

RRRRRRRRRRRR	TTTTTTTTTTTTTT	PPPPPPPPPPPP	AAAAAAA	DDDDDDDDDDDD
RRRRRRRRRRRR	TTTTTTTTTTTTTT	PPPPPPPPPPPP	AAAAAAA	DDDDDDDDDDDD
RRRRRRRRRRRR	TTTTTTTTTTTTTT	PPPPPPPPPPPP	AAAAAAA	DDDDDDDDDDDD
RRR	TTT	PPP	AAA	DDD
RRR	TTT	PPP	AAA	DDD
RRR	TTT	PPP	AAA	DDD
RRR	TTT	PPP	AAA	DDD
RRR	TTT	PPP	AAA	DDD
RRR	TTT	PPP	AAA	DDD
RRR	TTT	PPP	AAA	DDD
RRRRRRRRRRRR	TTTTTTTTTTTTTT	PPPPPPPPPPPP	AAA	DDD
RRRRRRRRRRRR	TTTTTTTTTTTTTT	PPPPPPPPPPPP	AAA	DDD
RRRRRRRRRRRR	TTTTTTTTTTTTTT	PPPPPPPPPPPP	AAA	DDD
RRR	TTT	PPP	AAAAAAA	DDD
RRR	TTT	PPP	AAAAAAA	DDD
RRR	TTT	PPP	AAAAAAA	DDD
RRR	TTT	PPP	AAA	DDD
RRR	TTT	PPP	AAA	DDD
RRR	TTT	PPP	AAA	DDD
RRR	TTT	PPP	AAA	DDD
RRR	TTT	PPP	AAA	DDD
RRR	TTT	PPP	AAA	DDDDDDDDDDDD
RRR	TTT	PPP	AAA	DDDDDDDDDDDD
RRR	TTT	PPP	AAA	DDDDDDDDDDDD

```

LL          IIIII!
LL          I!III!
LL          II
LL          II
LL          II
LL          II
LL          II
LL          II
LL          II
LL          II
LL          II
LL          II
LL          II
LL          II
LLLLLLLLLLL IIIII!
LLLLLLLLLLL I!III!
SSSSSSSSS
SSSSSSSSS
SS
SS
SS
SS
SSSSSS
SSSSSS
SS
SS
SS
SS
SSSSSSSSS
SSSSSSSSS

```

RTPAD
Table of contents

- REMOTE TERMINAL PROGRAM

H 14

16-SEP-1984 02:15:27 VAX/VMS Macro V04-00

Page 0

R1
VC

(1)	113	DECLARATIONS
(1)	146	RTPAD - MAIN ROUTINE
(1)	247	INIT - INITIALIZATION OF LINK, ETC.
(2)	611	RECORD QUIT - snapshot QUIT info
(3)	629	READ ONLY DATA
(3)	662	READ WRITE DATA
(3)	856	PROTOCOL TABLE PSECTS

```
0000 1 .TITLE RTPAD - REMOTE TERMINAL PROGRAM
0000 2 .IDENT 'V04-000'
00000000 3 .PSECT RTPAD,NOWRT
0000 4
0000 5 $DEBUGDEF
0000 6
0000 7
0000 8 *****
0000 9 *
0000 10 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0000 11 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0000 12 * ALL RIGHTS RESERVED.
0000 13 *
0000 14 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0000 15 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0000 16 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0000 17 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0000 18 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0000 19 * TRANSFERRED.
0000 20 *
0000 21 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0000 22 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0000 23 * CORPORATION.
0000 24 *
0000 25 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0000 26 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0000 27 *
0000 28 *
0000 29 *****
0000 30
0000 31 ++
0000 32 FACILITY: REMOTE TERMINAL SUPPORT
0000 33
0000 34 ABSTRACT:
0000 35
0000 36 THIS PROGRAM RUNS ON A LOCAL NODE TO ALLOW A TERMINAL TO APPEAR TO
0000 37 BE LOCALLY CONNECTED TO A REMOTE NODE.
0000 38
0000 39
0000 40 ENVIRONMENT: VMS - USER MODE
0000 41
0000 42
0000 43 AUTHOR: W M CARDOZA, CREATION DATE: 20-AUG-79
0000 44
0000 45 MODIFIED BY:
0000 46
0000 47 V03-017 JLV0362 Jake VanNoy 11-JUL-1984
0000 48 Add code to signal PC that QUOTA EXCEEDED occurred at.
0000 49
0000 50 V03-016 JLV0353 Jake VanNoy 10-APR-1984
0000 51 Add support for SET HOST/DTE ttcn:
0000 52
0000 53 V03-015 JLV0324 Jake VanNoy 10-JAN-1984
0000 54 Fix bug in setting of vax to vax flag.
0000 55
0000 56 V03-014 JLV0315 Jake VanNoy 7-DEC-1983
0000 57 Add logical name RTPAD$LOG for debug purposes.
```

```
0000 58 : Used Like FALSLOG, etc.
0000 59 :
0000 60 : V03-013 JLV0295 Jake VanNoy 28-JUL-1983
0000 61 : Add looping on non-zero WRITEQIO before exiting.
0000 62 : Add SET HOST/LOG[=filespec].
0000 63 :
0000 64 : V03-012 MHB0093 Mark Bramhall 7-Mar-1983
0000 65 : Moved MAXMSG to $RTPADDEF.
0000 66 : Reworked code to use dynamic descriptors.
0000 67 : Changed the CTERM detection algorithm.
0000 68 : Changed handling of PSTHRU messages.
0000 69 :
0000 70 : V03-011 MHB0089 Mark Bramhall 11-Feb-1983
0000 71 : Changed LIB$NET_CONNECT to UNS$NET_CONNECT.
0000 72 :
0000 73 : V03-010 JLV0 Jake VanNoy 17-Jan-1983
0000 74 : Added hooks for CTERM protocol. Broke up into two modules -
0000 75 : RTPAD and VMSRT. VMSRT now holds all of the VMS specific
0000 76 : protocol. General cleanup of existing code and comments.
0000 77 : Add use of new CLI interface, results in loss of first
0000 78 : command feature used by RSTSRT.
0000 79 :
0000 80 : V03-009 WMC0068 Wayne Cardoza 5-Oct-1982
0000 81 : Fix the previous DCL fix.
0000 82 :
0000 83 : V03-008 WMC0067 Wayne Cardoza 15-Oct-1982
0000 84 : Fix timing problem when link error arrives before the
0000 85 : mailbox message.
0000 86 : Work with new DCL keyword support.
0000 87 :
0000 88 : V03-007 JLV0214 Jake VanNoy 6-OCT-1982
0000 89 : Add Read Verify code as implemented (more or less)
0000 90 : by Steve Long.
0000 91 :
0000 92 : V03-006 WMC0066 Wayne Cardoza 20-Aug-1982
0000 93 : Take care of errors on SENSE MODE.
0000 94 :
0000 95 : V03-005 WMC0065 Wayne Cardoza 7-Jul-1982
0000 96 : Don't issue out of band set mode QIO if no change.
0000 97 :
0000 98 : V03-004 WMC0064 Wayne Cardoza 9-Apr-1982
0000 99 : Don't try to disaule ^T under MCR.
0000 100 :
0000 101 : V03-003 WMC0063 Wayne Cardoza 1-Apr-1982
0000 102 : Zero the buffer before a SENSE.
0000 103 :
0000 104 : V03-002 WMC0062 Wayne Cardoza 18-Mar-1982
0000 105 : Add TERMCHAR in DIB format back for RSX, etc.
0000 106 :
0000 107 : V03-001 WMC0061 Wayne Cardoza 15-Mar-1982
0000 108 : Send extended characteristics in configuration message.
0000 109 :
0000 110 : **
0000 111 : --
```

```
0000 113 .SBTTL DECLARATIONS
0000 114 :
0000 115 : DEFAULT ADDRESSING MODE
0000 116 :
0000 117 .DEFAULT DISPLACEMENT WORD
0000 118 :
0000 119 : INCLUDE FILES:
0000 120 :
0000 121 $DIBDEF ;DIB OFFSETS
0000 122 $DSCDEF ;DESCRIPTOR DEFINITIONS
0000 123 $SDVIDEF ;GETDVI DEFINITIONS
0000 124 $IODEF ;I/O OP CODES & MODIFIERS
0000 125 $RDPDEF ;REMOTE DEVICE PROTOCOL
0000 126 $RTPADDEF ; *** NEW
0000 127 $TSADEF ; *** NEW *** tsadef
0000 128 $TTYDEFS ;TERMINAL DRIVER SYMBOLS
0000 129 :
0000 130 : EQUATED SYMBOLS:
0000 131 :
0000 132 : AST CONTROL BLOCK
0000 133 :
00000026 0000 134 AST$_BUF = CTP$_PRO_MSGTYPE ; DATA BUFFER STARTS HERE
0000001FE 0000 135
0000 136 REM$_FACILITY = 510 ; REM$_code
0000 137
0000 138 .show me,MC,MD
0000 139 $SHR_MSGDEF NAME=REM, CODE=REM$_FACILITY, SCOPE=LOCAL, -
0000 140 MSGCODES = <<ATPC,INFO>
00000001 0000 .IF NDF,SHR$_SHRDEF
0000 SHR$_SHRDEF = 1
0000 $SHRDEF
0000 $DEFINI SHR,
0000 .SAVE LOCAL_BLOCK
0000 .NOCROSS
0000 .IIF DIF <> <GLOBAL>,.ENABLE SUPPRESSION
0000 .PSECT $ABS$,ABS
0000 $GBLINI
0000 .IF IDN <LOCAL> <GLOBAL>
0000 .MACRO $DEF SYM,ALLOC,SIZ
0000 .IIF NB,SYM,SYM::
0000 .IIF NB,ALLOC, ALLOC SIZ
0000 .ENDM $DEF
0000 .MACRO $SEQU SYM,VAL
0000 SYM=VAL
0000 .ENDM $SEQU
0000 .MACRO $VIELD1 MOD,SEP,SYM,SIZ,MSK
0000 SIZ...=1
0000 .IIF NB,SIZ, SIZ...=SIZ
0000 .IF NB,SYM
0000 MOD'SEP'V 'SYM==BIT::
0000 .IIF NB,SIZ, MOD'SEP'S 'SYM==SIZ
0000 .IIF NB,MSK, MOD'SEP'M 'SYM==<<<1@SIZ...>-1>@BIT...>
0000 .ENDC
0000 BIT...=BIT...+SIZ...
0000 .ENDM $VIELD1
0000 .IIF
0000 .IIF DIF <LOCAL> <LOCAL>,.ERROR ;ARG MUST BE 'GLOBAL','LOCAL',OR NULL
```



```
0000 .MACRO SDEF SYM,ALLOC,SIZ
0000 .IIF NB,SYM,SYM:
0000 .IIF NB,ALLOC,ALLOC SIZ
0000 .ENDM SDEF
0000 .MACRO SEQU SYM,VAL
0000 SYM=VAL
0000 .ENDM SEQU
0000 .MACRO $VIELD1 MOD,SEP,SYM,SIZ,MSK
0000 SIZ...=1
0000 .IIF NB,SIZ,SIZ...=SIZ
0000 .IF NB,SYM
0000 MOD'SEP'V'SYM=BIT...
0000 .IIF NB,SIZ,MOD'SEP'S'SYM=SIZ
0000 .IIF NB,MSK,MOD'SEP'M'SYM=<<<1@SIZ...>-1>@BIT...>
0000 .ENDC
0000 BIT...=BIT...+SIZ...
0000 .ENDM $VIELD1
0000 .ENDC
00000000 0000 .=0
00000000 0000 SEQU SHRS_FACILITY 0
00000000 0000 SHRS_FACILITY=0
00001000 0000 SEQU SHRS_APPENDED 4096
00001000 0000 SHRS_APPENDED=4096
00001008 0000 SEQU SHRS_APPENDED 4104
00001008 0000 SHRS_APPENDED=4104
00001010 0000 SEQU SHRS_BADBYTE 4112
00001010 0000 SHRS_BADBYTE=4112
00001018 0000 SEQU SHRS_BADFIELD 4120
00001018 0000 SHRS_BADFIELD=4120
00001020 0000 SEQU SHRS_BADLONG 4128
00001020 0000 SHRS_BADLONG=4128
00001028 0000 SEQU SHRS_BADWORD 4136
00001028 0000 SHRS_BADWORD=4136
00001030 0000 SEQU SHRS_BEGIN 4144
00001030 0000 SHRS_BEGIN=4144
00001038 0000 SEQU SHRS_BEGIN 4152
00001038 0000 SHRS_BEGIN=4152
00001040 0000 SEQU SHRS_BEGIN 4160
00001040 0000 SHRS_BEGIN=4160
00001048 0000 SEQU SHRS_CLICB 4168
00001048 0000 SHRS_CLICB=4168
00001050 0000 SEQU SHRS_CLOSEIN 4176
00001050 0000 SHRS_CLOSEIN=4176
0000
```

RTPAD
V04-000

- REMOTE TERMINAL PROGRAM
DECLARATIONS

M 14

16-SEP-1984 02:15:27 VAX/VMS Macro V04-00
5-SEP-1984 03:15:47 [RTPAD.SRC]RTPAD.MAR;1

Page 5
(1)

R
V

00001058	0000	\$EQU	SHR\$_CLOSEOUT 4184
	0000		SHR\$_CLOSEOUT=4184
	0000		
00001060	0000	\$EQU	SHR\$_COPIEDB 4192
	0000		SHR\$_COPIEDB=4192
	0000		
00001068	0000	\$EQU	SHR\$_COPIEDR 4200
	0000		SHR\$_COPIEDR=4200
	0000		
00001070	0000	\$EQU	SHR\$_CREATED 4208
	0000		SHR\$_CREATED=4208
	0000		
00001078	0000	\$EQU	SHR\$_ENDED 4216
	0000		SHR\$_ENDED=4216
	0000		
00001080	0000	\$EQU	SHR\$_ENEDD 4224
	0000		SHR\$_ENEDD=4224
	0000		
00001088	0000	\$EQU	SHR\$_ENEDT 4232
	0000		SHR\$_ENEDT=4232
	0000		
00001090	0000	\$EQU	SHR\$_NEWFILES 4240
	0000		SHR\$_NEWFILES=4240
	0000		
00001098	0000	\$EQU	SHR\$_OPENIN 4248
	0000		SHR\$_OPENIN=4248
	0000		
000010A0	0000	\$EQU	SHR\$_OPENOUT 4256
	0000		SHR\$_OPENOUT=4256
	0000		
000010A8	0000	\$EQU	SHR\$_OVERLAY 4264
	0000		SHR\$_OVERLAY=4264
	0000		
000010B0	0000	\$EQU	SHR\$_READERR 4272
	0000		SHR\$_READERR=4272
	0000		
000010B8	0000	\$EQU	SHR\$_REPLACED 4280
	0000		SHR\$_REPLACED=4280
	0000		
000010C0	0000	\$EQU	SHR\$_WILDCONCAT 4288
	0000		SHR\$_WILDCONCAT=4288
	0000		
000010C8	0000	\$EQU	SHR\$_WILDOUTVER 4296
	0000		SHR\$_WILDOUTVER=4296
	0000		
000010D0	0000	\$EQU	SHR\$_WRITEERR 4304
	0000		SHR\$_WRITEERR=4304
	0000		
000010D8	0000	\$EQU	SHR\$_ABEND 4312
	0000		SHR\$_ABEND=4312
	0000		
000010E0	0000	\$EQU	SHR\$_ABENDD 4320
	0000		SHR\$_ABENDD=4320
	0000		
000010E8	0000	\$EQU	SHR\$_ABENDT 4328
	0000		SHR\$_ABENDT=4328
	0000		

000010F0	0000	SEQU	SHRS_SYSEERRORPC 4336
	0000		SHRS_SYSEERRORPC=4336
	0000		
000010F8	0000	SEQU	SHRS_SYNTAX 4344
	0000		SHRS_SYNTAX=4344
	0000		
00001100	0000	SEQU	SHRS_NOVALUE 4352
	0000		SHRS_NOVALUE=4352
	0000		
00001108	0000	SEQU	SHRS_BADKEY 4360
	0000		SHRS_BADKEY=4360
	0000		
00001110	0000	SEQU	SHRS_BADVALUE 4368
	0000		SHRS_BADVALUE=4368
	0000		
00001118	0000	SEQU	SHRS_BADDELIM 4376
	0000		SHRS_BADDELIM=4376
	0000		
00001120	0000	SEQU	SHRS_BADLOGIC 4384
	0000		SHRS_BADLOGIC=4384
	0000		
00001128	0000	SEQU	SHRS_NOWILD 4392
	0000		SHRS_NOWILD=4392
	0000		
00001130	0000	SEQU	SHRS_TEXT 4400
	0000		SHRS_TEXT=4400
	0000		
00001138	0000	SEQU	SHRS_IDXCONCAT 4408
	0000		SHRS_IDXCONCAT=4408
	0000		
00001140	0000	SEQU	SHRS_RELCONCAT 4416
	0000		SHRS_RELCONCAT=4416
	0000		
00001148	0000	SEQU	SHRS_HIGHVER 4424
	0000		SHRS_HIGHVER=4424
	0000		
00001150	0000	SEQU	SHRS_BADLOGICPC 4432
	0000		SHRS_BADLOGICPC=4432
	0000		
00001158	0000	SEQU	SHRS_ATPC 4440
	0000		SHRS_ATPC=4440
	0000		
00001160	0000	SEQU	SHRS_BADCOPIES 4448
	0000		SHRS_BADCOPIES=4448
	0000		
00001168	0000	SEQU	SHRS_BADFORM 4456
	0000		SHRS_BADFORM=4456
	0000		
00001170	0000	SEQU	SHRS_BADJOBID 4464
	0000		SHRS_BADJOBID=4464
	0000		
00001178	0000	SEQU	SHRS_BADJOBNAME 4472
	0000		SHRS_BADJOBNAME=4472
	0000		
00001180	0000	SEQU	SHRS_BADPRTY 4480
	0000		SHRS_BADPRTY=4480
	0000		

00001188	0000	\$EQU	SHR\$_BADQNAME 4488
	0000		SHR\$_BADQNAME=4488
	0000		
00001190	0000	\$EQU	SHR\$_BADTIME 4496
	0000		SHR\$_BADTIME=4496
	0000		
00001198	0000	\$EQU	SHR\$_NOQUEUE 4504
	0000		SHR\$_NOQUEUE=4504
	0000		
000011A0	0000	\$EQU	SHR\$_NOJOBID 4512
	0000		SHR\$_NOJOBID=4512
	0000		
000011A8	0000	\$EQU	SHR\$_NOJOBNAME 4520
	0000		SHR\$_NOJOBNAME=4520
	0000		
000011B0	0000	\$EQU	SHR\$_SYSERROR 4528
	0000		SHR\$_SYSERROR=4528
	0000		
000011B8	0000	\$EQU	SHR\$_NOTCOPIED 4536
	0000		SHR\$_NOTCOPIED=4536
	0000		
000011C0	0000	\$EQU	SHR\$_NOTCMPLT 4544
	0000		SHR\$_NOTCMPLT=4544
	0000		
000011C8	0000	\$EQU	SHR\$_RMSERROR 4552
	0000		SHR\$_RMSERROR=4552
	0000		
000011D0	0000	\$EQU	SHR\$_UNXPCTSTS 4560
	0000		SHR\$_UNXPCTSTS=4560
	0000		
000011D8	0000	\$EQU	SHR\$_HASHCONCAT 4568
	0000		SHR\$_HASHCONCAT=4568
	0000		
000011E0	0000	\$EQU	SHR\$_INCOMPAT 4576
	0000		SHR\$_INCOMPAT=4576
	0000		
000011E8	0000	\$EQU	SHR\$_VALERR 4584
	0000		SHR\$_VALERR=4584
	0000		
000011F0	0000	\$EQU	SHR\$_FILNOTDEL 4592
	0000		SHR\$_FILNOTDEL=4592
	0000		
000011F8	0000	\$EQU	SHR\$_CONFDEL 4600
	0000		SHR\$_CONFDEL=4600
	0000		
00001200	0000	\$EQU	SHR\$_DELETED 4608
	0000		SHR\$_DELETED=4608
	0000		
00001208	0000	\$EQU	SHR\$_DELVER 4616
	0000		SHR\$_DELVER=4616
	0000		
00001210	0000	\$EQU	SHR\$_PURGEVER 4624
	0000		SHR\$_PURGEVER=4624
	0000		
00001218	0000	\$EQU	SHR\$_CLOSEDEL 4632
	0000		SHR\$_CLOSEDEL=4632
	0000		

00001220	0000	SEQU	SHRS_DIRTOOBUS 4640
	0000		SHRS_DIRTOOBUS=4640
	0000		
00001228	0000	SEQU	SHRS_NOFILPURG 4648
	0000		SHRS_NOFILPURG=4648
	0000		
00001230	0000	SEQU	SHRS_FILNOTPUR 4656
	0000		SHRS_FILNOTPUR=4656
	0000		
00001238	0000	SEQU	SHRS_SEARCHFAIL 4664
	0000		SHRS_SEARCHFAIL=4664
	0000		
00001240	0000	SEQU	SHRS_DELINTERR 4672
	0000		SHRS_DELINTERR=4672
	0000		
00001248	0000	SEQU	SHRS_PARSEFAIL 4680
	0000		SHRS_PARSEFAIL=4680
	0000		
00001250	0000	SEQU	SHRS_FILPURGED 4688
	0000		SHRS_FILPURGED=4688
	0000		
00001258	0000	SEQU	SHRS_ENDABORT 4696
	0000		SHRS_ENDABORT=4696
	0000		
00001260	0000	SEQU	SHRS_ENDDIAGS 4704
	0000		SHRS_ENDDIAGS=4704
	0000		
00001268	0000	SEQU	SHRS_ENDNOOBJ 4712
	0000		SHRS_ENDNOOBJ=4712
	0000		
00001270	0000	SEQU	SHRS_HALTED 4720
	0000		SHRS_HALTED=4720
	0000		
00001278	0000	SEQU	SHRS_NOCMDMEM 4728
	0000		SHRS_NOCMDMEM=4728
	0000		
00001280	0000	SEQU	SHRS_QEMPTY 4736
	0000		SHRS_QEMPTY=4736
	0000		
00001288	0000	SEQU	SHRS_CBT 4744
	0000		SHRS_CBT=4744
	0000		
00001290	0000	SEQU	SHRS_EXISTS 4752
	0000		SHRS_EXISTS=4752
	0000		
00001298	0000	SEQU	SHRS_UNLOCKED 4760
	0000		SHRS_UNLOCKED=4760
	0000		
000012A0	0000	SEQU	SHRS_RENAMED 4768
	0000		SHRS_RENAMED=4768
	0000		
000012A8	0000	SEQU	SHRS_PROTECTED 4776
	0000		SHRS_PROTECTED=4776
	0000		
000012B0	0000	SEQU	SHRS_NOTLOCKED 4784
	0000		SHRS_NOTLOCKED=4784
	0000		

000012B8	0000	SEQU	SHR\$_ACTIMAGE 4792
	0000		SHR\$_ACTIMAGE=4792
	0000		
000012C0	0000	SEQU	SHR\$_DIRNOTCRE 4800
	0000		SHR\$_DIRNOTCRE=4800
	0000		
000012C8	0000	SEQU	SHR\$_NODESTQUE 4808
	0000		SHR\$_NODESTQUE=4808
	0000		
000012D0	0000	SEQU	SHR\$_ILLDESQUE 4816
	0000		SHR\$_ILLDESQUE=4816
	0000		
000012D8	0000	SEQU	SHR\$_NOTTERM 4824
	0000		SHR\$_NOTTERM=4824
	0000		
000012E0	0000	SEQU	SHR\$_CONFQUAL 4832
	0000		SHR\$_CONFQUAL=4832
	0000		
000012E8	0000	SEQU	SHR\$_ILLDIRCOPY 4840
	0000		SHR\$_ILLDIRCOPY=4840
	0000		
000012F0	0000	SEQU	SHR\$_INSVIRMEM 4848
	0000		SHR\$_INSVIRMEM=4848
	0000		
000012F8	0000	SEQU	SHR\$_CREATEDSTM 4856
	0000		SHR\$_CREATEDSTM=4856
	0000		
00001300	0000	SEQU	SHR\$_NOTRUNC 4864
	0000		SHR\$_NOTRUNC=4864
	0000		
00001308	0000	SEQU	SHR\$_PRODNOTINS 4872
	0000		SHR\$_PRODNOTINS=4872
	0000		
00001310	0000	SEQU	SHR\$_TOTAL 4880
	0000		SHR\$_TOTAL=4880
	0000		
00001318	0000	SEQU	SHR\$_FILPURG 4888
	0000		SHR\$_FILPURG=4888
	0000		
00001320	0000	SEQU	SHR\$_FILDEL 4896
	0000		SHR\$_FILDEL=4896
	0000		
00001328	0000	SEQU	SHR\$_INVQUAVAL 4904
	0000		SHR\$_INVQUAVAL=4904
	0000		
00001330	0000	SEQU	SHR\$_NOFILES 4912
	0000		SHR\$_NOFILES=4912
	0000		
00001338	0000	SEQU	SHR\$_FILNOTACC 4920
	0000		SHR\$_FILNOTACC=4920
	0000		
00001340	0000	SEQU	SHR\$_QUALMISS 4928
	0000		SHR\$_QUALMISS=4928
	0000		
00001348	0000	SEQU	SHR\$_FILSPCSRCH 4936
	0000		SHR\$_FILSPCSRCH=4936
	0000		

```
00001350 0000 $EQU SHRS_NGSRCHLST 4944
00001350 0000 SHRS_NOSRCHLST=4944
00001358 0000 $EQU SHRS_NOSUCHID 4952
00001358 0000 SHRS_NOSUCHID=4952
00001358 0000 $DEFEND SHR,DEF
00001358 0000 .MACRO $SHRDEF A
00001358 0000 .ENDM $SHRDEF
00001358 0000 .IIF DIF <> <GLOBAL>,.DISABLE SUPPRESSION
00001358 0000 .CROSS
00001358 0000 .RESTORE
00001358 0000 .ENDC
00000000 0000 $$GBL = 0
00000000 0000 .IIF IDN,LOCAL,GLOBAL,$$GBL = 1
00000000 0000 .IRP MSGPAIR,<<ATPC,INFO>>
00000000 0000 $SHR,MSGCOD REM, REMS_FACILITY, MSGPAIR
00000000 0000 .ENDR
00000000 0000 $SHR,MSGCOD REM, REMS_FACILITY, ATPC,INFO
00000000 0000 .IF IDN,INFO,WARNING
00000000 0000 .IF EQ $$GBL
00000000 0000 REMS_ATPC = 0
00000000 0000 .IFF
00000000 0000 REMS_ATPC == 0
00000000 0000 .ENDC
00000000 0000 .IFF
00000000 0000 .IF IDN,INFO,SUCCESS
00000000 0000 .IF EQ $$GBL
00000000 0000 REMS_ATPC = 1
00000000 0000 .IFF
00000000 0000 REMS_ATPC == 1
00000000 0000 .ENDC
00000000 0000 .IFF
00000000 0000 .IF IDN,INFO,ERROR
00000000 0000 .IF EQ $$GBL
00000000 0000 REMS_ATPC = 2
00000000 0000 .IFF
00000000 0000 REMS_ATPC == 2
00000000 0000 .ENDC
00000000 0000 .IFF
00000000 0000 .IF IDN,INFO,INFO
00000000 0000 .IF EQ $$GBL
00000000 0000 REMS_ATPC = 3
00000000 0000 .IFF
00000000 0000 REMS_ATPC == 3
00000000 0000 .ENDC
00000000 0000 .IFF
00000000 0000 .IF IDN,INFO,SEVERE
00000000 0000 .IF EQ $$GBL
00000000 0000 REMS_ATPC = 4
00000000 0000 .IFF
00000000 0000 REMS_ATPC == 4
00000000 0000 .ENDC
00000000 0000 .IFF
00000000 0000 .IF EQ $$GBL
```

```

0000      REMS_ATPC = INFO
0000      .IFF
0000      REMS_ATPC == INFO
0000      .ENDC
0000      .ENDC
0000      .ENDC
0000      .ENDC
0000      .ENDC
0000      .ENDC
00000000 0000      .IF EQ $$GBL
01FE115B 0000      REMS_ATPC = REMS_ATPC+SHRS_ATPC+<REMS_FACILITY@16>
0000      .IFF
0000      REMS_ATPC == REMS_ATPC+SHRS_ATPC+<REMS_FACILITY@16>
0000      .ENDC
0000
0000      141 .noshow meb
0000      142
0000      143 ;
0000      144 ;

```



```
0000 146 .SBTTL RTPAD - MAIN ROUTINE
0000 147 :++
0000 148 : FUNCTIONAL DESCRIPTION:
0000 149 :
0000 150 :     MAIN ROUTINE
0000 151 :
0000 152 : CALLING SEQUENCE:
0000 153 :
0000 154 :     RUN FROM A TERMINAL
0000 155 :
0000 156 : INPUT PARAMETERS:
0000 157 :
0000 158 :     NONE
0000 159 :
0000 160 : IMPLICIT INPUTS:
0000 161 :
0000 162 :     NONE
0000 163 :
0000 164 : OUTPUT PARAMETERS:
0000 165 :
0000 166 :     NONE
0000 167 :
0000 168 : IMPLICIT OUTPUTS:
0000 169 :
0000 170 :     NONE
0000 171 :
0000 172 : COMPLETION CODES:
0000 173 :
0000 174 :     RETURNED IF A SYSTEM SERVICE HAS AN UNEXPECTED ERROR
0000 175 :
0000 176 : SIDE EFFECTS:
0000 177 :
0000 178 :     DECNET LINK SET UP WITH A REMOTE NODE
0000 179 :
0000 180 : --
0000 181 RTPAD:
0000 182 .WORD 0
0002 183
0002 184 .if df debug
0002 185 BSBW DEBUG_SETUP ; enable ^B
0002 186 .endc
0002 187
0002 188 ;
0002 189 ; INIT creates logical link and starts up protocol module,
0002 190 ; everything after that is AST driven.
0002 191 ;
0002 192
0000 30 0002 193 BSBW INIT ; INIT & set up logical link
0005 194
0005 195 10$: $HIBER_S ; Proceed asynchronous from now on
0005
0005 .GLOBL SYSSHIBER
0005 CALLS #0,G^SYSSHIBER
000C
000C
000C 196 TSTB WAKEFLAG ; Time to exit if > 0
000C 197 BEQL 10$ ; Spurious $WAKE
0010 198
0012 199 BLBC CTERM_FLAG,20$ ; branch if not cterm
11 02B2'CF E9 0012
```

```
0000'CF D5 0017 200 TSTL WRITEQIO ; must also be zero before exiting
OB 13 001B 201 BEQL 20$ ; yup, exit
001D 202 $SETAST_S ENBFLG = #1 ; Allow ast delivery (turned off by QUIT)
001D .GLOBL SYSS$SETAST
001D .PUSHL #1
001F .CALLS #1,G^SYSS$SETAST
0026 DD 11 0026 203 BRB 10$ ; loop
0028 204 20$:
0028 205 ;
0028 206 ; Wakeflag is set, exit back to DCL
0028 207 ;
0028 208
0028 209 $SETAST_S ENBFLG = #0 ; Shut down ast delivery
0028 .GLOBL SYSS$SETAST
0028 .PUSHL #0
002A .CALLS #1,G^SYSS$SETAST
0031
0031 210
0031 211 $CANCEL_S CHAN = READCHAN
0031 .GLOBL SYSS$CANCEL
0031 MOVZWL READCHAN,-(SP)
0036 .CALLS #1,G^SYSS$CANCEL
003D
003D 212 $CANCEL_S CHAN = TERMMBXCHAN
003D .GLOBL SYSS$CANCEL
003D MOVZWL TERMMBXCHAN,-(SP)
0042 .CALLS #1,G^SYSS$CANCEL
0049
0049 213 $CANCEL_S CHAN = MAILCHAN
0049 .GLOBL SYSS$CANCEL
0049 MOVZWL MAILCHAN,-(SP)
004E .CALLS #1,G^SYSS$CANCEL
0055
0055 214 $CANCEL_S CHAN = LINKCHAN
0055 .GLOBL SYSS$CANCEL
0055 MOVZWL LINKCHAN,-(SP)
005A .CALLS #1,G^SYSS$CANCEL
0061
0061 215
0061 216 $PUTMSG_S MSGVEC = EXITMSG ; Tell user why
0061 .GLOBL SYSS$PUTMSG
0061 .PUSHL #0
0063 $PUSHADR 0,CONTEXT=0
0063 .IF IDN,0,0
0063 .PUSHL #0
0065 .IFF
0065 $PUSHAQ 0
0065 .ENDC
0065
0065 $PUSHADR 0
0065 .IF IDN,0,0
0065 .PUSHL #0
0067 .IFF
0067 $PUSHAL 0
0067 .ENDC
0067
```



```
0067 $PUSHADR EXITMSG
0067 .IF IDN,0,EXITMSG
0067 PUSHL #0
0067 .IFF
0067 PUSHAL EXITMSG
0068 .ENDC
0068 CALLS #4,G^SYSS$PUTMSG
0072
0072 217
0072 218
0072 219
0072 $QIO_S CHAN = CNTRLCHAN - ; Get rid of the ^Y AST request
0072 FUNC = #IOS$SETMODE!IOSM_CTRLFAST
0072 .GLOBL SYS$QIO
0072 $PUSHTWO #0,#0
0072 $$T1 = 0
0072 .IF IDN,<#0>,<#0>
0072 .IF IDN,<#0>,<#0>
0072 $$T1 = 1
0072 .ENDC
0072 .ENDC
0072 .IF NE $$T1
0072 CLRQ -(SP)
0074 .IFF
0074 PUSHL #0
0074 PUSHL #0
0074 .ENDC
0074
0074 $PUSHTWO #0,#0
0074 $$T1 = 0
0074 .IF IDN,<#0>,<#0>
0074 .IF IDN,<#0>,<#0>
0074 $$T1 = 1
0074 .ENDC
0074 .ENDC
0074 .IF NE $$T1
0074 CLRQ -(SP)
0076 .IFF
0076 PUSHL #0
0076 PUSHL #0
0076 .ENDC
0076
0076 PUSHL #0
0078 $PUSHADR 0
0078 .IF IDN,0,0
0078 PUSHL #0
007A .IFF
007A PUSHAL 0
007A .ENDC
007A
007A $QIOPUSH #0,0
007A $$T1 = 0
007A .IF IDN,<#0>,<#0>
007A .IF IDN,<0>,<0>
007A $$T1 = 1
007A .ENDC
007A .ENDC
007A .IF NE $$T1
```

```

7E 7C 007A CLRQ -(SP)
      007C .IFF
      007C PUSHL #0
      007C $PUSHADR 0
      007C .ENDC
      007C $PUSHADR 0,CONTEXT=0
      007C .IF IDN,0,0
00 DD 007C PUSHL #0
      007E .IFF
      007E PUSHAQ 0
      007E .ENDC
      007E MOVZWL #IOS$ SETMODE!IOSM_CTRLVAST,-(SP)
7E 00A3 8F 3C 007E MOVZWL CNTRECHAN,-(SP)
7E 0154'CF 3C 0083
      00 DD 0088
00000000'GF 0C FB 008A
      0091
      0091 220
      0091 221
      0091 222
      0091 223
      0091 224
00000000'GF 031C'CF D4 0091 224
      02 DF 0093 225
      FB 0097 226
      009E 227
      009E 228
      009E 229
      009E
00000000'GF 0320'CF DD 009E
      01 FB 00A2
      00A9
      00A9 230
      FF54' 30 00A9 231
      00AC 232
50 50 0309'CF D0 00AC 233
50 00000000'8F D1 00B1 234
      1A 12 00B8 235
      00BA 236
      030D'CF DD 00BA 237
      01 DD 00BE 238
      01FE115B 8F DD 00C0 239
      50 DD 00C6 240
00000000'GF 04 FB 00C8 241
      00CF 242
      50 0309'CF D0 00CF 243
      04 00D4 244 100$:
      00D5 245

; Restore original out of band and resouce wait mode
;
CLRL -(SP)
PUSHAL OLDCTRL ; Reenable cli out of band characters
CALLS #2,G^LIB$ENABLE_CTRL

$SETRWM,S - ; Set resource wait mode
      WATFLG = OLDSETRWM ; to whatever it was upon entry
      .GLOBL SYSSSETRWM
      PUSHAL OLDSETRWM
      CALLS #1,G^SYSSSETRWM

BSBW CTERM$CLOSE_LOG ; Close log file if open

MOVL RETSTATUS,R0 ; Get saved status
CMPL #SS$_EXQUOTA,R0 ; Exceeded some quota?
BNEQ 100$ ; nope, exit

PUSHL QUIT_PC
PUSHL #1
PUSHL #REMS_ATPC
PUSHL R0
CALLS #4,G^LIB$SIGNAL ; signal error

MOVL RETSTATUS,R0 ; Get saved status
RET ; Exit program
```

```
00D5 247 .SBTTL INIT - INITIALIZATION OF LINK, ETC.
00D5 248 :++
00D5 249 : FUNCTIONAL DESCRIPTION:
00D5 250 :
00D5 251 : PERFORMS INITIALIZATION FUNCTIONS FOR RTPAD
00D5 252 :
00D5 253 : CALLING SEQUENCE:
00D5 254 :
00D5 255 : JSB INIT
00D5 256 :
00D5 257 : INPUT PARAMETERS:
00D5 258 :
00D5 259 : NONE
00D5 260 :
00D5 261 : IMPLICIT INPUTS:
00D5 262 :
00D5 263 : NONE
00D5 264 :
00D5 265 : OUTPUT PARAMETERS:
00D5 266 :
00D5 267 : NONE
00D5 268 :
00D5 269 : IMPLICIT OUTPUTS:
00D5 270 :
00D5 271 : CHANNEL NUMBERS, ETC.
00D5 272 :
00D5 273 : COMPLETION CODES:
00D5 274 :
00D5 275 : WILL RETURN COMPLETION CODES OF SYSTEM SERVICES WITH UNEXPECTED ERRORS
00D5 276 :
00D5 277 : SIDE EFFECTS:
00D5 278 :
00D5 279 : SETS UP DECNET LINK TO REMOTE NODE
00D5 280 :
00D5 281 :--
00D5 282 :
00D5 283 INIT:
00D5 284 $TRNLOG_S -
00D5 285 LOGNAM = SYS$NODE, - ; Translate a logical name
00D5 286 RSLBUF = NODENAME, - ; from SYS$NODE
00D5 287 RSLLEN = NODENAME ; to the ending message
; setting the correct length
00 DD 00D5 .GLOBL SYS$TRNLOG
00D5 PUSHL #0
00D7 $PUSHADR 0,CONTEXT=B
00 DD 00D7 .IF IDN,0,0
00D7 PUSHL #0
00D9 .IFF
00D9 PUSHAB 0
00D9 .ENDC
00D9
00D9 $PUSHADR 0,CONTEXT=B
00 DD 00D9 .IF IDN,0,0
00D9 PUSHL #0
00D8 .IFF
00D8 PUSHAB 0
00D8 .ENDC
00D8
```

```
00DB      $PUSHADR NODENAME,CONTEXT=Q
00DB      .IF      IDN,0,NODENAME
00DB      PUSHL    #0
00DB      .IFF
01B0'CF    7F 00DB      PUSHAQ  NODENAME
00DF      .ENDC
00DF      $PUSHADR NODENAME,CONTEXT=W
00DF      .IF      IDN,0,NODENAME
00DF      PUSHL    #0
00DF      .IFF
01B0'CF    3F 00DF      PUSHAQ  NODENAME
00E3      .ENDC
00E3      $PUSHADR SYSS$NODE,CONTEXT=Q
00E3      .IF      IDN,0,SYSS$NODE
00E3      PUSHL    #0
00E3      .IFF
04C6'CF    7F 00E3      PUSHAQ  SYSS$NODE
00E7      .ENDC
00E7      CALLS    #6,G^SYSS$TRNLOG
00EE      ONERROR RET          ; Exit on error
01 50      E8 00EE      BLBS    RO,30000$
04         04 00F1      RET
00F2      30000$:
00F2
00F2      289
00F2      290
00F2      291
00F2      292
00F2      293
00         DD 00F2      $TRNLOG_S - ; translate 'RTPAD$LOG'
00F2      LOGNAM = RTPAD_LOGNAM,-
00F4      RSLLEN = RTLOG_DESC, -
00F4      RSLBUF = RTLOG_DESC
00F4      .GLOBL  SYSS$TRNLOG
00F4      PUSHL    #0
00F4      $PUSHADR 0,CONTEXT=B
00F4      .IF      IDN,0,0
00F4      PUSHL    #0
00F6      .IFF
00F6      PUSHAB  0
00F6      .ENDC
00F6      $PUSHADR 0,CONTEXT=B
00F6      .IF      IDN,0,0
00F6      PUSHL    #0
00F6      .IFF
00         DD 00F6      PUSHAB  0
00F8      .ENDC
00F8      $PUSHADR RTLOG_DESC,CONTEXT=Q
00F8      .IF      IDN,0,RTLOG_DESC
00F8      PUSHL    #0
00F8      .IFF
02BA'CF    7F 00F8      PUSHAQ  RTLOG_DESC
00FC      .ENDC
00FC      $PUSHADR RTLOG_DESC,CONTEXT=W
00FC
```

