


```

TTTTTTTTT1  000000  PppppppP  PppppppP  AAAAAA  GGGGGGGG
TTTTTTTTTT  000000  PppppppP  PppppppP  AAAAAA  GGGGGGGG
  TT          00      00  PP          PP  PP          PP  AA          AA  GG
  TT          00      00  PP          PP  PP          PP  AA          AA  GG
  TT          00      00  PP          PP  PP          PP  AA          AA  GG
  TT          00      00  PP          PP  PP          PP  AA          AA  GG
  TT          00      00  PppppppP  PppppppP  AA          AA  GG
  TT          00      00  PppppppP  PppppppP  AA          AA  GG
  TT          00      00  PP          PP  PP          PP  AAAAAAAAAA  GG  GGGGGG
  TT          00      00  PP          PP  PP          PP  AAAAAAAAAA  GG  GGGGGG
  TT          00      00  PP          PP  PP          PP  AA          AA  GG          GG
  TT          00      00  PP          PP  PP          PP  AA          AA  GG          GG
  TT          00          PP          PP  PP          PP  AA          AA  GG          GG
  TT          000000  PP          PP  PP          PP  AA          AA  GG          GG
  TT          000000  PP          PP  PP          PP  AA          AA  GG          GG

```

```

LL          I111111  SSSSSSSS
LL          I111111  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
LL          II       SS
LLLLLLLLLLL I111111  SSSSSSSS
LLLLLLLLLLL I111111  SSSSSSSS

```



```

1 0001 0 %TITLE 'Top of the page processing'
2 0002 0 MODULE toppag ( 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: Handles the top of the page.
35 0035 1
36 0036 1 ENVIRONMENT: Transportable
37 0037 1
38 0038 1 AUTHOR: R.W.Friday CREATION DATE: November, 1981
39 0039 1

```

```

: 41 0040 1 %SBTTL 'Revision History'
: 42 0041 1   MODIFIED BY:
: 43 0042 1
: 44 0043 1   013  REM00013      Ray Marshall      21-Mar-1984
: 45 0044 1      Moved the conditionalized "page" text length to the head
: 46 0045 1      of the routine so it can be used everywhere it's needed.
: 47 0046 1      Also provided the logic for the other languages; even
: 48 0047 1      though we don't have the correct words yet.
: 49 0048 1
: 50 0049 1   012  REM00012      Ray Marshall      31-Jan-1984
: 51 0050 1      Added conditionals for the German word for "page": seite.
: 52 0051 1
: 53 0052 1   011  KFA00011      Ken Alden        5-Jul-1983
: 54 0053 1      Added extra conditional on tnout call to prevent
: 55 0054 1      calling if the topnote jobs are empty.
: 56 0055 1
: 57 0056 1   010  KFA00010      Ken Alden        14-Jun-1983
: 58 0057 1      Removed /down error messages.
: 59 0058 1
: 60 0059 1   009  KFA00009      Ken Alden        13-Jun-1983
: 61 0060 1      /DOWN calculations now do not affect line count on current page.
: 62 0061 1
: 63 0062 1   008  KFA00008      Ken Alden        10-May-1983
: 64 0063 1      Boosted the limit of the tuncation fixed strings to
: 65 0064 1      fix a truncation accvio.
: 66 0065 1
: 67 0066 1   007  RER00007      Ron Randall     05-Apr-1983
: 68 0067 1      For DSRPLUS: Improved topnote handling.
: 69 0068 1
: 70 0069 1   006  RER00006      Ron Randall     17-Mar-1983
: 71 0070 1      For DSRPLUS:
: 72 0071 1      Added code to output topnotes after title/subtitle.
: 73 0072 1
: 74 0073 1   005  REM00005      Ray Marshall     07-Mar-1983
: 75 0074 1      Global edit of all modules. Updated module names, idents,
: 76 0075 1      copyright dates. Changed require files to BLISS library.
: 77 0076 1      --
: 78 0077 1

```

```

80 0078 1 %SBTTL 'Module Level Declarations'
81 0079 1
82 0080 1 : TABLE OF CONTENTS:
83 0081 1
84 0082 1 FORWARD ROUTINE
85 0083 1   toppag      : NOVALUE,
86 0084 1   truncate  : NOVALUE;
87 0085 1
88 0086 1 : INCLUDE FILES:
89 0087 1
90 0088 1 LIBRARY 'NXPORT:XPORT';           ! XPORT Library
91 0089 1 REQUIRE 'REQ:RNODEF';         ! RUNOFF variant definitions
92 0220 1
93 U 0221 1 %IF DSRPLUS %THEN
94 U 0222 1 LIBRARY 'REQ:DPLLIB';       ! DSRPLUS BLISS Library
95 0223 1 %ELSE
96 0224 1 LIBRARY 'REQ:DSRLIB';       ! DSR BLISS Library
97 0225 1 %FI
98 0226 1
99 0227 1
100 0228 1 : MACROS:
101 0229 1
102 0230 1 : This macro makes certain that when blank lines at the top or bottom of a page
103 0231 1 : are skipped change bars don't get output.
104 0232 1
105 0233 1 MACRO
106 M 0234 1   NO_BAR_SKIP (N) =
107 M 0235 1     BEGIN
108 M 0236 1     LOCAL
109 M 0237 1     HOLD_BARS;
110 M 0238 1
111 M 0239 1     HOLD_BARS = .TSF_BARS;
112 M 0240 1     TSF_BARS  = FALSE;
113 M 0241 1     USKIPL (N);
114 M 0242 1     TSF_BARS  = .HOLD_BARS;
115 M 0243 1     END-%;
116 0244 1
117 0245 1 : EQUATED SYMBOLS:
118 0246 1
119 0247 1 LITERAL
120 0248 1   form_feed = %O'14';
121 0249 1
122 0250 1 : EXTERNAL REFERENCES:
123 0251 1
124 0252 1 EXTERNAL
125 0253 1   DATTIM      : VECTOR,           ! Binary date and time.
126 0254 1   FRA         : FIXED_STRING,
127 0255 1   GCA         : GCA_DEFINITION,
128 0256 1   HCT         : HCT_DEFINITION,
129 0257 1   MRA         : REF_FIXED_STRING,
130 0258 1   NPAGEN      : PAGE_DEFINITION,   ! Number of next page.
131 0259 1   PAGEN       : PAGE_DEFINITION,
132 0260 1   SCA         : SCA_DEFINITION,
133 0261 1   SPAGER      : BLOCKVECTOR [1, PAGE_SCT_SIZE], ! List of starting pages.
134 0262 1   TPAGER      : BLOCKVECTOR [1, PAGE_SCT_SIZE], ! List of terminating pages.
135 0263 1   PHAN        : PHAN_DEFINITION,
136 0264 1   SBTMRA     : FIXED_STRING,     ! Contains the subtitle, if any

```

```

: 137      0265 1      SBTTSF      : VECTOR [TSF_SIZE],      ! Subtitle descriptor.
: 138      0266 1      TITMRA      : FIXED STRING,          ! Contains the title, if any.
: 139      0267 1      TITTSF      : VECTOR [TSF_SIZE],      ! Title descriptor.
: 140      0268 1      TSF          : TSF_DEFINITION;
: 141      0269 1
: 142      U 0270 1      %IF DSRPLUS %THEN
: 143      U 0271 1      EXTERNAL
: 144      U 0272 1      TOPNOT      : TN_DEFINITION;
: 145      U 0273 1
: 146      U 0274 1      EXTERNAL ROUTINE
: 147      U 0275 1      TNOUT;
: 148      0276 1      %FI
: 149      0277 1
: 150      0278 1      EXTERNAL LITERAL
: 151      0279 1      RNFPWF;          ! 'Page number won't fit on title'
: 152      0280 1
: 153      0281 1      EXTERNAL LITERAL
: 154      0282 1      RINTES : UNSIGNED (8);      ! 'escape' character used in internal
: 155      0283 1                                          ! data structures
: 156      0284 1
: 157      0285 1      EXTERNAL ROUTINE
: 158      0286 1      CNVDAT,
: 159      0287 1      ERM,
: 160      0288 1      ERML,          ! Error message followed by location in file
: 161      0289 1      ERMN,          ! Error message w/number followed by location in file
: 162      0290 1      LOUT,
: 163      0291 1      PACPAG,
: 164      0292 1      PAGFND,
: 165      0293 1      PUTTPG,
: 166      0294 1      USKIPL;
: 167      0295 1

```

```

169 0296 1 %SBTTL 'TOPPAG --'
170 0297 1 GLOBAL ROUTINE toppag : NOVALUE =
171 0298 1
172 0299 1 ++
173 0300 1 FUNCTIONAL DESCRIPTION:
174 0301 1
175 0302 1     Does everything at the top of a page that is included
176 0303 1     in the header area.
177 0304 1
178 0305 1 FORMAL PARAMETERS:      None
179 0306 1
180 0307 1 IMPLICIT INPUTS:        None
181 0308 1
182 0309 1 IMPLICIT OUTPUTS:       None
183 0310 1
184 0311 1 ROUTINE VALUE:
185 0312 1 COMPLETION CODES:      None
186 0313 1
187 0314 1 SIDE EFFECTS:          None
188 0315 1 --
189 0316 1
190 0317 2 BEGIN
191 0318 2 OWN
192 0319 2     ! Text to add to the subtitle as truncated.
193 0320 2     FS_ALLOCATE (ADD_CHARS, 50),
194 0321 2     ! Saved copy of subtitle text overwritten by truncation.
195 0322 2     FS_ALLOCATE (SAVE_CHARS, 50);
196 0323 2
197 0324 2 LOCAL
198 0325 2     DATE_LENGTH,           ! Length of date string.
199 0326 2     FILL_COUNT,
200 0327 2     HOLD_MRA,             ! Areas for preserving current status.
201 0328 2     HOLD_MRA_LENGTH,
202 0329 2     HOLD_MRA_NEXT,
203 0330 2     HOLD_TSF,
204 0331 2     HOLD_TSF_ADJUST,
205 0332 2     HOLD_TSF_EXT_HL,
206 0333 2     HOLD_TSF_INT_HL,
207 0334 2     WORK_AREA : VECTOR [CH$ALLOCATION (50)],
208 0335 2     PAGE_LENGTH,         ! Number of characters in page number.
209 0336 2     page_string_length :
210 0337 2 %IF german %THEN
211 0338 2     INITIAL(6),
212 0339 2 %ELSE
213 0340 2 %IF french %THEN
214 0341 2     INITIAL(5),
215 0342 2 %ELSE
216 0343 2 %IF italian %THEN
217 0344 2     INITIAL(5),
218 0345 2 %ELSE
219 0346 2     INITIAL(5),
220 0347 2 %FI %FI %FI
221 0348 2     PTR;
222 0349 2
223 0350 2     FS_INIT (SAVE_CHARS);    ! Make sure this is initially zero
224 0351 2
225 0352 2     HOLD_MRA = .MRA;        ! Preserve current status

```

```

: 226 0353 2 HOLD TSF = .TSF;
: 227 0354 2 FS_INIT (FRA); ! Initialize output buffer.
: 228 0355 2
: 229 0356 2 ! Get to top of page if not the very first line to be output.
: 230 0357 2 IF
: 231 0358 2 (IF .PHAN_SIMULATE
: 232 0359 2 THEN
: 233 0360 2 (.PHAN_BOTTOM MOD .PHAN_PLINES) NEQ 0
: 234 0361 2 ELSE
: 235 0362 2 (.PHAN_BOTTOM MOD .PHAN_SLINES) NEQ 0)
: 236 0363 2 THEN
: 237 0364 2 ! Not already at top of page, so need a form feed.
: 238 0365 2 PHAN_FORM_PEND = FORM_FEED
: 239 0366 2 ELSE
: 240 0367 2 PHAN_FORM_PEND = 0;
: 241 0368 2
: 242 0369 2 ! If not top of first page advance page number and update page layout.
: 243 0370 2 IF NOT .PHAN_TOP_FIRST
: 244 0371 2 THEN
: 245 0372 2 BEGIN ! Advance the page number.
: 246 0373 2
: 247 0374 2 ! Copy next page number to current page number
: 248 0375 2 BEGIN
: 249 0376 2 MAP ! Make these structures
: 250 0377 2 PAGEN : VECTOR, ! vectors for easier
: 251 0378 2 NPAGEN : VECTOR; ! copying.
: 252 0379 2
: 253 0380 2 INCR I FROM 0 TO (PAGE_SCT_SIZE - 1) DO
: 254 0381 2 PAGEN [I] = .NPAGEN [I];
: 255 0382 2
: 256 0383 2 END;
: 257 0384 2
: 258 0385 2 ! Next, bump page number of next page.
: 259 0386 2 IF .NPAGEN [SCT_SUB_PAGE] EQL 0
: 260 0387 2 THEN ! Not a subpage
: 261 0388 2 NPAGEN [SCT_PAGE] = .NPAGEN [SCT_PAGE] + 1
: 262 0389 2 ELSE ! It's a subpage
: 263 0390 2 NPAGEN [SCT_SUB_PAGE] = .NPAGEN [SCT_SUB_PAGE] + 1;
: 264 0391 2
: 265 0392 2 ! Bump the running page number of the document
: 266 0393 2 NPAGEN [SCT_RUN_PAGE] = .NPAGEN [SCT_RUN_PAGE] + 1;
: 267 0394 2
: 268 0395 2 ! Update the page layout
: 269 0396 2 HCT_LAYOUT = .HCT_LAYOUT NP;
: 270 0397 2 HCT_LAYOUTN = .HCT_LAYOUTN NP;
: 271 0398 2 HCT_NUMBER_PAGE = .HCT_NMPG_NP;
: 272 0399 2 END;
: 273 0400 2
: 274 0401 2 ! If generating a table of contents, signal a new page by writing the
: 275 0402 2 ! number of the page being started.
: 276 0403 2 IF .GCA_CMD_BTC
: 277 0404 2 THEN
: 278 0405 2 ! Yup, user does want a binary table of contents,
: 279 0406 2 ! so save this page number.
: 280 0407 2
: 281 0408 2 ! The page number is written regardless of whether the user said
: 282 0409 2 ! .ENABLE TOC or .DISABLE TOC, as long as he has said /CONTENTS. This

