





```

1 0001 0 %title 'FILECMDS File-Oriented Phone Commands'
2 0002 0 module filecmds (
3 0003 1 ident='V04-000') = begin
4 0004 1
5 0005 1
6 0006 1 *****
7 0007 1 *
8 0008 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY *
9 0009 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS. *
10 0010 1 * ALL RIGHTS RESERVED. *
11 0011 1 *
12 0012 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED *
13 0013 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE *
14 0014 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER *
15 0015 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY *
16 0016 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY *
17 0017 1 * TRANSFERRED. *
18 0018 1 *
19 0019 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE *
20 0020 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT *
21 0021 1 * CORPORATION. *
22 0022 1 *
23 0023 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS *
24 0024 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL. *
25 0025 1 *
26 0026 1 *
27 0027 1 *****
28 0028 1
29 0029 1
30 0030 1 ++
31 0031 1 Facility: VAX/VMS Telephone Facility, File-Oriented Phone Commands
32 0032 1
33 0033 1 Abstract: This module handles the following user commands:
34 0034 1 FACSIMILE Include file contents in conversation.
35 0035 1 TRANSCRIBE Make a transcript of conversation.
36 0036 1
37 0037 1
38 0038 1 Environment:
39 0039 1
40 0040 1 Author: Paul C. Anagnostopoulos, Creation Date: 6 November 1980
41 0041 1
42 0042 1 Modified By:
43 0043 1
44 0044 1 : Version
45 0045 1 01 -
46 0046 1 --

```

B  
C  
C  
D  
E  
F  
G  
H  
I  
J  
K  
L  
M  
N  
O  
P  
Q  
R  
S  
T  
U  
V  
W  
X  
Y  
Z  
[  
\  
]  
^  
\_  
`  
a  
b  
c  
d  
e  
f  
g  
h  
i  
j  
k  
l  
m  
n  
o  
p  
q  
r  
s  
t  
u  
v  
w  
x  
y  
z  
{  
|  
}  
~  
!

```

: 48      0047 1 %sbttl 'Module Declarations'
: 49      0048 1
: 50      0049 1   Libraries and Requires:
: 51      0050 1
: 52      0051 1
: 53      0052 1 library 'sys$library:starlet.l32';
: 54      0053 1 require 'phonereq';
: 55      0382 1
: 56      0383 1
: 57      0384 1   Table of Contents:
: 58      0385 1
: 59      0386 1
: 60      0387 1 forward routine
: 61      0388 1     phn$facsimile_cmd: novalue,
: 62      0389 1     phn$facsimile2: novalue,
: 63      0390 1     phn$transcribe_cmd: novalue;
: 64      0391 1
: 65      0392 1
: 66      0393 1   The following macro is used in PHN$TRANSCRIBE_CMD:
: 67      0394 1
: 68      0395 1
: 69      M 0396 1 macro transcribe(length,address) =
: 70      M 0397 1     if dec(line_counter) lss 0 then
: 71      M 0398 1         page headings();
: 72      M 0399 1         trans_rab[ra$w_rsz] = length;
: 73      M 0400 1         trans_rab[ra$l_rbf] = address;
: 74      M 0401 1         status = $put(ra$=trans_rab);
: 75      M 0402 1         check(.status);
: 76      0403 1 %;
: 77      0404 1
: 78      0405 1
: 79      0406 1   External References:
: 80      0407 1
: 81      0408 1
: 82      0409 1 external routine
: 83      0410 1     phn$conversing,
: 84      0411 1     phn$fresh_screen,
: 85      0412 1     phn$inform,
: 86      0413 1     phn$kill_ctl,
: 87      0414 1     phn$queue_smb,
: 88      0415 1     phn$talk;
: 89      0416 1
: 90      0417 1
: 91      0418 1   Own Variables:
: 92      0419 1
: 93      0420 1
: 94      0421 1 own
: 95      P 0422 1     facs_fab: $fab( fac=get,           ! RMS FAB for facsimile file.
: 96      0423 1         shr=get),
: 97      0424 1     facs_buffer: block[79,byte],      ! Buffer for getting file records.
: 98      P 0425 1     facs_rab: $rab( fab=facs_fab,      ! RMS RAB for facsimile file.
: 99      P 0426 1         rac=seq,
: 100     P 0427 1         rop=loc,
: 101     P 0428 1         ubf=facs_buffer,
: 102     0429 1         usz=79);

```

```

104 0430 1 %sbttl 'PHN$FACSIMILE_CMD - Include File in Conversation'
105 0431 1 ++
106 0432 1 Functional Description:
107 0433 1 This routine handles the FACSIMILE command, which is used to
108 0434 1 include the contents of a file into the conversation.
109 0435 1
110 0436 1 Formal Parameters:
111 0437 1 facs_spec Address of descriptor of file specification.
112 0438 1
113 0439 1 Implicit Inputs:
114 0440 1 global data
115 0441 1
116 0442 1 Implicit Outputs:
117 0443 1 global data
118 0444 1
119 0445 1 Returned Value:
120 0446 1 none
121 0447 1
122 0448 1 Side Effects:
123 0449 1
124 0450 1 --
125 0451 1
126 0452 1
127 0453 2 global routine phn$facsimile_cmd(facs_spec): novalue = begin
128 0454 2
129 0455 2 bind
130 0456 2 spec_dsc = .facs_spec: descriptor;
131 0457 2
132 0458 2 local
133 0459 2 status: long;
134 0460 2
135 0461 2
136 0462 2 ! Check to make sure we are currently talking to someone. Otherwise facsing
137 0463 2 ! makes no sense.
138 0464 2
139 0465 2 if not phn$conversing() then (
140 0466 3 phn$inform(phn$_notconv);
141 0467 3 return;
142 0468 2 );
143 0469 2
144 0470 2 ! Now we have to open the file we are going to facs. It is also necessary to
145 0471 2 ! connect the RAB so we can read records. If either of these operations fails,
146 0472 2 ! tell the user why.
147 0473 2
148 0474 2 facs_fab[fab$b_fns] = .spec_dsc[len];
149 0475 2 facs_fab[fab$l_fna] = .spec_dsc[ptr];
150 0476 2 status = $open(fab=facs_fab);
151 0477 3 if .status nequ rms$normal then (
152 0478 3 phn$inform(.status);
153 0479 3 return;
154 0480 2 );
155 0481 2 status = $connect(rab=facs_rab);
156 0482 3 if .status nequ rms$normal then (
157 0483 3 $close(fab=facs_fab);
158 0484 3 phn$inform(.status);
159 0485 3 return;
160 0486 2 );

```

```

: 161 0487 2
: 162 0488 2 ! Tell the user how to cancel the operation.
: 163 0489 2
: 164 0490 2 phn$inform(phn$_facscan);
: 165 0491 2
: 166 0492 2 ! Set the "facsimile in progress" flag so it can be cancelled if necessary.
: 167 0493 2
: 168 0494 2 phn$gv_facsimile = true;
: 169 0495 2
: 170 0496 2 ! Talk a carriage return so the file will start on a new line. Queue a
: 171 0497 2 ! steering message that will force us to facs the first record.
: 172 0498 2
: 173 0499 2 phn$talk(describe(%char(ret)));
: 174 0500 2 phn$queue_smb(smb__facsimile2);
: 175 0501 2 return;
: 176 0502 2
: 177 0503 1 end;

```

```

.TITLE FILECMDS FILECMDS File-Oriented Phone Commands
.IDENT \V04-000\

```

```

.PSECT $PLITS$,NOWRT,NOEXE,2

```

```

0D 0000 P.AAB: .ASCII <13>
00001 .BLKB 3
00000001, 00004 P.AAA: .LONG 1
00000000' 00008 .ADDRESS P.AAB

```

```

.PSECT $OWNS$,NOEXE,2

```

```

03 0000 FACS_FAB:
50 0001 .BYTE 3
0000 0002 .BYTE 80
00000000 0004 .WORD 0
00000000 0008 .LONG 0
00000000 000C .LONG 0
00000000 0010 .LONG 0
0000 0014 .WORD 0
02 0016 .BYTE 2
02 0017 .BYTE 2
00000000 0018 .LONG 0
00 001C .BYTE 0
00 001D .BYTE 0
00 001E .BYTE 0
02 001F .BYTE 2
00000000 0020 .LONG 0
00000000 0024 .LONG 0
00000000 0028 .LONG 0
00000000 002C .LONG 0
00000000 0030 .LONG 0
00 0034 .BYTE 0
00 0035 .BYTE 0
0000 0036 .WORD 0
00000000 0038 .LONG 0
0000 003C .WORD 0

```

```
00 0003E .BYTE 0
00 0003F .BYTE 0
00000000 00040 .LONG 0
00000000 00044 .LONG 0
0000 00048 .WORD 0
00 0004A .BYTE 0
00 0004B .BYTE 0
00000000 0004C .LONG 0
00050 FACS_BUFFER:
      .BLKB 79
0004F .BLKB 1
01 000A0 FACS_RAB:
      .BYTE 1
44 000A1 .BYTE 68
0000 000A2 .WORD 0
00010000 000A4 .LONG 65536
00000000 000A8 .LONG 0
00000000 000AC .LONG 0
0000# 000B0 .WORD 0[3]
0000 000B6 .WORD 0
00000000 000B8 .LONG 0
0000 000BC .WORD 0
00 000BE .BYTE 0
00 000BF .BYTE 0
004F 000C0 .WORD 79
0000 000C2 .WORD 0
00000000' 000C4 .ADDRESS FACS_BUFFER
00000000 000C8 .LONG 0
00000000 000CC .LONG 0
00000000 000D0 .LONG 0
00 000D4 .BYTE 0
00 000D5 .BYTE 0
00 000D6 .BYTE 0
00 000D7 .BYTE 0
00000000 000D8 .LONG 0
00000000' 000DC .ADDRESS FACS_FAB
00000000 000E0 .LONG 0
```

```
.EXTRN PHNS_OK, PHNS_ANSWERED
.EXTRN PHNS_BUSYCALL, PHNS_CANCEL
.EXTRN PHNS_CANTREACH, PHNS_CONFCALL
.EXTRN PHNS_DEAD, PHNS_DECNETLINK
.EXTRN PHNS_DIRCAN, PHNS_FACSCAN
.EXTRN PHNS_HELPCAN, PHNS_HUNGUP
.EXTRN PHNS_JUSTRANG, PHNS_LOGGEDOFF
.EXTRN PHNS_REJECTED, PHNS_RING
.EXTRN PHNS_REJECTJUNK
.EXTRN PHNS_SENDINGMAIL
.EXTRN PHNS_BADCMD, PHNS_BADHELP
.EXTRN PHNS_BADMAILCMD
.EXTRN PHNS_BADSMB, PHNS_BADSPEC
.EXTRN PHNS_HELPMISSING
.EXTRN PHNS_IVREDUNANS
.EXTRN PHNS_IVREDUNCALL
.EXTRN PHNS_LINKERROR, PHNS_NEEDUSER
.EXTRN PHNS_NOCALL, PHNS_NOROLDS
.EXTRN PHNS_NOPORTS, PHNS_NOPRIV
```

.....

					.EXTRN	PHNS_NOPROC, PHNS_NOTCONV	
					.EXTRN	PHNS_ONLYNODE, PHNS_PHONEBUSY	
					.EXTRN	PHNS_REMOTEERROR	
					.EXTRN	PHNS_TARGTERM, PHNS_UNPLUGGED	
					.EXTRN	PHNS_BADTERM, PHNS_SHAREDMBX	
					.EXTRN	PHNS_INPUTTERM, PHNSGQ_NODE_NAME	
					.EXTRN	PHNSGQ_SWITCH_HOOK	
					.EXTRN	PHNSGL_VIEWPORT_SIZE	
					.EXTRN	PHNSGB_SCROLL, PHNSGQ_PUBHEAD	
					.EXTRN	PHNSGB_FLAGS, PHNSCONVERSING	
					.EXTRN	PHNSFRESH_SCREEN	
					.EXTRN	PHNSINFORM, PHNSKILL_CTL	
					.EXTRN	PHNSQUEUE_SMB, PHNSTALK	
					.EXTRN	SYSSOPEN, SYSSCONNECT	
					.EXTRN	SYSSCLOSE	
					.PSECT	\$CODE\$,NOWRT,2	
					.ENTRY	PHNSFACSIMILE_CMD, Save R2,R3	: 0453
					MOVAB	FACS_FAB, R3	:
					MOVL	FACS_SPEC, R2	: 0456
					CALLS	#0, PHNSCONVERSING	: 0465
					BLBS	R0, 1\$	:
					PUSHL	#PHNS_NOTCONV	: 0466
					BRB	3\$	:
					MOVB	(R2), FACS_FAB+52	: 0474
					MOVL	4(R2), FACS_FAB+44	: 0475
					PUSHL	R3	: 0476
					CALLS	#1, SYSSOPEN	:
					MOVL	R0, STATUS	:
					CMPL	STATUS, #5557	: 0477
					BNEQ	2\$	:
					PUSHAB	FACS_RAB	: 0481
					CALLS	#1, SYSSCONNECT	:
					MOVL	R0, STATUS	:
					CMPL	STATUS, #65537	: 0482
					BEQL	4\$	:
					PUSHL	R3	: 0483
					CALLS	#1, SYSSCLOSE	:
					PUSHL	STATUS	: 0484
					CALLS	#1, PHNSINFORM	:
					RET		: 0482
					PUSHL	#PHNS_FACSCAN	: 0490
					CALLS	#1, PHNSINFORM	:
					BISB2	#8, PHNSGB_FLAGS	: 0494
					PUSHAB	P.AAA	: 0499
					CALLS	#1, PHNSTALK	:
					PUSHL	#16	: 0500
					CALLS	#1, PHNSQUEUE_SMB	:
					RET		: 0503

; Routine Size: 130 bytes, Routine Base: \$CODE\$ + 0000



```

179 0504 1 %sbttl 'PHN$FACSIMILE2 - Facsimile a Record'
180 0505 1 ++
181 0506 1 Functional Description:
182 0507 1 This steering message routine is called to facsimile the next record
183 0508 1 from the file.
184 0509 1
185 0510 1 Formal Parameters:
186 0511 1 none
187 0512 1
188 0513 1 Implicit Inputs:
189 0514 1 global data
190 0515 1
191 0516 1 Implicit Outputs:
192 0517 1 global data
193 0518 1
194 0519 1 Returned Value:
195 0520 1 none
196 0521 1
197 0522 1 Side Effects:
198 0523 1
199 0524 1 --
200 0525 1
201 0526 1
202 0527 2 global routine phn$facsimile2: novalue = begin
203 0528 2
204 0529 2 local
205 0530 2 status: long,
206 0531 2 record_dsc: descriptor;
207 0532 2
208 0533 2
209 0534 2 ! If the facsimile operation has been cancelled, talk a little message to
210 0535 2 ! that effect and bag the operation.
211 0536 2
212 0537 2 if not .phn$gv facsimile then (
213 L 0538 2 phn$talk(describe(%string('(CANCELLED)',
214 0539 2 %char(ret))));
215 0540 2 $close(fab=facs_fab);
216 0541 2 return;
217 0542 2 );
218 0543 2
219 0544 2 ! Read the next record from the file. There are three cases:
220 0545 2 1. We get a record (even if it is truncated). Build a descriptor
221 0546 2 for it and talk it into the conversation, with a carriage
222 0547 2 return. Queue a steering message for the next record.
223 0548 2
224 0549 2 2. We get end of file. Talk a little message to that effect and
225 0550 2 cancel the operation.
226 0551 2
227 0552 2 3. We get an I/O error. Talk a little message and cancel.
228 0553 2
229 0554 2 status = $get(rab=facs_rab);
230 0555 2
231 0556 2 selectoneu .status of set
232 0557 2 [rms$normal,
233 0558 2 rms$_rtb]: (record_dsc[len] = .facs_rab[rab$w_rsz];
234 0559 2 record_dsc[ptr] = .facs_rab[rab$l_rbf];
235 0560 2 phn$talk(record_dsc);

```

```

: 236      0561 3      phn$talk(describe(%char(ret)));
: 237      0562 3      phn$queue_smb(smb_facsimile2););
: 238      0563 3
: 239      L 0564 3 [rms$_eof]: (phn$talk(describe(%string('(EOF)',
: 240      0565 3      %char(ret))));
: 241      0566 3      $close(fab=facs_fab);
: 242      0567 3      phn$gv_facsimile = false;);
: 243      0568 3
: 244      L 0569 3 [otherwise]: (phn$talk(describe(%string('(FILE ERROR)',
: 245      0570 3      %char(ret))));
: 246      0571 3      $close(fab=facs_fab);
: 247      0572 3      phn$gv_facsimile = false;);
: 248      0573 3      ter;
: 249      0574 3
: 250      0575 3      return;
: 251      0576 3      end;

```

```

.PSECT $SPLITS,NOWRT,NOEXE,2
OD 29 44 45 4C 4C 45 43 4E 41 43 28 0000C P.AAD: .ASCII \ (CANCELLED)\<13>
      0000000C 00018 P.AAC: .LONG 12
      00000000 0001C .ADDRESS P.AAD
      OD 00020 P.AAF: .ASCII <13>
      00021 .BLKB 3
      00000001 00024 P.AAE: .LONG 1
      00000000 00028 .ADDRESS P.AAF
      OD 29 46 4F 45 28 0002C P.AAH: .ASCII \ (EOF)\<13>
      00032 .BLKB 2
      00000006 00034 P.AAG: .LONG 6
      00000000 00038 .ADDRESS P.AAH
OD 29 52 4F 52 52 45 20 45 4C 49 46 28 0003C P.AAJ: .ASCII \ (FILE ERROR)\<13>
      00049 .BLKB 3
      0000000D 0004C P.AAI: .LONG 13
      00000000 00050 .ADDRESS P.AAJ

```

.EXTRN SYSSGET

.PSECT \$CODE\$,NOWRT,2

```

      000C 00000 .ENTRY PHN$FACSIMILE2, Save R2,R3 : 0527
      53 00000000G 00 9E 00002 MOVAB SYSSCLOSE, R3
      52 0000G CF 9E 00009 MOVAB PHN$TALK, R2
      5E 08 C2 0000E SUP ? #8, SP
      OF 0000G CF 03 E0 00011 BBS #3, PHN$GB_FLAGS, 1$ : 0537
      0000' CF 9F 00017 PUSHAB P.AAC : 0539
      62 0000' 01 FB 0001B CALLS #1, PHN$TALK
      0000' CF 9F 0001E PUSHAB FACS_FAB : 0540
      63 01 FB 00022 CALLS #1, SYSSCLOSE
      0000' 04 00025 RET : 0537
      0000' CF 9F 00026 1$: PUSHAB FACS_RAB : 0554
      00000000G 00 01 FB 0002A CALLS #1, SYSSGET
      00010001 8F 50 D1 00031 CMPL STATUS, #65537 : 0557
      000181A8 8F 09 13 00038 BEQL 2$
      50 D1 0003A CMPL STATUS, #98728
      1F 12 00041 BNEQ 3$

```

FILECMDS  
V04-000

FILECMDS File-Oriented Phone Commands  
PHNSFACSIMILE2 - Facsimile a Record

I 16  
16-Sep-1984 02:09:16  
14-Sep-1984 12:53:25

VAX-11 Bliss-32 V4.0-742  
[PHONE.SRC]FILECMDS.B32;1

Page 9  
(4)

	6E	0000'	CF	B0	00043	2\$:	MOVW	FACS_RAB+34, RECORD_DSC	:	0558
04	AE	0000'	CF	D0	00048		MOVL	FACS_RAB+40, RECORD_DSC+4	:	0559
			5E	DD	0004E		PUSHL	SP	:	0560
	62		01	FB	00050		CALLS	#1, PHNSTALK	:	
		0000'	CF	9F	00053		PUSHAB	P.AAE	:	0561
	62		01	FB	00057		CALLS	#1, PHNSTALK	:	
			10	DD	0005A		PUSHL	#16	:	0562
0000G	CF		01	FB	0005C		CALLS	#1, PHNSQUEUE_SMB	:	
			04	00061			RET		:	0556
0001827A	8F		50	D1	00062	3\$:	CMPL	STATUS, #98938	:	0564
			06	12	00069		BNEQ	4\$	:	
		0000'	CF	9F	0006B		PUSHAB	P.AAG	:	0565
			04	11	0006F		BRB	5\$	:	
		0000'	CF	9F	00071	4\$:	PUSHAB	P.AAI	:	0570
	62		01	FB	00075	5\$:	CALLS	#1, PHNSTALK	:	
		0000'	CF	9F	00078		PUSHAB	FACS_FAB	:	0571
	63		01	FB	0007C		CALLS	#1, SYSSCLOSE	:	
0000G	CF		08	8A	0007F		BICB2	#8, PHNSGB_FLAGS	:	0572
			04	00084			RET		:	0576

; Routine Size: 133 bytes, Routine Base: \$CODE\$ + 0082

```

: 253      0577 1 %sbttl 'PHN$TRANSCRIBE_CMD - Handle TRANSCRIBE Command'
: 254      0578 1 ++
: 255      0579 1 Functional Description:
: 256      0580 1 This routine handles the TRANSCRIBE command, which is used to make a
: 257      0581 1 transcript of the conversation. We record the conversation that
: 258      0582 1 has taken place UP TO THIS POINT, but only for people we are currently
: 259      0583 1 talking to.
: 260      0584 1
: 261      0585 1 Formal Parameters:
: 262      0586 1 trans_spec The address of the descriptor of the file to receive
: 263      0587 1 the transcript.
: 264      0588 1
: 265      0589 1 Implicit Inputs:
: 266      0590 1 global data
: 267      0591 1
: 268      0592 1 Implicit Outputs:
: 269      0593 1 global data
: 270      0594 1
: 271      0595 1 Returned Value:
: 272      0596 1 none
: 273      0597 1
: 274      0598 1 Side Effects:
: 275      0599 1
: 276      0600 1 --
: 277      0601 1
: 278      0602 1
: 279      0603 2 global routine phn$transcribe_cmd(trans_spec): novalue = begin
: 280      0604 2
: 281      0605 2 bind
: 282      0606 2 spec_dsc = .trans_spec: descriptor;
: 283      0607 2 own
: 284      0608 2 trans_fab: $fab(dnm='.PHN',
: 285      0609 2 fac=put,
: 286      0610 2 mrs=132,
: 287      0611 2 org=seq,
: 288      0612 2 rat=cr,
: 289      0613 2 rfm=var,
: 290      0614 2 shr=nil),
: 291      0615 2 trans_rab: $rab(fab=trans_fab,
: 292      0616 2 rac=seq);
: 293      0617 2
: 294      0618 2 own ! These variables are OWN because
: 295      0619 2 page_number: long, ! of up-level referencing.
: 296      0620 2 line_counter: long;
: 297      0621 2 literal
: 298      0622 2 array_size = 20/pub_k_minlines;
: 299      0623 2 local
: 300      0624 2 status: long,
: 301      0625 2 pub_array: vector[array_size,long],
: 302      0626 2 ctl_array: vector[array_size,long],
: 303      0627 2 p: ref pub, cur_p: ref pub,
: 304      0628 2 min_stamp: long,
: 305      0629 2 i: long, min_i: long,
: 306      0630 2 c: ref ctl,
: 307      0631 2 line_buf: block[132,byte];

```



```
00000000 20 000FB .BYTE 32
00000000 00 000FC .LONG 0
00000000 00 00100 .BYTE 0
00000000 00 00101 .BYTE 0
00000000 02 00102 .BYTE 2
00000000 02 00103 .BYTE 2
00000000 00 00104 .LONG 0
00000000 00 00108 .LONG 0
00000000 00 0010C .LONG 0
00000000 00 00110 .LONG 0
00000000 00 00114 .ADDRESS P.AAK
00000000 00 00118 .BYTE 0
00000000 04 00119 .BYTE 4
00000000 0084 0011A .WORD 132
00000000 0000 0011C .LONG 0
00000000 00 00120 .WORD 0
00000000 00 00122 .BYTE 0
00000000 00 00123 .BYTE 0
00000000 00 00124 .LONG 0
00000000 00 00128 .LONG 0
00000000 0000 0012C .WORD 0
00000000 00 0012E .BYTE 0
00000000 00 0012F .BYTE 0
00000000 00 00130 .LONG 0
00000000 01 00134 TRANS_RAB:
00000000 44 00135 .BYTE 1
00000000 0000 00136 .BYTE 68
00000000 00000000 00138 .WORD 0
00000000 00000000 0013C .LONG 0
00000000 00000000 00140 .LONG 0
00000000 0000# 00144 .WORD 0[3]
00000000 0000 0014A .WORD 0
00000000 00000000 0014C .LONG 0
00000000 0000 00150 .WORD 0
00000000 00 00152 .BYTE 0
00000000 00 00153 .BYTE 0
00000000 0000 00154 .WORD 0
00000000 0000 00156 .WORD 0
00000000 00000000 00158 .LONG 0
00000000 00000000 0015C .LONG 0
00000000 00000000 00160 .LONG 0
00000000 00000000 00164 .LONG 0
00000000 00 00168 .BYTE 0
00000000 00 00169 .BYTE 0
00000000 00 0016A .BYTE 0
00000000 00 0016B .BYTE 0
00000000 00000000 0016C .LONG 0
00000000 00000000 00170 .ADDRESS TRANS_FAB
00000000 00000000 00174 .LONG 0
00000000 00178 PAGE_NUMBER:
00000000 .BLKB 4
00000000 0017C LINE_COUNTER:
00000000 .BLKB 4
.EXTRN SYSSPUT, SYSSFAO
```

				.PSECT	\$CODE\$	NOWRT,2		
				007C	00000	PAGE_HEADINGS:		
	56	FB	AF	9E	00002	.WORD	Save R2,R3,R4,R5,R6	: 0635
	55	00000000G	00	9E	00006	MOVAB	PAGE HEADINGS, R6	
	54	00000000G	00	9E	0000D	MOVAB	SYSSPUT, R5	
	53	0000'	CF	9E	00014	MOVAB	LIB\$SIGNAL, R4	
	5E	FF78	CE	9E	00019	MOVAB	LINE_COUNTER, R3	
	7E	84	8F	9A	0001E	MOVZBL	-136(TSP), SP	
04	AE	08	AE	9E	00022	MOVAB	#132, FAO_BUF	: 0640
	63		37	D0	00027	MOVL	FAO_BUF+8, FAO_BUF+4	
		FC	A3	D5	0002A	TSTL	#55- LINE_COUNTER	: 0643
			06	13	0002D	BEQL	PAGE_NUMBER	: 0645
	03		63	F4	0002F	SOBGEQ	1\$	
	66		00	FB	00032	CALLS	LINE_COUNTER, 1\$	: 0646
DA	A3		01	B0	00035	1\$:	#0, PAGE HEADINGS	
EO	A3	0000'	CF	9E	00039	MOVW	#1, TRANS_RAB+34	
		B8	A3	9F	0003F	MOVAB	P.AAL, TRANS_RAB+40	
	65		01	FB	00042	PUSHAB	TRANS_RAB	
	52		50	D0	00045	CALLS	#1, SYSSPUT	
	05		52	E8	00048	MOVL	R0, STATUS	
			52	DD	0004B	BLBS	STATUS, 2\$	
	64		01	FB	0004D	PUSHL	STATUS	
		FC	A3	D6	00050	2\$:	#1, LIB\$SIGNAL	
		FC	A3	DD	00053	INCL	PAGE_NUMBER	: 0647
			7E	D4	00056	PUSHL	PAGE_NUMBER	: 0654
		08	AE	9F	00058	CLRL	-(SP)	
		0C	AE	9F	0005B	PUSHAB	FAO_BUF	
		0000'	CF	9F	0005E	PUSHAB	FAO_BUF	
00000000G	00		05	FB	00062	PUSHAB	P.AAM	
	52		50	D0	00069	CALLS	#5, SYSSFAO	
	05		52	E8	0006C	MOVL	R0, STATUS	
			52	DD	0006F	BLBS	STATUS, 3\$	: 0655
	64		01	FB	00071	PUSHL	STATUS	
	03		63	F4	00074	3\$:	#1, LIB\$SIGNAL	
	66		00	FB	00077	SOBGEQ	LINE_COUNTER, 4\$	: 0656
DA	A3		6E	B0	0007A	4\$:	#0, PAGE HEADINGS	
EO	A3	04	AE	D0	0007E	MOVW	FAO_BUF, TRANS_RAB+34	
		B8	A3	9F	00083	MOVL	FAO_BUF+4, TRANS_RAB+40	
	65		01	FB	00086	PUSHAB	TRANS_RAB	
	52		50	D0	00089	CALLS	#1, SYSSPUT	
	05		52	E8	0008C	MOVL	R0, STATUS	
			52	DD	0008F	BLBS	STATUS, 5\$	
	64		01	FB	00091	PUSHL	STATUS	
	03		63	F4	00094	5\$:	#1, LIB\$SIGNAL	
	66		00	FB	00097	SOBGEQ	LINE_COUNTER, 6\$	: 0657
DA	A3		01	B0	0009A	6\$:	#0, PAGE HEADINGS	
EO	A3	0000'	CF	9E	0009E	MOVW	#1, TRANS_RAB+34	
		B8	A3	9F	000A4	MOVAB	P.AAO, TRANS_RAB+40	
	65		01	FB	000A7	PUSHAB	TRANS_RAB	
	52		50	D0	000AA	CALLS	#1, SYSSPUT	
	05		52	E8	000AD	MOVL	R0, STATUS	
			52	DD	000B0	BLBS	STATUS, 7\$	
	64		01	FB	000B2	PUSHL	STATUS	
			04	000B5	7\$:	CALLS	#1, LIB\$SIGNAL	: 0660
						RET		

FILECMDS  
VJ4-000

FILECMDS File-Oriented Phone Commands  
PHN\$TRANSCRIBE\_CMD - Handle TRANSCRIBE Command

B 1  
16-Sep-1984 02:09:16  
14-Sep-1984 12:53:25

VAX-11 Bliss-32 V4.0-742  
[PHONE.SRC]FILECMDS.B32;1

Page 14  
(6)

; Routine Size: 182 bytes, Routine Base: \$CODE\$ + 0107

INF  
V04

.....



```
339 0661 2 ! We begin by opening a new file in which to place the transcript. We also
340 0662 2 ! have to connect the RAB so we can write records. If either of these actions
341 0663 2 ! fails, tell the user why.
342 0664 2
343 0665 2 trans_fab[fab$b_fns] = .spec_dsc[len];
344 0666 2 trans_fab[fab$l_fna] = .spec_dsc[ptr];
345 0667 2 status = $create(fab=trans_fab);
346 0668 2 if .status nequ rms$normal then (
347 0669 3     phn$inform(.status);
348 0670 3     return;
349 0671 2 );
350 0672 2 status = $connect(rab=trans_rab);
351 0673 2 if .status nequ rms$normal then (
352 0674 3     phn$inform(.status);
353 0675 3     $close(fab=trans_fab);
354 0676 3     return;
355 0677 2 );
356 0678 2
357 0679 2 ! Now we inform all of the people that we are going to record them.
358 0680 2
359 L 0681 2 phn$stalk(describe(%string(%char(ret),
360 L 0682 2     'I AM NOW RECORDING THE CONVERSATION UP TO THIS POINT.',
361 0683 2     %char(ret)))));
362 0684 2
363 0685 2 ! Now we are going to set up two arrays, containing entries for each person
364 0686 2 ! we are recording. The pub_array contain the address of the person's PUB.
365 0687 2 ! The ctl_array contains the address of the oldest CTL on their CTL chain.
366 0688 2
367 0689 2 ch$fill(0,%allocation(pub_array),pub_array);
368 0690 2 ch$fill(0,%allocation(ctl_array),ctl_array);
369 0691 2 p = .phn$gq_pubhead[0];
370 0692 2 i = 0;
371 0693 2
372 0694 2 until .p eq la phn$gq_pubhead do (
373 0695 3     if not .p[pub_v_temporary] and
374 0696 3         not .p[pub_v_uhaveheld] and
375 0697 4         not .p[pub_v_hasuheld] then (
376 0698 4
377 0699 4         pub_array[.i] = .p;
378 0700 4         ctl_array[.i] = .p[pub_q_ctlhead1];
379 0701 4         inc(i);
380 0702 3     );
381 0703 3     p = .p[pub_l_flink];
382 0704 2 );
383 0705 2
384 0706 2 ! Now we go into our merging loop. On each pass, we find the oldest stamp -
385 0707 2 ! that will merge the conversation text in the most accurate order. For
386 0708 2 ! each line we print one line in the transcript. It contains the conversation
387 0709 2 ! text and also the speaker's name unless it's the same speaker as the
388 0710 2 ! previous line.
389 0711 2
390 0712 2 page_number = 0;
391 0713 2 line_counter = 0;
392 0714 2 cur_p = 0;
393 0715 2
394 0716 3 loop (
395 0717 3
```

```
396 0718 3 ! Loop through arrays and find minimum stamp. If there are no
397 0719 3 ! CTLs left to merge, we're done.
398 0720 3
399 0721 3 min_stamp = -1;
400 0722 3 incfu i from 0 to array_size-1 do
401 0723 4 if .ctl_array[.i] nequ 0 then (
402 0724 4 bind
403 0725 4 c = .ctl_array[.i]: ctl;
404 0726 4
405 0727 5 if .c[ctl_l_stamp] lssu .min_stamp then (
406 0728 5 min_stamp = .c[ctl_l_stamp];
407 0729 5 min_i = .i;
408 0730 4 );
409 0731 3 );
410 0732 3
411 0733 3 exitif (.min_stamp eqlu -1);
412 0734 3
413 0735 3 ! If this is a different person from the last line, then put the
414 0736 3 ! person's name on the beginning of the line. Otherwise just clear
415 0737 3 ! the beginning of the line.
416 0738 3
417 0739 4 if .pub_array[.min_i] neqa .cur_p then (
418 0740 4 cur_p = .pub_array[.min_i];
419 0741 4 transcribe(1,uplit byte(%char(ret)));
420 0742 5 begin
421 0743 5 bind
422 0744 5 new_tsb = cur_p[pub_b_tsb]: tsb,
423 0745 5 new_dsc = new_tsb[tsb_q_tkndsc,0]: descriptor;
424 0746 5
425 0747 5 ch$copy(.new_dsc[len],.new_dsc[ptr],' ',40,line_buf);
426 0748 4 end;
427 0749 3 ) else
428 0750 3 ch$fill(' ',40,line_buf);
429 0751 3
430 0752 3 ! Move the conversation text into the line.
431 0753 3
432 0754 3 c = .ctl_array[.min_i];
433 0755 4 begin
434 0756 4 bind
435 0757 4 line_dsc = c[ctl_q_line]: descriptor;
436 0758 4
437 0759 4 ch$move(.line_dsc[len],.line_dsc[ptr],line_buf+40);
438 0760 4 transcribe(40+.line_dsc[len],line_buf);
439 0761 3 end;
440 0762 3
441 0763 3 ! Now we move to the previous CTL on this person's chain. If
442 0764 3 ! there isn't one, then zero the ctl_array entry.
443 0765 3
444 0766 4 ctl_array[.min_i] = (if .c[ctl_l_blink] eqlu cur_p[pub_q_ctlhead0]
445 0767 4 then 0
446 0768 3 else .c[ctl_l_blink]);
447 0769 2 );
448 0770 2
449 0771 2 ! All done. Close the transcription file and refresh the screen.
450 0772 2
451 0773 2 $close(fab=trans fab);
452 0774 2 phn$fresh_screen(false);
```

: 453  
: 454  
: 455  
0775 2 return;  
0776 2  
0777 1 end;

```

.PSECT $SPLITS$,NOWRT,NOEXE,2
52 4F 43 45 52 20 57 4F 4E 20 4D 41 20 49 0D 000A1 P.AAQ: .ASCII <13>\I AM NOW RECORDING THE CONVERSATION\
52 45 56 4E 4F 43 20 45 48 54 20 47 4E 49 44 000B0
4F 50 20 53 49 48 54 20 4F 54 20 50 55 20 4E 000C4 .ASCII \N UP TO THIS POINT.\<13>
OD 2E 54 4E 49 000D3
00000037' 000D8 P.AAP: .LONG 55
00000000' 000DC .ADDRESS P.AAQ
OD 000E0 P.AAR: .ASCII <13>

.EXTRN SYSS$CREATE
.PSECT $CODE$,NOWRT,2
OFFC 00000
.ENTRY PHN$TRANSCRIBE_CMD, Save R2,R3,R4,R5,R6,R7,-; 0603
5E FF4C CE 9E 00002 MOVAB R8,R9,R10,R11
50 04 AC D0 00007 -180(SP), SP
0000' CF 60 90 0000B MOVL TRANS_SPEC, R0
0000' CF 04 A0 D0 00010 MOVAB (R0), TRANS_FAB+52
00000000G 00 0000' CF 9F 00016 MOVL 4(R0), TRANS_FAB+44
5B 01 FB 0001A PUSHAB TRANS_FAB
00010001 8F 50 D0 00021 CALLS #1, SYSS$CREATE
0000G CF 5B D1 00024 MOVL R0, STATUS
0000' CF 01 FB 0002F CMPL STATUS, #65537
0000' CF 04 00034 BEQL 1$
00000000G 00 0000' CF 9F 00035 1$: PUSHAB TRANS_RAB
5B 01 FB 00039 CALLS #1, SYSS$CONNECT
00010001 8F 50 D0 00040 MOVL R0, STATUS
0000G CF 5B D1 00043 CMPL STATUS, #65537
00000000G 00 0000' CF 9F 00053 BEQL 2$
0000G CF 01 FB 0004E CALLS #1, PHN$INFORM
00000000G 00 0000' CF 9F 00057 PUSHAB TRANS_FAB
0000G CF 01 FB 0005E CALLS #1, SYSS$CLOSE
18 00 0000G CF 01 FB 00063 2$: PUSHAB P.AAP
6E E8 AD 00068 CALLS #1, PHN$TALK
18 00 6E 00 2C 00068 MOVCS #0, (SP), #0, #24, PUB_ARRAY
DO AD 00074 MOVCS #0, (SP), #0, #24, CTL_ARRAY
51 0000G CF D0 00076 MOVL PHN$GQ_PUBHEAD, P
52 0000G CF 50 D4 0007B CLRL I
52 0000G CF 9E 0007D 3$: MOVAB PHN$GQ_PUBHEAD, R2
52 51 D1 00082 CMPL P, R2
52 23 13 00085 BEQL 5$
52 00F0 C1 9E 00087 MOVAB 240(P), R2

```

15	62	02	E0	0008C	BBS	#2, (R2), 4\$	:	
	12	62	E8	00090	BLBS	(R2), 4\$	:	0696
OE	62	01	E0	00093	BBS	#1, (R2), 4\$	:	0697
	E8 AD40	51	D0	00097	MOVL	P, PUB_ARRAY[I]	:	0699
	DO AD40	0108	C1	D0 0009C	MOVL	264(P), CTL_ARRAY[I]	:	0700
		50	D6	000A3	INCL	I	:	0701
	51	61	D0	000A5 4\$:	MOVL	(P), P	:	0703
		D3	11	000A8	BRB	3\$	:	0694
		0000'	CF	7C 000AA 5\$:	CLR	PAGE NUMBER	:	0712
		57	D4	000AE	CLRL	CUR_P	:	0714
	5A	01	CE	000B0 6\$:	MNEGL	#1, MIN_STAMP	:	0721
		51	D4	000B3	CLRL	I	:	0722
	50	DO AD41	D0	000B5 7\$:	MOVL	CTL_ARRAY[I], R0	:	0723
		0D	13	000BA	BEQL	8\$	:	
	5A	OC	A0	D1 000BC	CMPL	12(R0), MIN_STAMP	:	0727
		07	1E	000C0	BGEQU	8\$	:	
	5A	OC	A0	D0 000C2	MOVL	12(R0), MIN_STAMP	:	0728
	58	51	D0	000C6	MOVL	I, MIN_I	:	0729
		51	D6	000C9 8\$:	INCL	I	:	0723
	05	51	D1	000CB	CMPL	I, #5	:	
		E5	1B	000CE	BLEQU	7\$	:	
FFFFFFFF	8F	5A	D1	000D0	CMPL	MIN_STAMP, #-1	:	0733
		03	12	000D7	BNEQ	9\$	:	
		00A8	31	000D9	BRW	18\$	:	
	57	E8 AD48	D1	000DC 9\$:	CMPL	PUB_ARRAY[MIN_I], CUR_P	:	0739
		42	13	000E1	BEQL	12\$	:	
	57	E8 AD48	D0	000E3	MOVL	PUB_ARRAY[MIN_I], CUR_P	:	0740
	05	0000'	CF	F4 000E8	SOBGEQ	LINE_COUNTER, -10\$	:	0741
	FE58	CF	00	FB 000ED	CALLS	#0, PAGE HEADINGS	:	
	0000'	CF	01	B0 000F2 10\$:	MOVW	#1, TRANS_RAB+34	:	
	0000'	CF	0000'	CF 9E 000F7	MOVAB	P.AAR, TRANS_RAB+40	:	
		0000'	CF	9F 000FE	PUSHAB	TRANS_RAB	:	
00000000G	00	01	FB	00102	CALLS	#1, SY\$PUT	:	
	5B	50	D0	00109	MOVL	R0, STATUS	:	
	09	5B	E8	0010C	BLBS	STATUS, 11\$	:	
		5B	DD	0010F	PUSHL	STATUS	:	
00000000G	00	01	FB	00111	CALLS	#1, LIB\$SIGNAL	:	
	50	10	A7	9E 00118 11\$:	MOVAB	16(CUR_P), R0	:	0745
28	20	04	B0	60 2C 0011C	MOVCS	(R0), #4(R0), #32, #40, LINE_BUF	:	0747
			6E	00122			:	
			06	11 00123	BRB	13\$	:	0739
28	20	6E	00	2C 00125 12\$:	MOVCS	#0, (SP), #32, #40, LINE_BUF	:	0750
			6E	0012A			:	
	56	DO AD48	D0	0012B 13\$:	MOVL	CTL_ARRAY[MIN_I], C	:	0754
	59	10	A6	9E 00130	MOVAB	16(C), R9	:	0757
28	AE	04	B9	69 28 00134	MOVCS	(R9), #4(R9), LINE_BUF+40	:	0759
			05	0000'	CF	F4 0013A	:	0760
	FE06	CF	00	FB 0013F	SOBGEQ	LINE_COUNTER, 14\$	:	
0000'	CF	69	28	A1 00144 14\$:	CALLS	#0, PAGE HEADINGS	:	
	0000'	CF	0000'	CF 6E 9E 0014A	ADDW3	#40, (R9), TRANS_RAB+34	:	
		0000'	CF	9F 0014F	MOVAB	LINE_BUF, TRANS_RAB+40	:	
00000000G	00	01	FB	00153	PUSHAB	TRANS_RAB	:	
	5B	50	D0	0015A	CALLS	#1, SY\$PUT	:	
	09	5B	E8	0015D	MOVL	R0, STATUS	:	
		5B	DD	00160	BLBS	STATUS, 15\$	:	
00000000G	00	01	FB	00162	PUSHL	STATUS	:	
	50	0104	C7	9E 00169 15\$:	CALLS	#1, LIB\$SIGNAL	:	
					MOVAB	260(R7), R0	:	0766



FILECMDS  
V04-000

FILECMDS File-Oriented Phone Commands  
PHN\$TRANSCRIBE\_CMD - Handle TRANSCRIBE Command

H 1  
16-Sep-1984 02:09:16

VAX-11 Bliss-32 V4.0-742

Page 20

: Memory Used: 190 pages  
: Compilation Complete

IN  
VO

.....

0304 AH-BT13A-SE  
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION  
CONFIDENTIAL AND PROPRIETARY

The image displays a grid of 100 terminal windows, arranged in 10 rows and 10 columns. Each window shows a different screen from a VAX/VMS system. The screens contain various types of data, including:

- Text-based reports and lists.
- Tables with columns and rows of data.
- Diagrams and flowcharts.
- Large blocks of text, possibly logs or error messages.
- Small graphical elements like bar charts or status indicators.

Several windows have prominent titles or headers:

- PHONE
- PHONE MAP
- BASICMDS LIS
- PATSYM LIS
- PATSTO LIS
- PATVEC LIS
- PHONREQ REQ
- PATWRT LIS
- FILEMDS LIS

The overall appearance is that of a multi-user terminal session on a mainframe or minicomputer system, showing a variety of system utilities and data processing screens.

