

RRRRRRRRRRRR		TTTTTTTTTTTT	PPPPPPPPPPPP		AAAAAAAAAA		DDDDDDDDDDDD	
RRRRRRRRRRRR		TTTTTTTTTTTT	PPPPPPPPPPPP		AAAAAAAAAA		DDDDDDDDDDDD	
RRRRRRRRRRRR		TTTTTTTTTTTT	PPPPPPPPPPPP		AAAAAAAAAA		DDDDDDDDDDDD	
RRR	RRR	TTT	PPP	PPP	AAA	AAA	DDD	DDD
RRR	RRR	TTT	PPP	PPP	AAA	AAA	DDD	DDD
RRR	RRR	TTT	PPP	PPP	AAA	AAA	DDD	DDD
RRR	RRR	TTT	PPP	PPP	AAA	AAA	DDD	DDD
RRR	RRR	TTT	PPP	PPP	AAA	AAA	DDD	DDD
RRR	RRR	TTT	PPP	PPP	AAA	AAA	DDD	DDD
RRRRRRRRRRRR		TTT	PPPPPPPPPPPP		AAA	AAA	DDD	DDD
RRRRRRRRRRRR		TTT	PPPPPPPPPPPP		AAA	AAA	DDD	DDD
RRRRRRRRRRRR		TTT	PPPPPPPPPPPP		AAA	AAA	DDD	DDD
RRR	RRR	TTT	PPP		AAAAAAAAAAAAAAAA		DDD	DDD
RRR	RRR	TTT	PPP		AAAAAAAAAAAAAAAA		DDD	DDD
RRR	RRR	TTT	PPP		AAAAAAAAAAAAAAAA		DDD	DDD
RRR	RRR	TTT	PPP		AAA	AAA	DDD	DDD
RRR	RRR	TTT	PPP		AAA	AAA	DDD	DDD
RRR	RRR	TTT	PPP		AAA	AAA	DDD	DDD
RRR	RRR	TTT	PPP		AAA	AAA	DDDDDDDDDDDD	
RRR	RRR	TTT	PPP		AAA	AAA	DDDDDDDDDDDD	
RRR	RRR	TTT	PPP		AAA	AAA	DDDDDDDDDDDD	

[illegible]

