

NNN	NNN	IIIIIIIIII	CCCCCCCCCCCC	NNN	NNN	FFFFFFFFFFFFFF
NNN	NNN	IIIIIIIIII	CCCCCCCCCCCC	NNN	NNN	FFFFFFFFFFFFFF
NNN	NNN	IIIIIIIIII	CCCCCCCCCCCC	NNN	NNN	FFFFFFFFFFFFFF
NNN	NNN	III	CCC	NNN	NNN	FFF
NNN	NNN	III	CCC	NNN	NNN	FFF
NNN	NNN	III	CCC	NNN	NNN	FFF
NNNNNN	NNN	III	CCC	NNNNNN	NNN	FFF
NNNNNN	NNN	III	CCC	NNNNNN	NNN	FFF
NNNNNN	NNN	III	CCC	NNNNNN	NNN	FFF
NNN	NNN	III	CCC	NNN	NNN	FFFFFFFFFFFFFF
NNN	NNN	III	CCC	NNN	NNN	FFFFFFFFFFFFFF
NNN	NNN	III	CCC	NNN	NNN	FFFFFFFFFFFFFF
NNN	NNN	III	CCC	NNN	NNN	FFF
NNN	NNN	III	CCC	NNN	NNN	FFF
NNN	NNN	III	CCC	NNN	NNN	FFF
NNN	NNN	III	CCC	NNN	NNN	FFF
NNN	NNN	III	CCC	NNN	NNN	FFF
NNN	NNN	III	CCC	NNN	NNN	FFF
NNN	NNN	IIIIIIIIII	CCCCCCCCCCCC	NNN	NNN	FFF
NNN	NNN	IIIIIIIIII	CCCCCCCCCCCC	NNN	NNN	FFF
NNN	NNN	IIIIIIIIII	CCCCCCCCCCCC	NNN	NNN	FFF

```

CCCCCCCC  NN      NN  FFFFFFFFFF  IIIIII  NN      NN  TTTTTTTTTT  RRRRRRRR  PPPPPPPP  TTTTTTTTTT
CCCCCCCC  NN      NN  FFFFFFFFFF  IIIIII  NN      NN  TTTTTTTTTT  RRRRRRRR  PPPPPPPP  TTTTTTTTTT
CC         NN      NN  FF          II       NN      NN  TT          RR      RR  PP      PP  TT
CC         NN      NN  FF          II       NN      NN  TT          RR      RR  PP      PP  TT
CC         NNNN    NN  FF          II       NNNN    NN  TT          RR      RR  PP      PP  TT
CC         NNNN    NN  FF          II       NNNN    NN  TT          RR      RR  PP      PP  TT
CC         NN  NN  NN  FFFFFFFF  II       NN  NN  NN  TT          RRRRRRRR  PPPPPPPP  TT
CC         NN  NN  NN  FFFFFFFF  II       NN  NN  NN  TT          RRRRRRRR  PPPPPPPP  TT
CC         NN      NNNN  FF          II       NN      NNNN  TT          RR  RR  PP      TT
CC         NN      NNNN  FF          II       NN      NNNN  TT          RR  RR  PP      TT
CC         NN      NN  FF          II       NN      NN  TT          RR      RR  PP      TT
CC         NN      NN  FF          II       NN      NN  TT          RR      RR  PP      TT
CCCCCCCC  NN      NN  FF          IIIIII  NN      NN  TT          RR      RR  PP      TT
CCCCCCCC  NN      NN  FF          IIIIII  NN      NN  TT          RR      RR  PP      TT

```

```

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
LLLLLLLLLL IIIIII  SSSSSSSS
LLLLLLLLLL IIIIII  SSSSSSSS

```

```

1 0001 0 %TITLE 'DECnet Ethernet Configurator Module'
2 0002 0 MODULE CNFINTRPT
3 0003 0 (
4 0004 0 LANGUAGE (BLISS32),
5 0005 0 IDENT = 'V04-000'
6 0006 0 ) =
7 0007 0 *****
8 0008 0 *
9 0009 0 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY *
10 0010 0 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS. *
11 0011 0 * ALL RIGHTS RESERVED. *
12 0012 0 *
13 0013 0 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED *
14 0014 0 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE *
15 0015 0 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER *
16 0016 0 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY *
17 0017 0 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY *
18 0018 0 * TRANSFERRED. *
19 0019 0 *
20 0020 0 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE *
21 0021 0 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT *
22 0022 0 * CORPORATION. *
23 0023 0 *
24 0024 0 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS *
25 0025 0 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL. *
26 0026 0 *
27 0027 0 *
28 0028 0 *****
29 0029 0
30 0030 0
31 0031 0 ++
32 0032 0 FACILITY: DECnet Configurator Module (NICONFIG)
33 0033 0
34 0034 0 ABSTRACT:
35 0035 0
36 0036 0 This module contains the routines for establishing, and breaking
37 0037 0 logical links to NICONFIG.
38 0038 0
39 0039 0 ENVIRONMENT: VAX/VMS Operating System
40 0040 0
41 0041 0 AUTHOR: Bob Grosso, CREATION DATE: 13-Oct-1982
42 0042 0
43 0043 0 MODIFIED BY:
44 0044 0
45 0045 0 V03-001 RPG0001 Bob Grosso 02-May-1983
46 0046 0 Ensure NICONFIG will die gracefully on Network Shutdown.
47 0047 0 --
48 0048 1 BEGIN

```

```

50 0049 1 %SBTTL 'Definitions'
51 0050 1
52 0051 1
53 0052 1 !! INCLUDE FILES:
54 0053 1 !!
55 0054 1
56 0055 1 LIBRARY 'SYSS$LIBRARY:STARLET'; ! VMS common definitions
57 0056 1
58 0057 1 LIBRARY 'SHRLIB$:NET'; ! Network definitions
59 0058 1
60 0059 1 LIBRARY 'SHRLIB$:NMALIBRY'; ! NICE Code definitions
61 0060 1
62 0061 1 REQUIRE 'LIB$:CNFDEF.R32';
63 0152 1
64 0153 1 REQUIRE 'SRC$:CNFPREFIX.REQ';
65 0250 1
66 0251 1
67 0252 1 !!
68 0253 1 !! BUILTIN functions
69 0254 1 !!
70 0255 1
71 0256 1 BUILTIN
72 0257 1 INSQUE, ! INSQUE instruction
73 0258 1 REMQUE; ! REMQUE instruction
74 0259 1
75 0260 1 !!
76 0261 1 !! Own storage
77 0262 1 !!
78 0263 1
79 0264 1 OWN
80 0265 1 SI_IOSB : BBLOCK [8]; ! IO status block for
81 0266 1
82 0267 1 !!
83 0268 1 !! TABLE OF CONTENTS:
84 0269 1 !!
85 0270 1
86 0271 1 FORWARD ROUTINE
87 0272 1
88 0273 1 CNF$SOLICIT_INTERRUPT : NOVALUE, ! Solicit work items
89 0274 1 NET_INTERRUPT : NOVALUE, ! Action routine to receive first SET command
90 0275 1 OPEN_REQUEST_LINK : NOVALUE, ! Open incoming request logical link
91 0276 1 CNF$CLOSE_REQUEST_LINK : NOVALUE, ! Close request logical link
92 0277 1 SHUTDOWN : NOVALUE, ! Shut down receiver gracefully
93 0278 1 CNF$SOLICIT_REQUEST : NOVALUE, ! Perform a read on the logical link
94 0279 1 REQUEST_RECEIVED : NOVALUE; ! Accept incoming request record
95 0280 1
96 0281 1
97 0282 1
98 0283 1 !!
99 0284 1 !! EXTERNAL REFERENCES:
100 0285 1 !!
101 0286 1
102 0287 1 EXTERNAL ROUTINE
103 0288 1
104 0289 1 ! Module CNFMAIN
105 0290 1
106 0291 1 CNF$EXIT, ! Clean up and exit

```

```

: 107 0292 1 CNF$TRACE, ! Log messages to log file
: 108 0293 1 CNF$LOG_DATA, ! Log messages to log file
: 109 0294 1 CNF$GET_ZVM, ! Get zeroed virtual memory
: 110 0295 1 CNF$FREE_VM, ! Free virtual memory
: 111 0296 1
: 112 0297 1 ! Module CNFREQUES
: 113 0298 1
: 114 0299 1 CNF$DISABLE_SURVEIL : NOVALUE, ! Discontinue surveillance on specified circuit
: 115 0300 1 CNF$PROCESS_REQUEST : NOVALUE, ! Parse and perform requested function
: 116 0301 1
: 117 0302 1 ! Module CNFWORKQ
: 118 0303 1
: 119 0304 1 WKQ$ADD_WORK_ITEM, ! Add work to work queue
: 120 0305 1 WKQ$DO_WORK_ITEM; ! Add work to work queue
: 121 0306 1
: 122 0307 1
: 123 0308 1 EXTERNAL LITERAL
: 124 0309 1
: 125 0310 1 CNF$_MAILBOX, ! Mailbox error
: 126 0311 1 CNF$_CHAN, ! Error assigning or deassigning channel
: 127 0312 1 CNF$_LINK, ! Error on logical link
: 128 0313 1
: 129 0314 1 CNF$_MAXMBXMSG, ! Maximum mailbox message size
: 130 0315 1 CNF$_SYNCH_EFN, ! Synchronous event flag number
: 131 0316 1 CNF$_ASYNCR_EFN; ! Asynchronous event flag number
: 132 0317 1
: 133 0318 1
: 134 0319 1 EXTERNAL
: 135 0320 1
: 136 0321 1 CNF$_A_MBXMSG : VECTOR [ ,BYTE], ! Mailbox message buffer
: 137 0322 1 CNF$_W_NETCHAN : WORD, ! Channel opened to network
: 138 0323 1 CNF$_W_MBXCHAN : WORD, ! Channel to mailbox
: 139 0324 1 CNF$_B_SURVEILLANCE_SET, ! Boolean: mark if useful work is being accomplished
: 140 0325 1 CNF$_G_LOGMASK : BITVECTOR [32], ! Logging control mask
: 141 0326 1 CNF$_G_CIRSURLST : VECTOR [2], ! List of circuit under surveillance
: 142 0327 1 CNF$_G_IRBLST : VECTOR [2]; ! Listhead for incoming links
: 143 0328 1

```

```

: 145 0329 1 %SBTTL 'CNF$SOLICIT_INTERRUPT Request network interrupts for Connect requests and Shutdown '
: 146 0330 1 GLOBAL ROUTINE CNF$SOLICIT_INTERRUPT: NOVALUE =
: 147 0331 1
: 148 0332 1 |++
: 149 0333 1 | FUNCTIONAL DESCRIPTION:
: 150 0334 1 |
: 151 0335 1 | Issue an asynchronous QIO on the associated mailbox
: 152 0336 1 | for the network channel in expectation of receiving
: 153 0337 1 | requests for connects, or Shutdown notification.
: 154 0338 1 | Called the first time from MAIN routine in user mode
: 155 0339 1 | and subsequent times from NET_INTERRUPT to execute in AST mode.
: 156 0340 1 |
: 157 0341 1 | FORMAL PARAMETERS:
: 158 0342 1 | NONE
: 159 0343 1 |
: 160 0344 1 | IMPLICIT INPUTS:
: 161 0345 1 |
: 162 0346 1 |     CNF$W_MBXCHAN  Channel number for mailbox
: 163 0347 1 |     CNF$A_MBXMSG   Buffer for mailbox msg
: 164 0348 1 |
: 165 0349 1 | IMPLICIT OUTPUTS:
: 166 0350 1 | NONE
: 167 0351 1 |
: 168 0352 1 | ROUTINE VALUE:
: 169 0353 1 | COMPLETION CODES:
: 170 0354 1 |
: 171 0355 1 |     Errors are signalled
: 172 0356 1 |
: 173 0357 1 | SIDE EFFECTS:
: 174 0358 1 | NONE
: 175 0359 1 |
: 176 0360 1 | --
: 177 0361 1 |
: 178 0362 2 BEGIN
: 179 0363 2
: 180 0364 2     LOCAL
: 181 0365 2     STATUS;
: 182 0366 2
: 183 0367 2     CNF$TRACE (DBG$C_TRACE, $DESCRIPTOR('TRACE'),
: 184 0368 2     $DESCRIPTOR('cnf$solicit_interrupt'));
: 185 0369 2
: 186 0370 2     STATUS = $QIO (
: 187 P 0371 2         FUNC = IOS$ READVBLK,      ! Request read on mailbox
: 188 P 0372 2         CHAN = .CNF$W_MBXCHAN, ! Use assigned channel
: 189 P 0373 2         EFN = CNF$C_ASYNC_EFN,  ! Asynchronous Event flag number
: 190 P 0374 2         IOSB = SI IOSB,        ! Interrupt request block
: 191 P 0375 2         ASTADR = RET_INTERRUPT, ! AST routine to execute on read completion
: 192 P 0376 2         P1 = CNF$A_MBXMSG,      ! Buffer to contain mailbox message
: 193 P 0377 2         P2 = CNF$C_MAXMBXMSG); ! Size maximum on mailbox message
: 194 0378 2
: 195 0379 2     IF NOT .STATUS
: 196 0380 2     THEN                                     ! report an error
: 197 0381 2         SIGNAL (CNF$_MAILBOX, 0, .STATUS);
: 198 0382 2
: 199 0383 2     RETURN;
: 200 0384 1 END;                                     ! End routine CNF$SOLICIT_INTERRUPT

```

```

                                .TITLE CNFINTRPT DECnet Ethernet Configurator Module
                                .IDENT \V04-000\
                                .PSECT $SPLITS,NOWRT,NOEXE,2
                                45 43 41 52 54 00000 P.AAB: .ASCII \TRACE\
                                00005 .BLKB 3
                                00000005 00008 P.AAA: .LONG 5
                                00000000' 0000C .ADDRESS P.AAB
74 6E 69 5F 74 69 63 69 6C 6F 73 24 66 6E 63 00010 P.AAD: .ASCII \cnf$solicit_interrupt\
74 70 75 72 72 65 0001F
                                00025 .BLKB 3
                                00000015 00028 P.AAC: .LONG 21
                                00000000' 0002C .ADDRESS P.AAD
                                .PSECT $OWNS,NOEXE,2
                                00000 SI_IOSB: .BLKB 8
                                .EXTRN CNF$EXIT, CNF$TRACE
                                .EXTRN CNF$LOG_DATA, CNF$GET_ZVM
                                .EXTRN CNF$FREE_VM, CNF$DISABLE_SURVEIL
                                .EXTRN CNF$PROCESS_REQUEST
                                .EXTRN WKQ$ADD_WORK_ITEM
                                .EXTRN WKQ$DO_WORK_ITEM
                                .EXTRN CNF$MAILBOX, CNF$CHAN
                                .EXTRN CNF$LINK, CNF$C_MAXMBXMSG
                                .EXTRN CNF$C_SYNCH_EFN
                                .EXTRN CNF$C_ASYNCH_EFN
                                .EXTRN CNF$A_MBXMSG, CNF$W_NETCHAN
                                .EXTRN CNF$W_MBXCHAN, CNF$B_SURVEILLANCE_SET
                                .EXTRN CNF$G_LOGMASK, CNF$G_CIRSURLST
                                .EXTRN CNF$G_IRBLST, SYSSQIO
                                .PSECT $CODES,NOWRT,2
                                0000 0000 .ENTRY CNF$SOLICIT_INTERRUPT, Save nothing
                                0000' CF 9F 00002 PUSHAB P.AAC
                                0000' CF 9F 00006 PUSHAB P.AAA
                                0000G CF 01 DD 0000A PUSHL #1
                                0000G CF 03 FB 0000C CALLS #3, CNF$TRACE
                                7E 7C 00011 CLRQ -(SP)
                                7E 7C 00013 CLRQ -(SP)
                                00000000G 8F DD 00015 PUSHL #CNF$C_MAXMBXMSG
                                0000G CF 9F 0001B PUSHAB CNF$A_MBXMSG
                                7E D4 0001F CLRL -(SP)
                                0000V CF 9F 00021 PUSHAB NET_INTERRUPT
                                0000' CF 9F 00025 PUSHAB SI_IOSB
                                31 DD 00029 PUSHL #49
                                7E 0000G CF 3C 0002B MOVZWL CNF$W_MBXCHAN, -(SP)
                                00000000G 8F DD 00030 PUSHL #CNF$C_ASYNCH_EFN
                                00000000G 00 0C FB 00036 CALLS #12, SYSSQIO
                                11 50 EB 0003D BLBS STATUS, 1$
                                50 DD 00040 PUSHL STATUS
                                7E D4 00042 CLRL -(SP)
                                00000000G 8F DD 00044 PUSHL #CNF$MAILBOX

```

0330  
0368  
0367  
0377  
0379  
0381





```

202 0385 1 %SBTTL 'net_interrupt Process Net interrupts for Shutdown or Connect Request'
203 0386 1 ROUTINE NET_INTERRUPT : NOVALUE =
204 0387 1
205 0388 1 !++
206 0389 1
207 0390 1 This ASI routine is called when the outstanding QIO
208 0391 1 on the associated mailbox completes. If the interrupt
209 0392 1 indicates a connect is pending, then the acceptance
210 0393 1 routine is added to the work queue.
211 0394 1
212 0395 1 !--
213 0396 1
214 0397 2 BEGIN
215 0398 2 BIND
216 0399 2 MESSAGE_TYPE = CNFSA_MBXMSG [0] : BYTE; ! First byte contains code for message type
217 0400 2
218 0401 2 CNF$TRACE (DBG$C TRACE, $DESCRIPTOR('TRACE'),
219 0402 2 $DESCRIPTOR ('Net_interrupt'));
220 0403 2
221 0404 2 !
222 0405 2 Check message type. If connect request, then connect and wait
223 0406 2 for a set. Anything else is bad news.
224 0407 2 !
225 0408 2 SELECTONEU .MESSAGE_TYPE OF
226 0409 2 SET
227 0410 2
228 0411 2 [MSG$_NETSHUT]: ! Network shutting down
229 0412 2
230 0413 3 BEGIN
231 0414 3 WKQ$ADD WORK_ITEM(SHUTDOWN); ! Shut down receiver gracefully
232 0415 3 CNF$TRACE (DBG$C TRACE, $DESCRIPTOR('TRACE'),
233 0416 3 $DESCRIPTOR ('Net_interrupt - Network shutting down'));
234 0417 3 RETURN; ! Do not re-issue mailbox read
235 0418 3 END;
236 0419 2
237 0420 2 [MSG$_CONNECT]: ! Incoming connect request
238 0421 2 BEGIN
239 0422 3 LOCAL
240 0423 3 PTR,LEN,
241 0424 3 IRB: REF BBLOCK; ! Incoming Request Block
242 0425 3
243 0426 3 CNF$TRACE (DBG$C TRACE, $DESCRIPTOR('TRACE'),
244 0427 3 $DESCRIPTOR ('Net_interrupt - Connect request received'));
245 0428 3
246 0429 3 LEN = IRB$C_LENGTH;
247 0430 3 CNF$GET ZVM(LEN,IRB); ! Allocate incoming request block
248 0431 3 IRB [IRB$W_SIZE] = IRB$C_LENGTH; ! Set length of block
249 0432 3 PTR = 5 + .CNFSA_MBXMSG [4]; ! Get index of start of ascic data
250 0433 3 IRB [IRB$B_NCBLEN] = .CNFSA_MBXMSG [.PTR]; ! Set length of NCB
251 0434 3 CH$COPY (.CNFSA_MBXMSG [.PTR], CNFSA_MBXMSG [.PTR+1],
252 0435 3 0, IRB$C_MAXNCBLEN, IRB [IRB$B_NCB]);
253 0436 3 IRB [IRB$B_BNR_FLINK] = IRB [IRB$B_BNR_FLINK]; ! Initialize list for Buffered NICE Messages
254 0437 3 IRB [IRB$B_BNR_BLINK] = IRB [IRB$B_BNR_FLINK];
255 0438 3 INSQUE (.IRB, .CNF$GQ_IRBLST [1]); ! Insert into list
256 0439 3
257 0440 3 WKQ$ADD_WORK_ITEM(OPEN_REQUEST_LINK,.IRB); ! Queue the connect accept
258 0441 3

```

```

: 259      0442      2      END;
: 260      0443      2
: 261      0444      2      [OTHERWISE]:
: 262      0445      2      BEGIN
: 263      0446      3      CNF$TRACE (DBG$C_TRACE, $DESCRIPTOR('TRACE *** ERROR'),
: 264      0447      3      $DESCRIPTOR ('Net_interrupt - Unprocessed Interrupt'));
: 265      0448      2      END;
: 266      0449      2
: 267      0450      2      TES;
: 268      0451      2
: 269      0452      2      CNF$SOLICIT_INTERRUPT();          ! Issue another read on mailbox
: 270      0453      2
: 271      0454      2      CNF$TRACE (DBG$C_TRACE, $DESCRIPTOR('TRACE'),
: 272      0455      2      $DESCRIPTOR ('Net_interrupt - Interrupt dispatched, another solicited'));
: 273      0456      2
: 274      0457      2      RETURN;
: 275      0458      1      END;          ! Routine net_interrupt

```

															.PSECT		\$SPLITS, NOWRT, NOEXE, 2																
															45	43	41	52	54	00030	P.AAF:	.ASCII	\TRACE\										
																				00035		.BLKB	3										
																				00000005	00038	P.AAE:	.LONG	5									
																				00000000	0003C		.ADDRESS	P.AAF									
															74	70	75	72	72	65	74	6E	49	5F	74	65	4E	00040	P.AAH:	.ASCII	\Net_Interrupt\		
																				0004D		.BLKB	3										
																				00000000	00050	P.AAG:	.LONG	13									
																				00C00000	00054		.ADDRESS	P.AAH									
															45	43	41	52	54	00058	P.AAJ:	.ASCII	\TRACE\										
																				0005D		.BLKB	3										
																				00000005	00060	P.AAI:	.LONG	5									
																				00000000	00064		.ADDRESS	P.AAJ									
2D	20	74	70	75	72	72	65	74	6E	49	5F	74	65	4E	00068	P.AAL:	.ASCII	\Net_Interrupt - Network shutting down\															
69	74	74	75	68	73	20	6B	72	6F	77	74	65	4E	20	00077																		
															6E	77	6F	64	20	67	6E	00086											
																				0008D		.BLKB	3										
																				00000025	00090	P.AAK:	.LONG	37									
																				00000000	00094		.ADDRESS	P.AAL									
															45	43	41	52	54	00098	P.AAN:	.ASCII	\TRACE\										
																				0009D		.BLKB	3										
																				00000005	000A0	P.AAM:	.LONG	5									
																				00000000	000A4		.ADDRESS	P.AAN									
2D	20	74	70	75	72	72	65	74	6E	49	5F	74	65	4E	000A8	P.AAP:	.ASCII	\Net_Interrupt - Connect request received\															
73	65	75	71	65	72	20	74	63	65	6E	6E	6F	43	20	000B7																		
															64	65	76	69	65	63	65	72	20	74	000C6								
																				00000028	000D0	P.AAO:	.LONG	40									
																				00000000	000D4		.ADDRESS	P.AAP									
52	4F	52	52	45	20	2A	2A	2A	20	45	43	41	52	54	000D8	P.AAR:	.ASCII	\TRACE *** ERROR\															
																				000E7		.BLKB	1										
																				0000000F	000E8	P.AAQ:	.LONG	15									
																				00000000	000EC		.ADDRESS	P.AAR									
2D	20	74	70	75	72	72	65	74	6E	49	5F	74	65	4E	000F0	P.AAT:	.ASCII	\Net_Interrupt - Unprocessed Interrupt\															
6E	49	20	64	65	73	73	65	63	6F	72	70	6E	55	20	000FF																		
															74	70	75	72	72	65	74	0010E											
																				00115		.BLKB	3										



			6E DD 0008B	PUSHL	IRB	:	0441
			CF 9F 0008D	PUSHAB	OPEN REQUEST LINK	:	
0000G	CF		02 FB 00091	CALLS	#2, WKQ\$ADD_WORK_ITEM	:	
			0D 11 00096	BRB	3\$	:	0408
		00C8	C7 9F 00098	PUSHAB	P.AAS	:	0447
		0098	C7 9F 0009C	PUSHAB	P.AAQ	:	0446
			01 DD 000A0	PUSHL	#1	:	
	69		03 FB 000A2	CALLS	#3, CNF\$TRACE	:	
FF04	CF		00 FB 000A5	CALLS	#0, CNF\$SOLICIT_INTERRUPT	:	0452
		0118	C7 9F 000AA	PUSHAB	P.AAW	:	0455
		00D8	C7 9F 000AE	PUSHAB	P.AAU	:	0454
			01 DD 000B2	PUSHL	#1	:	
	69		03 FB 000B4	CALLS	#3, CNF\$TRACE	:	
			04 000B7	RET		:	0458

; Routine Size: 184 bytes, Routine Base: \$CODE\$ + 0052

```

277 0459 1 %SBTTL 'open_request_link'
278 0460 1 ROUTINE OPEN_REQUEST_LINK (IRB): NOVALUE =
279 0461 1
280 0462 1 |---
281 0463 1 |
282 0464 1 |   Open the logical link for incoming request records.
283 0465 1 |   Executed off the work queue.
284 0466 1 |
285 0467 1 |   Inputs:
286 0468 1 |
287 0469 1 |       irb = Address of incoming request block
288 0470 1 |
289 0471 1 |   Outputs:
290 0472 1 |
291 0473 1 |       routine = True if link established, false if not
292 0474 1 |---
293 0475 1
294 0476 2 BEGIN
295 0477 2
296 0478 2 MAP
297 0479 2     IRB:          REF BBLOCK;                ! Address of request block
298 0480 2
299 0481 2 LOCAL
300 0482 2     STATUS,
301 0483 2     PTR,
302 0484 2     NCB_DESC:  VECTOR [2];                ! Descriptor of NCB
303 0485 2
304 0486 2 CNF$TRACE (DBG$C TRACE, $DESCRIPTOR('TRACE'),
305 0487 2     $DESCRIPTOR ('open_request_link'));
306 0488 2
307 0489 2 |
308 0490 2 |       Setup NCB for connect accept
309 0491 2 |
310 0492 2
311 0493 2 NCB_DESC [0] = .IRB [IRB$B_NCBLEN];          ! Get length of requestor ncb
312 0494 2 NCB_DESC [1] = IRB [IRB$T_NCB];          ! and address of ncb
313 0495 2
314 0496 2 |
315 0497 2 |       Get copy of NCB up to slash to enable its use in error reporting
316 0498 2 |
317 0499 2
318 0500 2 PTR = CH$FIND_CH (.IRB [IRB$B_NCBLEN], IRB [IRB$T_NCB], '/');
319 0501 2
320 0502 2 IF NOT CH$FAIL (.PTR)                        ! If ending slash found,
321 0503 2 THEN
322 0504 2     IRB [IRB$B_NCBLEN] = .PTR - IRB [IRB$T_NCB]; ! then truncate rest of junk
323 0505 2
324 0506 2 |
325 0507 2 |       Setup NCB for accept by zeroing optional data sent by requestor
326 0508 2 |
327 0509 2
328 0510 2 (.PTR+3) <0,8> = 0;                          ! Zero optional data
329 0511 2
330 P 0512 2 STATUS = $ASSIGN (DEVNAM = %ASCII ' NET:',          ! Get channel for incoming link
331 0513 2     CHAN = IRB [IRB$W_CRAN]);
332 0514 2
333 0515 2 IF NOT .STATUS                                ! If error assigning channel,

```

```

334 0516 2 THEN
335 0517 2 BEGIN
336 0518 2 SIGNAL (CNF$ CHAN, 0, .STATUS); ! then report the error
337 0519 2 CNF$CLOSE_REQUEST_LINK(.IRB); ! and deallocate the storage
338 0520 2 RETURN;
339 0521 2 END;
340 0522 2
341 P 0523 2 STATUS = $QIOW(FUNC = IOS ACCESS, ! Accept the logical link
342 P 0524 2 CHAN = .IRB [IRB$W CHAN],
343 P 0525 2 EFN = CNF$C SYNCH_EFN,
344 P 0526 2 IOSB = IRB [IRB$W_IOSB], ! Address of I/O status block
345 P 0527 2 P2 = NCB_DESC); ! Address of network control block
346 0528 2
347 0529 2 IF .STATUS ! If successfully submitted,
348 0530 2 THEN
349 0531 2 STATUS = .IRB [IRB$W_IOSB]; ! then pick up QIO final status
350 0532 2
351 0533 2 IF NOT .STATUS ! If error starting up link
352 0534 2 THEN
353 0535 2 BEGIN
354 0536 2 SIGNAL (CNF$ LINK, 0, .STATUS); ! then report the error
355 0537 2 CNF$CLOSE_REQUEST_LINK(.IRB); ! and deallocate the storage
356 0538 2 RETURN;
357 0539 2 END;
358 0540 2
359 0541 2
360 0542 2 CNF$SOLICIT_REQUEST (.IRB); ! Issue a QIO for an incoming request record
361 0543 2
362 0544 1 END; ! End routine open_request_link

```

```

.PSECT $SPLITS,NOWRT,NOEXE,2
45 43 41 52 54 00170 P.AAZ: .ASCII \TRACE\
00175 .BLKB 3
00000005 00178 P.AAY: .LONG 5
00000000' 0017C .ADDRESS P.AAZ
69 6C 5F 74 73 65 75 71 65 72 5F 6E 65 70 6F 00180 P.ABB: .ASCII \open_request_link\
6B 6E 0018F
00191 .BLKB 3
00000011 00194 P.ABA: .LONG 17
00000000' 00198 .ADDRESS P.ABB
00 00 00 3A 54 45 4E 5F 0019C P.ABD: .ASCII \NET:\<0><0><0>
010E0005 001A4 P.ABC: .LONG 17694725
00000000' 001A8 .ADDRESS P.ABD

.EXTRN SYSS$ASSIGN, SYSS$QIOW
.PSECT $CODE$,NOWRT,2
000C 0000 OPEN_REQUEST LINK:
5E .WORD Save R2,R3 : 0460
0000' CF 9F 00005 .SUBL2 #8, SP : 0487
0000' CF 9F 00009 .PUSHAB P.ABA : 0486
01 DD 0000D .PUSHL #1

```

0000G	CF	03	FB	0000F	CALLS	#3, CNF\$TRACE	
	52	04	AC	D0 00014	MOVL	IRB, R2	0493
	6E	24	A2	98 00018	CVTBL	36(R2), NCB_DESC	
04	AE	25	A2	9E 0001C	MOVAB	37(R2), NCB_DESC+4	0494
	50	24	A2	98 00021	CVTBL	36(R2), R0	0500
25	A2		2F	3A 00025	LOCC	#47, R0, 37(R2)	
	50		02	12 0002A	BNEQ	1\$	
			51	D4 0002C	CLRL	R1	
			51	D5 0002E	TSTL	PTR	0502
			09	13 0003C	BEQL	2\$	
	50	25	A2	9E 00032	MOVAB	37(R2), R0	0504
24	A2		50	83 00036	SUBB3	R0, PTR, 36(R2)	
	51		A1	94 0003B	CLRB	3(PTR)	0510
			7E	7C 0003E	CLRQ	-(SP)	0513
		0A	A2	9F 00040	PUSHAB	10(R2)	
		0000'	CF	9F 00043	PUSHAB	P.ABC	
00000000G	00		04	FB 00047	CALLS	#4, SYSS\$ASSIGN	
	53		50	D0 0004E	MOVL	R0, STATUS	
	0C		53	E8 00051	BLBS	STATUS, 3\$	0515
			53	DD 00054	PUSHL	STATUS	0518
			7E	D4 00056	CLRL	-(SP)	
		00000000G	8F	DD 00058	PUSHL	#CNF\$_CHAN	
			38	11 0005E	BRB	5\$	
			7E	7C 00060	CLRQ	-(SP)	0527
			7E	7C 00062	CLRQ	-(SP)	
		10	AE	9F 00064	PUSHAB	NCB_DESC	
			7E	7C 00067	CLRQ	-(SP)	
			7E	D4 00069	CLRL	-(SP)	
		0C	A2	9F 0006B	PUSHAB	12(R2)	
			32	DD 0006E	PUSHL	#50	
	7E	0A	A2	32 00070	CVTWL	10(R2), -(SP)	
00000000G	00	00000000G	8F	DD 00074	PUSHL	#CNF\$_SYNCH_EFN	
	53		0C	FB 0007A	CALLS	#12, SYSS\$QIO	
	07		50	D0 00081	MOVL	R0, STATUS	
	53		53	E9 00084	BLBC	STATUS, 4\$	0529
	19	0C	A2	32 00087	CVTWL	12(R2), STATUS	0531
			53	E8 0008B	BLBS	STATUS, 6\$	0533
			53	DD 0008E	PUSHL	STATUS	0536
			7E	D4 00090	CLRL	-(SP)	
00000000G	00	00000000G	8F	DD 00092	PUSHL	#CNF\$_LINK	
			05	FB 00098	CALLS	#3, LIB\$SIGNAL	
			52	DD 0009F	PUSHL	R2	0537
0000V	CF		01	FB 000A1	CALLS	#1, CNF\$CLOSE_REQUEST_LINK	
				04 000A6	RET		0535
			52	DD 000A7	PUSHL	R2	0542
0000V	CF		01	FB 000A9	CALLS	#1, CNF\$SOLICIT_REQUEST	
				04 000AE	RET		0544

; Routine Size: 175 bytes, Routine Base: %00025 + 010A

```

364 0545 1 %SBTTL 'CNF$CLOSE_REQUEST_LINK'
365 0546 1 GLOBAL ROUTINE CNF$CLOSE_REQUEST_LINK (IRB): NOVALUE =
366 0547 1
367 0548 1 ----
368 0549 1
369 0550 1     Close the logical link for incoming request records.
370 0551 1
371 0552 1     Inputs:
372 0553 1
373 0554 1         irb = Address of incoming request block
374 0555 1
375 0556 1     Outputs:
376 0557 1
377 0558 1     None
378 0559 1
379 0560 1     Value:
380 0561 1     Signal any errors
381 0562 1 ----
382 0563 1
383 0564 2 BEGIN
384 0565 2
385 0566 2 MAP
386 0567 2     IRB:          REF BBLOCK;          ! Address of incoming request channel
387 0568 2
388 0569 2 LOCAL
389 0570 2     FREE_BNR : REF BBLOCK,
390 0571 2     LENGTH,
391 0572 2     STATUS;
392 0573 2
393 0574 2 CNF$TRACE (DBG$C_TRACE, $DESCRIPTOR('TRACE'),
394 0575 2     $DESCRIPTOR ('CNF$CLOSE_REQUEST_LINK'));
395 0576 2
396 0577 2 IF .IRB [IRB$W_CHAN] NEQ 0          ! If channel was assigned,
397 0578 2 THEN
398 0579 2     BEGIN
399 0580 2     STATUS = $DASSGN (CHAN = .IRB [IRB$W_CHAN]); ! Deassign network channel
400 0581 2
401 0582 2     IF NOT .STATUS          ! If error detected,
402 0583 2     THEN
403 0584 2     SIGNAL (CNF$_CHAN, 0, .STATUS); ! then report error
404 0585 2
405 0586 2     END;
406 0587 2 REMQUE (.IRB, STATUS);          ! Remove from linked list
407 0588 2
408 0589 2 !
409 0590 2 !     If there are Buffered NICE responses in the IRB, deallocate them.
410 0591 2 !
411 0592 2 FREE_BNR = .IRB [IRB$B_BNR_FLINK];
412 0593 2 WHILE .FREE_BNR NEQ IRB [IRB$B_BNR_FLINK] DO
413 0594 2     BEGIN
414 0595 2     REMQUE (.FREE_BNR, STATUS);
415 0596 2     EXECUTE (CNF$FREE_VM (FREE_BNR [BNR$W_LENGTH], FREE_BNR [BNR$B_ADDRESS] ));
416 0597 2     EXECUTE (CNF$FREE_VM (%REF (BNR$C_LENGTH), FREE_BNR));
417 0598 2     FREE_BNR = .IRB [IRB$B_BNR_FLINK];
418 0599 2     END;
419 0600 2 LENGTH = .IRB [IRB$W_SIZE];          ! Get size of block
420 0601 2 EXECUTE (CNF$FREE_VM (LENGTH, IRB)); ! Deallocate storage

```



: 421  
: 422  
0602 2  
0603 1 END;

! End routine cnf\$close\_request\_link

```

.PSECT $SPLITS,NOWRT,NOEXE,2
      45 43 41 52 54 001AC P.ABF: .ASCII \TRACE\
      001B1 .BLKB 3
      00000005 001B4 P.ABE: .LONG 5
      00000000' 001B8 .ADDRESS P.ABF
45 55 51 45 52 5F 45 53 4F 4C 43 24 46 4E 43 001BC P.ABH: .ASCII \CNF$CLOSE_REQUEST_LINK\
      4B 4E 49 4C 5F 54 53 001CB
      001D2 .BLKB 2
      00000016 001D4 P.ABG: .LONG 22
      00000000' 001D8 .ADDRESS P.ABH

.EXTRN SYSSDASSGN
.PSECT $CODE$,NOWRT,2
      54 0000G CF 9E 00002 .ENTRY CNF$CLOSE_REQUEST_LINK, Save R2,R3,R4
      5E 0C C2 00007 MOVAB CNF$FREE_VM, R4
      0000' CF 9F 0000A SUBL2 #12, SP
      0000' CF 9F 0000E PUSHAB P.ABG
      01 DD 00012 PUSHAB P.ABE
      0000G CF 03 FB 00014 PUSHL #1
      52 04 AC D0 00019 CALLS #3, CNF$TRACE
      0A A2 B5 0001D MOVL IRB, R2
      22 13 00C20 TSTW 10(R2)
      7E 0A A2 32 00022 BEQL 1$
      00000000G 00 01 FB 00026 CVTWL 10(R2), -(SP)
      53 50 D0 0002D CALLS #1, SYSSDASSGN
      11 53 E8 00030 MOVL R0, STATUS
      53 DD 00033 BLBS STATUS, 1$
      7E D4 00035 PUSHL STATUS
      00000000GG 8F DD 00037 CLRL -(SP)
      00 03 FB 0003D PUSHL #CNF$ CHAN
      53 62 OF 00044 1$: CALLS #3, LIB$SIGNAL
      50 04 AC D0 00047 REMQUE (R2), STATUS
      04 AE 14 A0 D0 0004B MOVL IRB, R0
      52 04 AC 14 C1 00050 MOVL 20(R0), FREE_BNR
      52 04 AE D1 00055 2$: ADDL3 #20, IRB, R2
      2F 13 00059 CMPL FREE_BNR, R2
      53 04 BE OF 0005B BEQL 3$
      7E 04 AE 0C C1 0005F REMQUE @FREE_BNR, STATUS
      7E 08 AE 08 C1 00064 ADDL3 #12, FREE_BNR, -(SP)
      64 02 FB 00069 ADDL3 #8, FREE_BNR, -(SP)
      2D 50 E9 0006C CALLS #2, CNF$FREE_VM
      04 AE 9F 0006F BLBC STATUS, 4$
      04 AE 10 D0 00072 PUSHAB FREE_BNR
      04 AE 9F 00076 MOVL #16, -4(SP)
      64 02 FB 00079 PUSHL 4(SP)
      1D 50 E9 0007C CALLS #2, CNF$FREE_VM
      52 04 AC 14 C1 0007F BLBC STATUS, 4$
      04 AE 62 D0 00084 ADDL3 #20, IRB, R2
      MOVL (R2), FREE_BNR

```



```

: 424 0604 1 %SBTTL 'shutdown'
: 425 0605 1 ROUTINE SHUTDOWN: NOVALUE =
: 426 0606 1
: 427 0607 1 ---
: 428 0608 1
: 429 0609 1 This routine is called when the network is shutting down to
: 430 0610 1 gracefully close all incoming links so that NICONFIG goes away quietly.
: 431 0611 1
: 432 0612 1 Inputs:
: 433 0613 1
: 434 0614 1 None
: 435 0615 1
: 436 0616 1 Outputs:
: 437 0617 1
: 438 0618 1 None
: 439 0619 1
: 440 0620 1 Effect:
: 441 0621 1
: 442 0622 1 Disabling all surveillance will cause NICONFIG to terminate
: 443 0623 1 ---
: 444 0624 1
: 445 0625 2 BEGIN
: 446 0626 2
: 447 0627 2 LOCAL
: 448 0628 2 PTR: REF BBLOCK, ! Pointer to irb block
: 449 0629 2 NEXT_PTR;
: 450 0630 2
: 451 0631 2 PTR = .CNF$GQ_IRBLST; ! Start at first link context block
: 452 0632 2 WHILE .PTR NEQ CNF$GQ_IRBLST ! Until end of linked list,
: 453 0633 2 DO
: 454 0634 3 BEGIN
: 455 0635 3 NEXT_PTR = .PTR [IRB$L_LINK];
: 456 0636 3 CNF$[CLOSE_REQUEST_LINK(.PTR)]; ! Abort the incoming link
: 457 0637 3 PTR = .NEXT_PTR; ! and link to next in chain
: 458 0638 3 END;
: 459 0639 2
: 460 0640 2 PTR = .CNF$GQ_CIRSURLST; ! Start at first circuit block
: 461 0641 2 WHILE .PTR NEQ CNF$GQ_CIRSURLST ! Until end of linked list,
: 462 0642 2 DO
: 463 0643 3 BEGIN
: 464 0644 3 CNF$DISABLE_SURVEIL (.PTR); ! and delete the circuit
: 465 0645 3 PTR = .PTR [CIR$L_LINK]; ! Link to next one
: 466 0646 2 END;
: 467 0647 2
: 468 0648 2 CNF$_SURVEILLANCE_SET = FALSE; ! So it will die quietly
: 469 0649 2
: 470 0650 1 END; ! End routine shutdown

```

```

                                000C 0000 SHUTDOWN:
                                .WORD      Save R2,R3
52      0000G  CF  D0 00002      MOVL      CNF$GQ_IRBLST, PTR      : 0605
50      0000G  CF  9E 00007 1$:  MOVAB     CNF$GQ_IRBLST, R0      : 0631
50      0000G  52  D1 0000C      CMPL     PTR, R0          : 0632
                                :

```

			OF	13	0000F		BEQL	2\$		
	53		62	D0	00011		MOVL	(PTR), NEXT_PTR	:	0635
			52	DD	00014		PUSHL	PTR	:	0636
FF48	CF		01	FB	00016		CALLS	#1, CNF\$CLOSE_REQUEST_LINK	:	
	52		53	D0	0001B		MOVL	NEXT_PTR, PTR	:	0637
			E7	11	0001E		BRB	1\$	:	0632
	52	0000G	CF	D0	00020	2\$:	MOVL	CNF\$GQ_CIRSURLST, PTR	:	0640
	50	0000G	CF	9E	00025	3\$:	MOVAB	CNF\$GQ_CIRSURLST, R0	:	0641
	50		52	D1	0002A		CML	PTR, R0	:	
			0C	13	0002D		BEQL	4\$	:	
			52	DD	0002F		PUSHL	PiR	:	0644
0000G	CF		01	FB	00031		CALLS	#1, CNF\$DISABLE_SURVEIL	:	
	52		62	D0	00036		MOVL	(PTR), PTR	:	0645
			EA	11	00039		BRB	3\$	:	0641
		0000G	CF	D4	0003B	4\$:	CLRL	CNF\$B_SURVEILLANCE_SET	:	0648
			04	0003F			RET		:	0650

; Routine Size: 64 bytes, Routine Base: \$CODE\$ + 0256

```

472 0651 1 %SBTTL 'CNF$SOLICIT REQUEST'
473 0652 1 GLOBAL ROUTINE CNF$SOLICIT_REQUEST (irb): NOVALUE =
474 0653 1
475 0654 1 ---
476 0655 1
477 0656 1 This routine is called to obtain requests from the incoming
478 0657 1 logical link. Each incoming request is immediately queued
479 0658 1 to the disposal queue for the appropriate action.
480 0659 1 It is first called directly by OPEN_REQUEST_LINK which is executing
481 0660 1 off the work queue, and thereafter calls are placed on the work queue
482 0661 1 by the AST routine, REQUEST_RECEIVED.
483 0662 1
484 0663 1 Inputs:
485 0664 1
486 0665 1     irb = Address of incoming request block
487 0666 1
488 0667 1 Outputs:
489 0668 1
490 0669 1     None
491 0670 1 ---
492 0671 1
493 0672 2 BEGIN
494 0673 2
495 0674 2 MAP
496 0675 2     IRB:          REF BBLOCK;          ! Address of incoming request block
497 0676 2
498 0677 2 LOCAL
499 0678 2     STATUS;
500 0679 2
501 0680 2 CNF$TRACE (DBG$C TRACE, $DESCRIPTOR('TRACE'),
502 0681 2     $DESCRIPTOR ('cnf$solicit_request'));
503 0682 2
504 P 0683 2 STATUS = $QIO(FUNC = IOS_READVBLK,      ! Get request from incoming link
505 P 0684 2     CHAN = .IRB [IRB$W_CHAN],
506 P 0685 2     EFN = CNF$C ASYNCH_EFN,
507 P 0686 2     IOSB = IRB [IRB$W_IOSB], ! Address of I/O status block
508 P 0687 2     ASTADR = REQUEST_RECEIVED, ! Address of completion routine
509 P 0688 2     ASTPRM = .IRB,           ! Giving irb as routine parameter
510 P 0689 2     P1 = IRB [IRB$T_REQUEST], ! Address of request buffer
511 0690 2     P2 = IRB$C_MAXRQSTLEN); ! Length of request buffer
512 0691 2
513 0692 2 IF NOT .STATUS
514 0693 2 THEN
515 0694 3     BEGIN
516 0695 3     IF (.STATUS NEQ SSS_LINKABORT) AND ! Don't signal a fatal error just because
517 0696 3     (.STATUS NEQ SSS_LINKEXIT)      ! the partner went away.
518 0697 3     THEN
519 0698 3     SIGNAL (CNF$_LINK, 0, .STATUS); ! then report error
520 0699 3
521 0700 3 WKQ$ADD WORK ITEM( CNF$CLOSE_REQUEST_LINK, .IRB);
522 0701 3 CNF$TRACE (DBG$C TRACE, $DESCRIPTOR('TRACE'),
523 0702 3     $DESCRIPTOR ('cnf$solicit_request %CNF$CLOSE_REQUEST_LINK'));
524 0703 2 END;
525 0704 2
526 0705 1 END;          ! End routine cnf$solicit_request

```

```

.PSECT $SPLITS$,NOWRT,NOEXE,2
      45 43 41 52 54 001DC P.ABJ: .ASCII \TRACE\
      001E1 .BLKB 3
      00000005 001E4 P.ABI: .LONG 5
      00000000' 001E8 .ADDRESS P.ABJ
71 65 72 5F 74 69 63 69 6C 6F 73 24 66 6E 63 001EC P.ABL: .ASCII \cnf$solicit_request\
4F 4C 43 24 46 4F 43 25 25 20 20 74 73 65 75 001FB .BLKB 1
      001FF .LONG 19
      00000013 00200 P.ABK: .ADDRESS P.ABL
      00000000' 00204 .ADDRESS P.ABL
      45 43 41 52 54 00208 P.ABN: .ASCII \TRACE\
      0020D .BLKB 3
      00000005 00210 P.ABM: .LONG 5
      00000000' 00214 .ADDRESS P.ABN
71 65 72 5F 74 69 63 69 6C 6F 73 24 66 6E 63 00218 P.ABP: .ASCII \cnf$solicit_request %%CNF$CLOSE_REQUEST\
4F 4C 43 24 46 4F 43 25 25 20 20 74 73 65 75 00227 .BLKB 3
      00236 .LONG 45
      4B 4E 49 4C 5F 00240 .ADDRESS P.ABP
      00245 P.ABO: .ASCII \_LINK\
      0000002D 00248 .BLKB 3
      00000000' 0024C .LONG 45
      .ADDRESS P.ABP

```

```

.PSECT $CODE$,NOWRT,2
      0004 00000 .ENTRY CNF$SOLICIT_REQUEST, Save R2
      0000' CF 9F 00002 PUSHAB P.ABK
      0000' CF 9F 00006 PUSHAB P.ABI
      0000G CF 01 DD 0000A PUSHL #1
      03 FB 0000C CALLS #3, CNF$TRACE
      7E 7C 00011 CLRQ -(SP)
      7E 7C 00013 CLRQ -(SP)
      7E FA 8F 9A 00015 MOVZBL #250, -(SP)
      52 04 AC D0 00019 MOVL IRB, R2
      65 A2 9F 0001D PUSHAB 101(R2)
      52 DD 00020 PUSHL R2
      0000V CF 9F 00022 PUSHAB REQUEST_RECEIVED
      0C A2 9F 00026 PUSHAB 12(R2)
      31 DD 00029 PUSHL #49
      7E 0A A2 32 0002B CVTWL 10(R2), -(SP)
      00000000G 8F DD 0002F PUSHL #CNF$C_ASYNC_EFN
      00 0C FB 00035 CALLS #12, SYS$QIO
      3D 50 E8 0003C BLBS STATUS, 2$
      000020E4 8F 50 D1 0003F CMPL STATUS, #8420
      000020F4 8F 1A 13 00046 BEQL 1$
      50 D1 00048 CMPL STATUS, #8436
      11 13 0004F BEQL 1$
      50 DD 00051 PUSHL STATUS
      7E D4 00053 CLRL -(SP)
      00000000G 8F DD 00055 PUSHL #CNF$ LINK
      00 03 FB 00058 CALLS #3, LTB$SIGNAL
      52 DD 00062 1$: PUSHL R2
      FEBB CF 9F 00064 PUSHAB CNF$CLOSE_REQUEST LINK
      0000G CF 02 FB 00068 CALLS #2, WKQ$ADD_WORK_ITEM

```

```

: 0652
: 0681
: 0680
: 0690
: 0692
: 0695
: 0696
: 0698
: 0700

```

CNFINTRPT  
V04-000

DECnet Ethernet Configurator Module  
CNF\$SOLICIT\_REQUEST

N 10  
16-Sep-1984 02:03:38  
14-Sep-1984 12:49:50

VAX-11 Bliss-32 V4.0-742  
[NICNF.SRC]CNFINTRPT.B32;1

Page 21  
(8)

CN  
VO

0000:	CF	9F	0006D	PUSHAB	P.ABO
0000:	CF	9F	00071	PUSHAB	P.ABM
	01	DD	00075	PUSHL	#1
0000G	CF	03	FB	00077	CALLS #3, CNF\$TRACE
		04	0007C	2\$:	RET

: 0702  
: 0701  
:  
: 0705

: Routine Size: 125 bytes, Routine Base: \$CODE\$ + 0296

```

528 0706 1 %SBTTL 'request_received'
529 0707 1 ROUTINE REQUEST_RECEIVED (irb): NOVALUE =
530 0708 1
531 0709 1 ----
532 0710 1
533 0711 1 This AST routine is called when a new request has come
534 0712 1 in over the logical link. The request is queued to the
535 0713 1 work queue.
536 0714 1
537 0715 1 Inputs:
538 0716 1
539 0717 1 irb = Address of incoming request block
540 0718 1
541 0719 1 Outputs:
542 0720 1
543 0721 1 None
544 0722 1 ----
545 0723 1
546 0724 2 BEGIN
547 0725 2
548 0726 2 MAP
549 0727 2 IRB: REF BBLOCK; ! Address of incoming request block
550 0728 2
551 0729 2 CNF$TRACE (DBG$C_TRACE, $DESCRIPTOR('TRACE'),
552 0730 2 $DESCRIPTOR('request_received'));
553 0731 2
554 0732 2 IF NOT .IRB [IRB$W_IOSB] ! If error from QIO,
555 0733 2 THEN
556 0734 2 BEGIN
557 0735 2 IF (.IRB [IRB$W_IOSB] NEQ SSS_LINKABORT) AND
558 0736 2 (.IRB [IRB$W_IOSB] NEQ SSS_LINKEXIT)
559 0737 2 THEN
560 0738 2 SIGNAL (CNF$LINK, 0, .IRB [IRB$W_IOSB]); ! then report the error
561 0739 2 WKQ$ADD_WORK_ITEM (CNF$CLOSE_REQUEST_LINK, .IRB); ! close the link until re-established
562 0740 2 RETURN;
563 0741 2 END;
564 0742 2
565 0743 2 !
566 0744 2 ! Log the contents of the incoming message
567 0745 2 !
568 0746 2 BEGIN
569 0747 2 LOCAL DATA_DSC : BBLOCK [DSC$C_S_BLN];
570 0748 2
571 0749 2 DATA_DSC = 0;
572 0750 2 DATA_DSC [DSC$W_LENGTH] = .IRB [IRB$W_IOSB1];
573 0751 2 DATA_DSC [DSC$A_POINTER] = IRB [IRB$T_REQUEST];
574 0752 2 CNF$LOG_DATA (DBG$C_NICE, $DESCRIPTOR('NICE received'), 0, DATA_DSC);
575 0753 2 END;
576 0754 2
577 0755 2 WKQ$ADD_WORK_ITEM (CNF$PROCESS_REQUEST, ! Queue request
578 0756 2 .IRB);
579 0757 2
580 0758 1 END; ! End routine request_received

```

.PSECT SPLITS,NOWRT,NOEXE,2





CNFINTRPT  
V04-0 0

DECnet Ethernet Configurator Module  
request\_received

D 11  
16-Sep-1984 02:03:38  
14-Sep-1984 12:49:50

VAX-11 Bliss-32 V4.0-742  
[NICNF.SRC]CNFINTRPT.832;1

Page 24  
(10)

: 582 0759 1 END  
: 583 0760 0 ELUDOM

: End of module CNFINTRPT

.EXTRN LIB\$SIGNAL

PSECT SUMMARY

Name	Bytes	Attributes
\$OWNS	8	NOVEC, WRT, RD, NOEXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)
\$SPLITS	656	NOVEC, NOWRT, RD, NOEXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)
\$CODES	896	NOVEC, NOWRT, RD, EXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)

Library Statistics

File	Total	Symbols Loaded	Percent	Pages Mapped	Processing Time
-\$255\$DUA28:[SYSLIB]STARLET.L32;1	9776	16	0	581	00:01.0
-\$255\$DUA28:[SHRLIB]NET.L32;1	1279	0	0	63	00:00.9
-\$255\$DUA28:[SHRLIB]NMALIBRY.L32;1	887	0	0	47	00:00.8

COMMAND QUALIFIERS

BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS\$:CNFINTRPT/OBJ=OBJ\$:CNFINTRPT MSRC\$:CNFINTRPT/UPDATE=(ENHS:CNFINTRPT)

: Size: 896 code + 664 data bytes  
: Run Time: 00:19.2  
: Elapsed Time: 00:41.8  
: Lines/CPU Min: 2370  
: Lexemes/CPU-Min: 20292  
: Memory Used: 116 pages  
: Compilation Complete

0279 AH-BT13A-SE  
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION  
CONFIDENTIAL AND PROPRIETARY

The image displays a grid of terminal window outputs from a VAX/VMS system. Each window shows a different screen from the 'AH-BT13A-SE' menu.

Visible titles and labels include:

- NETDEF
- NETTRN LIS
- NETTREE LIS
- SERVER LIS
- NICNF
- NICONFIG MAP
- CNFDEF SDL
- CNFDEF LIS
- CNFMAIN LIS
- CNRFREQS LIS
- CNFINTRPT LIS
- CNFADDEF SDL
- CNFPREFX REQ
- CNFM5G LIS

The outputs consist of lists of data, tables, and configuration information, typical of a network management interface.