

```

PPPPPPPPPPPP      AAAAAAAAAA      SSSSSSSSSSSS      RRRRRRRRRRRR      TTTTTTTTTTTTTT      LLL
PPPPPPPPPPPPPP    AAAAAAAAAA      SSSSSSSSSSSS      RRRRRRRRRRRR      TTTTTTTTTTTTTT      LLL
PPPPPPPPPPPPPP    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
PPPPPPPPPPPPPP    AAA          AAA          SSSSSSSSSS      RRRRRRRRRRRR      TTT              LLL
PPPPPPPPPPPPPP    AAA          AAA          SSSSSSSSSS      RRRRRRRRRRRR      TTT              LLL
PPPPPPPPPPPPPP    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
PAS

PAS
PAS

PAS

PAS
PASPAS
PAS

PAS

PAS
PAS

PAS

PAS
PAS

PAS
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  AAAA  SSSSSSSS  RRRRRRRR  EEEEEEEEE  AAAA  IIIIII  NN  NN  TTTTTTTTTT
PPPPPPPP  AAAA  SSSSSSSS  RRRRRRRR  EEEEEEEEE  AAAA  IIIIII  NN  NN  TTTTTTTTTT
PP  PP  AA  AA  SS  RR  RR  EE  AA  AA  II  NN  NN  TT
PP  PP  AA  AA  SS  RR  RR  EE  AA  AA  II  NN  NN  TT
PP  PP  AA  AA  SS  RR  RR  EE  AA  AA  II  NN  NN  TT
PPPPPPPP  AA  AA  SSSSSS  RRRRRRRR  EEEEEEE  AA  AA  II  NN  NN  TT
PPPPPPPP  AA  AA  SSSSSS  RRRRRRRR  EEEEEEE  AA  AA  II  NN  NN  TT
PP  AA  AA  SS  RR  RR  EE  AA  AA  AA  II  NN  NN  TT
PP  AA  AA  SS  RR  RR  EE  AA  AA  AA  II  NN  NN  TT
PP  AA  AA  SS  RR  RR  EE  AA  AA  AA  II  NN  NN  TT
PP  AA  AA  SSSSSSSS  RR  RR  EE  AA  AA  IIIIII  NN  NN  TT
PP  AA  AA  SSSSSSSS  RR  RR  EE  AA  AA  IIIIII  NN  NN  TT
```

....
....
....
....

```
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
LLLLLLLLLL  IIIIII  SSSSSSSS
LLLLLLLLLL  IIIIII  SSSSSSSS
```



```
1 0001 0 MODULE PASSREAD_INTEGER ( %TITLE 'Read an integer'
2 0002 0 IDENT = '1-002' ! File: PASREAIINT.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 contains procedures which read an integer
36 0036 1 from a textfile or a string.
37 0037 1
38 0038 1 ENVIRONMENT: User mode - AST reentrant
39 0039 1
40 0040 1 AUTHOR: Steven B. Lionel, CREATION DATE: 1-April-1981
41 0041 1
42 0042 1 MODIFIED BY:
43 0043 1
44 0044 1 1-001 - Original. SBL 1-April-1981
45 0045 1 1-002 - Use PASS$END_READ. SBL 24-May-1982
46 0046 1 --
47 0047 1
```

```

: 49      0048 1 %SBTTL 'Declarations'
: 50      0049 1
: 51      0050 1 PROLOGUE DEFINITIONS:
: 52      0051 1
: 53      0052 1
: 54      0053 1 REQUIRE 'RTLIN:PASPROLOG';
: 55      0117 1
: 56      0118 1
: 57      0119 1 TABLE OF CONTENTS:
: 58      0120 1
: 59      0121 1
: 60      0122 1 FORWARD ROUTINE
: 61      0123 1     PASSREAD_INTEGER,
: 62      0124 1     PASSREADV_INTEGER;
: 63      0125 1
: 64      0126 1
: 65      0127 1 MACROS:
: 66      0128 1
: 67      0129 1     NONE
: 68      0130 1
: 69      0131 1 EQUATED SYMBOLS:
: 70      0132 1
: 71      0133 1     NONE
: 72      0134 1
: 73      0135 1 FIELDS:
: 74      0136 1
: 75      0137 1     NONE
: 76      0138 1
: 77      0139 1 OWN STORAGE:
: 78      0140 1
: 79      0141 1     NONE
: 80      0142 1

```

! Externals, linkages, PSECTs, structures

! Read from textfile
! Read from string


```

82 0143 1 %SBTTL 'PASS$READ_INTEGER - Read integer from textfile'
83 0144 1 GLOBAL ROUTINE PASS$READ_INTEGER (
84 0145 1   PFV: REF $PASS$PFV_FILE_VARIABLE,      ! File variable
85 0146 1   ERROR                                ! Error unwind address
86 0147 1   ) =
87 0148 1
88 0149 1 ++
89 0150 1 FUNCTIONAL DESCRIPTION:
90 0151 1
91 0152 1   This function reads an integer from the specified textfile
92 0153 1   and returns it as the function value.
93 0154 1
94 0155 1 CALLING SEQUENCE:
95 0156 1
96 0157 1   Integer.wl.v = PASS$READ_INTEGER (PFV.mr.r [, ERROR.ja.r])
97 0158 1
98 0159 1 FORMAL PARAMETERS:
99 0160 1
100 0161 1   PFV          - The Pascal File Variable (PFV) passed by reference.
101 0162 1                   The structure of the PFV is defined in PASPFV.REQ.
102 0163 1
103 0164 1   ERROR       - Optional. If specified, the address to unwind to
104 0165 1                   in case of an error.
105 0166 1
106 0167 1 IMPLICIT INPUTS:
107 0168 1
108 0169 1   NONE
109 0170 1
110 0171 1 IMPLICIT OUTPUTS:
111 0172 1
112 0173 1   NONE
113 0174 1
114 0175 1 ROUTINE VALUE:
115 0176 1
116 0177 1   The value of the integer read.
117 0178 1
118 0179 1 SIDE EFFECTS:
119 0180 1
120 0181 1   If the file is the standard file INPUT or OUTPUT, it is implicitly opened.
121 0182 1
122 0183 1 SIGNALLED ERRORS:
123 0184 1
124 0185 1   INVSYNINT - invalid syntax for integer value
125 0186 1   NOTVALTYP - "string" is not a value of type "type"
126 0187 1
127 0188 1 --
128 0189 1
129 0190 2 BEGIN
130 0191 2
131 0192 2 LOCAL
132 0193 2   RESULT,          ! Result value
133 0194 2   DESCR: BLOCK [8, BYTE], ! Descriptor for convert
134 0195 2   FCB: REF $PASS$FCB_CONTROL_BLOCK, ! File Control block
135 0196 2   PFV_ADDR: VOLATILE, ! Enable argument
136 0197 2   UNWIND_ACT: VOLATILE, ! Enable argument
137 0198 2   ERROR_ADDR: VOLATILE; ! Enable argument
138 0199 2

```



```

139      0200 2    BUILTIN
140      0201 2    ACTUALCOUNT;                ! Count of arguments
141      0202 2
142      0203 2    ENABLE
143      0204 2    PASS$IO_HANDLER (PFV_ADDR, UNWIND_ACT, ERROR_ADDR);    ! Enable error handler
144      0205 2
145      0206 2    !+
146      0207 2    !- Get ERROR parameter, if present.
147      0208 2
148      0209 2
149      0210 2    IF ACTUALCOUNT () GEQU 2
150      0211 2    THEN
151      0212 2    ERROR_ADDR = .ERROR;            ! Set unwind address
152      0213 2
153      0214 2    PFV_ADDR = PFV [PFV$R_PFV];      ! Set PFV address
154      0215 2
155      0216 2    !+
156      0217 2    !- Validate PFV and get PFV.
157      0218 2
158      0219 2
159      0220 2    PASS$VALIDATE_PFV (PFV [PFV$R_PFV]; FCB);
160      0221 2
161      0222 2    !+
162      0223 2    !- Set unwind action to unlock file.
163      0224 2
164      0225 2
165      0226 2    UNWIND_ACT = PASS$K_UNWIND_UNLOCK;
166      0227 2
167      0228 2    !+
168      0229 2    !- Do common initialization.
169      0230 2
170      0231 2
171      0232 2    PASS$INIT_READ (PFV [PFV$R_PFV], FCB [FCB$R_FCB]; FCB);
172      0233 2
173      0234 2    RESULT = 0;
174      0235 2
175      0236 2    !+
176      0237 2    !- Set up string descriptor for convert call.
177      0238 2
178      0239 2
179      0240 2    DESCR [DSC$B_CLASS] = DSC$K_CLASS_S;
180      0241 2    DESCR [DSC$B_DTYPE] = DSC$K_DTYPE_T;
181      0242 2
182      0243 2    !+
183      0244 2    !- Call utility routine to find a string that looks like an integer.
184      0245 2    !- If we can't find one, signal an error.
185      0246 2
186      0247 2
187      0248 2    IF NOT PASS$GET_INTEGER (PFV [PFV$R_PFV], FCB [FCB$R_FCB];
188      0249 2    DESCR [DSC$A_POINTER], DESCR [DSC$W_LENGTH], FCB)
189      0250 2    THEN
190      0251 2    $PASS$IO_ERROR (PASS$INVSYNINT, 2, DESCR,
191      0252 2    .FCB [FCB$L_RECORD_NUMBER]); ! Invalid syntax for integer
192      0253 2
193      0254 2    !+
194      0255 2    !- Call convert routine. If it fails, signal an error.
195      0256 2

```

```

: 196      0257 2
: 197      0258 2
: 198      0259 2
: 199      P 0260 2
: 200      P 0261 2
: 201      0262 2
: 202      0263 2
: 203      0264 2
: 204      0265 2
: 205      0266 2
: 206      0267 2
: 207      0268 2
: 208      0269 2
: 209      0270 2
: 210      0271 2
: 211      0272 1

IF NOT OTSSCVT_TI_L (DESCR, RESULT)
THEN
    SPASSIO ERROR (PASS NOTVALTYP,3,DESCR,
        UPLIT BYTE (%CHARCOUNT('INTEGER'),'INTEGER'),
        .FCB [FCB$L_RECORD_NUMBER]); ! Not a value

!+
!- Do end-of-READ processing.

PASS$END_READ (PFV [PFV$R_PFV], FCB [FCB$R_FCB]);

RETURN .RESULT;

END;

```

! End of routine PASS\$READ_INTEGER

```

.TITLE PASS$READ_INTEGER Read an integer
.IDENT \1-002\

.PSECT _PASS$CODE,NOWRT, SHR, PIC,2

52 45 47 45 54 4E 07 0000 P.AAA: .BYTE 7
49 00001 .ASCII \INTEGER\

.EXTRN PASS$READ_INTEGER
.EXTRN PASS$READ_INTEGER
.EXTRN PASS$IO_HANDLER
.EXTRN PASS$VALIDATE_PFV
.EXTRN PASS$INIT_READ, PASS$GET_INTEGER
.EXTRN PASS$SIGNAL, PASS$INVSYNINT
.EXTRN OTSSCVT_TI_L, PASS$NOTVALTYP
.EXTRN PASS$END_READ

.ENTRY PASS$READ_INTEGER, Save R2,R3,R4,R5,R6,R7,-
R8,R9
MOVAB PASS$SIGNAL, R9
SUBL2 #24, SP
CLRQ ERROR_ADDR
CLRL PFV_ADDR
MOVAL 5$,-(FP)
CMPB (AP), #2
BLSSU 1$
MOVL ERROR, ERROR_ADDR
MOVL PFV, R6
MOVL R6, PFV_ADDR
JSB PASS$VALIDATE_PFV
MOVL #1, UNWIND_ACT
JSB PASS$INIT_READ
CLRL RESULT
MOVW #270, DESCR+2
JSB PASS$GET_INTEGER
MOVL R4, DESCR+4
MOVW R5, DESCR
BLBS R0, 2$
PUSHL -56(FCB)

03FC 00000
59 00000000G 00 9E 00002
5E 04 AE 7C 0000C
0C 0C AE D4 0000F
6D 007D CF DE 00012
02 6C 91 00017
04 AE 08 AC D0 0001C
56 04 AC D0 00021 1$:
0C AE 56 D0 00025
08 AE 00000000G 00 16 00029
AE 01 D0 0002F
00000000G 00 16 00033
6E D4 00039
12 AE 010E 8F B0 0003B
00000000G 00 16 00041
14 AE 54 D0 00047
10 AE 55 B0 0004B
11 5G E8 0004F
C8 A7 DD 00052

```


PASSREAD_INTEGE Read an integer
1-002

PASSREAD_INTEGER - Read integer from textfile

F 8
16-Sep-1984 01:57:44
14-Sep-1984 12:51:47

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

Page 6
(3)

		14	AE	9F	00055	PUSHAB	DESCR		
			02	DD	00058	PUSHL	#2		
7E		00G	8F	9A	0005A	MOVZBL	#PASSK_INVSYNINT, -(SP)		
69			04	FB	0005E	CALLS	#4, PASS\$\$SIGNAL		
			2D	11	00061	BRB	4\$		
			5E	DD	00063	PUSHL	SP		0258
		14	AE	9F	00065	PUSHAB	DESCR		
00000000G	00		02	FB	00068	CALLS	#2, OTSS\$CVT_TI_L		
	14		50	E8	0006F	BLBS	R0, 3\$		
		C8	A7	DD	00072	PUSHL	-56(FCB)		0262
		80	AF	9F	00075	PUSHAB	P.AAA		
		18	AE	9F	00078	PUSHAB	DESCR		
			03	DD	0007B	PUSHL	#3		
7E		00G	8F	9A	0007D	MOVZBL	#PASSK_NOTVALTYP, -(SP)		
69			05	FB	00081	CALLS	#5, PASS\$\$SIGNAL		
			0A	11	00084	BRB	4\$		
		00000000G	00	16	00086	JSB	PASS\$END_READ		0268
			6E	D0	0008C	MOVL	RESULT, R0		0270
				04	0008F	RET			
			50	D4	00090	CLRL	R0		0272
				04	00092	RET			
				0000	00093	.WORD	Save nothing		0190
						MOVL	8(AP), R0		
50		08	AC	D0	00095	MOVL	4(R0), R0		
		04	A0	D0	00099	PUSHAB	ERROR_ADDR		
		EC	A0	9F	0009D	PUSHAB	UNWIND_ACT		
		F0	A0	9F	000A0	PUSHAB	PFV_ADDR		
		F4	A0	9F	000A3	PUSHL	#3		
			03	DD	000A6	PUSHL	SP		
			5E	DD	000A8	PUSHL	SP		
		04	AC	7D	000AA	MOVQ	4(AP), -(SP)		
00000000G	7E		03	FB	000AE	CALLS	#3, PASS\$\$IO_HANDLER		
	00			04	000B5	RET			

; Routine Size: 182 bytes, Routine Base: _PASS\$CODE + 0008

: 212 0273 1
: 213 0274 1 !<BLF/PAGE>


```
: 215      0275 1 %SBTTL 'PASSREADV_INTEGER - Read integer from string'
: 216      0276 1 GLOBAL ROUTINE PASSREADV_INTEGER (
: 217      0277 1     STRING: REF BLOCK [, BYTE],           ! String descriptor
: 218      0278 1     ERROR                                ! Error unwind address
: 219      0279 1     ) =
: 220      0280 1
: 221      0281 1 !++
: 222      0282 1 !FUNCTIONAL DESCRIPTION:
: 223      0283 1
: 224      0284 1     This function reads an integer from the specified string
: 225      0285 1     and returns it as the function value.
: 226      0286 1
: 227      0287 1 !CALLING SEQUENCE:
: 228      0288 1
: 229      0289 1     Integer.wl.v = PASSREADV_INTEGER (STRING.mt.ds [, ERROR.ja.r])
: 230      0290 1
: 231      0291 1 !FORMAL PARAMETERS:
: 232      0292 1
: 233      0293 1     STRING                                - The string to read from, passed as a class S
: 234      0294 1     (assumed) descriptor. The length and pointer
: 235      0295 1     are updated to reflect the unread string.
: 236      0296 1
: 237      0297 1     ERROR                                - Optional. If specified, the address to unwind to
: 238      0298 1     in case of an error.
: 239      0299 1
: 240      0300 1 !IMPLICIT INPUTS:
: 241      0301 1
: 242      0302 1     NONE
: 243      0303 1
: 244      0304 1 !IMPLICIT OUTPUTS:
: 245      0305 1
: 246      0306 1     NONE
: 247      0307 1
: 248      0308 1 !ROUTINE VALUE:
: 249      0309 1
: 250      0310 1     The value of the integer read.
: 251      0311 1
: 252      0312 1 !SIDE EFFECTS:
: 253      0313 1
: 254      0314 1     NONE
: 255      0315 1
: 256      0316 1 !SIGNALLED ERRORS:
: 257      0317 1
: 258      0318 1     See PASSREAD_INTEGER
: 259      0319 1
: 260      0320 1 !--
: 261      0321 1
: 262      0322 2 BEGIN
: 263      0323 2
: 264      0324 2 LOCAL
: 265      0325 2     PFV: $PASSPFV_FILE_VARIABLE,           ! Pascal File Variable
: 266      0326 2     RESULT,                                ! Result value
: 267      0327 2     ARG_LIST: VECTOR [2, LONG],            ! Argument list
: 268      0328 2     PFV_ADDR: VOLATILE,                    ! Enable argument
: 269      0329 2     UNWIND_ACT: VOLATILE,                  ! Enable argument
: 270      0330 2     ERROR_ADDR: VOLATILE;                 ! Enable argument
: 271      0331 2
```

```

272 0332 2 BUILTIN
273 0333 2 ACTUALCOUNT; ! Count of arguments
274 0334 2
275 0335 2
276 0336 2 ENABLE
277 0337 2 PASS$IO_HANDLER (PFV_ADDR, UNWIND_ACT, ERROR_ADDR); ! Enable error handler
278 0338 2
279 0339 2 !+
280 0340 2 ! Get ERROR parameter, if present.
281 0341 2 !-
282 0342 2 IF ACTUALCOUNT () GEQU 2
283 0343 2 THEN
284 0344 2 ERROR_ADDR = .ERROR; ! Set unwind address
285 0345 2
286 0346 2 PFV_ADDR = PFV [PFV$R_PFV]; ! Set PFV address
287 0347 2
288 0348 2 !+
289 0349 2 ! Set up ARG_LIST.
290 0350 2 !-
291 0351 2
292 0352 2 ARG_LIST [0] = 1; ! One argument
293 0353 2 ARG_LIST [1] = PFV [PFV$R_PFV]; ! PFV address
294 0354 2
295 0355 2 !+
296 0356 2 ! Call PASS$DO_READV to do the work, giving it the address of
297 0357 2 ! PASS$READ_INTEGER to call.
298 0358 2 !-
299 0359 2
300 0360 2 PASS$DO_READV (PFV [PFV$R_PFV], .STRING, ARG_LIST, PASS$READ_INTEGER;
301 0361 2 RESULT);
302 0362 2
303 0363 2 RETURN .RESULT;
304 0364 2
305 0365 1 END; ! End of routine PASS$READV_INTEGER

```

```

                                .EXTRN PASS$DO_READV
                                .ENTRY PASS$READV_INTEGER, Save R2,R3,R4,R6
                                SUBL2 #32, SP
                                CLRL ERROR_ADDR
                                CLRQ UNWIND_ACT
                                MOVAL 2$, (FP)
                                CMPB (AP), #2
                                BLSSU 1$
                                MOVL ERROR, ERROR_ADDR
                                MOVAB PFV, PFV_ADDR
                                MOVL #1, ARG_LIST
                                MOVAB PFV, ARG_LIST+4
                                MOVAB PASS$READ_INTEGER, R4
                                MOVAB ARG_LIST, R3
                                MOVAB PFV, R6
                                MOVL STRING, R2
                                JSB PASS$DO_READV
                                RET
                                .WORD Save nothing

```


PASSREAD_INTEGE Read an integer
1-002 PASSREADV_INTEGER - Read integer from string

I 8
16-Sep-1984 01:57:44
14-Sep-1984 12:51:47

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

Page 9
(4)

50	08	AC	D0	00040
50	04	A0	D0	00044
	DC	A0	9F	00048
	E0	A0	9F	0004B
	E4	A0	9F	0004E
		03	DD	00051
		5E	DD	00053
00000000G	7E	04	AC	7D 00055
00			03	FB 00059
			04	00060

```
MOVL 8(AP), R0
MOVL 4(R0), R0
PUSHAB ERROR_ADDR
PUSHAB UNWIND_ACT
PUSHAB PFV_ADDR
PUSHL #3
PUSHL SP
MOVQ 4(AP), -(SP)
CALLS #3, PASS$IO_HANDLER
RET
```

;
;
;
;
;
;
;
;
;
;
;

; Routine Size: 97 bytes, Routine Base: _PASS\$CODE + 00BE

; 306 0366 1
; 307 0367 1 !<BLF/PAGE>

PASS\$READ_INTEGE Read an integer
1-002 PASS\$READV_INTEGER - Read integer from string

J 8
16-Sep-1984 01:57:44
14-Sep-1984 12:51:47

VAX-11 Bliss-32 V4.0-742
[PASRTL.SRC]PAS\$REAIN.T.B32;1

Page 10
(5)

: 309 0368 1 END
: 310 0369 1
: 311 0370 0 ELUDOM

! End of module PASS\$READ_INTEGER

PSECT SUMMARY

Name	Bytes	Attributes
_PASS\$CODE	287	NOVEC,NOWRT, RD , EXE, SHR, LCL, REL, CON, PIC,ALIGN(2)

Library Statistics

File	----- Total	Symbols Loaded	----- Percent	Pages Mapped	Processing Time
_\$255\$DUA28:[SYSLIB]STARLET.L32;1	9776	6	0	581	00:01.0
_\$255\$DUA28:[PASRTL.OBJ]PASLIB.L32;1	427	99	23	33	00:00.4

COMMAND QUALIFIERS

BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/NOTRACE/LIS=LIS\$:PAS\$REAIN.T/OBJ=OBJ\$:PAS\$REAIN.T MSRC\$:PAS\$REAIN.T/UPDATE=(ENH\$:PAS\$REAIN.T)

: Size: 279 code + 8 data bytes
: Run Time: 00:07.2
: Elapsed Time: 00:28.7
: Lines/CPU Min: 3104
: Lexemes/CPU-Min: 12965
: Memory Used: 84 pages
: Compilation Complete

0296

AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY