


```

CCCCCCCC 000000 PPPPPPP YY YY SSSSSSS EEEEEEEEE MM MM AAAAAA NN NN
CCCCCCCC 000000 PPPPPPP YY YY SSSSSSS EEEEEEEEE MM MM AAAAAA NN NN
CC        00    00 PP      PP YY YY SS        EE        MMMM MMMM AA AA NN NN
CC        00    00 PP      PP YY YY SS        EE        MMMM MMMM AA AA NN NN
CC        00    00 PP      PP YY YY SS        EE        MM  MM  MM  AA AA NNNN NN
CC        00    00 PP      PP YY YY SS        EE        MM  MM  MM  AA AA NNNN NN
CC        00    00 PPPPPPP YY YY SSSSSS  EEEEEEEE MMM  MM  MM  AA AA NN NN
CC        00    00 PPPPPPP YY YY SSSSSS  EEEEEEEE MMM  MM  MM  AA AA NN NN
CC        00    00 PP      PP YY YY SS        EE        MM  MM  MM  AAAAAAAAAA NN NNNN
CC        00    00 PP      PP YY YY SS        EE        MM  MM  MM  AAAAAAAAAA NN NNNN
CC        00    00 PP      PP YY YY SS        EE        MM  MM  MM  AA AA NN NN
CC        00    00 PP      PP YY YY SS        EE        MM  MM  MM  AA AA NN NN
CCCCCCCC 000000 PP      PP YY YY SSSSSSS EEEEEEEEE MM  MM  MM  AA AA NN NN
CCCCCCCC 000000 PP      PP YY YY SSSSSSS EEEEEEEEE MM  MM  MM  AA AA NN NN

```

```

LL        111111 SSSSSSS
LL        111111 SSSSSSS
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 111111 SSSSSSS
LLLLLLLLLL 111111 SSSSSSS

```

```

1 0001 0 MODULE copyseman ( ! Semantics of input/output spec combinations for COPY command
2 0002 0
3 0003 0 LANGUAGE (BLISS32),
4 0004 0 IDENT = 'V04-000'
5 0005 1 BEGIN
6 0006 1
7 0007 1
8 0008 1 *****
9 0009 1 *
10 0010 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY *
11 0011 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS. *
12 0012 1 * ALL RIGHTS RESERVED. *
13 0013 1 *
14 0014 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED *
15 0015 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE *
16 0016 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER *
17 0017 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY *
18 0018 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY *
19 0019 1 * TRANSFERRED. *
20 0020 1 *
21 0021 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE *
22 0022 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT *
23 0023 1 * CORPORATION. *
24 0024 1 *
25 0025 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS *
26 0026 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL. *
27 0027 1 *
28 0028 1 *
29 0029 1 *****
30 0030 1
31 0031 1 ++
32 0032 1 FACILITY: COPY Command
33 0033 1
34 0034 1 ABSTRACT:
35 0035 1
36 0036 1 This module contains a decision table that lists combinations of valid
37 0037 1 and invalid specifications to the COPY command, and associated semantic
38 0038 1 actions. Also contained is a routine that sets bits in the COPY status
39 0039 1 words and and values in RMS data blocks that cause these semantic actions to occur.
40 0040 1
41 0041 1 ENVIRONMENT:
42 0042 1
43 0043 1 VAX/VMS operating system, unprivileged user mode utility,
44 0044 1 operates at non-AST level.
45 0045 1
46 0046 1 --
47 0047 1 ++
48 0048 1
49 0049 1 AUTHOR: Carol Peters, CREATION DATE: 19 April 1978 19:31
50 0050 1
51 0051 1 REVISION HISTORY:
52 0052 1
53 0053 1 V03-002 TSK0002 Tamar Krichevsky 7-Feb-83
54 0054 1 Move external declarations from COPY.REQ to here. Change the
55 0055 1 names for the offsets in COPY$CLI_STATUS and COPY$SEM_STATUS to
56 0056 1 facilitate using the new CLI.
57 0057 1

```

: 58
: 59
: 60
: 61
: 62
: 63
: 64
: 65
: 66
: 67
: 68
: 69

0058 1 !
0059 1 !
0060 1 !
0061 1 !
0062 1 !
0063 1 !
0064 1 !
0065 1 !
0066 1 !
0067 1 !
0068 1 !
0069 1 !--

V03-001 TSK0001 Tamar Krichevsky 16-Mar-82
Explicit input version numbers should be preserved if the
output file specification is wild.

V02-003 WMC0003 Wayne Cardoza 3-Nov-1981
Always maximize version number when concatenating files unless
an explicit output version was specified.

V02-002 WMC0002 Wayne Cardoza 2-Nov-1981
Explicit input versions should preserve version numbers if
no output file spec is specified.

```

71 0070 1 |
72 0071 1 | Table of Contents
73 0072 1 |
74 0073 1 | FORWARD ROUTINE
75 0074 1 |   copy$semantics;           ! Determines the semantics of a COPY command
76 0075 1 |
77 0076 1 |
78 0077 1 |   Include files
79 0078 1 |
80 0079 1 |
81 0080 1 | LIBRARY 'SYSSLIBRARY:STARLET.L32';   ! Common VAX/VMS definitions
82 0081 1 | REQUIRE 'SRCS:COPYMSG.REQ';         ! Definition of macros to SIGNAL a message
83 0162 1 |
84 0163 1 |
85 0164 1 |   Global variables
86 0165 1 |
87 0166 1 |
88 0167 1 |
89 0168 1 |   Own variables
90 0169 1 |
91 0170 1 |
92 0171 1 |
93 0172 1 |   Macros
94 0173 1 |
95 0174 1 | MACRO
96 M 0175 1 |   or_op (x) [] =           ! OR operator to be used in iterative macro.
97   0176 1 |     OR %,
98   0177 1 |
99 M 0178 1 |   stored_mask (x) [] =    ! Makes a mask of named bit numbers.
100  0179 1 |     1 * (x) or_op (%REMAINING) stored_mask (%REMAINING) %;
101 0180 1 |
102 0181 1 |
103 0182 1 |   Equated symbols
104 0183 1 |
105 0184 1 | LITERAL
106 0185 1 |   mask_value               = 0,   ! Mask value in decision table entry.
107 0186 1 |   action_value             = 1,   ! Action label in decision table entry
108 0187 1 |   next_entry               = 2,   ! Next entry in decision table
109 0188 1 |
110 0189 1 |   mul_quiet_slip           = 0,   ! Create multiple files, match versions, slip quietl
111 0190 1 |   multiple_slip            = 1,   ! Create multiple files, match versions, report slip
112 0191 1 |   multiple_max              = 2,   ! Create multiple files, maximize version numbers
113 0192 1 |   max_or_new                = 3,   ! Maximize or use new version number
114 0193 1 |   concat_slip              = 4,   ! Concatenate input files, slip version numbers
115 0194 1 |   error_spec                = 5,   ! Error in combination of input and output specs
116 0195 1 |   no_action                 = 6;  ! No action to take
117 0196 1 |
118 0197 1 |
119 0198 1 |   External variables
120 0199 1 |
121 0200 1 | EXTERNAL
122 0201 1 |   copy$cli_status : $BLOCK,      ! Command line parse results
123 0202 1 |   copy$sem_status : $BLOCK      ! Status of input and output fields
124 0203 1 |
125 0204 1 |
126 0205 1 | REQUIRE
127 0206 1 |   'SRCS:COPY.REQ'

```

COPYSEMAN
V04-000

E 13
15-Sep-1984 23:42:04
14-Sep-1984 12:14:19

VAX-11 Bliss-32 V4.0-742
[COPY.SRC]COPYSEMAN.B32.1

Page 4
(2)

; 128 0207 1 ;

COPYSEMAN
V04-000

F 13
15-Sep-1984 23:42:04
15-Sep-1984 22:42:03

VAX-11 Bliss-32 V4.0-742
_S255\$DUA28:[COPY.SRC]VMSMAC.REQ;1

Page 5
(1)

: %PRINT:

File: VMSMAC.B32, Version V04-000, Edit 1, WWC, 09-JAN-1978

COPYSEMAN
V04-000

G 13
15-Sep-1984 23:42:04
14-Sep-1984 12:14:19

VAX-11 Bliss-32 V4.0-742
[COPY.SRC]COPYSEMAN.B32;1

Page 6
(2)

: 129
: 130
: 131

0662 1
0663 1 EXTERNAL ROUTINE
0664 1 copy\$inopn_err;

! Reports an error resulting from an OPEN of an inpu


```

133 0665 1
134 0666 1 The decision table below, DECISION TABLE, is made up of two WORD entries, of
135 0667 1 which the first word is a mask of bit settings, and the second is a case label
136 0668 1 associated with the mask.
137 0669 1
138 0670 1 Bit 0 16 31
139 0671 1 -----
140 0672 1 | mask | action |
141 0673 1 |-----|
142 0674 1
143 0675 1
144 0676 1
145 0677 1
146 0678 1 BIND
147 0679 1 decision_table = UPLIT WORD
148 0680 1 (
149 0681 1
150 0682 1
151 0683 1 Rule 1:
152 0684 1 Explicit wildcard output fields specified with the /CONCATENATE qualifier is an error.
153 0685 1
154 0686 1 stored_mask (wild_input_bit, wild_output_bit, concat_qual_bit), error_spec
155 0687 1
156 0688 1
157 0689 1 Rule 2:
158 0690 1 Specification of the /CONCATENATE qualifier with an output specification that contains no
159 0691 1 explicit name, type, and version number fields is an error.
160 0692 1
161 0693 1 . stored_mask (concat_qual_bit, no_output_spec_bit), error_spec
162 0694 1
163 0695 1
164 0696 1 Rule 3:
165 0697 1 A wildcard input version number specified with an output specification that contains no
166 0698 1 explicit name, type, and version number fields causes output files to be created whose
167 0699 1 version numbers may be lower than those of existing files.
168 0700 1
169 0701 1 . stored_mask (wild_inp_ver_bit, no_output_spec_bit), mul_quiet_slip
170 0702 1
171 0703 1
172 0704 1 Rule 4:
173 0705 1 Specification of both input wildcard version number and output wildcard version number
174 0706 1 quietly allows output files to be created whose version numbers are lower than those of
175 0707 1 existing files. Input version numbers are matched if possible. Overwrites usually fail.
176 0708 1
177 0709 1 . stored_mask (wild_inp_ver_bit, wild_out_ver_bit), mul_quiet_slip
178 0710 1
179 0711 1
180 0712 1 Rule 5:
181 0713 1 A wildcard output version number produces matching version numbers and warnings
182 0714 1 if an attempt is made to create an output file whose version number is lower
183 0715 1 than that of an existing file.
184 0716 1
185 0717 1 . stored_mask (wild_out_ver_bit), multiple_slip
186 0718 1
187 0719 1
188 0720 1 Rule 6:
189 0721 1 A wildcard input version number specified with wildcard output fields (but not wildcard

```

```

190 0722 1  output version number field) causes a warning to be reported if an attempt is made to
191 0723 1  create an output file with a version number lower than that of an existing file.
192 0724 1
193 0725 1  . stored_mask (wild_inp_ver_bit, wild_output_bit), multiple_slip
194 0726 1
195 0727 1
196 0728 1  Rule 7:
197 0729 1  Wildcard output fields and explicit wildcard version number field causes multiple
198 0730 1  files to be created, and a warning to be reported if an attempt is made to create
199 0731 1  an output file with a version number lower than that of an existing file.
200 0732 1
201 0733 1  . stored_mask (wild_output_bit, exp_out_ver_bit), multiple_slip
202 0734 1
203 0735 1  Rule 7a:
204 0736 1  Explicit input versions and no output spec should cause multiple files
205 0737 1  to be created with a warning to be reported if an attempt is made to create
206 0738 1  an output file with a version number lower than that of an existing file.
207 0739 1
208 0740 1  . stored_mask (exp_inp_ver_bit, no_output_spec_bit), multiple_slip
209 0741 1
210 0742 1
211 0743 1  Rule 7b:
212 0744 1  Explicit input versions and wild output spec should cause multiple files
213 0745 1  to be created with a warning to be reported if an attempt is made to create
214 0746 1  an output file with a version number lower than that of an existing file.
215 0747 1
216 0748 1  . stored_mask (exp_inp_v r_bit, wild_output_bit), multiple_slip
217 0749 1
218 0750 1
219 0751 1  Rule 8:
220 0752 1  Wildcard output fields (but not wildcard output version number field) causes a warning
221 0753 1  to be reported if an attempt is made to create an output file with a version number
222 0754 1  lower than that of an existing file. Output version numbers are maximized.
223 0755 1
224 0756 1  . stored_mask (wild_output_bit), multiple_max
225 0757 1
226 0758 1
227 0759 1  Rule 9:
228 0760 1  Absence of output file name, type, and version number specification always cause
229 0761 1  no concatenation and maximized version numbers.
230 0762 1
231 0763 1  . stored_mask (no_output_spec_bit), multiple_max
232 0764 1
233 0765 1
234 0766 1  Rule 10:
235 0767 1  An explicit output version number creates a concatenated output file whose version number
236 0768 1  may be lower than that of an existing file.
237 0769 1
238 0770 1  . stored_mask (exp_out_ver_bit), concat_slip
239 0771 1
240 0772 1
241 0773 1  Rule 11:
242 0774 1  An explicit input version number specified with no explicit output version number (wildcard
243 0775 1  or otherwise) creates a concatenated output file with a maximized or new version number if
244 0776 1  there are multiple input files. Preserve version number for a single input file.
245 0777 1
246 0778 1  . stored_mask (exp_inp_ver_bit, multiple_input_bit), max_or_new

```

COPYSEMAN
V04-000

J 13
15-Sep-1984 23:42:04
14-Sep-1984 12:14:19

VAX-11 Bliss-32 V4.0-742
[COPY.SRC]COPYSEMAN.B32;1

Page 9
(3)

```
: 247      0779 1      , stored_mask (wild_input_bit), max_or_new
: 248      0780 1      ; 0, no_action
: 249      0781 1      );
```

```

251 0782 1 GLOBAL ROUTINE copy$semantics (status, input_fab, output_fab) =
252 0783 1                                     ! Chooses semantics for COPY command
253 0784 1
254 0785 1 +-+
255 0786 1 Functional description:
256 0787 1
257 0788 1     This routine consults a decision table (DECISION_TABLE) to decide
258 0789 1     the appropriate semantics for a given input/output specification pair.
259 0790 1
260 0791 1     The status passed as an input parameter is compared with various mask
261 0792 1     settings in the decision table. If an item in the decision table matches
262 0793 1     the status, then an associated action is taken.
263 0794 1
264 0795 1     Actions set or turn off status bits, and/or write values to fields in
265 0796 1     the output FAB, NAM, and XAB blocks.
266 0797 1
267 0798 1 Calling sequence:
268 0799 1
269 0800 1     copy$semantics (status.ra, input_fab.ra, output_fab.ra)
270 0801 1
271 0802 1 Input parameters
272 0803 1
273 0804 1     status          - address of a status word
274 0805 1     input_fab      - address of the FAB associated with the current input file
275 0806 1     output_fab     - address of the FAB associated with the current output file
276 0807 1
277 0808 1 Implicit inputs
278 0809 1
279 0810 1     The NAM and XAB blocks associated with the input and output FABs.
280 0811 1
281 0812 1 Output parameters
282 0813 1
283 0814 1     none
284 0815 1
285 0816 1 Implicit outputs
286 0817 1
287 0818 1     Bits that may be written in the status word include the following:
288 0819 1
289 0820 1     Fields in the NAM blocks that may be written include the following:
290 0821 1
291 0822 1     Fields in the XAB blocks that may be written include the following:
292 0823 1
293 0824 1 Routine value
294 0825 1
295 0826 1     OK              - success
296 0827 1     ERROR          - error in combination of input and output specifications
297 0828 1
298 0829 1 Side effects
299 0830 1
300 0831 1     none
301 0832 1
302 0833 1 --
303 0834 1
304 0835 2 BEGIN
305 0836 2
306 0837 2 MAP
307 0838 2     status          : REF VECTOR [, WORD],          ! Pointer to the status word

```

```

308      0839      input_fab      : REF BLOCK [, BYTE],      ! Input FAB block
309      0840      output_fab    : REF BLOCK [, BYTE];      ! Output FAB block
310      0841
311      0842      LOCAL
312      0843      decis_tbl_ptr  : REF VECTOR [, WORD];      ! Pointer to current entry in DECISION_TABLE.
313      0844
314      0845
315      0846      Set up a pointer to the first entry in the decision table.
316      0847
317      0848
318      0849      decis_tbl_ptr = decision_table;      ! The address of the table addresses the first entry
319      0850
320      0851
321      0852      Start out without maximized file versions.
322      0853
323      0854
324      0855      output_fab [fab$v_mxv] = FALSE;
325      0856
326      0857
327      0858      Iteratively move through the decision table. For each entry, see if all the bits set
328      0859      in the mask are also set in the status word. If they are, exit from the
329      0860      loop with the decision table pointer pointing to that entry.
330      0861
331      0862      If no masks match the status word, return from this routine with an error status code.
332      0863
333      0864
334      0865      WHILE 1 DO      ! Loop once for each entry in the decision table.
335      0866      BEGIN
336      0867      IF (.status [0] AND .decis_tbl_ptr [mask_value])!      If the bits in the mask are also set in
337      0868      EQL .decis_tbl_ptr [mask_value]      !      the status word,
338      0869      THEN
339      0870      EXITLOOP      !      then exit from this comparing loop.
340      0871      ELSE
341      0872      decis_tbl_ptr = decis_tbl_ptr [next_entry]; !      Otherwise, move to the next table entry.
342      0873      IF .decis_tbl_ptr [mask_value] EQL 0      !      If this entry is null,
343      0874      THEN
344      0875      EXITLOOP;      !      then perform associated null action.
345      0876      END;      !      Else go through the loop again.
346      0877
347      0878
348      0879      Now take the action associated with the matched mask value.
349      0880
350      0881
351      0882      CASE .decis_tbl_ptr [action_value]      ! This case statement performs the appropriate
352      0883      FROM mul_quiet_slip TO no_action OF      !      action routine.
353      0884
354      0885      SET
355      0886
356      0887      [mul_quiet_slip]:      ! Multiple output files, slip version numbers silent
357      0888      BEGIN
358      0889      multiple_output = TRUE;      ! Set the flag indicating multiple output
359      0890      quiet_slip = TRUE;      ! and slip versions without reporting them.
360      0891      END;      ! files are to be created.
361      0892
362      0893
363      0894      [multiple_slip]:      ! Multiple output files, slip version numbers and re
364      0895      BEGIN

```

```

: 365      0896      multiple_output = TRUE;      ! Set the flag indicating multiple output.
: 366      0897      END;
: 367      0898
: 368      0899
: 369      0900      [multiple_max]:      ! Multiple output files, maximize version numbers.
: 370      0901      BEGIN
: 371      0902      multiple_output = TRUE;      ! Set the flag indicating that multiple output
: 372      0903      ! files are to be created.
: 373      0904      output_fab [fab$v_mxv] = TRUE;      ! Set a bit in the output FAB that causes version
: 374      0905      ! numbers to be maximized.
: 375      0906      END;
: 376      0907
: 377      0908
: 378      0909      [max_or_new]:      ! Concatenate and maximize
: 379      0910      BEGIN
: 380      0911      concat_follows = TRUE;      ! Set flag saying that concatenating files.
: 381      0912      output_fab [fab$v_mxv] = TRUE;      ! Set a bit in the output FAB that causes version
: 382      0913      ! numbers to be maximized.
: 383      0914      END;
: 384      0915
: 385      0916
: 386      0917      [concat_slip]:      ! Concatenate and slip version numbers.
: 387      0918      BEGIN
: 388      0919      concat_follows = TRUE;      ! Set flag saying that concatenating files.
: 389      0920      END;
: 390      0921
: 391      0922
: 392      0923      [error_spec]:      ! Error in spec combination.
: 393      0924      BEGIN
: 394      0925      put message (msg$_wildconcat, 99);      ! Report fatal error.
: 395      0926      RETURN error;      ! Return an error code.
: 396      0927      END;
: 397      0928
: 398      0929
: 399      0930      [no_action]:      ! No action to take.
: 400      0931      BEGIN
: 401      0932      concat_follows = TRUE;      ! Set flag saying that concatenating files.
: 402      0933      END;
: 403      0934
: 404      0935      TES;
: 405      0936
: 406      0937      !
: 407      0938      ! Some special case to handle the /NOCONCATENATE qualifer when it is explicitly specified.
: 408      0939
: 409      0940
: 410      0941      IF .negated_concat_qual      ! If the /NOCONCATENATE qualifier was explicitly
: 411      0942      THEN      ! specified,
: 412      0943      BEGIN
: 413      0944      multiple_output := TRUE;      ! then note that multiple files are to be produce
: 414      0945      concat_follows = FALSE;      ! and turn off concatenation.
: 415      0946      END;
: 416      0947
: 417      0948
: 418      0949      !
: 419      0950      ! Test for more error conditions. Files with relative, indexed, and hashed organizations
: 420      0951      ! cannot be concatenated with other files.
: 421      0952

```

```

422 0953 2 IF NOT .multiple_output ! If multiple output files are not being created,
423 0954 2 THEN ! then concatenation or single file COPY is occur
424 0955 2 BEGIN
425 0956 2 IF .outfile_open OR ! If an output file is open, we must be appending
426 0957 2 .multiple_input OR ! if more than one input file is specified, COPY
427 0958 2 ! tries to append.
428 0959 2 .wild_input OR ! again, multiple input files;
429 0960 2 .append_command ! the APPEND command always concatenates.
430 0961 2 THEN ! This sort of COPY is only allowed if file
431 0962 2 BEGIN ! organization is a simple one.
432 0963 2 IF .input_fab [fab$b_org] EQL fab$c_rel ! If the file has relative organization,
433 0964 2 THEN
434 0965 2 BEGIN
435 0966 2 input_fab [fab$l_sts] = ! Report an error by loading a special
436 0967 2 copy$msg_number ( ! error code into the RMS status field of
437 0968 2 ! msg$_relconcat); ! the input FAB.
438 0969 2 input_fab [fab$l_stv] = 0; ! Clear the secondard RMS status value.
439 0970 2 copy$inopn_err ( ! Call the open error action routine
440 0971 2 ! .input_fab); ! with the address of the input FAB block.
441 0972 2 wildcard_active = FALSE; ! Turn off the WILDCARD_ACTIVE flag.
442 0973 2 RETURN no_file; ! Return an error message to the caller.
443 0974 2 END;
444 0975 2
445 0976 2 IF .input_fab [fab$b_org] EQL fab$c_idx ! If this is an indexed file organization,
446 0977 2 THEN !
447 0978 2 BEGIN ! construct a similar error.
448 0979 2 input_fab [fab$l_sts] =
449 0980 2 copy$msg_number (
450 0981 2 ! msg$_idxconcat);
451 0982 2 input_fab [fab$l_stv] = 0;
452 0983 2 copy$inopn_err (
453 0984 2 ! .input_fab);
454 0985 2 wildcard_active = FALSE;
455 0986 2 RETURN no_file;
456 0987 2 END;
457 0988 2
458 0989 2 IF .input_fab [fab$b_org] EQL fab$c_hsh ! If this is a hashed file organization,
459 0990 2 THEN !
460 0991 2 BEGIN ! construct a similar error.
461 0992 2 input_fab [fab$l_sts] =
462 0993 2 copy$msg_number (
463 0994 2 ! msg$_hashconcat);
464 0995 2 input_fab [fab$l_stv] = 0;
465 0996 2 copy$inopn_err (
466 0997 2 ! .input_fab);
467 0998 2 wildcard_active = FALSE;
468 0999 2 RETURN no_file;
469 1000 2 END;
470 1001 2
471 1002 2 END;
472 1003 2
473 1004 2 END;
474 1005 2
475 1006 2
476 1007 2 ! Return with success status code.
477 1008 2
478 1009 2

```

: 479 1010 2 RETURN ok;
: 480 1011 2
: 481 1012 1 END;

! Return successfully.

```

                                .TITLE COPYSEMAN
                                .IDENT \V04-000\
                                .PSECT $SPLITS,NOWRT,NOEXE,2
0001 0080 0000 00A0 0000 0028 0005 0009 0005 0051 00000 P.AAA: .WORD 81, 5, 9, 5, 40, 0, 160, 0, 128, 1, 96, -
0002 0040 0001 0042 0001 000A 0001 0044 0001 0060 00014 1, 68, 1, 10, 1, 66, 1, 64, 2, 8, 2, 4, -
0006 0000 0003 0010 0003 0202 0004 0004 0002 0008 00028 4, 514, 3, 16, 3, 0, 6
                                DECISION_TABLE= P.AAA
                                .EXTRN COPY$MSG_NUMBER
                                .EXTRN COPY$CLI_STATUS
                                .EXTRN COPY$SEM_STATUS
                                .EXTRN CLIS_PRESENT, CLIS_NEGATED
                                .EXTRN CLIS_LOCPRES, CLIS_LOCNEG
                                .EXTRN COPY$INOPN_ERR
                                .PSECT $CODE$,NOWRT,2
                                .ENTRY COPY$SEMANTICS, Save R2,R3,R4 : 0782
                                MOVAB COPY$SEM_STATUS, R4
                                MOVAB DECISION_TABLE, DECIS_TBL_PTR : 0849
                                MOVL OUTPUT_FAB, R0 : 0855
                                B:CB2 #2, 4(R0)
                                MOVZWL @STATUS, R1 : 0867
                                MOVZWL (DECIS_TBL_PTR), R3
                                MCOML R3, R3
                                BICL2 R3, R1
                                CMPZV #0, #16, (DECIS_TBL_PTR), R1 : 0868
                                BEQL 2$
                                ADDL2 #4, DECIS_TBL_PTR : 0872
                                TSTW (DECIS_TBL_PTR) : 0873
                                BNEQ 1$
                                CASEW 2(DECIS_TBL_PTR), #0, #6 : 0882
                                .WORD 4$-3$,-
                                5$-3$,-
                                6$-3$,-
                                7$-3$,-
                                10$-3$,-
                                9$-3$,-
                                10$-3$
                                01 A4 0101 8F A8 00042 4$: B:SW2 #257, COPY$SEM_STATUS+1 : 0890
                                35 11 00048 BRB 11$ : 0882
                                01 A4 01 88 0004A 5$: B:SB2 #1, COPY$SEM_STATUS+1 : 0896
                                2F 11 0004E RRB 11$ : 0882
                                01 A4 01 88 00050 6$: B:SB2 #1, COPY$SEM_STATUS+1 : 0902
                                04 11 00054 BRB 8$ : 0904
                                02 A4 08 88 00056 7$: B:SB2 #8, COPY$SEM_STATUS+2 : 0911
                                04 A0 02 88 0005A 8$: B:SB2 #2, 4(R0) : 0912
                                1F 11 0005E BRB 11$ : 0882
                                7E 63 8F 9A 00060 9$: MOVZBL #99, -(SP) : 0925
                                7E 10C2 8F 3C 00064 MOVZWL #4290, -(SP)

```


	0000G	CF		01	FB	00069		CALLS	#1, COPY\$MSG_NUMBER	
				50	DD	0006E		PUSHL	R0	0926
	00000000G	00		02	FB	00070		CALLS	#2, LIB\$SIGNAL	
		50		02	DO	00077		MOVL	#2, R0	
						04		RET		
	02	A4		08	88	0007B	10\$:	BISB2	#8, COPY\$SEM_STATUS+2	0932
08	0000G	CF		03	E1	0007F	11\$:	BBC	#3, COPY\$CLI_STATUS, 12\$	0941
	01	A4		01	88	00085		BISB2	#1, COPY\$SEM_STATUS+1	0944
	02	A4		08	8A	00089		BICB2	#8, COPY\$SEM_STATUS+2	0945
		55	01	A4	E8	0008D	12\$:	BLBS	COPY\$SEM_STATUS+1, 17\$	0953
0E	02	A4		01	E0	00091		BBS	#1, COPY\$SEM_STATUS+2, 13\$	0956
09	01	A4		01	E0	00096		BBS	#1, COPY\$SEM_STATUS+1, 13\$	0957
05		64		04	E0	0009B		BBS	#4, COPY\$SEM_STATUS, 13\$	0959
		42	0000G	CF	E9	0009F		BLBC	COPY\$CLI_STATUS, 17\$	0960
		52	08	AC	DO	000A4	13\$:	MOVL	INPUT FAB, R2	0963
		10	1D	A2	91	000A8		CMPB	29(R2), #16	
				07	12	000AC		BNEQ	14\$	
		7E	1142	8F	3C	000AE		MOVZWL	#4418, -(SP)	0967
				1B	11	000B3		BRB	16\$	
		20	1D	A2	91	000B5	14\$:	CMPB	29(R2), #32	0976
				07	12	000B9		BNEQ	15\$	
		7E	113A	8F	3C	000BB		MOVZWL	#4410, -(SP)	0980
				0B	11	000C0		BRB	16\$	
		30	1D	A2	91	000C2	15\$:	CMPB	29(R2), #48	0989
				1E	12	000C6		BNEQ	17\$	
		7E	11DA	8F	3C	000C8		MOVZWL	#4570, -(SP)	0993
	0000G	CF		01	FB	000CD	16\$:	CALLS	#1, COPY\$MSG_NUMBER	
	08	A2		50	DO	000D2		MOVL	R0, 8(R2)	
			0C	A2	D4	000D6		CLRL	12(R2)	0995
				52	DD	000D9		PUSHL	R2	0997
	0000G	CF		01	FB	000DB		CALLS	#1, COPY\$INOPN_ERR	
	02	A4		20	8A	000E0		BICB2	#32, COPY\$SEM_STATUS+2	0998
				04	11	000E4		BRB	18\$	0999
		50		01	DO	000E6	17\$:	MOVL	#1, R0	1010
						04		RET		
				50	D4	000EA	18\$:	CLRL	R0	1012
				04	000EC			RET		

: Routine Size: 237 bytes, Routine Base: \$CODE\$ + 0000

: 483 1013 1 END
: 484 1014 0 ELUDOM

.EXTRN LIB\$SIGNAL

PSECT SUMMARY

Name	Bytes	Attributes
\$SPLITS	60	NOVEC,NOWRT, RD ,NOEXE,NOSHR, LCL, REL, CON,NOPIC,ALIGN(2)
\$CODES	237	NOVEC,NOWRT, RD , EXE,NOSHR, LCL, REL, CON,NOPIC,ALIGN(2)

Library Statistics

File	Total	Symbols Loaded	Percent	Pages Mapped	Processing Time
_\$255\$DUA28:[SYSLIB]STARLET.L32;1	9776	52	0	581	00:01.0

COMMAND QUALIFIERS

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

: Size: 237 code + 60 data bytes
: Run Time: 00:13.7
: Elapsed Time: 00:38.1
: Lines/CPU Min: 4453
: Lexemes/CPU-Min: 33456
: Memory Used: 135 pages
: Compilation Complete

This image displays a grid of 100 terminal window screenshots, arranged in 10 rows and 10 columns. Each window shows a different system utility or report. The windows are densely packed and contain various types of data, including text-based tables, lists, and status information. Some of the visible window titles and content include:

- COPYMSG REQ**: A window showing a list of copy messages with columns for message ID, date, and status.
- COPY**: A window showing a list of copy jobs.
- COPYMAP**: A window showing a map or layout of copy jobs.
- VMSMAC REQ**: A window showing a list of VMSMAC requests.
- COPYREQ**: A window showing a list of copy requests.
- COPYCLI LIS**: A window showing a list of copy client information.
- COPYMAIN LIS**: A window showing a list of copy main information.
- COPYSEMAN LIS**: A window showing a list of copy semaphore information.

The screenshots are small and difficult to read in detail, but they clearly show the output of various system utilities. The overall appearance is that of a multi-terminal session on a VAX/VMS system.