


```

000000    UU    UU    TTTTTTTTTT    HH    HH    DDDDDDDD    RRRRRRRR
000000    UU    UU    TTTTTTTTTT    HH    HH    DDDDDDDD    RRRRRRRR
00    00    UU    UU    TT          HH    HH    DD          DD    RR          RR
00    00    UU    UU    TT          HH    HH    DD          DD    RR          RR
00    00    UU    UU    TT          HH    HH    DD          DD    RR          RR
00    00    UU    UU    TT          HH    HH    DD          DD    RR          RR
00    00    UU    UU    TT          HH    HH    DD          DD    RR          RR
00    00    UU    UU    TT          HH    HH    DD          DD    RR          RR
00    00    UU    UU    TT          HH    HH    DD          DD    RR          RR
00    00    UU    UU    TT          HH    HH    DD          DD    RR          RR
00    00    UU    UU    TT          HH    HH    DD          DD    RR          RR
000000    UU    UU    TT          HH    HH    DD          DD    RR          RR
000000    UU    UU    TT          HH    HH    DD          DD    RR          RR
UUUUUUUUUU    TT          HH    HH    DDDDDDDD    RR          RR
UUUUUUUUUU    TT          HH    HH    DDDDDDDD    RR          RR

```

```

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
LLLLLLLLLLL IIIIII    SSSSSSSS
LLLLLLLLLLL IIIIII    SSSSSSSS

```

```

....
....
....
....

```



```
1 0001 0 %TITLE 'O/P module for processing HEADER: figure, table, example, level'  
2 0002 0 MODULE OUTHDR (IDENT = 'V04-000'  
3 P 0003 0 %BLISS32 [ , ADDRESSING_MODE ( EXTERNAL = LONG_RELATIVE,  
4 0004 0 NONEXTERNAL = LONG_RELATIVE)]  
5 0005 0 ) =  
6 0006 1 BEGIN  
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: DSR (Digital Standard RUNOFF) / DSRPLUS  
33 0033 1  
34 0034 1 ABSTRACT: General purpose processor for headers of various kinds  
35 0035 1  
36 0036 1 ENVIRONMENT: Transportable  
37 0037 1  
38 0038 1 AUTHOR: R.W.Friday CR.LATION DATE: March, 1982  
39 0039 1
```

```

: 41      0040 1 %SBTTL 'Revision History'
: 42      0041 1
: 43      0042 1   MODIFIED BY:
: 44      0043 1
: 45      0044 1       017   REM00017   Ray Marshall   21-Mar-1984 -- 4-Apr-1984
: 46      0045 1       Implemented the foreign language conditionals for compiling
: 47      0046 1       fixed output words.
: 48      0047 1
: 49      0048 1       016   KFA00016   Ken Alden     15-Nov-1983
: 50      0049 1       Reversed action performed in #14, so now the toctxt and tochl
: 51      0050 1       records are shipped out for and header ...
: 52      0051 1
: 53      0052 1       015   KFA00015   Ken Alden     01-Nov-1983
: 54      0053 1       For FLIP, loosened restriction imposed in #014 to provide
: 55      0054 1       support for .HE, .HF, .HT.
: 56      0055 1
: 57      0056 1       014   KAD00014   Keith Dawson  25-May-1983
: 58      0057 1       For FLIP, forbid writing header information to the TOC for
: 59      0058 1       .HE, .HF, .HT. This is a FLIP Level 1 requirement that can
: 60      0059 1       be loosened for Level 2 support.
: 61      0060 1
: 62      0061 1       013   KAD00013   Keith Dawson  19-May-1983
: 63      0062 1       Moved the location at which the SCA_HEADER bit, used by FLIP,
: 64      0063 1       is set. This bit was not serving its purpose -- communica-
: 65      0064 1       ting between this module and LOHORI -- because a new save
: 66      0065 1       and restore of the SCA intervened.
: 67      0066 1
: 68      0067 1       012   KFA00012   Ken Alden     16-Mar-1983
: 69      0068 1       PUSH/POP_SCA now visible to dsr.
: 70      0069 1
: 71      0070 1       011   KFA00011   Ken Alden     11-Mar-1983
: 72      0071 1       For DSRPLUS: hl emphasis is always passed to the BRN,
: 73      0072 1       regardless of whether it is autosubtitle or not.
: 74      0073 1
: 75      0074 1       010   KAD00010   Keith Dawson  07-Mar-1983
: 76      0075 1       Global edit of all modules. Updated module names, idents,
: 77      0076 1       copyright dates. Changed require files to BLISS library.
: 78      0077 1
: 79      0078 1   --

```

```

81 0079 1 %SBTTL 'Module Level Declarations'
82 0080 1
83 0081 1 TABLE OF CONTENTS:
84 0082 1
85 0083 1 FORWARD ROUTINE
86 0084 1 OUTHDR : NOVALUE, ! Generate numbered or unnumbered header.
87 0085 1 OUTCTR : NOVALUE; ! Generate counter for the header.
88 0086 1
89 0087 1 INCLUDE FILES:
90 0088 1
91 0089 1
92 0090 1 LIBRARY 'NXPORT:XPORT'; ! XPORT Library
93 0091 1 REQUIRE 'REQ:RNODEF'; ! RUNOFF variant definitions
94 0222 1
95 U 0223 1 %IF DSRPLUS %THEN
96 U 0224 1 LIBRARY 'REQ:OPLLIB'; ! DSRPLUS BLISS Library
97 0225 1 %ELSE
98 0226 1 LIBRARY 'REQ:DSRLIB'; ! DSR BLISS Library
99 0227 1 %FI
100 0228 1
101 0229 1
102 0230 1 MACROS:
103 0231 1
104 0232 1
105 0233 1 EQUATED SYMBOLS:
106 0234 1
107 0235 1
108 0236 1 EXTERNAL LITERAL
109 0237 1 RINTES : UNSIGNED (8);
110 0238 1
111 0239 1 EXTERNAL LITERAL
112 0240 1 S_FMRA; !Allocated length for footnote MRA.
113 0241 1
114 0242 1 OWN STORAGE:
115 0243 1
116 0244 1 OWN
117 0245 1 PP_SCA : $H_R_SCA_BLOCK; !Used in PUSH_SCA, POP_SCA macros (defined in SCA.REQ).
118 0246 1
119 0247 1 EXTERNAL REFERENCES:
120 0248 1
121 0249 1
122 0250 1 EXTERNAL
123 0251 1 FOOMRA : FIXED STRING,
124 0252 1 FOOTSF : VECTOR [TSF_SIZE],
125 0253 1 FS01 : FIXED STRING,
126 0254 1 ECC : $ECC_BLOCKVECTOR, ! Counters and display codes for Examples, Figures, Tables
127 0255 1 GCA : GCA_DEFINITION,
128 0256 1 HLC : HLC_DEFINITION,
129 0257 1 HLDSP : VECTOR [MAX_LEVELS],
130 0258 1 HLLIST : COUNTED LIST,
131 0259 1 IRA : FIXED STRING,
132 0260 1 MRA : REF FIXED STRING,
133 0261 1 NPAGEN : PAGE_DEFINITION,
134 0262 1 PAGEN : PAGE_DEFINITION,
135 0263 1 PHAN : PHAN_DEFINITION,
136 0264 1 SBTMRA : FIXED STRING,
137 0265 1 SBTTSF : VECTOR,

```

OUTHDR
V04-000

O/P module for processing HEADER: figure, table
Module Level Declarations

E 10
16-Sep-1984 01:20:29
14-Sep-1984 13:07:31

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

Page 4
(3)

OUT
V04

```
: 138      0266 1      SCA : SCA_DEFINITION,  
: 139      0267 1      TSF : TSF_DEFINITION;  
: 140      0268 1  
: 141      0269 1 EXTERNAL  
: 142      0270 1      KHAR;  
: 143      0271 1  
: 144      0272 1 EXTERNAL ROUTINE  
: 145      0273 1      endchr,      endwrđ,      gcskip,      gtpc,      outnj,  
: 146      0274 1      outcrg,      pacsec,      pacxxx,      puttpg,      putcnt,  
: 147      0275 1      puttxt,      scant,      setcas,      rskips,      titles;
```

```

149 0276 1 GLOBAL ROUTINE OUTHDR
150 0277 1 (
151 0278 1     lines_before,          lines_after,          test_page_amount,
152 0279 1     counter_major_type,   counter_minor_type,
153 0280 1     counter_value,        counter_display_code, counter_spaces_after,
154 0281 1     counter_pre_string_length, counter_pre_string_ptr,
155 0282 1     counter_post_string_length, counter_post_string_ptr,
156 0283 1     caption_major_type,   caption_minor_type,   caption_case,
157 0284 1     caption_is_centered,  caption_is_flush_right, caption_is_run_in,
158 0285 1     caption_is_bold,      caption_is_underlined, put_into_mem_file,
159 0286 1     autosubtitle,         brn_open,              break_before_caption,
160 0287 1     lines_between,        new_page,              startodd,
161 0288 1     tocpage
162 0289 1 )
163 0290 1 : NOVALUE =
164 0291 1 +
165 0292 1 : FUNCTIONAL DESCRIPTION:
166 0293 1
167 0294 1     See the ABSTRACT, above.
168 0295 1
169 0296 1 : FORMAL PARAMETERS:
170 0297 1
171 0298 1 : LINES_BEFORE          Leave this many blank lines before the header
172 0299 1 : LINES_AFTER          Leave this many blank lines after the (non-run-in) header
173 0300 1 : TEST_PAGE_AMOUNT     start a new page unless this many lines remain
174 0301 1 : COUNTER_MAJOR_TYPE   (not implemented) always MAJ_RUNOFF
175 0302 1 : COUNTER_MINOR_TYPE   specifies type of header: HL, Example, Figure, Table
176 0303 1 : COUNTER_VALUE        (not used for Hs) numerical value of the counter
177 0304 1 : COUNTER_DISPLAY_CODE (not used for Hs) display code of the counter
178 0305 1 : COUNTER_SPACES_AFTER leave this many spaces between counter and caption
179 0306 1 : COUNTER_PRE_STRING_LENGTH (not used for Hs) length of pre-counter string
180 0307 1 : COUNTER_PRE_STRING_PTR (not used for Hs) pointer to pre-counter string
181 0308 1 : COUNTER_POST_STRING_LENGTH (not used for Hs) length of post-counter string
182 0309 1 : COUNTER_POST_STRING_PTR (not used for Hs) pointer to post-counter string
183 0310 1 : CAPTION_MAJOR_TYPE   (not implemented) always MAJ_RUNOFF
184 0311 1 : CAPTION_MINOR_TYPE   specifies whether or not to show the counter
185 0312 1 : CAPTION_CASE         code for the case rules to apply to the caption
186 0313 1 : CAPTION_IS_CENTERED  True if caption should be centered
187 0314 1 : CAPTION_IS_FLUSH_RIGHT (not implemented) True if caption should be flush-right
188 0315 1 : CAPTION_IS_RUN_IN    True if caption should be run-in with following text
189 0316 1 : CAPTION_IS_BOLD      True if caption should be bolded
190 0317 1 : CAPTION_IS_UNDERLINED True if caption should be underlined
191 0318 1 : PUT INTO MEM_FILE    (not implemented) True if header should be output in .MEM file
192 0319 1 : AUTOSUBTITLE        True if header should be picked up as an auto subtitle
193 0320 1 : BRN_OPEN             True if header should be sent to the TOC
194 0321 1
195 0322 1
196 0323 1 : IMPLICIT INPUTS:      None
197 0324 1
198 0325 1 : IMPLICIT OUTPUTS:    None
199 0326 1
200 0327 1 : ROUTINE VALUE:
201 0328 1 : COMPLETION CODES:    None
202 0329 1
203 0330 1 : SIDE EFFECTS:        None
204 0331 1 :--
205 0332 1

```

```

206 0333 2 BEGIN
207 0334 2
208 0335 2 LOCAL
209 0336 2 GEN CHAR,
210 0337 2 HOLD_KHAR,
211 0338 2 HOLD_LST_SP,
212 0339 2 HOLD_MRA,
213 0340 2 HOLD_TSF,
214 0341 2 HOLD_LM,
215 0342 2 HOLD_WRD_PNTR,
216 0343 2 IRA_HOLD : VECTOR [10],
217 0344 2 SCA_HOLD : VECTOR [SCA_SIZE],
218 0345 2 SHOW_COUNTER;
219 0346 2
220 0347 2 !Assume header is numbered (counter will be shown).
221 0348 2 SHOW_COUNTER = TRUE;
222 0349 2
223 0350 2 !Skip spaces and tabs before the text.
224 0351 2 RSKIPS (IRA);
225 0352 2
226 0353 2 !See if there is enough room on the page.
227 0354 2 GTPC (.TEST_PAGE_AMOUNT);
228 0355 2
229 0356 2 !Now, force the paper to be positioned as it should.
230 0357 2 !The reason for forcing it, rather than allowing it to happen
231 0358 2 !the "normal" way, is that the user may have said something such
232 0359 2 !as ".NUMBER INDEX", which, by definition, does not take effect
233 0360 2 !in the middle of a page. By forcing the paper to be positioned,
234 0361 2 !.HEADER LEVELS that would be forced onto the next page
235 0362 2 !anyway can get influenced by commands such as just mentioned.
236 0363 2 OUTCRG ();
237 0364 2
238 0365 2 !Generate the specified number of lines before the header text.
239 0366 2 GCSKIP (.LINES_BEFORE);
240 0367 2
241 0368 2 !If at the top of any page, output the page number. This is being done
242 0369 2 !because of problems with the the design of RUNOFF. NEWPAG won't output
243 0370 2 !the page number until this header text forces a title to be generated,
244 0371 2 !which may be too late in this case.
245 0372 2 IF (.BRN_OPEN AND .PHAN_TOP_PAGE)
246 0373 2 THEN
247 0374 2 IF .PHAN_TOP_FIRST
248 0375 2 THEN
249 0376 2 XIF FLIP XTHEN
250 0377 2 PUTTPG (PAGEN, FLIP$K_TOCPAG)
251 0378 2 ELSE
252 0379 2 PUTTPG (NPAGEN, FLIP$K_TOCPAG);
253 0380 2 XELSE
254 0381 2 PUTTPG (PAGEN, -1)
255 0382 2 ELSE
256 0383 2 PUTTPG (NPAGEN, -1);
257 0384 2 XFI
258 0385 2 ! If creating a binary table of contents file, write out the counter and
259 0386 2 ! display descriptor.
260 0387 2 IF .BRN_OPEN
261 0388 2 THEN
262 0389 2 !User is generating a table of contents.

```



```

263 0390 2 PUTCNT (.COUNTER_MAJOR_TYPE,
264 0391 2 .COUNTER_MINOR_TYPE,
265 0392 2 .COUNTER_VALUE, ! The actual number.
266 0393 2 .COUNTER_DISPLAY_CODE, ! How to display the number.
267 0394 2 .COUNTER_PRE_STRING_LENGTH,
268 0395 2 .COUNTER_PRE_STRING_PTR, ! String before the number.
269 0396 2 .COUNTER_POST_STRING_LENGTH,
270 0397 2 .COUNTER_POST_STRING_PTR ! String after the number.
271 0398 2 ),
272 0399 2
273 0400 2 !Save scanning information
274 0401 2 PUSH_SCA: !Save the SAVED SCA bits.
275 0402 2 INCR I FROM 0 TO SCA_SIZE - 1 DO
276 0403 2 SCA_HOLD [I] = .SCA [I];
277 0404 2 !Save left margin; it might be used if centering and wrapping.
278 0405 2 HOLD_LM = .SCA_LM;
279 0406 2 !Save IRA and KHAR.
280 0407 2 INCR I FROM 0 TO 10 - 1 DO
281 0408 2 IRA_HOLD [I] = .IRA [I];
282 0409 2 HOLD_KHAR = .KHAR;
283 0410 2
284 0411 2 !+ If either (1) this head will become an autosubtitle or (2) user is creating a
285 0412 2 .BRN file, then we must scan the text twice. The first time we scan it at full
286 0413 2 width, 150 characters, for subtitle and/or TOC. If the subtitle actually runs
287 0414 2 wider than the current right margin, however, we truncate it and append "...".
288 0415 2
289 0416 2 IF .AUTOSUBTITLE OR .BRN_OPEN
290 0417 2 THEN
291 0418 2 BEGIN
292 0419 2
293 0420 2 SCA_PRESCAN = FALSE; !A ';' does NOT terminate this command.
294 0421 2 SCA_RSKIPS = TRUE; !Ignore multiple spaces and tabs.
295 0422 2 SCA_FILL = FALSE;
296 0423 2 SCA_JUSTIFY = FALSE;
297 0424 2 !Set right margin big to catch long headers.
298 0425 2 SCA_LM = 0;
299 0426 2 SCA_RM = 150;
300 0427 2 SCA_DO_IND = FALSE; !Ignore indexing for subtitle/TOC scan.
301 0428 2
302 0429 2 !Make a subtitle out of this header, if that's what the user desires.
303 0430 2 !The routine TITLES, which normally processes
304 0431 2 !titles/subtitles, contains the necessary logic.
305 0432 2 IF .AUTOSUBTITLE
306 0433 2 THEN
307 0434 2 !Collect header text in subtitle buffer
308 0435 2 BEGIN
309 0436 2 LOCAL
310 0437 2 HOLD_NBITS;
311 0438 2
312 0439 2 !Set up case rules for the header text.
313 0440 2 SETCAS (.CAPTION_CASE);
314 0441 2
315 0442 2 !Use TITLES to get the subtitle.
316 0443 2 TITLES (-1);
317 0444 2
318 0445 2 !Temporarily switch back to the subtitle TSF
319 0446 2 HOLD_TSF = .TSF;

```

```

320      0447 4      TSF = SBTTSF;
321      0448 4
322      0449 4      !Carry over underlining/bolding, etc, as specified.
323      0450 4      TSF_NBITS = .TSF_NBITS AND .HLC_CO_NBITS;
324      0451 4
325      0452 4      !Return to the actual TSF
326      0453 4      TSF = .HOLD_TSF;
327      0454 4
328      0455 3      END;                                     !End of autosubtitle processing.
329      0456 3
330      0457 3      IF .brn_open
331      0458 3      THEN
332      0459 4      BEGIN
333      0460 4      !** -----
334      0461 4      !** Is this whole trip necessary? Looks like it does nothing.
335      0462 4      !Restore IRA and KHAR.
336      0463 4      INCR I FROM 0 TO 10 - 1 DO
337      0464 4      IRA [I] = .IRA_HOLD [I];
338      0465 4
339      0466 4      INCR I FROM 0 TO SCA_SIZE - 1 DO
340      0467 4      SCA [I] = .SCA_HOLD [I];
341      0468 4
342      0469 4      POP_SCA;      !Restore the SAVED SCA bits.
343      0470 4      !Save scanning information for another scan.
344      0471 4      PUSH_SCA;      !Save the SAVED SCA bits.
345      0472 4
346      0473 4      INCR I FROM 0 TO SCA_SIZE - 1 DO
347      0474 4      SCA_HOLD [I] = .SCA [I];
348      0475 4      !Save IRA and KHAR.
349      0476 4      INCR I FROM 0 TO 10 - 1 DO
350      0477 4      IRA_HOLD [I] = .IRA [I];
351      0478 4      !** -----
352      0479 4      !Going to TOC, may or may not be an auto-subtitle.
353      0480 4
354      0481 4      sca_prescan = false;      !A ';' does NOT terminate this command.
355      0482 4      sca_rskips = true;      !Ignore multiple spaces and tabs.
356      0483 4      sca_fill = false;
357      0484 4      sca_justify = false;
358      0485 4      sca_lm = 0;      !Set right margin big to catch long headers.
359      0486 4      sca_rm = 150;
360      0487 4      sca_do_ind = false;      !Ignore indexing for subtitle/TOC scan.
361      0488 4      sca_fc_case = true;
362      0489 4      sca_fc = true;
363      0490 4      khar = .hold_khar;
364      0491 4
365      0492 4      !Switch to the Footnote TSF and MRA (which are not otherwise used in headers) to
366      0493 4      !collect the header information for the TOC.
367      0494 4
368      0495 4      hold_mra = .mra;
369      0496 4      mra = foomra;
370      0497 4      hold_tsf = .tsf;
371      0498 4      tsf = footsf;
372      0499 4
373      0500 4      !Before initializing the text descriptors, it is necessary to explicitly
374      0501 4      !reset the maximum length of the footnote MRA. It is clobbered if any
375      0502 4      !footnotes have been processed so far. See the comments in FNONLY around
376      0503 4      !line 1545 for the detailed justification.

```

: R

```

377 0504 4 fs_maxsize (mra) = s_fmra;
378 0505 4
379 0506 4 !Now initialize the text descriptors.
380 0507 4 fs_init (mra);
381 0508 4 INCR i FROM 0 TO tsf_size - 1 DO
382 0509 4 tsf [.i] = 0;
383 0510 4
384 0511 4 !Set up bolding and underlining, if requested.
385 0512 5 IF (.caption_is_bold AND .sca_do_bld)
386 0513 4 THEN
387 0514 5 BEGIN
388 0515 5 sca_bld = true;
389 0516 5 sca_wrd_c_bld = true;
390 0517 4 END;
391 0518 4
392 0519 5 IF (.caption_is_underlined AND .sca_do_und)
393 0520 4 THEN
394 0521 5 BEGIN
395 0522 5 sca_und = true;
396 0523 5 sca_wrd_c_und = true;
397 0524 4 END;
398 0525 4
399 0526 4 ! Set up case rules for the header.
400 0527 4 setcas (.caption_case);
401 0528 4
402 0529 4 ! Scan 150-wide into footnote MRA.
403 0530 4 scant ();
404 0531 4 endwrd (false, false, false);
405 0532 4
406 0533 4 ! Write header text to .BRN file.
407 0534 4
408 0535 4 !** Needed for FLIP Level 1 support. The result is that PUTTXT is not called
409 0536 4 for .HE, .HF, .HT commands if /DEVICE=FLIP was specified.
410 0537 4
411 0538 4 !:XIF FLIP XTHEN
412 0539 4 IF (.gca_op_dev NEQ op_dev_flip)
413 0540 4 OR
414 0541 4 (.caption_minor_type LEQ min_hl_txt_u)
415 0542 4 THEN
416 0543 4 !:XFI
417 0544 4 !**
418 0545 4 puttxt (.fs_length(foomra),
419 0546 4 .fs_start(foomra),
420 0547 4 .caption_major_type,
421 0548 4 .caption_minor_type);
422 0549 4
423 0550 4 !Switch TSF and MRA back.
424 0551 4 mra = .hold_mra;
425 0552 4 tsf = .hold_tsf;
426 0553 4 END; !End of TOC-only processing.
427 0554 4
428 0555 4 !Restore IRA and KHAR.
429 0556 4 INCR i FROM 0 TO 10 - 1 DO
430 0557 4 ira [.i] = .ira_hold [.i];
431 0558 4 khar = .hold_khar;
432 0559 4
433 0560 2 END; !End of subtitle-or-TOC processing

```

```

434 0561 2
435 0562 2 INCR I FROM 0 TO SCA_SIZE - 1 DO
436 0563 2     SCA [I] = .SCA_HOLD [I];
437 0564 2 POP_SCA; !Restore the SAVED SCA bits.
438 0565 2 SCA_FILL = TRUE; !Fill the header even if not filling text.
439 0566 2 SCA_PRESCAN = FALSE; !A ';' does NOT terminate this command.
440 0567 2 SCA_RSKIPS = TRUE; !Ignore multiple spaces and tabs.
441 0568 2
442 0569 2
443 0570 2 !Set up bolding and underlining, if requested.
444 0571 2 IF (.caption_is_bold AND .sca_do_bld)
445 0572 2 THEN
446 0573 2     BEGIN
447 0574 2     sca_bld = true;
448 0575 2     sca_wrd_c_bld = true;
449 0576 2     END;
450 0577 2
451 0578 2 IF (.caption_is_underlined AND .sca_do_und)
452 0579 2 THEN
453 0580 2     BEGIN
454 0581 2     sca_und = true;
455 0582 2     sca_wrd_c_und = true;
456 0583 2     END;
457 0584 2
458 0585 2 ! Generate the number for the header title if not turned off. Note:
459 0586 2 ! the following line depends on the values of the literals in TOCRTY.
460 0587 2
461 0588 2 show_counter = NOT ((.caption_minor_type MOD 3) EQL 0);
462 0589 2 ! If centering the header, pull in the margins so that the caption
463 0590 2 ! looks more deliberately centered.
464 0591 2
465 0592 2 IF .caption_is_centered
466 0593 2 THEN
467 0594 2     BEGIN
468 0595 2     sca_lm = .sca_lm + 3;
469 0596 2     sca_rm = .sca_rm - 3;
470 0597 2     END;
471 0598 2
472 0599 2 IF .show_counter
473 0600 2 THEN
474 0601 2     ! User didn't turn off the numbering, so generate the header counter.
475 0602 2     BEGIN
476 0603 2     ! Generate the counter in using the case rules for the caption.
477 0604 2     !
478 0605 2     outctr (.counter_minor_type, .caption_case);
479 0606 2     !
480 0607 2     ! Put the counter into the output buffer.
481 0608 2     !
482 0609 2     fs_next (fs01) = .fs_start (fs01);
483 0610 2     INCR i FROM 1 TO .fs_length (fs01) DO
484 0611 2         BEGIN
485 0612 2         LOCAL
486 0613 2         temp_char;
487 0614 2         fs_rchar (fs01, temp_char);
488 0615 2         endchr (.temp_char);
489 0616 2         END;
490 0617 2

