

```

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

PAS
PAS

PAS
PAS

PAS

PAS
PAS

PAS
PAS

PAS

PAS

PAS
PASPAS
PAS

PAS

PAS

PAS
PAS

PAS

PAS

PAS
PAS

PAS
PAS

PAS

22

PAS

PAS

TABLE 1

PAS

PAS

PAS

PAS

PAS
PAS

PAS

PAS

PAS

PAS
PAS

PAS

PAS

PAS
PAS

PAS

```
PPPPPPPP      AAAAAA      SSSSSSSSS      IIIIII      000000      HH      HH      AAAAAA      NN      NN      DDDDDDDD
PPPPPPPP      AAAAAA      SSSSSSSSS      IIIIII      000000      HH      HH      AAAAAA      NN      NN      DDDDDDDD
PP      PP      AA      AA      SS      II      00      00      HH      HH      AA      AA      NN      NN      DD      DD
PP      PP      AA      AA      SS      II      00      00      HH      HH      AA      AA      NN      NN      DD      DD
PP      PP      AA      AA      SS      II      00      00      HH      HH      AA      AA      NNNN      NN      DD      DD
PP      PP      AA      AA      SS      II      00      00      HH      HH      AA      AA      NNNN      NN      DD      DD
PPPPPPPP      AA      AA      SSSSSS      II      00      00      HHHHHHHHHH      AA      AA      NN      NN      DD      DD
PPPPPPPP      AA      AA      SSSSSS      II      00      00      HHHHHHHHHH      AA      AA      NN      NN      DD      DD
PP      AAAAAAAAAA      SS      II      00      00      HH      HH      AAAAAAAAAA      NN      NNNN      DD      DD
PP      AAAAAAAAAA      SS      II      00      00      HH      HH      AAAAAAAAAA      NN      NNNN      DD      DD
PP      AA      AA      SS      II      00      00      HH      HH      AA      AA      NN      NN      DD      DD
PP      AA      AA      SS      II      00      00      HH      HH      AA      AA      NN      NN      DD      DD
PP      AA      AA      SSSSSSSS      IIIIII      000000      HH      HH      AA      AA      NN      NN      DDDDDDDD
PP      AA      AA      SSSSSSSS      IIIIII      000000      HH      HH      AA      AA      NN      NN      DDDDDDDD
                                     ....
                                     ....
                                     ....
                                     ....

LL      IIIIII      SSSSSSSSS
LL      IIIIII      SSSSSSSSS
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 PAS$$IO_HANDLER ( %TITLE 'I/O Error Handler'
2 0002 0 IDENT = '1-003' ! File: PASIOHAND.B32 Edit: SBL1003
3 0003 0 ) =
4 0004 1 BEGIN
5 0005 1
6 0006 1 *****
7 0007 1 *
8 0008 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
9 0009 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
10 0010 1 * ALL RIGHTS RESERVED.
11 0011 1 *
12 0012 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
13 0013 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
14 0014 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
15 0015 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
16 0016 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
17 0017 1 * TRANSFERRED.
18 0018 1 *
19 0019 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
20 0020 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
21 0021 1 * CORPORATION.
22 0022 1 *
23 0023 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
24 0024 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
25 0025 1 *
26 0026 1 *
27 0027 1 *****
28 0028 1
29 0029 1 ++
30 0030 1 FACILITY: Pascal Language Support
31 0031 1
32 0032 1 ABSTRACT:
33 0033 1
34 0034 1 This module contains PAS$$IO_HANDLER, the condition handler established
35 0035 1 by all Pascal I/O procedures.
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
44 0044 1 1-001 - Original. SBL 1-April-1981
45 0045 1 1-002 - If a zero-level access violation occurs while in PAS$$VALIDATE_PFV,
46 0046 1 convert the message to PASS_INVFILVAR. SBL 14-July-1982
47 0047 1 1-003 - Check for PASS_CONTINUE regardless of whether ERROR:=CONTINUE
48 0048 1 was specified. SBL 29-Oct-1982
49 0049 1 --
50 0050 1
```

```

: 52      0051 1 %SBTTL 'Declarations'
: 53      0052 1
: 54      0053 1 PROLOGUE DEFINITIONS:
: 55      0054 1
: 56      0055 1
: 57      0056 1 REQUIRE 'RTLIN:PASPROLOG';           ! Externals, linkages, PSECTs, structures
: 58      0120 1
: 59      0121 1
: 60      0122 1 TABLE OF CONTENTS:
: 61      0123 1
: 62      0124 1
: 63      0125 1 FORWARD ROUTINE
: 64      0126 1 PAS$$IO_HANDLER;                     ! I/O error handler
: 65      0127 1
: 66      0128 1
: 67      0129 1 MACROS:
: 68      0130 1
: 69      0131 1 NONE
: 70      0132 1
: 71      0133 1 EQUATED SYMBOLS:
: 72      0134 1
: 73      0135 1 NONE
: 74      0136 1
: 75      0137 1 FIELDS:
: 76      0138 1
: 77      0139 1 NONE
: 78      0140 1
: 79      0141 1 OWN STORAGE:
: 80      0142 1
: 81      0143 1 NONE

```



```

83 0144 1 %SBTTL 'PASS$IO_HANDLER - I/O Error handler'
84 0145 1 GLOBAL ROUTINE PASS$IO_HANDLER (
85 0146 1     SIGNAL_ARGS: REF BLOCK [, BYTE],
86 0147 1     MECH_ARGS: REF BLOCK [, BYTE],
87 0148 1     ENABLE_ARGS: REF VECTOR [, LONG]
88 0149 1 ) =
89 0150 1
90 0151 1 ++
91 0152 1 FUNCTIONAL DESCRIPTION:
92 0153 1
93 0154 1     This is the condition handler enabled by all VAX-11 Pascal I/O
94 0155 1     procedures. It implements the ERROR:=CONTINUE specifier by
95 0156 1     unwinding to the address specified in the enable argument ERROR_ADDR.
96 0157 1
97 0158 1     Upon an unwind, if the file is not fully opened, it is closed. The
98 0159 1     file is unlocked on unwind.
99 0160 1
100 0161 1 CALLING SEQUENCE:
101 0162 1
102 0163 1     status.wlc.v = PASS$IO_HANDLER (SIGNAL_ARGS.rl.ra, MECH_ARGS.rl.ra
103 0164 1     , ENABLE_ARGS.rl.ra)
104 0165 1
105 0166 1 FORMAL PARAMETERS:
106 0167 1
107 0168 1     SIGNAL_ARGS    - The signal argument list.
108 0169 1
109 0170 1     MECH_ARGS      - The mechanism argument list.
110 0171 1
111 0172 1     ENABLE_ARGS    - An array with the following
112 0173 1     format:
113 0174 1
114 0175 1     +-----+
115 0176 1     | ENB_COUNT | <-- ENABLE_ARGS
116 0177 1     +-----+
117 0178 1     | ENB_PFV_ADDR |
118 0179 1     +-----+
119 0180 1     | ENB_UNWIND_ACT |
120 0181 1     +-----+
121 0182 1     | ENB_ERROR_ADDR |
122 0183 1     +-----+
123 0184 1
124 0185 1     ENB_COUNT is the count of following enable arguments.
125 0186 1     The count is always at least 2.
126 0187 1
127 0188 1     ENB_PFV_ADDR - If non-zero, the address of a longword
128 0189 1     containing the PFV our establisher is operating on.
129 0190 1
130 0191 1     ENB_UNWIND_ACT - Specifies the action
131 0192 1     to take on an unwind. The values are:
132 0193 1     PASS$K_UNWIND_NOP - Do nothing
133 0194 1     PASS$K_UNWIND_UNLOCK - Unlock PFV
134 0195 1
135 0196 1     ENB_ERROR_ADDR - If present and non-zero, the address
136 0197 1     of a longword containing the PC of an instruction to
137 0198 1     unwind to. This is present if the user specified
138 0199 1     Error:=Continue in the I/O statement.
139 0200 1

```



```

140 0201 1 | IMPLICIT INPUTS:
141 0202 1 |
142 0203 1 |     The signaller's PFV placed as the first FA0 argument in the primary
143 0204 1 |     signalled message.
144 0205 1 |
145 0206 1 | IMPLICIT OUTPUTS:
146 0207 1 |
147 0208 1 |     May clear PFV$V_LOCK upon unwind.
148 0209 1 |
149 0210 1 | ROUTINE VALUE:
150 0211 1 |
151 0212 1 |     SSS_RESIGNAL
152 0213 1 |
153 0214 1 | SIDE EFFECTS:
154 0215 1 |
155 0216 1 |     May cause an unwind.
156 0217 1 |
157 0218 1 |     If a zero-level access violation occurs while in PASS$VALIDATE_PFV,
158 0219 1 |     the signal is converted to PASS_INVFILVAR.
159 0220 1 |
160 0221 1 | --
161 0222 1 |
162 0223 2 | BEGIN
163 0224 2 |
164 0225 2 | LITERAL
165 0226 2 |     ENB_COUNT = 0,           ! Count of enable arguments
166 0227 2 |     ENB_PFV_ADDR = 1,       ! Address of address of PFV
167 0228 2 |     ENB_UNWIND_ACT = 2,     ! Address of unwind action
168 0229 2 |     ENB_ERROR_ADDR = 3;     ! Address of address of unwind PC
169 0230 2 |
170 0231 2 | BUILTIN
171 0232 2 |     ACTUALCOUNT;
172 0233 2 |
173 0234 2 | !+
174 0235 2 | ! Determine if this is an unwind.
175 0236 2 | !-
176 0237 2 |
177 0238 2 | IF .SIGNAL_ARGS [CHF$L_SIG_NAME] EQLU SSS_UNWIND
178 0239 2 | THEN
179 0240 3 |     BEGIN
180 0241 3 |     IF ..ENABLE_ARGS [ENB_UNWIND_ACT] NEQ PASS$K_UNWIND_NOP
181 0242 3 |     THEN
182 0243 4 |         BEGIN
183 0244 4 |
184 0245 4 |         !+
185 0246 4 |         ! Get our establisher's PFV.
186 0247 4 |         !-
187 0248 4 |
188 0249 4 |         LOCAL
189 0250 4 |         PFV: REF $PASS$PFV FILE VARIABLE;
190 0251 4 |         PFV = ..ENABLE_ARGS [ENB_PFV_ADDR]; ! Get PFV address
191 0252 4 |
192 0253 4 |         PFV [PFV$V_LOCK] = 0;           ! Unlock PFV
193 0254 3 |         END;
194 0255 3 |     END
195 0256 3 |
196 0257 2 | ELSE

```



```

197      0258 2
198      0259 3 BEGIN
199      0260 3
200      0261 3 LOCAL
201      0262 3 COND_NAME: BLOCK [4, BYTE]; ! Primary condition name
202      0263 3
203      0264 3
204      0265 3 !+
205      0266 3 ! Get primary condition name.
206      0267 3 !-
207      0268 3 COND_NAME = .SIGNAL_ARGS [CHF$L_SIG_NAME];
208      0269 3
209      0270 3 !+
210      0271 3 ! If this is a zero-level access violation, and if the PC is inside
211      0272 3 ! PASS$VALIDATE_PFV, we assume that we have been passed an invalid
212      0273 3 ! file variable. Convert the signal to PASS$INVFILVAR.
213      0274 3 !-
214      0275 3
215      0276 4 IF (.COND_NAME EQLU SS$_ACCVIO) AND (.MECH_ARGS [CHF$L_MCH_DEPTH] EQL 0)
216      0277 3 THEN
217      0278 3 !+
218      0279 3 ! Compare PC against range for PASS$VALIDATE_PFV.
219      0280 3 !-
220      0281 3 IF (.SIGNAL_ARGS [16,0,32,0] GEQA PASS$VALIDATE_PFV) AND
221      0282 4 (.SIGNAL_ARGS [16,0,32,0] LSSA PASS$VALIDATE_PFV_END)
222      0283 3 THEN
223      0284 4 BEGIN
224      0285 4 SIGNAL_ARGS [CHF$L_SIG_NAME] = PASS$INVFILVAR;
225      0286 4 SIGNAL_ARGS [8,0,32,0] = 1; ! FAO count
226      0287 4 SIGNAL_ARGS [12,0,32,0] = ..ENABLE_ARGS [ENB_PFV_ADDR];
227      0288 4 COND_NAME = .SIGNAL_ARGS [CHF$L_SIG_NAME];
228      0289 4 END;
229      0290 3
230      0291 3 !+
231      0292 3 ! Is this a PASS$ error? If not, resignal.
232      0293 3 !-
233      0294 3
234      0295 3 IF .COND_NAME [STSSV_FAC_NO] NEQU PASS$_FACILITY
235      0296 3 THEN
236      0297 3 RETURN SS$_RESIGNAL;
237      0298 3
238      0299 3 !+
239      0300 3 ! See if the PFVs match. The signaller's PFV is the
240      0301 3 ! first FAO parameter in the primary message. If so, then
241      0302 3 ! continue with Pascal-specific processing.
242      0303 3 !-
243      0304 3
244      0305 3 IF .SIGNAL_ARGS [12,0,32,0] EQLA ..ENABLE_ARGS [ENB_PFV_ADDR]
245      0306 3 THEN
246      0307 4 BEGIN
247      0308 4
248      0309 4 !+
249      0310 4 ! Check to see if the current exception code is PASS$CONTINUE.
250      0311 4 ! This is generated by PASS$SIGNAL if a user condition handler
251      0312 4 ! continued from the signal. If so, do a default unwind to
252      0313 4 ! the caller of our establisher.
253      0314 4 !-

```



```

254      0315  4
255      0316  4      IF .COND_NAME EQLU PASS_CONTINUE
256      0317  4      THEN
257      0318  5          IF NOT $UNWIND ( )
258      0319  4          THEN
259      0320  4              $PASS$BUGCHECK (BUG_UNWINDFAIL);
260      0321  4
261      0322  4      !+
262      0323  4      ! Has ERROR:=CONTINUE been specified by our establisher? This
263      0324  4      ! is determined by having ENB_ERROR_ADDR be non-zero. If so,
264      0325  4      ! we may want to unwind.
265      0326  4      !-
266      0327  4
267      0328  4      IF .ENABLE_ARGS [ENB_COUNT] GEQU ENB_ERROR_ADDR
268      0329  4      THEN
269      0330  5          BEGIN
270      0331  5              IF ..ENABLE_ARGS [ENB_ERROR_ADDR] NEQ 0
271      0332  5              THEN
272      0333  6                  BEGIN
273      0334  6
274      0335  6                  !+
275      0336  6                  ! See if the error message is one which is "trapped"
276      0337  6                  ! by ERROR:=CONTINUE. This is done by comparing the
277      0338  6                  ! message number against a select range.
278      0339  6                  !-
279      0340  6
280      0341  6                  IF .COND_NAME [ST$V_CODE] GEQU PASS$K_MSGCONTLO AND ! Lowest number
281      0342  6                  .COND_NAME [ST$V_CODE] LEQU PASS$K_MSGCONTHI
282      0343  6                  THEN
283      0344  7                      BEGIN
284      0345  7
285      0346  7                      !+
286      0347  7                      ! We want to do the ERROR:=CONTINUE. Unwind to the caller
287      0348  7                      ! of our establisher at the specified PC.
288      0349  7                      !-
289      0350  7
290      0351  8                      IF NOT $UNWIND (NEWPC = ..ENABLE_ARGS [ENB_ERROR_ADDR])
291      0352  7                      THEN
292      0353  7                          $PASS$BUGCHECK (BUG_UNWINDFAIL);
293      0354  6                      END;
294      0355  5                  END;
295      0356  4              END;
296      0357  3          END;
297      0358  2      END;
298      0359  2      RETURN SS$_RESIGNAL;
299      0360  2
300      0361  2
301      0362  1      END;

```

! Resignal error

! End of routine PASS\$IO_HANDLER

.TITLE PASS\$IO_HANDLER I/O Error Handler
.IDENT \1-003\

.EXTRN PASS\$IO_HANDLER
.EXTRN PASS\$VALIDATE_PFV
.EXTRN PASS\$VALIDATE_PFV_END
.EXTRN PASS_INVFILVAR, PASS_FACILITY

Address	Hex	Label	Op	Op2	Op3	Op4	Op5	Op6	Op7	Op8	Op9	Op10	Op11	Op12	Op13	Op14	Op15	Op16	Op17	Op18	Op19	Op20	Op21	Op22	Op23	Op24	Op25	Op26	Op27	Op28	Op29	Op30	Op31	Op32	Op33	Op34	Op35	Op36	Op37	Op38	Op39	Op40	Op41	Op42	Op43	Op44	Op45	Op46	Op47	Op48	Op49	Op50	Op51	Op52	Op53	Op54	Op55	Op56	Op57	Op58	Op59	Op60	Op61	Op62	Op63	Op64	Op65	Op66	Op67	Op68	Op69	Op70	Op71	Op72	Op73	Op74	Op75	Op76	Op77	Op78	Op79	Op80	Op81	Op82	Op83	Op84	Op85	Op86	Op87	Op88	Op89	Op90	Op91	Op92	Op93	Op94	Op95	Op96	Op97	Op98	Op99	Op100	Op101	Op102	Op103	Op104	Op105	Op106	Op107	Op108	Op109	Op110	Op111	Op112	Op113	Op114	Op115	Op116	Op117	Op118	Op119	Op120	Op121	Op122	Op123	Op124	Op125	Op126	Op127	Op128	Op129	Op130	Op131	Op132	Op133	Op134	Op135	Op136	Op137	Op138	Op139	Op140	Op141	Op142	Op143	Op144	Op145	Op146	Op147	Op148	Op149	Op150	Op151	Op152	Op153	Op154	Op155	Op156	Op157	Op158	Op159	Op160	Op161	Op162	Op163	Op164	Op165	Op166	Op167	Op168	Op169	Op170	Op171	Op172	Op173	Op174	Op175	Op176	Op177	Op178	Op179	Op180	Op181	Op182	Op183	Op184	Op185	Op186	Op187	Op188	Op189	Op190	Op191	Op192	Op193	Op194	Op195	Op196	Op197	Op198	Op199	Op200	Op201	Op202	Op203	Op204	Op205	Op206	Op207	Op208	Op209	Op210	Op211	Op212	Op213	Op214	Op215	Op216	Op217	Op218	Op219	Op220	Op221	Op222	Op223	Op224	Op225	Op226	Op227	Op228	Op229	Op230	Op231	Op232	Op233	Op234	Op235	Op236	Op237	Op238	Op239	Op240	Op241	Op242	Op243	Op244	Op245	Op246	Op247	Op248	Op249	Op250	Op251	Op252	Op253	Op254	Op255	Op256	Op257	Op258	Op259	Op260	Op261	Op262	Op263	Op264	Op265	Op266	Op267	Op268	Op269	Op270	Op271	Op272	Op273	Op274	Op275	Op276	Op277	Op278	Op279	Op280	Op281	Op282	Op283	Op284	Op285	Op286	Op287	Op288	Op289	Op290	Op291	Op292	Op293	Op294	Op295	Op296	Op297	Op298	Op299	Op300	Op301	Op302	Op303	Op304	Op305	Op306	Op307	Op308	Op309	Op310	Op311	Op312	Op313	Op314	Op315	Op316	Op317	Op318	Op319	Op320	Op321	Op322	Op323	Op324	Op325	Op326	Op327	Op328	Op329	Op330	Op331	Op332	Op333	Op334	Op335	Op336	Op337	Op338	Op339	Op340	Op341	Op342	Op343	Op344	Op345	Op346	Op347	Op348	Op349	Op350	Op351	Op352	Op353	Op354	Op355	Op356	Op357	Op358	Op359	Op360	Op361	Op362	Op363	Op364	Op365	Op366	Op367	Op368	Op369	Op370	Op371	Op372	Op373	Op374	Op375	Op376	Op377	Op378	Op379	Op380	Op381	Op382	Op383	Op384	Op385	Op386	Op387	Op388	Op389	Op390	Op391	Op392	Op393	Op394	Op395	Op396	Op397	Op398	Op399	Op400	Op401	Op402	Op403	Op404	Op405	Op406	Op407	Op408	Op409	Op410	Op411	Op412	Op413	Op414	Op415	Op416	Op417	
---------	-----	-------	----	-----	-----	-----	-----	-----	-----	-----	-----	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	--

PASS\$IO_HANDLER I/O Error Handler
1-003 PASS\$IO_HANDLER - I/O Error handler

F 3
16-Sep-1984 01:42:08
14-Sep-1984 12:51:33

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

Page 8
(3)

50	0918	06	11	000C4	BRB	7\$
		8F	3C	000C6	MOVZWL	#2328, R0
			04	000CB	RET	
		50	D4	000CC	CLRL	R0
			04	000CE	RET	

: 0360
:
:
:
: 0362

; Routine Size: 207 bytes, Routine Base: _PAS\$CODE + 0000

; 302 0363 1

PASS\$IO_HANDLER I/O Error Handler
1-003 PASS\$IO_HANDLER - I/O Error handler

G 3
16-Sep-1984 01:42:08 VAX-11 Bliss-32 V4.0-742
14-Sep-1984 12:51:33 [PASRTL.SRC]PASIOHAND.B32;1

Page 9
(4)

: 304 0364 1 END
: 305 0365 1
: 306 0366 0 ELUDOM

! End of module PASS\$IO_HANDLER

PSECT SUMMARY

Name	Bytes	Attributes
_PASS\$CODE	207	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	9	0	581	00:01.0
-\$255\$DUA28:[PASRTL.OBJ]PASLIB.L32;1	427	32	7	33	00:00.4

COMMAND QUALIFIERS

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

: Size: 207 code + 0 data bytes
: Run Time: 00:07.0
: Elapsed Time: 00:31.1
: Lines/CPU Min: 3150
: Lexemes/CPU-Min: 12086
: Memory Used: 91 pages
: Compilation Complete

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

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY