


```

RRRRRRRR      EEEEEEEEEEE MM      MM      XX      XX      PPPPPPPP      000000      RRRRRRRR      TTTTTTTTTT      DDDDDDDD
RRRRRRRR      EEEEEEEEEEE MM      MM      XX      XX      PPPPPPPP      000000      RRRRRRRR      TTTTTTTTTT      DDDDDDDD
RR      RR      EE      MMMM      MMMM      XX      XX      PP      PP      00      00      RR      RR      TT      DD      DD
RR      RR      EE      MMMM      MMMM      XX      XX      PP      PP      00      00      RR      RR      TT      DD      DD
RR      RR      EE      MM      MM      MM      XX      XX      PP      PP      00      00      RR      RR      TT      DD      DD
RR      RR      EE      MM      MM      MM      XX      XX      PP      PP      00      00      RR      RR      TT      DD      DD
RRRRRRRR      EEEEEEEEEEE MM      MM      XX      XX      PPPPPPPP      00      00      RRRRRRRR      TT      DD      DD
RRRRRRRR      EEEEEEEEEEE MM      MM      XX      XX      PPPPPPPP      00      00      RRRRRRRR      TT      DD      DD
RR      RR      EE      MM      MM      XX      XX      PP      PP      00      00      RR      RR      TT      DD      DD
RR      RR      EE      MM      MM      XX      XX      PP      PP      00      00      RR      RR      TT      DD      DD
RR      RR      EE      MM      MM      XX      XX      PP      PP      00      00      RR      RR      TT      DD      DD
RR      RR      EE      MM      MM      XX      XX      PP      PP      00      00      RR      RR      TT      DD      DD
RR      RR      EEEEEEEEEEE MM      MM      XX      XX      PP      000000      RR      RR      TT      DDDDDDDD
RR      RR      EEEEEEEEEEE MM      MM      XX      XX      PP      000000      RR      RR      TT      DDDDDDDD

```

```

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
LLLLLLLLLLLL IIIIII SSSSSSSS
LLLLLLLLLLLL IIIIII SSSSSSSS

```

```
0000 1 .TITLE REXPORT - PROVIDE THE ACP TRANSPORT MECHANISM
0000 2 .IDENT 'V04-000'
0000 3
0000 4 :*****
0000 5 :*
0000 6 :* COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0000 7 :* DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0000 8 :* ALL RIGHTS RESERVED.
0000 9 :*
0000 10 :* THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0000 11 :* ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0000 12 :* INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0000 13 :* COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0000 14 :* OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0000 15 :* TRANSFERRED.
0000 16 :*
0000 17 :* THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0000 18 :* AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0000 19 :* CORPORATION.
0000 20 :*
0000 21 :* DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0000 22 :* SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0000 23 :*
0000 24 :*
0000 25 :*****
0000 26
0000 27 :++
0000 28 : FACILITY: REMOTE I/O ACP
0000 29
0000 30 : ABSTRACT:
0000 31 : This module handles the mechanism for transporting
0000 32 : ACP messages.
0000 33
0000 34 : This version of the module uses:
0000 35 : *****
0000 36 : DECnet
0000 37 : *****
0000 38 : as its transport mechanism.
0000 39
0000 40 : ENVIRONMENT:
0000 41 : MODE = KERNEL
0000 42 :--
```

```
0000 44 :  
0000 45 : AUTHOR: SCOTT G. DAVIS, CREATION DATE: 11-JUL-79  
0000 46 :  
0000 47 : MODIFIED BY:  
0000 48 :  
0000 49 : V03-001 JLV0218 Jake VanNoy 7-Jan-1982  
0000 50 : Add code to declare object type 42. TSA requires  
0000 51 : this new object type. Store object type requested  
0000 52 : in UCB.  
0000 53 :  
0000 54 : V02-005 SPF0050 Steve Forgey 31-Dec-1981  
0000 55 : Store logical link number in RT UCB.  
0000 56 :  
0000 57 : V02-004 DJD2002 Darrell Duffy 2-May-1981  
0000 58 : Fix to ignore link error mailbox messages.  
0000 59 :  
0000 60 : V02-003 DJD2001 Darrell Duffy 27-Feb-1981  
0000 61 : Allow for max links at init time.  
0000 62 :  
0000 63 : V02-002 SGD2001 Scott G. Davis 23-Sep-1980  
0000 64 : Remove multiple devices, ignore unknown messages
```

```

0000 66
0000 67 : INCLUDE FILES:
0000 68 :
0000 69      $AQBDEF
0000 70      $DIBDEF
0000 71      $MSGDEF
0000 72      $NFBDEF
0000 73      $REMDEF
0000 74      $UCBDEF
0000 75      $RTTUCBEXT
0000 76
0000 77 :
0000 78 : MACROS:
0000 79 :
0000 80
0000 81 :
0000 82 : EQUATED SYMBOLS:
0000 83 :