(1)	55	DECLARATIONS
(2)	99	DTE_DF03 - DF03 autodial routine
(3)	197	WRITE_STR - write string to port channel
(4)	227	READ_CHAR - read status character from port
(5)	261	READ_DONE - ast for read completion
(6)	306	WRITE_STR_TO_USER - write string to command channel

```
0000 1      .TITLE DTE_DF03.- Sample SET HOST/DTE dialer module
0000 2      .IDENT 'V04-000'
0000 3
0000 4
0000 5      :*****
0000 6      :
0000 7      :*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0000 8      :*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0000 9      :*  ALL RIGHTS RESERVED.
0000 10     :
0000 11     :*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0000 12     :*  ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0000 13     :*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0000 14     :*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0000 15     :*  OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0000 16     :*  TRANSFERRED.
0000 17     :
0000 18     :*  THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0000 19     :*  AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0000 20     :*  CORPORATION.
0000 21     :
0000 22     :*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0000 23     :*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0000 24     :
0000 25     :*****
0000 26     :
0000 27     :
0000 28     :
0000 29     :++
0000 30     :
0000 31     : FACILITY:
0000 32     :
0000 33     :     SET HOST/DTE
0000 34     :
0000 35     : ABSTRACT:
0000 36     :
0000 37     :     Provide modem-specific support for autodialing on a DF03, and
0000 38     :     serve as example for other modem types. Activated as a sharable
0000 39     :     image when SET HOST ttcn: /DTE /DIAL=(number:string,MODEM_TYPE=DF03)
0000 40     :     is run.
0000 41     :
0000 42     : ENVIRONMENT:
0000 43     :
0000 44     :     VAX/VMS, user mode.
0000 45     :
0000 46     : --
0000 47     :
0000 48     : AUTHOR: Jake VanNoy, CREATION DATE: 11-Apr-1984
0000 49     :
0000 50     : MODIFIED BY:
0000 51     :
0000 52     : **
0000 53     :
0000 54     :
0000 55     : .SBTTL DECLARATIONS
0000 56     :
0000 57     : INCLUDE FILES:
```



```
0000 58 ;
0000 59 ;      $SHRDEF      ; shared messages
0000 60      $STSDEF      ; status fields
0000 61
0000 62 ;
0000 63 ; MACROS:
0000 64 ;
0000 65 ;
0000 66 ;
0000 67 ; EQUATED SYMBOLS:
0000 68 ;
000001FE 0000 69 REM$_FACILITY = ^X1FE
01FE1110 0000 70 REM$_BADVALUE = SHR$_BADVALUE!<REM$_FACILITY@16>
0000000D 0000 71 CR = 13
0000000A 0000 72 LF = 10
0000 73
0000 74 ;
0000 75 ; OWN STORAGE:
0000 76 ;
0000 77
00000001 0000 78 CTRLB_DESC: .LONG 1 ; length
00000000 0004 79 .LONG 0 ; will get filled in by code
00000002 0008 80
00000002 0008 81 CTRLB_STR: .LONG 2 ; '2' is ^B
000C 82
00000022' 000C 83 CONN_DESC: .LONG CONN_STR_LEN ; length
00000000 0010 84 .LONG 0 ; will get filled in by code
6E 6F 69 74 63 65 6E 6E 6F 43 0A 0D 0014 85 CONN_STR: .ASCII <CR><LF>/Connection made to remote port/<CR><LF>
6D 65 72 20 6F 74 20 65 64 61 6D 20 0020
0A 0D 74 72 6F 70 20 65 74 6F 002C
00000022 0036 86 CONN_STR_LEN = .-CONN_STR
0036 87
00000024' 0036 88 FAIL_DESC: .LONG FAIL_STR_LEN ; length
00000000 003A 89 .LONG 0 ; will get filled in by code
20 6F 74 20 64 65 6C 69 61 46 0A 0D 003E 90 FAIL_STR: .ASCII <CR><LF>/Failed to connect to remote port/<CR><LF>
72 20 6F 74 20 74 63 65 6E 6E 6F 63 004A
0A 0D 74 72 6F 70 20 65 74 6F 6D 65 0056
00000024 0062 91 FAIL_STR_LEN = .-FAIL_STR
0062 92
0000006C 0062 93 READ_BUFFER: .BLKB 10 ; read buffer
00000000 006C 94 IOSB: .LONG 0,0 ; I/O status
00000000 0074 95 READ_STATUS: .LONG 0 ; completion status
0078 96
00000000 0078 97 USER_CHAN: .LONG 0 ; command channel own storage
```

```
007C 99 .SBTTL DTE_DF03 - DF03 autodial routine
007C 100
007C 101 :++
007C 102 :
007C 103 : FUNCTIONAL DESCRIPTION:
007C 104 :
007C 105 : Perform the necessary autodial protocol on a DF03-AC modem.
007C 106 :
007C 107 : CALLING SEQUENCE:
007C 108 :
007C 109 : DIAL_ROUTINE (number_desc, port_chan, command_chan)
007C 110 :
007C 111 : INPUT PARAMETERS:
007C 112 :
007C 113 : 4(AP) - descriptor of string specified in NUMBER:string
007C 114 : 8(AP) - channel number of port DF03 is connected to
007C 115 : 12(AP) - channel number of user's terminal
007C 116 :
007C 117 : IMPLICIT INPUTS:
007C 118 : NONE
007C 119 :
007C 120 : OUTPUT PARAMETERS:
007C 121 : NONE
007C 122 :
007C 123 : IMPLICIT OUTPUTS:
007C 124 : NONE
007C 125 :
007C 126 : COMPLETION CODES:
007C 127 :
007C 128 : R0 - status
007C 129 :
007C 130 : SIDE EFFECTS:
007C 131 : NONE
007C 132 :
007C 133 : --
007C 134 :
00000004 007C 135 number = 4
00000008 007C 136 port_chan = 8
0000000C 007C 137 command_chan = 12
007C 138
007C 139 .TRANSFER DIAL_ROUTINE
007C 140 .MASK DIAL_ROUTINE
0002' 31 007E 141 BRW DIAL_ROUTINE+2
0081 142
001C 0081 143 .ENTRY DIAL_ROUTINE,^M<R2,R3,R4>
0083 144
FO AF 0C AC 3C 0083 145 MOVZWL command_chan(AP),user_chan ; save for later
52 04 AC D0 0088 146 MOVL number(AP),R2 ; fetch address of descriptor
53 62 3C 008C 147 MOVZWL (R2),R3 ; length of string
54 04 A2 D0 008F 148 MOVL 4(R2),R4 ; address
0093 149
0093 150 ; Loop through string to check for illegal characters
0093 151
0093 152 10$:
64 3D 91 0093 153 CMPB #^A/=/, (R4) ; '=' is pause character
0A 13 0096 154 BEQL 20$ ; branch if match
64 30 91 0098 155 CMPB #^A/0/, (R4) ; check for number
```



```

        64 0C 1A 009B 156      BGTRU 30$      ; Branch if less than legal
        39 91 009D 157      CMPB  #^A/9/,(R4) ; check for number
        07 1F 00A0 158      BLSSU 30$      ; Branch if more than legal
        54 D6 00A2 159 20$: INCL  R4      ; next character
        EC 53 F5 00A4 160      SOBGTR R3,10$ ; legal character, loop
        1A 11 00A7 161      BRB  40$      ; continue, number ok
           00A9 162
           00A9 163      ; error in number string
           00A9 164
        04 AC DD 00A9 165 30$: PUSHL number(AP) ; signal error
        01 DD 00AC 166      PUSHL #1      ; number of FA0 args
        01FE1110 8F DD 00AE 167      PUSHL #REMS_BADVALUE ; error type
00000000'GF 03 FB 00B4 168      CALLS #3,G^CIB$SIGNAL ; error
50 11FE1110 8F D0 00BB 169      MOVL #REMS_BADVALUE!ST$M_INHIB_MSG,R0 ; return status
           04 00C2 170      RET          ; return
           00C3 171 40$:
           00C3 172
           00C3 173      ; number string ok, continue.
           00C3 174      ; queue read for character
           00C3 175
        004B 30 00C3 176      BSBW READ_CHAR ; read status character
        27 50 E9 00C6 177      BLBC R0,100$ ; exit on error
           00C9 178
           00C9 179      ; Write string to modem
           00C9 180
        FF34 CF FF3B CF 9E 00C9 181      MOVAB CTRLB_STR,CTRLB_DESC+4 ; set address
        52 FF2C CF 9E 00D0 182      MOVAB CTRLB_DESC,R2 ; ^B initiates dial
           0019 30 00D5 183      BSBW WRITE_STR ; write string
           15 50 E9 00D8 184      BLBC R0,100$ ; exit on error
           00DB 185
        52 04 AC D0 00DB 186      MOVL number(AP),R2 ; fetch address of descriptor
           000F 30 00DF 187      BSBW WRITE_STR ; write number string
           0B 50 E9 00E2 188      BLBC R0,100$ ; exit on error
           00E5 189
           00E5 190      $HIBER_S ; wait for read to complete
           00EC 191
        50 85 AF D0 00EC 192      MOVL READ_STATUS,R0 ; set status
           00F0 193 100$:
           04 00F0 194      RET
           00F1 195
```

```

00F1 197 .SBTTL WRITE_STR - write string to port channel
00F1 198 :++
00F1 199 :
00F1 200 : FUNCTIONAL DESCRIPTION:
00F1 201 :
00F1 202 :     write a string to the DTE port
00F1 203 :
00F1 204 : CALLING SEQUENCE:
00F1 205 :
00F1 206 :     BSBW    WRITE_STR
00F1 207 :
00F1 208 : INPUT PARAMETERS:
00F1 209 :
00F1 210 :     R2 - address of descriptor to write
00F1 211 :
00F1 212 : COMPLETION CODES:
00F1 213 :
00F1 214 :     R0 - status
00F1 215 :
00F1 216 :--
00F1 217 :
00F1 218 WRITE_STR:
00F1 219
00F1 220     $QIOW_S -
00F1 221     CHAN = port_chan(AP),-           ; channel
00F1 222     FUNC = #IOS_WRITEVBLK!IOSM_NOFORMAT,- ; write no format
00F1 223     P1   = @4(R2),-               ; address
00F1 224     P2   = (R2)                  ; length
05 0110 225     RSB

```



```
0111 227 .SBTTL READ_CHAR - read status character from port
0111 228 :++
0111 229 :
0111 230 : FUNCTIONAL DESCRIPTION:
0111 231 :
0111 232 :     Read the status character from the DF03, allowing a maximum
0111 233 :     of 60 seconds for the event to occur.
0111 234 :
0111 235 : CALLING SEQUENCE:
0111 236 :
0111 237 :     BSBW    READ_CHAR
0111 238 :
0111 239 : INPUT PARAMETERS:
0111 240 :     NONE
0111 241 :
0111 242 : COMPLETION CODES:
0111 243 :
0111 244 :     R0 - status
0111 245 :
0111 246 :--
0111 247 :
0111 248 READ_CHAR:
0111 249
0111 250     $QIO_S -
0111 251         CHAN = port_chan(AP),- ; channel
0111 252         FUNC = #IOS_READVBLK!IOSM_TIMED!IOSM_PURGE,-
0111 253         - ; read timed, purge
0111 254         IOSB = IOSB,- ; I/O status
0111 255         ASTADR = READ_DONE,- ; ast routine
0111 256         P1 = READ_BUFFER,- ; address
0111 257         P2 = #1,- ; length
0111 258         P3 = #60 ; timeout
05 013B 259     RSB ; exit with status
```

```
013C 261 .SBTTL READ_DONE - ast for read completion
013C 262 :++
013C 263 :
013C 264 : FUNCTIONAL DESCRIPTION:
013C 265 :
013C 266 :     Check for timeout or status character
013C 267 :
013C 268 : CALLING SEQUENCE:
013C 269 :
013C 270 :     CALled as AST routine
013C 271 :
013C 272 : INPUT PARAMETERS:
013C 273 :     NONE
013C 274 :
013C 275 : COMPLETION CODES:
013C 276 :
013C 277 :     R0 - status
013C 278 :
013C 279 :--
013C 280 :
0004 013C 281 .ENTRY READ_DONE,^M<R2>
013E 282
50  FF2A CF 3C 013E 283 MOVZWL IOSB,R0 ; get status of read
    38 50 E9 0143 284 BLBC R0,100$ ; branch if timeout
    0146 285
52  FF18 CF 9A 0146 286 MOVZBL READ_BUFFER,R2 ; fetch data
    52 41 8F 91 014B 287 CMPB #^A/^A/,R2 ; status ok?
    1A 12 014F 288 BNEQ 10$ ; branch if not
52  FEB7 CF 9E 0151 289 MOVAB conn_desc,R2 ; set up string
04 A2 FEBA CF 9E 0156 290 MOVAB conn_str,4(R2) ; set up string
    0030 30 015C 291 BSBW WRITE_STR_TO_USER ; tell user, ready
    1C 50 E9 015F 292 BLBC R0,100$ ; exit on error
50  00000000'8F D0 0162 293 MOVL #SS$_NORMAL,R0 ; ready
    13 11 0169 294 BRB 100$ ; exit
    016B 295 10$:
52  FEC7 CF 9E 016B 296 MOVAB fail_desc,R2 ; set up string
04 A2 FECA CF 9E 0170 297 MOVAB fail_str,4(R2) ; set up string
    0016 30 0176 298 BSBW WRITE_STR_TO_USER ; tell user, ready
50  0000'8F 3C 0179 299 MOVZWL #SS$_RANGOP,R0 ; status
    FEF1 CF 50 D0 017E 300
    017E 301 100$: MOVL R0,READ_STATUS ; save status
    0183 302 $WAKE_S ; wake main stream
    018E 303
04  018E 304 RET
```



```
018F 306 .SBTTL WRITE_STR_TO_USER - write string to command channel
018F 307 :++
018F 308 :
018F 309 : FUNCTIONAL DESCRIPTION:
018F 310 :
018F 311 :     write a string to the user terminal channel
018F 312 :
018F 313 : CALLING SEQUENCE:
018F 314 :
018F 315 :     BSBW    WRITE_STR_TO_USER
018F 316 :
018F 317 : INPUT PARAMETERS:
018F 318 :
018F 319 :     R2 - address of descriptor to write
018F 320 :
018F 321 : COMPLETION CODES:
018F 322 :
018F 323 :     R0 - status
018F 324 :
018F 325 :--
018F 326 :
018F 327 WRITE_STR_TO_USER:
018F 328 :
018F 329 :     $QIOW_S -
018F 330 :         CHAN = user_chan,-           ; channel
018F 331 :         FUNC = #IOS$WRITEVBLK,-      ; write
018F 332 :         P1   = @4(R2),-              ; address
018F 333 :         P2   = (R2)                  ; length
05 01AF 334 :     RSB
01B0 335 :
01B0 336 : .END
```