			00FC			.IF	IDN,0,RTLOG_DESC	
			00FC			PUSHL	#0	
			00FC			.IFF		
02BA'CF	3F		00FC			PUSHAW	RTLOG_DESC	
			0100			.ENDC		
			0100					
			0100			\$PUSHADR	RTPAD_LOGNAM,CONTEXT=0	
			0100			.IF	IDN,0,RTPAD_LOGNAM	
			0100			PUSHL	#0	
			0100			.IFF		
02D2'CF	7F		0100			PUSHAQ	RTPAD_LOGNAM	
			0104			.ENDC		
			0104					
00000000'GF	06	FB	0104			CALLS	#6,G^SYS\$TRNLOG	
			010B					
0000'8F	50	B1	010B	294		CMPW	R0,#SS\$_NOTRAN	
	1E	13	0110	295		BEQL	SS	; continue if no definition
1B	50	E9	0112	296		BLBC	R0,SS	; or error
			0115	297				
			0115	298				; translate hex byte string to binary value
			0115	299				
02B6'CF	DF		0115	300		PUSHAL	RTLOG_FLAGS	; flags
02C2'CF	9F		0119	301		PUSHAB	RTLOG_BUF	; string
7E			011D	302		MOVZWL	RTLOG_DESC,-(SP)	; length
00000000'GF	03	FB	0122	303		CALLS	#3,G^CIB\$CVT_HTB	; convert hex to binary
	04	50	0129	304		BLBS	R0,SS	; go if ok
02B6'CF	D4		012C	305		CLRL	RTLOG_FLAGS	; otherwise zero
			0130	306				
			0130	307	SS:	\$GETDVI_S -		; Get the device characteristics
			0130	308		DEVNAM = TTYDESC, -		; of the translated SYSS\$INPUT
			0130	309		ITMLST = DVILIST		
			0130			.GLOBL	SYSS\$GETDVI	
			0130			\$ASNPUSH	0,#0	
00000000			0130			\$ST1 = 0		
			0130			.IF	IDN,<0>,<0>	
00000001			0130			.IF	IDN,<#0>,<#0>	
			0130			\$ST1 = 1		
			0130			.ENDC		
00000001			0130			.ENDC		
			0130			.IF	NE \$ST1	
7E	7C		0130			CLRQ	-(SP)	
			0132			.IFF		
			0132			\$PUSHADR	0,CONTEXT=0	
			0132			PUSHL	#0	
			0132			.ENDC		
			0132					
			0132			\$PUSHADR	0	
00	DD		0132			.IF	IDN,0,0	
			0134			PUSHL	#0	
			0134			.IFF		
			0134			PUSHAL	0	
			0134			.ENDC		
			0134					
			0134			\$PUSHADR	0,CONTEXT=0	
00	DD		0134			.IF	IDN,0,0	
			0134			PUSHL	#0	
			0136			.IFF		

```
0136          PUSHAQ 0
0136          .ENDC
0136          $PUSHADR DVILIST
0136          .IF      IDN,0,DVILIST
0136          PUSHL  #0
0136          .IFF
055E'CF      DF 0136          PUSHAL DVILIST
0136          .ENDC
013A          $PUSHADR TTYDESC,CONTEXT=Q
013A          .IF      IDN,0,TTYDESC
013A          PUSHL  #0
013A          .IFF
04D6'CF      7F 013A          PUSHAQ TTYDESC
013E          .ENDC
013E
7E      00      3C 013E          MOVZWL #0,-(SP)
00      00      DD 0141          PUSHL  #0
00000000'GF  08  FB 0143          CALLS  #8,G^SYSS$GETDVI
014A
01 50      E8 014A          ONERROR RET          ; Die if any error
04      04 014A          BLBS  R0,30001$
014D          RET
014E          30001$:
014E          311
014E          312
014E          313
014E          314
014E          $GETDEV_S -          ; RSX, ETC. WANTS THIS
014E          DEVNAM = TTYDESC,-
014E          PRIBUF = TERMCHAR
014E          .GLOBL SYSS$GETDEV
014E          $PUSHADR 0,CONTEXT=Q
014E          .IF      IDN,0,0
00      DD 014E          PUSHL  #0
0150          .IFF
0150          PUSHAQ 0
0150          .ENDC
0150          $PUSHADR 0,CONTEXT=W
0150          .IF      IDN,0,0
00      DD 0150          PUSHL  #0
0152          .IFF
0152          PUSHAW 0
0152          .ENDC
0152          $PUSHADR TERMCHAR,CONTEXT=Q
0152          .IF      IDN,0,TERMCHAR
0152          PUSHL  #0
0152          .IFF
0084'CF      7F 0152          PUSHAQ TERMCHAR
0156          .ENDC
0156          $PUSHADR 0,CONTEXT=W
0156          .IF      IDN,0,0
00      DD 0156          PUSHL  #0
0158          .IFF
0158          PUSHAW 0
```



```
0158 .ENDC
0158
0158 $PUSHADR TTYDESC,CONTEXT=0
0158 .IF IDN,0,TTYDESC
0158 PUSHL #0
0158 .IFF
0158 PUSHAQ TTYDESC
0158 .ENDC
04D6'CF 7F 0158
015C
015C
00000000'GF 05 FB 015C
0163
0163 315 ONERROR RET
0163 BLBS RO,30002$
0166 RET
0167 30002$:
0167
0167 316
0060'CF 0054'CF 90 0167 317 MOVB DEVCLASS TEMP,DEVCLASS ; Pack the data correctly
0061'CF 0058'CF 90 016E 318 MOVB DEVTYPE TEMP,DEVTYPE
0062'CF 005C'CF 80 0175 319 MOVW DEVBUFSIZ_TEMP,DEVBUFSIZ
017C 320
00'8F 0060'CF 91 017C 321 CMPB DEVCLASS, #DC$_TERM ; Is it a terminal?
12 13 0182 322 BEQL 10$ ; Yes
0184 323 $PUTMSG_S - ; Output an error message
0184 324 MSGVEC = NOTTERM ; saying SYS$COMMAND not a terminal
0184
00 DD 0184
0186
0186
00 DD 0186
0188
0188
0188
0188
0188
0188
00 DD 0188
018A
018A
018A
018A
018A
018A
018A
018A
011C'CF DF 018A
018E
018E
00000000'GF 04 FB 018E
0195
0195 325 RET
0196 326
0196 327
0196 328 ; Call RTL routine to assign channel and associate mailbox to terminal
0196 329
0196 330 10$:
0158'CF 3F 0196 331 PUSHAW TERMMBXCHAN ; Arg #5 is the terminal mailbox chan
```

```
014C'CF 3F 019A 332 PUSHAW READCHAN ; Arg #4 is the terminal input channel
0764'CF DF 019E 333 PUSHAL MAXMSGSI2 ; Arg #3 is the message buffer quota
0764'CF DF 01A2 334 PUSHAL MAXMSGSI2 ; Arg #2 is the maximum message size
04D6'CF 7F 01A6 335 PUSHAQ TTYDESC ; Arg #1 is the terminal device name
00000000'GF 05 FB 01AA 336 CALLS #5, G^LIB$ASN_WTH_MBX ; Assign a channel w/ a mailbox
01 50 E8 01B1 337 ONERROR RET ; Die if any error
04 01B1 BLBS R0,30003$
01B4 RET
01B5 30003$:
01B5
01B5 338
01B5 339 ; Assign a terminal write channel
01B5 340
01B5 341 $ASSIGN_S - ; Assign a channel
01B5 342 DEVNAM = TTYDESC, - ; to the terminal device
01B5 343 CHAN = WRITECHAN ; for terminal output
01B5 .GLOBL SYSS$ASSIGN
01B5 $ASNPU$H 0,#0
00000000 01B5 $$T1 = 0
01B5 .IF IDN,<0>,<0>
00000001 01B5 .IF IDN,<#0>,<#0>
01B5 $$T1 = 1
01B5 .ENDC
01B5 .ENDC
00000001 01B5 .IF NE $$T1
7E 7C 01B5 CLRQ -(SP)
01B7 .IFF
01B7 $PUSHADR 0,CONTEXT=Q
01B7 PUSHL #0
01B7 .ENDC
01B7
01B7 $PUSHADR WRITECHAN,CONTEXT=W
01B7 .IF IDN,0,WRITECHAN
01B7 PUSHL #0
01B7 .IFF
0150'CF 3F 01B7 PUSHAW WRITECHAN
01B8 .ENDC
01B8
01B8 $PUSHADR TTYDESC,CONTEXT=Q
01B8 .IF IDN,0,TTYDESC
01B8 PUSHL #0
01B8 .IFF
04D6'CF 7F 01B8 PUSHAQ TTYDESC
01B8 .ENDC
01B8
00000000'GF 04 FB 01B8 CALLS #4,G^SYSS$ASSIGN
01C6 344 ONERROR RET ; Die if any error
01 50 E8 01C6 BLBS R0,30004$
04 01C9 RET
01CA 30004$:
01CA
01CA 345
01CA 346 ; See if SYSS$INPUT is a file
01CA 347
01CA 348 $OPEN FAB = SYSINFAB ; Open SYSS$INPUT
01CA $RMSCALL OPEN,SYSINFAB,,
```



```
01CA .GLOBL SYSSOPEN
01CA .IF B <SYSINFAB>
01CA CALLG (AP),G^SYSSOPEN
01CA $$TMP=0
01CA .IF NB <>
01CA $$TMP=1
01CA .ENDC
01CA .IF NB <>
01CA $$TMP=1
01CA .ENDC
01CA .IF NE $$TMP
01CA .ERROR ; SYSINFAB= parameter missing;
01CA .ENDC
01CA .ENDC
00000001 01CA .IF NB <SYSINFAB>
01CA $$TMP1=1
01CA .IF NB <>
01CA PUSHAL
01CA $$TMP1=3
01CA .ENDC
01CA .IF NB <>
01CA PUSHAL
01CA .IF EQ <$$TMP1-1>
01CA $$TMP1=2
01CA .ENDC
01CA .IFF
01CA .IF EQ <$$TMP1-3>
01CA PUSHL #0
01CA .ENDC
01CA .ENDC
000000CF 01CA .NTYPE $$TMP2, SYSINFAB
00000070 01CA .IF EQ <<$$TMP2&^XF0>-^X50>
01CA PUSHL SYSINFAB
01CA .IFF
01CA .IF EQ <<$$TMP2&^XF0>-^X10>
01CA PUSHL SYSINFAB
01CA .IFF
021C'CF DF 01CA PUSHAL SYSINFAB
01CE .ENDC
01CE .ENDC
00000000'GF 01 FB 01CE CALLS #$$TMP1,G^SYSSOPEN
01D5 .ENDC
01D5
01D5
01 50 E8 01D5 349 ONERROR RET
04 01D8 BLBS R0,30005$
01D9 RET
01D9 30005$:
00000000'8F E0 01D9 350 BBS #DEV$V TRM,-
15 025C'CF 01DF 351 SYSINFAB+FAB$L_DEV,20$ ; Branch if terminal
02B0'CF 96 01E3 352 INCB INDFLAG ; Indicate indirect file
01E7 353 $CONNECT RAB = SYSINRAB ; Connect to input stream
01E7 $RMSCALL CONNECT, SYSINRAB,,
01E7 .GLOBL SYSSCONNECT
01E7 .IF B <SYSINRAB>
01E7 CALLG (AP),G^SYSSCONNECT
```

```
01E7          $$TMP=0
01E7          .IF      NB <>
01E7          $$TMP=1
01E7          .ENDC
01E7          .IF      NB <>
01E7          $$TMP=1
01E7          .ENDC
01E7          .IF      NE $$TMP
01E7          .ERROR                                ; SYSINRAB= parameter missing;
01E7          .ENDC
01E7          .ENDC
01E7          .IF      NB <SYSINRAB>
00000001 01E7          $$TMP1=1
01E7          .IF      NB <>
01E7          PUSHAL
01E7          $$TMP1=3
01E7          .ENDC
01E7          .IF      NB <>
01E7          PUSHAL
01E7          .IF      EQ <$$TMP1-1>
01E7          $$TMP1=2
01E7          .ENDC
01E7          .IFF
01E7          .IF      EQ <$$TMP1-3>
01E7          .IF      EQ <$$TMP1-3>
01E7          .ENDC
01E7          .ENDC
000000CF 01E7          .NTYPE  $$TMP2,SYSINRAB
000J007G 01E7          .IF      EQ <<$$TMP2&^XF0>-^X50>
01E7          PUSHL  SYSINRAB
01E7          .IFF
01E7          .IF      EQ <<$$TMP2&^XF0>-^X10>
01E7          PUSHL  SYSINRAB
01E7          .IFF
01E7          .IF      EQ <<$$TMP2&^XF0>-^X10>
01E7          PUSHL  SYSINRAB
01E7          .IFF
026C'CF  DF 01E7          PUSHAL  SYSINRAB
01EB          .ENDC
01EB          .ENDC
0000G000'GF 01  FB 01EB          CALLS  #$$TMP1,G^SYSS$CONNECT
01F2          .ENDC
01F2
01F2
01F2 354          ONERROR RET
01F2          BLBS    R0,30006$
01F5          RET
01F6          30006$:
01F6
01F6 355          BRB    30$
01F8 356
01F8 357 20$:      $CLOSE  FAB = SYSINFAB          ; Won't use it
01F8          $RMSCALL  CLOSE,SYSINFAB,,
01F8          .GLOBL  SYS$CLOSE
01F8          .IF      B <SYSINFAB>
01F8          CALLG  (AP),G^SYSS$CLOSE
01F8          $$TMP=C
01F8          .IF      NB <>
01F8          $$TMP=1
01F8          .ENDC
```

```
01F8 .IF NB <>
01F8 $$TMP=1
01F8 .ENDC
01F8 .IF NE $$TMP
01F8 .ERROR ; SYSINFAB= parameter missing;
01F8 .ENDC
01F8 .ENDC
01F8 .IF NB <SYSINFAB>
00000001 01F8 $$TMP1=1
01F8 .IF NB <>
01F8 PUSHAL
01F8 $$TMP1=3
01F8 .ENDC
01F8 .IF NB <>
01F8 PUSHAL
01F8 .IF EQ <$$TMP1-1>
01F8 $$TMP1=2
01F8 .ENDC
01F8 .IFF
FFFFF0FE 01F8 .IF EQ <$$TMP1-3>
01F8 PUSHL #0
01F8 .ENDC
01F8 .ENDC
000000CF 01F8 .NTYPE $$TMP2, SYSINFAB
0000007J 01F8 .IF EQ <<$$TMP2&^XF0>-^X50>
01F8 PUSHL SYSINFAB
01F8 .IFF
000000B0 01F8 .IF EQ <<$$TMP2&^XF0>-^X10>
01F8 PUSHL SYSINFAB
01F8 .IFF
021C'CF DF 01F8 PUSHAL SYSINFAB
01FC .ENDC
01FC .ENDC
0000000J'GF 01 FB 01FC CALLS #$$TMP1,G^SYSSCLOSE
0203 .ENDC
0203
0203
0203 358 30$:
0203 359
0203 ;
0203 ; Check for /LOG [=filespec]
0203 360
0203 361
0203 362
0167'CF 7F 0203 363
00000000'GF 01 FB 0207 364
21 50 E9 020E 365
0211 366
01A0'CF 7F 0211 367
0167'CF 7F 0215 368
00000000'GF 02 FB 0219 369
0E 50 E9 0220 370
0223 371
01A0'CF 7F 0223 372
00000000'GF 01 FB 0227 373
01 50 E8 022E 374
04 0231 375 61$:
0232 376 62$:
0232 377
;
PUSHAQ LOG_DESC ; LOG label
CALLS #1,G^CLIS$PRESENT ; See if present
BLBC R0, 62$ ; branch if not present
PUSHAQ LOG_FILE_DESC ; return buffer
PUSHAQ LOG_DESC ; LOG label
CALLS #2,G^CLIS$GET_VALUE ; get value
BLBC R0, 61$ ; continue if no error
PUSHAQ LOG_FILE_DESC ; use this file
CALLS #1,G^CTERM$OPEN_LOG ; Open log file
BLBS R0, 62$ ; Branch if ok
RET ; exit on error
```

```
0232 378 ; Get node name from CLI
0232 379 ;
0198'CF 7F 0232 380 PUSHAQ NODE_NAME_DESC ; Return buffer
017D'CF 7F 0236 381 PUSHAQ NODEDESC ; Parameter name
00000000'GF 02 FB 023A 382 CALLS #2, G^CLISGET_VALUE ; Get node name
0241 383 ONERROR RET ; Exit on error
01 50 E8 0241 BLBS RO,30007$
04 0244 RET
0245 30007$:
0245
0245 384
0245 385 ;
0245 386 ; Check for /DTE
0245 387 ;
0245 388
0172'CF 7F 0245 389 PUSHAQ DTE_DESC ; DTE label
00000000'GF 01 FB 0249 390 CALLS #1, G^CLISPRESENT ; See if present
29 50 E9 0250 391 BLBC RO, NOT_DTE ; branch if not present
0253 392
0253 393 ;
0253 394 ; Kludge NODENAME so CR,LF happen on exit
0253 395 ;
50 01B0'CF 3C 0253 396 MOVZWL NODENAME,RO ; length
50 01B4'CF C0 0258 397 ADDL2 NODENAME+4,RO ; plus address
01B0'CF 02 A0 025D 398 ADDW #2,NODENAME ; add to length
60 0DOA 8F B0 0262 399 MOVW #^XODOA,(RO) ; CR,LF
0267 400
7E 0150'CF 3C 0267 401 MOVZWL WRITECHAN,-(SP) ; command channel
0198'CF 7F 026C 402 PUSHAQ NODE_NAME_DESC ; value of P1
00000000'GF 02 FB 0270 403 CALLS #2, G^TERM$EMULATE ; call terminal emulation code
0277 404 ONERROR RET ; exit immediately on error
01 50 E8 0277 BLBS RO,30008$
04 027A RET
027B 30008$:
027B
05 027B 405 RSB ; otherwise, return to hiber code
027C 406
027C 407 NOT_DTE:
027C 408 ;
027C 409 ; Assign a channel for ^C and ^Y handling
027C 410 ;
027C 411 $ASSIGN_S - ; Assign a channel
027C 412 DEVNAM = TTYDESC, - ; to the terminal device
027C 413 CHAN = CNTRLCHAN ; for control (AST's)
027C
027C .GLOBL SYSS$ASSIGN
027C $ASNUPUSH 0,#0
00000000 027C $ST1 = 0
027C .IF IDN,<0>,<0>
027C .IF IDN,<#0>,<#0>
00000001 027C $ST1 = 1
027C .ENDC
027C .ENDC
00000001 027C .IF NE $ST1
7E 7C 027C CLRQ -(SP)
027E .IFF
027E $PUSHADR 0,CONTEXT=0
027E PUSHL #0
```

```
027E .ENDC
027E $PUSHADR CNTRLCHAN,CONTEXT=W
027E .IF IDN,0,CNTRLCHAN
027E PUSHL #0
027E .IFF
0154'CF 3F 027E PUSHAW CNTRLCHAN
0282 .ENDC
0282 $PUSHADR TTYDESC,CONTEXT=Q
0282 .IF IDN,0,TTYDESC
0282 PUSHL #0
0282 .IFF
04D6'CF 7F 0282 PUSHAQ TTYDESC
0286 .ENDC
0286 CALLS #4,G^SYSS$ASSIGN
00000000'GF 04 FB 0286
028D 028D 414 ONERROR RET ; Die if any error
01 50 E8 028D BLBS R0,30009$
04 04 0290 RET
0291 30009$:
0291
0291 415
0291 416 ;
0291 417 ; Loop through node name to remove trailing ':'s
0291 418 ;
0291 419
52 0198'CF 7D 0291 420 MOVQ NODE_NAME_DESC, R2 ; Get node name length, address
52 52 3C 0296 421 MOVZWL R2, R2 ; and isolate its real length
0A 13 0299 422 BEQL 50$ ; No length??
3A FF A342 91 029B 423 40$: CMPB -1(R3)[R2], #^A': ; A trailing colon?
09 12 02A0 424 BNEQ 60$ ; Nope
F6 52 F5 02A2 425 SOBGTR R2, 40$ ; Yep, remove it from count and loop
50 0000'8F 3C 02A5 426 50$: MOVZWL #SS$_NOSUCHNODE, R0 ; Error
04 04 02AA 427 RET ; and exit
0000'CF 52 7D 02AB 428
02AB 429 60$: MOVQ R2, REMOTENODE ; Save the node name descriptor
02B0 430
02B0 431 ;
02B0 432 ; Form the network connection string
02B0 433 ;
02B0 434
0189'CF 7F 02B0 435 PUSHAQ OBJ_DESC ; Arg #3 is the right part (obj type)
0000'CF 7F 02B4 436 PUSHAQ REMOTENODE ; Arg #2 is the left part (node name)
01A8'CF 7F 02B8 437 PUSHAQ CONNDESC ; Arg #1 is the resultant string
00000000'GF 03 FB 02BC 438 CALLS #3, G^STR$CONCAT ; Go concatenate for connection string
02C3 439 ONERROR RET ; Exit on error
01 50 E8 02C3
04 04 02C6
02C7 30010$:
02C7
58 01AC'CF 52 C1 02C7 440 ADDL3 R2, CONNDESC+4, R8 ; Address just beyond node name
58 03' C0 02CD 441 ADDL S^#OBJ_C_PREFIX, R8 ; then offset to object number
02D0 442
02D0 443 ;
02D0 444 ; *** TEMPORARY CODE TO DETECT /OLD QUALIFIER
```

```

00000000 015C'CF 7F 02D0 445 :
00000000 GF 01 FB 02D0 446 : pushaq old_desc : *** TEMP
00000000 05 50 E9 02D4 447 : calls #1, g^cli$present : *** TEMP
68 3332 8F B0 02DB 448 : blbc r0, 65$ : *** TEMP; /NOOLD, go try 42 first
00000000 0208'CF 7F 02DE 449 : movw #'A'23', (r8) : *** TEMP; /OLD, change to 23
00000000 GF 01 FB 02E3 450 65$:
00000000 01 50 E8 02E3 451 :
00000000 04 04 02E3 452 :
00000000 0764'CF 3F 02E3 453 :
00000000 0421'CF 9F 02E3 454 :
00000000 0200'CF 7F 02E3 455 :
00000000 01F8'CF 7F 02E3 456 70$: PUSHAB FSTHRU MSG : Address the PSTHRU message desc
00000000 0148'CF 3F 02E7 457 : CALLS #1, G^STR$FREE1_DX : and free up anything in it
00000000 0144'CF 3F 02EE 458 : ONERROR RET : Exit on error
00000000 01A8'CF 7F 02EE 458 : BLBS R0,30011$
00000000 GF 07 FB 02F1 458 : RET
00000000 13 50 E8 02F2 458 : 30011$:
00000000 51 68 B0 02F2 459 : PUSHAB MAXMSGISZ : Arg #7 is the maximum message size
00000000 3332 8F B0 02F6 460 : PUSHAB 180$ : Arg #6 is the message call routine
00000000 BC 12 0325 461 : PUSHAB FINALACS : Arg #5 is the final ACS desc
00000000 00E0 30 0325 462 : PUSHAB FINALPATH : Arg #4 is the final path desc
00000000 00DC 30 0302 463 : PUSHAB MAILCHAN : Arg #3 is the link mailbox channel
00000000 55 0324'CF 9E 0306 464 : PUSHAB LINKCHAN : Arg #2 is the link channel
00000000 0324'CF 9E 030A 465 : PUSHAB CONNDESC : Arg #1 is the "device" desc
00000000 0324'CF 9E 030E 466 : CALLS #7, G^UNSS$NET_CONNECT : Do the remote node connection
00000000 0324'CF 9E 0315 467 : BLBS R0, 80$ : Continue if success completion
00000000 0324'CF 9E 0318 468 :
00000000 0324'CF 9E 0318 469 :
00000000 0324'CF 9E 0318 470 :
00000000 0324'CF 9E 0318 471 :
00000000 0324'CF 9E 0318 472 :
00000000 51 68 B0 0318 473 : MOVW (R8), R1 : Save the object type we just tried
00000000 3332 8F B0 031B 474 : MOVW #'A'23', (R8) : then replace it object type 23
00000000 BC 12 0325 475 : CMPW R1, #'A'23' : Did we just try object type 23?
00000000 00E0 30 0325 476 : BNEQ 70$ : Nope, so go try object type 23
00000000 00DC 30 0327 477 : BSBW 170$ : Go output any saved PSTHRU message(s)
00000000 55 0324'CF 9E 032A 478 : RET : then exit with the error
00000000 0324'CF 9E 032B 479 :
00000000 0324'CF 9E 032B 480 80$: BSBW 170$ : Go output any saved PSTHRU message(s)
00000000 0324'CF 9E 032E 481 :
00000000 0324'CF 9E 032E 482 :
00000000 0324'CF 9E 032E 483 :
00000000 0324'CF 9E 032E 484 :
00000000 0324'CF 9E 032E 485 :
00000000 0324'CF 9E 032E 486 :
00000000 0324'CF 9E 032E 487 :
00000000 0324'CF 9E 0333 488 : MOVAB FIRSTMSG, R5 : Address of area to receive message
00000000 0324'CF 9E 0333 489 : $QIOW_S - : Read CONFIG message
00000000 0324'CF 9E 0333 490 : CHAN = LINKCHAN, - : on the link channel
00000000 0324'CF 9E 0333 491 : FUNC = #IOS$ READVBLK, - : reading obviously
00000000 0324'CF 9E 0333 492 : IOSB = AST$Q IOSB(R5), - : use an IOSB
00000000 0324'CF 9E 0333 493 : P1 = AST$T BOF(R5), - : into this buffer
00000000 0324'CF 9E 0333 494 : P2 = #MAXMSG : which is this long
00000000 0324'CF 9E 0333 495 : .GLOBL SYSSQIOW
00000000 0324'CF 9E 0333 496 : $PUSHTWO #0,#0
00000000 0324'CF 9E 0333 497 : $ST1 = 0
00000000 0324'CF 9E 0333 498 : .IF IDN,<#0>,<#0>

```



```
00000001 0333 .IF IDN,<#0>,<#0>
0333 $ST1 = 1
0333 .ENDC
0333 .ENDC
00000001 0333 .IF NE $ST1
7E 7C 0333 CLRQ -(SP)
0335 .IFF
0335 PUSHL #0
0335 PUSHL #0
0335 .ENDC
0335
0335 $PUSHTWO #0,#0
00000000 0335 $ST1 = 0
0335 .IF IDN,<#0>,<#0>
0335 .IF IDN,<#0>,<#0>
00000001 0335 $ST1 = 1
0335 .ENDC
0335 .ENDC
00000001 0335 .IF NE $ST1
7E 7C 0335 CLRQ -(SP)
0337 .IFF
0337 PUSHL #0
0337 PUSHL #0
0337 .ENDC
0337
0000041A 8F DD 0337 PUSHL #MAXMSG
0337 $PUSHADR AST$T_BUF(R5)
0337 .IF IDN,0,AST$T_BUF(R5)
0337 PUSHL #0
0337 .IFF
26 A5 DF 0337 PUSHAL AST$T_BUF(R5)
0340 .ENDC
0340
0340 $QIOPUSH #0,0
00000000 0340 $ST1 = 0
0340 .IF IDN,<#0>,<#0>
0340 .IF IDN,<0>,<0>
00000001 0340 $ST1 = 1
0340 .ENDC
0340 .ENDC
00000001 0340 .IF NE $ST1
7E 7C 0340 CLRQ -(SP)
0342 .IFF
0342 PUSHL #0
0342 $PUSHADR 0
0342 .ENDC
0342
0342 $PUSHADR AST$Q_IOSB(R5),CONTEXT=0
0342 .IF IDN,0,AST$Q_IOSB(R5)
0342 PUSHL #0
0342 .IFF
04 A5 7F 0342 PUSHAQ AST$Q_IOSB(R5)
0345 .ENDC
0345
0345 MOVZWL #IOS_READVBLK,-(SP)
7E 7E 31 3C 0345 MOVZWL LINK$CHAN,-(SP)
7E 0144'LF 3C 0348
00 DD 0340 PUSHL #0
```

Address	Op	Op2	Op3	Op4	Op5	Op6	Op7	Op8	Op9	Op10	Op11	Op12	Op13	Op14	Op15	Op16	Op17	Op18	Op19	Op20	Op21	Op22	Op23	Op24	Op25	Op26	Op27	Op28	Op29	Op30	Op31	Op32	Op33	Op34	Op35	Op36	Op37	Op38	Op39	Op40	Op41	Op42	Op43	Op44	Op45	Op46	Op47	Op48	Op49	Op50	Op51	Op52	Op53	Op54	Op55	Op56	Op57	Op58	Op59	Op60	Op61	Op62	Op63	Op64	Op65	Op66	Op67	Op68	Op69	Op70	Op71	Op72	Op73	Op74	Op75	Op76	Op77	Op78	Op79	Op80	Op81	Op82	Op83	Op84	Op85	Op86	Op87	Op88	Op89	Op90	Op91	Op92	Op93	Op94	Op95	Op96	Op97	Op98	Op99	Op100	Op101	Op102	Op103	Op104	Op105	Op106	Op107	Op108	Op109	Op110	Op111	Op112	Op113	Op114	Op115	Op116	Op117	Op118	Op119	Op120	Op121	Op122	Op123	Op124	Op125	Op126	Op127	Op128	Op129	Op130	Op131	Op132	Op133	Op134	Op135	Op136	Op137	Op138	Op139	Op140	Op141	Op142	Op143	Op144	Op145	Op146	Op147	Op148	Op149	Op150	Op151	Op152	Op153	Op154	Op155	Op156	Op157	Op158	Op159	Op160	Op161	Op162	Op163	Op164	Op165	Op166	Op167	Op168	Op169	Op170	Op171	Op172	Op173	Op174	Op175	Op176	Op177	Op178	Op179	Op180	Op181	Op182	Op183	Op184	Op185	Op186	Op187	Op188	Op189	Op190	Op191	Op192	Op193	Op194	Op195	Op196	Op197	Op198	Op199	Op200	Op201	Op202	Op203	Op204	Op205	Op206	Op207	Op208	Op209	Op210	Op211	Op212	Op213	Op214	Op215	Op216	Op217	Op218	Op219	Op220	Op221	Op222	Op223	Op224	Op225	Op226	Op227	Op228	Op229	Op230	Op231	Op232	Op233	Op234	Op235	Op236	Op237	Op238	Op239	Op240	Op241	Op242	Op243	Op244	Op245	Op246	Op247	Op248	Op249	Op250	Op251	Op252	Op253	Op254	Op255	Op256	Op257	Op258	Op259	Op260	Op261	Op262	Op263	Op264	Op265	Op266	Op267	Op268	Op269	Op270	Op271	Op272	Op273	Op274	Op275	Op276	Op277	Op278	Op279	Op280	Op281	Op282	Op283	Op284	Op285	Op286	Op287	Op288	Op289	Op290	Op291	Op292	Op293	Op294	Op295	Op296	Op297	Op298	Op299	Op300	Op301	Op302	Op303	Op304	Op305	Op306	Op307	Op308	Op309	Op310	Op311	Op312	Op313	Op314	Op315	Op316	Op317	Op318	Op319	Op320	Op321	Op322	Op323	Op324	Op325	Op326	Op327	Op328	Op329	Op330	Op331	Op332	Op333	Op334	Op335	Op336	Op337	Op338	Op339	Op340	Op341	Op342	Op343	Op344	Op345	Op346	Op347	Op348	Op349	Op350	Op351	Op352	Op353	Op354	Op355	Op356	Op357	Op358	Op359	Op360	Op361	Op362	Op363	Op364	Op365	Op366	Op367	Op368	Op369	Op370	Op371	Op372	Op373	Op374	Op375	Op376	Op377	Op378	Op379	Op380	Op381	Op382	Op383	Op384	Op385	Op386	Op387	Op388	Op389	Op390	Op391	Op392	Op393	Op394	Op395	Op396	Op397	Op398	Op399	Op400	Op401	Op402	Op403	Op404	Op405	Op406	Op407	Op408	Op409	Op410	Op411	Op412	Op413	Op414	Op415	Op416	Op417	Op418	Op419
---------	----	-----	-----	-----	-----	-----	-----	-----	-----	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------


```
00 DD 03A0 .IF IDN,0,0
      03A0 PUSHL #0
      03A2 .IFF
      03A2 PUSHAQ 0
      03A2 .ENDC
      03A2 $PUSHADR 0
      00 DD 03A2 .IF IDN,0,0
      03A2 PUSHL #0
      03A4 .IFF
      03A4 PUSHAL 0
      03A4 .ENDC
      03A4 $PUSHADR NCTVMS
      03A4 .IF IDN,0,NOTVMS
      03A4 PUSHL #0
      03A4 .IFF
      0138'CF DF 03A4 PUSHAL NOTVMS
      03A8 .ENDC
00000000'GF 04 FB 03A8 CALLS #4,G^SYSS$PUTMSG
      04 03AF 521 RET
      03B0 522
      03B0 523
      03B0 524 ; Use the support bit mask that was returned to determine which
      03B0 525 ; of the protocol modules to call.
      03B0 526
      03B0 527
      0319'CF 28 A5 90 03B0 528 140$: MOVB AST$T_BUF+2(R5),PROTO_ECO ; Save eco level
      031A'CF 2A A5 B0 03B6 529 MOVW AST$T_BUF+4(R5),HOST_OPSYS ; Save host operating system
      52 7FFC'CF 9E 03BC 530
      0319'CF 52 04 C0 03C1 531 150$: MOVAB PROTOTBL-4, R2 ; Get (biased) pointer to protocols
      031A'CF 52 04 C0 03C1 532 ADDL #4, R2 ; Index over address to bit number
      00000000'8F 52 D1 03C4 533 CMPL R2, #ENDPROTO ; Are we out of protocols?
      031A'CF 52 D1 03CB 534 BGEQU 130$ ; Yep, protocol not supported
      82 2C A5 B3 03CD 535 BITW AST$T_BUF+6(R5),(R2)+ ; Not yet, does the bit match?
      EE 13 03D1 536 BEQL 150$ ; No match, loop for next protocol
      03D3 537
      03D3 538 ; R2 is now address of protocol module routine
      03D3 539
      031C'CF DF 03D3 540
      031C'CF DF 03D7 541
      00000000'GF 02 FB 03DB 542
      03E2 543
      031C'CF DF 03E2 544
      031C'CF DF 03E6 545
      00000000'GF 02 FB 03EA 546
      03F1 547
      03F1 548
      03F1 549
      03F1 550
      00000000'GF 01 DD 03F1
      00000000'GF 01 FB 03F3
      03FA
      0000'8F 50 B1 03FA 551
      CMPW R0, #SS$_WASCLR ; Was it already off?
```

```
0320'CF 04 13 03FF 552      BEQL 160$      ; Nope, we'll turn it back on at exit
                                D6 0401 553      INCL OLDSETRWM ; Yep, we'll keep it off at exit
                                0405 554
                                0405 555      ;
                                0405 556      ; Call the appropriate protocol module...
                                0405 557      ;
                                0405 558 160$:
00 B2 00 FB 0405 559      CALLS #0, @(<2)      ; Go start up selected protocol
                                05 0409 560      RSD ; Done
                                040A 561
0208'CF B5 040A 562 170$: TSTW PSTHRU_MSG      ; Any saved PSTHRU message(s)?
                                10 13 040E 563      BEQL 175$ ; Nope
                                50 DD 0410 564      PUSHL R0 ; Yep, save the status code
0208'CF 7F 0412 565      PUSHAQ PSTHRU_MSG      ; Address the saved message(s)
00000000'GF 01 FB 0416 566      CALLS #1, G^IB$PUT_OUTPUT ; and go output them
                                50 8E D0 041D 567      MOVL (SP)+, R0 ; Restore the status code
                                05 0420 568 175$: RSB ; Exit
                                0421 569
06 AC 01FE 8F 0000 0421 570 180$: .WORD ^M<> ; Message call back routine
                                B0 0423 571      MOVW #REMS_FACILITY, 4+2(AP) ; Replace facility code with ours
                                0429 572      $PUTMSG_S - ; Use $PUTMSG
                                0429 573      MSGVEC = (AP), - ; with the supplied message vector
                                0429 574      ACTRTN = 190$ ; calling this action routine
                                0429
                                00 DD 0429 .GLOBL SYSS$PUTMSG
                                042B PUSHL #0
                                042B $PUSHADR 0,CONTEXT=0
                                00 DD 042B .IF IDN,0,0
                                042D PUSHL #0
                                042D .IFF
                                042D PUSHAQ 0
                                042D .ENDC
                                042D
                                042D $PUSHADR 190$
                                042D .IF IDN,0,190$
                                042D PUSHL #0
                                042D .IFF
                                043B'CF DF 0431 C'2D
                                0431 .ENDC
                                0431
                                0431 $PUSHADR (AP)
                                0431 .IF IDN,0,(AP)
                                0431 PUSHL #0
                                0431 .IFF
                                6C DF 0431 PUSHAL (AP)
                                0433 .ENDC
                                0433
00000000'GF 04 FB 0433      CALLS #4,G^SYSS$PUTMSG
                                043A
                                04 043A 575      RET ; Return
                                043B 576
                                0000 043B 577 190$: .WORD ^M<> ; Action routine for $PUTMSG
                                B5 043D 578      TSTW PSTHRU_MSG ; Anything saved yet?
                                OF 13 0441 579      BEQL 200$ ; Nope
                                0210'CF 7F 0443 580      PUSHAQ PSTHRU_CRLF ; Arg #2 is the string to append
                                0208'CF 7F 0447 581      PUSHAQ PSTHRU_MSG ; Arg #1 is the string to append to
00000000'GF 02 FB 044B 582      CALLS #2, G^STR$APPEND ; Go save message by appending
                                04 BC 7F 0452 583 200$: PUSHAQ @4(AP) ; Arg #2 is the string to append
```

```
00000000'0208'CF 7F 0455 584 PUSHAQ PSTHRU MSG ; Arg #1 is the string to append to
00000000'GF 02 FB 0459 585 CALLS #2, G^STR$APPEND ; Gc save message by appending
50 04 D4 0460 586 CLRL R0 ; Say we don't want any output now
04 0462 587 RET ; then return
0463 588
0000 0463 589 CTERM_RT: ; CTERM protocol initialization
0465 590 .WORD ^M<> ; No register(s) to save
02B2'CF 96 0465 592 INCB CTERM_FLAG ; Indicate CTERM protocol
0469 593
50 04E9'CF 7E 0469 594 movaq infomsg1,R0 ; assume not vax to vax
031A'CF 07 B1 046E 595 C^PW #7,HOST_OPSYS ; Talking to VMS?
0A 12 0473 596 BNLQ 10$ ; nope
10 A8 0475 597 BLSW #FLGSM,VAXHOST,-
02B2'CF 0477 598 CTERM_FLAG ; Set flag
50 0522'CF 7E 047A 599 movaq infomsg2,R0 ; vax to vax
047F 600 10$:
00 E1 047F 601 BBC #RTLOG$V,BANNER,- ; Branch if not requested
11 02B6'CF 0481 602 RTLOG_FLAGS,20$
50 DD 0485 603 pushl R0
01 DD 0487 604 pushl #1
01FE1130'8F DD 0489 605 pushl #<shr$text:sts$k_info>!<rem$facility@16> ;
00000000'GF 03 FB 048F 606 calls #3,g^lib$signal ;
0496 607
0000'CF 00 ^B 0496 608 20$: CALLS #0,VMSRT ; Go join the VMSRT protocol
04 049B 609 RET ; then exit
```

```
049C 611 .SBTTL RECORD_QUIT - snapshot QUIT info
049C 612
049C 613 RECORD_QUIT::
049C 614
030D'CF 6E DO 049C 615      MOVL      (SP),QUIT_PC      ; save caller PC
04A1 616
04A1 617      $GETJPIW,S -
04A1 618      EFN = #5,-
04A1 619      PIDADR = LOCAL_PID,-
04A1 620      ITMLST = GETJPI_ITMLST,-
04A1 621      IOSB = JPI_IOSB
04A1      .GLOBL SYSS$GETJPIW
04A1      $IOPUSH #0,0
00000000 04A1      $ST1 = 0
04A1      .IF      IDN,<#0>,<#0>
04A1      .IF      IDN,<0>,<0>
00000001 04A1      $ST1 = 1
04A1      .ENDC
04A1      .ENDC
00000001 04A1      .IF      NE $ST1
7E 7C 04A1      CLRQ      -(SP)
04A3      .IFF
04A3      PUSHL      #0
04A3      $PUSHADR 0
04A3      .ENDC
04A3
04A3      $PUSHADR JPI_IOSB,CONTEXT=0
04A3      .IF      IDN,0,JPI_IOSB
04A3      PUSHL      #0
0004'CF 7F 04A3      .IFF
04A3      PUSHAQ     JPI_IOSB
04A7      .ENDC
04A7
04A7      $PUSHADR GETJPI_ITMLST
04A7      .IF      IDN,0,GETJPI_ITMLST
04A7      PUSHL      #0
04A7      .IFF
000C'CF DF 04A7      PUSHAL     GETJPI_ITMLST
04AB      .ENDC
04AB
04AB      $PUSHADR 0,CONTEXT=0
04AB      .IF      IDN,0,0
00  DD 04AB      PUSHL      #0
04AD      .IFF
04AD      PUSHAQ     0
04AD      .ENDC
04AD
04AD      $PUSHADR LOCAL_PID
04AD      .IF      IDN,0,LOCAL_PID
04AD      PUSHL      #0
04AD      .IFF
0000'CF DF 04AD      PUSHAL     LOCAL_PID
04B1      .ENDC
04B1
00000000'GF 05 DD 04B1      PUSHL      #5
07 FB 04B3      CALLS      #7,G*SYSS$GETJPIW
04BA
```

RTPAD
V04-000

- FEMOTE TERMINAL PROGRAM
RECORD_QUIT - snapshot QUIT info

D 1

16-SEP-1984 02:15:27 VAX/VMS Macro V04-00
5-SEP-1984 03:15:47 [RTPAD.SRC]RTPAD.MAR;1

Page 34
(2)

RTP
V04

50	08 50	E9	04BA	622	BLBC	RO,100\$
	0004 CF	7C	04BD	623	MOVZWL	JPI,IOSB,RO
	00 50	E9	04C2	624	BLBC	RO,100\$
			04C5	625		100\$:
		05	04C5	626	RSB	
			04C6	627		

```
04C6 629 .SBTTL READ ONLY DATA
04C6 630
4F 4E 24 53 59 53 000004CE'010E0000' 04C6 631 SYS$NODE: .ASCID /SYS$NODE/
45 44 04D4
4F 43 24 53 59 53 000004DE'010E0000' 04D6 632 TTYDESC:: .ASCID /SYS$COMMAND/
44 4E 41 4D 4D 04E4
04E9 633
65 74 6F 6D 65 72 000004F1'010E0000' 04E9 634 infomsg1: .ascid /remote terminal is using CTERM protocol (non-vax)/ ; *** TE
73 69 20 6C 61 6E 69 6D 72 65 74 20 04F7
4D 52 45 54 43 20 67 6E 69 73 75 20 0503
6E 28 20 6C 6F 63 6F 74 6F 72 70 20 050F
29 78 61 76 2D 6E 6F 051B
65 74 6F 6D 65 72 0000052A'010E0000' 0522 635 infomsg2: .ascid /remote terminal is using CTERM protocol (vax-to-vax)/ ; ***
73 69 20 6C 61 6E 69 6D 72 65 74 20 0530
4D 52 45 54 43 20 67 6E 69 73 75 20 053C
76 2C 20 6C 6F 63 6F 74 6F 72 70 20 0548
79 78 61 76 2D 6F 74 2D 78 61 0554
055E 636
0004 0004 055E 637 DVILIST:
00000000'00000054' 0562 638 .WORD 4,DVIS_DEVCLASS ; Device class
056A 639 .ADDRESS DEVCLASS_TEMP,0
0006 0004 056A 640
00000000'00000058' 056A 641 .WORD 4,DVIS_DEVTYPE ; Device type
056E 642 .ADDRESS DEVTYPE_TEMP,0
0576 643
0008 0004 0576 644 .WORD 4,DVIS_DEVBUFSIZ ; Device buffer size
00000000'0000005C' 057A 645 .ADDRESS DEVBUFSIZ_TEMP,0
0582 646
000A 0004 0582 647 .WORD 4,DVIS_DEVDEPEND ; Device dependant data (1)
00000000'00000064' 0586 648 .ADDRESS DEVDEPEND,0
058E 649
001C 0004 058E 650 .WORD 4,DVIS_DEVDEPEND2 ; Device dependant data (2)
00000000'00000068' 0592 651 .ADDRESS DEVDEPEND2,0
059A 652
0020 0010 059A 653 .WORD 16,DVIS_DEVNAM ; Device name and length
0000006C'00000070' 059E 654 .ADDRESS DEVNAM,DEVNAMLEN
05A6 655
000C 0004 05A6 656 .WORD 4,DVIS_UNIT ; Device unit number
00000000'00000080' 05AA 657 .ADDRESS TERMUNIT,0
05B2 658
00000000 05B2 659 .LONG 0 ;END OF LIST
05B6 660
```



```
00000000 05B6 662 .SBTTL READ WRITE DATA
00000000 663 .PSECT _RTPAD, LONG
00000000 664
00000000 665 LOCAL PID: .LONG 0 ; no pid
0000000C 0004 666 JPI_IOSB: .BLKL 2 ; iosb
000C 667
000C 668 GETJPI_ITMLST:
000C 669 .WORD 4, JPI$ ASTLM
0000005C 00000040 0010 670 .LONG ASTLM, LEN
00000000 00000004 0018 671 .WORD 4, JPI$ ASTCNT
000C0050 00000044 001C 672 .LONG ASTCNT, LEN
00000000 00000004 0024 673 .WORD 4, JPI$ BIOLM
00000050 00000048 0028 674 .LONG BIOLM, LEN
00000000 00000004 0030 675 .WORD 4, JPI$ BIOCNT
00000050 0000004C 0034 676 .LONG BIOCNT, LEN
003C 677
00000000 003C 678 .LONG 0 ; end of list
0040 679
00000000 0040 680 ASTLM: .LONG 0
00000000 0044 681 ASTCNT: .LONG 0
00000000 0048 682 BIOLM: .LONG 0
00000000 004C 683 BIOCNT: .LONG 0
00000000 0050 684 LEN: .LONG 0
0054 685
0054 686 ;
0054 687 ; Returned data area for $GETDVI
0054 688 ;
0054 689
00000000 0054 690 DEVCLASS_TEMP: .LONG 0 ; Temp locations since getdvi wants longs
00000000 0058 691 DEVTYPE_TEMP: .LONG 0
00000000 005C 692 DEVBUFSIZ_TEMP: .LONG 0
0060 693
0060 694 CHAR_BLOCK:: ; VMS characteristics
00000061 0060 695 DEVCLASS: .BLKB 1
00000062 0061 696 DEVTYPE: .BLKB 1
00000064 0062 697 DEVBUFSIZ: .BLKW 1
00000068 0064 698 DEVDEPEND: .BLKL 1
0000006C 0068 699 DEVDEPEND2: .BLKL 1
006C 700
00000000 006C 701 DEVNAMLEN:: .LONG 0 ;
0070 702
00000080 0070 703 DEVNAM:: .BLKB 16 ;
0080 704
00000000 0080 705 TERMUNIT:: .LONG 0 ;
0084 706
0084 707 TERMCHAR::
0000008C 00000084 0084 708 .LONG DIB$K_LENGTH+16, 1$ ; Term characteristics for RSA, etc.
00000110 008C 709 1$: .BLKB DIB$K_LENGTH+16
0110 710
0110 711 ;
0110 712 ; Message vectors for $PUTMSG
0110 713 ;
0110 714
00000000 00000000 00000002 0110 715 DECNETERR:: .LONG 2, REMS_NETERR, 0
011C 716
00000000 00000000 00000002 011C 717 NOTTERM: .LONG 2, REMS_NOTERM, 0
0128 718
```

RTPAD
V04-000

- REMOTE TERMINAL PROGRAM
READ WRITE DATA

G 1

16-SEP-1984 02:15:27 VAX/VMS Macro V04-00
5-SEP-1984 03:15:47 [RTPAD.SRC]RTPAD.MAR;1

Page 37
(3)

RT
V04

```
000001B0'00000001 00000000'00000003 0128 719 EXITMSG: .LONG 3,REMS_END,1,NODENAME
                                           0138 720
00000000 00000000'00000002 0138 721 NOTVMS: .LONG 2,REMS_NOPROT,0
                                           0144 722
                                           0144 723 :
                                           0144 724 : Channels assigned in INIT and used by other modules
                                           0144 725 :
                                           0144 726
00000000 0144 727 LINKCHAN:: .LONG 0 ; DECnet link channel
00000000 0148 728
014C 729 MAILCHAN:: .LONG 0 ; DECnet link mailbox channel
014C 730
00000000 014C 731 RDWRTCHAN:: .LONG 0 ; Terminal reads channel
0150 732 READCHAN::
00000000 0150 734 WRITECHAN:: .LONG 0 ; Terminal writes channel
0154 735
00000000 0154 736 CNTRLCHAN:: .LONG 0 ; Terminal ^C and ^Y enables
0158 737
00000000 0158 738 TERMMBXCHAN:: .LONG 0 ; Terminal Unsolicited data mailbox channel
015C 739
```



```
015C 741 :  
015C 742 : Cli interface storage  
015C 743 :  
44 4C 4F 00000164'010E0000' 015C 744 OLD_DESC: .ASCID /OLD/ ; /OLD qualifier  
0167 745  
47 4F 4C 0000016F'010E0000' 0167 746 LOG_DESC: .ASCID /LOG/ ; /LOG qualifier  
0172 747  
45 54 44 0000017A'010E0000' 0172 748 DTE_DESC: .ASCID /DTE/ ; /DTE qualifier  
017D 749  
45 44 4F 4E 00000185'010E0000' 017D 750 NODEDESC: .ASCID /NODE/ ; name of parameter  
0189 751  
0189 752 :  
0189 753 : Network names and descriptors  
0189 754 :  
0189 755  
0189 756 OBJ_DESC: ; Object type descriptor  
00000007' 0189 757 .LONG 20$-10$  
00000191' 018D 758 .LONG 10$  
0191 759  
22 3A 3A 0191 760 10$: .ASCII /:='' ; FINAL FORMAT: node_name::'nn='  
00000003 0194 761 OBJ_C_PREFIX = -10$ ; Offset to object number  
22 3D 32 34 0194 762 .ASCII /42='' ; 42 is TSA, 23 is old remote terminal  
0198 763 20$:  
0198 764  
0198 765 NODE_NAME_DESC: ; Returned CLI node name parameter  
02000000 0198 766 .LONG DSC$K_CLASS_Da<DSC$B_CLASS*8>  
00000000 019C 767 .LONG 0  
01A0 768 LOG_FILE_DESC: ; Returned /LOG= file spec  
02000000 01A0 769 .LONG DSC$K_CLASS_Da<DSC$B_CLASS*8>  
00000000 01A4 770 .LONG 0  
01A8 771  
01A8 772 CONNDESC: ; Network connection string  
02000000 01A8 773 .LONG DSC$K_CLASS_Da<DSC$B_CLASS*8>  
00000000 01AC 774 .LONG 0  
01B0 775  
00000040' 01B0 776 NODENAME:: .LONG 20$-10$ ; Translation of SYS$NODE held here  
000001B8' 01B4 777 .LONG 10$  
01B8 778  
000001F8 01B8 779 10$: .BLKB 64  
01F8 780 20$:  
01F8 781  
01F8 782 FINALPATH:: ; Final path descriptor  
02000000 01F8 783 .LONG DSC$K_CLASS_Da<DSC$B_CLASS*8>  
00000000 01FC 784 .LONG 0  
0200 785  
0200 786 FINALACS:: ; Final ACS descriptor  
02000000 0200 787 .LONG DSC$K_CLASS_Da<DSC$B_CLASS*8>  
00000000 0204 788 .LONG 0  
0208 789  
0208 790 PSTHRU_MSG: ; Saved PSTHRU message descriptor  
02000000 0208 791 .LONG DSC$K_CLASS_Da<DSC$B_CLASS*8>  
00000000 020C 792 .LONG 0  
0210 793  
0A 0D 00000218'010E0000' 0210 794 PSTHRU_CRLF: .ASCID <13><10> ; <CR><LF> for PSTHRU messages  
021A 795  
021A 796 :  
021A 797 ; RMS storage
```

```
021A 798 ;
021A 799 .ALIGN LONG
021C 800
021C 801 SYSINFAB:: $FAB FAC=GET,FNM=<SYSS$INPUT> ; To open SYSS$INPUT
021C $FABDEF
021C $DEFINI FAB,
021C .SAVE LOCAL_BLOCK
021C .NOCROSS
021C .IIF DIF <> <GLOBAL>,.ENABLE SUPPRESSION
021C .PSECT $ABSS,ABS
0000 $GBLINI
0000 .IF IDN <LOCAL> <GLOBAL>
0000 .MACRO $DEF SYM,ALLOC,SIZ
0000 .IIF NB,SYM,SYM::
0000 .IIF NB,ALLOC, ALLOC SIZ
0000 .ENDM $DEF
0000 .MACRO $EQU SYM,VAL
0000 SYM=VAL
0000 .ENDM $EQU
0000 .MACRO $VIELD1 MOD,SEP,SYM,SIZ,MSK
0000 SIZ...=1
0000 .IIF NB,SIZ, SIZ...=SIZ
0000 .IF NB,SYM
0000 MOD'SEP'V'SYM=BIT...
0000 .IIF NB,SIZ, MOD'SEP'S'SYM=SIZ
0000 .IIF NB,MSK, MOD'SEP'M_'SYM=<<<1@SIZ...>-1>@BIT...>
0000 .ENDC
0000 BIT...=BIT...+SIZ...
0000 .ENDM $VIELD1
0000 .IIF
0000 .IIF DIF <LOCAL> <LOCAL>,.ERROR ;ARG MUST BE 'GLOBAL','LOCAL',OR NULL
0000 .MACRO $DEF SYM,ALLOC,SIZ
0000 .IIF NB,SYM,SYM:
0000 .IIF NB,ALLOC, ALLOC SIZ
0000 .ENDM $DEF
0000 .MACRO $EQU SYM,VAL
0000 SYM=VAL
0000 .ENDM $EQU
0000 .MACRO $VIELD1 MOD,SEP,SYM,SIZ,MSK
0000 SIZ...=1
0000 .IIF NB,SIZ, SIZ...=SIZ
0000 .IF NB,SYM
0000 MOD'SEP'V'SYM=BIT...
0000 .IIF NB,SIZ, MOD'SEP'S'SYM=SIZ
0000 .IIF NB,MSK, MOD'SEP'M_'SYM=<<<1@SIZ...>-1>@BIT...>
0000 .ENDC
0000 BIT...=BIT...+SIZ...
0000 .ENDM $VIELD1
0000 .ENDC
00000000 0000 . =0
00000003 0000 $EQU FAB$L_BID 3
0000 FAB$C_BID=3
00000000 0000 $EQU FAB$M_PPF_RAT 16320
00003FC0 0000 FAB$M_PPF_RAT=16320
```

00004000	0000	SEQU	FABSM_PPF_IND	16384
	0000		FABSM_PPF_IND=16384	
	0000			
00000002	0000	SEQU	FABSM_MXV	2
	0000		FABSM_MXV=2	
	0000			
00000004	0000	SEQU	FABSM_SUP	4
	0000		FABSM_SUP=4	
	0000			
00000008	0000	SEQU	FABSM_TMP	8
	0000		FABSM_TMP=8	
	0000			
00000010	0000	SEQU	FABSM_TMD	16
	0000		FABSM_TMD=16	
	0000			
00000020	0000	SEQU	FABSM_DFW	32
	0000		FABSM_DFW=32	
	0000			
00000040	0000	SEQU	FABSM_SQO	64
	0000		FABSM_SQO=64	
	0000			
00000080	0000	SEQU	FABSM_RWO	128
	0000		FABSM_RWO=128	
	0000			
00000100	0000	SEQU	FABSM_POS	256
	0000		FABSM_POS=256	
	0000			
00000200	0000	SEQU	FABSM_WCK	512
	0000		FABSM_WCK=512	
	0000			
00000400	0000	SEQU	FABSM_NEF	1024
	0000		FABSM_NEF=1024	
	0000			
00000800	0000	SEQU	FABSM_RWC	2048
	0000		FABSM_RWC=2048	
	0000			
00001000	0000	SEQU	FABSM_DMO	4096
	0000		FABSM_DMO=4096	
	0000			
00002000	0000	SEQU	FABSM_SPL	8192
	0000		FABSM_SPL=8192	
	0000			
00004000	0000	SEQU	FABSM_SCF	16384
	0000		FABSM_SCF=16384	
	0000			
00008000	0000	SEQU	FABSM_DLT	32768
	0000		FABSM_DLT=32768	
	0000			
00010000	0000	SEQU	FABSM_NFS	65536
	0000		FABSM_NFS=65536	
	0000			
00020000	0000	SEQU	FABSM_UFO	131072
	0000		FABSM_UFO=131072	
	0000			
00040000	0000	SEQU	FABSM_PPF	262144
	0000		FABSM_PPF=262144	

	0000			
	0000			
00080000	0000	\$EQU	FABSM_INP	524288
	0000		FABSM_INP=	524288
	0000			
	0000			
00100000	0000	\$EQU	FABSM_CTG	1048576
	0000		FABSM_CTG=	1048576
	0000			
	0000			
00200000	0000	\$EQU	FABSM_CBT	2097152
	0000		FABSM_CBT=	2097152
	0000			
	0000			
00800000	0000	\$EQU	FABSM_RCK	8388608
	0000		FABSM_RCK=	8388608
	0000			
	0000			
01000000	0000	\$EQU	FABSM_NAM	16777216
	0000		FABSM_NAM=	16777216
	0000			
	0000			
02000000	0000	\$EQU	FABSM_CIF	33554432
	0000		FABSM_CIF=	33554432
	0000			
	0000			
08000000	0000	\$EQU	FABSM_ESC	134217728
	0000		FABSM_ESC=	134217728
	0000			
	0000			
10000000	0000	\$EQU	FABSM_TEF	268435456
	0000		FABSM_TEF=	268435456
	0000			
	0000			
20000000	0000	\$EQU	FABSM_OFP	536870912
	0000		FABSM_OFP=	536870912
	0000			
	0000			
40000000	0000	\$EQU	FABSM_KFO	1073741824
	0000		FABSM_KFO=	1073741824
	0000			
	0000			
00000001	0000	\$EQU	FABSM_PUT	1
	0000		FABSM_PUT=	1
	0000			
	0000			
00000002	0000	\$EQU	FABSM_GET	2
	0000		FABSM_GET=	2
	0000			
	0000			
00000004	0000	\$EQU	FABSM_DEL	4
	0000		FABSM_DEL=	4
	0000			
	0000			
00000008	0000	\$EQU	FABSM_UPD	8
	0000		FABSM_UPD=	8
	0000			
	0000			
00000010	0000	\$EQU	FABSM_TRN	16
	0000		FABSM_TRN=	16
	0000			
	0000			
00000020	0000	\$EQU	FABSM_BIO	32
	0000		FABSM_BIO=	32
	0000			
	0000			
00000040	0000	\$EQU	FABSM_BRO	64
	0000		FABSM_BRO=	64
	0000			
	0000			
00000080	0000	\$EQU	FABSM_EXE	128
	0000		FABSM_EXE=	128
	0000			
	0000			
00000001	0000	\$EQU	FABSM_SHRPUT	1
	0000		FABSM_SHRPUT=	1

	0000			
	0000			
00000002	0000	SEQU	FABSM_SHRGET	2
	0000		FABSM_SHRGET=2	
	0000			
	0000			
00000004	0000	SEQU	FABSM_SHRDEL	4
	0000		FABSM_SHRDEL=4	
	0000			
	0000			
00000008	0000	SEQU	FABSM_SHRUPD	8
	0000		FABSM_SHRUPD=8	
	0000			
	0000			
00000010	0000	SEQU	FABSM_MSE	16
	0000		FABSM_MSE=16	
	0000			
	0000			
00000020	0000	SEQU	FABSM_NIL	32
	0000		FABSM_NIL=32	
	0000			
	0000			
00000040	0000	SEQU	FABSM_UPI	64
	0000		FABSM_UPI=64	
	0000			
	0000			
00000000	0000	SEQU	FABSC_SEQ	0
	0000		FABSC_SEQ=0	
	0000			
	0000			
00000010	0000	SEQU	FABSC_REL	16
	0000		FABSC_REL=16	
	0000			
	0000			
00000020	0000	SEQU	FABSC_IDX	32
	0000		FABSC_IDX=32	
	0000			
	0000			
00000030	0000	SEQU	FABSC_HSH	48
	0000		FABSC_HSH=48	
	0000			
	0000			
00000001	0000	SEQU	FABSM_FTN	1
	0000		FABSM_FTN=1	
	0000			
	0000			
00000002	0000	SEQU	FABSM_CR	2
	0000		FABSM_CR=2	
	0000			
	0000			
00000004	0000	SEQU	FABSM_PRN	4
	0000		FABSM_PRN=4	
	0000			
	0000			
00000008	0000	SEQU	FABSM_BLK	8
	0000		FABSM_BLK=8	
	0000			
	0000			
00000002	0000	SEQU	FABSC_RFM_DFLT	2
	0000		FABSC_RFM_DFLT=2	
	0000			
	0000			
00000000	0000	SEQU	FABSC_UDF	0
	0000		FABSC_UDF=0	
	0000			
	0000			
00000001	0000	SEQU	FABSC_FIX	1
	0000		FABSC_FIX=1	
	0000			
	0000			
00000002	0000	SEQU	FABSC_VAR	2
	0000		FABSC_VAR=2	
	0000			
	0000			
00000003	0000	SEQU	FABSC_VFC	3
	0000		FABSC_VFC=3	

	0000			
	0000			
00000004	0000	\$EQU	FAB\$C_STM	4
	0000		FAB\$C_STM=4	
	0000			
	0000			
00000005	0000	\$EQU	FAB\$C_STMLF	5
	0000		FAB\$C_STMLF=5	
	0000			
	0000			
00000006	0000	\$EQU	FAB\$C_STMCR	6
	0000		FAB\$C_STMCR=6	
	0000			
	0000			
00000006	0000	\$EQU	FAB\$C_MAXRFM	6
	0000		FAB\$C_MAXRFM=6	
	0000			
	0000			
00000001	0000	\$EQU	FAB\$M_RU	1
	0000		FAB\$M_RU=1	
	0000			
	0000			
00000002	0000	\$EQU	FAB\$M_AI	2
	0000		FAB\$M_AI=2	
	0000			
	0000			
00000004	0000	\$EQU	FAB\$M_BI	4
	0000		FAB\$M_BI=4	
	0000			
	0000			
00000050	0000	\$EQU	FAB\$K_BLN	80
	0000		FAB\$K_BLN=80	
	0000			
	0000			
00000050	0000	\$EQU	FAB\$C_BLN	80
	0000		FAB\$C_BLN=80	
	0000			
	0000			
00000050	0000	\$EQU	FAB\$S_FABDEF	80
	0000		FAB\$S_FABDEF=80	
	0000			
	0000			
00000000	0000	\$EQU	FAB\$B_BID	0
	0000		FAB\$B_BID=0	
	0000			
	0000			
00000001	0000	\$EQU	FAB\$B_BLN	1
	0000		FAB\$B_BLN=1	
	0000			
	0000			
00000002	0000	\$EQU	FAB\$R_IFI_OVERLAY	2
	0000		FAB\$R_IFI_OVERLAY=2	
	0000			
	0000			
00000002	0000	\$EQU	FAB\$W_IFI	2
	0000		FAB\$W_IFI=2	
	0000			
	0000			
00000002	0000	\$EQU	FAB\$R_IFI_BITS	2
	0000		FAB\$R_IFI_BITS=2	
	0000			
	0000			
00000008	0000	\$EQU	FAB\$S_PPF_RAT	8
	0000		FAB\$S_PPF_RAT=8	
	0000			
	0000			
00000006	0000	\$EQU	FAB\$V_PPF_RAT	6
	0000		FAB\$V_PPF_RAT=6	
	0000			
	0000			
0000000E	0000	\$EQU	FAB\$V_PPF_IND	14
	0000		FAB\$V_PPF_IND=14	
	0000			
	0000			
00000004	0000	\$EQU	FAB\$R_FOP_OVERLAY	4
	0000		FAB\$R_FOP_OVERLAY=4	

00000004	0000	SEQU	FABSL_FOP	4
	0000		FABSL_FOP=4	
	0000			
00000004	0000	SEQU	FABSR_FOP_BITS	4
	0000		FABSR_FOP_BITS=4	
	0000			
00000001	0000	SEQU	FABSV_MXV	1
	0000		FABSV_MXV=1	
	0000			
00000002	0000	SEQU	FABSV_SUP	2
	0000		FABSV_SUP=2	
	0000			
00000003	0000	SEQU	FABSV_TMP	3
	0000		FABSV_TMP=3	
	0000			
00000004	0000	SEQU	FABSV_TMD	4
	0000		FABSV_TMD=4	
	0000			
00000005	0000	SEQU	FABSV_DFW	5
	0000		FABSV_DFW=5	
	0000			
00000006	0000	SEQU	FABSV_SQO	6
	0000		FABSV_SQO=6	
	0000			
00000007	0000	SEQU	FABSV_RWO	7
	0000		FABSV_RWO=7	
	0000			
00000008	0000	SEQU	FABSV_POS	8
	0000		FABSV_POS=8	
	0000			
00000009	0000	SEQU	FABSV_WCK	9
	0000		FABSV_WCK=9	
	0000			
0000000A	0000	SEQU	FABSV_NEF	10
	0000		FABSV_NEF=10	
	0000			
0000000B	0000	SEQU	FABSV_RWC	11
	0000		FABSV_RWC=11	
	0000			
0000000C	0000	SEQU	FABSV_DMO	12
	0000		FABSV_DMO=12	
	0000			
0000000D	0000	SEQU	FABSV_SPL	13
	0000		FABSV_SPL=13	
	0000			
0000000E	0000	SEQU	FABSV_SCF	14
	0000		FABSV_SCF=14	
	0000			
0000000F	0000	SEQU	FABSV_DLT	15
	0000		FABSV_DLT=15	
	0000			
00000010	0000	SEQU	FABSV_NFS	16
	0000		FABSV_NFS=16	
	0000			
00000011	0000	SEQU	FABSV_UFO	17
	0000		FABSV_UFO=17	

00000012	0000	SEQU	FABSV_PPF	18
	0000		FABSV_PPF=18	
	0000			
00000013	0000	SEQU	FABSV_INP	19
	0000		FABSV_INP=19	
	0000			
00000014	000C	SEQU	FABSV_CTG	20
	0000		FABSV_CTG=20	
	0000			
00000015	0000	SEQU	FABSV_CBT	21
	0000		FABSV_CBT=21	
	0000			
00000017	0000	SEQU	FABSV_RCK	23
	0000		FABSV_RCK=23	
	0000			
00000018	0000	SEQU	FABSV_NAM	24
	0000		FABSV_NAM=24	
	0000			
00000019	000C	SEQU	FABSV_CIF	25
	0000		FABSV_CIF=25	
	0000			
0000001B	0000	SEQU	FABSV_ESC	27
	0000		FABSV_ESC=27	
	0000			
0000001C	0000	SEQU	FABSV_TEF	28
	0000		FABSV_TEF=28	
	0000			
0000001D	0000	SEQU	FABSV_OFP	29
	0000		FABSV_OFP=29	
	0000			
0000001E	0000	SEQU	FABSV_KFO	30
	0000		FABSV_KFO=30	
	0000			
00000008	0000	SEQU	FABSL_STS	8
	0000		FABSL_STS=8	
	0000			
0000000C	0000	SEQU	FABSL_STV	12
	0000		FABSL_STV=12	
	0000			
00000010	0000	SEQU	FABSL_ALQ	16
	0000		FABSL_ALQ=16	
	0000			
00000014	0000	SEQU	FABSW_DEQ	20
	0000		FABSW_DEQ=20	
	0000			
00000016	0000	SEQU	FABSR_FAC_OVERLAY	22
	0000		FABSR_FAC_OVERLAY=22	
	0000			
00000016	0000	SEQU	FABSB_FAC	22
	0000		FABSB_FAC=22	
	0000			
00000016	0000	SEQU	FABSR_FAC_BITS	22
	0000		FABSR_FAC_BITS=22	
	0000			
00000000	0000	SEQU	FABSV_PUT	0
	0000		FABSV_PUT=0	

00000001	0000	SEQU	FABSV_GET	1
	0000		FABSV_GET=1	
00000002	0000	SEQU	FABSV_DEL	2
	0000		FABSV_DEL=2	
00000003	0000	SEQU	FABSV_UPD	3
	0000		FABSV_UPD=3	
00000004	0000	SEQU	FABSV_TRN	4
	0000		FABSV_TRN=4	
00000005	0000	SEQU	FABSV_BIO	5
	0000		FABSV_BIO=5	
00000006	0000	SEQU	FABSV_BRO	6
	0000		FABSV_BRO=6	
00000007	0000	SEQU	FABSV_EXE	7
	0000		FABSV_EXE=7	
00000017	0000	SEQU	FABSR_SHR_OVERLAY	23
	0000		FABSR_SHR_OVERLAY=23	
00000017	0000	SEQU	FABSB_SHR	23
	0000		FABSB_SHR=23	
00000017	0000	SEQU	FABSR_SHR_BITS	23
	0000		FABSR_SHR_BITS=23	
00000000	0000	SEQU	FABSV_SHRPUT	0
	0000		FABSV_SHRPUT=0	
00000001	0000	SEQU	FABSV_SHRGET	1
	0000		FABSV_SHRGET=1	
00000002	0000	SEQU	FABSV_SHRDEL	2
	0000		FABSV_SHRDEL=2	
00000003	0000	SEQU	FABSV_SHRUPD	3
	0000		FABSV_SHRUPD=3	
00000004	0000	SEQU	FABSV_MSE	4
	0000		FABSV_MSE=4	
00000005	0000	SEQU	FABSV_NIL	5
	0000		FABSV_NIL=5	
00000006	0000	SEQU	FABSV_UPI	6
	0000		FABSV_UPI=6	
00000018	0000	SEQU	FABSL_CTX	24
	0000		FABSL_CTX=24	
0000001C	0000	SEQU	FABSB_RTV	28
	0000		FABSB_RTV=28	

```
0000
0000001D 0000 $EQU FAB$R_ORG_OVERLAY 29
0000001D 0000 FAB$R_ORG_OVERLAY=29
0000001D 0000 $EQU FAB$B_ORG 29
0000001D 0000 FAB$B_ORG=29
0000001D 0000 $EQU FAB$R_ORG_BITS 29
0000001D 0000 FAB$R_ORG_BITS=29
00000004 0000 $EQU FAB$S_ORG 4
00000004 0000 FAB$S_ORG=4
00000004 0000 $EQU FAB$V_ORG 4
00000004 0000 FAB$V_ORG=4
0000001E 0000 $EQU FAB$R_RAT_OVERLAY 30
0000001E 0000 FAB$R_RAT_OVERLAY=30
0000001E 0000 $EQU FAB$B_RAT 30
0000001E 0000 FAB$B_RAT=30
0000001E 0000 $EQU FAB$R_RAT_BITS 30
0000001E 0000 FAB$R_RAT_BITS=30
00000000 0000 $EQU FAB$V_FTN 0
00000000 0000 FAB$V_FTN=0
00000001 0000 $EQU FAB$V_CR 1
00000001 0000 FAB$V_CR=1
00000002 0000 $EQU FAB$V_PRN 2
00000002 0000 FAB$V_PRN=2
00000003 0000 $EQU FAB$V_BLK 3
00000003 0000 FAB$V_BLK=3
0000001F 0000 $EQU FAB$B_RFM 31
0000001F 0000 FAB$B_RFM=31
00000020 0000 $EQU FAB$L_JNL 32
00000020 0000 FAB$L_JNL=32
00000024 0000 $EQU FAB$L_XAB 36
00000024 0000 FAB$L_XAB=36
00000028 0000 $EQU FAB$L_NAM 40
00000028 0000 FAB$L_NAM=40
0000002C 0000 $EQU FAB$L_FNA 44
0000002C 0000 FAB$L_FNA=44
00000030 0000 $EQU FAB$L_DNA 48
00000030 0000 FAB$L_DNA=48
00000034 0000 $EQU FAB$B_FNS 52
00000034 0000 FAB$B_FNS=52
```

	0000			
00000035	0000	SEQU	FAB\$B_DNS	53
	0000		FAB\$B_DNS=53	
	0000			
00000036	0000	SEQU	FAB\$W_MRS	54
	0000		FAB\$W_MRS=54	
	0000			
00000038	0000	SEQU	FAB\$L_MRN	56
	0000		FAB\$L_MRN=56	
	0000			
0000003C	0000	SEQU	FAB\$W_BLS	60
	0000		FAB\$W_BLS=60	
	0000			
0000003E	0000	SEQU	FAB\$B_BKS	62
	0000		FAB\$B_BKS=62	
	0000			
0000003F	0000	SEQU	FAB\$B_FSZ	63
	0000		FAB\$B_FSZ=63	
	0000			
00000040	0000	SEQU	FAB\$L_DEV	64
	0000		FAB\$L_DEV=64	
	0000			
00000044	0000	SEQU	FAB\$L_SDC	68
	0000		FAB\$L_SDC=68	
	0000			
00000048	0000	SEQU	FAB\$W_GBC	72
	0000		FAB\$W_GBC=72	
	0000			
0000004A	0000	SEQU	FAB\$R_ACMODES_OVERLAY	74
	0000		FAB\$R_ACMODES_OVERLAY=74	
	0000			
0000004A	0000	SEQU	FAB\$B_ACMODES	74
	0000		FAB\$B_ACMODES=74	
	0000			
0000004A	0000	SEQU	FAB\$R_ACMODES_BITS	74
	0000		FAB\$R_ACMODES_BITS=74	
	0000			
00000002	0000	SEQU	FAB\$S_LNM_MODE	2
	0000		FAB\$S_LNM_MODE=2	
	0000			
00000000	0000	SEQU	FAB\$V_LNM_MODE	0
	0000		FAB\$V_LNM_MODE=0	
	0000			
00000002	0000	SEQU	FAB\$S_CHAN_MODE	2
	0000		FAB\$S_CHAN_MODE=2	
	0000			
00000002	0000	SEQU	FAB\$V_CHAN_MODE	2
	0000		FAB\$V_CHAN_MODE=2	
	0000			
00000002	0000	SEQU	FAB\$S_FILE_MODE	2
	0000		FAB\$S_FILE_MODE=2	
	0000			
00000004	0000	SEQU	FAB\$V_FILE_MODE	4
	0000		FAB\$V_FILE_MODE=4	
	0000			
0000004B	0000	SEQU	FAB\$R_RCF_OVERLAY	75
	0000		FAB\$R_RCF_OVERLAY=75	

```
00000000 0000
0000004B 0000      SEQU  FAB$B_RCF      75
0000004B 0000      FAB$B_RCF=75
0000004B 0000      SEQU  FAB$R_RCF_BITS 75
0000004B 0000      FAB$R_RCF_BITS=75
00000000 0000      SEQU  FAB$V_RU      0
00000000 0000      FAB$V_RU=0
00000001 0000      SEQU  FAB$V_AI      1
00000001 0000      FAB$V_AI=1
00000002 0000      SEQU  FAB$V_BI      2
00000002 0000      FAB$V_BI=2
0000021C 0000      $DEFEND FAB,,DEF
0000021C 0000      .MACRO  $FABDEF A
0000021C 0000      .ENDM   $FABDEF
0000021C 0000      .IIF    DIF <> <GLOBAL>,,DISABLE      SUPPRESSION
0000021C 0000      .CROSS
0000021C 0000      .RESTORE
0000021C 021C      $$R_TABINIT      FAB$C_BID, FAB$C_BLN
0000021C 021C      .IIF NE .83, .print ;%MACRO-I=GENINFO, Generated INFO: RMS BLOCK NOT LONGWORD ALIGNED
0000021C 021C      $$TAB=.
0000021C 021C      .BYTE FAB$C_BID
0000021C 021D      .BYTE FAB$C_BLN
0000026C 021E      .BLKB FAB$C_BLN-2
0000026C 026C      $$TABEND=.
00000000 026C      $$R_VBFSET      FAB,<>
00000000 026C      $$TMP=0
00000000 026C      .IRP X,<>
00000000 026C      .IF DF FAB$V_'X'
00000000 026C      $$TMP=$$TMP!<1@FAB$V_'X'>
00000000 026C      .IFF
00000000 026C      .ERROR      ; UNDEFINED BIT VALUE CODE: X;
00000000 026C      .ENDC
00000000 026C      .ENDR
00000220 026C      .=$$TAB+FAB$S_FOP
00000000 0220      .ADDRESS      $$TMP
0000022C 0224      .=$$TAB+FAB$S_ALQ
00000000 022C      .ADDRESS      0
00000000 0230      .WORD      0
00000000 0232      $$R_VBFSET      FAB,<GET>
00000000 0232      $$TMP=0
00000000 0232      .IRP X,<GET>
00000000 0232      .IF DF FAB$V_'X'
00000000 0232      $$TMP=$$TMP!<1@FAB$V_'X'>
00000000 0232      .IFF
00000000 0232      .ERROR      ; UNDEFINED BIT VALUE CODE: X;
00000000 0232      .ENDC
00000000 0232      .ENDR
00000000 0232      .IF DF FAB$V_GET
```

```
00000002 0232          $$TMP=$$TMP!<1@FAB$V_GET^
0232          .IFF
0232          .ERROR          ; UNDEFINED BIT VALUE CODE: GET;
0232          .ENDC
0232
02 0232          .BYTE  $$TMP
0232          $$R_VBFSET      FAB,<>
00000000 0233          $$TMP=0
0233          .IRP X,<>
0233          .IF DF FAB$V 'X
0233          $$TMP=$$TMP!<1@FAB$V_'X>
0233          .IFF
0233          .ERROR          ; UNDEFINED BIT VALUE CODE: X;
0233          .ENDC
0233          .ENDR
00 0233          .BYTE  $$TMP
00000000' 0234          .ADDRESS 0
00 0238          .BYTE  0
0239          .IF DF FAB$C_SEQ
00 0239          .BYTE  FAB$C_SEQ
023A          .IFF
023A          .BYTE
023A          .ERROR          ; UNDEFINED VALUE FOR FIELD : SEQ;
023A          .ENDC
023A          $$R_VBFSET      FAB,<>
00000000 023A          $$TMP=0
023A          .IRP X,<>
023A          .IF DF FAB$V 'X
023A          $$TMP=$$TMP!<1@FAB$V_'X>
023A          .IFF
023A          .ERROR          ; UNDEFINED BIT VALUE CODE: X;
023A          .ENDC
023A          .ENDR
00 023A          .BYTE  $$TMP
02 023B          .IF DF FAB$C_VAR
023B          .BYTE  FAB$C_VAR
023C          .IFF
023C          .BYTE
023C          .ERROR          ; UNDEFINED VALUE FOR FIELD : VAR;
023C          .ENDC
00000000' 023C          .ADDRESS 0
00000000' 0240          .ADDRESS 0
00000000' 0244          .ADDRESS 0
00000000' 0248          .ADDRESS 0
00000000' 024C          .ADDRESS 0
00 0250          .BYTE  0
00 0251          .BYTE  0
0000 0252          .WORD  0
00000000' 0254          .ADDRESS 0
0000 0258          .WORD  0
00 025A          .BYTE  0
00 025B          .BYTE  0
00000264 025C          =$$TAB+FAB$W_GBC
0000 0264          .WORD  0
```

```
00 0266 .BYTE <<0@FAB$V_LNM_MODE> + <0@FAB$V_CHAN_MODE> + -
0267 <0@FAB$V_FILE_MODE>>
0267 .IIF NE 8-8 .ERROR ; INVALID BYTE SIZE
0267 .IF NB <SYS$INPUT>
0267 .SAVE
00000000 .PSECT $RMSNAM
00000000 0000 $$TMPX=.
54 55 50 4E 49 24 53 59 53 0000 .ASCII %SYS$INPUT%
00000009 0009 $$TMPX1=-$$TMPX
00000267 .RESTORE
00000248 0267 .=$$TAB+FAB$L_FNA
00000000' 0248 .ADDRESS $$TMPX
00000250 024C .=$$TAB+FAB$B_FNS
09 0250 .BYTE $$TMPX1
0251 .ENDC
0251 .IF NB <>
0251 .SAVE
0251 .PSECT $RMSNAM
0251 $$TMPX=.
0251 .ASCII %
0251 $$TMPX1=-$$TMPX
0251 .RESTORE
0251 .=$$TAB+FAB$L_DNA
0251 .ADDRESS $$TMPX
0251 .=$$TAB+FAB$B_DNS
0251 .BYTE $$TMPX1
0251 .ENDC
0000026C 0251 .=$$TABEND
026C
026C
026C 802 SYSINRAB:: $RAB FAB=SYSINFAB
026C 803 $RABDEF
026C $DEFINI RAB,
026C .SAVE LOCAL_BLOCK
026C .NOCROSS
026C .IIF DIF <> <GLOBAL>,.ENABLE SUPPRESSION
026C .PSECT $ABSS,ABS
0000 $GBLINI
0000 .IF IDN <LOCAL> <GLOBAL>
0000 .MACRO $DEF SYM,ALLOC,SIZ
0000 .IIF NB,SYM,SYM::
0000 .IIF NB,ALLOC, ALLOC SIZ
0000 .ENDM $DEF
0000 .MACRO $EQU SYM,VAL
0000 SYM==VAL
0000 .ENDM $EQU
0000 .MACRO $VIELD1 MOD,SEP,SYM,SIZ,MSK
0000 SIZ...=1
0000 .IIF NB,SIZ, SIZ...=SIZ
0000 .IF NB,SYM
0000 MOD'SEP'V 'SYM==BIT...
0000 .IIF NB,SIZ, MOD'SEP'S 'SYM==SIZ
0000 .IIF NB,MSK, MOD'SEP'M 'SYM==<<<1@SIZ...>-1>@BIT...>
0000 .ENDC
0000 BIT...=BIT...+SIZ...
0000 .ENDM $VIELD1
0000 .IIF
```



```
0000 .IIF DIF <LOCAL> <LOCAL> .ERROR ;ARG MUST BE 'GLOBAL','LOCAL',OR NULL
0000 .MACRO SDEF SYM,ALLOC,SIZ
0000 .IIF NB,SYM,SYM:
0000 .IIF NB,ALLOC, ALLOC SIZ
0000 .ENDM SDEF
0000 .MACRO SEQU SYM,VAL
0000 SYM=VAL
0000 .ENDM SEQU
0000 .MACRO $VIELD1 MOD,SEP,SYM,SIZ,MSK
0000 SIZ...=1
0000 .IIF NB,SIZ, SIZ...=SIZ
0000 .IF NB,SYM
0000 MOD'SEP'V'SYM=BIT...
0000 .IIF NB,SIZ, MOD'SEP'S'SYM=SIZ
0000 .IIF NB,MSK, MOD'SEP'M'SYM=<<<1@SIZ...>-1>@BIT...>
0000 .ENDC
0000 BIT...=BIT...+SIZ...
0000 .ENDM $VIELD1
0000 .ENDC
00000000 0000 .=0
00000001 0000 SEQU RAB$C_BID 1
0000 RAB$C_BID=1
00003FC0 0000 SE RAB$M_PPF_RAT 16320
0000 RAB$M_PPF_RAT=16320
00004000 0000 SEQU RAB$M_PPF_IND 16384
0000 RAB$M_PPF_IND=16384
00000001 0000 SEQU RAB$M_ASY 1
0000 RAB$M_ASY=1
00000002 0000 SEQU RAB$M_TPT 2
0000 RAB$M_TPT=2
00000004 0000 SEQU RAB$M_REA 4
0000 RAB$M_REA=4
00000008 0000 SEQU RAB$M_RRL 8
0000 RAB$M_RRL=8
00000010 0000 SEQU RAB$M_UIF 16
0000 RAB$M_UIF=16
00000020 0000 SEQU RAB$M_MAS 32
0000 RAB$M_MAS=32
00000040 0000 SEQU RAB$M_FDL 64
0000 RAB$M_FDL=64
00000080 0000 SEQU RAB$M_HSH 128
0000 RAB$M_HSH=128
00000100 0000 SEQU RAB$M_EOF 256
0000 RAB$M_EOF=256
```

	0000			
	0000	SEQU	RABSM_RAH	512
00000200	0000		RABSM_RAH=512	
	0000			
	0000	SEQU	RABSM_WBH	1024
00000400	0000		RABSM_WBH=1024	
	0000			
	0000	SEQU	RABSM_BIO	2048
00000800	0000		RABSM_BIO=2048	
	0000			
	0000	SEQU	RABSM_LV2	4096
00001000	0000		RABSM_LV2=4096	
	0000			
	0000	SEQU	RABSM_LOA	8192
00002000	0000		RABSM_LOA=8192	
	0000			
	0000	SEQU	RABSM_LIM	16384
00004000	0000		RABSM_LIM=16384	
	0000			
	0000	SEQU	RABSM_LOC	65536
00010000	0000		RABSM_LOC=65536	
	0000			
	0000	SEQU	RABSM_WAT	131072
00020000	0000		RABSM_WAT=131072	
	0000			
	0000	SEQU	RABSM_ULK	262144
00040000	0000		RABSM_ULK=262144	
	0000			
	0000	SEQU	RABSM_RLK	524288
00080000	0000		RABSM_RLK=524288	
	0000			
	0000	SEQU	RABSM_NLK	1048576
00100000	0000		RABSM_NLK=1048576	
	0000			
	0000	SEQU	RABSM_KGE	2097152
00200000	0000		RABSM_KGE=2097152	
	0000			
	0000	SEQU	RABSM_KGT	4194304
00400000	0000		RABSM_KGT=4194304	
	0000			
	0000	SEQU	RABSM_NXR	8388608
00800000	0000		RABSM_NXR=8388608	
	0000			
	0000	SEQU	RABSM_RNE	16777216
01000000	0000		RABSM_RNE=16777216	
	0000			
	0000	SEQU	RABSM_TMO	33554432
02000000	0000		RABSM_TMO=33554432	
	0000			
	0000	SEQU	RABSM_CVT	67108864
04000000	0000		RABSM_CVT=67108864	
	0000			
	0000	SEQU	RABSM_RNF	134217728
08000000	0000		RABSM_RNF=134217728	
	0000			
	0000	SEQU	RABSM_ETO	268435456
10000000	0000		RABSM_ETO=268435456	


```
0000
20000000 0000      $EQU  RAB$M_PTA      536870912
          0000      RAB$M_PTA=536870912
          0000
          0000      $EQU  RAB$M_PMT      1073741824
40000000 0000      RAB$M_PMT=1073741824
          0000
          0000      $EQU  RAB$M_CCO      -2147483648
80000000 0000      RAB$M_CCO=-2147483648
          0000
          0000      $EQU  RAB$C_SEQ      0
00000000 0000      RAB$C_SEQ=0
          0000
          0000      $EQU  RAB$C_KEY      1
00000001 0000      RAB$C_KEY=1
          0000
          0000      $EQU  RAB$C_RFA      2
00000002 0000      RAB$C_RFA=2
          0000
          0000      $EQU  RAB$C_STM      3
00000003 0000      RAB$C_STM=3
          0000
          0000      $EQU  RAB$K_BLN      68
00000044 0000      RAB$K_BLN=68
          0000
          0000      $EQU  RAB$C_BLN      68
00000044 0000      RAB$C_BLN=68
          0000
          0000      $EQU  RAB$S_RABDEF    68
00000044 0000      RAB$S_RABDEF=68
          0000
          0000      $EQU  RAB$B_BID      0
00000000 0000      RAB$B_BID=0
          0000
          0000      $EQU  RAB$B_BLN      1
00000001 0000      RAB$B_BLN=1
          0000
          0000      $EQU  RAB$R_ISI_OVERLAY  2
00000002 0000      RAB$R_ISI_OVERLAY=2
          0000
          0000      $EQU  RAB$W_ISI      2
00000002 0000      RAB$W_ISI=2
          0000
          0000      $EQU  RAB$R_ISI_BITS  2
00000002 0000      RAB$R_ISI_BITS=2
          0000
          0000      $EQU  RAB$S_PPF_RAT    8
00000008 0000      RAB$S_PPF_RAT=8
          0000
          0000      $EQU  RAB$V_PPF_RAT    6
00000006 0000      RAB$V_PPF_RAT=6
          0000
          0000      $EQU  RAB$V_PPF_IND    14
0000000E 0000      RAB$V_PPF_IND=14
          0000
          0000      $EQU  RAB$R_ROP_OVERLAY  4
00000004 0000      RAB$R_ROP_OVERLAY=4
```

	0000			
	0000			
00000004	0000	SEQU	RABSL_ROP	4
	0000		RABSL_ROP=4	
	0000			
	0000			
00000004	0000	SEQU	RABSR_ROP_BITSO	4
	0000		RABSR_ROP_BITSO=4	
	0000			
	0000			
00000000	0000	SEQU	RABSV_ASY	0
	0000		RABSV_ASY=0	
	0000			
	0000			
00000001	0000	SEQU	RABSV_TPT	1
	0000		RABSV_TPT=1	
	0000			
	0000			
00000002	0000	SEQU	RABSV_REA	2
	0000		RABSV_REA=2	
	0000			
	0000			
00000003	0000	SEQU	RABSV_RRL	3
	0000		RABSV_RRL=3	
	0000			
	0000			
00000004	0000	SEQU	RABSV_UIF	4
	0000		RABSV_UIF=4	
	0000			
	0000			
00000005	0000	SEQU	RABSV_MAS	5
	0000		RABSV_MAS=5	
	0000			
	0000			
00000006	0000	SEQU	RABSV_FDL	6
	0000		RABSV_FDL=6	
	0000			
	0000			
00000007	0000	SEQU	RABSV_HSH	7
	0000		RABSV_HSH=7	
	0000			
	0000			
00000008	0000	SEQU	RABSV_EOF	8
	0000		RABSV_EOF=8	
	0000			
	0000			
00000009	0000	SEQU	RABSV_RAH	9
	0000		RABSV_RAH=9	
	0000			
	0000			
0000000A	0000	SEQU	RABSV_WBH	10
	0000		RABSV_WBH=10	
	0000			
	0000			
0000000B	0000	SEQU	RABSV_BIO	11
	0000		RABSV_BIO=11	
	0000			
	0000			
0000000C	0000	SEQU	RABSV_LV2	12
	0000		RABSV_LV2=12	
	0000			
	0000			
0000000D	0000	SEQU	RABSV_LOA	13
	0000		RABSV_LOA=13	
	0000			
	0000			
0000000E	0000	SEQU	RABSV_LIM	14
	0000		RABSV_LIM=14	
	0000			
	0000			
00000010	0000	SEQU	RABSV_LOC	16
	0000		RABSV_LOC=16	
	0000			
	0000			
00000011	0000	SEQU	RABSV_WAT	17
	0000		RABSV_WAT=17	

00000012	0000	SEQU	RABSV_ULK	18
	0000		RABSV_ULK=18	
	0000			
00000013	0000	SEQU	RABSV_RLK	19
	0000		RABSV_RLK=19	
	0000			
00000014	0000	SEQU	RABSV_NLK	20
	0000		RABSV_NLK=20	
	0000			
00000015	0000	SEQU	RABSV_KGE	21
	0000		RABSV_KGE=21	
	0000			
00000016	0000	SEQU	RABSV_KGT	22
	0000		RABSV_KGT=22	
	0000			
00000017	0000	SEQU	RABSV_NXR	23
	0000		RABSV_NXR=23	
	0000			
00000018	0000	SEQU	RABSV_RNE	24
	0000		RABSV_RNE=24	
	0000			
00000019	0000	SEQU	RABSV_TMO	25
	0000		RABSV_TMO=25	
	0000			
0000001A	0000	SEQU	RABSV_CVT	26
	0000		RABSV_CVT=26	
	0000			
0000001B	0000	SEQU	RABSV_RNF	27
	0000		RABSV_RNF=27	
	0000			
0000001C	0000	SEQU	RABSV_ETO	28
	0000		RABSV_ETO=28	
	0000			
0000001D	0000	SEQU	RABSV_PTA	29
	0000		RABSV_PTA=29	
	0000			
0000001E	0000	SEQU	RABSV_PMT	30
	0000		RABSV_PMT=30	
	0000			
0000001F	0000	SEQU	RABSV_CCO	31
	0000		RABSV_CCO=31	
	0000			
00000004	0000	SEQU	RABSR_ROP_FIELDS	4
	0000		RABSR_ROP_FIELDS=4	
	0000			
00000005	0000	SEQU	RABSB_ROP1	5
	0000		RABSB_ROP1=5	
	0000			
00000006	0000	SEQU	RABSB_ROP2	6
	0000		RABSB_ROP2=6	
	0000			
00000007	0000	SEQU	RABSB_ROP3	7
	0000		RABSB_ROP3=7	
	0000			
00000008	0000	SEQU	RABSL_STS	8
	0000		RABSL_STS=8	

0000000C	0000	SEQU	RAB\$R_STV_OVERLAY	12
	0000		RAB\$R_STV_OVERLAY=12	
	0000			
0000000C	0000	SEQU	RAB\$L_STV	12
	0000		RAB\$L_STV=12	
	0000			
0000000C	0000	SEQU	RAB\$R_STV_FIELDS	12
	0000		RAB\$R_STV_FIELDS=12	
	0000			
0000000C	0000	SEQU	RAB\$W_STV0	12
	0000		RAB\$W_STV0=12	
	0000			
0000000E	0000	SEQU	RAB\$W_STV2	14
	0000		RAB\$W_STV2=14	
	0000			
00000010	0000	SEQU	RAB\$R_RFA_OVERLAY	16
	0000		RAB\$R_RFA_OVERLAY=16	
	0000			
00000006	0000	SEQU	RAB\$S_RFA	6
	0000		RAB\$S_RFA=6	
	0000			
00000010	0000	SEQU	RAB\$W_RFA	16
	0000		RAB\$W_RFA=16	
	0000			
00000010	0000	SEQU	RAB\$R_RFA_FIELDS	16
	0000		RAB\$R_RFA_FIELDS=16	
	0000			
00000010	0000	SEQU	RAB\$L_RFA0	16
	0000		RAB\$L_RFA0=16	
	0000			
00000014	0000	SEQU	RAB\$W_RFA4	20
	0000		RAB\$W_RFA4=20	
	0000			
00000018	0000	SEQU	RAB\$L_CTX	24
	0000		RAB\$L_CTX=24	
	0000			
0000001E	0000	SEQU	RAB\$B_RAC	30
	0000		RAB\$B_RAC=30	
	0000			
0000001F	0000	SEQU	RAB\$B_TMO	31
	0000		RAB\$B_TMO=31	
	0000			
00000020	0000	SEQU	RAB\$W_USZ	32
	0000		RAB\$W_USZ=32	
	0000			
00000022	0000	SEQU	RAB\$W_RSZ	34
	0000		RAB\$W_RSZ=34	
	0000			
00000024	0000	SEQU	RAB\$L_UBF	36
	0000		RAB\$L_UBF=36	
	0000			
00000028	0000	SEQU	RAB\$L_RBF	40
	0000		RAB\$L_RBF=40	
	0000			
0000002C	0000	SEQU	RAB\$L_RHB	44
	0000		RAB\$L_RHB=44	

RTPAD
V04-000

- REMOTE TERMINAL PROGRAM
READ WRITE DATA

83

```
16-SEP-1984 02:15:27 VAX/VMS Macro V04-00
5-SEP-1984 03:15:47 [RTPAD.SRC]RTPAD.MAR:1
```

Page 58
(3)

RT
VO

```
000002B0 02B0    $$ .TABEND=.
02B0
02B0
00000000 02B0    $$ .TMP=0
02B0    .SSR_VBFSET RAB,<>
02B0    .IRP X,<>
02B0    .IF DF RAB$V 'X
02B0        $$ .TMP=$$ .TMP!<1@RAB$V_'X>
02B0    .IFF
02B0        .ERROR          ; UNDEFINED BIT VALUE CODE: X;
02B0    .ENDC
02B0    .ENDR
02B0
00000270 02B0    .=$$ .TAB+RAB$SL_ROP
00000000 0270    .ADDRESS    $$ .TMP
00000284 0274    .=$$ .TAB+RAB$SL_CTX
00000000 0284    .ADDRESS    0
0000028A 0288    .=$$ .TAB+RAB$B_RAC
028A    .IF DF RAB$C_SEQ
00 028A    .BYTE    RAB$C_SEQ
0288    .IFF
0288    .BYTE
0288    .ERROR          ; UNDEFINED VALUE FOR FIELD: CNST;
0288    .ENDC
00 0288    .BYTE    0
0000 028C    .WORD    0
0000 028E    .WORD    0
00000000 0290    .ADDRESS    0
00000000 0294    .ADDRESS    0
00000000 0298    .ADDRESS    0
00000000 029C    .ADDRESS    0
02A0    .IF NB <>
02A0    .=$$ .TAB+RAB$SL_PBF
02A0    .ADDRESS
02A0    .ENDC
00 02A0    .BYTE    0
02A1    .IF NB <>
02A1    .=$$ .TAB+RAB$B_PSZ
02A1    .BYTE
02A1    .ENDC
00 02A1    .BYTE    0
00 02A2    .BYTE    0
00 02A3    .BYTE    0
00000000 02A4    .ADDRESS    0
0000021C 02A8    .ADDRESS    SYSINFAB
00000000 02AC    .ADDRESS    0
000002B0 02B0    .=$$ .TABEND
02B0
02B0
```



```
0280 806
0280 807 ;
0280 808 ; Flags
0280 809 ;
0280 810
00 0280 811 INDFLAG:: .BYTE 0 ; Indicate indirect command file
0281 812
00 0281 813 WAKEFLAG:: .BYTE 0 ; Flag for legitimate $WAKE
0282 814
00000000 0282 815 CTERM_FLAG:: .LONG 0 ; TSA/CTERM flags (see $RTPADDEF)
0286 816
00000000 0286 817 RTLOG_FLAGS:: .LONG 0 ; value of RTPAD$LOG
028A 818 ;
028A 819 ; other misc. global storage
028A 820 ;
028A 821
028A 822 RTLOG_DESC: ; RTPAD$LOG value
00000010 028A 823 .LONG 16
000002C2 028E 824 .LONG RTLOG_BUF
000002D2 02C2 825 RTLOG_BUF: .BLKB 16
02D2 826
02D2 827 RTPAD_LOGNAM:
24 44 41 50 54 52 000002DA 010E0000 02D2 828 .ASCII /RTPAD$LOG/
47 4F 4C 02E0
02E3 829
00000309 02E3 830 RT$AB_ASTBLK:: .BLKB AST$T_BUF ; Dummy AST block
0309 831
00000000 0309 832 RETSTATUS:: .LONG 0 ; Save a system service status
00000000 030D 833 QUIT_PC:: .LONG 0 ; Save PC where error happened
0311 834
0311 835 FIRSTCMD:: ; First command descriptor
02000000 0311 836 .LONG DSC$K_CLASS_DA<DSC$B_CLASS*8>
00000000 0315 837 .LONG 0
0319 838
00 0319 839 PROTO_ECO:: .BYTE 0 ; protocol eco level
031A 840
0000 031A 841 HOST_OPSYS: .WORD 0 ; host system
031C 842 ;
031C 843 ; local storage
031C 844 ;
031C 845 ;
031C 846
00000000 031C 847 OLDCTRL: .LONG 0 ; CLI out of band enable flags
0320 848
00000000 0320 849 OLDSETRWM: .LONG 0 ; Original resource wait mode
0324 850
00000764 0324 851 FIRSTMSG: .BLKB AST$T_BUF+MAXMSG ; Buffer for BIND message
0764 852
0000041A 0764 853 MAXMSGsiz: .LONG MAXMSG ; Maximum message size
0768 854
```

RTPAD
V04-000

- REMOTE TERMINAL PROGRAM
PROTOCOL TABLE PSECTS

E 3

16-SEP-1984 02:15:27 VAX/VMS Macro V04-00
5-SEP-1984 03:15:47 [RTPAD.SRC]RTPAD.MAR;1

Page 61
(3)

RT
V0

```
      0768      856      .SBTTL  PROTOCOL TABLE PSECTS
00000000      857      .PSECT  PROTOTB,BYTE,NOEXE
      0000      858
      0000      859  PROTOTBL:
      0000      860
0010      0000      861      .WORD  <104>
00000463      0002      862      .LONG  CTERM_RT
      0004      0006      863      .WORD  <102>
00000000      0008      864      .LONG  VMSRT
      000C      865
00000000      866      .PSECT  PROTOTBL,BYTE,NOEXE
      0000      867
00000000      868      .PSECT  PROTOTBL1,BYTE,NOEXE
      0000      869
      0000      870  ENDPROTO:
      0000      871
      0000      872      .END    RTPAD
```

; <4> => CTERM protocol
; This is the initialization entry
; <2> => VMS remote terminal protocol
; This is the initialization entry


```

$$ .TAB
$$ .TABEND
$$ .TMP
$$ .TMP1
$$ .TMP2
$$ .TMPX
$$ .TMPX1
$$GBL
$$T1
ASTSQ_IOSB
ASTST_BUF
ASTCNT
ASTLM
BIOCNT
BIOLM
CHAR_BLOCK
CLISGET_VALUE
CLISPRESENT
CNTRLCHAN
CONNDESC
CTERMSCLOSE_LOG
CTERMSOPEN_LOG
CTERM_FLAG
CTERM_RT
CTPSB_PRO_MSGTYPE
DCS_TERM
DECRETERR
DEVSV_TRM
DEVBUFSIZ
DEVBUFSIZ_TEMP
DEVCLASS
DEVCLASS_TEMP
DEVDEPEND
DEVDEPEND2
DEVNAM
DEVNAMLEN
DEVTYPE
DEVTYPE_TEMP
DIBSK_LENGTH
DSCSB_CLASS
DSCSK_CLASS_D
DTE_DESC
DVIS_DEVBUFSIZ
DVIS_DEVCLASS
DVIS_DEVDEPEND
DVIS_DEVDEPEND2
DVIS_DEVNAM
DVIS_DEVTYPE
DVIS_UNIT
DVILIST
ENDPROTO
EXITHSG
FABSB_FNS
FABSC_BID
FABSC_BLN
FABSC_SEQ
FABSC_VAR

```

=	00000026C	R	03
=	000000280	R	03
=	000000000		
=	000000001		
=	0000000CF		
=	000000000	R	04
=	000000009		
=	000000000		
=	000000001		
=	000000004		
=	000000026		
	000000044	R	03
	000000040	R	03
	00000004C	R	03
	000000048	R	03
	000000060	RG	03
	*****	X	01
	*****	X	01
	00000154	RG	03
	000001A8	R	03
	*****	X	01
	*****	X	01
	000002B2	RG	03
	00000463	R	01
=	00000026		
	*****	X	01
	00000110	RG	03
	*****	X	01
	00000062	R	03
	0000005C	R	03
	00000060	R	03
	00000054	R	03
	00000064	R	03
	00000068	R	03
	00000070	RG	03
	0000006C	RG	03
	00000061	R	03
	00000058	R	03
=	00000074		
=	00000003		
=	00000002		
=	00000172	R	03
=	00000008		
=	00000004		
=	0000000A		
=	0000001C		
=	00000020		
=	00000006		
=	0000000C		
	0000055E	R	01
	00000000	R	07
	00000128	R	03
=	00000034		
=	00000003		
=	00000050		
=	00000000		
=	00000002		

FABSL_ALQ
FABSL_FNA
FABSL_FOP
FABSV_CHAN_MODE
FABSV_FILE_MODE
FABSV_GET
FABSV_LNM_MODE
FABSW_GBC
FINALACS
FINALPATH
FIRSTCMD
FIRSTMSG
FLGSM_VAXHOST
FOO_RSTS 1
GETJPI_ITMI.ST
HOST_OPSYS
INDFCAG
INFOMSG1
INFOMSG2
INIT
IOSM_CTRLYAST
IOS_READVBLK
IOS_SETMODE
JPIS_ASTCNT
JPIS_ASTLM
JPIS_BIOCNT
JPIS_BIOLM
JPI_IOSB
LEN
LIB\$ASN_WTH_MBX
LIB\$CVT_HTB
LIB\$DISABLE_CTRL
LIB\$ENABLE_CTRL
LIB\$PUT_OUTPUT
LIB\$SIGNAL
LINKCHAN
LOCAL_PID
LOG_DESC
LOG_FILE_DESC
MAILCHAN
MAXMSG
MAXMSGSIZE
NODEDESC
NODENAME
NODE_NAME_DESC
NOTTERM
NOTVMS
NOT_DTE
OBJ_C_PREFIX
OBJ_DESC
OLDCTRL
OLDSETRWM
OLD_DESC
PROTOTBL
PROTO_ECO
PSTHRU_CRLF
PSTHRU_MSG

=	00000010		
=	0000002C		
=	00000004		
=	00000002		
=	00000004		
=	00000001		
=	00000000		
=	00000048		
	00000200	RG	03
	000001F8	RG	03
	00000311	RG	03
	00000324	R	03
	00000010		
=	00000392	R	01
	0000000C	R	03
	0000031A	R	03
	00000280	RG	03
	000004E9	R	01
	00000522	R	01
	000000D5	R	01
=	00000080		
=	00000031		
=	00000023		
	*****	X	03
	*****	X	03
	*****	X	03
	*****	X	03
	00000004	R	03
	00000050	R	03
	*****	X	01
	*****	X	01
	*****	X	01
	*****	X	01
	*****	X	01
	*****	X	01
	*****	X	01
	00000144	RG	03
	00000000	R	03
	00000167	R	03
	000001A0	R	03
	00000148	RG	03
=	0000041A		
	00000764	R	03
	0000017D	R	03
	00000180	RG	03
	00000198	R	03
	0000011C	R	03
	00000138	R	03
	0000027C	R	01
=	00000003		
	00000189	R	03
	0000031C	R	03
	00000320	R	03
	0000015C	R	03
	00000000	R	05
	00000319	RG	03
	00000210	R	03
	00000208	R	03

RT		AS		PS
Sy		AS		--
		CT		.
		CT		\$A
		CT		R
		CT		RT
		CT		
		CT		
		CT		
		CT		
		CT		
		CT		
		CT		
		GE		
		LI		
		PA		
		PR		
		QC		
		QF		
		QR		
		QR		
		QR		
		QR		
		QR		
		SH		
		VM		
		VM		
		VM		
		VM		
		WR		

RTPAD
Symbol table

- REMOTE TERMINAL PROGRAM

G 3

16-SEP-1984 02:15:27 VAX/VMS Macro V04-00
5-SEP-1984 03:15:47 [RTPAD.SRC]RTPAD.MAR;1

Page 63
(3)

QUIT_PC	= 0000030D	RG	03
RAB\$B_RAC	= 0000001E		
RAB\$C_BID	= 00000001		
RAB\$C_BLN	= 00000044		
RAB\$C_SEQ	= 00000000		
RAB\$C_CTX	= 00000018		
RAB\$C_ROP	= 00000004		
RDWRTCHAN	0000014C	RG	03
READCHAN	0000014C	RG	03
RECORD_QUIT	0000049C	RG	01
REMS_ATPC	= 01FE115B		
REMS_END	*****	X	03
REMS_FACILITY	= 000001FE		
REMS_NETERR	*****	X	03
REMS_NOPROT	*****	X	03
REMS_NOTERM	*****	X	03
REMOTENODE	*****	X	01
RETSTATUS	00000309	RG	03
RT\$AB_ASTBLK	000002E3	RG	03
RTLOG\$V_BANNER	= 00000000		
RTLOG_BUF	000002C2	R	03
RTLOG_DESC	000002BA	R	03
RTLOG_FLAGS	000002B6	RG	03
RTPAD	00000000	R	01
RTPAD_LOGNAM	000002D2	R	03
SHR\$K_SHRDEF	= 00000001		
SHR\$ATPL	= 00001158		
SHR\$TEXT	= 00001130		
SS\$EXQUOTA	*****	X	01
SS\$NOSUCHNODE	*****	X	01
SS\$NOTRAN	*****	X	01
SS\$WASCLR	*****	X	01
STR\$APPEND	*****	X	01
STR\$CONCAT	*****	X	01
STR\$FREE1_DX	*****	X	01
SY\$K_INFO	*****	X	01
SY\$S\$SIGN	*****	GX	01
SY\$S\$CANCEL	*****	GX	01
SY\$S\$CLOSE	*****	GX	01
SY\$S\$CONNECT	*****	GX	01
SY\$S\$GETDEV	*****	GX	01
SY\$S\$GETDVI	*****	GX	01
SY\$S\$GETJPIW	*****	GX	01
SY\$S\$HIBER	*****	GX	01
SY\$S\$NODE	000004C6	R	01
SY\$S\$OPEN	*****	GX	01
SY\$S\$PUTMSG	*****	GX	01
SY\$S\$QIO	*****	GX	01
SY\$S\$QIOW	*****	GX	01
SY\$S\$SETAST	*****	GX	01
SY\$S\$SETRUM	*****	GX	01
SY\$S\$TRNLOG	*****	GX	01
SYSINFAB	0000021C	RG	03
SYSINRAB	0000026C	RG	03
TERM\$EMULATE	*****	X	01
TERMCHAR	00000084	RG	03
TERMMBXCHAN	00000158	RG	03

TERMUNIT
TTYDESC
UN\$NET_CONNECT
VMSRT
WAKEFLAG
WRITECHAN
WRITEQIO

00000080	RG	03
000004D6	RG	01
*****	X	01
*****	X	01
000002B1	RG	03
00000150	RG	03
*****	X	01

RT
VA

Ph
--
In
Co
Pa
Sy
Pa
Sy
Ps
Cr
As
Th
25
Th
33
9

Ma
--
\$
--
\$
--
\$
TO
52
Th
MA

+-----+
! Psect synopsis !
+-----+

PSECT name	Allocation	PSECT No.	Attributes
. ABS .	00000000 (0.)	00 (0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
RTPAD	000005B6 (1462.)	01 (1.)	NOPIC USR CON REL LCL NOSHR EXE RD NOWRT NOVEC BYTE
\$ABSS	00000000 (0.)	02 (2.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE
RTPAD	00000768 (1896.)	03 (3.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC LONG
\$RMSNAM	00000009 (9.)	04 (4.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC BYTE
PROTOTB	0000000C (12.)	05 (5.)	NOPIC USR CON REL LCL NOSHR NOEXE RD WRT NOVEC BYTE
PROTOTBL	00000000 (0.)	06 (6.)	NOPIC USR CON REL LCL NOSHR NOEXE RD WRT NOVEC BYTE
PROTOTBL1	00000000 (0.)	07 (7.)	NOPIC USR CON REL LCL NOSHR NOEXE RD WRT NOVEC BYTE

+-----+
! Performance indicators !
+-----+

Phase	Page faults	CPU Time	Elapsed Time
Initialization	31	00:00:00.08	00:00:00.77
Command processing	129	00:00:00.47	00:00:04.39
Pass 1	694	00:00:17.42	00:01:07.76
Symbol table sort	31	00:00:02.85	00:00:13.32
Pass 2	578	00:00:04.24	00:00:19.02
Symbol table output	1	00:00:00.13	00:00:00.56
Psect synopsis output	0	00:00:00.02	00:00:00.02
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	1466	00:00:25.22	00:01:45.84

The working set limit was 2400 pages.
173467 bytes (339 pages) of virtual memory were used to buffer the intermediate code.
There were 150 pages of symbol table space allocated to hold 2667 non-local and 46 local symbols.
872 source lines were read in Pass 1, producing 31 object records in Pass 2.
63 pages of virtual memory were used to define 56 macros.

+-----+
! Macro library statistics !
+-----+

Macro library name	Macros defined
-\$255\$DUA28:[RTPAD.OBJ]RTPAD.MLB;1	4
-\$255\$DUA28:[SYS.OBJ]LIB.MLB;1	12
-\$255\$DUA28:[SYSLIB]STARLET.MLB;2	36
TOTALS (all libraries)	52

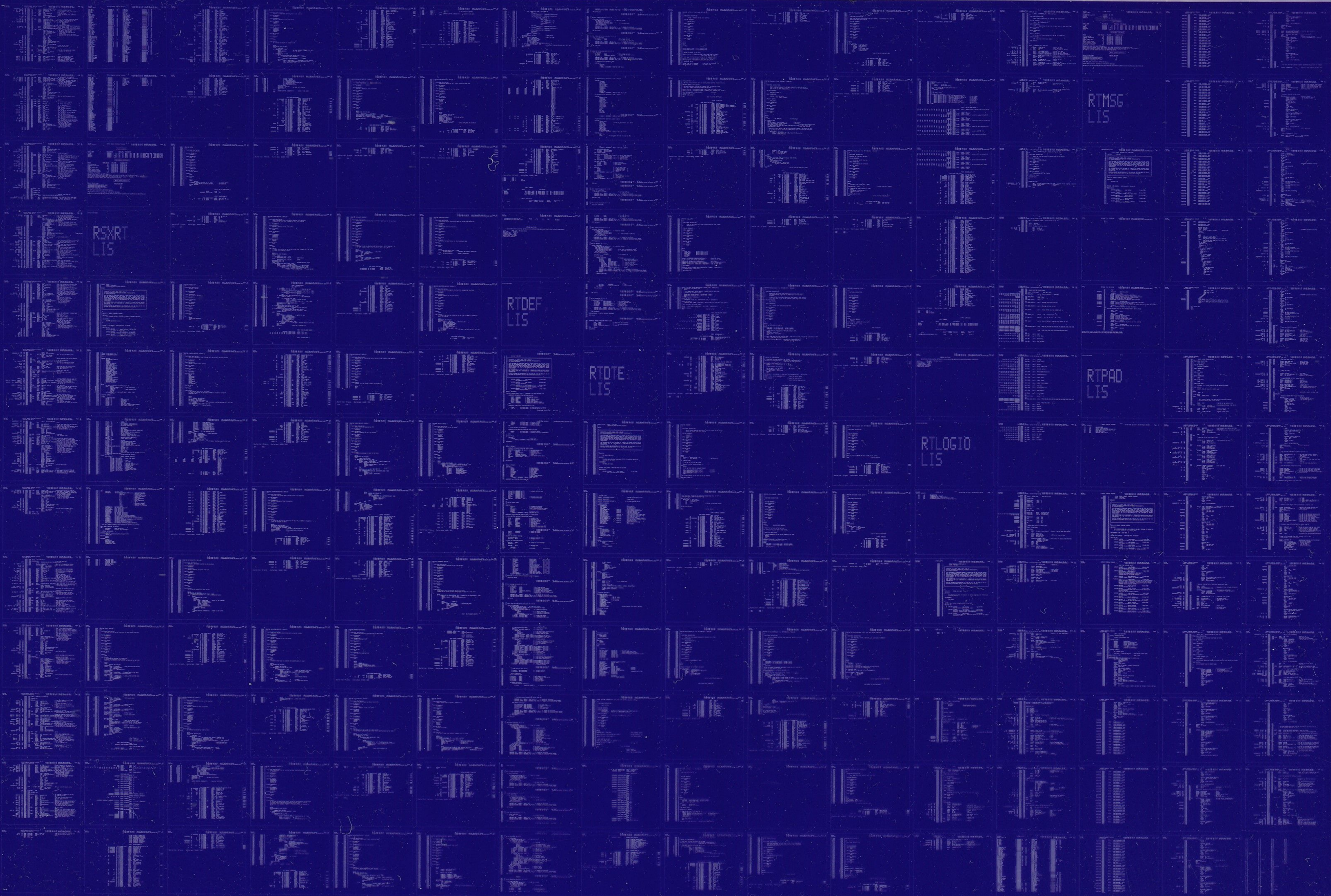
3096 GETS were required to define 52 macros.

There were no errors, warnings, or information messages.

MACRO/LIS=LIS\$:RTPAD/OBJ=OBJ\$:RTPAD MSRC\$:RTPAD/UPDATE=(ENH\$:RTPAD)+EXECMLS/LIB+LIB\$:RTPAD/LIB

0334 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY



0335

AH-BT13A-SE
 VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY