

CCCCCCCC	AAAAAA	PPPPPPP	TTTTTTTT	IIIIII	000000	NN	NN	
CCCCCCCC	AAAAAA	PPPPPPP	TTTTTTTT	IIIIII	000000	NN	NN	
CC	AA	AA	PP	PP	TT	II	00	00
CC	AA	AA	PP	PP	TT	II	00	00
CC	AA	AA	PP	PP	TT	II	00	00
CC	AA	AA	PP	PP	TT	II	00	00
CC	AA	AA	PP	PP	TT	II	00	00
CC	AA	AA	PP	PP	TT	II	00	00
CC	AA	AA	PP	PP	TT	II	00	00
CC	AA	AA	PP	PP	TT	II	00	00
CC	AA	AA	PP	PP	TT	II	00	00
CC	AA	AA	PP	PP	TT	II	00	00
CC	AA	AA	PP	PP	TT	II	00	00
CC	AA	AA	PP	PP	TT	II	00	00
CCCCCCCC	AAAAAA	PPPPPPP	TTTTTTTT	IIIIII	000000	NN	NN
CCCCCCCC	AAAAAA	PPPPPPP	TTTTTTTT	IIIIII	000000	NN	NN

LL	IIIIII	SSSSSSSS
LL	IIIIII	SSSSSSSS
LL	II	SS
LL	II	SS
LL	II	SS
LL	II	SS
LL	II	SSSSSS
LL	II	SSSSSS
LL	II	SS
LL	II	SS
LL	II	SS
LL	II	SS
LLLLLLLLLL	IIIIII	SSSSSSSS
LLLLLLLLLL	IIIIII	SSSSSSSS

.....

:

```
1 0001 0 %TITLE 'CAPTION - Generate figure, table and example captions'  
2 0002 0 !<BLF/NOFORMAT>  
3 0003 0  
4 0004 0 MODULE caption (IDENT = 'V04-000'  
5 0005 0 %BLISS32C, ADDRESSING_MODE (EXTERNAL = LONG_RELATIVE, NONEXTERNAL = LONG_RELATIVE)]  
6 0006 0 ) =  
7 0007 1 BEGIN  
8 0008 1  
9 0009 1 !<BLF/FORMAT>  
10 0010 1 !<BLF/LOWERCASE_USER>  
11 0011 1 !<BLF/UPPERCASE_KEY>  
12 0012 1 !<BLF/MACRO>  
13 0013 1  
14 0014 1  
15 0015 1  
16 0016 1 *****  
17 0017 1 *  
18 0018 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY  
19 0019 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.  
20 0020 1 * ALL RIGHTS RESERVED.  
21 0021 1 *  
22 0022 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED  
23 0023 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE  
24 0024 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER  
25 0025 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY  
26 0026 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY  
27 0027 1 * TRANSFERRED.  
28 0028 1 *  
29 0029 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE  
30 0030 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT  
31 0031 1 * CORPORATION.  
32 0032 1 *  
33 0033 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS  
34 0034 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.  
35 0035 1 *  
36 0036 1 *****  
37 0037 1  
38 0038 1  
39 0039 1 ++  
40 0040 1 FACILITY:  
41 0041 1 DSR (Digital Standard RUNOFF) /DSRPLUS DSRTOC/CONTENTS Utility  
42 0042 1  
43 0043 1 ABSTRACT:  
44 0044 1 Format figure, table and example captions  
45 0045 1  
46 0046 1 ENVIRONMENT: Transportable  
47 0047 1  
48 0048 1 AUTHOR: JPK  
49 0049 1  
50 0050 1 CREATION DATE: March 1982  
51 0051 1  
52 0052 1 MODIFIED BY:  
53 0053 1  
54 0054 1 007 JPK00010 05-May-1983  
55 0055 1 Modified CONTENTS to change text for index record group to all  
56 0056 1 caps. Modified CAPTION to change the number of blank lines  
57 0057 1 after the entity title line.
```

```

58 0058 1
59 0059 1
60 0060 1
61 0061 1
62 0062 1
63 0063 1
64 0064 1
65 0065 1
66 0066 1
67 0067 1
68 0068 1
69 0069 1
70 0070 1
71 0071 1
72 0072 1
73 0073 1
74 0074 1
75 0075 1
76 0076 1
77 0077 1
78 0078 1
79 0079 1
80 0080 1
81 0081 1
82 0082 1
83 0083 1
84 0084 1
85 0085 1
86 0086 1
87 0087 1
88 0088 1
89 0089 1
90 0090 1
91 0091 1
92 0092 1
93 0093 1
94 0094 1
95 0095 1
96 0096 1
97 0097 1
98 0098 1
99 0099 1
100 0100 1
101 0101 1
102 0102 1
103 0103 1
104 0104 1
105 0105 1
106 0106 1
107 0107 1
108 0108 1
109 0109 1
110 0110 1
111 0111 1
112 0112 1
113 0193 1
114 0194 1

```

006 JPK00008 09-Mar-1983
Modified CONTENTS and CAPTION to support new BRN formats,
support SEND CONTENTS, /DOUBLE_SPACE, page numbered chapters,
guarantee space after section number and to write new
prologue and epilog for RUNOFF output.
Modified FORMAT to quote only the RUNOFF flags used by CONTENTS.
Modified CNTVMS to fix default for /DOUBLE_SPACE and do more
value checking.

005 JPK00007 14-Feb-1983
Global edit of all sources for CONTENTS/DSRTOC:
- module names are now consistent with file names
- copyright dates have been updated
- facility names have been updated
- revision history was updated to be consistent with DSR/DSRPLUS

004 JPK00004 11-Feb-1983
Changed the global variable name INDENT to LINE_INDENT in
modules CONTENTS, CAPTION, FORMAT and GBLDCL.
Removed declarations of PDENTS in modules CNTVMS, CONTENTS,
and CAPTION and replaced with a module wide BIND using the
new name INDENTS.
Changed handling of INDENTS [1]. It no longer represents the
sum of the chapter and title indents.

003 KFA00002 14-Oct-1982
Modified CNTVMS, CONTENTS, CAPTION and CNTCLI.REQ to handle
new syntax for /INDENT qualifier. All indents are now
stored in the vector CMDBLK [CONTENTSSAG_HL_INDENT []],
where 0 = chapter indent, 1 = header level one indent, etc.

002 KFA00001 13-Oct-1982
Modified CONTENTS and CAPTION to allow null captions for
headers, examples, figures, & tables.

--

TABLE OF CONTENTS:

FORWARD ROUTINE
hdr_x : NOVALUE ; Process Figure, Table and Example captions
stcHdr : NOVALUE ; Process SEND TOC temporary file
strip : NOVALUE ; Remove leading and trailing whitespace
insrnt : NOVALUE ; Insert text directly into output file

INCLUDE FILES:

LIBRARY 'NXPORT:XPORT';
REQUIRE 'REQ:PAGEN'; ! Page reference format
REQUIRE 'REQ:TOCRTY'; ! Table of Contents file formats

```

115 0304 1
116 0305 1 REQUIRE 'REQ:CNTCLI';           ! Command line information block formats
117 0433 1
118 0434 1
119 0435 1   ! MACROS:
120 0436 1   !
121 0437 1   !
122 0438 1 MACRO
123 0439 1
124 0440 1   ! Write a character to output line
125 0441 1
126 M 0442 1   write_char (ch) [] =
127 M 0443 1   BEGIN
128 M 0444 1   CH$WCHAR_A (ch, lp);
129 M 0445 1   intlin = .intlin + 1;
130 M 0446 1
131 M 0447 1   %IF NOT %NULL (%REMAINING)
132 M 0448 1   %THEN
133 M 0449 1     extlin = .extlin + 1;
134 M 0450 1   %FI
135 M 0451 1
136 M 0452 1   END
137 M 0453 1   !
138 M 0454 1   !
139 M 0455 1   ! Write a text literal to the output line
140 M 0456 1
141 M 0457 1   ! literal text (str) =
142 M 0458 1   BEGIN
143 M 0459 1   CH$MOVE (%CHARCOUNT (str), CH$PTR (UPLIT (str)), .lp);
144 M 0460 1   lp = CH$PLUS (.lp, %CHARCOUNT (str));
145 M 0461 1   intlin = .intlin + %CHARCOUNT (str);
146 M 0462 1   END
147 M 0463 1   !
148 M 0464 1   !
149 M 0465 1   ! Pad the output line with blanks
150 M 0466 1
151 M 0467 1   pad (n_blanks) =
152 M 0468 1   BEGIN
153 M 0469 1
154 M 0470 1   IF n_blanks GTR 0
155 M 0471 1   THEN
156 M 0472 1     BEGIN
157 M 0473 1     CH$FILL (%C' ', n_blanks, .lp);
158 M 0474 1     lp = CH$PLUS (.lp, n_blanks);
159 M 0475 1     intlin = .intlin + n_blanks;
160 M 0476 1     extlin = .extlin + n_blanks;
161 M 0477 1     END;
162 M 0478 1
163 M 0479 1   END
164 M 0480 1   !
165 M 0481 1   !
166 M 0482 1   ! Clear the text lines being built up.
167 M 0483 1
168 M 0484 1   ! clr_line (_) =
169 M 0485 1   BEGIN
170 M 0486 1   lp = CH$PTR (line);
171 M 0487 1   intlin = 0;

```

```

172 M 0488 1      extlin = 0;
173 M 0489 1      END
174      0490 1      |
175      0491 1      |
176      0492 1      | Insert specified character sequence into file, as is.
177      0493 1      |
178 M 0494 1      put (str) =
179 M 0495 1      BEGIN
180 M 0496 1      $str_copy (string = str, target = tmpstr);
181 M 0497 1      chROUT = .chROUT + .tmpstr [str$h_length];
182 M 0498 1      $xpo_put (iob = tocoob, string = tmpstr);
183 M 0499 1      |
184 M 0500 1      | For TMS output, split the output file if it gets too large
185 M 0501 1      |
186 M 0502 1      |
187 M 0503 1      IF .cmdblk [contents$v_tms11] THEN split ();
188 M 0504 1      |
189 M 0505 1      END
190      0506 1      |
191      0507 1      |
192      0508 1      |
193      0509 1      | EQUATED SYMBOLS:
194      0510 1      |
195      0511 1      |
196      0512 1      LITERAL
197      0513 1      true = 1
198      0514 1      false = 0
199      0515 1      tab = %0'11' : UNSIGNED (8),
200      0516 1      rintex = %0'34' : UNSIGNED (8),
201      0517 1      infinity = 1^(%BPVAL - 1) - 1;
202      0518 1      |
203      0519 1      |
204      0520 1      | OWN STORAGE:
205      0521 1      |
206      0522 1      |
207      0523 1      | EXTERNAL REFERENCES:
208      0524 1      |
209      0525 1      |
210      0526 1      EXTERNAL
211      0527 1      cmdblk : $contents_cmd,
212      0528 1      tocoob : $xpo_iob (?),
213      0529 1      stciob : $xpo_iob (?),
214      0530 1      chROUT,
215      0531 1      tmpstr : $str_descriptor (),
216      0532 1      txtlen,
217      0533 1      hl_n,
218      0534 1      hlList : VECTOR [100],
219      0535 1      hldsp : VECTOR [100],
220      0536 1      major,
221      0537 1      minor,
222      0538 1      pagen : page_definition,
223      0539 1      nxtstc,
224      0540 1      rno_ex : $str_descriptor (),
225      0541 1      rno_fg : $str_descriptor (),
226      0542 1      rno_tb : $str_descriptor (),
227      0543 1      tms_ex : $str_descriptor (),
228      0544 1      tms_fg : $str_descriptor (),

```

```

| Command line information block
| IOB for the resulting .RNT file
| IOB for send toc temp file
| Number of characters written to output file
| Temporary strings
| Length of input text.
| "n" from latest .HL n command
| Header level counters
| Header level display characteristics
| Major record type code
| Minor record type code
| Current page number.
| Next SEND IOC "n" value to process
| String constant 'EXAMPLES'
| String constant 'FIGURES'
| String constant 'TABLES'
| String constant 'Examples'
| String constant 'Figures'

```

```

: 229      0545 1      tms_tb : $str_descriptor (),      | String constant 'Tables'
: 230      0546 1      prestr : $str_descriptor (),      | Caption prefix
: 231      0547 1      sfxstr : $str_descriptor (),      | Caption suffix
: 232      0548 1      lp,      | CH$PTR along line being built up
: 233      0549 1      intlin,      | Number of characters needed to represent text
: 234      0550 1      extlin,      | Number of resulting print positions
: 235      0551 1      line : VECTOR [CH$ALLOCATION (10000)],      | Buffer in which line is being built up.
: 236      0552 1      lenpag,      | Number of characters in the converted page number.
: 237      0553 1      txtpag : VECTOR [CH$ALLOCATION (50)],      | The text (lots of room)
: 238      0554 1      ptrpag,      | A CH$PTR (working pointer)
: 239      0555 1      lensct,      | Number of characters in the converted section number
: 240      0556 1      txtsct : VECTOR [CH$ALLOCATION (150)],      | The text (lots of room)
: 241      0557 1      ptrsct,      | A CH$PTR (working pointer)
: 242      0558 1      rmargin,      | Used by ENDWRD for controlling filling lines.
: 243      0559 1      wrap,      | Wrap long lines around to here.
: 244      0560 1      line_indent;      | Assume this standard indentation before the text.
: 245      0561 1
: 246      0562 1 BIND
: 247      0563 1      indents = cmdblk[contents$ag_hl_indent] : VECTOR;
: 248      0564 1
: 249      0565 1 EXTERNAL ROUTINE
: 250      0566 1      split : NOVALUE,      | Start new output file for TMS
: 251      0567 1      fmttxt : NOVALUE,      | Generate formatted text
: 252      0568 1      insref : NOVALUE,      | Insert a page reference
: 253      0569 1      pacpag,      | Format a page number
: 254      0570 1      pacbas,      | Convert a number to ASCII
: 255      0571 1      pacsec,      | Format a section number
: 256      0572 1      pacxxx;      | Format a number using display code
: 257      0573 1
: 258      L 0574 1 %IF %BLISS (BLISS32)
: 259      0575 1 %THEN
: 260      0576 1
: 261      0577 1 EXTERNAL ROUTINE
: 262      0578 1      open_error;      | File open error handling routine
: 263      0579 1
: 264      0580 1 %FI

```

```

266 0581 1 %SBTTL 'HDR X - Process .HEADER x captions'
267 0582 1 GLOBAL ROUTINE hdr_x (
268 0583 1     iob : REF $xpo_iob (),
269 0584 1     title : REF $str_descriptor (),
270 0585 1     info,
271 0586 1     txt,
272 0587 1     txt_u
273 0588 1 ) : NOVALUE =
274 0589 1 ++
275 0590 1
276 0591 1 FUNCTIONAL DESCRIPTION:
277 0592 1
278 0593 1     This routine processes the .HEADER x Captions
279 0594 1
280 0595 1 FORMAL PARAMETERS:
281 0596 1
282 0597 1     iob     - address of temp file iob
283 0598 1     title  - address of string descriptor of title
284 0599 1     info   - information record type
285 0600 1     txt    - text record type
286 0601 1     txt_u  - unnumbered text record type
287 0602 1
288 0603 1 IMPLICIT INPUTS:
289 0604 1
290 0605 1     cmdblk - command line information block
291 0606 1
292 0607 1 IMPLICIT OUTPUTS:
293 0608 1
294 0609 1     None
295 0610 1
296 0611 1 ROUTINE VALUE:
297 0612 1 COMPLETION CODES:
298 0613 1
299 0614 1     None
300 0615 1
301 0616 1 SIDE EFFECTS:
302 0617 1
303 0618 1     None
304 0619 1
305 0620 1 --
306 0621 1
307 0622 2 BEGIN
308 0623 2
309 0624 2 LOCAL
310 0625 2     first_record;
311 0626 2
312 0627 2 IF .cmdblk [contents$g_headers] EQL 0 THEN RETURN;
313 0628 2
314 0629 2 first_record = true;
315 P 0630 2 $xpo_open (iob = .iob, options = input, attributes = BINARY
316 P 0631 2     %IF %BLISS (BLISS32) %THEN , failure = open_error %FI
317 0632 2 );
318 0633 2
319 0634 2 WHILE $xpo_get (iob = .iob, fullwords = 2) EQL xpo$_normal DO
320 0635 2     BEGIN
321 0636 2         !
322 0637 3         ! Process a record group

```



```

323 0638 3 !
324 0639 3 BIND
325 0640 3 record_header = .iob [iob$a_data] : VECTOR;
326 0641 3
327 0642 3 major = .record_header [0];
328 0643 3 minor = .record_header [1];
329 0644 3
330 0645 3 SELECTONE true OF
331 0646 3 SET
332 0647 3
333 0648 3 [(.major EQL maj_runoff) AND (.minor EQL min_page)] :
334 0649 4 BEGIN
335 0650 4 |
336 0651 4 | Process a page reference
337 0652 4 |
338 0653 4 | $xpo_get (iob = .iob, fullwords = page_sct_size);
339 0654 5 BEGIN
340 0655 5 BIND
341 0656 5 | page_ref = .iob [iob$a_data] : VECTOR;
342 0657 5 |
343 0658 5 MAP
344 0659 5 | pagen : VECTOR;
345 0660 5 |
346 0661 5 | INCR i FROM 0 TO page_sct_size - 1 DO
347 0662 5 | | pagen [.i] = .page_ref [.i];
348 0663 5 |
349 0664 4 END;
350 0665 3 END;
351 0666 3
352 0667 3 [.major EQL maj_send_contents] :
353 0668 4 BEGIN
354 0669 4 |
355 0670 4 | Process a SEND CONTENTS record
356 0671 4 |
357 0672 4 | LOCAL
358 0673 4 | type;
359 0674 4 |
360 0675 4 | type = .minor<0, %BPVAL/2>;
361 0676 4 | $xpo_get (iob = .iob, fullwords = 1); ! Get length
362 0677 4 | txtlen = ..iob [iob$a_data];
363 0678 4 |
364 0679 4 | IF .txtlen GTR 0
365 0680 4 | THEN
366 0681 4 | | $xpo_get (iob = .iob, fullwords = CH$ALLOCATION (.txtlen));
367 0682 4 | |
368 0683 4 | IF .type EQL scon_comm
369 0684 4 | THEN
370 0685 4 | |
371 0686 4 | | A command. Insert directly into table of contents
372 0687 4 | |
373 0688 5 | | put ((.txtlen, CH$PTR (.iob [iob$a_data])))
374 0689 4 | ELSE
375 0690 5 | BEGIN
376 0691 5 | |
377 0692 5 | | Text. Format it.
378 0693 5 | |
379 0694 5 | IF .first_record

```

```

: 380 0695 5 THEN
: 381 0696 6 BEGIN
: 382 0697 6
: 383 0698 6 | First one seen - generate title
: 384 0699 6
: 385 0700 6 IF .cmdblk [contents$v_tms11]
: 386 0701 6 THEN
: 387 0702 7 BEGIN
: 388 0703 7 put ('');
: 389 0704 7
: 390 0705 7 IF .hl_n NEQ 0 THEN put ('[va15]');
: 391 0706 7
: 392 0707 7 put ($str_concat ('*c0n*', .title, '@@'));
: 393 0708 7 END
: 394 0709 6 ELSE
: 395 0710 7 BEGIN
: 396 0711 7 put ('.TEST PAGE 7');
: 397 0712 7 IF .hl_n NEQ 0 THEN put ('.SK2') ELSE put ('.SK1');
: 398 0713 7
: 399 0714 7 IF .indents [0] NEQ 0
: 400 0715 7 THEN
: 401 0716 7 put ($str_concat ('.I-', $str_ascii (.indents [0])));
: 402 0717 7
: 403 0718 7 put (,title);
: 404 0719 7 put ('.BLANK');
: 405 0720 6 END;
: 406 0721 6
: 407 0722 6 hl_n = 1;
: 408 0723 6 first_record = false;
: 409 0724 5 END;
: 410 0725 5
: 411 0726 5 |
: 412 0727 5 | Initialize formatting parameters
: 413 0728 5 |
: 414 0729 5 | clr_line ();
: 415 0730 5 | line_indent = .indents [0];
: 416 0731 5 | extlin = .line_indent;
: 417 0732 5 | wrap = .indents [0] + .indents [1];
: 418 0733 5 | rmargin = .cmdblk [contents$g_page_width];
: 419 0734 5
: 420 0735 5 IF .cmdblk [contents$v_tms11]
: 421 0736 5 THEN
: 422 0737 6 literal_text ('*c1n*@')
: 423 0738 5 ELSE
: 424 0739 5 pad ((.indents [0] + .indents [1] - .extlin));
: 425 0740 5
: 426 0741 5 IF .type EQL scon_txt
: 427 0742 5 THEN
: 428 0743 6 BEGIN
: 429 0744 6 |
: 430 0745 6 | Generate the page number so as to get its length.
: 431 0746 6 | This is needed so that the proper number of dots can be supplied.
: 432 0747 6 |
: 433 0748 6 ptrpag = CH$PTR (txtpag);
: 434 0749 6
: 435 0750 6 IF NOT .cmdblk [contents$v_standard_page]
: 436 0751 6 THEN

```

```

: 437 0752 6 lenpag = pacbas (.pagen [sct_run_page], ptrpag, 10)
: 438 0753 6 ELSE
: 439 0754 6 lenpag = pacpag (pagen, ptrpag);
: 440 0755 6
: 441 0756 6
: 442 0757 5 END
: 443 0758 5 ELSE
: 444 0759 5 lenpag = 0;
: 445 0760 5
: 446 0761 5 | Set limit beyond which words must be wrapped
: 447 0762 5 |
: 448 0763 5 rmargin = .cmdblk [contents$g_page_width] - .lenpag - 1;
: 449 0764 5 wrap = .extlin;
: 450 0765 5 |
: 451 0766 5 | Scan text(if any) and put it(may be null caption) onto the line.
: 452 0767 5 |
: 453 0768 5 fmttxt (.txtlen, CH$PTR (.iob [iob$a_data]));
: 454 0769 5 |
: 455 0770 5 IF .type EQL scon_txt
: 456 0771 5 THEN
: 457 0772 5 insref () ! Insert page reference
: 458 0773 5 ELSE
: 459 0774 6 BEGIN
: 460 0775 6
: 461 0776 6 IF .cmdblk [contents$v_tms11] THEN write_char (%'a');
: 462 0777 5 END;
: 463 0778 5 |
: 464 0779 5 |
: 465 0780 5 | Now actually put this line of text into the file
: 466 0781 5 |
: 467 0782 5 put ((.intlin, CH$PTR (line)));
: 468 0783 5 clr_line ();
: 469 0784 4 END;
: 470 0785 4
: 471 0786 3 END;
: 472 0787 3
: 473 0788 3 [(.major EQL maj_runoff) AND (.minor EQL .info)] :
: 474 0789 4 BEGIN
: 475 0790 4 |
: 476 0791 4 | A caption information record
: 477 0792 4 |
: 478 0793 4 LOCAL
: 479 0794 4 prefix_len,
: 480 0795 4 suffix_len;
: 481 0796 4 |
: 482 0797 4 |
: 483 0798 4 | Get caption counter, display descriptor and prefix text length
: 484 0799 4 |
: 485 0800 4 $xpo get (iob = .iob, fullwords = 3);
: 486 0801 5 BEGIN
: 487 0802 5 |
: 488 0803 5 |
: 489 0804 5 BIND
: 490 0805 5 info_vector = .iob [iob$a_data] : VECTOR [3];
: 491 0806 5 hllist [0] = .info_vector [0]; ! Save the counter
: 492 0807 5 hldsp [0] = .info_vector [1]; ! the display descriptor
: 493 0808 5 prefix_len = .info_vector [2]; ! and prefix string length

```

```

494 0809 4      END;
495 0810 4
496 0811 4      | Get prefix text if any
497 0812 4
498 0813 4      | IF .prefix_len NEQ 0 THEN $xpo_get (iob = .iob, fullwords = CH$ALLOCATION (.prefix_len));
499 0814 4
500 0815 4      | $str_copy (string = (.prefix_len, CH$PTR (.iob [iob$a_data])), target = prestr);
501 0816 4
502 0817 4      | Get length of suffix text
503 0818 4
504 0819 4      | $xpo_get (iob = .iob, fullwords = 1);
505 0820 4      | suffix_len = (.iob [iob$a_data]);
506 0821 4
507 0822 4      | Get suffix text if any
508 0823 4
509 0824 4      | IF .suffix_len NEQ 0 THEN $xpo_get (iob = .iob, fullwords = CH$ALLOCATION (.suffix_len));
510 0825 4
511 0826 4      | $str_copy (string = (.suffix_len, CH$PTR (.iob [iob$a_data])), target = sfxstr);
512 0827 3      END;
513 0828 3
514 0829 3      [(.major EQL maj_runoff) AND (.minor EQL .txt OR .minor EQL .txt_u)] :
515 0830 4      BEGIN
516 0831 4
517 0832 4      | Process the text part of a caption.
518 0833 4
519 0834 4      LOCAL
520 0835 4      | needed_fill;
521 0836 4
522 0837 4
523 0838 4      | First read the number of characters needed to represent the caption
524 0839 4
525 0840 4      | $xpo_get (iob = .iob, fullwords = 1);
526 0841 4      | txtlen = (.iob [iob$a_data]); ! Save text length.
527 0842 4
528 0843 4      | Generate a table of contents entry even if the caption
529 0844 4      | has no text associated with it
530 0845 4
531 0846 4      | If .txtlen NEQ 0
532 0847 4      | THEN
533 0848 4      |   $xpo_get (iob = .iob, fullwords = CH$ALLOCATION (.txtlen));
534 0849 4
535 0850 4
536 0851 4      | Initialize formatting parameters
537 0852 4
538 0853 4      | clr_line ();
539 0854 4      | line_indent = .indents [0];
540 0855 4      | extlin = .line_indent;
541 0856 4      | wrap = .indents [0] + .indents [1];
542 0857 4      | rmargin = .cmdblk [contents$g_page_width];
543 0858 4
544 0859 4      | IF .first_record
545 0860 4      | THEN
546 0861 5      | BEGIN
547 0862 5      |   | First one seen - generate title
548 0863 5      |   |
549 0864 5      |   | IF .cmdblk [contents$v_tms11]
550 0865 5

```

```

551 0866 5      THEN
552 0867 6      BEGIN
553 0868 6      put ('');
554 0869 6      IF .hl_n NEQ 0 THEN put ('[va15]');
555 0870 6      IF .hl_n NEQ 0 THEN put ('[va15]');
556 0871 6      put ($str_concat ('*c0n*', .title, '@@'));
557 0872 6      END
558 0873 6      ELSE
559 0874 5      BEGIN
560 0875 6      put ('.TEST PAGE 7');
561 0876 6      IF .hl_n NEQ 0 THEN put ('.SK2') ELSE put ('.SK1');
562 0877 6      IF .indents [0] NEQ 0
563 0878 6      THEN
564 0879 6      put ($str_concat ('.I-', $str_ascii (.indents [0])));
565 0880 6      put (.title);
566 0881 6      put ('.BLANK');
567 0882 6      END;
568 0883 6      hl_n = 1;
569 0884 6      first_record = false;
570 0885 5      END;
571 0886 5      IF .cmdblk [contents$v_tms11] THEN literal_text ('*c1n*');
572 0887 5      IF .cmdblk [contents$v_include_sections]
573 0888 5      AND (
574 0889 4      (.cmdblk [contents$g_sections] GEQ .hl_n)
575 0890 4      OR (
576 0891 4      (.cmdblk [contents$g_sections] EQL -1)
577 0892 4      AND (.minor NEQ .txt_u)
578 0893 4      )
579 0894 5      )
580 0895 6      THEN
581 0896 6      BEGIN
582 0897 7      : Generate number if either the user said include
583 0898 7      : all header numbers up to some level GEQ 1 or
584 0899 6      : header numbers should be displayed as input and
585 0900 5      : this was a numbered head.
586 0901 4      ptrsct = CH$PTR (txtsct);
587 0902 5      lensct = 0;
588 0903 5      IF .pagen [sct_typ] NEQ 0
589 0904 5      THEN
590 0905 5      BEGIN
591 0906 5      : Document is divided into sections
592 0907 5      lensct = .lensct + pacsec (pagen, ptrsct) + 1;
593 0908 5      CH$WCHAR_A (XC'-', ptrsct);
594 0909 5      END;
595 0910 5      lensct = .lensct + pacxxx (.hllist [0], ptrsct, .hldsp [0]);
596 0911 5      IF .pagen [sct_typ] NEQ 0
597 0912 5      THEN
598 0913 5      BEGIN
599 0914 6      : Document is divided into sections
600 0915 6      lensct = .lensct + pacsec (pagen, ptrsct) + 1;
601 0916 6      CH$WCHAR_A (XC'-', ptrsct);
602 0917 6      END;
603 0918 6      lensct = .lensct + pacxxx (.hllist [0], ptrsct, .hldsp [0]);
604 0919 6      IF .pagen [sct_typ] NEQ 0
605 0920 5      THEN
606 0921 5      BEGIN
607 0922 5      : Document is divided into sections

```

```

: 608 0923 5
: 609 0924 5
: 610 0925 5
: 611 0926 5
: 612 0927 5
: 613 0928 5
: 614 0929 5
: 615 0930 5
: 616 0931 5
: 617 0932 5
: 618 0933 5
: 619 0934 5
: 620 0935 5
: 621 0936 5
: 622 0937 5
: 623 0938 5
: 624 0939 6
: 625 0940 6
: 626 0941 6
: 627 0942 6
: 628 0943 6
: 629 0944 6
: 630 0945 6
: 631 0946 5
: 632 0947 5
: 633 0948 5
: 634 0949 4
: 635 0950 4
: 636 0951 4
: 637 0952 4
: 638 0953 5
: 639 0954 4
: 640 0955 4
: 641 0956 4
: 642 0957 4
: 643 0958 4
: 644 0959 4
: 645 0960 4
: 646 0961 4
: 647 0962 4
: 648 0963 4
: 649 0964 4
: 650 0965 5
: 651 0966 5
: 652 0967 5
: 653 0968 5
: 654 0969 5
: 655 0970 5
: 656 0971 4
: 657 0972 5
: 658 0973 5
: 659 0974 5
: 660 0975 5
: 661 0976 5
: 662 0977 4
: 663 0978 4
: 664 0979 4

```

```

:
: Write the prefix string if any
: IF .prestr [str$h_length] NEQ 0
: THEN
:   fmttxt (.prestr [str$h_length], .prestr [str$a_pointer]);
:
: Write the entity number
: fmttxt (.lensct, CH$PTR (txtsct));
:
: Write the suffix string if any
: IF .sfxstr [str$h_length] NEQ 0
: THEN
:   BEGIN
:     IF CH$NEQ(1, CH$PTR (UPLIT(%':')),
:       .sfxstr [str$h_length], .sfxstr [str$a_pointer])
:     THEN
:       fmttxt (.sfxstr [str$h_length], .sfxstr [str$a_pointer]);
:
:     END;
:
:   IF NOT .cmdblk [contents$v_tms11] THEN write_char (%' ', counts_visually);
:   END;
:
: IF .cmdblk [contents$v_tms11]
: THEN
:   write_char (%'@')
: ELSE
:   pad ((.indents [0] + .indents [1] - .extlin));
:
: Generate the page number so as to get its length.
: This is needed so that the proper number of dots can be supplied.
: ptrpag = CH$PTR (txtpag);
:
: IF NOT .cmdblk [contents$v_standard_page]
: THEN
:   BEGIN
:     Use the running page counter instead of a real page number.
:     lenpag = pacbas (.pagen [sct_run_page], ptrpag, 10);
:   END
: ELSE
:   BEGIN
:     Supply the usual page number.
:     lenpag = pacpag (pagen, ptrpag);
:   END;
:
:
:

```

```

: 665 0980 4 : Set limit beyond which words must be wrapped
: 666 0981 4
: 667 0982 4 rmargin = .cndblk [contents$g_page_width] - .lenpag - 1;
: 668 0983 4 wrap = .extlin;
: 669 0984 4
: 670 0985 4 : Scan text(if any) and put it(may be null caption) onto the line.
: 671 0986 4
: 672 0987 4 fmttxt (.txtlen, CHSPTR (.iob [iob$a_data]));
: 673 0988 4
: 674 0989 4 : Insert page reference into line if desired.
: 675 0990 4
: 676 0991 4 insref ();
: 677 0992 4
: 678 0993 4 : Now actually put this line of text into the file
: 679 0994 4
: 680 0995 4 put ((.intlin, CHSPTR (line)));
: 681 0996 4 clr_line ();
: 682 0997 4 END;
: 683 0998 3
: 684 0999 3 TES;
: 685 1000 3 END;
: 686 1001 2 $xpo_close (iob = .iob, options = remember);
: 687 1002 2 END;
: 688 1003 2
: 689 1004 1

```

```

:
: .TITLE CAPTION CAPTION - Generate figure, table and ex
: .IDENT \V04-000\
: .PSECT $SPLITS,NOWRT,NOEXE,2
:
:          5D 35 31 61 76 5B 00000 P.AAA: .BLKB 0
:          2A 6E 30 63 2A 00000 P.AAB: .ASCII \[va15]\
: 37 20 45 47 41 50 20 54 53 45 54 2E 00006 P.AAE: .ASCII \*c0n*\
:          40 40 00008 P.AAF: .ASCII \@@\
:          32 48 53 2E 0000D P.AAI: .ASCII \.TEST PAGE 7\
:          31 48 53 2E 00019 P.AAJ: .ASCII \.SK2\
:          2D 49 2E 0001D P.AAK: .ASCII \.SK1\
:          4B 4E 41 4C 42 2E 00021 P.AAM: .ASCII \.I-\
:          00 00 40 2A 6E 31 63 2A 00024 P.AAO: .ASCII \.BLANK\
:          0002A .BLKB 2
:          00 00 40 2A 6E 31 63 2A 0002C P.AAP: .ASCII \*c1n*@\<0><0>
:          5D 35 31 61 76 5B 00034 P.AAQ: .BLKB 0
:          2A 6E 30 63 2A 00034 P.AAR: .ASCII \[va15]\
: 37 20 45 47 41 50 20 54 53 45 54 2E 0003A P.AAU: .ASCII \*c0n*\
:          40 40 0003F P.AAV: .ASCII \@@\
:          32 48 53 2E 00041 P.AAY: .ASCII \.TEST PAGE 7\
:          31 48 53 2E 0004D P.AAZ: .ASCII \.SK2\
:          2D 49 2E 00051 P.ABA: .ASCII \.SK1\
:          4B 4E 41 4C 42 2E 00055 P.ABC: .ASCII \.I-\
:          00 00 00 2A 6E 31 63 2A 00058 P.ABE: .ASCII \.BLANK\
:          0005E .BLKB 2
:          00060 P.ABF: .ASCII \*c1n*\<0><0><0>
:          0000003A 00068 P.ABG: .LONG 58
:

```

.PSECT \$OWNS,NOEXE,2

```

0000 00000 $STR$STRING:
01 0E 00002 .WORD 0
00000000' 00004 .BYTE 14, 1
0006 00008 $STR$STRING:
01 0E 0000A .WORD 6
00000000' 0000C .BYTE 14, 1
0005 00010 $STR$STRING:
01 0E 00012 .WORD 5
00000000' 00014 .BYTE 14, 1
0002 00018 $STR$STRING:
01 0E 0001A .WORD 2
00000000' 0001C .BYTE 14, 1
000C 00020 $STR$STRING:
01 0E 00022 .WORD 12
00000000' 00024 .BYTE 14, 1
0004 00028 $STR$STRING:
01 0E 0002A .WORD 4
00000000' 0002C .BYTE 14, 1
0004 00030 $STR$STRING:
01 0E 00032 .WORD 4
00000000' 00034 .BYTE 14, 1
0003 00038 $STR$STRING:
01 0E 0003A .WORD 3
00000000' 0003C .BYTE 14, 1
0006 00040 $STR$STRING:
01 0E 00042 .WORD 6
00000000' 00044 .BYTE 14, 1
0000 00048 $STR$STRING:
01 0E 0004A .WORD 0
00000000' 0004C .BYTE 14, 1
0006 00050 $STR$STRING:
01 0E 00052 .WORD 6
00000000' 00054 .BYTE 14, 1
0005 00058 $STR$STRING:
01 0E 0005A .WORD 5
00000000' 0005C .BYTE 14, 1
0002 00060 $STR$STRING:
01 0E 00062 .WORD 2
00000000' 00064 .BYTE 14, 1
000C 00068 $STR$STRING:
01 0E 0006A .WORD 12
.BYTE 14, 1

```

.....

: R


```

00000000' 0006C .ADDRESS P.AAV
0004 00070 $STR$STRING:
      .WORD 4
01 0E 00072 .BYTE 14, 1
00000000' 00074 .ADDRESS P.AAZ
0004 00078 $STR$STRING:
      .WORD 4
01 0E 0007A .BYTE 14, 1
00000000' 0007C .ADDRESS P.ABA
0003 00080 $STR$STRING:
      .WORD 3
01 0E 00082 .BYTE 14, 1
00000000' 00084 .ADDRESS P.ABC
0006 00088 $STR$STRING:
      .WORD 6
01 0E 0008A .BYTE 14, 1
00000000' 0008C .ADDRESS P.ABE

```

```

.EXTRN CMDBLK, TOCOOB, STCI0B
.EXTRN CHROUT, TMPSTR, TXTLEN
.EXTRN HL N, HLLIST, HLDSP
.EXTRN MAJOR, MINOR, PAGEN
.EXTRN NXTSTC, RNO_EX, RNO_FG
.EXTRN RNO_TB, TMS_EX, TMS_FG
.EXTRN TMS_TB, PRESTR, SFXSTR
.EXTRN LP, INFLIN, EXTLIN
.EXTRN LINE, LENPAG, TXTPAG
.EXTRN PTRPAG, LENSCT, TXTSCT
.EXTRN PTRSCT, RMARGIN
.EXTRN WRAP, LINE INDENT
.EXTRN SPLIT, FMTXT, INSREF
.EXTRN PACPAG, PACBAS, PACSEC
.EXTRN PACXXX, OPEN ERROR
.EXTRN XPOSOPEN, XPOSGET
.EXTRN XPOSFAILURE, XST$COPY
.EXTRN STR$FAILURE, XPOSPUT
.EXTRN XST$JOIN, XST$ASCII
.EXTRN XPOS$CLOSE

```

.PSECT \$CODE\$,NOWRT,2

```

OFFC 00000
5B 0000000G EF 9E 00002 .ENTRY HDR X, Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11 ; 0582
5E 0000000G 08 C2 00009 MOVAB $STR$TARGET, R11
      EF D5 0000C SUBL2 #8, SP
      01 12 00012 TSTL CMDBLK+4
      04 00014 BNEQ 1$
      01 D0 00015 RET
2E 5A 01 D0 00015 1$: MOVL #1, FIRST_RECORD
56 04 AC D0 00018 MOVL IOB, R6
2C A6 00010001 8F C8 0001C BISL2 #65537, 46(R6)
A6 0000000G 01 90 00024 MOVB #1, 44(R6)
      EF 9F 00028 PUSHAB OPEN ERROR
      7E D4 0002E CLRL -(SP)
      56 DD 00030 PUSHL R6
0000000G EF 03 FB 00032 CALLS #3, XPOSOPEN
58 38 A6 9E 00039 MOVAB 56(R6), R8
34 A6 08 B0 0003D 2$: MOVW #8, 52(R6)

```

.....

.....
0627
0629
0632
0640
0634

36	A6	02	90	00041	MOV	#2, 54(R6)		
2C	A6	06	90	00045	MOV	#6, 44(R6)		
		00000000G	EF	9F 00049	PUSHAB	XPOS\$FAILURE		
			7E	D4 0004F	CLRL	-(SP)		
			56	DD 00051	PUSHL	R6		
00000000G	EF	03	FB	00053	CALLS	#3, XPOS\$GET		
00208001	8F	50	D1	0005A	CMPL	R0, #2129921		
			03	13 00061	BEQL	3\$		
			OCF0	31 00063	BRW	74\$		
	50	68	D0	00066	MOVL	(R8), R0		0640
00000000G	EF	04	D0	00069	MOVL	(R0), MAJOR		0642
00000000G	EF	04	A0	00070	MOVL	4(R0), MINOR		0643
			51	D4 00078	CLRL	R1		0648
	02	00000000G	EF	D1 0007A	CMPL	MAJOR, #2		
			02	12 00081	BNEQ	4\$		
			51	D6 00083	INCL	R1		
			50	D4 00085	CLRL	R0		
	01	00000000G	EF	D1 00087	CMPL	MINOR, #1		
			02	12 0008E	BNEQ	5\$		
			50	D6 00090	INCL	R0		
	52		51	D2 00092	MCOML	R1, R2		
50			52	CA 00095	BICL2	R2, R0		
01			50	D1 00098	CMPL	R0, #1		
			30	12 0009B	BNEQ	7\$		
	34	A6	10	B0 0009D	MOVW	#16, 52(R6)		0653
	36	A6	02	90 000A1	MOV	#2, 54(R6)		
	2C	A6	06	90 000A5	MOV	#6, 44(R6)		
		00000000G	EF	9F 000A9	PUSHAB	XPOS\$FAILURE		
			7E	D4 000AF	CLRL	-(SP)		
			56	DD 000B1	PUSHL	R6		
00000000G	EF	03	FB	000B3	CALLS	#3, XPOS\$GET		
			50	D4 000BA	CLRL	I		0661
00000000G	EF	40	00	B840 D0 000BC	MOVL	@(R8)[I], PAGEN[I]		0662
F2			50	03 F3 000C6	AOBLEQ	#3, I, 6\$		
			FF70	31 000CA	BRW	2\$		0645
	04	00000000G	EF	D1 000CD	CMPL	MAJOR, #4		0667
			03	13 000D4	BEQL	8\$		
			054D	31 000D6	BRW	35\$		
	59	00000000G	EF	3C 000D9	MOVZWL	MINOR, TYPE		0675
	34	A6	04	B0 000E0	MOV	#4, 52(R6)		0676
	36	A6	02	90 000E4	MOV	#2, 54(R6)		
	2C	A6	06	90 000E8	MOV	#6, 44(R6)		
		00000000G	EF	9F 000EC	PUSHAB	XPOS\$FAILURE		
			7E	D4 000F2	CLRL	-(SP)		
			56	DD 000F4	PUSHL	R6		
00000000G	EF	03	FB	000F6	CALLS	#3, XPOS\$GET		
00000000G	EF	00	B8	D0 000FD	MOVL	@(R8), TXTLEN		0677
	50	00000000G	EF	D0 00105	MOVL	TXTLEN, R0		0679
			24	15 0010C	BLEQ	9\$		
	50		03	C0 0010E	ADDL2	#3, R0		
	50		04	C6 00111	DIVL2	#4, R0		
	50		04	A5 00114	MULW3	#4, R0, 52(R6)		
34	A6		02	90 00119	MOV	#2, 54(R6)		
	36	A6	06	90 0011D	MOV	#6, 44(R6)		
	2C	A6	00000000G	EF	9F 00121	PUSHAB	XPOS\$FAILURE	
				7E	D4 00127	CLRL	-(SP)	
			56	DD 00129	PUSHL	R6		0681

00000000G	EF	03	FB	0012B	CALLS	#3, XPOSGET	
	01	59	D1	00132	CMPL	TYPE, #1	0683
		68	12	00135	BNEQ	11\$	
	6E	00000000G	EF	B0	MOVW	TXLEN, \$STR\$STRING	0688
	02		AE	0E	MOVB	#14, \$STR\$STRING+2	
	03		AE	01	MOVB	#1, \$STR\$STRING+3	
	04		AE	68	MOVL	(R8), \$STR\$STRING+4	
		00000000G	EF	9F	PUSHAB	STR\$FAILURE	
			7E	D4	CLRL	-(SP)	
			5B	DD	PUSHL	R11	
		0C	AE	9F	PUSHAB	\$STR\$STRING	
			7E	D4	CLRL	-(SP)	
00000000G	EF	05	FB	00159	CALLS	#5, XST\$COPY	
00000000G	EF	6B	3C	00160	MOVZWL	TMPSTR, R0	
00000000G	EF	50	C0	00163	ADDL2	R0, CHR0UT	
00000000G	EF	6B	9E	0016A	MOVAB	\$IOB\$OUTPUT, IOB\$+68	
00000000G	EF	07	90	00171	MOVB	#7, IOB\$+44	
		00000000G	EF	9F	PUSHAB	XPOSFAILURE	
			7E	D4	CLRL	-(SP)	
		00000000G	EF	9F	PUSHAB	IOB\$	
00000000G	EF	03	FB	00186	CALLS	#3, XPOSPUT	
00000000G	EF	01	E1	0018D	BBC	#1, CMDBLK, 10\$	
00000000G	EF	00	FB	00195	CALLS	#0, SPLIT	
			FE9E	31	BRW	2\$	0683
		03	5A	E8	BLBS	FIRST_RECORD, 12\$	0694
			02CF	31	BRW	25\$	
03 00000000G	EF	01	E0	001A5	BBS	#1, CMDBLK, 13\$	0700
		00D7	31	001AD	BRW	16\$	
		00000000G	EF	9F	PUSHAB	STR\$FAILURE	0703
			7E	D4	CLRL	-(SP)	
			5B	DD	PUSHL	R11	
		00000000'	EF	9F	PUSHAB	\$STR\$STRING	
			7E	D4	CLRL	-(SP)	
00000000G	EF	05	FB	001C2	CALLS	#5, XST\$COPY	
00000000G	50	6B	3C	001C9	MOVZWL	TMPSTR, R0	
00000000G	EF	50	C0	001CC	ADDL2	R0, CHR0UT	
00000000G	EF	6B	9E	001D3	MOVAB	\$IOB\$OUTPUT, IOB\$+68	
00000000G	EF	07	90	001DA	MOVB	#7, IOB\$+44	
		00000000G	EF	9F	PUSHAB	XPOSFAILURE	
			7E	D4	CLRL	-(SP)	
		00000000G	EF	9F	PUSHAB	IOB\$	
00000000G	EF	03	FB	001EF	CALLS	#3, XPOSPUT	
00000000G	EF	01	E1	001F6	BBC	#1, CMDBLK, 14\$	
00000000G	EF	00	FB	001FE	CALLS	#0, SPLIT	
		00000000G	EF	D5	TSTL	HL N	0705
			55	13	BEQL	15\$	
		00000000G	EF	9F	PUSHAB	STR\$FAILURE	
			7E	D4	CLRL	-(SP)	
			5B	DD	PUSHL	R11	
		00000000'	EF	9F	PUSHAB	\$STR\$STRING	
			7E	D4	CLRL	-(SP)	
00000000G	EF	05	FB	0021F	CALLS	#5, XST\$COPY	
00000000G	50	6B	3C	00226	MOVZWL	TMPSTR, R0	
00000000G	EF	50	C0	00229	ADDL2	R0, CHR0UT	
00000000G	EF	6B	9E	00230	MOVAB	\$IOB\$OUTPUT, IOB\$+68	
00000000G	EF	07	90	00237	MOVB	#7, IOB\$+44	
		00000000G	EF	9F	PUSHAB	XPOSFAILURE	

		7E	D4	00244	CLRL	-(SP)	
		EF	9F	00246	PUSHAB	IOB\$	
07	00000000G	03	FB	00247	CALLS	#3, XPO\$PUT	
	00000000G	01	E1	00253	BBC	#1, CMDBLK, 15\$	
	00000000G	00	FB	0025B	CALLS	#0, SPLIT	
	00000000'	EF	9F	00262	PUSHAB	\$STR\$STRING2	15\$:
	08	AC	DD	00268	PUSHL	TITLE	
	00000000'	EF	9F	0026B	PUSHAB	\$STR\$STRING0	
00000000G		03	FB	00271	CALLS	#3, XST\$JOIN	
	00000000G	EF	9F	00278	PUSHAB	STR\$FAILURE	
	0801	7E	D4	0027E	CLRL	-(SP)	
		8F	BB	00280	PUSHR	#^M<R0,R11>	
	01	9F	31	00284	BRW	23\$	
	00000000G	EF	9F	00287	PUSHAB	STR\$FAILURE	16\$:
		7E	D4	0028D	CLRL	-(SP)	
	00000000'	5B	DD	0028F	PUSHL	R11	
		EF	9F	00291	PUSHAB	\$STR\$STRING	
		7E	D4	00297	CLRL	-(SP)	
00000000G		05	FB	00299	CALLS	#5, XST\$COPY	
		6B	3C	002A0	MOVZWL	TMPSTR, R0	
00000000G		50	C0	002A3	ADDL2	R0, CHROUT	
00000000G		6B	9E	002AA	MOVAB	\$IOB\$OUTPUT, IOB\$+68	
00000000G		07	90	002B1	MOVB	#7, IOB\$+44	
	00000000G	EF	9F	002B8	PUSHAB	XPO\$FAILURE	
		7E	D4	002BE	CLRL	-(SP)	
	00000000G	EF	9F	002C0	PUSHAB	IOB\$	
07	00000000G	03	FB	002C6	CALLS	#3, XPO\$PUT	
	00000000G	01	E1	002CD	BBC	#1, CMDBLK, 17\$	
	00000000G	00	FB	002D5	CALLS	#0, SPLIT	
	00000000G	EF	D5	002DC	TSTL	HL N	17\$:
		12	13	002E2	BEQL	18\$	
	00000000G	EF	9F	002E4	PUSHAB	STR\$FAILURE	
		7E	D4	002EA	CLRL	-(SP)	
	00000000'	5B	DD	002EC	PUSHL	R11	
		EF	9F	002EE	PUSHAB	\$STR\$STRING	
	00000000G	10	11	002F4	BRB	19\$	
	00000000G	EF	9F	002F6	PUSHAB	STR\$FAILURE	18\$:
		7E	D4	002FC	CLRL	-(SP)	
	00000000'	5B	DD	002FE	PUSHL	R11	
		EF	9F	00300	PUSHAB	\$STR\$STRING	
		7E	D4	00306	CLRL	-(SP)	19\$:
00000000G		05	FB	00308	CALLS	#5, XST\$COPY	
		6B	3C	0030F	MOVZWL	TMPSTR, R0	
00000000G		50	C0	00312	ADDL2	R0, CHROUT	
00000000G		6B	9E	00319	MOVAB	\$IOB\$OUTPUT, IOB\$+68	
00000000G		07	90	00320	MOVB	#7, IOB\$+44	
	00000000G	EF	9F	00327	PUSHAB	XPO\$FAILURE	
		7E	D4	0032D	CLRL	-(SP)	
	00000000G	EF	9F	0032F	PUSHAB	IOB\$	
07	00000000G	03	FB	00335	CALLS	#3, XPO\$PUT	
	00000000G	01	E1	0033C	BBC	#1, CMDBLK, 20\$	
	00000000G	00	FB	00344	CALLS	#0, SPLIT	
	00000000G	50	D0	0034B	MOVL	INDENTS, R0	20\$:
		70	13	00352	BEQL	21\$	
		7E	D4	00354	CLRL	-(SP)	
		50	DD	00356	PUSHL	R0	
	0903	7E	8F	3C	00358	MOVZWL	#2307, -(SP)

0707

0711

0712

0714

0716

00000000G	EF		03	FB	0035D	CALLS	#3, XST\$ASCII		
			50	DD	00364	PUSHL	R0		
00000000G	EF	00000000'	02	FB	00366	PUSHAB	\$STR\$STRING0		
			02	FB	0036C	CALLS	#2, XST\$JOIN		
00000000G	EF	00000000G	07	9F	00373	PUSHAB	STR\$FAILURE		
			7E	D4	00379	CLRL	-(SP)		
		0801	8F	BB	0037B	PUSHR	#*M<R0,R11>		
			7E	D4	0037F	CLRL	-(SP)		
00000000G	EF		05	FB	00381	CALLS	#5, XST\$COPY		
	50		6B	3C	00388	MOVZWL	TMPSTR, R0		
00000000G	EF		50	C0	0038B	ADDL2	R0, CHR0UT		
00000000G	EF		6B	9E	00392	MOVAB	\$IOB\$OUTPUT, IOB\$+68		
00000000G	EF		07	90	00399	MOVB	#7, IOB\$+44		
		00000000G	07	9F	003A0	PUSHAB	XPO\$FAILURE		
			7E	D4	003A6	CLRL	-(SP)		
		00000000G	07	9F	003A8	PUSHAB	IOB\$		
07 00000000G	EF		03	FB	003AE	CALLS	#3, XPO\$PUT		
00000000G	EF		01	F1	003B5	BBC	#1, CMDBLK, 21\$		
00000000G	EF		00	FB	003BD	CALLS	#0, SPLIT		
		00000000G	07	9F	003C4	PUSHAB	STR\$FAILURE		0718
			7E	D4	003CA	CLRL	-(SP)		
		08	5B	DD	003CC	PUSHL	R11		
			7E	D4	003CE	PUSHL	TITLE		
00000000G	EF		05	FB	003D3	CALLS	#5, XST\$COPY		
	50		6B	3C	003DA	MOVZWL	TMPSTR, R0		
00000000G	EF		50	C0	003DD	ADDL2	R0, CHR0UT		
00000000G	EF		6B	9E	003E4	MOVAB	\$IOB\$OUTPUT, IOB\$+68		
00000000G	EF		07	90	003EB	MOVB	#7, IOB\$+44		
		00000000G	07	9F	003F2	PUSHAB	XPO\$FAILURE		
			7E	D4	003F8	CLRL	-(SP)		
		00000000G	07	9F	003FA	PUSHAB	IOB\$		
07 00000000G	EF		03	FB	00400	CALLS	#3, XPO\$PUT		
00000000G	EF		01	E1	00407	BBC	#1, CMDBLK, 22\$		
00000000G	EF		00	FB	0040F	CALLS	#0, SPLIT		
		00000000G	07	9F	00416	PUSHAB	STR\$FAILURE		0719
			7E	D4	0041C	CLRL	-(SP)		
		00000000'	5B	DD	0041E	PUSHL	R11		
			7E	D4	00420	PUSHAB	\$STR\$STRING		
00000000G	EF		05	FB	00428	CALLS	#5, XST\$COPY		
	50		6B	3C	0042F	MOVZWL	TMPSTR, R0		
00000000G	EF		50	C0	00432	ADDL2	R0, CHR0UT		
00000000G	EF		6B	9E	00439	MOVAB	\$IOB\$OUTPUT, IOB\$+68		
00000000G	EF		07	90	00440	MOVB	#7, IOB\$+44		
		00000000G	07	9F	00447	PUSHAB	XPO\$FAILURE		
			7E	D4	0044D	CLRL	-(SP)		
		00000000G	07	9F	0044F	PUSHAB	IOB\$		
07 00000000G	EF		03	FB	00455	CALLS	#3, XPO\$PUT		
00000000G	EF		01	E1	0045C	BBC	#1, CMDBLK, 24\$		
00000000G	EF		00	FB	00464	CALLS	#0, SPLIT		
00000000G	EF		01	D0	00468	MOVL	#1, HL N		0722
			5A	D4	00472	CLRL	FIRST_RECORD		0723
00000000G	EF	00000000G	07	9E	00474	MOVAB	LINE, LP		0729
		00000000G	07	D4	0047F	CLRL	INTLIN		
		00000000G	07	D4	00485	CLRL	EXTLIN		
00000000G	EF	00000000G	07	D0	0048B	MOVL	INDENTS, LINE_INDENT		0730

	00000000G	EF	00000000G	EF	D0	00496	MOVL	LINE INDENT, EXTLIN	0731	
57	00000000G	EF	00000000G	EF	C1	004A1	ADDL3	INDENTS+4, INDENTS, R7	0732	
	00000000G	EF		57	D0	004AD	MOVL	R7, WRAP		
	00000000G	EF	00000000G	EF	D0	004B4	MOVL	CMDBLK+24, RMARGIN	0733	
1C	00000000G	EF		01	E1	004BF	BBC	#1, CMDBLK, 26\$	0735	
00000000G	FF	EF		06	28	004C7	MOV3	#6, P.AAP, @LP	0737	
	00000000G	EF		06	C0	004D3	ADDL2	#6, LP		
	00000000G	EF		06	C0	004DA	ADDL2	#6, INTLIN		
	00000000G	EF		2F	11	004E1	BRB	27\$	0735	
	00000000G	EF		57	D1	004E3	26\$:	CMPL	R7, EXTLIN	0739
				26	15	004EA	BLEQ	27\$		
57	00000000G	EF	00000000G	EF	C2	004EC	SUBL2	EXTLIN, R7		
20	6E	00000000G	FF	2C	004F3	MOV3	#0, (SP), #32, R7, @LP			
	00000000G	EF		FF	004F8					
	00000000G	EF		57	C0	004FD	ADDL2	R7, LP		
	00000000G	EF		57	C0	00504	ADDL2	R7, INTLIN		
	00000000G	EF		57	C0	0050B	ADDL2	R7, EXTLIN		
				52	D4	00512	27\$:	CLRL	R2	0741
				59	D1	00514	CMPL	TYPE, #2		
				49	12	00517	BNEQ	30\$		
				52	D6	00519	INCL	R2		
18	00000000G	EF	00000000G	EF	9E	0051B	MOVAB	TXTPAG, PTRPAG	0748	
	00000000G	EF		03	E0	00526	BBS	#3, CMDBLK, 28\$	0750	
				0A	DD	0052E	PUSHL	#10	0752	
				EF	9F	00530	PUSHAB	PTRPAG		
	00000000G	7E	00000000G	EF	3C	00536	MOVZWL	PAGEN+14, -(SP)		
		EF		03	FB	0053D	CALLS	#3, PACBAS		
				13	11	00544	BRB	29\$		
				EF	9F	00546	28\$:	PUSHAB	PTRPAG	0754
				EF	9F	0054C	PUSHAB	PAGEN		
	00000000G	EF	00000000G	02	FB	00552	CALLS	#2, PACPAG		
	00000000G	EF		50	D0	00559	29\$:	MOVL	R0, LENPAG	
				06	11	00560	BRB	31\$		0741
				EF	D4	00562	30\$:	CLRL	LENPAG	0758
50	00000000G	EF	00000000G	EF	C3	00568	31\$:	SUBL3	LENPAG, CMDBLK+24, R0	0763
	00000000G	EF	FF	A0	9E	00574	MOVAB	-1(R0), RMARGIN		
	00000000G	EF	00000000G	EF	D0	0057C	MOVL	EXTLIN, WRAP	0764	
				68	DD	00587	PUSHL	(R8)	0768	
				EF	DD	00589	PUSHL	TXLEN		
	00000000G	EF		02	FB	0058F	CALLS	#2, FMTTXT		
				52	E9	00596	BLBC	R2, 32\$	0770	
	00000000G	EF		00	FB	00599	CALLS	#0, INSREF	0772	
				1C	11	005A0	BRB	33\$		
14	00000000G	EF		01	E1	005A2	32\$:	BBC	#1, CMDBLK, 33\$	0776
	00000000G	FF		8F	90	005AA	MOVB	#64, @LP		
				EF	D6	005B2	INCL	LP		
				EF	D6	005B8	INCL	INTLIN		
				6E	B0	005BE	33\$:	MOVW	INTLIN, \$STR\$STRING	0782
				0E	90	005C5	MOVB	#14, \$STR\$STRING+2		
02	AE			01	90	005C9	MOVB	#1, \$STR\$STRING+3		
03	AE			EF	9E	005CD	MOVAB	LINE, \$STR\$STRING+4		
04	AE	00000000G		EF	9F	005D5	PUSHAB	STR\$FAILURE		
		00000000G		7E	D4	005DB	CLRL	-(SP)		
				5B	DD	005DD	PUSHL	R11		
				AE	9F	005DF	PUSHAB	\$STR\$STRING		
				7E	D4	005E2	CLRL	-(SP)		
	00000000G	EF		05	FB	005E4	CALLS	#5, XST\$COPY		

		50	6B	3C	005EB	MOVZWL	TMPSTR, R0		
	00000000G	EF	50	C0	005EE	ADDL2	R0, CHROUT		
	00000000G	EF	6B	9E	005F5	MOVAB	\$IOB\$OUTPUT, IOB\$+68		
	00000000G	EF	07	90	005FC	MOVB	#7, IOB\$+44		
		00000000G	EF	9F	00603	PUSHAB	XPOS\$FAILURE		
			7E	D4	00609	CLRL	-(SP)		
		00000000G	EF	9F	0060B	PUSHAB	IOB\$		
	00000000G	EF	03	FB	00611	CALLS	#3, XPOS\$PUT		
03	00000000G	EF	01	E0	00618	BBS	#1, CMDBLK, 34\$		
			0719	31	00620	BRW	73\$		
			070F	31	00623	BRW	72\$		
			50	D4	00626	CLRL	R0	0788	
	OC	AC	00000000G	EF	D1	00628	CMPL	MINOR, INFO	
				02	12	00630	BNEQ	36\$	
				50	D6	00632	INCL	R0	
				52	D2	00634	MCOML	R1, R2	
				50	CA	00637	BICL2	R2, R0	
				01	D1	0063A	CMPL	R0, #1	
				03	13	0063D	BEQL	37\$	
				00F7	31	0063F	BRW	41\$	
				OC	B0	00642	MOVW	#12, 52(R6)	0800
	34	A6		02	90	00646	MOVB	#2, 54(R6)	
	36	A6		06	90	0064A	MOVB	#6, 44(R6)	
	2C	A6		EF	9F	0064E	PUSHAB	XPOS\$FAILURE	
			00000000G	7E	D4	00654	CLRL	-(SP)	
				56	DD	00656	PUSHL	R6	
				03	FB	00658	CALLS	#3, XPOS\$GET	
	00000000G	EF	50	D0	0065F	MOVL	(R8), R0	0804	
				60	D0	00662	MOVL	(R0), HLLIST	0806
	00000000G	EF	04	A0	00669	MOVL	4(R0), HLDSP	0807	
	00000000G	EF	52	08	A0	00671	MOVL	8(R0), PREFIX_LEN	0808
				25	13	00675	BEQL	38\$	0813
				50	03	A2	9E	00677	
				50	04	C6	0067B	MOVAB	3(R2), R0
				50	04	A5	0067E	DIVL2	#4, R0
34	A6			02	90	00683	MULW3	#4, R0, 52(R6)	
				06	90	00687	MOVB	#2, 54(R6)	
	36	A6		EF	9F	0068B	MOVB	#6, 44(R6)	
	2C	A6		7E	D4	00691	PUSHAB	XPOS\$FAILURE	
			00000000G	56	DD	00693	CLRL	-(SP)	
				03	FB	00695	PUSHL	R6	
				52	B0	0069C	CALLS	#3, XPOS\$GET	
				0E	90	0069F	MOVW	PREFIX_LEN, \$STR\$STRING	0815
	02	AE		01	90	006A3	MOVB	#14, \$STR\$STRING+2	
	03	AE		68	D0	006A7	MOVB	#1, \$STR\$STRING+3	
	04	AE		EF	9F	006AB	MOVL	(R8), \$STR\$STRING+4	
			00000000G	7E	D4	006B1	PUSHAB	STR\$FAILURE	
				EF	9F	006B3	CLRL	-(SP)	
			00000000G	AE	9F	006B9	PUSHAB	\$STR\$TARGET	
			OC	7E	D4	006BC	PUSHAB	\$STR\$STRING	
				05	FB	006BE	CLRL	-(SP)	
	00000000G	EF	04	B0	006C5	CALLS	#5, XST\$COPY	0819	
				02	90	006C9	MOVW	#4, 52(R6)	
	34	A6		06	90	006CD	MOVB	#2, 54(R6)	
	36	A6		EF	9F	006D1	MOVB	#6, 44(R6)	
	2C	A6		7E	D4	006D7	PUSHAB	XPOS\$FAILURE	
			00000000G	56	DD	006D9	CLRL	-(SP)	
							PUSHL	R6	

	00000000G	EF	00	03	FB	006DB	CALLS	#3, XPOSGET	
		52		88	D0	006E2	MOVL	@0(R8), SUFFIX_LEN	0820
				25	13	006E6	BEQL	39\$	0824
		50	03	A2	9E	006E8	MOVAB	3(R2), R0	
34	A6	50		04	C6	006EC	DIVL2	#4, R0	
		50		04	A5	006EF	MULW3	#4, R0, 52(R6)	
	36	A6		02	90	006F4	MOVB	#2, 54(R6)	
	2C	A6		06	90	006F8	MOVB	#6, 44(R6)	
			00000000G	EF	9F	006FC	PUSHAB	XPOSFAILURE	
				7E	D4	00702	CLRL	-(SP)	
				56	DD	00704	PUSHL	R6	
	00000000G	EF		03	FB	00706	CALLS	#3, XPOSGET	
		6E		52	B0	0070D	MOVW	SUFFIX_LEN, \$STR\$STRING	0826
		02		0E	90	00710	MOVB	#14, \$STR\$STRING+2	
		03		01	90	00714	MOVB	#1, \$STR\$STRING+3	
		04		68	D0	00718	MOVL	(R8), \$STR\$STRING+4	
			00000000G	EF	9F	0071C	PUSHAB	STR\$FAILURE	
				7E	D4	00722	CLRL	-(SP)	
			00000000G	EF	9F	00724	PUSHAB	\$STR\$TARGET	
			0C	AE	9F	0072A	PUSHAB	\$STR\$STRING	
				7E	D4	0072D	CLRL	-(SP)	
	00000000G	EF		05	FB	0072F	CALLS	#5, XST\$COPY	
				F904	31	00736	BRW	2\$	0645
				52	D4	00739	CLRL	R2	0829
				52	D1	0073B	CMPL	MINOR, TXT	
	10	AC	00000000G	EF	12	00743	BNEQ	42\$	
				52	D6	00745	INCL	R2	
				50	D4	00747	CLRL	R0	
	14	AC	00000000G	EF	D1	00749	CMPL	MINOR, TXT_U	
				02	12	00751	BNEQ	43\$	
				50	D6	00753	INCL	R0	
				52	C8	00755	BISL2	R2, R0	
				53	D2	00758	MCOML	R1, R3	
				50	CA	0075B	BICL2	R3, R0	
				01	D1	0075E	CMPL	R0, #1	
				D3	12	00761	BNEQ	40\$	
	34	A6		04	B0	00763	MOVW	#4, 52(R6)	0840
	36	A6		02	90	00767	MOVB	#2, 54(R6)	
	2C	A6		06	90	0076B	MOVB	#6, 44(R6)	
			00000000G	EF	9F	0076F	PUSHAB	XPOSFAILURE	
				7E	D4	00775	CLRL	-(SP)	
				56	DD	00777	PUSHL	R6	
	00000000G	EF		03	FB	00779	CALLS	#3, XPOSGET	
	00000000G	EF	00	88	D0	00780	MOVL	@0(R8), TXTLEN	0841
		50	00000000G	EF	D0	00788	MOVL	TXTLEN, R0	0846
				24	13	0078F	BEQL	44\$	
				03	C0	00791	ADDL2	#3, R0	0848
				04	C6	00794	DIVL2	#4, R0	
34	A6	50		04	A5	00797	MULW3	#4, R0, 52(R6)	
		50		02	90	0079C	MOVB	#2, 54(R6)	
	36	A6		06	90	007A0	MOVB	#6, 44(R6)	
	2C	A6		EF	9F	007A4	PUSHAB	XPOSFAILURE	
			00000000G	7E	D4	007AA	CLRL	-(SP)	
				56	DD	007AC	PUSHL	R6	
	00000000G	EF		03	FB	007AE	CALLS	#3, XPOSGET	
	00000000G	EF	00000000G	EF	9E	007B5	MOVAB	LINE, LP	0853
			00000000G	EF	D4	007C0	CLRL	INTLIN	

00000000G	EF	00000000G	EF	D4	007C6	CLRL	EXTLIN	0854
00000000G	EF	00000000G	EF	DO	007CC	MOVL	INDENTS, LINE INDENT	0855
00000000G	EF	00000000G	EF	DO	007D7	MOVL	LINE INDENT, EXTLIN	0856
00000000G	EF	00000000G	EF	C1	007E2	ADDL3	INDENTS+4, INDENTS, WRAP	0857
00000000G	EF	00000000G	EF	DO	007F2	MOVL	CMDBLK+24, RMARGIN	0859
		03		5A	007FD	BLBS	FIRST_RECORD, 45\$	
				02CF	31 00800	BRW	58\$	
03	00000000G	EF		01	E0 00803	BBS	#1, CMDBLK, 46\$	0865
				00D7	31 0080B	BRW	49\$	
	00000000G	EF		9F	0080E	PUSHAB	STR\$FAILURE	0868
				7E	D4 00814	CLRL	-(SP)	
				5B	DD 00816	PUSHL	R11	
	00000000'	EF		9F	00818	PUSHAB	\$STR\$STRING	
				7E	D4 0081E	CLRL	-(SP)	
	00000000G	EF		05	FB 00820	CALLS	#5, XST\$COPY	
		50		6B	3C 00827	MOVZWL	TMPSTR, R0	
	00000000G	EF		50	C0 0082A	ADDL2	R0, CHROUT	
	00000000G	EF		6B	9E 00831	MOVAB	\$IOB\$OUTPUT, IOB\$+68	
	00000000G	EF		07	90 00838	MOVB	#7, IOB\$+44	
	00000000G	EF		9F	0083F	PUSHAB	XPOS\$FAILURE	
				7E	D4 00845	CLRL	-(SP)	
	00000000G	EF		9F	00847	PUSHAB	IOB\$	
	00000000G	EF		03	FB 0084D	CALLS	#3, XPOS\$PUT	
07	00000000G	EF		01	E1 00854	BBC	#1, CMDBLK, 47\$	
	00000000G	EF		00	FB 0085C	CALLS	#0, SPLIT	
	00000000G	EF		D5	00863	TSTL	HL N	0870
				55	13 00869	BEQL	48\$	
	00000000G	EF		9F	0086B	PUSHAB	STR\$FAILURE	
				7E	D4 00871	CLRL	-(SP)	
				5B	DD 00873	PUSHL	R11	
	00000000'	EF		9F	00875	PUSHAB	\$STR\$STRING	
				7E	D4 0087B	CLRL	-(SP)	
	00000000G	EF		05	FB 0087D	CALLS	#5, XST\$COPY	
		50		6B	3C 00884	MOVZWL	TMPSTR, R0	
	00000000G	EF		50	C0 00887	ADDL2	R0, CHROUT	
	00000000G	EF		6B	9E 0088E	MOVAB	\$IOB\$OUTPUT, IOB\$+68	
	00000000G	EF		07	90 00895	MOVB	#7, IOB\$+44	
	00000000G	EF		9F	0089C	PUSHAB	XPOS\$FAILURE	
				7E	D4 008A2	CLRL	-(SP)	
	00000000G	EF		9F	008A4	PUSHAB	IOB\$	
	00000000G	EF		03	FB 008AA	CALLS	#3, XPOS\$PUT	
07	00000000G	EF		01	E1 008B1	BBC	#1, CMDBLK, 48\$	
	00000000G	EF		00	FB 008B9	CALLS	#0, SPLIT	
	00000000'	EF		9F	008C0	PUSHAB	\$STR\$STRING2	0872
				08	AC DD 008C6	PUSHL	TITLE	
	000000C0'	EF		9F	008C9	PUSHAB	\$STR\$STRINGO	
	00000000G	EF		03	FB 008CF	CALLS	#3, XST\$JOIN	
	00000000G	EF		9F	008D6	PUSHAB	STR\$FAILURE	
				7E	D4 008DC	CLRL	-(SP)	
	0801			8F	BB 008DE	PUSHR	#*M<R0,R11>	
				019F	31 008E2	BRW	56\$	
	00000000G	EF		9F	008E5	PUSHAB	STR\$FAILURE	0876
				7E	D4 008EB	CLRL	-(SP)	
				5B	DD 008ED	PUSHL	R11	
	00000000'	EF		9F	008EF	PUSHAB	\$STR\$STRING	
				7E	D4 008F5	CLRL	-(SP)	
	00000000G	EF		05	FB 008F7	CALLS	#5, XST\$COPY	

			6B 3C 008FE	MOVZWL	TMPSTR, R0	
			50 C0 00901	ADDL2	R0, CHROUT	
			6B 9E 00908	MOVAB	\$IOB\$OUTPUT, IOB\$+68	
			07 90 0090F	MOVB	#7, IOB\$+44	
		00000000G	EF 9F 00916	PUSHAB	XPOSFAILURE	
			7E D4 0091C	CLRL	-(SP)	
		00000000G	EF 9F 0091E	PUSHAB	IOB\$	
07		00000000G	03 FB 00924	CALLS	#3, XPO\$PUT	
		00000000G	01 E1 0092B	BBC	#1, CMDBLK, 50\$	
		00000000G	00 FB 00933	CALLS	#0, SPLIT	
		00000000G	E: D5 0093A 50\$:	TSTL	HL N	
			12 13 00940	BEQL	51\$	
		00000000G	EF 9F 00942	PUSHAB	STR\$FAILURE	
			7E D4 00948	CLRL	-(SP)	
		00000000'	5B DD 0094A	PUSHL	R11	
			EF 9F 0094C	PUSHAB	\$STR\$STRING	
		00000000G	10 11 00952	BRB	52\$	
			EF 9F 00954 51\$:	PUSHAB	STR\$FAILURE	
			7E D4 0095A	CLRL	-(SP)	
		00000000'	5B DD 0095C	PUSHL	R11	
			EF 9F 0095E	PUSHAB	\$STR\$STRING	
			7E D4 00964 52\$:	CLRL	-(SP)	
		00000000G	05 FB 00966	CALLS	#5, XST\$COPY	
			6B 3C 0096D	MOVZWL	TMPSTR, R0	
		00000000G	50 C0 00970	ADDL2	R0, CHROUT	
		00000000G	6B 9E 00977	MOVAB	\$IOB\$OUTPUT, IOB\$+68	
		00000000G	07 90 0097E	MOVB	#7, IOB\$+44	
		00000000G	EF 9F 00985	PUSHAB	XPOSFAILURE	
			7E D4 0098B	CLRL	-(SP)	
		00000000G	EF 9F 0098D	PUSHAB	IOB\$	
07		00000000G	03 FB 00993	CALLS	#3, XPO\$PUT	
		00000000G	01 E1 0099A	BBC	#1, CMDBLK, 53\$	
		00000000G	00 FB 009A2	CALLS	#0, SPLIT	
		00000000G	EF D0 009A9 53\$:	MOVL	INDENTS, R0	
			70 13 009B0	BEQL	54\$	
			7E D4 009B2	CLRL	-(SP)	
			50 DD 009B4	PUSHL	R0	
		00000000G	7E 0903 8F 3C 009B6	MOVZWL	#2307, -(SP)	
			03 FB 009BB	CALLS	#3, XST\$ASCII	
		00000000'	50 DD 009C2	PUSHL	R0	
			EF 9F 009C4	PUSHAB	\$STR\$STRINGO	
		00000000G	02 FB 009CA	CALLS	#2, XST\$JOIN	
			EF 9F 009D1	PUSHAB	STR\$FAILURE	
		00000000G	7E D4 009D7	CLRL	-(SP)	
			8F BB 009D9	PUSHR	#*M<R0,R11>	
		0801	7E D4 009DD	CLRL	-(SP)	
		00000000G	05 FB 009DF	CALLS	#5, XST\$COPY	
			6B 3C 009E6	MOVZWL	TMPSTR, R0	
		00000000G	50 C0 009E9	ADDL2	R0, CHROUT	
		00000000G	6B 9E 009F0	MOVAB	\$IOB\$OUTPUT, IOB\$+68	
		00000000G	07 90 009F7	MOVB	#7, IOB\$+44	
		00000000G	EF 9F 009FE	PUSHAB	XPOSFAILURE	
			7E D4 00A04	CLRL	-(SP)	
		00000000G	EF 9F 00A06	PUSHAB	IOB\$	
07		00000000G	03 FB 00A0C	CALLS	#3, XPO\$PUT	
		00000000G	01 E1 00A13	BBC	#1, CMDBLK, 54\$	
		00000000G	00 FB 00A1B	CALLS	#0, SPLIT	

0877

0879

0881

		00000000G	EF	9F	00A22	54\$:	PUSHAB	STR\$FAILURE	0883
			7E	D4	00A28		CLRL	-(SP)	
			5B	DD	00A2A		PUSHL	R11	
		08	AC	DD	00A2C		PUSHL	TITLE	
			7E	D4	00A2F		CLRL	-(SP)	
		00000000G	EF	05	FB	00A31	CALLS	#5, XST\$COPY	
			50	6B	3C	00A38	MOVZWL	TMPSTR, R0	
		00000000G	EF	50	C0	00A3B	ADDL2	R0, CHR0UT	
		00000000G	EF	6B	9E	00A42	MOVAB	\$IOB\$OUTPUT, IOB\$+68	
		00000000G	EF	07	90	00A49	MOVAB	#7, IOB\$+44	
		00000000G	EF	9F	00A50		PUSHAB	XPO\$FAILURE	
			7E	D4	00A56		CLRL	-(SP)	
		00000000G	EF	9F	00A58		PUSHAB	IOB\$	
		07 00000000G	EF	03	FB	00A5E	CALLS	#3, XPO\$PUT	
		00000000G	EF	01	E1	00A65	BBC	#1, CMDBLK, 55\$	
		00000000G	EF	00	FB	00A6D	CALLS	#0, SPLIT	
		00000000G	EF	9F	00A74	55\$:	PUSHAB	STR\$FAILURE	0884
			7E	D4	00A7A		CLRL	-(SP)	
			5B	DD	00A7C		PUSHL	R11	
		00000000'	EF	9F	00A7E		PUSHAB	\$STR\$STRING	
			7E	D4	00A84	56\$:	CLRL	-(SP)	
		00000000G	EF	05	FB	00A86	CALLS	#5, XST\$COPY	
			50	6B	3C	00A8D	MOVZWL	TMPSTR, R0	
		00000000G	EF	50	C0	00A90	ADDL2	R0, CHR0UT	
		00000000G	EF	6B	9E	00A97	MOVAB	\$IOB\$OUTPUT, IOB\$+68	
		00000000G	EF	07	90	00A9E	MOVAB	#7, IOB\$+44	
		00000000G	EF	9F	00AA5		PUSHAB	XPO\$FAILURE	
			7E	D4	00AAB		CLRL	-(SP)	
		00000000G	EF	9F	00AAD		PUSHAB	IOB\$	
		07 00000000G	EF	03	FB	00AB3	CALLS	#3, XPO\$PUT	
		00000000G	EF	01	E1	00ABA	BBC	#1, CMDBLK, 57\$	
		00000000G	EF	00	FB	00AC2	CALLS	#0, SPLIT	
		00000000G	EF	01	D0	00AC9	MOVL	#1, HL N	
			5A	D4	00AD0	57\$:	CLRL	FIRST RECORD	0887
		1A 00000000G	EF	01	E1	00AD2	BBC	#1, CMDBLK, 59\$	0888
		FF 00000000'	EF	05	28	00ADA	MOV3	#5, P.ABF, @LP	0891
		00000000G	EF	05	C0	00AE6	ADDL2	#5, LP	
		00000000G	EF	05	C0	00AED	ADDL2	#5, INTLIN	
		03 00000000G	EF	04	E0	00AF4	BBS	#4, CMDBLK, 61\$	0893
			01	31	00AFC	60\$:	BRW	66\$	
		00000000G	EF	D1	00AFF	61\$:	CMP	CMDBLK+20, HL_N	0895
			17	18	00B0A		BGEQ	62\$	
		FFFFFFFF	8F	D1	00B0C		CMP	CMDBLK+20, #-1	0897
			E3	12	00B17		BNEQ	60\$	
		14 AC	00000000G	EF	D1	00B19	CMP	MINOR, TXT_U	0898
			D9	13	00B21		BEQ	60\$	
		00000000G	EF	9E	00B23	62\$:	MOVAB	TXTSCT, PTRSCT	0909
			0F	D4	00B2E		CLRL	LENSCT	0910
			0F	93	00B34		BITB	PAGEN, #15	0912
			2F	13	00B3B		BEQ	63\$	
		00000000G	EF	9F	00B3D		PUSHAB	PTRSCT	0918
		00000000G	EF	9F	00B43		PUSHAB	PAGEN	
		00000000G	EF	02	FB	00B49	CALLS	#2, PACSEC	
		50 00000000G	EF	C0	00B50		ADDL2	LENSCT, R0	
		00000000G	EF	A0	9E	00B57	MOVAB	1(R0), LENSCT	
		00000000G	FF	2D	90	00B5F	MOVAB	#45, @PTRSCT	0919
			00000000G	EF	D6	00B66	INCL	PTRSCT	

		00000000G	EF	DD	00B6C	63\$:	PUSHL	HLDSP		0922
		00000000G	EF	9F	00B72		PUSHAB	PTRSCT		
		00000000G	EF	DD	00B78		PUSHL	HLLIST		
		00000000G	EF	03	FB	00B7E	CALLS	#3, PACXXX		
		00000000G	EF	50	C0	00B85	ADDL2	R0, LENSCT		
		00000000G	50	EF	3C	00B8C	MOVZWL	PRESTR, R0		0926
		00000000G	OF	13	00B93		BEQL	64\$		
		00000000G	EF	DD	00B95		PUSHL	PRESTR+4		0928
		00000000G	50	DD	00B9B		PUSHL	R0		
		00000000G	EF	02	FB	00B9D	CALLS	#2, FMTTXT		
		00000000G	EF	9F	00BA4	64\$:	PUSHAB	TXTSCT		0933
		00000000G	EF	DD	00BAA		PUSHL	LENSCT		
		00000000G	EF	02	FB	00BB0	CALLS	#2, FMTTXT		
		00000000G	54	EF	3C	00BB7	MOVZWL	SFXSTR, R4		0937
		00000000G	1F	13	00BBE		BEQL	65\$		
54	00	00000000'	EF	01	2D	00BC0	CMPCS	#1, P.ABG, #0, R4, @SFXSTR+4		0941
		00000000G	FF		00BC9					
		00000000G	OF	13	00BCE		BEQL	65\$		
		00000000G	EF	DD	00BD0		PUSHL	SFXSTR+4		0944
		00000000G	54	DD	00BD6		PUSHL	R4		
		00000000G	EF	02	FB	00BD8	CALLS	#2, FMTTXT		
	21	00000000G	EF	01	E0	00BDF	65\$:	BBS	#1, CMDBLK, 67\$	0948
		00000000G	FF	20	90	00BE7	MOVB	#32, @LP		
		00000000G	EF	D6	00BEE		INCL	LP		
		00000000G	EF	D6	00BF4		INCL	INTLIN		
		00000000G	EF	D6	00BFA		INCL	EXTLIN		
	16	00000000G	EF	01	E1	00C00	66\$:	BBC	#1, CMDBLK, 68\$	0951
		00000000G	FF	40	8F	90	67\$:	MOVB	#64, @LP	0953
		00000000G	EF	D6	00C10		INCL	LP		
		00000000G	EF	D6	00C16		INCL	INTLIN		
		00000000G	3B	11	00C1C		BRB	69\$		0951
	57	00000000G	EF	01	00C1E	68\$:	ADDL3	INDENTS+4, INDENTS, R7		0955
		00000000G	EF	57	D1	00C2A	CMPL	R7, EXTLIN		
		00000000G	EF	26	15	00C31	BLEQ	69\$		
		00000000G	EF	C2	00C33		SUBL2	EXTLIN, R7		
	57	00000000G	6E	00	2C	00C3A	MOVCS	#0, (SP), #32, R7, @LP		
		00000000G	FF		00C3F					
		00000000G	EF	57	C0	00C44	ADDL2	R7, LP		
		00000000G	EF	57	C0	00C4B	ADDL2	R7, INTLIN		
		00000000G	EF	57	C0	00C52	ADDL2	R7, EXTLIN		
		00000000G	EF	9E	00C59	69\$:	MOVAB	TXPAG, PTRPAG		0961
	18	00000000G	EF	03	E0	00C64	BBS	#3, CMDBLK, 70\$		0963
		00000000G	0A	DD	00C6C		PUSHL	#10		0969
		00000000G	EF	9F	00C6E		PUSHAB	PTRPAG		
		00000000G	7E	EF	3C	00C74	MOVZWL	PAGEN+14, -(SP)		
		00000000G	EF	03	FB	00C7B	CALLS	#3, PACBAS		
		00000000G	13	11	00C82		BRB	71\$		
		00000000G	EF	9F	00C84	70\$:	PUSHAB	PTRPAG		0976
		00000000G	EF	9F	00C8A		PUSHAB	PAGEN		
		00000000G	EF	02	FB	00C90	CALLS	#2, PACPAG		
		00000000G	EF	50	D0	00C97	71\$:	MOVL	R0, LENPAG	
		00000000G	EF	C3	00C9E		SUBL3	LENPAG, CMDBLK+24, R0		0982
	20	00000000G	EF	A0	9E	00CAA	MOVAB	-1(R0), RMARGIN		
		00000000G	EF	FF			MOVAB			
		00000000G	EF	D0	00CB2		MOVL	EXTLIN, WRAP		0983
		00000000G	EF	68	DD	00CBD	PUSHL	(R8)		0987
		00000000G	EF	DD	00CBF		PUSHL	TXTLEN		
		00000000G	EF	02	FB	00CC5	CALLS	#2, FMTTXT		

00000000G	EF	00	FB	00CCC	CALLS	#0, INSREF	:	0991
	6E	00000000G	EF	80 00CD3	MOVW	INTLIN, \$STR\$STRING	:	0995
02	AE		0E	90 00CDA	MOVB	#14, \$STR\$STRING+2	:	
03	AE		01	90 00CDE	MOVB	#1, \$STR\$STRING+3	:	
04	AE	00000000G	EF	9E 00CE2	MOVAB	LINE, \$STR\$STRING+4	:	
		00000000G	EF	9F 00CEA	PUSHAB	STR\$FAILURE	:	
			7E	D4 00CF0	CLRL	-(SP)	:	
			5B	DD 00CF2	PUSHL	R11	:	
		0C	AE	9F 00CF4	PUSHAB	\$STR\$STRING	:	
			7E	D4 00CF7	CLRL	-(SP)	:	
00000000G	EF		05	FB 00CF9	CALLS	#5, XST\$COPY	:	
	50		6B	3C 00D00	MOVZWL	TMPSTR, R0	:	
00000000G	EF		50	C0 00D03	ADDL2	R0, CHR\$OUT	:	
00000000G	EF		6B	9E 00D0A	MOVAB	\$IOB\$OUTPUT, IOB\$+68	:	
00000000G	EF		07	90 00D11	MOVB	#7, IOB\$+44	:	
		00000000G	EF	9F 00D18	PUSHAB	XPOS\$FAILURE	:	
			7E	D4 00D1E	CLRL	-(SP)	:	
		00000000G	EF	9F 00D20	PUSHAB	IOB\$:	
00000000G	EF		03	FB 00D26	CALLS	#3, XPOS\$PUT	:	
07 00000000G	EF		01	E1 00D2D	BBC	#1, CMDBLK, 73\$:	
00000000G	EF		00	FB 00D35 72\$:	CALLS	#0, SPLIT	:	
00000000G	EF	00000000G	EF	9E 00D3C 73\$:	MOVAB	LINE, LP	:	0996
		00000000G	EF	D4 00D47	CLRL	INTLIN	:	
		00000000G	EF	D4 00D4D	CLRL	EXTLIN	:	
			F2E7	31 00D53	BRW	2\$:	0634
	2E	A6	10	88 00D56 74\$:	BISB2	#16, 46(R6)	:	1003
	2C	A6	02	90 00D5A	MOVB	#2, 44(R6)	:	
		00000000G	EF	9F 00D5E	PUSHAB	XPOS\$FAILURE	:	
			7E	D4 00D64	CLRL	-(SP)	:	
			56	DD 00D66	PUSHL	R6	:	
00000000G	EF		03	FB 00D68	CALLS	#3, XPOS\$CLOSE	:	1004
			04	00D6F	RET		:	

; Routine Size: 3440 bytes, Routine Base: \$CODE\$ + 0000

```

: 691 1005 1 %SBTTL 'STCHDR - Process the .SEND TOC n captions temp file'
: 692 1006 1 GLOBAL ROUTINE stchr : NOVALUE =
: 693 1007 1 ++
: 694 1008 1
: 695 1009 1 FUNCTIONAL DESCRIPTION:
: 696 1010 1
: 697 1011 1 This routine reads the temp file, which contains page references
: 698 1012 1 and .SEND TOC records.
: 699 1013 1
: 700 1014 1 The .SEND TOC records are formatted into captions.
: 701 1015 1
: 702 1016 1 FORMAL PARAMETERS:
: 703 1017 1
: 704 1018 1 None
: 705 1019 1
: 706 1020 1 IMPLICIT INPUTS:
: 707 1021 1
: 708 1022 1     nxtstc - next .SEND TOC 'n' value to process
: 709 1023 1     cmdblk - command line information block
: 710 1024 1
: 711 1025 1 IMPLICIT OUTPUTS:
: 712 1026 1
: 713 1027 1 None
: 714 1028 1
: 715 1029 1 ROUTINE VALUE:
: 716 1030 1 COMPLETION CODES:
: 717 1031 1
: 718 1032 1 None
: 719 1033 1
: 720 1034 1 SIDE EFFECTS:
: 721 1035 1
: 722 1036 1 None
: 723 1037 1
: 724 1038 1 --
: 725 1039 1
: 726 1040 2 BEGIN
: 727 1041 2 |
: 728 1042 2 | Walk through the file until there are no more .SEND TOC 'n'
: 729 1043 2 | groups left to process only if we are processing HL 1's
: 730 1044 2 |
: 731 1045 2 WHILE (.nxtstc NEQ infinity) AND (.cmdblk [contents$g_headers] NEQ 0) DO
: 732 1046 2 BEGIN
: 733 1047 2
: 734 1048 2 LOCAL
: 735 1049 2     stcn,
: 736 1050 2     first_record;
: 737 1051 2
: 738 1052 2     first_record = true;
: 739 1053 2     stcn = .nxtstc;                                ! Value we're processing this time
: 740 1054 2     nxtstc = infinity;                              ! Value to process next time
: 741 1055 2     $xpo_open (iob = stciob, options = input, attributes = BINARY
: 742 1056 2     %IF %BLISS (BLISS32) %THEN , failure = open_error %FI
: 743 1057 2     );
: 744 1058 2
: 745 1059 2 WHILE $xpo_get (iob = stciob, fullwords = 2) EQL xpo$_normal DO
: 746 1060 2 BEGIN
: 747 1061 2 !

