

```

PPPPPPPPPPPP      AAAAAAAAAA      SSSSSSSSSSSS      RRRRRRRRRRRR      TTTTTTTTTTTTTT      LLL
PPPPPPPPPPPP      AAAAAAAAAA      SSSSSSSSSSSS      RRRRRRRRRRRR      TTTTTTTTTTTTTT      LLL
PPPPPPPPPPPP      AAAAAAAAAA      SSSSSSSSSSSS      RRRRRRRRRRRR      TTTTTTTTTTTTTT      LLL
PPP              PPP      AAA              AAA      SSS              RRR              RRR              TTT              LLL
PPP              PPP      AAA              AAA      SSS              RRR              RRR              TTT              LLL
PPP              PPP      AAA              AAA      SSS              RRR              RRR              TTT              LLL
PPP              PPP      AAA              AAA      SSS              RRR              RRR              TTT              LLL
PPP              PPP      AAA              AAA      SSS              RRR              RRR              TTT              LLL
PPP              PPP      AAA              AAA      SSS              RRR              RRR              TTT              LLL
PPPPPPPPPPPP      AAA              AAA              SSSSSSSSSS      RRRRRRRRRRRR      TTT              LLL
PPPPPPPPPPPP      AAA              AAA              SSSSSSSSSS      RRRRRRRRRRRR      TTT              LLL
PPPPPPPPPPPP      AAA              AAA              SSSSSSSSSS      RRRRRRRRRRRR      TTT              LLL
PPP              AAAAAAAAAAAAAAAAAA      SSS              RRR              RRR              TTT              LLL
PPP              AAAAAAAAAAAAAAAAAA      SSS              RRR              RRR              TTT              LLL
PPP              AAAAAAAAAAAAAAAAAA      SSS              RRR              RRR              TTT              LLL
PPP              AAA              AAA      SSS              RRR              RRR              TTT              LLL
PPP              AAA              AAA      SSS              RRR              RRR              TTT              LLL
PPP              AAA              AAA      SSS              RRR              RRR              TTT              LLL
PPP              AAA              AAA      SSS              RRR              RRR              TTT              LLL
PPP              AAA              AAA      SSSSSSSSSSSS      RRR              RRR              TTT              LLLLLLLLLLLLLLLLLL
PPP              AAA              AAA      SSSSSSSSSSSS      RRR              RRR              TTT              LLLLLLLLLLLLLLLLLL
PPP              AAA              AAA      SSSSSSSSSSSS      RRR              RRR              TTT              LLLLLLLLLLLLLLLLLL

