

| | | | | | | | | | | |
|------------|-----|--------------|-----|------------|--|------------|-----|------------|--|------------------|
| PPPPPPPPPP | | AAAAA | | SSSSSSSSSS | | RRRRRRRRRR | | TTTTTTTTTT | | LLL |
| PPPPPPPPPP | | AAAAA | | SSSSSSSSSS | | RRRRRRRRRR | | TTTTTTTTTT | | LLL |
| PPPPPPPPPP | | AAAAA | | SSSSSSSSSS | | RRRRRRRRRR | | TTTTTTTTTT | | 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 |
| PPPPPPPPPP | | AAA | AAA | SSSSSSSS | | RRRRRRRRRR | | TTT | | LLL |
| PPPPPPPPPP | | AAA | AAA | SSSSSSSS | | RRRRRRRRRR | | TTT | | LLL |
| PPPPPPPPPP | | AAA | AAA | SSSSSSSS | | RRRRRRRRRR | | TTT | | LLL |
| PPP | | AAAAAAAAAAAA | | SSS | | RRR | RRR | TTT | | LLL |
| PPP | | AAAAAAAAAAAA | | SSS | | RRR | RRR | TTT | | LLL |
| PPP | | AAAAAAAAAAAA | | 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 | SSSSSSSSSS | | RRR | RRR | TTT | | LLLLLLLLLLLLLLLL |
| PPP | | AAA | AAA | SSSSSSSSSS | | RRR | RRR | TTT | | LLLLLLLLLLLLLLLL |
| PPP | | AAA | AAA | SSSSSSSSSS | | RRR | RRR | TTT | | LLLLLLLLLLLLLLLL |

```

PPPPPPPP      AAAAAA      SSSSSSSS      RRRRRRRR      EEEEEEEEEE      AAAAAA      UU      UU      NN      NN      SSSSSSSS
PPPPPPPP      AAAAAA      SSSSSSSS      RRRRRRRR      EEEEEEEEEE      AAAAAA      UU      UU      NN      NN      SSSSSSSS
PP      PP      AA      AA      SS      RR      RR      EE      AA      AA      UU      UU      NN      NN      SS
PP      PP      AA      AA      SS      RR      RR      EE      AA      AA      UU      UU      NN      NN      SS
PP      PP      AA      AA      SS      RR      RR      EE      AA      AA      UU      UU      NN      NN      SS
PP      PP      AA      AA      SS      RR      RR      EE      AA      AA      UU      UU      NN      NN      SS
PPPPPPPP      AA      AA      SSSSSS      RRRRRRRR      EEEEEEEE      AA      AA      UU      UU      NN      NN      SSSSSS
PPPPPPPP      AA      AA      SSSSSS      RRRRRRRR      EEEEEEEE      AA      AA      UU      UU      NN      NN      SSSSSS
PP      AAAAAAAAAA      SS      RR      RR      EE      AAAAAAAAAA      UU      UU      NN      NN      SSSSSS
PP      AAAAAAAAAA      SS      RR      RR      EE      AAAAAAAAAA      UU      UU      NN      NN      SSSSSS
PP      AA      AA      SS      RR      RR      EE      AA      AA      UU      UU      NN      NN      SS
PP      AA      AA      SS      RR      RR      EE      AA      AA      UU      UU      NN      NN      SS
PP      AA      AA      SSSSSSSS      RR      RR      EEEEEEEEEE      AA      AA      UUUUUUUUUU      NN      NN      SSSSSSSS
PP      AA      AA      SSSSSSSS      RR      RR      EEEEEEEEEE      AA      AA      UUUUUUUUUU      NN      NN      SSSSSSSS

```

```

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 PASSREAD_UNSIGN ( XTITLE 'Read an unsigned integer'
2 0002 0 IDENT = '1-002' ! File: PASREAUNS.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 unsigned 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 26-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';           ! Externals, linkages, PSECTs, structures  
.. 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 UNSIGNED,           ! Read from textfile  
.. 62      0124 1     PASSREADV_UNSIGNED;         ! Read from string  
.. 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 !
```

```

82 0143 1 %SBTTL 'PASS$READ UNSIGNED - Read an unsigned integer from textfile'
83 0144 1 GLOBAL ROUTINE PASS$READ_UNSIGNED (
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 unsigned 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     Unsigned.wlu.v = PASS$READ_UNSIGNED (PFV.mr.r [, ERROR.ja.r])
97 0158 1
98 0159 1 FORMAL PARAMETERS:
99 0160 1
100 0161 1     PFV           - The Pascal File Variable (PFV) passed by reference.
101 0162 1                 The structure of the PFV is defined in P$PFV.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 unsigned 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     INVSYNUNS - invalid syntax for unsigned 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     STRING_LENGTH,  ! Length of string
134 0195 2     STRING_ADDR,     ! Address of string
135 0196 2     FCB: REF $PASS$FCB_CONTROL_BLOCK, ! File Control block
136 0197 2     PFV_ADDR: VOLATILE, ! Enable argument
137 0198 2     UNWIND_ACT: VOLATILE, ! Enable argument
138 0199 2     ERROR_ADDR: VOLATILE; ! Enable argument

```

```

139 0200 2
140 0201 2 BUILTIN
141 0202 2 ACTUALCOUNT; . Count of arguments
142 0203 2
143 0204 2 ENABLE
144 0205 2 PASS$IO_HANDLER (PFV_ADDR, UNWIND_ACT, ERROR_ADDR); ! Enable error handler
145 0206 2
146 0207 2 !+
147 0208 2 ! Get ERROR parameter, if present.
148 0209 2 !-
149 0210 2
150 0211 2 IF ACTUALCOUNT () GEQU 2
151 0212 2 THEN
152 0213 2 ERROR_ADDR = .ERROR; ! Set unwind address
153 0214 2
154 0215 2 PFV_ADDR = PFV [PFV$R_PFV]; ! Set PFV address
155 0216 2
156 0217 2 !+
157 0218 2 ! Validate PFV and get PFV.
158 0219 2 !-
159 0220 2
160 0221 2 PASS$VALIDATE_PFV (PFV [PFV$R_PFV]; FCB);
161 0222 2
162 0223 2 !+
163 0224 2 ! Set unwind action to unlock file.
164 0225 2 !-
165 0226 2
166 0227 2 UNWIND_ACT = PASS$K_UNWIND_UNLOCK;
167 0228 2
168 0229 2 !+
169 0230 2 ! Do common initialization.
170 0231 2 !-
171 0232 2
172 0233 2 PASS$INIT_READ (PFV [PFV$R_PFV], FCB [FCB$R_FCB]; FCB);
173 0234 2
174 0235 2 !+
175 0236 2 ! Call utility routine to find a string that looks like an unsigned integer.
176 0237 2 ! If we can't find one, signal an error.
177 0238 2 !-
178 0239 2
179 0240 2 IF NOT PASS$GET_UNSIGNED (PFV [PFV$R_PFV], FCB [FCB$R_FCB];
180 0241 2 STRING_ADDR, STRING_LENGTH, FCB)
181 0242 2 THEN
182 P 0243 2 $PASS$IO_ERROR (PASS$INVSYNUNS, 3, .STRING_LENGTH, .STRING_ADDR,
183 0244 2 .FCB [FCB$R_RECORD_NUMBER])
184 0245 2
185 0246 2 !+
186 0247 2 ! Call convert routine. If it fails, signal an error.
187 0248 2 !-
188 0249 2
189 0250 2 ELSE IF NOT LIB$CVT_DTB (.STRING_LENGTH, .STRING_ADDR, RESULT)
190 0251 2 THEN
191 0252 2 BEGIN
192 0253 2 LOCAL
193 0254 2 DESCR: BLOCK [8, BYTE];
194 0255 2 DESCR [DSC$W_LENGTH] = .STRING_LENGTH;
195 0256 2 DESCR [DSC$B_CLASS] = 0;

```

: 196
: 197
: 198
: 199
: 200
: 201
: 202
: 203
: 204
: 205
: 206
: 207
: 208
: 209
: 210
: 211

0257 3
0258 3
P 0259 3
P 0260 3
0261 3
0262 2
0263 2
0264 2
0265 2
0266 2
0267 2
0268 2
0269 2
0270 2
0271 2
0272 1

```
DESCR [DSC$B_DTYPE] = 0;
DESCR [DSC$A_POINTER] = .STRING_ADDR;
$PASSIO_ERROR (PASS_NOTVALTYP, 3, DESCR,
  UPLIT BYTE (%CHARCOUNT('UNSIGNED'), 'UNSIGNED'),
  .FCB [FCB$L_RECORD_NUMBER]);
END;

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

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

RETURN .RESULT;

END;
```

! End of routine PASS\$READ_UNSIGNED

| | | | | | | |
|----|----|----|-----------|--------|---|---------------|
| | | | | .TITLE | PASS\$READ_UNSIGNED Read an unsigned integer | |
| | | | | .IDENT | \1-002\ | |
| | | | | .PSECT | _PASS\$CODE, NOWRT, SHR, PIC, 2 | |
| 44 | 45 | 4E | 47 | 49 | 53 | 4E |
| | | | | | 08 | 00000 |
| | | | | | 55 | 00001 |
| | | | | | P.AAA: | |
| | | | | .BYTE | 8 | |
| | | | | .ASCII | \UNSIGNED\ | : |
| | | | | .EXTRN | PASS\$READ_UNSIGNED | |
| | | | | .EXTRN | PASS\$READV_UNSIGNED | |
| | | | | .EXTRN | PASS\$IO_HANDLER | |
| | | | | .EXTRN | PASS\$VACIDATE_PFV | |
| | | | | .EXTRN | PASS\$INIT_READ, PASS\$GET_UNSIGNED | |
| | | | | .EXTRN | PASS\$SIGNAL, PASS\$K_INVSYNUNS | |
| | | | | .EXTRN | LIB\$CVT_DTB, PASS\$K_NOTVALTYP | |
| | | | | .EXTRN | PASS\$END_READ | |
| | | | | .ENTRY | PASS\$READ_UNSIGNED, Save R2,R3,R4,R5,R6,R7,- | : 0144 |
| | | | | | R8 | |
| | 5E | | | | 18 | C2 00002 |
| | | 0C | | | AE | 7C 00005 |
| | | 14 | | | AE | D4 00008 |
| | 6D | | 0078 | | CF | DE 0000B |
| | 02 | | | | 6C | 91 00010 |
| | | | | | 05 | 1F 00013 |
| | 0C | AE | | 08 | AC | DD 00015 |
| | | 56 | | 04 | AC | DD 0001A 1\$: |
| | 14 | AE | | | 56 | DD 0001E |
| | | | 00000000G | | 00 | 16 00022 |
| | 10 | AE | | | 01 | DD 00028 |
| | | | 00000000G | | 00 | 16 0002C |
| | | | 00000000G | | 00 | 16 00032 |
| | | 0F | | | 50 | E8 00038 |
| | | | C8 | | A7 | DD 0003B |
| | | | | | 54 | DD 0003E |
| | | | | | 55 | DD 00040 |
| | | | | | 03 | DD 00042 |
| | | 7E | | 00G | 8F | 9A 00044 |
| | | | | | 27 | 11 00048 |
| | | | | SUBL2 | #24, SP | |
| | | | | CLRG | ERROR_ADDR | : 0190 |
| | | | | CLRL | PFV_ADDR | |
| | | | | MOVAL | 6\$, (FP) | |
| | | | | CMPB | (AP), #2 | : 0211 |
| | | | | BLSSU | 1\$ | |
| | | | | MOVL | ERROR, ERROR_ADDR | : 0213 |
| | | | | MOVL | PFV, R6 | : 0215 |
| | | | | MOVL | R6, PFV_ADDR | |
| | | | | JSB | PASS\$VACIDATE_PFV | : 0221 |
| | | | | MOVL | #1, UNWIND_ACT | : 0227 |
| | | | | JSB | PASS\$INIT_READ | : 0233 |
| | | | | JSB | PASS\$GET_UNSIGNED | : 0240 |
| | | | | BLBS | R0, 2\$ | |
| | | | | PUSHL | -56(FCB) | : 0244 |
| | | | | PUSHL | STRING_ADDR | |
| | | | | PUSHL | STRING_LENGTH | |
| | | | | PUSHL | #3 | |
| | | | | MOVZBL | #PASS\$K_INVSYNUNS, -(SP) | |
| | | | | BRB | 3\$ | |

PASSREAD_UNSIGN Read an unsigned integer
1-002

PASSREAD_UNSIGNED - Read an unsigned integer fr

E 14
16-Sep-1984 02:03:53
14-Sep-1984 12:51:52

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

Page 6
(3)

PAS
1-0

| | | | | | | | | | | |
|-----------|----|-----------|----|------|-------|------|--------|-------------------------|--|------|
| | | 4010 | 8F | BB | 0004A | 2\$: | PUSHR | #^M<R4,SP> | | 0250 |
| | | | 55 | DD | 0004E | | PUSHL | STRING_LENGTH | | |
| 00000000G | 00 | | 03 | FB | 00050 | | CALLS | #3, LIB\$CVT_DTB | | |
| | 20 | | 50 | E8 | 00057 | | BLBS | R0, 4\$ | | |
| | 04 | AE | 55 | 3C | 0005A | | MOVZWL | STRING_LENGTH, DESCR | | 0255 |
| | 08 | AE | 54 | D0 | 0005E | | MOVL | STRING_ADDR, DESCR+4 | | 0258 |
| | | C8 | A7 | DD | 00062 | | PUSHL | -56(FCB) | | 0261 |
| | | 8F | AF | 9F | 00065 | | PUSHAB | P.AAA | | |
| | | 0C | AE | 9F | 00068 | | PUSHAB | DESCR | | |
| | | | 03 | DD | 0006B | | PUSHL | #3 | | |
| | 7E | 00G | 8F | 9A | 0006D | | MOVZBL | #PASSK NOTVALTYP, -(SP) | | |
| 00000000G | 0C | | 05 | FB | 00071 | 3\$: | CALLS | #5, PASS\$SIGNAL | | |
| | | | 0A | 11 | 00078 | | BRB | 5\$ | | |
| | | 00000000G | 00 | 16 | 0007A | 4\$: | JSB | PASS\$END_READ | | 0268 |
| | 50 | | 6E | D0 | 00080 | | MOVL | RESULT, R0 | | 0270 |
| | | | | 04 | 00083 | | RET | | | |
| | | | 50 | D4 | 00084 | 5\$: | CLRL | R0 | | 0272 |
| | | | | 04 | 00086 | | RET | | | |
| | | | | 0000 | 00087 | 6\$: | .WORD | Save nothing | | 0190 |
| | 50 | 08 | AC | D0 | 00089 | | MOVL | 8(AP), R0 | | |
| | 50 | 04 | A0 | D0 | 0008D | | MOVL | 4(R0), R0 | | |
| | | F4 | A0 | 9F | 00091 | | PUSHAB | ERROR_ADDR | | |
| | | F8 | A0 | 9F | 00094 | | PUSHAB | UNWIND_ACT | | |
| | | FC | A0 | 9F | 00097 | | PUSHAB | PFV_ADDR | | |
| | | | 03 | DD | 0009A | | PUSHL | #3 | | |
| | | | 5E | DD | 0009C | | PUSHL | SP | | |
| | 7E | 04 | AC | 7D | 0009E | | MOVQ | 4(AP), -(SP) | | |
| 00000000G | 00 | | 03 | FB | 000A2 | | CALLS | #3, PASS\$IO_HANDLER | | |
| | | | | 04 | 000A9 | | RET | | | |

: Routine Size: 170 bytes, Routine Base: _PASS\$CODE + 0009

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


```

215 0275 1 %SBTTL 'PASS$READV UNSIGNED - Read an unsigned from string'
216 0276 1 GLOBAL ROUTINE PASS$READV UNSIGNED (
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 unsigned 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     Unsigned.wlu.v = PASS$READV_UNSIGNED (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 unsigned 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     NONE
259 0319 1
260 0320 1 !--
261 0321 1
262 0322 2     BEGIN
263 0323 2
264 0324 2     LOCAL
265 0325 2         PFV: $PASS$PFV_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 ENABLE
276 0336 2 PASS$IO_HANDLER (PFV_ADDR, UNWIND_ACT, ERROR_ADDR); ! Enable error handler
277 0337 2
278 0338 2 !+
279 0339 2 ! Get ERROR parameter, if present.
280 0340 2 !-
281 0341 2
282 0342 2 IF ACTUALCOUNT () GEQU 2
283 0343 2 THEN
284 0344 2 ERROR_ADDR = .ERROR; ! Set unwind address
285 0345 2
286 0346 2 PFV_ADDR = PFV [PFV$R_P V]; ! 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$R_P FV]; ! PFV address
294 0354 2
295 0355 2 !+
296 0356 2 ! Call PASS$DO_READV to do the work, giving it the address of
297 0357 2 ! PASS$READ_UNSIGNED to call.
298 0358 2 !-
299 0359 2
300 0360 2 PASS$DO_READV (PFV [PFV$R_P FV], .STRING, ARG_LIST, PASS$READ_UNSIGNED;
301 0361 2 RESULT);
302 0362 2
303 0363 2 RETURN .RESULT;
304 0364 2
305 0365 1 END; ! End of routine PASS$READV_UNSIGNED

```

```

                .EXTRN PASS$DO_READV
                .ENTRY PASS$READV_UNSIGNED, Save R2,R3,R4,R6 : 0276
5E              20 C2 00002  SUBL2 #32, SP : 0322
                7E D4 00005  CLRL ERROR_ADDR : 0322
                04 AE 7C 00007  CLRQ UNWIND_ACT
6D              0030 CF DE 0000A  MOVAL 2$, (FP)
02              6C 91 0000F  CMPB (AP), #2 : 0342
                04 1F 00012  BLSSU 1$
                08 AC D0 00014  MOVL ERROR, ERROR_ADDR : 0344
08              14 AE 9E 00018 1$: MOVAB PFV, PFV_ADDR : 0346
0C              01 D0 0001D  MOVL #1, ARG_LIST : 0352
10              14 AE 9E 00021  MOVAB PFV, ARG_LIST+4 : 0353
                FF2C CF 9E 00026  MOVAB PASS$READ_UNSIGNED, R4 : 0360
                53 0C AE 9E C002B  MOVAB ARG_LIST, R3
                56 14 AE 9E 0002F  MOVAB PFV, R6
                52 04 AC D0 00033  MOVL STRING, R2
                00000000G 00 16 00037  JSB PASS$DO_READV
                04 0003D  RET : 0365
                0000 0003E 2$: .WORD Save nothing : 0322

```

PASSREAD_UNSIGN Read an unsigned integer
1-002

PASSREADV_UNSIGNED - Read an unsigned from stri

H 14
16-Sep-1984 02:03:53
14-Sep-1984 12:51:52

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

Page 9
(4)

| | | | | | | |
|-----------|----|----|-------|-------|--------|----------------------|
| 50 | 08 | AC | D0 | 00040 | MOVL | 8(AP), R0 |
| 50 | 04 | A0 | D0 | 00044 | MOVL | 4(R0), R0 |
| | DC | A0 | 7F | 00048 | PUSHAB | ERROR_ADDR |
| | E0 | A0 | 9F | 0004B | PUSHAB | UNWIND_ACT |
| | E4 | A0 | 9F | 0004E | PUSHAB | PFV_ADDR |
| | | 03 | DD | 00051 | PUSHL | #3 |
| | | 5E | DD | 00053 | PUSHL | SP |
| 00000000G | 7E | 04 | AC | 7D | MOVQ | 4(AP), -(SP) |
| 00 | 00 | 03 | FB | 00059 | CALLS | #3, PASS\$IO_HANDLER |
| | | 04 | 00060 | | RET | |

.....

: Routine Size: 97 bytes, Routine Base: _PASSCODE + 00B3

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

PASSREAD_UNSIGN Read an unsigned integer
1-002

PASSREADV_UNSIGNED - Read an unsigned from stri

I 14
16-Sep-1984 02:03:53
14-Sep-1984 12:51:52

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

Page 10
(5)

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

! End of module PASSREAD_UNSIGNED

PSECT SUMMARY

| Name | Bytes | Attributes |
|-----------|-------|--|
| _PASSCODE | 276 | 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 | 4 | 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\$:PASREAUNS/OBJ=OBJ\$:PASREAUNS MSRC\$:PASREAUNS/UPDATE=(ENH\$:PASREAUNS)

: Size: 267 code + 9 data bytes
: Run Time: 00:07.2
: Elapsed Time: 00:25.2
: Lines/CPU Min: 3087
: Lexemes/CPU-Min: 13276
: Memory Used: 87 pages
: Compilation Complete

This page contains a grid of approximately 140 small diagrams or code snippets, each enclosed in a rectangular frame. The diagrams are organized in a roughly rectangular grid pattern. Many of the diagrams have titles at the top, such as:

- PASREABOO LIS
- PASREAST LIS
- PASREAREH LIS
- PASREASETK LIS
- PASREAREG LIS
- PASRAB LIS
- PASREASET2 LIS
- PASREADLN LIS
- PASREAREF LIS
- PASREAVAR LIS
- PASREARED LIS
- PASREACHA LIS
- PASREADUT LIS
- PASREAENU LIS
- PASREAUNS LIS
- PASREASR LIS
- PASREASR2 LIS

The diagrams themselves contain various elements, including text labels, numerical values, and small schematic-like structures. Some diagrams include a title, a list of items, or a small diagram with lines and nodes. The overall appearance is that of a technical manual page for a VAX/VMS system, showing various procedures and their outputs or internal structures. The text is small and difficult to read in detail, but the layout is consistent across the grid.