DTE_DF03
Symbol table

- Sample SET HOST/DTE dialer module B 15

16-SEP-1984 02:18:23
5-SEP-1984 03:15:05

VAX/VMS Macro V04-00
[RTPAD.SRC]DTE_DF03.MAR;1

Page 9
(6)

```

$ST1          = 00000001
COMMAND_CHAN  = 0000000C
CONN_DESC     = 0000000C R    01
CONN_STR      = 00000014 R    01
CONN_STR_LEN  = 00000022
CR            = 0000000D
CTRLB_DESC    = 00000000 R    01
CTRLB_STR     = 00000008 R    01
DIAL_ROUTINE  = 00000081 RG   01
FAIL_DESC     = 00000036 R    01
FAIL_STR      = 0000003E R    01
FAIL_STR_LEN  = 00000024
IOSM_NOFORMAT = ***** X    01
IOSM_PURGE    = ***** X    01
IOSM_TIMED    = ***** X    01
IOS_READVBLK  = ***** X    01
IOS_WRITEVBLK = ***** X    01
IOSB          = 0000006C R    01
LF            = 0000000A
LIB$SIGNAL    = ***** X    01
NUMBER        = 00000004
PORT_CHAN     = 00000008
READ_BUFFER   = 00000062 R    01
READ_CHAR     = 00000111 R    01
READ_DONE     = 0000013C RG   01
READ_STATUS   = 00000074 R    01
REMS_BADVALUE = 01FE1110
REMS_FACILITY = 000001FE
SHRS_BADVALUE = 00001110
SS$_RANGUP    = ***** X    01
SS$_NORMAL    = ***** X    01
ST$M_INHIB_MSG = 10000000
SY$SHIBER     = ***** GX   01
SY$QIO        = ***** GX   01
SY$QIOW       = ***** GX   01
SY$WAKE       = ***** GX   01
USER_CHAN     = 00000078 R    01
WRITE_STR     = 000000F1 R    01
WRITE_STR_TO_USER = 0000018F R    01

```

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

PSECT name	Allocation	PSECT No.	Attributes
. ABS .	00000000 (0.)	00 (0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
. BLANK .	000001B0 (432.)	01 (1.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC BYTE
\$AB\$	00000000 (0.)	02 (2.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE

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

Phase	Page faults	CPU Time	Elapsed Time
-----	-----	-----	-----
Initialization	30	00:00:00.06	00:00:00.58
Command processing	123	00:00:00.39	00:00:04.09
Pass 1	166	00:00:01.98	00:00:10.35
Symbol table sort	0	00:00:00.13	00:00:00.44
Pass 2	68	00:00:00.56	00:00:03.19
Symbol table output	6	00:00:00.03	00:00:00.03
Psect synopsis output	1	00:00:00.02	00:00:00.01
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	396	00:00:03.17	00:00:18.70

The working set limit was 1200 pages.
14239 bytes (28 pages) of virtual memory were used to buffer the intermediate code.
There were 10 pages of symbol table space allocated to hold 178 non-local and 7 local symbols.
336 source lines were read in Pass 1, producing 18 object records in Pass 2.
16 pages of virtual memory were used to define 15 macros.

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

Macro library name	Macros defined
-----	-----
_\$255\$DUA28:[RTPAD.OBJ]RTPAD.MLB;1	0
-\$255\$DUA28:[SYS.OBJ]LIB.MLB;1	0
-\$255\$DUA28:[SYSLIB]STARLET.MLB;2	12
TOTALS (all libraries)	12

277 GETS were required to define 12 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS\$:DTE_DF03/OBJ=OBJ\$:DTE_DF03 MSRC\$:DTE_DF03/UPDATE=(ENH\$:DTE_DF03)+EXECML\$/LIB+LIB\$:RTPAD/LIB

0333 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

CTDSENSE
LIS

CTDUMSPEC
LIS

CTSENSE
LIS

RSTRT
LIS

CTERMRT
LIS

DTE_DF03
LIS

CTSETRT
LIS

CTOSET
LIS