```

```

00000096 0000 84 MBX_MSG_LTH = 150 ; Size of a mailbox message
00000017 0000 85 RT_OBJ = 23 ; Object number for network cmd term
0000002A 0000 86 RT_OBJ2 = 42 ; Object number for TSA remote term
0000 87
0000 88 :
0000 89 : OWN STORAGE:
0000 90 :

```

```

00000000 91 .PSECT REM_XP_PURE,NOWRT,NOEXE
0000 92

```

```

50 43 41 4D 45 52 00000008'010E0000' 0000 93 MBXDESC: .ASCID /REMACP_MBX/ ; Logical name of mailbox
      58 42 4D 5F 000E
3A 54 45 4E 5F 0000001A'010E0000' 0012 94 NETDESC: .ASCID /_NET:/ ; Device name for channel assignment
00000005' 001F 95 NFBDESC: .LONG 5 ; NFB (FIB) descriptor
00000000' 0023 96 .ADDRESS NFB
00000074' 0027 97 DIBDESC: .LONG DIB$K_LENGTH ; Descriptor for device info bfr
000000A7' 002B 98 .ADDRESS DIBBUF
002F 99

```

```

00000000 100
00000000 101 .PSECT REM_XP_IMPURE NOSHR,NOEXE,RD,WRT
0000 102

```

```

16 0000 103 NFB: .BYTE NFB$C_DECLOBJ ; Function code (DECLARE OBJ)
00000017 0001 104 NFB_OBJ: .LONG RT_OBJ ; Object number
0005 105 MBX_Q_IOSB: ; I/O status block
0005 106 NCBDESC: ; Multiplex with MBX IOSB
0000000D 0005 107 .BLKQ 1 ; Status of I/O completion
000D 108 MBX_T_BUFFER: ; Buffer for mailbox message
0000000F 000D 109 .BLKW 1 ; Message identification
00000011 000F 110 ; Not used
0011 111 MBX_T_DATA: ; On connect initiates, et cetera
000000A7 0011 112 .BLKB MBX_MSG_LTH ; Leave room for message
00000118 00A7 113 DIBBUF: .BLKB DIB$K_LENGTH ; Device information buffer

```



```

0072 172
0072 173
0072 174 :+
0072 175 : REM$MBX_READ - This routine puts a read out on the mailbox for
0072 176 : process termination and inbound connect notifications
0072 177 :-
0072 178 REM$MBX_READ:: : Post read to mailbox
0072 179 $QIO_S CHAN = W^REMSGW MBX CHAN,-
0072 180 FUNC = S^#IOS$ READVBLK,-
0072 181 ASTADR = B^MBX_AST,-
0072 182 IOSB = W^MBX_Q_IOSB,-
0072 183 P1 = W^MBX_T_BUFFER,-
0072 184 P2 = #MBX_MSG_LTH
SC 50 E9 009A 185 BLBC R0,MBX_FAILURE ; If LBC error
05 009D 186 RSB ; return
009E 187 :++
009E 188 :
009E 189 : MBX_AST - This routine services process terminations
009E 190 : and inbound connect notifications.
009E 191 :-
009E 192 :-
OFFC 009E 193 MBX_AST: .WORD ^M<R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>
00' 0005'CF B1 00A0 194 CMPW W^MBX_Q_IOSB,S^#SS$_ABORT ; Was the I/O cancelled?
34 13 00A5 195 BEQL 50$ ; If EQL yes
0000'8F 0005'CF B1 00A7 196 CMPW W^MBX_Q_IOSB,#SS$_CANCEL ; Try this code, too
28 13 00AE 197 BEQL 50$ ; If EQL yes
59 000D'CF 9E 00B0 198 MOVAB W^MBX_T_BUFFER,R9 ; Point to message buffer
56 89 3C 00B5 199 MOVZWL (R9)+,R6 ; Save mailbox message code
57 89 B0 00B8 200 MOVW (R9)+,R7 ; Get device dev unit number
50 69 9A 00BB 201 MOVZBL (R9),R0 ; Get length of device name
00BE 202 ; Counted ASCII string
59 50 D6 00BE 203 INCL R0
59 50 C0 00C0 204 ADDL R0,R9 ; Pass device name string
00C3 205 :
00C3 206 : Determine device index
00C3 207 :
5B 0000'CF 9A 00C3 208 MOVZBL W^REMSGB_MAXLINKS,R11 ; Init count
00C3 209 10$:
0000'DF4B 57 B1 00C8 210 CMPW R7,@W^REM$GL_MBX_UNITS[R11] ; Match?
03 13 00CE 211 BEQL 20$ ; If EQL yes
F5 5B F5 00D0 212 SOBGTR R11,10$ ; Loop
00D3 213 :*****
00D3 214 : FATAL - unit number not found
00D3 215 :*****
00D3 216 :
00D3 217 20$:
00D3 218 :
00D3 219 : Dispatch
00D3 220 :
07 10 00D3 221 BSBB 100$ ; Allow for easy return
00D5 222 ; by action routines
5A 01 90 00D5 223 MOVB #REM$C MBX_READ,R10 ; Set up for a mailbox read
FF25' 30 00D8 224 BSBW REM$ENQUEUE ; Queue the request
04 00DB 225 50$: RET ; Done
00DC 226 :
00DC 227 : Dispatch to action routine matching logical link event
00DC 228 :

```

Mac

-\$2
-\$2
TOT
217
The
MAC

```
00DC 229 ; R9 points to mailbox data
00DC 230 ; R11 contains device index
00DC 231 ;
C0DC 232 100$: ; Entry point
00DC 233 $DISPATCH R6,- ; Case on event code
00DC 234
00DC 235 <-; event action
00DC 236
00DC 237 <MSG$_CONNECT, CONN_REQ> -; Connect request received
00DC 238 <MSG$_ABORT, LINK_BROKEN> -; Link broken
00DC 239 <MSG$_DISCON, LINK_BROKEN> -; Link broken
00DC 240 <MSG$_EXIT, LINK_BROKEN> -; Link broken
00DC 241 <MSG$_PATHLOST, LINK_BROKEN> -; Link broken
00DC 242 <MSG$_PROTOCOL, LINK_BROKEN> -; Link broken
00DC 243 <MSG$_THIRDPARTY, LINK_BROKEN> -; Link broken
00DC 244 <MSG$_NETSHUT, NET_SHUTDOWN> -; Network shutting down
00DC 245 >
05 00F8 246 RSB ; Ignore other messages
00F9 247 MBX_FAILURE:
00F9 248 BUG_CHECK ACPMBFAIL,FATAL
```



```

00FD 250
00FD 251 :++
00FD 252 :
00FD 253 : CONFIRM_AST - This is the AST routine for connect accepts.
00FD 254 : Issue a read over the link.
00FD 255 :
00FD 256 :
00FD 257 :--
00FD 258 :
50 5B 04 AC OFFC 00FD 259 CONFIRM_AST: .WORD ^M<R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>
0000'DF4B 7D 00FF 260 -MOVL 4(AP),R11 ; Get the device index
04 50 E8 0103 261 MOVQ @W^REMSGL_RIOSBVEC[R11],R0 ; Get the IOSB
0109 262 BLBS R0,5$ ; If LBS I/O was OK
010C 263 :
010C 264 : I/O failed
010C 265 :
FEF1' 30 010C 266 BSBW REM$KILL_UCB ; Clean up the channel
04 010F 267 RET ; Done
0110 268 5$:
55 0000'DF4B D0 0110 269 MOVL @W^REMSGL_UCBVEC[R11],R5 ; Get the UCB address
64 A5 10 8A 0116 270 BICB #UCB$M_ONLINE,UCB$W_STS(R5) ; Waiting for CONFIGURE
FEE3' 30 011A 271 BSBW REM$SEND_CONFIG ; Send one, too
SA 02 90 011D 272 MOVB #REM$C_LNK_READ,R10 ; Set up for a link read
FEDD' 30 0120 273 BSBW REM$ENQUEUE ; Queue the work element
04 0123 274 RET ; Dismiss the AST
0124 275 :
0124 276 :
0124 277 :
0124 278 :++
0124 279 :
0124 280 : CONN_REQ - Connection request received - figure out who and why.
0124 281 : Allocate a channel and a UCB, set up the data bases,
0124 282 : and confirm the link, if everything is copasetic,
0124 283 : else reject the logical link.
0124 284 :
0124 285 : INPUTS:
0124 286 :
0124 287 : R9 - points at counted NCB
0124 288 :
0124 289 :--
0124 290 :
5A 0005'CF 9E 0124 291 CONN_REQ:
8A 89 9A 0129 292 MOVAB W^NCBDESC,R10 ; Point to descriptor
6A 59 D0 012C 293 MOVZBL (R9)+,(R10)+ ; Store length
5A 04 C2 012F 294 MOVL R9,(R10) ; Pointer to data
59 00' D0 0132 295 SUBL2 #4,R10 ; Point to descriptor
58 C5 AF 9E 0135 296 MOVL S^#IOS$_ACCESS,R9 ; Set the initial function code
0139 297 MOVAB B^CONFIRM_AST,R8 ; For confirm, take an AST
0139 298 :
0139 299 : Try to set up a database for this channel
0139 300 :
FEC4' 30 0139 301 BSBW REM$SET_UP_IN ; Do it
13 50 E8 013C 302 BLBS R0,200$ ; If LBS OK
013F 303 :
013F 304 : Can't take this link - reject it
013F 305 :
013F 306 100$:

```


REXPORT
Symbol table

- PROVIDE THE ACP TRANSPORT MECHANISM

H 12

16-SEP-1984 02:13:14 VAX/VMS Macro V04-00
5-SEP-1984 02:54:18 [REM.SRC]REXPORTD.MAR;1

Page 11
(11)

\$\$T1	=	00000000		
AQBSB_MNTCNT	=	0000000B		
BIN_TO_ASC		000001D6	R	04
BUGS_ACPMBFAIL		*****	X	04
CONFIRM_AST		000000FD	R	04
CONN_REQ		00000124	R	04
DIBSK_LENGTH	=	00000074		
DIBSW_UNIT	=	0000000C		
DIBBUF		000000A7	R	03
DIBDESC		00000027	R	02
IOSM_ABORT		*****	X	04
IOS_ACCESS		*****	X	04
IOS_ACPCONTROL		*****	X	04
IOS_READVBLK		*****	X	04
LINK_BROKEN		000001CC	R	04
MBXDESC		00000000	R	02
MBX_AST		0000009E	R	04
MBX_FAILURE		000000F9	R	04
MBX_MSG_LTH	=	00000096		
MBX_Q_IOSB		00000005	R	03
MBX_T_BUFFER		0000000D	R	03
MBX_T_DATA		00000011	R	03
MSG\$_ABORT	=	00000030		
MSG\$_CONNECT	=	00000032		
MSG\$_DISCON	=	00000033		
MSG\$_EXIT	=	00000034		
MSG\$_NETSHUT	=	0000003B		
MSG\$_PATHLOST	=	00000036		
MSG\$_PROTOCOL	=	00000037		
MSG\$_THIRDPARTY	=	00000039		
NCBDESC		00000005	R	03
NETDESC		00000012	R	02
NET_SHUTDOWN		000001CD	R	04
NFB		00000000	R	03
NFB\$C_DECLOBJ	=	00000016		
NFBDESC		0000001F	R	02
NFB_OBJ		00000001	R	03
REMSC_CURECO	=	00000001		
REMSC_CURVRS	=	00000001		
REMSC_LNK_READ	=	00000002		
REMSC_MAXDEVS	=	0000000A		
REMSC_MAXLINKS	=	00000010		
REMSC_MAXUNITS	=	00000010		
REMSC_MBX_READ	=	00000001		
REMSC_ST_ATTRIB	=	00000002		
REMSC_ST_CONFIG	=	00000001		
REMSDQ_QIO		*****	X	04
REMSSEND_QUEUE		*****	X	04
REMSG\$B_MAXLINKS		*****	X	04
REMSG\$GL_CHANVEC		*****	X	04
REMSG\$GL_MBX_UNITS		*****	X	04
REMSG\$GL_Q_HEAD		*****	X	04
REMSG\$GL_REJ_CHAN		*****	X	04
REMSG\$GL_RIOSBVEC		*****	X	04
REMSG\$GL_UCBVEC		*****	X	04
REMSGW_MBX_CHAN		*****	X	04
REMSKICL_UCB		*****	X	04

REMSMBX_READ		00000072	RG	04
REMSSEND_CONFIG		*****	X	04
REMSSET_OP_IN		*****	X	04
REMSXPORT_CHAN		00000181	RG	04
REMSXPORT_START		00000000	RG	04
RT_OBJ	=	00000017		
RT_OBJ2	=	0000002A		
SS\$_ABORT		*****	X	04
SS\$_CANCEL		*****	X	04
SYSS\$ASSIGN		*****	GX	04
SYSS\$CREMBX		*****	GX	04
SYSS\$GETCHN		*****	GX	04
SYSS\$QIO		*****	GX	04
SYSS\$QIOW		*****	GX	04
UCBSB_RTT_OBJ	=	000000D8		
UCBSK_RTT_LEN	=	00000138		
UCBSK_RTT_LENGTH	=	00000138		
UCBSL_DEVDEPND2	=	00000048		
UCBSL_RTT_CTRLC	=	00000094		
UCBSL_RTT_CTRLY	=	00000090		
UCBSL_RTT_DEVDEPEND2	=	00000048		
UCBSL_TL_BANDQUE	=	0000009C		
UCBSL_TL_CTRLC	=	00000094		
UCBSL_TL_CTRLY	=	00000090		
UCBSL_TL_OUTBAND	=	00000098		
UCBSM_ONLINE	=	00000010		
UCBSW_RTT_LINK	=	000000D6		
UCBSW_STS	=	00000064		

EX
Mo
--
RM
RM
IN
SY
CJ
MA
SY
RM

! Psect synopsis !

PSECT name	Allocation	PSECT No.	Attributes
. ABS .	00000000 (0.)	00 (0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
\$AB\$\$	00000000 (0.)	01 (1.)	NOPIC JSR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE
REM_XP_PURE	0000002F (47.)	02 (2.)	NOPIC USR CON REL LCL NOSHR NOEXE RD NOWRT NOVEC BYTE
REM_XP_IMPURE	0000011B (283.)	03 (3.)	NOPIC USR CON REL LCL NOSHR NOEXE RD WRT NOVEC BYTE
REM_XP_CODE	000001EA (490.)	04 (4.)	NOPIC USR CON REL LCL NOSHR EXE RD NOWRT NOVEC BYTE

! Performance indicators !

Phase	Page faults	CPU Time	Elapsed Time
Initialization	30	00:00:00.08	00:00:00.33
Command processing	107	00:00:00.63	00:00:02.35
Pass 1	422	00:00:16.31	00:00:34.77
Symbol table sort	0	00:00:02.40	00:00:04.37
Pass 2	92	00:00:02.68	00:00:05.19
Symbol table output	11	00:00:00.12	00:00:00.58
Psect synopsis output	2	00:00:00.03	00:00:00.03
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	666	00:00:22.25	00:00:47.62

The working set limit was 1500 pages.
82688 bytes (162 pages) of virtual memory were used to buffer the intermediate code.
There were 90 pages of symbol table space allocated to hold 1601 non-local and 24 local symbols.
443 source lines were read in Pass 1, producing 20 object records in Pass 2.
47 pages of virtual memory were used to define 40 macros.

! Macro library statistics !

Macro library name	Macros defined
-\$255\$DUA28:[REM.OBJ]REM.MLB;1	3
-\$255\$DUA28:[SYS.OBJ]LIB.MLB;1	14
-\$255\$DUA28:[SYSLIB]STARLET.MLB;2	14
TOTALS (all libraries)	31

1839 GETS were required to define 31 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS\$:REXPORTD/OBJ=OBJ\$:REXPORTD MSRCS\$:REXPORTD/UPDATE=(ENH\$:REXPORTD)+EXECMLS/LIB+LIB\$:REM/LIB

