


```

1 0001 0 MODULE COM$ERRSET_TST (%TITLE'FORTRAN compatibility error set and test'
2 0002 0 IDENT = '1-024' ! File: COMSETST.B32 Edit: SBL1024
3 0003 0 ) =
4 0004 1 BEGIN
5 0005 1
6 0006 1 *****
7 0007 1 *
8 0008 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY *
9 0009 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS. *
10 0010 1 * ALL RIGHTS RESERVED. *
11 0011 1 *
12 0012 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED *
13 0013 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE *
14 0014 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER *
15 0015 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY *
16 0016 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY *
17 0017 1 * TRANSFERRED. *
18 0018 1 *
19 0019 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE *
20 0020 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT *
21 0021 1 * CORPORATION. *
22 0022 1 *
23 0023 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS *
24 0024 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL. *
25 0025 1 *
26 0026 1 *
27 0027 1 *****
28 0028 1
29 0029 1
30 0030 1 **
31 0031 1 FACILITY: FORTRAN compatibility Library
32 0032 1
33 0033 1 ABSTRACT:
34 0034 1
35 0035 1 FORTRAN compatibility error set and test and condition handler.
36 0036 1 FOR$COM_HANDLE performs default error handling
37 0037 1 which creates an environment compatible with the PDP-11
38 0038 1 FORTRAN IV-PLUS error environment.
39 0039 1 This environment is only created if the user has a:
40 0040 1 CALL ERRST or CALL ERRSET in his program.
41 0041 1
42 0042 1 ENVIRONMENT: User Mode - AST re-entrant
43 0043 1
44 0044 1 AUTHOR: Thomas N. Hastings, CREATION DATE: 14-Dec-1977
45 0045 1
46 0046 1 MODIFIED BY:
47 0047 1
48 0048 1 Thomas N. Hastings, 14-Dec-1977: VERSION 0
49 0049 1 [Previous edit history removed. SBL 11-Nov-1980]
50 0050 1 1-13 - Change file name to COMSETST.B32, and change the names of
51 0051 1 the REQUIRE files similarly. JBS 14-NOV-78
52 0052 1 1-014 - Update the copyright notice. JBS 16-NOV-78
53 0053 1 1-015 - Change the REQUIRE file names from FOR... to OTS... JBS 07-DEC-78
54 0054 1 1-016 - Add error numbers 49-57. SBL 09-Feb-1979
55 0055 1 1-017 - Rename errors 36, 50. SBL 6-Apr-79
56 0056 1 1-018 - Change error 93 to allow continuation. SBL 25-May-1979
57 0057 1 1-019 - Correct a typo in the PSECT declaration that was not caught by

```

```

: 58      0058 1  !      earlier BLISS compilers. Also, remove unnecessary REQUIRE
: 59      0059 1  !      files and put through PRETTY. JBS 06-SEP-1979
: 60      0060 1  ! 1-020 - Change MTH$K_SINCOSSIG to MTH$K_SIGLOSMAT. JBS 19-SEP-1979
: 61      0061 1  ! 1-021 - Add handling of floating faults. Update names of error
: 62      0062 1  !      codes to match latest list. SBL 20-Nov-1979
: 63      0063 1  ! 1-022 - Add errors 17-19. SBL 11-Nov-1980
: 64      0064 1  ! 1-023 - Don't change severity of non-FORTRAN errors. SPR 11-46349
: 65      0065 1  !      SBL 16-June-1982
: 66      0066 1  ! 1-024 - Add call to FOR$$INIT_ERRSET to notify shared RTL that it should
: 67      0067 1  !      record errors in our tables. SBL 29-Jun-1983
: 68      0068 1  ! --
: 69      0069 1  !
: 70      0070 1  !<BLF/PAGE>
```

```

72 0071 1 |
73 0072 1 | SWITCHES:
74 0073 1 |
75 0074 1 |
76 0075 1 | SWITCHES ADDRESSING_MODE (EXTERNAL = GENERAL, NONEXTERNAL = WORD_RELATIVE);
77 0076 1 |
78 0077 1 |
79 0078 1 | TABLE OF CONTENTS:
80 0079 1 |
81 0080 1 |
82 0081 1 | FORWARD ROUTINE
83 0082 1 |     COM_STARTUP,           ! Compatibility startup routine
84 0083 1 |     COM_HANDLER,         ! FORTRAN compatibility error handler
85 0084 1 |     RECORD_ERROR: NOVALUE; ! Record an error number
86 0085 1 |
87 0086 1 |
88 0087 1 | INCLUDE FILES:
89 0088 1 |
90 0089 1 |
91 0090 1 | LIBRARY 'RTLSTARLE';      ! STARLET library for macros and symbols
92 0091 1 |
93 0092 1 | REQUIRE 'RTLIN:COMEST';  ! Common ERRST, ERRSET macros
94 0161 1 |
95 0162 1 | REQUIRE 'RTLML:FORERR';  ! FORTRAN error numbers
96 0230 1 |
97 0231 1 | REQUIRE 'RTLIN:RTLPSECT'; ! Define DECLARE_PSECTS macro
98 0326 1 |
99 0327 1 | REQUIRE 'RTLML:MTHERR';  ! Math library error numbers
100 0343 1 |
101 0344 1 |
102 0345 1 | MACROS:
103 0346 1 |
104 0347 1 |     NONE
105 0348 1 |
106 0349 1 | EQUATED SYMBOLS:
107 0350 1 |
108 0351 1 |
109 0352 1 | LITERAL
110 0353 1 |     S_SEVERITY = 3,       ! size of condition value severity field
111 0354 1 |     K_ERROR_LIMIT = 15;  ! initial image error count limit
112 0355 1 |
113 0356 1 |
114 0357 1 | OWN STORAGE:
115 0358 1 |
116 0359 1 |     NONE
117 0360 1 |
118 0361 1 | PSECT DECLARATIONS:
119 0362 1 |
120 0363 1 |
121 0364 1 | PSECT
122 0365 1 |     PLIT = LIB$INITIALIZE ( READ, NOWRITE, NOEXECUTE, NOSHARE, NOPIC, CONCATENATE, GLOBAL, ALIGN (2),
123 0366 1 |     ADDRESSING_MODE (GENERAL));
124 0367 1 |
125 0368 1 |
126 0369 1 | !+ Make LIB$INITIALIZE PSECT contribution so LIBINITIALIZE will call
127 0370 1 | ! COM_STARTUP which will setup default handler and co-routine back.
128 0371 1 | ! Don't use LIB$INITIALIZD_ PSECT, since that for Common Run-time library standard features.

```

```
129 0372 1 !-
130 0373 1
131 0374 1 BIND
132 0375 1 VECT = UPLIT (COM_STARTUP);
133 0376 1
134 0377 1 !+
135 0378 1 ! Now declare usual PSECTS
136 0379 1 !-
137 0380 1 DECLARE_PSECTS (F4PCOMPAT); ! declare PSECTS for F4PCOMPAT$ facility
138 0381 1 ! Keep separate from sharable library
139 0382 1
140 0383 1 GLOBAL
141 0384 1 COM$ERRORCOUNT : INITIAL (K_ERROR_LIMIT) VOLATILE; ! Image error count limit.
142 0385 1
143 0386 1 ! Decremented by COM_HANDLER.
144 0387 1 ! EXIT on continuable error if 0.
145 0388 1 !+
146 0389 1 Define FORTRAN compatibility error byte table
147 0390 1 One byte per FORTRAN error number.
148 0391 1 !-
149 0392 1 Define macro to generate byte data for initializing OWN COM$ERR_TAB.
150 0393 1 !-
151 0394 1
152 0395 1 MACRO
153 0396 1 A (OCCURRED, ! error occurred since last CALL ERTST
154 0397 1 EREQ_ALLOW, ! ERR= allowed
155 0398 1 CONT_ALLOW, ! continue allowed
156 0399 1 LOG, ! print error message
157 0400 1 CONT_TYPE, ! 1 = take ERR= if present, else EXIT; 0 = continue
158 0401 1 COUNT, ! count against image limit
159 0402 1 CONTINUE ! 1 = continuable (or ERR=), 0 = EXIT
160 0403 1 ) =
161 0404 1 (OCCURRED^7 + EREQ_ALLOW^6 + CONT_ALLOW^5 +
162 0405 1 LOG^3 + CONT_TYPE^2 + COUNT^1 + CONTINUE) %;
163 0406 1
164 0407 1 !+
165 0408 1 ! Statically allccate a byte table, one byte for each FORTRAN error number
166 0409 1 ! Each byte contains the error control status bits.
167 0410 1 !-
168 0411 1
169 0412 1 GLOBAL
170 0413 1 COM$ERR_TAB : RLOCKVECTOR [FOR$K_MAX_ERR + 1, 1, BYTE] INITIAL (BYTE (
171 0414 1
172 0415 1 !+
173 0416 1 O C R O L C C
174 0417 1 C E R O G U N C
175 0418 1 U R E N G T N T
176 0419 1 D I A A T C
177 0420 1 R E L L T Y C
178 0421 1 E A L L T Y N
179 0422 1 D L L O P E I
180 0423 1 W O O W F U
181 0424 1 !-
182 0425 1
183 0426 1
184 0427 1
185 0428 1 0, ! 0
```

186	0429	1	A(0,1,0,1,1,1,1),	NOTFORSPE,	1	! NOT A FORTRAN-SPECIFIC ERROR
187	0430	1		Used as a catch-all error number		
188	0431	1		for error other than FOR\$ and MTH\$		
189	0432	1	REP 17-1-1 OF BYTE (0),	skip 2-16		
190	0433	1	A(0,1,0,1,1,1,1),	SYNERRNAM,	17	! SYNTAX ERROR IN NAMELIST
191	0434	1	A(0,1,0,1,1,1,1),	TOOMANVAL,	18	! TOO MANY VALUES IN NAMELIST
192	0435	1	A(0,1,0,1,1,1,1),	INVREFVAR,	19	! INVALID REFERENCE TO VARIABLE
193	0436	1	A(0,1,0,1,1,1,1),	REWERR,	20	! REWIND ERROR
194	0437	1	A(0,1,0,1,1,1,1),	DUPFILSPE,	21	! DUPLICATE FILE SPECIFICATIONS
195	0438	1	A(0,1,0,1,1,1,1),	INPRECTOO,	22	! INPUT RECORD TOO LONG
196	0439	1	A(0,1,0,1,1,1,1),	BACERR,	23	! BACKSPACE ERROR
197	0440	1	A(0,1,0,1,1,1,1),	ENDDURREA,	24	! END-OF-FILE DURING READ
198	0441	1	A(0,1,0,1,1,1,1),	RECNUMOUT,	25	! RECORD NUMBER OUTSIDE RANGE
199	0442	1	A(0,1,0,1,1,1,1),	OPEDEFREQ,	26	! OPEN OR DEFINEFILE REQUIRED TO SPECIFY DIRECT OR K
200	0443	1	A(0,1,0,1,1,1,1),	TOOMANREC,	27	! TOO MANY RECORDS IN I/O STATEMENT
201	0444	1	A(0,1,0,1,1,1,1),	CLOERR,	28	! CLOSE ERROR
202	0445	1	A(0,1,0,1,1,1,1),	FILNOTFOU,	29	! FILE NOT FOUND
203	0446	1	A(0,1,0,1,1,1,1),	OPEFAI,	30	! OPEN FAILURE
204	0447	1	A(0,1,0,1,1,1,1),	MIXFILACC,	31	! MIXED FILE ACCESS MODES
205	0448	1	A(0,1,0,1,1,1,1),	INVLOGUNI,	32	! INVALID LOGICAL UNIT NUMBER
206	0449	1	A(0,1,0,1,1,1,1),	ENDFILERR,	33	! ENDFILE ERROR
207	0450	1	A(0,1,0,1,1,1,1),	UNIALROPE,	34	! UNIT ALREADY OPEN
208	0451	1	A(0,1,0,1,1,1,1),	SEGRECFOR,	35	! SEGMENTED RECORD FORMAT ERROR
209	0452	1	A(0,1,0,1,1,1,1),	ATTACNON,	36	! ATTEMPT TO ACCESS NON-EXISTENT RECORD
210	0453	1	A(0,1,0,1,1,1,1),	INCRECLEN,	37	! INCONSISTENT RECORD LENGTH
211	0454	1	A(0,1,0,1,1,1,1),	ERRDURWRI,	38	! ERROR DURING WRITE
212	0455	1	A(0,1,0,1,1,1,1),	ERRDURREA,	39	! ERROR DURING READ
213	0456	1	A(0,1,0,1,1,1,1),	RECIO OPE,	40	! RECURSIVE I/O OPERATION
214	0457	1	A(0,1,0,1,1,1,1),	INSVIRMEM,	41	! INSUFFICIENT VIRTUAL MEMORY
215	0458	1	A(0,1,0,1,1,1,1),	NO SUCDEV,	42	! NO SUCH DEVICE
216	0459	1	A(0,1,0,1,1,1,1),	FICNAMSPE,	43	! FILE NAME SPECIFICATION ERROR
217	0460	1	A(0,1,0,1,1,1,1),	INCRECTYP,	44	! INCONSISTENT RECORD TYPE
218	0461	1	A(0,1,0,1,1,1,1),	KEYVALERR,	45	! KEYWORD VALUE ERROR IN OPEN STATEMENT
219	0462	1	A(0,1,0,1,1,1,1),	INCOPECLO,	46	! INCONSISTENT OPEN/CLOSE PARAMETERS
220	0463	1	A(0,1,0,1,1,1,1),	WRIREAFIL,	47	! WRITE TO READYONLY FILE
221	0464	1	A(0,1,0,1,1,1,1),	INVARGFOR,	48	! INVALID ARGUMENT TO FORTRAN I/O LIBRARY
222	0465	1	A(0,1,0,1,1,1,1),	INVKEYSPE,	49	! INVALID KEY SPECIFICATION
223	0466	1	A(0,1,0,1,1,1,1),	INCKEYCHG,	50	! INCONSISTENT KEY CHANGE OR DUPLICATE KEY
224	0467	1	A(0,1,0,1,1,1,1),	INCFILORG,	51	! INCONSISTENT FILE ORGANIZATION
225	0468	1	A(0,1,0,1,1,1,1),	SPERECLOC,	52	! SPECIFIED RECORD LOCKED
226	0469	1	A(0,1,0,1,1,1,1),	NO CURREC,	53	! NO CURRENT RECORD
227	0470	1	A(0,1,0,1,1,1,1),	REWRITERR,	54	! REWRITE ERROR
228	0471	1	A(0,1,0,1,1,1,1),	DELERR,	55	! DELETE ERROR
229	0472	1	A(0,1,0,1,1,1,1),	UNLERR,	56	! UNLOCK ERROR
230	0473	1	A(0,1,0,1,1,1,1),	FINERR,	57	! FIND ERROR
231	0474	1	0,	Unused: 58		
232	0475	1	A(0,1,1,1,1,0,1),	LISIO SYN,	59	! LIST-DIRECTED I/O SYNTAX ERROR
233	0476	1	A(0,1,0,1,1,1,1),	INFFORLOO,	60	! INFINITE FORMAT LOOP
234	0477	1	A(0,1,1,1,1,1,1),	FORVARMIS,	61	! FORMAT/VARIABLE-TYPE MISMATCH
235	0478	1	A(0,1,0,1,1,1,1),	SYNERRFOR,	62	! SYNTAX ERROR IN FORMAT
236	0479	1	A(0,1,1,1,0,0,1),	OUTCONERR,	63	! OUTPUT CONVERSION ERROR
237	0480	1	A(0,1,1,1,1,1,1),	INPCONERR,	64	! INPUT CONVERSION ERROR
238	0481	1				
239	0482	1	0,	skip 65		
240	0483	1				
241	0484	1	A(0,1,0,1,1,1,1),	OUTSTAOVE,	66	! OUTPUT STATEMENT OVERFLOWED RECORD
242	0485	1	A(0,1,0,1,1,1,1),	INPSTAREQ,	67	! INPUT STATEMENT REQUIRED TOO MUCH DATA

```

: 243      0486 1      A(0,1,1,1,1,1),      ! VFEVALERR,    68      ! VARIABLE FORMAT EXPRESSION VALUE ERROR
: 244      0487 1
: 245      0488 1      0,                      ! skip 69
: 246      0489 1
: 247      0490 1      A(0,0,1,1,0,1,1),      ! INTOVF,      70      ! INTEGER OVERFLOW
: 248      0491 1      A(0,0,1,1,0,1,1),      ! INTZERDIV,   71      ! INTEGER ZERO DIVIDE
: 249      0492 1      A(0,0,1,1,0,1,1),      ! FLOOVE,      72      ! FLOATING OVERFLOW
: 250      0493 1      A(0,0,1,1,0,1,1),      ! FLOZERDIV,   73      ! FLOATING ZERO DIVIDE
: 251      0494 1      A(0,0,1,1,0,1,1),      ! FLOUND,      74      ! FLOATING UNDERFLOW
: 252      0495 1
: 253      0496 1      0,                      ! skip 75
: 254      0497 1
: 255      0498 1      A(0,0,1,1,0,1,1),      ! DECSTROVE,   76      ! DECIMAL STRING OVERFLOW
: 256      0499 1      A(0,0,1,1,0,1,1),      ! ARRREFOUT,   77      ! ARRAY REFERENCE OUTSIDE ARRAY
: 257      0500 1
: 258      0501 1      0,0,                  ! skip 78, 79
: 259      0502 1
: 260      0503 1      A(0,0,0,1,0,1,1),      ! WRONUMARG,   80      ! wrong number of arguments
: 261      0504 1      A(0,0,0,1,0,1,1),      ! INVARGMTH,   81      ! invalid argument to math library
: 262      0505 1      A(0,0,1,1,0,1,1),      ! UNDEXP,      82      ! undefined exponentiation
: 263      0506 1      A(0,0,1,1,0,1,1),      ! LOGZERNEG,   83      ! logarithm of zero or negative value
: 264      0507 1      A(0,0,1,1,0,1,1),      ! SQUROONEG,   84      ! square root of negative value
: 265      0508 1
: 266      0509 1      0, 0,                ! Skip 85 and 86 (used on PDP-11),
: 267      0510 1
: 268      0511 1      A(0,0,1,1,0,1,1),      ! SIGLOSMAT,   87      ! significance lost in math library
: 269      0512 1      A(0,0,1,1,0,1,1),      ! FLOOVEMAT,   88      ! floating overflow in math library
: 270      0513 1      A(0,0,1,1,0,1,1),      ! FLOUNDMAT,   89      ! floating underflow in math library
: 271      0514 1
: 272      0515 1      REP 93-89-1 OF BYTE (0),      ! skip 90-92
: 273      0516 1
: 274      0517 1
: 275      0518 1      A (0, 0, 1, 1, 0, 1, 1)); ! ADJARRDIM,   93      ! ADJUSTABLE ARRAY DIMENSION ERROR
: 276      0519 1
: 277      0520 1
: 278      0521 1      ! EXTERNAL REFERENCES:
: 279      0522 1
: 280      0523 1
: 281      0524 1      EXTERNAL ROUTINE
: 282      0525 1      FOR$ERRSNS_SAV : NOVALUE,      ! Save FORTRAN error info for ERRSNS
: 283      0526 1      ! No logical unit associated with it.
: 284      0527 1      FOR$INIT_ERRSET: NOVALUE,      ! Pass recording routine addr
: 285      0528 1      LIB$FIXUP_FLT,              ! Fixup reserved floating operands
: 286      0529 1      LIB$SIM_TRAP,                ! Simulate floating traps
: 287      0530 1      LIB$INITIALIZE;              ! Cause LIB$INITIALIZE to be
: 288      0531 1
: 289      0532 1      ! Linked in so called before main program
: 290      0533 1      ! causes COM$STARTUP to be called
: 291      0534 1      ! before main program.

```



```

293 0535 1 ROUTINE COM_STARTUP (           ! Initialize FORTRAN compatibility ERRST, ERRSET support
294 0536 1     CO_ROUT_INIT,           ! Adr. of co-routine entry in LIB$INITIALIZE
295 0537 1     CLT_CO_ROUT)           ! Adr. of CLI co-routine
296 0538 1     =                       ! Value returned is main program completion code
297 0539 1
298 0540 1 !++
299 0541 1 FUNCTIONAL DESCRIPTION:
300 0542 1
301 0543 1     It establishes a ERRST/ERRSET default handler.
302 0544 1     Next, it calls FOR$$INIT_ERRSET, a routine in the shared
303 0545 1     RTL, to give it the address of RECORD_ERROR. This will be called
304 0546 1     whenever an I/O error occurs that is trapped by error handling
305 0547 1     so that we know about all errors, not just the untrapped ones.
306 0548 1     Then it performs a co-routine call-back to LIB$INITIALIZE
307 0549 1     which keeps the handler on the stack before the main program.
308 0550 1
309 0551 1 FORMAL PARAMETERS:
310 0552 1
311 0553 1     co_rout_init           Adr. of a procedure to be called in order
312 0554 1                     to affect a co-routine linkage with the
313 0555 1                     caller (LIB$INITIALIZE).
314 0556 1     cli_co_rout           Adr. of CLI co-routine
315 0557 1     ..                   Any other arguments passed to main programs
316 0558 1
317 0559 1 IMPLICIT INPUTS:
318 0560 1
319 0561 1     NONE
320 0562 1
321 0563 1 IMPLICIT OUTPUTS:
322 0564 1
323 0565 1     NONE
324 0566 1
325 0567 1 ROUTINE VALUE:
326 0568 1 COMPLETION CODES:
327 0569 1
328 0570 1     The completion code returned by the main program
329 0571 1
330 0572 1 SIDE EFFECTS:
331 0573 1
332 0574 1     Sets up a handler and does a co-routine call-back using co_rout_init.
333 0575 1 --
334 0576 1
335 0577 2 BEGIN
336 0578 2
337 0579 2 ENABLE
338 0580 2     COM_HANDLER;           ! Establish language independent handler
339 0581 2
340 0582 2 !+
341 0583 2     Call FOR$$INIT_ERRSET and pass it the address of RECORD_ERROR. The
342 0584 2     shared RTL will call RECORD_ERROR whenever it finds an I/O error that
343 0585 2     is to be unwound, so that it never gets to our handler. Therefore,
344 0586 2     our table always knows which errors have occurred.
345 0587 2     -
346 0588 2
347 0589 2 FOR$$INIT_ERRSET (RECORD_ERROR);
348 0590 2
349 0591 2 !+

```

```

: 350      0592  2  ! Perform co-routine call-back to caller (LIB$INITIALIZE)
: 351      0593  2  ! which will call any other library and user initialization procedures
: 352      0594  2  ! and then call the main program.
: 353      0595  2  ! When the main program return, return its completion status to caller
: 354      0596  2  ! which will get back to debugger or CLI.
: 355      0597  2  !
: 356      0598  2  !
: 357      0599  2  RETURN (.CO_ROUT_INIT) ();
: 358      0600  1  END;

```

! End of COM_STARTUP routine

.TITLE COM\$ERRSET_TST FORTRAN compatibility error set
and test

.IDENT \1-024\

.PSECT _F4PCOMPAT\$DATA,NOEXE, PIC,2

0000000F 0000 COM\$ERRORCOUNT::

4F 00 00004 COM\$ERR_TAB::

```

4F 4F 4F 4F 4F 4F 4F 4F 4F 4F 4F 4F 4F 4F 4F 4F 00# 00006
4F 4F 4F 4F 4F 4F 4F 4F 4F 4F 4F 4F 4F 4F 4F 4F 00015
6F 4F 6D 00 4F 4F 4F 4F 4F 4F 4F 4F 4F 4F 4F 4F 00024
2B 00 2B 2B 2B 2B 2B 00 6F 4F 4F 00 6F 69 4F 4F 00033
2B 2B 2B 00 00 2B 2B 2B 0B 0B 00 00 2B 00042
2B 2B 2B 00 00 2B 2B 2B 0B 0B 00 00 2B 00051

```

```

.LONG 15
.BYTE 0, 79
.BYTE 0[15]
.BYTE 79, 79, 79, 79, 79, 79, 79, 79, 79, 79, 79, -
79, 79, 79, 79, 79, 79, 79, 79, 79, 79, 79, -
79, 79, 79, 79, 79, 79, 79, 79, 79, 79, 79, -
79, 79, 79, 79, 79, 79, 79, 79, 79, 79, 79, -
79, 0, 109, 79, 111, 79, 105, 111, 0, 79, -
79, 111, 0, 43, 43, 43, 43, 43, 0, 43, -
43, 0, 0, 11, 11, 43, 43, 43, 0, 0, 43, -
43, 43

```

00# 0005E
2B 00061

.PSECT LIB\$INITIALIZE,NOWRT,NOEXE, GBL,2

00000000' 00000 P.AAA: .ADDRESS COM_STARTUP

VECT=

```

.P.AAA
.EXTRN FOR$ERRSNS SAV
.EXTRN FOR$INIT_ERRSET
.EXTRN LIB$FIXUP_FLT, LIB$SIM_TRAP
.EXTRN LIB$INITIALIZE

```

.PSECT _F4PCOMPAT\$CODE,NOWRT, SHR, PIC,2

0000 00000 COM_STARTUP:

```

6D 0011 CF DE 00002 .WORD Save nothing
0000V CF 9F 00007 MOVAL 1$, (FP)
00000000G 00 01 FB 0000B PUSHAB RECORD ERROR
04 BC 00 FB 00012 CALLS #1, FOR$INIT_ERRSET
04 04 00016 CALLS #0, @CO_ROUT_INIT
0000 00017 1$: RET
7E D4 00019 .WORD Save nothing
5E DD 0001B CLRL -(SP)
0000V 7E 04 AC 7D 0001D PUSHL SP
0000V CF 03 FB 00021 MOVQ 4(AP), -(SP)
CALLS #3, COM_HANDLER

```

```

: 0535
: 0577
: 0589
: 0599
: 0600
: 0577
:

```

COM\$ERRSET_TST FORTRAN compatibility error set and test
1-024

B 3
16-Sep-1984 00:11:31
14-Sep-1984 12:31:34

VAX-11 Bliss-32 V4.0-742
[FORRTL.SRC]COMSETST.B32;1

Page 9
(3)

04 00026 RET

; Routine Size: 39 bytes, Routine Base: _F4PCOMPAT\$CODE + 0000

```

360 0601 1 ROUTINE COM_HANDLER (          | FORTRAN compatibility default error handler
361 0602 1     SIG_ARGS_ADR,                |   Adr. of signal arg list
362 0603 1     MCH_ARGS_ADR)                |   Adr. of mechanism arg list
363     =                                |   Value is that of a condition handler
364 0605 1
365 0606 1
366 0607 1 ++
367 0608 1   FUNCTIONAL DESCRIPTION:
368 0609 1     COM_HANDLER performs default error handling
369 0610 1     which creates an environment compatible with the PDP-11
370 0611 1     FORTRAN IV-PLUS error environment.
371 0612 1     This environment is only created if the user has a:
372 0613 1     The actions taken depends on the conditon code being signaled:
373 0614 1
374 0615 1     SSS_UNWIND      Just return and keep unwinding.
375 0616 1     SSS_SSFAIL      Return SSS_CONTINUE so O.S. caller
376 0617 1                 can check error status at call site.
377 0618 1     SSS_ROPRAND      Fixup floating/double operand by calling LIB$FIXUP FLT.
378 0619 1     SSS_FLTOVF F...SS$ FLTUND F Convert to trap by calling LIB$SIM_TRAP.
379 0620 1     SSS_INTTOVF...SS$ SOBRNG Convert to FORTRAN error number from
380 0621 1                 arithmetic trap condition value.
381 0622 1
382 0623 1     If a math error which returns a reserved operand, change it to +0.0 and continue
383 0624 1     execution with no message printed.
384 0625 1     If not a FOR$ error, save error info for ERRSNS
385 0626 1     (If a FOR$ error, FOR$$ERRSNS_SAV already called when signaled).
386 0627 1     Determine FORTRAN error number from condition value.
387 0628 1     Use 1 (NOT FORTRAN SPECIFIC ERROR) if not a FORTRAN specific error.
388 0629 1     Flag that error number has had an occurrence (for ERRIST)
389 0630 1     If error can never continue or ERR=, set severity to SEVERE and resignal (which EXITs).
390 0631 1     If error is ERR= type, set severity to SEVERE and resignal (which EXITs),
391 0632 1     since ERR= transfer would already have taken place if present.
392 0633 1     If error wants to be counted, decrement image error count.
393 0634 1     If exceed limit, set severity to SEVERE and resignal (which EXITs).
394 0635 1     If error is to continue, set severity to ERROR, else to SEVERE.
395 0636 1     If error is to be logged, resignal else continue.
396 0637 1
397 0638 1   FORMAL PARAMETERS:
398 0639 1
399 0640 1     SIG_ARGS_ADR.ml.ra   Adr. of signal arg list
400 0641 1     MCH_ARGS_ADR.ml.ra   Adr. of mechanism arg list
401 0642 1
402 0643 1   IMPLICIT INPUTS:
403 0644 1
404 0645 1     NONE
405 0646 1
406 0647 1   IMPLICIT OUTPUTS:
407 0648 1
408 0649 1     Decrements OWN count of no. of continuable error remaining before EXIT (ERROR_LIMIT).
409 0650 1
410 0651 1   ROUTINE VALUE:
411 0652 1   COMPLETION CODES:
412 0653 1
413 0654 1     SSS_CONTINUE if signaled code was SSS_SSFAIL or error is to be continued without logging.
414 0655 1     SSS_RESIGNAL otherwise.
415 0656 1
416 0657 1   SIDE EFFECTS:

```

```
417 0658 1 |
418 0659 1 |   Signaled condition value is changed to ERROR or SEVERE
419 0660 1 |   in order to cause image to continue or EXIT after printing message.
420 0661 1 | --
421 0662 1 |
422 0663 2 | BEGIN
423 0664 2 |
424 0665 2 | LOCAL
425 0666 2 |   FORT_ERR_NO;           ! FORTRAN error number
426 0667 2 |
427 0668 2 | MAP
428 0669 2 |   SIG_ARGS_ADR : REF BLOCK [8, BYTE], ! SIGNAL arg list
429 0670 2 |   MCH_ARGS_ADR : REF BLOCK [5, BYTE]; ! mechanism arg list
430 0671 2 |
431 0672 2 | +
432 0673 2 | | Check for unwinding since handler gets called when it does an unwind.
433 0674 2 | | Just return to continue unwinding, since nothing to do.
434 0675 2 | |
435 0676 2 | |
436 0677 2 | IF .SIG_ARGS_ADR [CHF$L_SIG_NAME] EQL SSS_UNWIND THEN RETURN SSS_NORMAL;
437 0678 2 |
438 0679 2 | +
439 0680 2 | | Check if this is a system service failure exception (i.e., SSFAIL mode
440 0681 2 | | set by user in violation of modularity standards). If yes just return
441 0682 2 | | SSS_CONTINUE status to condition handling facility which will return to the
442 0683 2 | | OTS-call-site to operating system which will check error code returned in R0.
443 0684 2 | |
444 0685 2 | |
445 0686 2 | IF .SIG_ARGS_ADR [CHF$L_SIG_NAME] EQL SSS_SSFAIL THEN RETURN SSS_CONTINUE;
446 0687 2 |
447 0688 2 | +
448 0689 2 | | If signal is SSS_ROPRAND (reserved operand fault),
449 0690 2 | | go fixup if floating point by setting operand to +0.0.
450 0691 2 | | Then continue execution without a message. Do not convert this error
451 0692 2 | | to a FORTRAN error # since this error didn't happen on the PDP-11.
452 0693 2 | |
453 0694 2 | |
454 0695 2 | IF LIB$FIXUP_FLT (.SIG_ARGS_ADR, .MCH_ARGS_ADR) THEN RETURN SSS_CONTINUE;
455 0696 2 |
456 0697 2 | +
457 0698 2 | | If this is a floating fault, convert it to a trap by calling
458 0699 2 | | LIB$SIM_TRAP. LIB$SIM_TRAP will return SSS_RESIGNAL if this is
459 0700 2 | | not a floating fault, else it will simulate a trap and 'resignal'.
460 0701 2 | | If this happens, control will never return here. The return
461 0702 2 | | of SSS_CONTINUE is just to keep BLISS happy - it will never execute.
462 0703 2 | |
463 0704 2 | |
464 0705 2 | IF LIB$SIM_TRAP (.SIG_ARGS_ADR, .MCH_ARGS_ADR) THEN RETURN SSS_CONTINUE;
465 0706 2 |
466 0707 2 | +
467 0708 2 | | If this is a math library error which is about to return a reserved operand,
468 0709 2 | | convert R0/R1 to +0.0 in mechanism arg list so user program
469 0710 2 | | will get the same function value as on the PDP-11.
470 0711 2 | |
471 0712 2 | |
472 0713 2 | IF .STS_FAC_NO EQL MTH$K_FAC_NO
473 0714 2 | THEN
```

```

474 0715 2
475 0716 2
476 0717 2
477 0718 2
478 0719 2
479 0720 2
480 0721 2
481 0722 2
482 0723 2
483 0724 2
484 0725 2
485 0726 2
486 0727 2
487 0728 2
488 0729 2
489 0730 2
490 0731 2
491 0732 2
492 0733 2
493 0734 2
494 0735 2
495 0736 2
496 0737 2
497 0738 2
498 0739 2
499 0740 2
500 0741 2
501 0742 2
502 0743 2
503 0744 2
504 0745 2
505 0746 2
506 0747 2
507 0748 2
508 0749 2
509 0750 2
510 0751 2
511 0752 2
512 0753 2
513 0754 2
514 0755 2
515 0756 2
516 0757 2
517 0758 2
518 0759 2
519 0760 2
520 0761 2
521 0762 2
522 0763 2
523 0764 2
524 0765 2
525 0766 2
526 0767 2
527 0768 2
528 0769 2
529 0770 2
530 0771 2

CASE .STS_CODE FROM MTH$K_UNDEXP TO MTH$K_FLOOVEMAT OF
  SET
    [MTH$K_UNDEXP, MTH$K_LOGZERNEG, MTH$K_SQUROONEG, MTH$K_SIGLOSMAT, MTH$K_FLOOVEMAT] :
      BEGIN
        MCH_ARGS_ADR [CHF$MCH_SAVR0] = 0;
        MCH_ARGS_ADR [CHF$MCH_SAVR1] = 0;
      END;

  [INRANGE] :
  ;
  ! do nothing

  [OUTRANGE] :
  ;
  ! do nothing

  TES;

!+
! If this is an arithmetic trap error, map to FORTRAN error number.
! Condition values are: INTOVF, INTDIV, FLTOVF, FLTDIV, FLTUND, DEVOFV, SUBRNG.
! and are assumed to be consecutive and in ascending order according
! to hardware trap number. Trap codes 1:5 map to error 70:74
! and trap code 7 maps to error 77. Other codes map to error 1.
!-

IF .STS_COND_ID GEQU SSS_INTOVF^-S_SEVERITY AND .STS_COND_ID LEQU SSS_SUBRNG^-S_SEVERITY
THEN
  BEGIN
    BIND
      TRAN = UPLIT BYTE(70, 71, 72, 73, 74, 1, 77);

    MAP
      TRAN : VECTOR [7, BYTE];

    FORT_ERR_NO = .TRAN [.STS_COND_ID - (SSS_INTOVF^-S_SEVERITY)];
  END
ELSE
  !+
  ! Setup FORTRAN error number and check range for index into error table.
  ! If out of range, map to error 1 (NOT FORTRAN SPECIFIC ERROR)
  !-

  FORT_ERR_NO = (SELECTONEU .STS_FAC_NO OF
    SET
      [FOR$K_FAC_NO] : .STS_CODE;
      [MTH$K_FAC_NO] : .STS_CODE;
      [OTHERWISE] : FOR$K_NOTFORSPE;
    TES);
  ! FOR$ errors
  ! MTH$ errors
  ! other, use error code 1
  ! NOT FORTRAN SPECIFIC ERROR

  !+
  ! If error table has no bits set for this error, also map to 1
  ! since error is not defined.
  !-

IF .COMSSERR_TAB [.FORT_ERR_NO, B_EC_ANY] EQL 0 THEN FORT_ERR_NO = FOR$K_NOTFORSPE;

```

C
S
F
P
S
U

P
I
P
S
P
S
C
A
T
I
T
I
2

M
-
I
T
M

```

531 0772 2
532 0773 2
533 0774 2
534 0775 2
535 0776 2
536 0777 2
537 0778 2
538 0779 2
539 0780 2
540 0781 2
541 0782 2
542 0783 2
543 0784 2
544 0785 2
545 0786 2
546 0787 2
547 0788 2
548 0789 2
549 0790 2
550 0791 2
551 0792 2
552 0793 2
553 0794 2
554 0795 2
555 0796 2
556 0797 2
557 0798 2
558 0799 2
559 0800 2
560 0801 2
561 0802 2
562 0803 2
563 0804 2
564 0805 2
565 0806 2
566 0807 2
567 0808 2
568 0809 2
569 0810 2
570 0811 2
571 0812 2
572 0813 2
573 0814 2
574 0815 2
575 0816 2
576 0817 2
577 0818 2
578 0819 2
579 0820 2
580 0821 4
581 0822 4
582 0823 4
583 0824 3
584 0825 2
585 0826 2
586 0827 2
587 0828 2

!+
!+ Save error info for ERRSNS if this error is not FOR$ error.
!+ (FOR$ errors already called FOR$$ERRSNS_SAV when error was signaled).
!-
IF .STS_FAC_NO NEQ FOR$K_FAC_NO
THEN
FOR$$ERRSNS_SAV (.FORT_ERR_NO, 0, 0, 0,
.SIG_ARGS_ADR [CHF$L_STG_NAME]);

!+
!+ Flag that this FORTRAN error number happened for subsequent CALL ERRST
!-
COM$ERR_TAB [.FORT_ERR_NO, V_EC_OCCURRED] = 1;

!+
!+ If this error can never continue (or ERR=) or continue type
!+ is ERR= (if present in I/O statement), set severity to SEVERE and resignal.
!+ Note: if continue type is ERR=, user did not supply if control
!+ gets here to default condition handler setup before main program called.
!-
IF NOT .COM$ERR_TAB [.FORT_ERR_NO, V_EC_CONTINUE] OR .COM$ERR_TAB [.FORT_ERR_NO, V_EC_CONT_TYPE]
THEN
BEGIN
IF .STS_FAC_NO EQL FOR$K_FAC_NO
THEN
STS_SEVERITY = STS$K_SEVERE;
RETURN SS$RESIGNAL
END;

!+
!+ If this error wants to be counted, decrement image error count limit.
!-
IF .COM$ERR_TAB [.FORT_ERR_NO, V_EC_COUNT]
THEN
BEGIN
COM$ERRORCOUNT = .COM$ERRORCOUNT - 1;

!+
!+ If error limit goes to 0, set error severity
!+ to SEVERE and resignal so that image exits after printing message.
!-

IF .COM$ERRORCOUNT LEQ 0
THEN
BEGIN
STS_SEVERITY = STS$K_SEVERE;
RETURN SS$RESIGNAL
END;

END;

!+

```

```

: 588      0829 2      ! If here, error is to continue (V_EC_CONTINUE = 0),
: 589      0830 2      ! set severity to ERROR so continue if resigaled (ie if logged).
: 590      0831 2      !
: 591      0832 2      !
: 592      0833 2      STS_SEVERITY = STS$K_ERROR;
: 593      0834 2      !
: 594      0835 2      !+
: 595      0836 2      ! If this error is to be logged, log it by resignaling and continue image.
: 596      0837 2      ! Otherwise just continue image (without resignaling so no message).
: 597      0838 2      !
: 598      0839 2      !
: 599      0840 2      IF .COMSSERR_TAB [.FORT_ERR_NO, V_EC_LOG] THEN RETURN SSS_RESIGNAL ELSE RETURN SSS_CONTINUE
: 600      0841 2      !
: 601      0842 1      END;
                                ! End of COM_HANDLER
    
```

```

4D 01 4A 49 48 47 46 00027 P.AAB: .BYTE 70, 71, 72, 73, 74, 1, 77
                                TRAN=      P.AAB
    
```

```

                                003C 00000 COM_HANDLER:
                                .WORD      Save R2,R3,R4,R5
                                55 00000000' EF 9E 00002      MOVAB      COMSSERR_TAB, R5
                                53      04 AC D0 00009      MOVL      SIG_ARGS_ADR, R3
                                54      04 A3 9E 0000C      MOVAB      4(R3), R4
                                00000920 8F      64 D1 00011      CMPL      (R4), #2336
                                0000045C 8F      07 13 00018      BEQL      1$
                                00000000G 00      64 D1 0001A      CMPL      (R4), #1116
                                EB      03 12 00021 1$:      BNEQ      3$
                                00DF      31 00023 2$:      BRW      18$
                                52      08 AC D0 00026 3$:      MOVL      MCH_ARGS_ADR, R2
                                53      DD 0002A      PUSHL     R2
                                00000000G 00      53 DD 0002C      PUSHL     R3
                                EB      02 FB 0002E      CALLS     #2, LIB$FIXUP_FLT
                                52      DD 00038      PUSHL     R2
                                00000000G 00      53 DD 0003A      PUSHL     R3
                                DD      02 FB 0003C      CALLS     #2, LIB$SIM_TRAP
                                16      02 A4      0C      50 E8 00043      BLBS     R0, 2$
                                50      20 12 0004C      BNEQ     6$
                                0013      06 00000052 8F      03 EF 0004E      EXTZV    #3, #12, (R4), R0
                                0010      50 CF 00053      CASEL    R0, #82, #6
                                0010      0010 0005B 4$:      .WORD    5$-4$, -
                                0010      0013 00063      5$-4$, -
                                0010      0013 00063      5$-4$, -
                                0010      0013 00063      5$-4$, -
                                0010      0013 00063      6$-4$, -
                                0010      0013 00063      6$-4$, -
                                0010      0013 00063      5$-4$, -
                                0010      0013 00063      5$-4$
                                0C      03 11 00069      BRB      6$
                                0000008F 8F      64      19      0C      A2 7C 0006B 5$:      CLRQ     12(R2)
                                00000095 8F      64      19      03      03 ED 0006E 6$:      CMPZV    #3, #25, (R4), #143
                                18 1F 00077      BLSSU    7$
                                03      ED 00079      CMPZV    #3, #25, (R4), #149
    
```


50		64	19		0D 1A 00082	BGTRU	7\$		
			53	FEDC	03 EF 00084	EXTZV	#3, #25, (R4), R0		0749
					40 9A 00089	MOVZBL	TRAN-143[R0], FORT_ERR_NO		
18	02	A4	0C		1A 11 0008F	BRB	10\$		0739
					00 ED 00091	CMPZV	#0, #12, 2(R4), #24		0760
16	02	A4	0C		08 13 00097	BEQL	8\$		
					00 ED 00099	CMPZV	#0, #12, 2(R4), #22		0761
53		64	0C		07 12 0009F	BNEQ	9\$		
					03 EF 000A1	EXTZV	#3, #12, (R4), FORT_ERR_NO		
			53		03 11 000A6	BRB	10\$		
					01 D0 000A8	MOVL	#1, FORT_ERR_NO		0762
				6543	95 000AB	TSTB	COMSSERR_TAB[FORT_ERR_NO]		0771
					03 12 000AE	BNEQ	11\$		
18	02	A4	0C		01 D0 000B0	MOVL	#1, FORT_ERR_NO		
					00 ED 000B3	CMPZV	#0, #12, 2(R4), #24		0778
					0F 13 000B9	BEQL	12\$		
					64 DD 000BB	PUSHL	(R4)		0781
					7E 7C 000BD	CLRQ	-(SP)		0780
					7E D4 000BF	CLRL	-(SP)		
					53 DD 000C1	PUSHL	FORT_ERR_NO		
		00G	00		05 FB 000C3	CALLS	#5, FORSSERRSNS_SAV		
		50	53		55 C1 000CA	ADDL3	R5, FORT_ERR_NO, R0		0787
			60	80	8F 88 000CE	BISB2	#128, (R0)		
			04		60 E9 000D2	BLBC	(R0), 13\$		0796
		0A	60		02 E1 000D5	BBC	#2, (R0), 14\$		
18	02	A4	0C		00 ED 000D9	CMPZV	#0, #12, 2(R4), #24		0799
					1E 12 000DF	BNEQ	17\$		
					0C 11 000E1	BRB	15\$		0801
		0F	60		01 E1 000E3	BBC	#1, (R0), 16\$		0809
				FC	A5 D7 000E7	DECL	COMSSERRORCOUNT		0812
				FC	A5 D5 000EA	TSTL	COMSSERRORCOUNT		0819
					07 14 000ED	BGTR	16\$		
64		03	00		04 F0 000EF	INSV	#4, #0, #3, (R4)		0822
					09 11 000F4	BRB	17\$		0823
64		03	00		02 F0 000F6	INSV	#2, #0, #3, (R4)		0833
		06	60		03 E1 000FB	BBC	#3, (R0), 18\$		0840
			50	0918	8F 3C 000FF	MOVZWL	#2328, R0		
					04 00104	RET			
			50		01 D0 00105	MOVL	#1, R0		0842
					04 00108	RET			

; Routine Size: 265 bytes, Routine Base: _F4PCOMPAT\$CODE + 002E

```

: 603 0843 1 %SBTTL'RECORD_ERROR'
: 604 0844 1 ROUTINE RECORD_ERROR (
: 605 0845 1   ERROR_NUMBER
: 606 0846 1   ): NOVALUE =
: 607 0847 1
: 608 0848 1 !++
: 609 0849 1 ! FUNCTIONAL DESCRIPTION:
: 610 0850 1
: 611 0851 1   This routine is called by FOR$$ERRSNS_SAV in the shared RTL to
: 612 0852 1   record that an I/O error occurred. Because an error can get
: 613 0853 1   trapped by ERR=, END= or IOSTAT=, COM HANDLER doesn't see all
: 614 0854 1   errors. This mechanism ensures that ERRST will always know
: 615 0855 1   that an error occurred. Note that COM STARTUP passes the address
: 616 0856 1   of this routine to FOR$$INIT_ERRSET, which stores it for later use.
: 617 0857 1
: 618 0858 1 ! FORMAL PARAMETERS:
: 619 0859 1
: 620 0860 1   ERROR_NUMBER   - The FORTRAN error number of the error that
: 621 0861 1   occurred.
: 622 0862 1
: 623 0863 1 ! IMPLICIT INPUTS:
: 624 0864 1
: 625 0865 1   NONE
: 626 0866 1
: 627 0867 1 ! IMPLICIT OUTPUTS:
: 628 0868 1
: 629 0869 1   COM$$ERR_TAB
: 630 0870 1
: 631 0871 1 ! ROUTINE VALUE:
: 632 0872 1
: 633 0873 1   NONE
: 634 0874 1
: 635 0875 1 ! SIDE EFFECTS:
: 636 0876 1
: 637 0877 1   NONE
: 638 0878 1 !--
: 639 0879 1
: 640 0880 2   BEGIN
: 641 0881 2
: 642 0882 2   !+
: 643 0883 2   ! Flag that this FORTRAN error number happened for subsequent CALL ERRST
: 644 0884 2   !-
: 645 0885 2
: 646 0886 2   COM$$ERR_TAB [.ERROR_NUMBER, V_EC_OCCURRED] = 1;
: 647 0887 2
: 648 0888 2   RETURN;
: 649 0889 2
: 650 0890 1   END;                                     ! End of RECORD_ERROR
  
```

```

                                0000 0000 RECORD_ERROR:
                                .WORD   Save nothing
                                MOVAB   COM$$ERR_TAB, R0
04 BC40 50 00000000' EF 9E 00002    BISB2 #128, @ERROR_NUMBER[R0]
                                80 8F 88 00009
  
```

```

: 0844
: 0886
:
  
```

04 0000F RET

: 0890

: Routine Size: 16 bytes, Routine Base: _F4PCOMPAT\$CODE + 0137

: 651 0891 1 END ! End of module
: 652 0892 1
: 653 0893 0 ELUDOM

PSECT SUMMARY

Name	Bytes	Attributes
LIB\$INITIALIZE	4	NOVEC,NOWRT, RD ,NOEXE,NOSHR, GBL, REL, CON,NOPIC,ALIGN(2)
_F4PCOMPAT\$DATA	98	NOVEC, WRT, RD ,NOEXE,NOSHR, LCL, REL, CON, PIC,ALIGN(2)
_F4PCOMPAT\$CODE	327	NOVEC,NOWRT, RD , EXE, SHR, LCL, REL, CON, PIC,ALIGN(2)

Library Statistics

File	----- Symbols -----		Pages Mapped	Processing Time
	Total	Loaded Percent		
_\$255\$DUA28:[SYSLIB]STARLET.L32;1	9776	16 0	581	00:01.0

COMMAND QUALIFIERS

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

: Size: 320 code + 109 data bytes
: Run Time: 00:15.1
: Elapsed Time: 00:44.8
: Lines/CPU Min: 3543
: Lexemes/CPU-Min: 28222
: Memory Used: 117 pages
: Compilation Complete