```

2

Sym

PAS

PAS
PASPAS
PAS

PAS

PAS
PASPAS
PAS

PAS

PAS
PASPAS
PAS

PAS

PAS
PAS

PAS

PAS
PASPAS
PAS

PAS

PAS
PASPAS
PAS

PAS

PAS
PAS

PAS
PAS

PAS

PAS

PAS

PAS

PAC

PAS

PAS

PAS
PAS

PAS

PAS

PAS

PAS

PAS
PASPAS
PAS

PAS

PAS
PAS

PAS


```
PPPPPPPP      AAAAAA      SSSSSSSS      PPPPPPPP      AAAAAA      GGGGGGGG      EEEEEEEEEEE      222222
PPPPPPPP      AAAAAA      SSSSSSSS      PPPPPPPP      AAAAAA      GGGGGGGG      EEEEEEEEEEE      222222
PP      PP      AA      AA      SS      PP      PP      AA      AA      GG      EE      22      22
PP      PP      AA      AA      SS      PP      PP      AA      AA      GG      EE      22      22
PP      PP      AA      AA      SS      PP      PP      AA      AA      GG      EE      22      22
PP      PP      AA      AA      SS      PP      PP      AA      AA      GG      EE      22      22
PPPPPPPP      AA      AA      SSSSSS      PPPPPPPP      AA      AA      GG      EEEEEEEEE      22
PPPPPPPP      AA      AA      SSSSSS      PPPPPPPP      AA      AA      GG      EEEEEEEEE      22
PP      AAAAAAAAAA      SS      PP      AAAAAAAAAA      GG      GGGGGG      EE      22
PP      AAAAAAAAAA      SS      PP      AAAAAAAAAA      GG      GGGGGG      EE      22
PP      AA      AA      SS      PP      AA      AA      GG      GG      EE      22
PP      AA      AA      SSSSSSSS      PP      AA      AA      GG      GG      EE      22
PP      AA      AA      SSSSSSSS      PP      AA      AA      GGGGGG      EEEEEEEEEEE      2222222222
PP      AA      AA      SSSSSSSS      PP      AA      AA      GGGGGG      EEEEEEEEEEE      2222222222
                                     ....
                                     ....
                                     ....
                                     ....

LL      IIIIII      SSSSSSSS
LL      IIIIII      SSSSSSSS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SSSSSS
LL      II      SSSSSS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SS
LLLLLLLLLLLL      IIIIII      SSSSSSSS
LLLLLLLLLLLL      IIIIII      SSSSSSSS
```



```
1 0001 0 MODULE PASSPAGE2 ( %TITLE 'PAGE procedure'
2 0002 0 IDENT = '1-002' ! File: PASSPAGE2.B32 Edit: SBL1002
3 0003 0 ) =
4 0004 1 BEGIN
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: Pascal Language Support
32 0032 1
33 0033 1 ABSTRACT:
34 0034 1
35 0035 1 This module implements the Pascal PAGE procedure.
36 0036 1
37 0037 1 ENVIRONMENT: User mode - AST reentrant
38 0038 1
39 0039 1 AUTHOR: Steven B. Lionel, CREATION DATE: 1-April-1981
40 0040 1
41 0041 1 MODIFIED BY:
42 0042 1
43 0043 1 1-001 - Original. SBL 1-April-1981
44 0044 1 1-002 - Correct argument count check. SBL 21-Feb-1983
45 0045 1 --
46 0046 1
```


PASSPAGE2
1-002

PAGE procedure
Declarations

D 16
16-Sep-1984 01:50:54
14-Sep-1984 12:51:41

VAX-11 Bliss-32 V4.0-742
[PASRTL.SRC]PASPAGE2.B32;1

Page 2
(2)

```
: 48      0047 1 %SBTTL 'Declarations'
: 49      0048 1
: 50      0049 1 : PROLOGUE DEFINITIONS:
: 51      0050 1 :
: 52      0051 1
: 53      0052 1 REQUIRE 'RTLIN:PASPROLOG';           ! Externals, linkages, PSECTs, structures
: 54      0116 1
: 55      0117 1 :
: 56      0118 1 : TABLE OF CONTENTS:
: 57      0119 1 :
: 58      0120 1
: 59      0121 1 FORWARD ROUTINE
: 60      0122 1 PASSPAGE2: NOVALUE;                 ! Skip to top of page
: 61      0123 1
: 62      0124 1 :
: 63      0125 1 : MACROS:
: 64      0126 1 :
: 65      0127 1 :     NONE
: 66      0128 1 :
: 67      0129 1 : EQUATED SYMBOLS:
: 68      0130 1 :
: 69      0131 1 :     NONE
: 70      0132 1 :
: 71      0133 1 : FIELDS:
: 72      0134 1 :
: 73      0135 1 :     NONE
: 74      0136 1 :
: 75      0137 1 : OWN STORAGE:
: 76      0138 1 :
: 77      0139 1 :     NONE
: 78      0140 1 :
```



```

80 0141 1 %SBTTL 'PAS$PAGE2 - Skip to top of page'
81 0142 1 GLOBAL ROUTINE PAS$PAGE2 (
82 0143 1     PFV: REF $PAS$PFV_FILE_VARIABLE,
83 0144 1     ERROR
84 0145 1     ): NOVALUE =
85 0146 1
86 0147 1 ++
87 0148 1 FUNCTIONAL DESCRIPTION:
88 0149 1
89 0150 1     This procedure implements the Pascal PAGE procedure for textfiles.
90 0151 1     It flushes the current buffer (if any), causes a 'page eject' in
91 0152 1     a manner appropriate for the carriage control, and begins a new line.
92 0153 1
93 0154 1 CALLING SEQUENCE:
94 0155 1
95 0156 1     CALL PAS$PAGE2 (PFV.mr.r [, ERROR.j.r])
96 0157 1
97 0158 1 FORMAL PARAMETERS:
98 0159 1
99 0160 1     PFV
100 0161 1         - The Pascal File Variable (PFV) passed by reference.
101 0162 1         The structure of the PFV is defined in PAS$PFV.REQ.
102 0163 1
103 0164 1     ERROR
104 0165 1         - Optional. If specified, the address to unwind to
105 0166 1         in case of an error.
106 0167 1
107 0168 1 IMPLICIT INPUTS:
108 0169 1
109 0170 1     NONE
110 0171 1
111 0172 1 IMPLICIT OUTPUTS:
112 0173 1
113 0174 1     NONE
114 0175 1
115 0176 1 ROUTINE VALUE:
116 0177 1
117 0178 1     NONE
118 0179 1
119 0180 1 SIDE EFFECTS:
120 0181 1
121 0182 1     If the file is the standard file INPUT or OUTPUT, it is implicitly opened.
122 0183 1
123 0184 1 SIGNALLED ERRORS:
124 0185 1
125 0186 1     LINTOOLON - line too long
126 0187 1
127 0188 2 --
128 0189 2 BEGIN
129 0190 2
130 0191 2     LOCAL
131 0192 2         FCB: REF $PAS$FCB_CONTROL_BLOCK, ! File Control block
132 0193 2         PFV_ADDR: VOLATILE, ! Enable argument
133 0194 2         UNWIND_ACT: VOLATILE, ! Enable argument
134 0195 2         ERROR_ADDR: VOLATILE; ! Enable argument
135 0196 2
136 0197 2     BIND
137 0198 2         RAB = FCB: REF BLOCK [, BYTE]; ! RAB is FCB
```



```

: 137 0198 2
: 138 0199 2 BUILTIN
: 139 0200 2 ACTUALCOUNT; ! Count of arguments
: 140 0201 2
: 141 0202 2 ENABLE
: 142 0203 2 PASS$IO_HANDLER (PFV_ADDR, UNWIND_ACT, ERROR_ADDR); ! Enable error handler
: 143 0204 2
: 144 0205 2 !+
: 145 0206 2 ! Get ERROR parameter, if present.
: 146 0207 2 !-
: 147 0208 2
: 148 0209 2 IF ACTUALCOUNT () GEQU 2
: 149 0210 2 THEN
: 150 0211 2 ERROR_ADDR = .ERROR; ! Set unwind address
: 151 0212 2
: 152 0213 2 PFV_ADDR = PFV [PFV$R_PFV]; ! Set PFV address
: 153 0214 2
: 154 0215 2 !+
: 155 0216 2 ! Validate PFV and get PFV.
: 156 0217 2 !-
: 157 0218 2
: 158 0219 2 PASS$VALIDATE_PFV (PFV [PFV$R_PFV]; FCB);
: 159 0220 2
: 160 0221 2 !+
: 161 0222 2 ! Set unwind action to unlock file.
: 162 0223 2 !-
: 163 0224 2
: 164 0225 2 UNWIND_ACT = PASS$K_UNWIND_UNLOCK;
: 165 0226 2
: 166 0227 2 !+
: 167 0228 2 ! Do common initialization.
: 168 0229 2 !-
: 169 0230 2
: 170 0231 2 PASS$INIT_WRITE (PFV [PFV$R_PFV], FCB [FCB$R_FCB]; FCB);
: 171 0232 2
: 172 0233 2 !+
: 173 0234 2 ! If the buffer has a partial line, flush it.
: 174 0235 2 !-
: 175 0236 2
: 176 0237 2 IF .FCB [FCB$A_RECORD_CUR] GTRA .FCB [FCB$A_RECORD_BEG]
: 177 0238 2 THEN
: 178 0239 2 PASS$WRITELN (PFV [PFV$R_PFV], FCB [FCB$R_FCB]);
: 179 0240 2
: 180 0241 2 !+
: 181 0242 2 ! If buffer length is zero, signal an error because there is no
: 182 0243 2 ! room to put the page-eject character.
: 183 0244 2 !-
: 184 0245 2
: 185 0246 2 IF .FCB [FCB$L_RECORD_LEN] EQL 0
: 186 0247 2 THEN
: 187 0248 2 $PASS$IO_ERROR (PASS$LINTOOLON,1,1);
: 188 0249 2
: 189 0250 2 !+
: 190 0251 2 ! If the carriagecontrol attribute is FTN, move a '1' to the first
: 191 0252 2 ! byte of the buffer, otherwise an <FF>.
: 192 0253 2 !-
: 193 0254 2

```



```
194 0255 3 BEGIN
195 0256 LOCAL
196 0257 FAB: REF BLOCK [, BYTE];
197 0258 FAB = .RAB [RAB$L_FAB]; ! Get FAB address
198 0259 IF .FAB [FAB$V_FTN]
199 0260 THEN
200 0261 CH$WCHAR_A (%C'1', FCB [FCB$A_RECORD_CUR])
201 0262 ELSE
202 0263 CH$WCHAR_A (%X'0C', FCB [FCB$A_RECORD_CUR]);
203 0264 END;
204 0265
205 0266 !+
206 0267 ! Do a WRITELN to write the record.
207 0268 !-
208 0269
209 0270 PASS$WRITELN (PFV [PFV$R_PFV], FCB [FCB$R_FCB]);
210 0271
211 0272 !+
212 0273 ! Undefine the file buffer.
213 0274 ! Indicate successful completion
214 0275 ! Unlock the file variable.
215 0276 !-
216 0277
217 0278 FCB [FCB$L_STATUS] = 0;
218 0279 PFV [PFV$V_DFB] = 0;
219 0280 PFV [PFV$V_LOCK] = 0;
220 0281
221 0282 RETURN;
222 0283
223 0284 1 END;
```

! End of routine PASSPAGE2

```
.TITLE PASSPAGE2 PAGE procedure
.IDENT \1-002\
```

```
.EXTRN PASSPAGE2, PASS$IO_HANDLER
.EXTRN PASS$VALIDATE_PFV
.EXTRN PASS$INIT_WRITE
.EXTRN PASS$WRITELN, PASS$SIGNAL
.EXTRN PASSK_LINTOOLON
```

```
.PSECT _PASSCODE, NOWRT, SHR, PIC, 2
```

			03FC 00000	.ENTRY PASSPAGE2, Save R2,R3,R4,R5,R6,R7,R8,R9	: 0142
59	00000000G	00	9E 00002	MOVAB PASS\$WRITELN, R9	
5E		08	C2 00009	SUBL2 #8, SP	
		7E	D4 0000C	CLRL ERROR_ADDR	: 0188
	04	AE	7C 0000E	CLRQ UNWIND_ACT	
6D	0063	CF	DE 00011	MOVAL 6\$, (FP)	
02		6C	91 00016	CMPB (AP), #2	: 0209
		04	1F 00019	BLSSU 1\$	
6E	08	AC	D0 0001B	MOVL ERROR, ERROR_ADDR	: 0211
56	04	AC	D0 0001F	MOVL PFV, R6	: 0213
08	AE	56	D0 00023	MOVL R6, PFV_ADDR	
	00000000G	00	16 00027	JSB PASS\$VALIDATE_PFV	: 0219
04	AE	01	D0 0002D	MOVL #1, UNWIND_ACT	: 0225
	00000000G	00	16 00031	JSB PASS\$INIT_WRITE	: 0231

PASSPAGE2
1-002

PAGE procedure
PASSPAGE2 - Skip to top of page

H 16
16-Sep-1984 01:50:54
14-Sep-1984 12:51:41

VAX-11 Bliss-32 V4.0-742
[PASRTL.SRC]PASSPAGE2.B32;1

Page 6
(3)

	58	EC	A7	9E	00037	MOVAB	-20(FCB), R8	:	0237
E8	A7		68	D1	0003B	CMPL	(R8), -24(FCB)	:	
			02	1B	0003F	BLEQU	2\$	:	
			69	16	00041	JSB	PASS\$WRITELN	:	0239
		F4	A7	D5	00043	TSTL	-12(FCB)	:	0246
			10	12	00046	BNEQ	3\$	:	
			01	DD	00048	PUSHL	#1	:	0248
			01	DD	0004A	PUSHL	#1	:	
	7E	00G	8F	9A	0004C	MOVZBL	#PASS\$ LINTOOLON, -(SP)	:	
00000000G	00		03	FB	00050	CALLS	#3, PASS\$\$SIGNAL	:	
				04	00057	RET		:	
	50	3C	A7	D0	00058	MOVL	60(FCB), FAB	:	0258
	06	1E	A0	E9	0005C	BLBC	30(FAB), 4\$	:	0259
00	B8		31	90	00060	MOVB	#49, a0(R8)	:	0261
			04	11	00064	BRB	5\$	:	
00	B8		0C	90	00066	MOVB	#12, a0(R8)	:	0263
			68	D6	0006A	INCL	(R8)	:	
			69	16	0006C	JSB	PASS\$WRITELN	:	0270
		D4	A7	D4	0006E	CLRL	-44(FCB)	:	0278
06	A6	8002	8F	AA	00071	BICW2	#32770, 6(R6)	:	0280
				04	00077	RET		:	0284
				0000	00078	.WORD	Save nothing	:	0188
	50	08	AC	D0	0007A	MOVL	8(AP), R0	:	
	50	04	A0	D0	0007E	MOVL	4(R0), R0	:	
		F4	A0	9F	00082	PUSHAB	ERROR_ADDR	:	
		F8	A0	9F	00085	PUSHAB	UNWIND_ACT	:	
		FC	A0	9F	00088	PUSHAB	PFV_ADDR	:	
			03	DD	0008B	PUSHL	#3	:	
			5E	DD	0008D	PUSHL	SP	:	
	7E	04	AC	7D	0008F	MOVQ	4(AP), -(SP)	:	
00000000G	00		03	FB	00093	CALLS	#3, PASS\$\$IO_HANDLER	:	
				04	0009A	RET		:	

; Routine Size: 155 bytes, Routine Base: _PASS\$CODE + 0000

: 224 0285 1
: 225 0286 1 !<BLF/PAGE>

PASSPAGE2
1-002

PAGE procedure
PASSPAGE2 - Skip to top of page

I 16
16-Sep-1984 01:50:54 VAX-11 Bliss-32 V4.0-742
14-Sep-1984 12:51:41 [PASRTL.SRC]PASSPAGE2.B32;1

Page 7
(4)

: 227 0287 1 END
: 228 0288 1
: 229 0289 0 ELUDOM

! End of module PASSPAGE2

PSECT SUMMARY

:
: Name Bytes Attributes
: _PASSCODE 155 NOVEC,NOWRT, RD , EXE, SHR, LCL, REL, CON, PIC,ALIGN(2)

Library Statistics

:
: File ----- Symbols ----- Pages Processing
: Total Loaded Percent Mapped Time
: \$255\$DUA28:[SYSLIB]STARLET.L32;1 9776 2 0 581 00:01.0
: \$255\$DUA28:[PASRTL.OBJ]PASLIB.L32;1 427 92 21 33 00:00.4

COMMAND QUALIFIERS

:
: BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/NOTRACE/LIS=LIS\$:PASSPAGE2/OBJ=OBJ\$:PASSPAGE2 MSRC\$:PASSPAGE2/UPDATE=(ENH\$:PASSPAGE2)

: 230 0290 0
: Size: 155 code + 0 data bytes
: Run Time: 00:05.7
: Elapsed Time: 00:14.7
: Lines/CPU Min: 3047
: Lexemes/CPU-Min: 12451
: Memory Used: 84 pages
: Compilation Complete

0295 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY