'HELLO',
          : IAC, 'INACTIVE',
          : LIN, 'LINE',
          : LIT, 'LISTEN',
          : MAX, 'MAXIMUM',
          : NUM, 'NUMBER',
          : OWN, 'OWNER',
          : POL, 'POLLING',
          : RCT, 'RECALL',
          : RPR, 'ROUTER',
          : SER, 'SERVICE',
          : STA, 'STATE',
          : TRT, 'TRANSMIT',
          : XPT, 'TRANSPORT',
          : TRI, 'TRIBUTARY',
          : TYP, 'TYPE',
          : USE, 'USAGE',
          : VER, 'VERIFICATION',
          :
          : )
          :
          : (TPAS_EOS, ST_CCI_DOIT)
          : ;
```



```

: 823      0808 1
: 824      0809 1
: 825      0810 1
: 826      0811 1
: 827      0812 1
: 828      0813 1
: 829      P 0814 1
: 830      P 0815 1
: 831      P 0816 1
: 832      P 0817 1
: 833      P 0818 1
: 834      P 0819 1
: 835      P 0820 1
: 836      P 0821 1
: 837      P 0822 1
: 838      P 0823 1
: 839      P 0824 1
: 840      P 0825 1
: 841      P 0826 1
: 842      P 0827 1
: 843      P 0828 1
: 844      P 0829 1
: 845      P 0830 1
: 846      P 0831 1
: 847      P 0832 1
: 848      P 0833 1
: 849      P 0834 1
: 850      P 0835 1
: 851      P 0836 1
: 852      P 0837 1
: 853      P 0838 1
: 854      P 0839 1
: 855      P 0840 1
: 856      P 0841 1
: 857      P 0842 1
: 858      P 0843 1
: 859      P 0844 1
: 860      P 0845 1
: 861      P 0846 1
: 862      P 0847 1
: 863      P 0848 1
: 864      P 0849 1
: 865      P 0850 1
: 866      P 0851 1
: 867      P 0852 1
: 868      P 0853 1
: 869      P 0854 1
: 870      P 0855 1
: 871      P 0856 1
: 872      0857 1

      These states take care of noise words
      and call the subexpressions to do the work

      PROCESS_STATES
      (CCI,

      ALL, ,
      STA, ,
      SER, ,
      CTM, 'TIMER',
      COS, ,
      MRT, , ! From MAX below
      RPR, 'PRIORITY',
      HET, 'TIMER',
      LIT, 'TIMER',
      BLK, ,
      MRC, , ! From MAX below
      RCT, 'TIMER',
      NUM, ,
      POL, 'STATE',
      OWN, ,
      LIN, ,
      USE, ,
      TYP, ,
      DTE, ,
      CHN, ,
      MBL, , ! From MAX below
      MWI, , ! From MAX below
      TRI, ,
      BBT, 'TIMER',
      TRT, 'TIMER',
      MRB, , ! From MAX below
      MTR, , ! From MAX below
      ACB, , ! From ACT below
      ACI, , ! From ACT below
      IAB, , ! From IAC below
      IAI, , ! From IAC below
      IAT, , ! From IAC below
      DYB, , ! From DYE below
      DYI, , ! From DYE below
      DYT, , ! From DYE below
      DTH, 'THRESHOLD',
      VER, ,
      XPT, 'TYPE',
      )
```

```

: 874      0858 1
: 875      0859 1
: 876      0860 1
: 877      0861 1
: 878      0862 1
: 879      P 0863 1
: 880      P 0864 1
: 881      P 0865 1
: 882      P 0866 1
: 883      P 0867 1
: 884      P 0868 1
: 885      P 0869 1
: 886      P 0870 1
: 887      P 0871 1
: 888      P 0872 1
: 889      P 0873 1
: 890      P 0874 1
: 891      P 0875 1
: 892      P 0876 1
: 893      P 0877 1
: 894      P 0878 1
: 895      P 0879 1
: 896      P 0880 1
: 897      P 0881 1
: 898      P 0882 1
: 899      P 0883 1
: 900      P 0884 1
: 901      P 0885 1
: 902      P 0886 1
: 903      P 0887 1
: 904      P 0888 1
: 905      P 0889 1
: 906      P 0890 1
: 907      P 0891 1
: 908      P 0892 1
: 909      P 0893 1
: 910      P 0894 1
: 911      P 0895 1
: 912      P 0896 1
: 913      P 0897 1
: 914      P 0898 1
: 915      P 0899 1
: 916      P 0900 1
: 917      P 0901 1
: 918      P 0902 1
: 919      P 0903 1
: 920      P 0904 1
: 921      P 0905 1
: 922      P 0906 1
: 923      0907 1

      These subexpressions store away the parameter id's

      SUB_EXPRESSIONS
      (CCI,

      ALL, TPAS_EOS,

      STA, TPAS_LAMBDA,
      SER, TPAS_LAMBDA,
      CTM, TPAS_LAMBDA,
      COS, TPAS_LAMBDA,
      MRT, TPAS_LAMBDA,
      RPR, TPAS_LAMBDA,
      HET, TPAS_LAMBDA,
      LIT, TPAS_LAMBDA,
      BLK, TPAS_LAMBDA,
      MRC, TPAS_LAMBDA,
      RCT, TPAS_LAMBDA,
      NUM, TPAS_LAMBDA,
      POL, TPAS_LAMBDA,
      OWN, TPAS_LAMBDA,
      LIN, TPAS_LAMBDA,
      USE, TPAS_LAMBDA,
      TYP, TPAS_LAMBDA,
      DTE, TPAS_LAMBDA,
      CHN, TPAS_LAMBDA,
      MBL, TPAS_LAMBDA,
      MWI, TPAS_LAMBDA,
      TRI, TPAS_LAMBDA,
      BBT, TPAS_LAMBDA,
      TRT, TPAS_LAMBDA,
      MRB, TPAS_LAMBDA,
      MTR, TPAS_LAMBDA,
      ACB, TPAS_LAMBDA,
      ACI, TPAS_LAMBDA,
      IAB, TPAS_LAMBDA,
      IAI, TPAS_LAMBDA,
      IAT, TPAS_LAMBDA,
      DYB, TPAS_LAMBDA,
      DYI, TPAS_LAMBDA,
      DYT, TPAS_LAMBDA,
      DTH, TPAS_LAMBDA,

      VER, TPAS_LAMBDA,
      XPT, TPAS_LAMBDA,

      )
```

```

925      0908 1
926      0909 1
927      0910 1      Dispatch on MAXIMUM keyword
928      0911 1
929      0912 1
930      P 0913 1 $STATE (ST_CCI_PRC_MAX,
931      P 0914 1
932      P 0915 1      DISPATCH_STATES
933      P 0916 1      (CCI,
934      P 0917 1
935      P 0918 1      MBL, 'DATA',
936      P 0919 1      MRC, 'RECALLS',
937      P 0920 1      MRT, 'ROUTERS',
938      P 0921 1      MTR, 'TRANSMITS',
939      P 0922 1      MWI, 'WINDOW',
940      P 0923 1
941      P 0924 1      )
942      0925 1      );
943      0926 1
944      0927 1
945      0928 1      Dispatch on ACTIVE keyword
946      0929 1
947      0930 1
948      P 0931 1 $STATE (ST_CCI_PRC_ACT,
949      P 0932 1
950      P 0933 1      DISPATCH_STATES
951      P 0934 1      (CCI,
952      P 0935 1
953      P 0936 1      ACB, 'BASE',
954      P 0937 1      ACI, 'INCREMENT',
955      P 0938 1
956      P 0939 1      )
957      0940 1      );
958      0941 1
959      0942 1
960      0943 1      Dispatch for INACTIVE keyword
961      0944 1
962      0945 1
963      P 0946 1 $STATE (ST_CCI_PRC_IAC,      ! Inactive things
964      P 0947 1
965      P 0948 1      DISPATCH_STATES
966      P 0949 1      (CCI,
967      P 0950 1
968      P 0951 1      IAB, 'BASE',
969      P 0952 1      IAI, 'INCREMENT',
970      P 0953 1      IAT, 'THRESHOLD',
971      P 0954 1
972      P 0955 1      )
973      0956 1      );
974      0957 1
975      0958 1
976      0959 1      Dispatch for DYING keyword
977      0960 1
978      0961 1
979      P 0962 1 $STATE (ST_CCI_PRC_DYE,
980      P 0963 1
981      P 0964 1      DISPATCH_STATES
```


NCPSTACLP
V04-000

Clear/Purge Parse States and Data
Clear/Purge Circuits

M 10
16-Sep-1984 00:05:04
14-Sep-1984 12:48:17

VAX-11 Bliss-32 V4.0-742
[NCP.SRC]NCPSTACLP.B32;1

Page 25
(18)

:	982	P	0965	1	(CCI,
:	983	P	0966	1	
:	984	P	0967	1	DYB, 'BASE',
:	985	P	0968	1	DYI, 'INCREMENT',
:	986	P	0969	1	DYT, 'THRESHOLD',
:	987	P	0970	1	)
:	988	P	0971	1	);
:	989		0972	1	

NC
VO

```

: 991      0973 1 %SBTTL 'Clear/Purge Lines'
: 992      0974 1
: 993      0975 1
: 994      0976 1
: 995      0977 1
: 996      0978 1
: 997      0979 1
: 998      0980 1
: 999      0981 1
1000      0982 1
1001      P 0983 1
1002      P 0984 1
1003      P 0985 1
1004      P 0986 1
1005      P 0987 1
1006      0988 1
1007      0989 1
1008      0990 1
1009      0991 1
1010      0992 1
1011      0993 1
1012      0994 1
1013      P 0995 1
1014      P 0996 1
1015      P 0997 1
1016      0998 1
1017      0999 1
1018      P 1000 1
1019      P 1001 1
1020      P 1002 1
1021      P 1003 1
1022      P 1004 1
1023      1005 1
1024      1006 1
1025      1007 1
1026      1008 1
1027      1009 1
1028      1010 1
1029      P 1011 1
1030      P 1012 1
1031      1013 1
1032      1014 1
1033      P 1015 1
1034      P 1016 1
1035      1017 1

%SBTTL 'Clear/Purge Lines'

Clear purge lines

Collect the line id and save it, prompt if none

COMMAND PROMPT
(CLI, ENT, NCP$_INVVAL,
 ( (SE_LINE_ID), , ACT$SAVPRM, , , PBK$G_CLI_ENT)
)

Come here after KNOWN and either prompt
or dispatch to process args

$STATE (ST_CLI_GO,
        (TPA$_EOS),
        (TPA$_LAMBDA, ST_CLI_PRC)
);

QUERY_STATES ! Build states to query for params
(CLI,

ALL

)

Perform the function if there is no junk on the line

$STATE (ST_CLI_DOIT, ! Set the option mask
        (TPA$_LAMBDA, , , NMA$C_ENT_LIN, NCP$GL_OPTION, )
);

$STATE ( ! Perform the function
        (TPA$_EOS, TPA$_EXIT, ACT$VRB_UTILITY, , , SDB$G_CLI)
);
```

```
: 1037      1018 1
: 1038      1019 1
: 1039      1020 1
: 1040      1021 1
: 1041      1022 1
: 1042      P 1023 1
: 1043      P 1024 1
: 1044      P 1025 1
: 1045      P 1026 1
: 1046      P 1027 1
: 1047      P 1028 1
: 1048      P 1029 1
: 1049      P 1030 1
: 1050      P 1031 1
: 1051      P 1032 1
: 1052      P 1033 1
: 1053      P 1034 1
: 1054      P 1035 1
: 1055      P 1036 1
: 1056      P 1037 1
: 1057      P 1038 1
: 1058      P 1039 1
: 1059      P 1040 1
: 1060      P 1041 1
: 1061      P 1042 1
: 1062      P 1043 1
: 1063      P 1044 1
: 1064      P 1045 1
: 1065      P 1046 1
: 1066      P 1047 1
: 1067      P 1048 1
: 1068      P 1049 1
: 1069      P 1050 1
: 1070      P 1051 1
: 1071      P 1052 1
: 1072      P 1053 1
: 1073      P 1054 1
: 1074      1055 1

      !
      ! Dispatch for line parameters
      !
      $STATE (ST_CLI_PRC,
      DISPATCH_STATES
      (CLI,
      ALL, 'ALL',
      CLO, 'CLOCK',
      CON, 'CONTROLLER',
      COS, 'COST',
      CTM, 'COUNTÉ',
      DDT, 'DEAD',
      DLT, 'DELAY',
      DEV, 'DEVICE',
      DPX, 'DUPLEX',
      HTI, 'HOLDBACK',
      MAX, 'MAXIMUM',
      MCD, 'MICROCODÉ',
      XMD, 'MODE',
      NTM, 'RETRANSMIT',
      NTM, 'NORMAL',
      PRO, 'PROTOCOL',
      BFN, 'RECEIVE',
      SLT, 'SCHEDULING',
      SVC, 'SERVICE',
      STA, 'STATE',
      SRT, 'STREAM',
      TRB, 'TRIBUTARY',
      TYP, 'TYPE',
      )
      (TPAS_EOS, ST_CLI_DOIT)
      );
```

! [V2 only]


```

: 1076      1056 1
: 1077      1057 1
: 1078      1058 1
: 1079      1059 1
: 1080      1060 1
: 1081      1061 1
: 1082      P 1062 1
: 1083      P 1063 1
: 1084      P 1064 1
: 1085      P 1065 1
: 1086      P 1066 1
: 1087      P 1067 1
: 1088      P 1068 1
: 1089      P 1069 1
: 1090      P 1070 1
: 1091      P 1071 1
: 1092      P 1072 1
: 1093      P 1073 1
: 1094      P 1074 1
: 1095      P 1075 1
: 1096      P 1076 1
: 1097      P 1077 1
: 1098      P 1078 1
: 1099      P 1079 1
: 1100      P 1080 1
: 1101      P 1081 1
: 1102      P 1082 1
: 1103      P 1083 1
: 1104      P 1084 1
: 1105      P 1085 1
: 1106      P 1086 1
: 1107      P 1087 1
: 1108      P 1088 1
: 1109      P 1089 1
: 1110      P 1090 1
: 1111      1091 1

      These states take care of noise words
      and call the subexpressions to do the work

      PROCESS_STATES
      (CLI,

      ALL, ,
      CLO, ,
      CON, ,
      COS, ,
      CTM, 'TIMER',
      DDT, 'TIMER',
      DLT, 'TIMER',
      DEV, ,
      DPX, ,
      HTI, 'TIMER',
      MCD, 'DUMP',
      MWI, ,
      MRT, ,
      MBL, ,
      NTM, 'TIMER',
      PRO, ,
      BFN, 'BUFFERS',
      SLT, 'TIMER',
      STM, ,
      SVM, ,
      STA, ,
      SRT, 'TIMER',
      TRB, ,
      TYP, ,
      XMD, ,

      )

      ! From MAX below
      ! From MAX below
      ! From MAX below
```

```
: 1113      1092  1
: 1114      1093  1
: 1115      1094  1
: 1116      1095  1
: 1117      1096  1
: 1118      P 1097  1
: 1119      P 1098  1
: 1120      P 1099  1
: 1121      P 1100  1
: 1122      P 1101  1
: 1123      P 1102  1
: 1124      P 1103  1
: 1125      P 1104  1
: 1126      P 1105  1
: 1127      P 1106  1
: 1128      P 1107  1
: 1129      P 1108  1
: 1130      P 1109  1
: 1131      P 1110  1
: 1132      P 1111  1
: 1133      P 1112  1
: 1134      P 1113  1
: 1135      P 1114  1
: 1136      P 1115  1
: 1137      P 1116  1
: 1138      P 1117  1
: 1139      P 1118  1
: 1140      P 1119  1
: 1141      P 1120  1
: 1142      P 1121  1
: 1143      P 1122  1
: 1144      P 1123  1
: 1145      P 1124  1
: 1146      P 1125  1
: 1147      P 1126  1
: 1148      1127  1
```

These subexpressions store away the parameter id's

SUB_EXPRESSIONS
(CLI,
ALL, TPAS_EOS,
CLO, TPAS_LAMBDA,
CON, TPAS_LAMBDA,
CCS, TPAS_LAMBDA,
CTM, TPAS_LAMBDA,
DDT, TPAS_LAMBDA,
DLT, TPAS_LAMBDA,
DEV, TPAS_LAMBDA,
DPX, TPAS_LAMBDA,
HTI, TPAS_LAMBDA,
MBL, TPAS_LAMBDA,
MCD, TPAS_LAMBDA,
MRT, TPAS_LAMBDA,
MWI, TPAS_LAMBDA,
NTM, TPAS_LAMBDA,
PRO, TPAS_LAMBDA,
BFN, TPAS_LAMBDA,
SLT, TPAS_LAMBDA,
STM, TPAS_LAMBDA,
SVM, TPAS_LAMBDA,
STA, TPAS_LAMBDA,
SRT, TPAS_LAMBDA,
TRB, TPAS_LAMBDA,
TYP, TPAS_LAMBDA,
XMD, TPAS_LAMBDA,
)


```
: 1150      1128 1 |
: 1151      1129 1 | Dispatch on MAXIMUM keyword
: 1152      1130 1 |
: 1153      1131 1 |
: 1154      P 1132 1 $STATE (ST_CLI_PRC_MAX,
: 1155      P 1133 1
: 1156      P 1134 1 DISPATCH_STATES
: 1157      P 1135 1 (CLI,
: 1158      P 1136 1
: 1159      P 1137 1 MBL, 'BLOCK',
: 1160      P 1138 1 MRT, 'RETRANSMITS',
: 1161      P 1139 1 MWI, 'WINDOW',
: 1162      P 1140 1
: 1163      P 1141 1 )
: 1164      1142 1 );
: 1165      1143 1
: 1166      1144 1 |
: 1167      1145 1 | Dispatch on line SERVICE keyword
: 1168      1146 1 |
: 1169      1147 1 |
: 1170      P 1148 1 $STATE (ST_CLI_PRC_SVC,
: 1171      P 1149 1
: 1172      P 1150 1 DISPATCH_STATES
: 1173      P 1151 1 (CLI,
: 1174      P 1152 1
: 1175      P 1153 1 SVM, 'MODE',
: 1176      P 1154 1 STM, 'TIMER',
: 1177      P 1155 1
: 1178      P 1156 1 )
: 1179      1157 1 );
```



```
: 1181      1158 1 %SBTTL 'Clear/Purge Logging'
: 1182      1159 1
: 1183      1160 1
: 1184      1161 1
: 1185      1162 1
: 1186      1163 1
: 1187      P 1164 1
: 1188      P 1165 1
: 1189      P 1166 1
: 1190      P 1167 1
: 1191      P 1168 1
: 1192      1169 1
: 1193      1170 1
: 1194      P 1171 1
: 1195      P 1172 1
: 1196      1173 1
: 1197      1174 1
: 1198      P 1175 1
: 1199      P 1176 1
: 1200      P 1177 1
: 1201      1178 1
: 1202      1179 1
: 1203      P 1180 1
: 1204      P 1181 1
: 1205      P 1182 1
: 1206      P 1183 1
: 1207      P 1184 1
: 1208      1185 1
: 1209      1186 1
: 1210      P 1187 1
: 1211      P 1188 1
: 1212      P 1189 1
: 1213      P 1190 1
: 1214      P 1191 1
: 1215      1192 1
: 1216      1193 1
: 1217      P 1194 1
: 1218      P 1195 1
: 1219      P 1196 1
: 1220      P 1197 1
: 1221      P 1198 1
: 1222      1199 1
: 1223      1200 1
: 1224      P 1201 1
: 1225      P 1202 1
: 1226      1203 1
: 1227      1204 1
: 1228      P 1205 1
: 1229      P 1206 1
: 1230      1207 1
: 1231      1208 1

      %SBTTL 'Clear/Purge Logging'

      Clear purge logging

      COMMAND PROMPT
      (CLO, ENT, NCP$_INVVAL,
      ( (SE_LOG_TYP) )
      )
      ! Collect the logging type or prompt

      $STATE (ST_CLO_GO,
      (TPAS_$LAMBDA, , ACT$_SAVPRM, , , PBK$_EVE_ESET)
      );
      ! Come here from elsewhere

      $STATE (
      (TPAS_$EOS),
      (TPAS_$LAMBDA, ST_CLO_PRC)
      );
      ! And prompt or
      ! Process command line

      QUERY_STATES
      (CLO,
      ALL
      )
      ! All is special, must query

      PROMPT_STATES
      (CLO,
      SRC, EVK, SNO
      )
      ! Obtain parameters

      QUERY_STATES
      (CLO,
      NAM
      )
      ! Only a query here

      $STATE (ST_CLO_DOIT,
      (TPAS_$LAMBDA, , , NMA$_C_ENT_LOG, NCP$_G_OPTION, )
      );
      ! Process the function

      $STATE (
      (TPAS_$EOS, TPAS_$EXIT, ACT$_VRB_UTILITY, , , SDB$_G_CLO)
      );
```

```

: 1233      1209 1
: 1234      1210 1
: 1235      1211 1
: 1236      1212 1
: 1237      1213 1
: 1238      P 1214 1 $STATE (ST_CLO_PRC,      ! Dispatch to process parameters
: 1239      P 1215 1
: 1240      P 1216 1 DISPATCH_STATES
: 1241      P 1217 1 (CLO,
: 1242      P 1218 1
: 1243      P 1219 1 ALL, 'ALL',
: 1244      P 1220 1 CIR, 'CIRCUIT',
: 1245      P 1221 1 EVL, 'EVENTS',
: 1246      P 1222 1 KNE, 'KNOWN',
: 1247      P 1223 1 LIN, 'LINE',
: 1248      P 1224 1 NOD, 'NODE',
: 1249      P 1225 1 NAM, 'NAME',
: 1250      P 1226 1 SNK, 'SINK',
: 1251      P 1227 1
: 1252      P 1228 1 )
: 1253      P 1229 1 ,(TPAS_EOS, ST_CLO_DOIT)      ! Only if end of string
: 1254      1230 1

```



```
: 1256      1231 1
: 1257      1232 1
: 1258      1233 1
: 1259      1234 1
: 1260      1235 1
: 1261      P 1236 1
: 1262      P 1237 1
: 1263      P 1238 1
: 1264      P 1239 1
: 1265      P 1240 1
: 1266      P 1241 1
: 1267      P 1242 1
: 1268      P 1243 1
: 1269      P 1244 1
: 1270      P 1245 1
: 1271      P 1246 1
: 1272      P 1247 1
: 1273      1248 1
: 1274      1249 1
: 1275      P 1250 1
: 1276      P 1251 1
: 1277      P 1252 1
: 1278      P 1253 1
: 1279      1254 1
: 1280      1255 1
: 1281      P 1256 1
: 1282      P 1257 1
: 1283      P 1258 1
: 1284      1259 1
: 1285      1260 1
: 1286      1261 1
: 1287      1262 1
: 1288      1263 1
: 1289      1264 1
: 1290      1265 1
: 1291      P 1266 1
: 1292      P 1267 1
: 1293      P 1268 1
: 1294      P 1269 1
: 1295      P 1270 1
: 1296      P 1271 1
: 1297      P 1272 1
: 1298      P 1273 1
: 1299      P 1274 1
: 1300      P 1275 1
: 1301      P 1276 1
: 1302      P 1277 1
: 1303      P 1278 1
: 1304      1279 1
: 1305      1280 1
: 1306      1281 1
: 1307      1282 1
: 1308      1283 1
: 1309      1284 1
: 1310      1285 1
: 1311      P 1286 1
: 1312      P 1287 1

Process states

PROCESS_STATES
(CLO,

ALL, ,
CIR, ,
EVL, ,
KNE, 'EVENTS',
LIN, ,
NOD, ,
NAM, ,
SNK, ,

)

$STATE (ST_CLO_SRC,          ! Source must be line, circuit or node
('CIRCUIT', ST_CLO_CIR),
('LINE', ST_CLO_LIN),
('NODE', ST_CLO_NOD),
);

$STATE (ST_CLO_SNK,          ! Sub-dispatch for sink node or exec
('EXECUTOR', ST_CLO_SEX),
('NODE', ST_CLO_SNO)
);

Subexpressions

SUB_EXPRESSIONS
(CLO,

ALL, TPAS EOS,
CIR, (SE_CIRC_ID),
EVL, (SE_EVENT_LIST),
KNE, (SE_EVENT_KNOWN),
LIN, (SE_LINE_ID),
NOD, (SE_NODE_ID),
NAM, TPAS LAMBDA,
SNO, (SE_NODE_ID),
SEX, TPAS_LAMBDA,

)

Subexpression to prompt for events

$STATE (ST_CLO_EVK,
('SE_EVENT_LIST), TPAS_EXIT, ACT$SAVPRM, , , PBK$G_CLO_EVL),
```

NCPSTACLP
V04-000

Clear/Purge Parse States and Data
Clear/Purge Logging

I 11
16-Sep-1984 00:05:04
14-Sep-1984 12:48:17

VAX-11 Bliss-32 V4.0-742
[NCP.SRC]NCPSTACLP.B32;1

Page 34
(26)

```
: 1313      P 1288 1      ('KNOWN', , ACT$SAVPRM, , , PBK$G_CLO_EVL)
: 1314      1289 1      );
: 1315      1290 1
: 1316      P 1291 1 $STATE (,
: 1317      P 1292 1      ( (SE_EVENT_KNOWN), TPAS_EXIT)      ! Store known event code
: 1318      1293 1
: 1319      1294 1
```



```
: 1321      1295 1 %SBTTL 'Clear/Purge Objects'
: 1322      1296 1
: 1323      1297 1
: 1324      1298 1
: 1325      1299 1
: 1326      1300 1
: 1327      1301 1
: 1328      1302 1
: 1329      1303 1
: 1330      1304 1
: 1331      P 1305 1
: 1332      P 1306 1
: 1333      P 1307 1
: 1334      P 1308 1
: 1335      P 1309 1
: 1336      1310 1
: 1337      1311 1
: 1338      1312 1
: 1339      1313 1
: 1340      1314 1
: 1341      1315 1
: 1342      P 1316 1
: 1343      P 1317 1
: 1344      P 1318 1
: 1345      1319 1
: 1346      1320 1
: 1347      P 1321 1
: 1348      P 1322 1
: 1349      P 1323 1
: 1350      P 1324 1
: 1351      P 1325 1
: 1352      1326 1
: 1353      1327 1
: 1354      1328 1
: 1355      P 1329 1
: 1356      P 1330 1
: 1357      1331 1
: 1358      1332 1
: 1359      P 1333 1
: 1360      P 1334 1
: 1361      1335 1

      %SBTTL 'Clear/Purge Objects'

      Clear purge objects

      Collect the object name/number or prompt

      COMMAND PROMPT
      (COB, ENT, NCP$_INVVAL,
      ( (SE_OBJECT_ID), , ACT$SAVPRM, , , PBK$G_COB_ENT)
      )

      Should we prompt or just process the remainder of the command?

      $STATE (ST_COB_GO,
      (TPA$_EOS),
      (TPA$_LAMBDA, ST_COB_PRC),
      );

      QUERY_STATES ! Prompt for selected parameters
      (COB,
      ALL, NUM, FIL, PRV, USR, PSW, PRX
      )

      $STATE (ST_COB_DOIT,
      (TPA$_LAMBDA, , , NMA$C_SENT_OBJ, NCP$GL_OPTION, ),
      );

      $STATE (
      (TPA$_EOS, TPA$_EXIT, ACT$VRB_UTILITY, , , SDB$G_COB)
      );
```

```
: 1363      1336 1 %SBTTL 'Dispatch States'
: 1364      1337 1
: 1365      1338 1
: 1366      1339 1
: 1367      1340 1
: 1368      1341 1
: 1369      P 1342 1 $STATE (ST_COB_PRC,
: 1370      P 1343 1
: 1371      P 1344 1 DISPATCH_STATES
: 1372      P 1345 1 (COB,
: 1373      P 1346 1
: 1374      P 1347 1 ALL, 'ALL',
: 1375      P 1348 1 ACC, 'ACCOUNT',
: 1376      P 1349 1 FIL, 'FILE',
: 1377      P 1350 1 NUM, 'NUMBER',
: 1378      P 1351 1 PRV, 'PRIVILEGES',
: 1379      P 1352 1 PRX, 'PROXY',
: 1380      P 1353 1 PSW, 'PASSWORD',
: 1381      P 1354 1 USR, 'USER',
: 1382      P 1355 1
: 1383      P 1356 1 )
: 1384      P 1357 1 (TPAS_EOS, ST_COB_DOIT)
: 1385      1358 1
: 1386      1359 1
: 1387      1360 1 %SBTTL 'Process States'
: 1388      1361 1
: 1389      1362 1
: 1390      1363 1
: 1391      1364 1
: 1392      1365 1
: 1393      P 1366 1 PROCESS_STATES
: 1394      P 1367 1 (COB,
: 1395      P 1368 1
: 1396      P 1369 1 ALL, ,
: 1397      P 1370 1 ACC, ,
: 1398      P 1371 1 FIL, ,
: 1399      P 1372 1 NUM, ,
: 1400      P 1373 1 PRV, ,
: 1401      P 1374 1 PRX, ,
: 1402      P 1375 1 PSW, ,
: 1403      P 1376 1 USR, ,
: 1404      P 1377 1
: 1405      1378 1 )
```

```

: 1407      1379 1 %SBTTL 'Subexpressions'
: 1408      1380 1
: 1409      1381 1
: 1410      1382 1
: 1411      1383 1
: 1412      1384 1
: 1413      P 1385 1
: 1414      P 1386 1
: 1415      P 1387 1
: 1416      P 1388 1
: 1417      P 1389 1
: 1418      P 1390 1
: 1419      P 1391 1
: 1420      P 1392 1
: 1421      P 1393 1
: 1422      P 1394 1
: 1423      P 1395 1
: 1424      P 1396 1
: 1425      1397 1

```

Sub_expressions

SUB EXPRESSIONS
(COB,

ALL, TPAS_EOS,
ACC, TPAS_LAMBDA,
FIL, TPAS_LAMBDA,
NUM, TPAS_LAMBDA,
PRV, TPAS_LAMBDA,
PRX, TPAS_LAMBDA,
PSW, TPAS_LAMBDA,
USR, TPAS_LAMBDA,

)


```
: 1427      1398 1 %SBTTL 'Define Subexpressions from Library'
: 1428      1399 1
: 1429      1400 1
: 1430      1401 1
: 1431      1402 1
: 1432      1403 1
: 1433      1404 1
: 1434      1405 1
: 1435      1406 1
: 1436      1407 1
: 1437      1408 1
: 1438      1409 1
: 1439      1410 1
: 1440      1411 1

                Define subexpressions from library

                SEM_NODE_ID      : Node id strings
                SEM_CIRC_ID      : Circuit id strings
                SEM_LINE_ID      : Line id strings
                SEM_QUOT_STR      : Quoted strings
                SEM_OBJECT_ID     : Object name/number
                SEM_LOG_TYP      : Logging entity type
                SEM_EVENT_LIST    : List of event numbers
                SEM_QUERY         : Query state subexpressions
```

NCPSTACLP
V04-000

Clear/Purge Parse States and Data
Object Listing of Parse Table

N 11
16-Sep-1984 00:05:04
14-Sep-1984 12:48:17

VAX-11 Bliss-32 V4.0-742
[NCP.SRC]NCPSTACLP.B32;1

Page 39
(31)

: 1442	1412	1	%SBTTL	'Object Listing of Parse Table'	
: 1443	1413	1			
: 1444	1414	1	END		!End of module
: 1445	1415	0	ELUDOM		

0269 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