```

```

491 0618 3      ! End the string so the spaces_after will not get under inded (if
492 0619 3      ! in effect)
493 0620 3      endwrđ (false, false, false);
494 0621 3
495 0622 3      ! Insert spaces after the counter if user didn't say BREAK or BETWEEN.
496 0623 3      ! Do not put out more than 75 spaces regardless of what the user said.
497 0624 3
498 0625 4      IF NOT (.break_before_caption OR (.lines_between GTR 0))
499 0626 3      THEN
500 0627 4      BEGIN
501 0628 4      LOCAL
502 0629 4      sca_hold_c_bldun,
503 0630 4      sca_hold_ac_bln;
504 0631 4      ! ENDCHR carries over underlining rules even for spaces.
505 0632 4      ! We must turn off these bits before we add the extra spaces.
506 0633 4
507 0634 4      sca_hold_c_bldun = .sca_wrd_c_bldun;
508 0635 4      sca_hold_ac_bln = .sca_wrd_ac_bln;
509 0636 4      INCR i FROM 1 TO (MIN (.counter_spaces_after, 75)) DO
510 0637 5      BEGIN
511 0638 5      sca_wrd_c_bldun = 0;
512 0639 5      sca_wrd_ac_bln = 0;
513 0640 5      endchr (' ');
514 0641 4      END;
515 0642 4
516 0643 4      ! Restore the SCA bits.
517 0644 4      sca_wrd_c_bldun = .sca_hold_c_bldun;
518 0645 4      sca_wrd_ac_bln = .sca_hold_ac_bln;
519 0646 4      END;
520 0647 3
521 0648 3      ! And now end this 'word'.
522 0649 3
523 0650 3      endwrđ (false,false,false);
524 0651 3
525 0652 3      ! If the user wants a break before the caption, we must
526 0653 3      ! make any adjustments if not flush_left.
527 0654 3
528 0655 3      IF .caption_is_centered OR .caption_is_flush_right
529 0656 3      THEN
530 0657 4      BEGIN
531 0658 5      IF .break_before_caption OR (.lines_between GTR 0)
532 0659 4      THEN
533 0660 5      BEGIN
534 0661 5      tsf_adjust =
535 0662 6      (IF .caption_is_centered      !Center the counter.
536 0663 6      THEN (.sca_rm - (.tsf_ext_hl))/2
537 0664 6      ELSE      !Force the counter right.
538 0665 5      .sca_rm - .tsf_ext_hl);
539 0666 5      END
540 0667 4      ELSE
541 0668 5      BEGIN
542 0669 5      ! No break to be done so we must set the left margin so
543 0670 5      ! wrapping will left justify on the caption left margin.
544 0671 5      hold_lm = .tsf_ext_hl; !Remember for use after SCA is restored.
545 0672 5      sca_lm = .tsf_ext_hl;
546 0673 4      END;
547 0674 4      END

```

```

548 0675 4
549 0676 4
550 0677 4
551 0678 4
552 0679 4
553 0680 4
554 0681 4
555 0682 4
556 0683 4
557 0684 4
558 0685 4
559 0686 4
560 0687 4
561 0688 4
562 0689 4
563 0690 4
564 0691 4
565 0692 4
566 0693 4
567 0694 4
568 0695 4
569 0696 4
570 0697 4
571 0698 4
572 0699 4
573 0700 4
574 0701 4
575 0702 4
576 0703 4
577 0704 4
578 0705 4
579 0706 4
580 0707 4
581 0708 4
582 0709 4
583 0710 4
584 0711 4
585 0712 4
586 0713 4
587 0714 4
588 0715 4
589 0716 4
590 0717 4
591 0718 4
592 0719 4
593 0720 4
594 0721 4
595 0722 4
596 0723 4
597 0724 4
598 0725 4
599 0726 4
600 0727 4
601 0728 4
602 0729 4
603 0730 4
604 0731 4

```

```

+
The header is flush-left. If the caption is to immediately follow, set the
left margin so that following text will wrap properly.
-
ELSE
    If we aren't doing a break after the counter, we must pull in
    the left margin to prevent writing over the counter.
    IF NOT (.break_before_caption OR (.lines_between GTR 0))
    THEN
        Set left margin beyond the counter so if the caption wraps,
        it will be nicely justified with itself.
        sca_lm = .tsf_ext_hl;
    END
    !End of code to generate header counter.
ELSE
    If we are not generating a number, then the first character
    scanned should be the first character in this MRA.
    sca_fc = true;
+
If the header is other than flush-left and user asked for break or
some lines between, then we have already computed amount to shift it;
we now put it out on a line by itself.
-
gca_line_pend = 1; !Until we throw the header, a line is pending.
IF (.break_before_caption) OR (.lines_between GTR 0)
THEN
    BEGIN
        outnj (); ! Put out the counter.
        gcskip (.lines_between); ! Skip the number of lines requested.
    END;
    ! Remember number of characters generated for use later.
    gen_char = .tsf_int_hl;
    ! Set up case rules for the heading.
    setcas (.caption_case);
    ! Tell SCANT that the first character of the header is the first
    ! character of a word. (ENDCHR turned this stuff off before.)
    sca_fc_case = true;
    scant (); !Go get the caption.
    ! SCA_WRD CPEND equals rintes IFF there was a trailing space/tab
    ! after the text. In such a case ENDWRD has already been called.
    ! Calling it again would have the effect of forcing an additional
    ! space out into TSF/MRA.
    IF .sca_wrd_cpend NEQ rintes
    THEN
        endwrđ (false, false, false)
    ELSE
        IF .sca_wrd_lst_und EQL 0

```

```

605 0732 2      AND                ! This fixes null-text-on-
606 0733 2      .sca_wrd_lst_sp GTR 0 ! run-in-header bug.
607 0734 2      THEN
608 0735 2      ! Chop off trailing spaces/tabs. When doing so, also back up
609 0736 2      ! intra-line pointer and counter appropriately. If justification
610 0737 2      ! was in effect, also cancel the justification mark that got writ-
611 0738 2      ! ten onto the MRA.
612 0739 2      ! Note that trailing spaces that are underlined are not discarded.
613 0740 2      BEGIN
614 0741 2      LOCAL
615 0742 2      chars_to_drop;
616 0743 3      chars_to_drop = .sca_wrd_lst_sp +
617 0744 4      (-IF .sca_justify
618 0745 4      THEN 3
619 0746 4      ELSE 0
620 0747 3      );
621 0748 3      fs_length (mra) = .fs_length (mra) - .chars_to_drop;
622 0749 3      fs_next (mra) = ch$plus (.fs_next (mra), -.chars_to_drop);
623 0750 3      sca_wrd_lst_sp = 0;
624 0751 2      END;
625 0752 2
626 0753 2      ! Restore previous scanner rules and set standard rules. Usually, it
627 0754 2      ! suffices to restore the previous scanner rules, and that's all. That
628 0755 2      ! works because after ENDWRD is called there is nothing left in the SCA
629 0756 2      ! that needs to be remembered. Simply restoring the SCA is then
630 0757 2      ! equivalent to a type of block structure. However, there is one case
631 0758 2      ! when the pointer to the NEXT word needs to be carried over; that is
632 0759 2      ! when the text of the header is too long for the current line,
633 0760 2      ! and it's a runin header. In that case, if the pointer is not carried
634 0761 2      ! over, generating the '-' after the text will not work, and will
635 0762 2      ! actually cause an internal logic error (RNFILE) when ENDWRD eventually
636 0763 2      ! gets called by some other routine, later.
637 0764 2      hold_wrd_pntr = .sca_wrd_pntr; !Remember start of next word.
638 0765 2      hold_lst_sp = .sca_wrd_lst_sp; !Remember last-space info.
639 0766 2      INCR i FROM 0 TO sca_size - 1 DO
640 0767 2      sca [i] = .sca_hold [i];
641 0768 2      pop_sca; !Restore the SAVED SCA bits again.
642 0769 2      sca_wrd_pntr = .hold_wrd_pntr; !Restore start of next word.
643 0770 2      sca_wrd_lst_sp = .hold_lst_sp; !Restore last-space info.
644 0771 2
645 0772 2      ! For FLIP, tell LOHORI to generate a TOCHL record (for .HL's only
646 0773 2      ! -- not for .HE, .HF, .HT commands).
647 0774 2      ! yes for now.
648 0775 2      %IF FLIP %THEN
649 0776 2      IF .brn_open
650 0777 2      !! AND
651 0778 2      !! (.caption_minor_type LEQ min_hl_txt_u)
652 0779 2      THEN
653 0780 2      sca_header = true;
654 0781 2      %FI
655 0782 2
656 0783 2      ! Differentiate between headers that are on a single line, and
657 0784 2      ! those for which a '-' has to be generated with following text
658 0785 2      ! running on.
659 0786 2      IF .caption_is_run_in
660 0787 2      THEN
661 0788 3      BEGIN

```

U
U
U
U
U

%IF
!!
!!
%FI

```

662 0789 3      IF .gen_char NEQ .tsf_int_hl
663 0790 3      THEN
664 0791 3      ! There was some text after the counter, so generate the '- '.
665 0792 4      BEGIN
666 0793 4      endchr (%C' ');
667 0794 4      endchr (%C'-');
668 0795 4      endwrd (true, false, false);
669 0796 4      END
670 0797 3      END
671 0798 2      ELSE
672 0799 2      ! Not a 'run-on' header.
673 0800 3      BEGIN
674 0801 3      ! This section basically outputs the last line of the header. This
675 0802 3      ! may also be the first (if only one line long). What must happen here
676 0803 3      ! is if the header is flush-left, the last line (if different from the
677 0804 3      ! first, must start at the caption left margin. The same goes for cen-
678 0805 3      ! tered captions if nobreak is in effect and if the caption is so long
679 0806 3      ! that it wraps. Now if it doesn't wrap, then this section will output
680 0807 3      ! the whole counter and caption.
681 0808 3      ! By using GCA_LINE_PEND, we may determine if this is the first line output.
682 0809 3      ! If a line has already gone out, then SCANT has already adjusted the left
683 0810 3      ! margin and no TSF_adjust is needed.
684 0811 3
685 0812 3      ! If the header is other than flush-left, compute amount to shift it.
686 0813 3      ! If user asked for a BREAK or had some LINES_BETWEEN, then the shift
687 0814 3      ! for the counter has already been taken care of and this section
688 0815 3      ! is only for the caption, otherwise the shift is for the whole header.
689 0816 3      IF .caption_is_centered OR .caption_is_flush_right
690 0817 3      THEN
691 0818 3      IF .gca_line_pend !NO line thrown yet.
692 0819 3      THEN
693 0820 3      tsf_adjust =
694 0821 4      (IF .caption_is_centered !Center the counter/and/or caption.
695 0822 5      THEN ((.sca_rm - .tsf_ext_hl)/2 - 2)
696 0823 4      ELSE !Force the counter/caption right.
697 0824 4      .sca_rm - .tsf_ext_hl)
698 0825 3      ELSE
699 0826 4      IF (.break_before_caption) OR (.lines_between GTR 0)
700 0827 3      THEN
701 0828 3      tsf_adjust =
702 0829 4      (IF .caption_is_centered !Center the counter/and/or caption.
703 0830 5      THEN ((.sca_rm - .tsf_ext_hl)/2 - 2) !This is centered.
704 0831 4      ELSE !Force the counter/caption right.
705 0832 3      .sca_rm - .tsf_ext_hl);
706 0833 3
707 0834 3      outnj (); !force out the caption.
708 0835 3
709 0836 3      gcskip (.lines_after);
710 0837 3
711 0838 3      sca_sect_empty = true; !This section is empty now.
712 0839 2      END;
713 0840 2
714 0841 2      !Set proper case conversion for remainder of section.
715 0842 2      setcas (.gca_case);
716 0843 2
717 0844 1      END; !End of OUTHDR

```



```

;
        .TITLE  OUTHDR O/P module for processing HEADER: figure
        .IDENT  \V04-000\
        .PSECT  $OWNS$,NOEXE,2
                00000 PP_SCA: .BLKB 48
        .EXTRN  RINTES, S FMRA, FOOMRA
        .EXTRN  FOOTSF, FS01, ECC
        .EXTRN  GCA, HLC, HLDSP
        .EXTRN  HLLIST, IRA, MRA
        .EXTRN  NPAGEN, PAGEN, PHAN
        .EXTRN  SBTMRA, SBTTSF, SCA
        .EXTRN  TSF, KHAR, ENDCHR
        .EXTRN  ENDWRD, GCSKIP, GTPC
        .EXTRN  OUTNJ, OUTCRG, PACSEC
        .EXTRN  PACXXX, PUTTPG, PUTCNT
        .EXTRN  PUTTXT, SCANT, SETCAS
        .EXTRN  RSKIPS, TITLES
        .PSECT  $CODE$,NOWRT,2
                OFFC 00000
        .ENTRY  OUTHDR, Save R2,R3,R4,R5,R6,R7,R8,R9,R10,-
                R11
                MOVAB  TSF, R11
                MOVAB  PP_SCA, R10
                MOVAB  SCA+120, R9
                MOVAB  -424(SP), SP
                MOVL   #1, SHOW_COUNTER
                PUSHAB IRA
                CALLS  #1, RSKIPS
                PUSHL  TEST_PAGE_AMOUNT
                CALLS  #1, GTPC
                CALLS  #0, OUTCRG
                PUSHL  LINES_BEFORE
                CALLS  #1, GCSKIP
                BLBC   BRN_OPEN, 4$
                BLBC   PHAN, 3$
                BLBC   PHAN+24, 1$
                MNEGL  #1, -(SP)
                PUSHAB PAGEN
                BRB    2$
                MNEGL  #1, -(SP)
                PUSHAB NPAGEN
                CALLS  #2, PUTTPG
                BLBC   BRN_OPEN, 4$
                MOVQ   COUNTER_POST_STRING_LENGTH, -(SP)
                MOVQ   COUNTER_PRE_STRING_LENGTH, -(SP)
                MOVQ   COUNTER_VALUE, -(SP)
                MOVQ   COUNTER_MAJOR_TYPE, -(SP)
                CALLS  #8, PUTCNT
                MOVL   @SCA+100, PP_SCA
                MOVL   @SCA+104, PP_SCA+4
                MOVL   @SCA+108, PP_SCA+8
                MOVL   @SCA+112, PP_SCA+12

```

1
1
1
1
1
1
1
1
1
1

0276
0348
0351
0354
0363
0366
0372
0374
0381
0383
0387
0396
0394
0392
0390
0398

10	AA	FC	B9	D0	000A2	MOVL	@SCA+116, PP_SCA+16				
14	AA	00	B9	D0	000A7	MOVL	@SCA+120, PP_SCA+20				
18	AA	04	B9	D0	000AC	MOVL	@SCA+124, PP_SCA+24				
1C	AA	08	B9	D0	000B1	MOVL	@SCA+128, PP_SCA+28				
20	AA	0C	B9	D0	000B6	MOVL	@SCA+132, PP_SCA+32				
24	AA	10	B9	D0	000BB	MOVL	@SCA+136, PP_SCA+36				
28	AA	14	B9	D0	000C0	MOVL	@SCA+140, PP_SCA+40				
2C	AA	18	B9	D0	000C5	MOVL	@SCA+144, PP_SCA+44				
				50	D4	000CA	CLRL	I	0402		
F2	6E40	88	A940	D0	000CC	5\$:	MOVL	SCA[I], SCA_HOLD[I]	0403		
	50	0000005F	8F	F3	00CD2		AOBLEQ	#95, I, 5\$			
	58		FC	B9	D0	000DA	MOVL	@SCA+116, HOLD_LM	0405		
				50	D4	000DE	CLRL	I	0407		
F2	DB	AD40	00000000GEF40	D0	000E0	6\$:	MOVL	IRA-4[I], IRA_HOLD[I]	0408		
	50			09	F3	000EA	AOBLEQ	#9, I, 6\$			
	55	00000000G	EF	D0	000EE		MOVL	KHAR, HOLD KHAR	0409		
	07		58	AC	E8	000F5	BLBS	AUTOSUBTITLE, 7\$	0416		
	03		5C	AC	E8	000F9	BLBS	BRN_OPEN, 7\$			
				01E6	31	000FD	BRW	19\$			
				34	A9	D4	00100	7\$:	CLRL	SCA+172	0420
54	A9			01	D0	00103	MOVL	#1, SCA+204	0421		
				F0	B9	D4	00107	CLRL	@SCA+104	0422	
				EC	B9	D4	0010A	CLRL	@SCA+100	0423	
				FC	B9	D4	0010D	CLRL	@SCA+116	0425	
00	B9			96	8F	9A	00110	MOVZBL	#150, @SCA+120	0426	
30	A9				08	8A	00115	BICB2	#8, SCA+168	0427	
	2F			58	AC	E9	00119	BLBC	AUTOSUBTITLE, 8\$	0432	
				3C	AC	DD	0011D	PUSHL	CAPTION CASE	0440	
00000000G	EF			01	FB	00120	CALLS	#1, SETCAS			
	7E			01	CE	00127	MNEGL	#1, -(SP)	0443		
C0000000G	EF			01	FB	0012A	CALLS	#1, TITLES			
	53			6B	D0	00131	MOVL	TSF, HOLD TSF	0446		
	6B	00000000G	EF	9E	00134		MOVAB	SBTTSF, TSF	0447		
	50			6B	D0	0013B	MOVL	TSF, R0			
	51	00000000G	EF	D2	0013E		MCOML	HLC+12, R1	0450		
08	A0			51	CA	00145	BICL2	R1, 8(R0)			
	6B			53	D0	00149	MOVL	HOLD TSF, TSF	0453		
	03			5C	AC	E8	0014C	8\$:	BLBS	BRN_OPEN, 9\$	0457
				017C	31	00150	BRW	17\$			
				50	D4	00153	9\$:	CLRL	I	0463	
F2	00000000GEF40		DB	AD40	D0	00155	10\$:	MOVL	IRA_HOLD[I], IRA-4[I]	0464	
	50			09	F3	0015F		AOBLEQ	#9, I, 10\$		
				50	D4	00163		CLRL	I	0466	
F2	88	A940	0000005F	6E40	D0	00165	11\$:	MOVL	SCA_HOLD[I], SCA[I]	0467	
	50			8F	F3	00168		AOBLEQ	#95, I, 11\$		
	EC	B9		6A	D0	00173		MOVL	PP_SCA, @SCA+100		
	F0	B9		04	AA	D0	00177	MOVL	PP_SCA+4, @SCA+104		
	F4	B9		08	AA	D0	0017C	MOVL	PP_SCA+8, @SCA+108		
	F8	B9		0C	AA	D0	00181	MOVL	PP_SCA+12, @SCA+112		
	FC	B9		10	AA	D0	00186	MOVL	PP_SCA+16, @SCA+116		
	00	B9		14	AA	D0	0018B	MOVL	PP_SCA+20, @SCA+120		
	04	B9		18	AA	D0	00190	MOVL	PP_SCA+24, @SCA+124		
	08	B9		1C	AA	D0	00195	MOVL	PP_SCA+28, @SCA+128		
	0C	B9		20	AA	D0	0019A	MOVL	PP_SCA+32, @SCA+132		
	10	B9		24	AA	D0	0019F	MOVL	PP_SCA+36, @SCA+136		
	14	B9		28	AA	D0	001A4	MOVL	PP_SCA+40, @SCA+140		
	18	B9		2C	AA	D0	001A9	MOVL	PP_SCA+44, @SCA+144		

	04	6A	EC	B9	D0	001AE	MOVL	@SCA+100, PP_SCA	0469	
	08	AA	F0	B9	D0	001B2	MOVL	@SCA+104, PP_SCA+4		
	0C	AA	F4	B9	D0	001B7	MOVL	@SCA+108, PP_SCA+8		
	10	AA	F8	B9	D0	001BC	MOVL	@SCA+112, PP_SCA+12		
	14	AA	FC	B9	D0	001C1	MOVL	@SCA+116, PP_SCA+16		
	18	AA	00	B9	D0	001C6	MOVL	@SCA+120, PP_SCA+20		
	1C	AA	04	B9	D0	001CB	MOVL	@SCA+124, PP_SCA+24		
	20	AA	08	B9	D0	001D0	MOVL	@SCA+128, PP_SCA+28		
	24	AA	0C	B9	D0	001D5	MOVL	@SCA+132, PP_SCA+32		
	28	AA	10	B9	D0	001DA	MOVL	@SCA+136, PP_SCA+36		
	2C	AA	14	B9	D0	001DF	MOVL	@SCA+140, PP_SCA+40		
			18	B9	D0	001E4	MOVL	@SCA+144, PP_SCA+44		
				50	D4	001E9	CLRL	I	0473	
F2	6E40		88	A940	D0	001EB	12\$: MOVL	SCA[I], SCA_HOLD[I]	0474	
	50	0000005F		8F	F3	001F1	AOBLEQ	#95, I, 12\$		
				50	D4	001F9	CLRL	I	0476	
F2	D8	AD40	00000000GEF40		D0	001FB	13\$: MOVL	IRA-4[I], IRA_HOLD[I]	0477	
	50			09	F3	00205	AOBLEQ	#7, I, 13\$		
			34	A9	D4	00209	CLRL	SCA+172	0481	
	54	A9		01	D0	0020C	MOVL	#1, SCA+204	0482	
			F0	B9	D4	00210	CLRL	@SCA+104	0483	
			EC	B9	D4	00213	CLRL	@SCA+100	0484	
			FC	B9	D4	00216	CLRL	@SCA+116	0485	
	00	B9	96	8F	9A	00219	MOVZBL	#150, @SCA+120	0486	
	30	A9		08	8A	0021E	BICB2	#8, SCA+168	0487	
	58	A9		01	D0	00222	MOVL	#1, SCA+208	0488	
	1C	A9		01	D0	00226	MOVL	#1, SCA+148	0489	
00000000G		EF		55	D0	0022A	MOVL	HOLD_KHAR, KHAR	0490	
	54	00000000G		EF	D0	00231	MOVL	MRA, -HOLD MRA	0495	
00000000G		EF	00000000G		EF	9E	00238	MOVAB	FOOMRA, MRA	0496
	53			6B	D0	00243	MOVL	TSF, HOLD TSF	0497	
	6B	00000000G		EF	9E	00246	MOVAB	FOOTSF, TSF	0498	
	50	00000000G		EF	D0	0024D	MOVL	MRA, R0	0504	
	08	A0	00000000G		8F	D0	00254	MOVL	#S FMRA, 8(R0)	
			0C	A0	D4	0025C	CLRL	12(R0)	0507	
			10	A0	9E	0025F	MOVAB	16(R0), (R0)		
	04	A0		60	D0	00263	MOVL	(R0), 4(R0)		
				50	D4	00267	CLRL	I	0509	
F8			00	BB40	D4	00269	14\$: CLRL	@TSF[I]		
	50			27	F3	0026D	AOBLEQ	#39, I, 14\$		
	0C		4C	AC	E9	00271	BLBC	CAPTION IS BOLD, 15\$	0512	
	08		30	A9	E9	00275	BLBC	SCA+168, 15\$		
	20	A9		01	88	00279	BISB2	#1, SCA+152	0515	
	4C	A9		01	88	0027D	BISB2	#1, SCA+196	0516	
08			50	AC	E9	00281	15\$: BLBC	CAPTION IS UNDERLINED, 16\$	0519	
	30	A9		01	E1	00285	BBC	#1, SCA+168, 16\$		
	20	A9		02	88	0028A	BISB2	#2, SCA+152	0522	
	4C	A9		02	88	0028E	BISB2	#2, SCA+196	0523	
			3C	AC	DD	00292	16\$: PUSHL	CAPTION CASE	0527	
00000000G		EF		01	FB	00295	CALLS	#1, SETCAS		
00000000G		EF		00	FB	0029C	CALLS	#0, SCANT	0530	
				7E	7C	002A3	CLRQ	-(SP)	0531	
				7E	D4	002A5	CLRL	-(SP)		
00000000G		EF		03	FB	002A7	CALLS	#3, ENDWRD		
		7E		34	AC	7D	002AE	MOVQ	CAPTION_MAJOR_TYPE, -(SP)	0547
		00000000G		EF	DD	002B2	PUSHL	FOOMRA	0546	
		00000000G		EF	DD	002B8	PUSHL	FOOMRA+12	0545	

	00000000G	EF		04	FB	002BE	CALLS	#4, PUTTXT				
	00000000G	EF		54	D0	002C5	MOVL	HOLD_MRA, MRA		0551		
		6B		53	D0	002CC	MOVL	HOLD_TSF, TSF		0552		
				50	D4	002CF	CLRL	I		0556		
	00000000G	EF	40	D8	AD40	D0	002D1	17\$:	18\$:	0557		
F2		50		09	F3	002DB	MOVL	IRA_HOLD[I], IRA-4[I]				
	00000000G	EF		55	D0	002DF	AOBLEQ	#9, I, 18\$				
				50	D4	002E6	MOVL	HOLD_KHAR, KHAR		0558		
				50	D4	002E8	CLRL	I		0562		
F2	88	A940		6E	40	D0	002E8	20\$:		0563		
		50	0000005F	8F	F3	002EE	MOVL	SCA_HOLD[I], SCA[I]				
	EC	B9		6A	D0	002F6	AOBLEQ	#95, I, 20\$				
	FO	B9	04	AA	D0	002FA	MOVL	PP_SCA, @SCA+100				
	F4	B9	08	AA	D0	002FF	MOVL	PP_SCA+4, @SCA+104				
	F8	B9	0C	AA	D0	00304	MOVL	PP_SCA+8, @SCA+108				
	FC	B9	10	AA	D0	00309	MOVL	PP_SCA+12, @SCA+112				
	00	B9	14	AA	D0	0030E	MOVL	PP_SCA+16, @SCA+116				
	04	B9	18	AA	D0	00313	MOVL	PP_SCA+20, @SCA+120				
	08	B9	1C	AA	D0	00318	MOVL	PP_SCA+24, @SCA+124				
	0C	B9	20	AA	D0	0031D	MOVL	PP_SCA+28, @SCA+128				
	10	B9	24	AA	D0	00322	MOVL	PP_SCA+32, @SCA+132				
	14	B9	28	AA	D0	00327	MOVL	PP_SCA+36, @SCA+136				
	18	B9	2C	AA	D0	0032C	MOVL	PP_SCA+40, @SCA+140				
	FO	B9		01	D0	00331	MOVL	PP_SCA+44, @SCA+144				
				01	D0	00331	MOVL	#1, @SCA+104		0565		
				34	A9	D4	00335	CLRL	SCA+172	0566		
	54	A9		01	D0	00338	MOVL	#1, SCA+204		0567		
		0C	4C	AC	E9	0033C	BLBC	CAPTION_IS_BOLD, 21\$		0571		
		08	30	A9	E9	00340	BLBC	SCA+168, 21\$				
	20	A9		01	88	00344	BISB2	#1, SCA+152		0574		
	4C	A9		01	88	00348	BISB2	#1, SCA+196		0575		
		0D	50	AC	E9	0034C	21\$:	BLBC	CAPTION_IS_UNDERLINED, 22\$	0578		
08	30	A9		01	E1	00350	BBC	#1, SCA+168, 22\$				
	20	A9		02	88	00355	BISB2	#2, SCA+152		0581		
	4C	A9		02	88	00359	BISB2	#2, SCA+196		0582		
7E	00	38	AC	01	7A	0035D	22\$:	EMUL	#1, CAPTION_MINOR_TYPE, #0, -(SP)	0588		
51	51	8E		03	7B	00363	EDIV	#3, (SP)+, R1, R1				
				50	D4	00368	CLRL	R0				
				51	D5	0036A	TSTL	R1				
				02	12	0036C	BNEQ	23\$				
				50	D6	0036E	INCL	R0				
				52	50	D2	00370	23\$:	MCOML	R0, SHOW_COUNTER		
	57		40	AC	D0	00373	MOVL	CAPTION_IS_CENTERED, R7		0592		
	08			57	E9	00377	BLBC	R7, 24\$				
	FC	B9		03	C0	0037A	ADDL2	#3, @SCA+116		0595		
	00	B9		03	C2	0037E	SUBL2	#3, @SCA+120		0596		
		03		52	E8	00382	24\$:	BLBS	SHOW_COUNTER, 25\$	0599		
				00EE	31	00385	BRW	40\$				
				3C	AC	DD	00388	25\$:	PUSHL	CAPTION_CASE	0605	
				14	AC	DD	0038B	PUSHL	COUNTER_MINOR_TYPE			
	00000000V	EF		02	FB	0038E	CALLS	#2, OUTCTR				
	00000000G	EF	00000000G	EF	D0	00395	MOVL	FS01, FS01+4		0609		
		53	00000000G	EF	D0	003A0	MOVL	FS01+12, R3		0610		
				52	D4	003A7	CLRL	I				
				1C	11	003A9	BRB	27\$				
				50	00000000G	FF	9A	003AB	26\$:	MOVZBL	@FS01+4, TEMP_CHAR	0614
					00000000G	EF	D6	003B2	INCL	FS01+4		
					00000000G	EF	D7	003B8	DECL	FS01+12		
						50	DD	003BE	PUSHL	TEMP_CHAR		0615

: R
: 1
: 1
:

					01	FB	003C0		CALLS	#1, ENDCHR		
					53	F3	003C7	27\$:	AOBLEQ	R3, I, 26\$		0610
					7E	7C	003CB		CLRQ	-(SP)		0620
					7E	D4	003CD		CLRL	-(SP)		
					03	FB	003CF		CALLS	#3, ENDWRD		
					54	D4	003D6		CLRL	R4		0625
				64	AC	D5	003D8		TSTL	LINES_BETWEEN		
					02	15	003DB		BLEQ	28\$		
					54	D6	003DD		INCL	R4		
				54	AC	C8	003DF	28\$:	BISL2	BREAK_BEFORE_CAPTION, R4		
				42	54	E8	003E3		BLBS	R4, 32\$		
56	4C	A9		02	00	EF	003E6		EXTZV	#0, #2, SCA+196, SCA_HOLD_C_BLDUN		0634
55	50	A9		02	00	EF	003EC		EXTZV	#0, #2, SCA+200, SCA_HOLD_AC_BLUN		0635
				52	AC	D0	003F2	20	MOVL	COUNTER_SPACES_AFTER, R2		0636
				8F	52	D1	003F6		CMPL	R2, #75		
					04	15	003FD		BLEQ	29\$		
				52	48	8F	9A	003FF	MOVZBL	#75, R2		
					53	D4	00403	29\$:	CLRL	I		
					11	11	00405		BRB	31\$		
	4C	A9		03	8A	00407	30\$:	BICB2	#3, SCA+196		0638	
	50	A9		03	8A	0040B		BICB2	#3, SCA+200		0639	
				20	DD	0040F		PUSHL	#32		0640	
				01	FB	00411		CALLS	#1, ENDCHR			
4C	A9	EB		53	52	F3	00418	51\$:	AOBLEQ	R2, I, 30\$		0636
50	A9	02		00	56	F0	0041C		INSV	SCA_HOLD_C_BLDUN, #0, #2, SCA+196		0644
		02		00	55	F0	00422		INSV	SCA_HOLD_AC_BLUN, #0, #2, SCA+200		0645
					7E	7C	00428	32\$:	CLRQ	-(SP)		0650
					7E	D4	0042A		CLRL	-(SP)		
					03	FB	0042C		CALLS	#3, ENDWRD		
				04	57	E8	00433		BLBS	R7, 33\$		0655
				2F	44	AC	E9	00436	BLBC	CAPTION_IS_FLUSH_RIGHT, 38\$		
				50	6B	D0	0043A	33\$:	MOVL	TSF, R0		0663
				05	60	AC	E8	0043D	BLBS	BREAK_BEFORE_CAPTION, 34\$		
					64	AC	D5	00441	TSTL	LINES_BETWEEN		0658
					1D	15	00444		BLEQ	37\$		
				51	6B	D0	00446	34\$:	MOVL	TSF, R1		0660
				0B	57	E9	00449		BLBC	R7, 35\$		0663
				50	04	A0	C3	0044C	SUBL3	4(R0), @SCA+120, R0		
					02	C6	00452		DIVL2	#2, R0		
					06	11	00455		BRB	36\$		
				50	04	A0	C3	00457	SUBL3	4(R0), @SCA+120, R0		0665
				28	50	D0	0045D	36\$:	MOVL	R0, 40(R1)		0662
					17	11	00461		BRB	41\$		0658
				58	04	A0	D0	00463	MOVL	4(R0), HOLD_LM		0671
					06	11	00467		BRB	39\$		0672
				0E	54	E8	00469	38\$:	BLBS	R4, 41\$		0684
				50	6B	D0	0046C		MOVL	TSF, R0		0689
				FC	04	A0	D0	0046F	MOVL	4(R0), @SCA+116		
					04	11	00474		BRB	41\$		0599
				1C	01	D0	00476	40\$:	MOVL	#1, SCA+148		0697
				00000000G	01	D0	0047A	41\$:	MOVL	#1, SCA+224		0703
					05	60	AC	E8	BLBS	BREAK_BEFORE_CAPTION, 42\$		0704
					64	AC	D5	00485	TSTL	LINES_BETWEEN		
					11	15	00488		BLEQ	43\$		
				00000000G	00	FB	0048A	42\$:	CALLS	#0, OUTNJ		0707
				00000000G	64	AC	DD	00491	PUSHL	LINES_BETWEEN		0708
					01	FB	00494		CALLS	#1, GCSKIP		

	54	00	BB	D0	0049B	43\$:	MOVL	@TSF, GEN CHAR	0712
		3C	AC	DD	0049F		PUSHL	CAPTION CASE	0715
	00000000G	EF	01	FB	004A2		CALLS	#1, SETCAS	
	58	A9	01	D0	004A9		MOVL	#1, SCA+208	0719
	00000000G	EF	00	FB	004AD		CALLS	#0, SCANT	0721
	00000000G	8F	00A0	C9	D1 004B4		CMPL	SCA+280, #RINTES	0727
			0D	13	004BD		BEQL	44\$	
			7E	7C	004BF		CLRQ	-(SP)	0729
			7E	D4	004C1		CLRL	-(SP)	
	00000000G	EF	03	FB	004C3		CALLS	#3, ENDWRD	
			30	11	004CA		BRB	47\$	
			00DC	C9	D5 004CC	44\$:	TSTL	SCA+340	0731
			2A	12	004D0		BNEQ	47\$	
			00D4	C9	D5 004D2		TSTL	SCA+332	0733
			24	15	004D6		BLEQ	47\$	
	05	EC	B9	E9	004D8		BLBC	@SCA+100, 45\$	0744
	50		03	D0	004DC		MOVL	#3, R0	
			02	11	004DF		BRB	46\$	
			50	D4	004E1	45\$:	CLRL	R0	
51	50	00D4	C9	C1	004E3	46\$:	ADDL3	SCA+332, R0, CHARS_TO_DROP	
	50	00000000G	EF	D0	004E9		MOVL	MRA, R0	0748
	OC	A0	51	C2	004F0		SUBL2	CHARS_TO_DROP, 12(R0)	
	04	A0	51	C2	004F4		SUBL2	CHARS_TO_DROP, 4(R0)	0749
		00D4	C9	D4	004F8		CLRL	SCA+332	0750
	51	0080	C9	D0	004FC	47\$:	MOVL	SCA+248, HOLD_WRD_PNTR	0764
	52	00D4	C9	D0	00501		MOVL	SCA+332, HOLD_LST_SP	0765
			50	D4	00506		CLRL	I	0766
F2	88	A940	6E40	D0	00508	48\$:	MOVL	SCA_HOLD[I], SCA[I]	0767
	50	0000005F	8F	F3	0050E		AOBLEQ	#95, I, 48\$	
	EC	B9	6A	D0	00516		MOVL	PP_SCA, @SCA+100	
	FO	B9	AA	D0	0051A		MOVL	PP_SCA+4, @SCA+104	
	F4	B9	AA	D0	0051F		MOVL	PP_SCA+8, @SCA+108	
	F8	B9	AA	D0	00524		MOVL	PP_SCA+12, @SCA+112	
	FC	B9	AA	D0	00529		MOVL	PP_SCA+16, @SCA+116	
	00	B9	AA	D0	0052E		MOVL	PP_SCA+20, @SCA+120	
	04	B9	AA	D0	00533		MOVL	PP_SCA+24, @SCA+124	
	08	B9	AA	D0	00538		MOVL	PP_SCA+28, @SCA+128	
	OC	B9	AA	D0	0053D		MOVL	PP_SCA+32, @SCA+132	
	10	B9	AA	D0	00542		MOVL	PP_SCA+36, @SCA+136	
	14	B9	AA	D0	00547		MOVL	PP_SCA+40, @SCA+140	
	18	B9	AA	D0	0054C		MOVL	PP_SCA+44, @SCA+144	
	0080	C9	51	D0	00551		MOVL	HOLD_WRD_PNTR, SCA+248	0769
	00D4	C9	52	D0	00556		MOVL	HOLD_LST_SP, SCA+332	0770
	25	48	AC	E9	0055B		BLBC	CAPTION IS RUN_IN, 49\$	0786
	00	BB	54	D1	0055F		CMPL	GEN_CHAR, @TSF	0789
			69	13	00563		BEQL	55\$	
			20	DD	00565		PUSHL	#32	0793
	00000000G	EF	01	FB	00567		CALLS	#1, ENDCHR	
	00000000G	EF	2D	DD	0056E		PUSHL	#45	0794
			01	FB	00570		CALLS	#1, ENDCHR	
			7E	7C	00577		CLRQ	-(SP)	0795
			01	DD	00579		PUSHL	#1	
	00000000G	EF	03	FB	0057B		CALLS	#3, ENDWRD	
			4A	11	00582		BRB	55\$	0788
	04		57	E8	00584	49\$:	BLBS	R7, 50\$	0816
	2E	44	AC	E9	00587		BLBC	CAPTION IS FLUSH_RIGHT, 54\$	
	09	00000000G	EF	E8	0058B	50\$:	BLBS	GCA+224, 5T\$	0818

		05	60	AC	E8	00592		BLBS	BREAK BEFORE CAPTION, 51\$: 0826
			64	AC	D5	00596		TSTL	LINES_BETWEEN		:
		50		1E	15	00599		BLEQ	54\$:
		OE		6B	D0	0059B	51\$:	MOVL	TSF, R0		: 0827
51	00	B9	04	57	E9	0059E		BLBC	R7, 52\$: 0830
		51		A0	C3	005A1		SUBL3	4(R0), @SCA+120, R1		:
		51		02	C6	005A7		DIVL2	#2, R1		:
		51		02	C2	005AA		SUBL2	#2, R1		:
		51		06	11	005AD		BRB	53\$:
	00	B9	04	A0	C3	005AF	52\$:	SUBL3	4(R0), @SCA+120, R1		: 0832
	28	A0		51	D0	005B5	53\$:	MOVL	R1, 40(R0)		: 0829
	00000000G	EF		00	FB	005B9	54\$:	CALLS	#0, OUTNJ		: 0834
			08	AC	DD	005C0		PUSHL	LINES_AFTER		: 0836
	00000000G	EF		01	FB	005C3		CALLS	#1, GCSKIP		:
	3C	A9		01	D0	005CA		MOVL	#1, SCA+180		: 0838
			00000000G	FF	DD	005CE	55\$:	PUSHL	@GCA+128		: 0842
	00000000G	EF		01	FB	005D4		CALLS	#1, SETCAS		:
				04	005DB			RET			: 0844

; Routine Size: 1500 bytes, Routine Base: \$CODE\$ + 0000

```

719 0845 1 GLOBAL ROUTINE OUTCTR (HEADER_TYPE, HEADER_CASE) : NOVALUE =
720 0846 1
721 0847 1 !++
722 0848 1 FUNCTIONAL DESCRIPTION:
723 0849 1
724 0850 1     OUTCTR generates (for header levels) a section number of the form
725 0851 1     1.2.3... For Example-Figure-Table titles it generates a
726 0852 1     counter of the form Example 2-4: ...
727 0853 1
728 0854 1 FORMAL PARAMETERS:
729 0855 1
730 0856 1     HEADER_TYPE     is the counter minor type of this header; it indicates
731 0857 1     whether this is an HC or an E/F/T title.
732 0858 1
733 0859 1     HEADER_CASE     is the case rules for this header, indicating upper/lower case.
734 0860 1
735 0861 1 IMPLICIT INPUTS:     None
736 0862 1
737 0863 1 IMPLICIT OUTPUTS:    None
738 0864 1
739 0865 1 ROUTINE VALUE:
740 0866 1 COMPLETION CODES:    None
741 0867 1
742 0868 1 SIDE EFFECTS:        None
743 0869 1
744 0870 1 --
745 0871 1
746 0872 2 BEGIN
747 0873 2 OWN
748 0874 2     OFFSET:                !As an own for debug purposes only***
749 0875 2
750 0876 2 LOCAL
751 0877 2     ENTITY_LEN,            !Length of ENTITY_TEXT.
752 0878 2     ENTITY_TEXT;          !Contains 'Example ' or 'Figure ' or 'Table '.
753 0879 2 !**     OFFSET;            !Offset into entity counter and display-code structures.
754 0880 2
755 0881 2 FS_INIT (FS01);
756 0882 2
757 0883 2 SELECT .HEADER_TYPE OF
758 0884 2 SET
759 0885 2 !
760 0886 2 [MIN_HL_INF] :
761 0887 2 !
762 0888 2 BEGIN
763 0889 3 IF .PAGEN [SCT_TYP] NEQ 0
764 0890 3 THEN
765 0891 3 !This is a document divided into sections.
766 0892 4 BEGIN
767 0893 4     FS_LENGTH (FS01) = PACSEC (PAGEN, FS_NEXT (FS01));
768 0894 4
769 0895 4     INCR I FROM 1 TO .HLLIST [CL_INDEX] DO
770 0896 5 BEGIN
771 0897 5     FS_WCHAR (FS01, %C'.');
772 0898 5     FS_LENGTH (FS01) = .FS_LENGTH (FS01) + PACXXX (.HLLIST [I], FS_NEXT (FS01), .HLDSP [I]
773 0899 4     END;
774 0900 4 END
775 0901 3 ELSE

```



```

776 0902 3 !Not a document divided into sections.
777 0903 4 BEGIN
778 0904 4 IF .HLLIST [CL_INDEX] EQL 1
779 0905 4 THEN
780 0906 4 !Special case for .HEADER LEVEL 1
781 0907 4 !Don't output a subsection number of zero.
782 0908 5 BEGIN
783 0909 5 FS_LENGTH (FS01) = PACXXX (.HLLIST [1], FS_NEXT (FS01), .HLDSP [0]);
784 0910 5 RETURN;
785 0911 4 END;
786 0912 4
787 0913 4 !Not a .HL1
788 0914 4 INCR I FROM 1 TO .HLLIST [CL_INDEX] DO
789 0915 5 BEGIN
790 0916 5 FS_LENGTH (FS01) = .FS_LENGTH (FS01) + PACXXX (.HLLIST [I], FS_NEXT (FS01), .HLDSP [I
791 0917 5 FS_WCHAR (FS01, %C'.');
792 0918 4 END;
793 0919 4
794 0920 4 !Always get an extra dot, so remove it.
795 0921 4 FS_LENGTH (FS01) = .FS_LENGTH (FS01) - 1;
796 0922 4 CH$PLUS (FS_NEXT (FS01), -1);
797 0923 3 END;
798 0924 3
799 0925 3 RETURN;
800 0926 2 END; ! End of HL counter processing.
801 0927 2 !
802 0928 2 [MIN_EXAMP_INF] :
803 0929 2 !
804 0930 3 BEGIN
805 0931 3 ENTITY_TEXT =
806 0932 4 (CASE .HEADER_CASE FROM LEAVE_CASE TO FIRST_CAPS OF
807 0933 4 SET
808 0934 4
809 U 0935 4 %IF german %THEN
810 U 0936 4 [LEAVE_CASE] : CH$PTR (UPLIT ('Beispiel'));
811 U 0937 4 [FORCE_UPPER] : CH$PTR (UPLIT ('BEISPIEL'));
812 U 0938 4 [FIRST_CAPS] : CH$PTR (UPLIT ('Beispiel'));
813 0939 4 %ELSE
814 U 0940 4 %IF french %THEN
815 U 0941 4 [LEAVE_CASE] : CH$PTR (UPLIT ('Example'));
816 U 0942 4 [FORCE_UPPER] : CH$PTR (UPLIT ('EXAMPLE'));
817 U 0943 4 [FIRST_CAPS] : CH$PTR (UPLIT ('Example'));
818 0944 4 %ELSE
819 U 0945 4 %IF italian %THEN
820 U 0946 4 [LEAVE_CASE] : CH$PTR (UPLIT ('Example'));
821 U 0947 4 [FORCE_UPPER] : CH$PTR (UPLIT ('EXAMPLE'));
822 U 0948 4 [FIRST_CAPS] : CH$PTR (UPLIT ('Example'));
823 0949 4 %ELSE
824 0950 4 [LEAVE_CASE] : CH$PTR (UPLIT ('Example'));
825 0951 4 [FORCE_UPPER] : CH$PTR (UPLIT ('EXAMPLE'));
826 0952 4 [FIRST_CAPS] : CH$PTR (UPLIT ('Example'));
827 0953 4 %FI %FI %FI
828 0954 3 TES);
829 0955 3
830 U 0956 3 %IF german %THEN
831 U 0957 3 ENTITY_LEN = 8;
832 0958 3 %ELSE

```

```

833 U 0959 3 %IF french %THEN
834 U 0960 3     ENTITY_LEN = 7;
835 0961 3 %ELSE
836 U 0962 3 %IF italian %THEN
837 U 0963 3     ENTITY_LEN = 7;
838 0964 3 %ELSE
839 0965 3     ENTITY_LEN = 7;
840 0966 3 %FI %FI %FI
841 0967 3     OFFSET = EXAMP_OFFSET;
842 0968 3     END;
843 0969 3     !
844 0970 3     [MIN_FIGUR_INF] :
845 0971 3     !
846 0972 3     BEGIN
847 0973 3     ENTITY_TEXT =
848 0974 4         (CASE .HEADER_CASE FROM LEAVE_CASE TO FIRST_CAPS OF
849 0975 4             SET
850 0976 4
851 U 0977 4 %IF german %THEN
852 U 0978 4     [LEAVE_CASE] : CH$PTR (UPLIT ('Bild'));
853 U 0979 4     [FORCE_UPPER] : CH$PTR (UPLIT ('BILD'));
854 U 0980 4     [FIRST_CAPS] : CH$PTR (UPLIT ('Bild'));
855 0981 4 %ELSE
856 U 0982 4 %IF french %THEN
857 U 0983 4     [LEAVE_CASE] : CH$PTR (UPLIT ('Figure'));
858 U 0984 4     [FORCE_UPPER] : CH$PTR (UPLIT ('FIGURE'));
859 U 0985 4     [FIRST_CAPS] : CH$PTR (UPLIT ('Figure'));
860 0986 4 %ELSE
861 U 0987 4 %IF italian %THEN
862 U 0988 4     [LEAVE_CASE] : CH$PTR (UPLIT ('Figure'));
863 U 0989 4     [FORCE_UPPER] : CH$PTR (UPLIT ('FIGURE'));
864 U 0990 4     [FIRST_CAPS] : CH$PTR (UPLIT ('Figure'));
865 0991 4 %ELSE
866 0992 4     [LEAVE_CASE] : CH$PTR (UPLIT ('Figure'));
867 0993 4     [FORCE_UPPER] : CH$PTR (UPLIT ('FIGURE'));
868 0994 4     [FIRST_CAPS] : CH$PTR (UPLIT ('Figure'));
869 0995 4 %FI %FI %FI
870 0996 4
871 0997 3     TES);
872 0998 3
873 U 0999 3 %IF german %THEN
874 U 1000 3     ENTITY_LEN = 4;
875 1001 3 %ELSE
876 U 1002 3 %IF french %THEN
877 U 1003 3     ENTITY_LEN = 6;
878 1004 3 %ELSE
879 U 1005 3 %IF italian %THEN
880 U 1006 3     ENTITY_LEN = 6;
881 1007 3 %ELSE
882 1008 3     ENTITY_LEN = 6;
883 1009 3 %FI %FI %FI
884 1010 3     OFFSET = FIGUR_OFFSET;
885 1011 3     END;
886 1012 3     !
887 1013 3     [MIN_TABLE_INF] :
888 1014 3     !
889 1015 3     BEGIN

```

```

: 890      1016 3      ENTITY TEXT =
: 891      1017 4      (CASE .HEADER_CASE FROM LEAVE_CASE TO FIRST_CAPS OF
: 892      1018 4      SET
: 893      1019 4
: 894      U 1020 4  %IF german %THEN
: 895      U U 1021 4      [LEAVE_CASE] : CH$PTR (UPLIT ('Tabelle'));
: 896      U U 1022 4      [FORCE_UPPER] : CH$PTR (UPLIT ('TABELLE'));
: 897      U 1023 4      [FIRST_CAPS] : CH$PTR (UPLIT ('Tabelle'));
: 898      1024 4  %ELSE
: 899      U 1025 4  %IF french %THEN
: 900      U U 1026 4      [LEAVE_CASE] : CH$PTR (UPLIT ('Table'));
: 901      U U 1027 4      [FORCE_UPPER] : CH$PTR (UPLIT ('TABLE'));
: 902      U 1028 4      [FIRST_CAPS] : CH$PTR (UPLIT ('Table'));
: 903      1029 4  %ELSE
: 904      U 1030 4  %IF italian %THEN
: 905      U U 1031 4      [LEAVE_CASE] : CH$PTR (UPLIT ('Table'));
: 906      U U 1032 4      [FORCE_UPPER] : CH$PTR (UPLIT ('TABLE'));
: 907      U 1033 4      [FIRST_CAPS] : CH$PTR (UPLIT ('Table'));
: 908      1034 4  %ELSE
: 909      1035 4      [LEAVE_CASE] : CH$PTR (UPLIT ('Table'));
: 910      1036 4      [FORCE_UPPER] : CH$PTR (UPLIT ('TABLE'));
: 911      1037 4      [FIRST_CAPS] : CH$PTR (UPLIT ('Table'));
: 912      1038 4  %FI %FI %FI
: 913      1039 3      TES);
: 914      1040 3
: 915      U 1041 3  %IF german %THEN
: 916      U 1042 3      ENTITY_LEN = 7;
: 917      1043 3  %ELSE
: 918      U 1044 3  %IF french %THEN
: 919      U 1045 3      ENTITY_LEN = 5;
: 920      1046 3  %ELSE
: 921      U 1047 3  %IF italian %THEN
: 922      U 1048 3      ENTITY_LEN = 5;
: 923      1049 3  %ELSE
: 924      1050 3      ENTITY_LEN = 5;
: 925      1051 3  %FI %FI %FI
: 926      1052 3      OFFSET = TABLE_OFFSET;
: 927      1053 2      END;
: 928      1054 2
: 929      1055 2      [MIN_APPEND_INF] :
: 930      1056 2
: 931      1057 3      BEGIN
: 932      1058 3      ENTITY TEXT =
: 933      1059 4      (CASE .HEADER_CASE FROM LEAVE_CASE TO FIRST_CAPS OF
: 934      1060 4      SET
: 935      1061 4
: 936      U 1062 4  %IF german %THEN
: 937      U U 1063 4      [LEAVE_CASE] : CH$PTR (UPLIT ('Anhang'));
: 938      U U 1064 4      [FORCE_UPPER] : CH$PTR (UPLIT ('ANHANG'));
: 939      U 1065 4      [FIRST_CAPS] : CH$PTR (UPLIT ('Anhang'));
: 940      1066 4  %ELSE
: 941      U 1067 4  %IF french %THEN
: 942      U U 1068 4      [LEAVE_CASE] : CH$PTR (UPLIT ('Appendix'));
: 943      U U 1069 4      [FORCE_UPPER] : CH$PTR (UPLIT ('APPENDIX'));
: 944      U 1070 4      [FIRST_CAPS] : CH$PTR (UPLIT ('Appendix'));
: 945      1071 4  %ELSE
: 946      U 1072 4  %IF italian %THEN

```

```

: 947 U 1073 4 [LEAVE_CASE] : CH$PTR (UPLIT ('Appendix'));
: 948 U 1074 4 [FORCE_UPPER] : CH$PTR (UPLIT ('APPENDIX'));
: 949 U 1075 4 [FIRST_CAPS] : CH$PTR (UPLIT ('Appendix'));
: 950 1076 4 %ELSE
: 951 1077 4 [LEAVE_CASE] : CH$PTR (UPLIT ('Appendix'));
: 952 1078 4 [FORCE_UPPER] : CH$PTR (UPLIT ('APPENDIX'));
: 953 1079 4 [FIRST_CAPS] : CH$PTR (UPLIT ('Appendix'));
: 954 1080 4 %FI %FI %FI
: 955 1081 4
: 956 1082 4 TES);
: 957 1083 4
: 958 U 1084 4 %IF german %THEN
: 959 U 1085 4 ENTITY_LEN = 6;
: 960 1086 4 %ELSE
: 961 U 1087 4 %IF french %THEN
: 962 U 1088 4 ENTITY_LEN = 8;
: 963 1089 4 %ELSE
: 964 U 1090 4 %IF italian %THEN
: 965 U 1091 4 ENTITY_LEN = 8;
: 966 1092 4 %ELSE
: 967 1093 4 ENTITY_LEN = 8;
: 968 1094 4 %FI %FI %FI
: 969 1095 4 OFFSET = APPEND_OFFSET;
: 970 1096 4 END;
: 971 1097 4
: 972 1098 4 [MIN_CHAPT_INF] :
: 973 1099 4
: 974 1100 4 BEGIN
: 975 1101 4 ENTITY TEXT =
: 976 1102 4 (CASE .HEADER_CASE FROM LEAVE_CASE TO FIRST_CAPS OF
: 977 1103 4 SET
: 978 1104 4
: 979 U 1105 4 %IF german %THEN
: 980 U 1106 4 [LEAVE_CASE] : CH$PTR (UPLIT ('Kapitel'));
: 981 U 1107 4 [FORCE_UPPER] : CH$PTR (UPLIT ('KAPITEL'));
: 982 U 1108 4 [FIRST_CAPS] : CH$PTR (UPLIT ('Kapitel'));
: 983 1109 4 %ELSE
: 984 U 1110 4 %IF french %THEN
: 985 U 1111 4 [LEAVE_CASE] : CH$PTR (UPLIT ('Chapter'));
: 986 U 1112 4 [FORCE_UPPER] : CH$PTR (UPLIT ('CHAPTER'));
: 987 U 1113 4 [FIRST_CAPS] : CH$PTR (UPLIT ('Chapter'));
: 988 1114 4 %ELSE
: 989 U 1115 4 %IF italian %THEN
: 990 U 1116 4 [LEAVE_CASE] : CH$PTR (UPLIT ('Chapter'));
: 991 U 1117 4 [FORCE_UPPER] : CH$PTR (UPLIT ('CHAPTER'));
: 992 U 1118 4 [FIRST_CAPS] : CH$PTR (UPLIT ('Chapter'));
: 993 1119 4 %ELSE
: 994 1120 4 [LEAVE_CASE] : CH$PTR (UPLIT ('Chapter'));
: 995 1121 4 [FORCE_UPPER] : CH$PTR (UPLIT ('CHAPTER'));
: 996 1122 4 [FIRST_CAPS] : CH$PTR (UPLIT ('Chapter'));
: 997 1123 4 %FI %FI %FI
: 998 1124 4 TES);
: 999 1125 4
: 1000 U 1126 4 %IF french %THEN
: 1001 U 1127 4 ENTITY_LEN = 7;
: 1002 1128 4 %ELSE
: 1003 U 1129 4 %IF italian %THEN

```

```

1004 U 1130 3 ENTITY_LEN = 7;
1005 1131 3 %ELSE ! The German & English words are the same length.
1006 1132 3 ENTITY_LEN = 7;
1007 1133 3 %FI %FI
1008 1134 3 OFFSET = CHAP_OFFSET;
1009 1135 3 END;
1010 1136 3 !
1011 1137 3 [MIN_EXAMP_INF, MIN_FIGUR_INF, MIN_TABLE_INF, MIN_APPEND_INF, MIN_CHAPT_INF] :
1012 1138 3 !
1013 1139 3 BEGIN
1014 1140 3
1015 1141 3 !Write the entity identifier ('Example ' or ...).
1016 1142 3 INCR I FROM 1 TO .ENTITY_LEN DO
1017 1143 3     ENDCHR (CH$RCHAR_A (ENTITY_TEXT));
1018 1144 3
1019 1145 3 !End the entity with a space.
1020 1146 3 ENDWRD (TRUE,FALSE,FALSE);
1021 1147 3
1022 1148 3 !Write the pre-counter string, if any.
1023 1149 3 BEGIN
1024 1150 3 LOCAL
1025 1151 3     PTR,
1026 1152 3     LEN;
1027 1153 3 PTR = .ECC [.OFFSET, ECC$A_PRE_PTR];
1028 1154 3 LEN = .ECC [.OFFSET, ECC$H_PRE_LEN];
1029 1155 3
1030 1156 3 INCR I FROM 1 TO .LEN DO
1031 1157 3     FS_WCHAR (FS01, CH$RCHAR_A (PTR));
1032 1158 3 END;
1033 1159 3
1034 1160 3 !If the document is divided into sections, write the chapter number and '-'.
1035 1161 3 !However if this is a chapter/appendix beign processed, we don't
1036 1162 3 !want to pick up the section number.
1037 1163 3 IF .OFFSET LSS CHAP_OFFSET
1038 1164 3 THEN !This isn't a chapter/appendix
1039 1165 3     IF .PAGEN [SCT_TYP] NEQ 0
1040 1166 3     THEN
1041 1167 3     BEGIN
1042 1168 3     FS_LENGTH (FS01) = .FS_LENGTH (FS01) + PACSEC (PAGEN, FS_NEXT (FS01));
1043 1169 3     FS_WCHAR (FS01, %C'-');
1044 1170 3     END;
1045 1171 3
1046 1172 3 !Write the entity's counter in the current display mode.
1047 1173 3 !Note that this is the section number for chapter/appendix
1048 1174 3 FS_LENGTH (FS01) = .FS_LENGTH (FS01) +
1049 1175 3     PACXXX (.ECC [.OFFSET, ECC$H_COUNTER],
1050 1176 3     FS_NEXT (FS01),
1051 1177 3     .ECC [.OFFSET, ECC$H_DISPLAY_DESC]);
1052 1178 3
1053 1179 3 !Write the post-counter string, if any.
1054 1180 3 BEGIN
1055 1181 3 LOCAL
1056 1182 3     PTR,
1057 1183 3     LEN;
1058 1184 3 PTR = .ECC [.OFFSET, ECC$A_POST_PTR];
1059 1185 3 LEN = .ECC [.OFFSET, ECC$H_POST_LEN];
1060 1186 3

```

```

: 1061      1187  4      INCR I FROM 1 TO .LEN DO
: 1062      1188  4      FS_WCHAR (FS01, CH$RCHAR_A (PTR));
: 1063      1189  4
: 1064      1190  3      END:
: 1065      1191  3
: 1066      1192  2      END:
: 1067      1193  2      TES:
: 1068      1194  2      ! End of E/F/T counter processing.
: 1069      1195  1      END:
                                !End of OUTCTR

```

```

                                .PSECT $PLITS,NOWRT,NOEXE,2
00 65 6C 70 6D 61 78 45 00000 P.AAA: .ASCII \Example\<0>
00 45 4C 50 4D 41 58 45 00008 P.AAB: .ASCII \EXAMPLE\<0>
00 65 6C 70 6D 61 78 45 00010 P.AAC: .ASCII \Example\<0>
00 00 65 72 75 67 69 46 00018 P.AAD: .ASCII \Figure\<0><0>
00 00 45 52 55 47 49 46 00020 P.AAE: .ASCII \FIGURE\<0><0>
00 00 65 72 75 67 69 46 00028 P.AAF: .ASCII \Figure\<0><0>
00 00 00 65 6C 62 61 54 00030 P.AAG: .ASCII \Table\<0><0><0>
00 00 00 45 4C 42 41 54 00038 P.AAH: .ASCII \TABLE\<0><0><0>
00 00 00 65 6C 62 61 54 00040 P.AAI: .ASCII \Table\<0><0><0>
78 69 64 6E 65 70 70 41 00048 P.AAJ: .ASCII \Appendix\
58 49 44 4E 45 50 50 41 00050 P.AAK: .ASCII \APPENDIX\
78 69 64 6E 65 70 70 41 00058 P.AAL: .ASCII \Appendix\
00 72 65 74 70 61 68 43 00060 P.AAM: .ASCII \Chapter\<0>
00 52 45 54 50 41 48 43 00068 P.AAN: .ASCII \CHAPTER\<0>
00 72 65 74 70 61 68 43 00070 P.AAO: .ASCII \chapter\<0>

```

```

                                .PSECT $OWNS,NOEXE,2
00030 OFFSET: .BLKB 4

```

```

                                .PSECT $CODE$,NOWRT,2
                                .ENTRY OUTCTR, Save R2,R3,R4,R5,R6,R7,R8,R9,R10,-
                                R11
00000000G EF 9E 00002 MOVAB PACXXX, R11
00000000G EF 9E 00009 MOVAB PAGEN, R10
00000000G EF 9F 00010 MOVAB ECC+20, R9
00000000G EF 9E 00017 MOVAB HLLIST+4, R8
00000000' EF 9E 0001E MOVAB OFFSET, R7
00000000' EF 9E 00025 MOVAB P.AAA, R6
00000000G EF 9E 0002C MOVAB FS01+4, R5
                                C8 A5 D4 00033 CLRL FS01+12
                                FC A5 9E 00036 MOVAB FS01+16, FS01
                                65 FC A5 D0 0003B MOVL FS01, FS01+4
                                53 04 AC D0 0003F MOVL HEADER_TYPE, R3
                                7F 12 00043 BNEQ 7$
                                OF 6A 93 00045 BITB PAGEN, #15
                                37 13 00048 BEQL 3$
                                55 DD 0004A PUSHL R5
                                SA DD 0004C PUSHL R10
00000000G EF 02 FB 0004E CALLS #2, PACSEC
                                0845
                                0881
                                0883
                                0886
                                0889
                                0893

```

	08	A5	50	D0	00055		MOV	R0, FS01+12		
		54	68	D0	00059		MOV	HLLIST+4, R4		0895
			52	D4	0005C		CLRL	I		
			1C	11	0005E		BRB	2\$		
	00	B5	2E	90	00060	1\$:	MOV	#46, @FS01+4		0897
			65	D6	00064		INCL	FS01+4		
		08	A5	D6	00066		INCL	FS01+12		
			42	DD	00069		PUSHL	HLDSP-4[I]		0898
			55	DD	00070		PUSHL	R5		
			68	DD	00072		PUSHL	HLLIST+4[I]		
		68	03	FB	00075		CALLS	#3, PACXXX		
	08	A5	50	C0	00078		ADDL2	R0, FS01+12		
E0		52	54	F3	0007C	2\$:	AOBLEQ	R4, I, 1\$		0895
				04	00080		RET			0889
		01	68	D1	00081	3\$:	CMP	HLLIST+4, #1		0904
			13	12	00084		BNEQ	4\$		
			00	DD	00086		PUSHL	HLDSP		0909
			55	DD	0008C		PUSHL	R5		
			04	DD	0008E		PUSHL	HLLIST+8		
		68	03	FB	00091		CALLS	#3, PACXXX		
	08	A5	50	D0	00094		MOV	R0, FS01+12		
				04	00098		RET			0908
		54	68	D0	00099	4\$:	MOV	HLLIST+4, R4		0914
			52	D4	0009C		CLRL	I		
			1C	11	0009E		BRB	6\$		
			42	DD	000A0	5\$:	PUSHL	HLDSP-4[I]		0916
			55	DD	000A7		PUSHL	R5		
			68	DD	000A9		PUSHL	HLLIST+4[I]		
		68	03	FB	000AC		CALLS	#3, PACXXX		
	08	A5	50	C0	000AF		ADDL2	R0, FS01+12		
	00	B5	2E	90	000B3		MOV	#46, @FS01+4		0917
			65	D6	000B7		INCL	FS01+4		
			08	A5	D6	000B9	INCL	FS01+12		
E0		52	54	F3	000BC	6\$:	AOBLEQ	R4, I, 5\$		0914
			08	A5	D7	000C0	DECL	FS01+12		0921
				04	000C3		RET			0888
		07	53	D1	000C4	7\$:	CMP	R3, #7		0928
			1F	12	000C7		BNEQ	13\$		
02	00	00	08	AC	000C9		CASEL	HEADER_CASE, #0, #2		0932
0011	000B		00	CF	000C9		CASEL	9\$-8\$, -		
				00	000CE	8\$:	.WORD	10\$-8\$, -		
								11\$-8\$, -		
		52	66	9E	000D4	9\$:	MOVAB	P.AAA, ENTITY_TEXT		0950
			0A	11	000D7		BRB	12\$		
		52	08	A6	9E	000D9	10\$:	MOVAB	P.AAB, ENTITY_TEXT	0951
			04	11	000DD		BRB	12\$		
		52	10	A6	9E	000DF	11\$:	MOVAB	P.AAC, ENTITY_TEXT	0952
		54	07	D0	000E3	12\$:	MOV	#7, ENTITY_LEN		0965
			67	D4	000E6		CLRL	OFFSET		0967
		0A	53	D1	000E8	13\$:	CMP	R3, #10		0970
			21	12	000EB		BNEQ	19\$		
02	00	00	08	AC	000ED		CASEL	HEADER_CASE, #0, #2		0974
0012	000C		00	CF	000ED		CASEL	15\$-14\$, -		
				00	000F2	14\$:	.WORD	16\$-14\$, -		
								17\$-14\$, -		
		52	18	A6	9E	000F8	15\$:	MOVAB	P.AAD, ENTITY_TEXT	0992
			0A	11	000FC		BRB	18\$		

	52	20	A6	9E	000FE	16\$:	MOVAB	P.AAE, ENTITY_TEXT	0993
			04	11	00102		BRB	18\$	
	52	28	A6	9E	00104	17\$:	MOVAB	P.AAF, ENTITY_TEXT	0994
	54		06	D0	00108	18\$:	MOVL	#6, ENTITY_LEN	1008
	67		01	D0	0010B		MOVL	#1, OFFSET	1010
	0D		53	D1	0010E	19\$:	CMPL	R3, #13	1013
			21	12	00111		BNEQ	25\$	
02	00	08	AC	CF	00113		CASEL	HEADER CASE, #0, #2	1017
0012	000C	0006			0011B	20\$:	.WORD	21\$-20\$,- 22\$-20\$,- 23\$-20\$	
	52	30	A6	9E	0011E	21\$:	MOVAB	P.AAG, ENTITY_TEXT	1035
			0A	11	00122		BRB	24\$	
	52	38	A6	9E	00124	22\$:	MOVAB	P.AAH, ENTITY_TEXT	1036
			04	11	00128		BRB	24\$	
	52	40	A6	9E	0012A	23\$:	MOVAB	P.AAI, ENTITY_TEXT	1037
	54		05	D0	0012E	24\$:	MOVL	#5, ENTITY_LEN	1050
	67		02	D0	00131		MOVL	#2, OFFSET	1052
	13		53	D1	00134	25\$:	CMPL	R3, #19	1055
			21	12	00137		BNEQ	31\$	
02	00	08	AC	CF	00139		CASEL	HEADER CASE, #0, #2	1059
0012	000C	0006			0013E	26\$:	.WORD	27\$-26\$,- 28\$-26\$,- 29\$-26\$	
	52	48	A6	9E	00144	27\$:	MOVAB	P.AAJ, ENTITY_TEXT	1077
			0A	11	00148		BRB	30\$	
	52	50	A6	9E	0014A	28\$:	MOVAB	P.AAK, ENTITY_TEXT	1078
			04	11	0014E		BRB	30\$	
	52	58	A6	9E	00150	29\$:	MOVAB	P.AAL, ENTITY_TEXT	1079
	54		08	D0	00154	30\$:	MOVL	#8, ENTITY_LEN	1093
	67		0B	D0	00157		MOVL	#11, OFFSET	1095
	10		53	D1	0015A	31\$:	CMPL	R3, #16	1098
			21	12	0015D		BNEQ	37\$	
02	00	08	AC	CF	0015F		CASEL	HEADER CASE, #0, #2	1102
0012	000C	0006			00164	32\$:	.WORD	33\$-32\$,- 34\$-32\$,- 35\$-32\$	
	52	60	A6	9E	0016A	33\$:	MOVAB	P.AAM, ENTITY_TEXT	1120
			0A	11	0016E		BRB	36\$	
	52	68	A6	9E	00170	34\$:	MOVAB	P.AAN, ENTITY_TEXT	1121
			04	11	00174		BRB	36\$	
	52	70	A6	9E	00176	35\$:	MOVAB	P.AAO, ENTITY_TEXT	1122
	54		07	D0	0017A	36\$:	MOVL	#7, ENTITY_LEN	1132
	67		0A	D0	0017D		MOVL	#10, OFFSET	1134
	07		53	D1	00180	37\$:	CMPL	R3, #7	1137
			15	13	00183		BEQL	38\$	
	0A		53	D1	00185		CMPL	R3, #10	
			10	13	00188		BEQL	38\$	
	0D		53	D1	0018A		CMPL	R3, #13	
			0B	13	0018D		BEQL	38\$	
	10		53	D1	0018F		CMPL	R3, #16	
			06	13	00192		BEQL	38\$	
	13		53	D1	00194		CMPL	R3, #19	
			01	13	00197		BEQL	38\$	
				04	00199		RET		
			53	D4	0019A	38\$:	CLRL	1	1142
			0A	11	0019C		BRB	40\$	

		7E	82	9A	0019E	39\$:	MOVZBL	(ENTITY TEXT)+, -(SP)	1143	
		EF	01	FB	001A1		CALLS	#1, ENDCHR		
	F2	53	54	F3	001A8	40\$:	AOBLEQ	ENTITY_LEN, 1, 39\$		
			7E	7C	001AC		CLRQ	-(SP)	1146	
			01	DD	001AE		PUSHL	#1		
		00000000G	03	FB	001B0		CALLS	#3, ENDWRD		
	50	67	24	C5	001B7		MULL3	#36, OFFSET, R0	1153	
52	6940	20	18	EF	001BB		EXTZV	#24, #32, ECC+20[R0], PTR		
			01	A940	9F	001C1	PUSHAB	ECC+21[R0]	1154	
			51	9E	32	001C5	CVTWL	@(SP)+, LEN		
			50	D4	001C8		CLRL	I	1156	
			09	11	001CA		BRB	42\$		
		00	82	90	001CC	41\$:	MOVB	(PTR)+, @FS01+4	1157	
			65	D6	001D0		INCL	FS01+4		
			08	A5	D6	001D2	INCL	FS01+12		
	F3	50	51	F3	001D5	42\$:	AOBLEQ	LEN, 1, 41\$	1156	
		0A	67	D1	001D9		CMPL	OFFSET, #10	1163	
			1D	18	001DC		BGEQ	43\$		
			0F	6A	93	001DE	BITB	PAGEN, #15	1165	
			18	13	001E1		BEQL	43\$		
			55	DD	001E3		PUSHL	R5	1168	
			5A	DD	001E5		PUSHL	R10		
		00000000G	02	FB	001E7		CALLS	#2, PACSEC		
		08	50	C0	001EE		ADDL2	R0, FS01+12		
		00	2D	90	001F2		MOVB	#45, @FS01+4	1169	
			65	D6	001F6		INCL	FS01+4		
			08	A5	D6	001F8	INCL	FS01+12		
	50	67	24	C5	001FB	43\$:	MULL3	#36, OFFSET, R0	1177	
7E	FC	A940	18	EE	001FF		EXTV	#24, #16, ECC+16[R0], -(SP)		
			55	DD	00206		PUSHL	R5	1176	
7E	F8	A940	18	EE	00208		EXTV	#24, #32, ECC+12[R0], -(SP)	1175	
			03	FB	0020F		CALLS	#3, PACXXX		
			50	C0	00212		ADDL2	R0, FS01+12		
		08	24	C5	00216		MULL3	#36, OFFSET, R0	1184	
	50	67	08	EF	0021A		EXTZV	#8, #32, ECC+28[R0], PTR		
52	08	A940	18	EE	00221		EXTV	#24, #16, ECC+24[R0], LEN	1185	
51	04	A940	50	D4	00228		CLRL	I	1187	
			09	11	0022A		BRB	45\$		
			00	82	90	0022C	44\$:	MOVB	(PTR)+, @FS01+4	1188
			65	D6	00230		INCL	FS01+4		
			08	A5	D6	00232	INCL	FS01+12		
	F3	50	51	F3	00235	45\$:	AOBLEQ	LEN, 1, 44\$	1187	
			04	00239			RET		1195	

: Routine Size: 570 bytes, Routine Base: *CLOS\$ + 05DC

: 1070 1196 1
: 1071 1197 1 END
: 1072 1198 0 ELUDOM

!End of module

PSECT SUMMARY

Name	Bytes	Attributes
\$OWNS	52	NOVEC, WRT, RD, NOEXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)
\$CODES	2070	NOVEC, NOWRT, RD, EXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)
\$SPLITS	120	NOVEC, NOWRT, RD, NOEXE, 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	0	0	252	00:00.2
_\$255\$DUA28:[RUNOFF.SRC]DSRLIB.L32:1	1248	133	10	86	00:00.3

COMMAND QUALIFIERS

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

Size: 2070 code + 172 data bytes
 Run Time: 00:45.6
 Elapsed Time: 01:41.5
 Lines/CPU Min: 1575
 Lexemes/CPU-Min: 18065
 Memory Used: 341 pages
 Compilation Complete