```

```

748      1062  4      ! Process a record group
749      1063  4      !
750      1064  4      BIND
751      1065  4      record_header = .stciob [iob$a_data] : VECTOR;
752      1066  4
753      1067  4      major = .record_header [0];
754      1068  4      minor = .record_header [1];
755      1069  4
756      1070  4      SELECTONE true OF
757      1071  4      SET
758      1072  4
759      1073  4      [(.major EQL maj_runoff) AND (.minor EQL min_page)] :
760      1074  5      BEGIN
761      1075  5      ! Process a page reference
762      1076  5
763      1077  5      $xpo_get (iob = stciob, fullwords = page_sct_size);
764      1078  5      BEGIN
765      1079  6
766      1080  6      BIND
767      1081  6      page_ref = .stciob [iob$a_data] : VECTOR;
768      1082  6
769      1083  6      MAP
770      1084  6      pagen : VECTOR;
771      1085  6
772      1086  6      INCR i FROM 0 TO page_sct_size - 1 DO
773      1087  6      pagen [i] = .page_ref [i];
774      1088  6
775      1089  6
776      1090  5      END;
777      1091  4      END;
778      1092  4
779      1093  4      [(.major EQL maj_send) AND (ABS (.minor) NEQ .stcn)] :
780      1094  5      BEGIN
781      1095  5      !
782      1096  5      ! A .SEND TOC with a value of n which does not match the
783      1097  5      ! value we're processing. Remember it if it is the smallest
784      1098  5      ! one which is larger than the value we're processing.
785      1099  5
786      1100  5      IF ABS (.minor) GTR .stcn THEN nextstc = MIN (ABS (.minor), .nextstc);
787      1101  5
788      1102  5      !
789      1103  5      ! Now skip over the text if any
790      1104  5
791      1105  5      $xpo_get (iob = stciob, fullwords = 1);
792      1106  5
793      1107  5      IF (.stciob [iob$a_data]) NEQ 0
794      1108  5      THEN
795      1109  5      $xpo_get (iob = stciob,
796      1110  5      fullwords = CH$ALLOCATION (.(.stciob [iob$a_data])));
797      1111  5
798      1112  4      END;
799      1113  4
800      1114  4      [(.major EQL maj_send) AND (.minor EQL .stcn)] :
801      1115  5      BEGIN
802      1116  5      !
803      1117  5      ! A caption title
804      1118  5

```


862 1176 5
863 1177 5
864 1178 5
865 1179 5
866 1180 5
867 1181 5
868 1182 5
869 1183 5
870 1184 6
871 1185 6
872 1186 6
873 1187 6
874 1188 6
875 1189 6
876 1190 7
877 1191 7
878 1192 7
879 1193 7
880 1194 7
881 1195 7
882 1196 7
883 1197 7
884 1198 7
885 1199 7
886 1200 7
887 1201 7
888 1202 7
889 1203 7
890 1204 8
891 1205 8
892 1206 8
893 1207 8
894 1208 8
895 1209 8
896 1210 9
897 1211 9
898 1212 9
899 1213 9
900 1214 9
901 1215 9
902 1216 9
903 1217 9
904 1218 8
905 1219 8
906 1220 8
907 1221 8
908 1222 8
909 1223 8
910 1224 8
911 1225 8
912 1226 7
913 1227 7
914 1228 6
915 1229 6
916 1230 5
917 1231 5
918 1232 5

```

!
$ppo_get (iob = stciob, fullwords = 1);
len = (.stciob [iob$a_data]);
$ppo_get (iob = stciob, fullwords = CH$ALLOCATION (.len));
ptr = CH$FIND_CH (.len, CH$PTR (.stciob [iob$a_data]), '%C',');

IF .cmdblk [contents$v_include_sections]
THEN
  BEGIN
    : Generate a section number if the user requested one.
    IF NOT CH$FAIL (.ptr)
    THEN
      BEGIN
        : User supplied a section number
        LOCAL
          sec_l,
          sec_p;

        sec_p = CH$PTR (.stciob [iob$a_data]);
        sec_l = CH$DIFF (.ptr, .sec_p);
        strip (sec_l, sec_p);

        IF .sec_l NEQ 0
        THEN
          BEGIN
            : User specified a non-null section number
            IF .pagen [sct_typ] NEQ 0
            THEN
              BEGIN
                : Document is divided into sections
                ptrsct = CH$PTR (txtsct);
                lensct = pacsec (pagen, ptrsct);
                fmttxt (.lensct, CH$PTR (txtsct));
                fmttxt (1, CH$PTR (UPLIT ('-')));
                END;

                fmttxt (.sec_l, .sec_p);

                IF NOT .cmdblk [contents$v_tms11]
                THEN
                  write_char (%C' ', counts_visually);

                END;

            END;

          END;

        END;

      END;

    IF .cmdblk [contents$v_tms11]
  
```

```

919 1233 5 THEN
920 1234 6 write_char (XC'a')
921 1235 5 ELSE
922 1236 5 pad ((.indents [0] + .indents [1] - .extlin));
923 1237 5
924 1238 5
925 1239 5 | Generate page number to get its length.
926 1240 5 ptrpag = CH$PTR (txtpag);
927 1241 5
928 1242 5
929 1243 5 IF NOT .cmdblck [contents$v_standard_page]
930 1244 5 THEN
931 1245 6 BEGIN
932 1246 6 | Use running page number
933 1247 6 lenpag = pacbas (.pagen [sct_run_page], ptrpag, 10);
934 1248 6 END
935 1249 6 ELSE
936 1250 6 BEGIN
937 1251 6 | Supply the usual page number
938 1252 6 lenpag = pacpag (pagen, ptrpag);
939 1253 6 END;
940 1254 6
941 1255 6
942 1256 6
943 1257 5
944 1258 5
945 1259 5 rmargin = .cmdblck [contents$g_page_width] - .lenpag - 1;
946 1260 5 wrap = .extlin;
947 1261 5
948 1262 5 | Process the text
949 1263 5
950 1264 5 IF CH$FAIL (.ptr)
951 1265 5 THEN
952 1266 5 ptr = CH$PTR (.stciob [iob$a_data])
953 1267 5 ELSE
954 1268 6 BEGIN
955 1269 6 | There was a section number in the line. Point past
956 1270 6 | the comma and update the length.
957 1271 6 ptr = CH$PLUS (.ptr, 1);
958 1272 6 len = .len - CH$DIFF (.ptr, CH$PTR (.stciob [iob$a_data]));
959 1273 6 END;
960 1274 6
961 1275 5
962 1276 5
963 1277 5 strip (len, ptr);
964 1278 5 fmttxt (.len, .ptr);
965 1279 5
966 1280 5 | Insert page reference into page if desired.
967 1281 5
968 1282 5 insref ();
969 1283 5
970 1284 5 | Write line to output file
971 1285 5
972 1286 5 put ((.intlin, CH$PTR (line)));
973 1287 5 clr_line ();
974 1288 4
975 1289 4
    
```



```

01 0E 000A2 .BYTE 14, 1
00000000' 000A4 .ADDRESS P.ABM
0002 000A8 $STR$STRING2:
          .WORD 2
01 0E 000AA .BYTE 14, 1
00000000' 000AC .ADDRESS P.ABN
000C 000B0 $STR$STRING:
          .WORD 12
01 0E 000B2 .BYTE 14, 1
00000000' 000B4 .ADDRESS P.ABR
0004 000B8 $STR$STRING:
          .WORD 4
01 0E 000BA .BYTE 14, 1
00000000' 000BC .ADDRESS P.ABS
0004 000C0 $STR$STRING:
          .WORD 4
01 0E 000C2 .BYTE 14, 1
00000000' 000C4 .ADDRESS P.ABT
0003 000C8 $STR$STRING0:
          .WORD 3
01 0E 000CA .BYTE 14, 1
00000000' 000CC .ADDRESS P.ABV
0006 000D0 $STR$STRING:
          .WORD 6
01 0E 000D2 .BYTE 14, 1
00000000' 000D4 .ADDRESS P.ABX

```

TMS_TITLES= P.ABH
RNO_TITLES= P.ABQ

```

          .PSECT $CODE$,NOWRT,2
          OFFC 00000
          .ENTRY STCHDR, Save R2,R3,R4,R5,R6,R7,R8,R9,R10,-
          R11
          MOVAB IOB$+68, R11
          MOVAB $STR$TARGET, R10
          MOVAB IOB$+52, R9
          SUBL2 #24, SP
          CMPL NXTSTC, #2147483647
          BEQL 2$
          TSTL CMDBLK+4
          BNEQ 3$
          RET
          MOVL #1, FIRST RECORD
          MOVL NXTSTC, STCN
          MOVL #2147483647, NXTSTC
          BISL2 #65537, IOB$+46
          MOVB #1, IOB$+44
          PUSHAB OPEN ERROR
          CLRL -(SP)
          PUSHAB IOB$
          CALLS #3, XPOSOPEN
          MOVW #8, IOB$+52
          MOVB #2, IOB$+54
          MOVB #6, IOB$+44
          PUSHAB XPOSFAILURE

```

			7E	D4	00074	CLRL	-(SP)		
		CC	A9	9F	00076	PUSHAB	IOB\$		
00000000G	EF		03	FB	00079	CALLS	#3, XPOSGET		
00208001	8F		50	D1	00080	CMPL	R0, #2129921		
			03	13	00087	BEQL	5\$		
			0715	31	00089	BRW	52\$		
	50	04	A9	D0	0008C	5\$:	MOVL	STCIOB+56, R0	1065
00000000G	EF		60	D0	00090		MOVL	(R0), MAJOR	1067
00000000G	EF	04	A0	D0	00097		MOVL	4(R0), MINOR	1068
			51	D4	0009F		CLRL	R1	1073
	02	00000000G	EF	D1	000A1		CMPL	MAJOR, #2	
			02	12	000A8		BNEQ	6\$	
			51	D6	000AA		INCL	R1	
			50	D4	000AC	6\$:	CLRL	R0	
	01	00000000G	EF	D1	000AE		CMPL	MINOR, #1	
			02	12	000B5		BNEQ	7\$	
			50	D6	000B7		INCL	R0	
	52		51	D2	000B9	7\$:	MCOML	R1, R2	
	50		52	CA	000BC		BICL2	R2, R0	
	01		50	D1	000BF		CMPL	R0, #1	
			30	12	000C2		BNEQ	10\$	
	69		10	B0	000C4		MOVW	#16, IOB\$+52	1078
	02		02	90	000C7		MOVB	#2, IOB\$+54	
	FB		06	90	000CB		MOVB	#6, IOB\$+44	
		00000000G	EF	9F	000CF		PUSHAB	XPOSFAILURE	
			7E	D4	000D5		CLRL	-(SP)	
		CC	A9	9F	000D7		PUSHAB	IOB\$	
00000000G	EF		03	FB	000DA		CALLS	#3, XPOSGET	
			50	D4	000E1		CLRL	I	1087
00000000GEF40		04	B940	D0	000E3	8\$:	MOVL	@STCIOB+56[I], PAGEN[I]	1088
F2	50		03	F3	000ED		AOBLEQ	#3, I, 8\$	
			FF6F	31	000F1	9\$:	BRW	4\$	1070
			52	D4	000F4	10\$:	CLRL	R2	1093
	03	00000000G	EF	D1	000F6		CMPL	MAJOR, #3	
			02	12	000FD		BNEQ	11\$	
			52	D6	0C0FF		INCL	R2	
	50	00000000G	EF	D0	00101	11\$:	MOVL	MINOR, R0	
			03	18	00108		BGEQ	12\$	
	50		50	CE	0010A		MNEGL	R0, R0	
			51	D4	0010D	12\$:	CLRL	R1	
	56		50	D1	0010F		CMPL	R0, STCN	
			02	13	00112		BEQL	13\$	
			51	D6	00114		INCL	R1	
	53		52	D2	00116	13\$:	MCOML	R2, R3	
	51		53	CA	00119		BICL2	R3, R1	
	01		51	D1	0011C		CMPL	R1, #1	
			72	12	0011F		BNEQ	17\$	
	50	00000000G	EF	D0	00121		MOVL	MINOR, R0	1100
			03	18	00128		BGEQ	14\$	
	50		50	CE	0012A		MNEGL	R0, R0	
	56		50	D1	0012D	14\$:	CMPL	R0, STCN	
			17	15	00130		BLEQ	16\$	
00000000G	EF		50	D1	00132		CMPL	R0, NXTSTC	
			07	15	00139		BLEQ	15\$	
	50	00000000G	EF	D0	00138		MOVL	NXTSTC, R0	
00000000G	EF		50	D0	00142	15\$:	MOVL	R0, NXTSTC	
	69		04	B0	00149	16\$:	MOVW	#4, IOB\$+52	1105

02	A9	02	90	0014C	MOVB	#2, IOB\$+54	
F8	A9	06	90	00150	MOVB	#6, IOB\$+44	
		00000000G	EF	9F 00154	PUSHAB	XPOSFAILURE	
			7E	D4 0015A	CLRL	-(SP)	
			CC	A9 9F 0015C	PUSHAB	IOB\$	
00000000G	EF	03	FB	0015F	CALLS	#3, XPOSGET	1107
	50	04	B9	D0 00166	MOVL	@STCIOB+56, R0	
			85	13 0016A	BEQL	9\$	1110
	50		03	C0 0016C	ADDL2	#3, R0	
	50		04	C6 0016F	DIVL2	#4, R0	
69	50		04	A5 00172	MULW3	#4, R0, IOB\$+52	
	F8		02	90 00176	MOVB	#2, IOB\$+54	
	A9		06	90 0017A	MOVB	#6, IOB\$+44	
		00000000G	EF	9F 0017E	PUSHAB	XPOSFAILURE	
			7E	D4 00184	CLRL	-(SP)	
			CC	A9 9F 00186	PUSHAB	IOB\$	
00000000G	EF	03	FB	00189	CALLS	#3, XPOSGET	1070
			FED0	31 00190	BRW	4\$	1114
			50	D4 00193	CLRL	R0	
	56	00000000G	EF	D1 00195	CMPL	MINOR, STCN	
			02	12 0019C	BNEQ	18\$	
			50	D6 0019E	INCL	R0	
	51		52	D2 001A0	MCOML	R2, R1	18\$:
	50		51	CA 001A3	BICL2	R1, R0	
	01		50	D1 001A6	CMPL	R0, #1	
			03	13 001A9	BEQL	19\$	
			0583	31 001AB	BRW	49\$	
00000000G	EF	00000000G	EF	9E 001AE	MOVAB	LINE, LP	19\$:
		00000000G	EF	D4 001B9	CLRL	INTLIN	1123
		00000000G	EF	D4 001BF	CLRL	EXTLIN	
00000000G	EF	00000000G	EF	D0 001C5	MOVL	INDENTS, LINE INDENT	1124
00000000G	EF	00000000G	EF	D0 001D0	MOVL	LINE INDENT, EXTLIN	1125
00000000G	EF	00000000G	EF	C1 001DB	ADDL3	INDENTS+4, INDENTS, WRAP	1126
00000000G	EF	00000000G	EF	D0 001EB	MOVL	CMDBLK+24, RMARGIN	1127
00000000G	EF		02	D0 001F6	MOVL	#2, MAJOR	1128
	03		58	E8 001FD	BLBS	FIRST_RECORD, 21\$	1130
			029A	31 00200	BRW	34\$	
	03		56	D1 00203	CMPL	STCN, #3	20\$:
			F8	14 00206	BGTR	20\$	21\$:
	52		02	78 00208	ASHL	#2, STCN, R2	1147
03	00000000G	EF	01	E0 0020C	BBS	#1, CMDBLK, 22\$	1136
			00C9	31 00214	BRW	25\$	
		00000000G	EF	9F 00217	PUSHAB	STR\$FAILURE	22\$:
			7E	D4 0021D	CLRL	-(SP)	1143
			5A	DD 0021F	PUSHL	R10	
		00000000'	EF	9F 00221	PUSHAB	\$STR\$STRING	
			7E	D4 00227	CLRL	-(SP)	
00000000G	EF		05	FB 00229	CALLS	#5, XST\$COPY	
	50		6A	3C 00230	MOVZWL	TMPSTR, R0	
00000000G	EF		50	C0 00233	ADDL2	R0, CHROUT	
	6B		6A	9E 0023A	MOVAB	\$IOB\$OUTPUT, IOB\$+68	
E8	AB		07	90 0023D	MOVB	#7, IOB\$+44	
		00000000G	EF	9F 00241	PUSHAB	XPOSFAILURE	
			7E	D4 00247	CLRL	-(SP)	
			BC	AB 9F 00249	PUSHAB	IOB\$	
00000000G	EF		03	FB 0024C	CALLS	#3, XPOSPUT	
07	00000000G	EF	01	E1 00253	BBC	#1, CMDBLK, 23\$	

00000000G	EF	00	FB	0025B	CALLS	#0, SPLIT	
00000000G		EF	D5	00262	TSTL	HL N	1145
00000000G		4B	13	00268	BEQL	24\$	
00000000G		EF	9F	0026A	PUSHAB	STR\$FAILURE	
00000000'		7E	D4	00270	CLRL	-(SP)	
00000000'		5A	DD	00272	PUSHL	R10	
00000000'		EF	9F	00274	PUSHAB	\$STR\$STRING	
00000000G	EF	7E	D4	0027A	CLRL	-(SP)	
00000000G	50	05	FB	0027C	CALLS	#5, XST\$COPY	
00000000G	EF	6A	3C	00283	MOVZWL	TMPSTR, R0	
00000000G	6B	50	C0	00286	ADDL2	R0, CHROUT	
E8	AB	6A	9E	0028D	MOVAB	\$IOB\$OUTPUT, IOB\$+68	
00000000G		07	90	00290	MOVB	#7, IOB\$+44	
00000000G		EF	9F	00294	PUSHAB	XPOS\$FAILURE	
00000000G		7E	D4	0029A	CLRL	-(SP)	
00000000G		AB	9F	0029C	PUSHAB	IOB\$	
00000000G	EF	03	FB	0029F	CALLS	#3, XPOS\$PUT	
00000000G	EF	01	E1	002A6	BBC	#1, CMDBLK, 24\$	
00000000G	EF	00	FB	002AE	CALLS	#0, SPLIT	
00000000'		EF	9F	002B5	PUSHAB	\$STR\$STRING2	1147
00000000'	EF	42	9F	002BB	PUSHAB	TMS TITLES[R2]	
00000000'		9E	DD	002C2	PUSHL	@(SP)+	
00000000G	EF	03	FB	002CA	PUSHAB	\$STR\$STRINGO	
00000000G		EF	9F	002D1	CALLS	#3, XST\$JOIN	
00000000G		7E	D4	002D7	PUSHAB	STR\$FAILURE	
00000000G		0401	8F	002D9	CLRL	-(SP)	
00000000'		0180	31	002DD	PUSHR	#*M<R0,R10>	
00000000'	EF	42	9F	002E0	BRW	32\$	1154
00000000G	52	9E	DD	002E7	PUSHAB	RND TITLES[R2]	
00000000G		EF	9F	002EA	MOVL	@(SP)+, R2	
00000000'		7E	D4	002F0	PUSHAB	STR\$FAILURE	1156
00000000'		5A	DD	002F2	CLRL	-(SP)	
00000000'		EF	9F	002F4	PUSHL	R10	
00000000G	EF	05	FB	002FC	PUSHAB	\$STR\$STRING	
00000000G	50	6A	3C	00303	CLRL	-(SP)	
00000000G	EF	50	C0	00306	CALLS	#5, XST\$COPY	
00000000G	6B	6A	9E	0030D	MOVZWL	TMPSTR, R0	
E8	AB	07	90	00310	ADDL2	R0, CHROUT	
00000000G		EF	9F	00314	MOVAB	\$IOB\$OUTPUT, IOB\$+68	
00000000G		7E	D4	0031A	MOVB	#7, IOB\$+44	
00000000G		AB	9F	0031C	PUSHAB	XPOS\$FAILURE	
00000000G	EF	03	FB	0031F	CLRL	-(SP)	
00000000G	EF	01	E1	00326	PUSHAB	IOB\$	
00000000G	EF	00	FB	0032E	CALLS	#3, XPOS\$PUT	
00000000G		EF	D5	00335	BBC	#1, CMDBLK, 26\$	
00000000G		12	13	0033B	CALLS	#0, SPLIT	
00000000G		EF	9F	0033D	TSTL	HL N	1157
00000000'		7E	D4	00343	BEQL	27\$	
00000000'		5A	DD	00345	PUSHAB	STR\$FAILURE	
00000000'		EF	9F	00347	CLRL	-(SP)	
00000000G		10	11	0034D	PUSHL	R10	
00000000G		EF	9F	0034F	PUSHAB	\$STR\$STRING	
00000000G		7E	D4	00355	BRB	28\$	
00000000G		5A	DD	00357	PUSHAB	STR\$FAILURE	27\$:
00000000G		EF	9F	00359	CLRL	-(SP)	
00000000G		EF	9F	00359	PUSHL	R10	
00000000G		EF	9F	00359	PUSHAB	\$STR\$STRING	

00000000G	EF		7E	D4	0035F	28\$:	CLRL	-(SP)	
	50		05	FB	00361		CALLS	#5, XST\$COPY	
00000000G	EF		6A	3C	00368		MOVZWL	TMPSTR, R0	
E8	6B		50	C0	0036B		ADDL2	R0, CHROUT	
	AB		6A	9E	00372		MOVAB	\$IOB\$OUTPUT, IOB\$+68	
		00000000G	07	90	00375		MOVB	#7, IOB\$+44	
			EF	9F	00379		PUSHAB	XPO\$FAILURE	
			7E	D4	0037F		CLRL	-(SP)	
		BC	AB	9F	00381		PUSHAB	IOB\$	
07	EF		03	FB	00384		CALLS	#3, XPO\$PUT	
00000000G	EF		01	E1	0038B		BBC	#1, CMDBLK, 29\$	
00000000G	EF		00	FB	00393		CALLS	#0, SPLIT	
	50	00000000G	EF	D0	0039A	29\$:	MOVL	INDENTS, R0	1159
			66	13	003A1		BEQL	30\$	
			7E	D4	003A3		CLRL	-(SP)	1161
			50	DD	003A5		PUSHL	R0	
00000000G	7E	0903	8F	3C	003A7		MOVZWL	#2307, -(SP)	
	EF		03	FB	003AC		CALLS	#3, XST\$ASCII	
			50	DD	003B3		PUSHL	R0	
00000000G	EF	00000000'	EF	9F	003B5		PUSHAB	\$STR\$STRINGO	
			02	FB	003BB		CALLS	#2, XST\$JOIN	
		00000000G	EF	9F	003C2		PUSHAB	STR\$FAILURE	
			7E	D4	003C8		CLRL	-(SP)	
		0401	8F	BB	003CA		PUSHR	#*M<R0,R10>	
00000000G	EF		7E	D4	003CE		CLRL	-(SP)	
	50		05	FB	003D0		CALLS	#5, XST\$COPY	
00000000G	EF		6A	3C	003D7		MOVZWL	TMPSTR, R0	
E8	6B		50	C0	003DA		ADDL2	R0, CHROUT	
	AB		6A	9E	003E1		MOVAB	\$IOB\$OUTPUT, IOB\$+68	
		00000000G	07	90	003E4		MOVB	#7, IOB\$+44	
			EF	9F	003E8		PUSHAB	XPO\$FAILURE	
			7E	D4	003EE		CLRL	-(SP)	
		BC	AB	9F	003F0		PUSHAB	IOB\$	
07	EF		03	FB	003F3		CALLS	#3, XPO\$PUT	
00000000G	EF		01	E1	003FA		BBC	#1, CMDBLK, 30\$	
00000000G	EF		00	FB	00402		CALLS	#0, SPLIT	
		00000000G	EF	9F	00409	30\$:	PUSHAB	STR\$FAILURE	1163
			7E	D4	0040F		CLRL	-(SP)	
		0404	8F	BB	00411		PUSHR	#*M<R2,R10>	
00000000G	EF		7E	D4	00415		CLRL	-(SP)	
	50		05	FB	00417		CALLS	#5, XST\$COPY	
00000000G	EF		6A	3C	0041E		MOVZWL	TMPSTR, R0	
E8	6B		50	C0	00421		ADDL2	R0, CHROUT	
	AB		6A	9E	00428		MOVAB	\$IOB\$OUTPUT, IOB\$+68	
		00G00000G	07	90	0042B		MOVB	#7, IOB\$+44	
			EF	9F	0042F		PUSHAB	XPO\$FAILURE	
			7E	D4	00435		CLRL	-(SP)	
		BC	AB	9F	00437		PUSHAB	IOB\$	
07	EF		03	FB	0043A		CALLS	#3, XPO\$PUT	
00000000G	EF		01	E1	00441		BBC	#1, CMDBLK, 31\$	
00000000G	EF		00	FB	00449		CALLS	#0, SPLIT	
		00000000G	EF	9F	00450	31\$:	PUSHAB	STR\$FAILURE	1164
			7E	D4	00456		CLRL	-(SP)	
			5A	DD	00458		PUSHL	R10	
		00000000'	EF	9F	0045A		PUSHAB	\$STR\$STRING	
00000000G	EF		7E	D4	00460	32\$:	CLRL	-(SP)	
			05	FB	00462		CALLS	#5, XST\$COPY	

		50	6A	3C	00469	MOVZWL	TMPSTR, R0		
	00000000G	EF	50	C0	0046C	ADDL2	R0, CHROUT		
	E8	6B	6A	9E	00473	MOVAB	\$IOR%OUTPUT, IOB\$+68		
		AB	07	90	00476	MOVB	#7, IOB\$+44		
			00000000G	EF	9F	0047A	PUSHAB	XPOSFAILURE	
				7E	D4	00480	CLRL	-(SP)	
			BC	AB	9F	00482	PUSHAB	IOB\$	
	00000000G	EF	03	FB	00485	CALLS	#3, XPOSPUT		
07	00000000G	EF	01	E1	0048C	BBC	#1, CMDBLK, 33\$		
	00000000G	EF	00	FB	00494	CALLS	#0, SPLIT		
			58	D4	0049B	CLRL	FIRST RECORD		1167
	00000000G	EF	01	D0	0049D	MOVL	#1, HE N		1170
1A	00000000G	EF	01	E1	004A4	BBC	#1, CMDBLK, 35\$		1172
00000000G	FF	00000000	05	28	004AC	MOV3	#5, P.ABY, @LP		
			05	C0	004B8	ADDL2	#5, LP		
			05	C0	004BF	ADDL2	#5, INTLIN		
			04	B0	004C6	MOVW	#4, IOB\$+52		1177
	02	A9	02	90	004C9	MOVB	#2, IOB\$+54		
	F8	A9	06	90	004CD	MOVB	#6, IOB\$+44		
			00000000G	EF	9F	004D1	PUSHAB	XPOSFAILURE	
				7E	D4	004D7	CLRL	-(SP)	
			CC	A9	9F	004D9	PUSHAB	IOB\$	
	00000000G	EF	03	FB	004DC	CALLS	#3, XPOSGET		
	OC	AE	04	B9	D0	004E3	MOVL	@STCIOB+56, LEN	1178
50	OC	AE	03	C1	004E8	ADDL3	#3, LEN, R0		1179
		50	04	C6	004ED	DIVL2	#4, R0		
69		50	04	A5	004F0	MULW3	#4, R0, IOB\$+52		
	02	A9	02	90	004F4	MOVB	#2, IOB\$+54		
	F8	A9	06	90	004F8	MOVB	#6, IOB\$+44		
			00000000G	EF	9F	004FC	PUSHAB	XPOSFAILURE	
				7E	D4	00502	CLRL	-(SP)	
			CC	A9	9F	00504	PUSHAB	IOB\$	
	00000000G	EF	03	FB	00507	CALLS	#3, XPOSGET		
04	B9	OC	2C	3A	0050E	LOCC	#4, LEN, @STCIOB+56		1180
			02	12	00514	BNEQ	36\$		
	08	AE	51	D4	00516	CLRL	R1		
	03	00000000G	51	D0	00518	MOVL	R1, PTR		
			04	E0	0051C	BBS	#4, CMDBLK, 38\$		1182
			009A	31	00524	BRW	40\$		
			FB	13	00527	BEQL	37\$		1188
			04	A9	D0	00529	MOVL	STCIOB+56, SEC_P	1198
04	AE	08	6E	C3	0052D	SUBL3	SEC_P, PTR, SEC_L		1199
			5E	DD	00533	PUSHL	SP		1200
			08	AE	9F	00535	PUSHAB	SEC_L	
	00000000V	EF	02	FB	00538	CALLS	#2, STRIP		
			04	AE	D5	0053F	TSTL	SEC_L	1202
				7D	13	00542	BEQL	40\$	
			OF	93	00544	BITB	PAGEN, #15		1208
				47	13	0054B	BEQL	39\$	
	00000000G	EF	00000000G	EF	9E	0054D	MOVAB	TXTSCT, PTRSCT	1214
			00000000G	EF	9F	00558	PUSHAB	PTRSCT	1215
			00000000G	EF	9F	0055E	PUSHAB	PAGEN	
	00000000G	EF	02	FB	00564	CALLS	#2, PACSEC		
	00000000G	EF	50	D0	0056B	MOVL	R0, LENSCT		
			00000000G	EF	9F	00572	PUSHAB	TXTSCT	1216
			00000000G	EF	DD	00578	PUSHL	LENSCT	
	00000000G	EF	02	FB	0057E	CALLS	#2, FMTXT		

57

			00000000'	EF	9F	00585		PUSHAB	P, ABZ	1217
				01	DD	00588		PUSHL	#1	
	00000000G	EF		02	FB	0058D		CALLS	#2, FMTTXT	
				6E	DD	00594	39\$:	PUSHL	SEC_P	1220
			08	AE	DD	00596		PUSHL	SEC_L	
21	00000000G	EF		02	FB	00599		CALLS	#2, FMTTXT	
	00000000G	EF		01	E0	005A0		BBS	#1, CMDBLK, 41\$	1222
	00000000G	FF		20	90	005A8		MOVB	#32, @LP	1224
			00000000G	EF	D6	005AF		INCL	LP	
			00000000G	EF	D6	005B5		INCL	INTLIN	
			00000000G	EF	D6	005BB		INCL	EXTLIN	
16	00000000G	EF		01	E1	005C1	40\$:	BBC	#1, CMDBLK, 42\$	1232
	00000000G	FF		8F	90	005C9	41\$:	MOVB	#64, @LP	1234
			00000000G	EF	D6	005D1		INCL	LP	
			00000000G	EF	D6	005D7		INCL	INTLIN	
				3B	11	005DD		BRB	43\$	1232
57	00000000G	EF	00000000G	EF	C1	005DF	42\$:	ADDL3	INDENTS+4, INDENTS, R7	1236
	00000000G	EF		57	D1	005EB		CMPL	R7, EXTLIN	
				26	15	005F2		BLEQ	43\$	
			00000000G	EF	C2	005F4		SUBL2	EXTLIN, R7	
20		57	00000000G	00	2C	005FB		MOVCS	#0, (SP), #32, R7, @LP	
		6E		FF		00600				
			00000000G	57	C0	00605		ADDL2	R7, LP	
			00000000G	57	C0	0060C		ADDL2	R7, INTLIN	
			00000000G	57	C0	00613		ADDL2	R7, EXTLIN	
18	00000000G	EF	00000000G	EF	9E	0061A	43\$:	MOVAB	TXTPAG, PTRPAG	1241
	00000000G	EF		03	E0	00625		BBS	#3, CMDBLK, 44\$	1243
				0A	DD	0062D		PUSHL	#10	1249
			00000000G	EF	9F	0062F		PUSHAB	PTRPAG	
			00000000G	EF	3C	00635		MOVZWL	PAGEN+14, -(SP)	
	00000000G	7E	00000000G	03	FB	0063C		CALLS	#3, PACBAS	
		EF		13	11	00643		BRB	45\$	
			00000000G	EF	9F	00645	44\$:	PUSHAB	PTRPAG	1256
			00000000G	EF	9F	0064B		PUSHAB	PAGEN	
			00000000G	02	FB	00651		CALLS	#2, PACPAG	
			00000000G	50	D0	00658	45\$:	MOVL	R0, LENPAG	
50	00000000G	EF	00000000G	EF	C3	0065F		SUBL3	LENPAG, CMDBLK+24, R0	1259
	00000000G	EF	FF	A0	9E	0066B		MOVAB	-1(R0), RMARGIN	
	00000000G	EF	00000000G	EF	D0	00673		MOVL	EXTLIN, WRAP	1260
				08	AE	0067E		TSTL	PTR	1264
				07	12	00681		BNEQ	46\$	
	08	AE	04	A9	D0	00683		MOVL	STCIOB+56, PTR	1266
				0D	11	00688		BRB	47\$	
				AE	D6	0068A	46\$:	INCL	PTR	1273
50			08	AE	C3	0068D		SUBL3	PTR, STCIOB+56, R0	1274
	04	A9	08	50	C0	00693		ADDL2	R0, LEN	
	0C	AE		AE	9F	00697	47\$:	PUSHAB	PTR	1277
				AE	9F	0069A		PUSHAB	LEN	
			10	AE	9F	0069A		PUSHAB	LEN	
	00000000V	EF		02	FB	0069D		CALLS	#2, STRIP	
				AE	DD	006A4		PUSHL	PTR	1278
				AE	DD	006A7		PUSHL	LEN	
			00000000G	02	FB	006AA		CALLS	#2, FMTTXT	
	00000000G	EF		00	FB	006B1		CALLS	#0, INSREF	1282
				EF	B0	006B8		MOVW	INTLIN, \$STR\$STRING	1286
	10	AE	00000000G	0E	90	006C0		MOVB	#14, \$STR\$STRING+2	
	12	AE		01	90	006C4		MOVB	#1, \$STR\$STRING+3	
	13	AE		9E	006C8			MOVAB	LINE, \$STR\$STRING+4	
	14	AE	00000000G	EF	9E	006C8				

		00000000G	EF 9F 006D0	PUSHAB	STR\$FAILURE	
			7E D4 006D6	CLRL	-(SP)	
			5A DD 006D8	PUSHL	R10	
		1C	AE 9F 006DA	PUSHAB	\$STR\$STRING	
			7E D4 006DD	CLRL	-(SP)	
	00000000G	EF	05 FB 006DF	CALLS	#5, XST\$COPY	
		50	6A 3C 006E6	MOVZWL	TMPSTR, R0	
	00000000G	EF	50 C0 006E9	ADDL2	R0, CHR0UT	
		6B	6A 9E 006F0	MOVAB	\$IOB\$OUTPUT, IOB\$+68	
	E8	AB	07 90 006F3	MOVB	#7, IOB\$+44	
		00000000G	EF 9F 006F7	PUSHAB	XPO\$FAILURE	
			7E D4 006FD	CLRL	-(SP)	
		BC	AB 9F 006FF	PUSHAB	IOB\$	
	00000000G	EF	03 FB 00702	CALLS	#3, XPO\$PUT	
07	00000000G	EF	01 E1 00709	BBC	#1, CMDBLK, 48\$	
	00000000G	EF	00 FB 00711	CALLS	#0, SPLIT	
	00000000G	EF	00 FB 00711	MOVAB	LINE, LP	1287
		00000000G	EF D4 00723	CLRL	INTLIN	
		00000000G	EF D4 00729	CLRL	EXTLIN	
			6D 11 0072F	BRB	51\$	1279
	51		56 CE 00731	MNEGL	STCN, R1	1290
			50 D4 00734	CLRL	R0	
	51	00000000G	EF D1 00736	CMPL	MINOR, R1	
			02 12 0073D	BNEQ	50\$	
			50 D6 0073F	INCL	R0	
	53		52 D2 00741	MCOML	R2, R3	
	50		53 CA 00744	BICL2	R3, R0	
	01		50 D1 00747	CMPL	R0, #1	
			52 12 0074A	BNEQ	51\$	
	69		04 B0 0074C	MOVW	#4, IOB\$+52	1298
	02		02 90 0074F	MOVB	#2, IOB\$+54	
	F8	A9	06 90 00753	MOVB	#6, IOB\$+44	
		00000000G	EF 9F 00757	PUSHAB	XPO\$FAILURE	
			7E D4 0075D	CLRL	-(SP)	
		CC	A9 9F 0075F	PUSHAB	IOB\$	
	00000000G	EF	03 FB 00762	CALLS	#3, XPO\$GET	
		52	04 B9 D0 00769	MOVL	@STCIOB+56, LEN	1299
		50	03 A2 9E 0076D	MOVAB	3(R2), R0	1300
		50	04 C6 00771	DIVL2	#4, R0	
69		50	04 A5 00774	MULW3	#4, R0, IOB\$+52	
	02	A9	02 90 00778	MOVB	#2, IOB\$+54	
	F8	A9	06 90 0077C	MOVB	#6, IOB\$+44	
		00000000G	EF 9F 00780	PUSHAB	XPO\$FAILURE	
			7E D4 00786	CLRL	-(SP)	
		CC	A9 9F 00788	PUSHAB	IOB\$	
	00000000G	EF	03 FB 0078B	CALLS	#3, XPO\$GET	
		04	A9 DD 00792	PUSHL	STCIOB+56	1301
			52 DD 00795	PUSHL	LEN	
	00000000V	EF	02 FB 00797	CALLS	#2, INSRNT	
		F8C2	31 0079E	BRW	4\$	1059
	FA	A9	10 88 007A1	BISB2	#16, IOB\$+46	1308
	F8	A9	02 90 007A5	MOVB	#2, IOB\$+44	
		00000000G	EF 9F 007A9	PUSHAB	XPO\$FAILURE	
			7E D4 007AF	CLRL	-(SP)	
		CC	A9 9F 007B1	PUSHAB	IOB\$	
	00000000G	EF	03 FB 007B4	CALLS	#3, XPO\$CLOSE	
		F85C	31 007BB	BRW	1\$	1045

CAPTION
VO4-000

CAPTION - Generate figure, table and example ca
STCHDR - Process the .SEND TOC n captions temp

^{J 12}
15-Sep-1984 23:50:23
14-Sep-1984 13:05:37

VAX-11 Bliss-32 V4.0-742
[RUNOFF.SRC]CAPTION.BLI;1

Page 42
(3)

04 007BE

RET

: 1311

; Routine Size: 1983 bytes, Routine Base: \$CODE\$ + 0D70

CAP
VO4

.....

```

: 999      1312 1 %SBTTL 'STRIP - Strip leading and trailing whitespace from text'
: 1000     1313 1 ROUTINE strip (l_addr, p_addr) : NOVALUE =
: 1001     1314 1 ++
: 1002     1315 1
: 1003     1316 1 FUNCTIONAL DESCRIPTION:
: 1004     1317 1
: 1005     1318 1     This routine strips leading and trailing whitespace from the input string
: 1006     1319 1
: 1007     1320 1 FORMAL PARAMETERS:
: 1008     1321 1
: 1009     1322 1     l_addr - address of length of string
: 1010     1323 1     p_addr - address of CH$PTR to string
: 1011     1324 1
: 1012     1325 1 IMPLICIT INPUTS:
: 1013     1326 1
: 1014     1327 1     None
: 1015     1328 1
: 1016     1329 1 IMPLICIT OUTPUTS:
: 1017     1330 1
: 1018     1331 1     The length and string pointer are modified to reflect stripped whitespace
: 1019     1332 1
: 1020     1333 1 ROUTINE VALUE:
: 1021     1334 1 COMPLETION CODES:
: 1022     1335 1
: 1023     1336 1     None
: 1024     1337 1
: 1025     1338 1 SIDE EFFECTS:
: 1026     1339 1
: 1027     1340 1     None
: 1028     1341 1
: 1029     1342 1 --
: 1030     1343 1
: 1031     1344 2 BEGIN
: 1032     1345 2
: 1033     1346 2 LOCAL
: 1034     1347 2     len,
: 1035     1348 2     ptr;
: 1036     1349 2
: 1037     1350 2     len = ..l_addr;
: 1038     1351 2     ptr = ..p_addr;
: 1039     1352 2
: 1040     1353 2     DECR i FROM .len TO 1 DO
: 1041     1354 2         BEGIN
: 1042     1355 2             |
: 1043     1356 2             | Remove leading whitespace from section number
: 1044     1357 2             |
: 1045     1358 2             LOCAL
: 1046     1359 2                 ch;
: 1047     1360 2
: 1048     1361 2                 ch = CH$RCHAR (.ptr);
: 1049     1362 2
: 1050     1363 2                 IF (.ch EQL %C' ') OR (.ch EQL tab)
: 1051     1364 2                 THEN
: 1052     1365 2                     BEGIN
: 1053     1366 2                         len = .len - 1;
: 1054     1367 2                         ptr = CH$PLUS (.ptr, 1);
: 1055     1368 2                     END

```

```

1056 1369 ELSE
1057 1370 EXITLOOP;
1058 1371
1059 1372 END;
1060 1373
1061 1374 .p_addr = .ptr; ! Set pointer to beginning of text
1062 1375 ptr = CH$PLUS (.ptr, .len - 1); ! Point to last character
1063 1376
1064 1377 DECR i FROM .len TO 1 DO
1065 1378 BEGIN
1066 1379 ! Remove trailing blanks and tabs
1067 1380
1068 1381 LOCAL
1069 1382 ch;
1070 1383
1071 1384 ch = CH$RCHAR (.ptr);
1072 1385
1073 1386 IF (.ch EQL %C' ') OR (.ch EQL tab)
1074 1387 THEN
1075 1388 BEGIN
1076 1389 len = .len - 1;
1077 1390 ptr = CH$PLUS (.ptr, -1);
1078 1391 END
1079 1392 ELSE
1080 1393 EXITLOOP;
1081 1394
1082 1395 END;
1083 1396
1084 1397 WHILE .len GEQ 3 DO
1085 1398 BEGIN
1086 1399 ! Strip trailing emphasis
1087 1400
1088 1401 IF CH$RCHAR (CH$PLUS (.ptr, -2)) EQL rintes
1089 1402 THEN
1090 1403 BEGIN
1091 1404 len = .len - 3;
1092 1405 ptr = CH$PLUS (.ptr, -3);
1093 1406 END
1094 1407 ELSE
1095 1408 EXITLOOP;
1096 1409
1097 1410 END;
1098 1411
1099 1412 .l_addr = .len; ! Return length of string
1100 1413 END;
1101 1414
1102 1415

```

```

50 04 BC D0 00002 STRIP: .WORD Save R2,R3
52 08 BC D0 00006 MOVL @L_ADDR, LEN
51 01 A0 9E 0000A MOVAB 1(R0), I
BRB 3$

```

1313
1350
1351
1353

CAPTION
V04-000

CAPTION - Generate figure, table and example ca
STRIP - Strip leading and trailing whitespace f

M 12
15-Sep-1984 23:50:23
14-Sep-1984 13:05:37

VAX-11 Bliss-32 V4.0-742
[RUNOFF.SRC]CAPTION.BLI;1

Page 45
(4)

	53		62	9A	00010	1\$:	MOVZBL	(PTR), CH	1361
	20		53	D1	00013		CMPL	CH, #32	1363
			05	13	00016		BEQL	2\$	
	09		53	D1	00018		CMPL	CH, #9	
			07	12	0001B		BNEQ	4\$	
			50	D7	0001D	2\$:	DECL	LEN	1366
			52	D6	0001F		INCL	PTR	1367
08	EC		51	F5	00021	3\$:	SOBGTR	I, 1\$	1353
	BC		52	D0	00024	4\$:	MOVL	PTR, @P_ADDR	1374
	52	FF	A042	9E	00028		MOVAB	-1(LEN)[PTR], PTR	1375
	51	01	A0	9E	0002D		MOVAB	1(R0), I	1377
			11	11	00031		BRB	7\$	
	53		62	9A	00033	5\$:	MOVZBL	(PTR), CH	1385
	20		53	D1	00036		CMPL	CH, #32	1387
			05	13	00039		BEQL	6\$	
	09		53	D1	0003B		CMPL	CH, #9	
			07	12	0003E		BNEQ	8\$	
			50	D7	00040	6\$:	DECL	LEN	1390
			52	D7	00042		DECL	PTR	1391
	EC		51	F5	00044	7\$:	SOBGTR	I, 5\$	1377
	03		50	D1	00047	8\$:	CMPL	LEN, #3	1398
			0E	19	0004A		BLSS	9\$	
	1C	FE	A2	91	0004C		CMPB	-2(PTR), #28	1403
			08	12	00050		BNEQ	9\$	
	50		03	C2	00052		SUBL2	#3, LEN	1406
	52		03	C2	00055		SUBL2	#3, PTR	1407
			ED	11	00058		BRB	8\$	1403
04	BC		50	D0	0005A	9\$:	MOVL	LEN, @L_ADDR	1414
			04	0005E			RET		1415

; Routine Size: 95 bytes, Routine Base: \$CODE\$ + 152F

CAP
V04

RE
L
C

```

: 1104 1416 1 %SBTTL 'INSRNT - Insert text directly into .RNT file'
: 1105 1417 1 GLOBAL ROUTINE insrnt (len, ptr) : NOVALUE =
: 1106 1418 1 ++
: 1107 1419 1
: 1108 1420 1 FUNCTIONAL DESCRIPTION:
: 1109 1421 1
: 1110 1422 1 This routine inserts the given text directly into the .RNT file
: 1111 1423 1
: 1112 1424 1 FORMAL PARAMETERS:
: 1113 1425 1
: 1114 1426 1 len - length of text
: 1115 1427 1 ptr - CH$PTR to text
: 1116 1428 1
: 1117 1429 1 IMPLICIT INPUTS:
: 1118 1430 1
: 1119 1431 1 cmdblk - command line information block
: 1120 1432 1
: 1121 1433 1 IMPLICIT OUTPUTS:
: 1122 1434 1
: 1123 1435 1 None
: 1124 1436 1
: 1125 1437 1 ROUTINE VALUE:
: 1126 1438 1 COMPLETION CODES:
: 1127 1439 1
: 1128 1440 1 None
: 1129 1441 1
: 1130 1442 1 SIDE EFFECTS:
: 1131 1443 1
: 1132 1444 1 None
: 1133 1445 1
: 1134 1446 1 --
: 1135 1447 1
: 1136 1448 2 BEGIN
: 1137 1449 2
: 1138 1450 2 IF .len GTR 0
: 1139 1451 2 THEN
: 1140 1452 2 BEGIN
: 1141 1453 2 |
: 1142 1454 2 | There is something to process
: 1143 1455 2 |
: 1144 1456 2 | hl_n = -1;
: 1145 1457 2 | clr_line ();
: 1146 1458 2 | line_indent = .indents [0];
: 1147 1459 2 | wrap = .line_indent;
: 1148 1460 2 | extlin = .line_indent;
: 1149 1461 2 | rmargin = .cmdblk [contents$g_page_width];
: 1150 1462 2 |
: 1151 1463 2 | IF .cmdblk [contents$v_tms11] THEN write_char (%C'<');
: 1152 1464 2 |
: 1153 1465 2 | fmttxt (.len, .ptr);
: 1154 1466 2 |
: 1155 1467 2 | IF .cmdblk [contents$v_tms11] THEN write_char (%C'>');
: 1156 1468 2 |
: 1157 1469 2 | put ((.intlin, CH$PTR (line)));
: 1158 1470 2 | clr_line ();
: 1159 1471 2 | END;
: 1160 1472 2
    
```


			03FC	00000	.ENTRY	INSRNT, Save R2,R3,R4,R5,R6,R7,R8,R9	1417
	59	00000000G	EF	9E	00002	MOVAB	IOB\$+68, R9
	58	00000000G	EF	9E	00009	MOVAB	\$STR\$TARGET, R8
	57	00000000G	EF	9E	00010	MOVAB	LINE, INDENT, R7
	56	00000000G	EF	9E	00017	MOVAB	EXTLIN, R6
	55	00000000G	EF	9E	0001E	MOVAB	LINE, R5
	54	00000000G	EF	9E	00025	MOVAB	CMDBLK, R4
	53	00000000G	EF	9E	0002C	MOVAB	INTLIN, R3
	52	00000000G	EF	9E	00033	MOVAB	LP, R2
	5E		08	C2	0003A	SUBL2	#8, SP
		04	AC	D5	0003D	TSTL	LEN
			01	14	00040	BGTR	1\$
				04	00042	RET	
	00000000G	EF	01	CE	00043	MNEGL	#1, HL N
		62		65	9E	0004A	MOVAB
				63	D4	0004D	CLRL
				66	D4	0004F	CLRL
		67	20	A4	D0	00051	MOVL
	00000000G	EF		67	D0	00055	MOVL
		66		67	D0	0005C	MOVL
	00000000G	EF	18	A4	D0	0005F	MOVL
08		64		01	E1	00067	BBC
	00	B2		3C	90	0006B	MOVAB
				62	D6	0006F	INCL
				63	D6	00071	INCL
		7E	04	AC	7D	00073	MOVQ
	00000000G	EF		02	FB	00077	CALLS
08		64		01	E1	0007E	BBC
	00	B2		3E	90	00082	MOVAB
				62	D6	00086	INCL
				63	D6	00088	INCL
		6E		63	B0	0008A	MOVW
	02	AE		0E	90	0008D	MOVAB
	03	AE		01	90	00091	MOVAB
	04	AE		65	9E	00095	MOVAB
			00000000G	EF	9F	00099	PUSHAB
				7E	D4	0009F	CLRL
				58	DD	000A1	PUSHL
			0C	AE	9F	000A3	PUSHAB
				7E	D4	000A6	CLRL
	00000000G	EF		05	FB	000A8	CALLS
		50		68	3C	000AF	MOVZWL
	00000000G	EF		50	C0	000B2	ADDL2
		69		68	9E	000B9	MOVAB
	E8	A9		07	90	000BC	MOVAB
			00000000G	EF	9F	000C0	PUSHAB
				7E	D4	000C6	CLRL
			BC	A9	9F	000C8	PUSHAB
	00000000G	EF		03	FB	000CB	CALLS
07		64		01	E1	000D2	BBC
	00000000G	EF		00	FB	000D6	CALLS
							#0, SPLIT
							#1, CMDBLK, 2\$
							#60, @LP
							LP
							INTLIN
							LEN, -(SP)
							#2, FMTTXT
							#1, CMDBLK, 3\$
							#62, @LP
							LP
							INTLIN
							INTLIN, \$STR\$STRING
							#14, \$STR\$STRING+2
							#1, \$STR\$STRING+3
							LINE, \$STR\$STRING+4
							STR\$FAILURE
							-(SP)
							R8
							\$STR\$STRING
							-(SP)
							#5, XST\$COPY
							TMPSTR, R0
							R0, CHROUT
							\$IOB\$OUTPUT, IOB\$+68
							#7, IOB\$+44
							XPOS\$FAILURE
							-(SP)
							IOB\$
							#3, XPOS\$PUT
							#1, CMDBLK, 4\$

.....

CAPTION
V04-000

CAPTION - Generate figure, table and example ca 15-Sep-1984 23:50:23
INSRNT - Insert text directly into .RNT file 14-Sep-1984 13:05:37

VAX-11 Bliss-32 V4.0-742
[RUNOFF.SRC]CAPTION.BLI;1

Page 48
(5)

CE
VO

62	65	9E 000DD 4\$:	MOVAB	LINE, LP
	63	D4 000E0	CLRL	INTLIN
	66	D4 000E2	CLRL	EXTLIN
		04 000E4	RET	

: 1470
:
:
: 1473

: Routine Size: 229 bytes, Routine Base: \$CODE\$ + 158E

: 1162	1474	1		
: 1163	1475	1	END	! End of module
: 1164	1476	0	ELUDOM	

PSECT SUMMARY

Name	Bytes	Attributes
\$SPLITS	200	NOVEC, NOWRT, RD, NOEXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)
\$OWNS	216	NOVEC, WRT, RD, NOEXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)
\$CODE\$	5747	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]XPORT.L32;1	590	156	26	252	00:00.1

COMMAND QUALIFIERS

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

: Size: 5747 code + 416 data bytes
: Run Time: 02:43.9
: Elapsed Time: 04:53.0
: Lines/CPU Min: 540
: Lexemes/CPU-Min: 96060
: Memory Used: 901 pages
: Compilation Complete