```

B
C
D
E
F
G
H
I
J
K
L
M
N
O
P
Q
R
S
T
U
V
W
X
Y
Z
[
,
-
.
/


```
: 283      0410      2      ! is done so that page numbers are current in the .BTC file so the user
: 284      0411      2      ! can .ENABLE the TOC in the middle of a page.
: 285      0412      2
: 286      U 0413      2      %IF FLIP %THEN
: 287      U 0414      2      PUTTPG (PAGEN, FLIP%K_NEWPAG);
: 288      0415      2      %ELSE
: 289      0416      2      PUTTPG (PAGEN, -1);
: 290      0417      2      %FI
: 291      0418      2
: 292      0419      2      ! Update page count.
: 293      0420      2      IF NOT .GCA_SKIP_OUT AND
: 294      0421      2      NOT .PHAN_TOP_FIRST
: 295      0422      2      THEN
: 296      0423      2      ! Count pages only if not skipping output.
: 297      0424      2      GCA_PAGE_CNT = .GCA_PAGE_CNT + 1;
: 298      0425      2
: 299      0426      2      ! It's important to reset PHAN_TOP_PAGE and PHAN_LINES_TP now, because
: 300      0427      2      ! otherwise, if LOUT gets called to output a title, LOUT will call NEWPAG
: 301      0428      2      ! which will call this routine, which will call LOUT again, etc, in a loop.
: 302      0429      2      ! That's because LOUT does not know who is calling it, would think that
: 303      0430      2      ! whoever called it is trying to output lines at the top of the page, and
: 304      0431      2      ! so would try to get NEWPAG to put in the titles first, thereby leading to
: 305      0432      2      ! an incessant (recursive) loop.
: 306      0433      2      PHAN_TOP_PAGE = FALSE;
: 307      0434      2      PHAN_LINES_TP = 0;
: 308      0435      2      ! This page is officially empty.
: 309      0436      2      ! Compute 'ODD' or 'EVEN'-ness of this page whose titles, etc, are about to
: 310      0437      2      ! be output.
: 311      0438      2      HCT_ODD_EVEN = NOT .HCT_ODD_EVEN;
: 312      0439      2
: 313      0440      2      ! We are now just about to start processing a new page. If we were
: 314      0441      2      ! previously skipping because the user specified a starting page, see if
: 315      0442      2      ! this is it.
: 316      0443      2
: 317      0444      2      ! If there is a list of pages to be output, and output is
: 318      0445      2      ! currently being skipped, see if this page begins a new
: 319      0446      2      ! output sequence.
: 320      0447      2      IF .GCA_SKIP_OUT
: 321      0448      2      THEN
: 322      0449      2      IF NOT .GCA_CMD_QUICK ! Don't reset output-skipping state if user
: 323      0450      2      THEN ! said /QUICK (that is, continue to skip).
: 324      0451      2      GCA_SKIP_OUT = NOT
: 325      0452      2      (PAGFND (PAGEN, SPAGER, .GCA_ORANGE_CNT, FALSE) NEQ 0);
: 326      0453      2
: 327      0454      2      !
: 328      0455      2      ! Process the /DOWN switch.
: 329      0456      2
: 330      0457      2
: 331      0458      2      ! Turn on /down flag so as to not count lines when skipping.
: 332      0459      2
: 333      0460      2      gca_down_flag = true;
: 334      0461      2
: 335      0462      2      IF .phan_down GTR 0
: 336      0463      2      THEN
: 337      0464      2      no_bar_skip (.phan_down);
: 338      0465      2
: 339      0466      2      ! Turn off the /down flag for skipping.
```

```
340 0467 2      !
341 0468 2      gca_down_flag = false;
342 0469 2
343 0470 2      IF .hct_headers
344 0471 2      THEN
345 0472 2      !
346 0473 2      ! Output the title.
347 0474 2      !
348 0475 3      BEGIN
349 0476 3
350 0477 3      IF .PHAN_HEADER
351 0478 3      THEN
352 0479 4          BEGIN          ! This page should have a header.
353 0480 4          LOCAL
354 0481 4              HOLD_PAGING;
355 0482 4
356 0483 4          ! Temporarily turn off paging to avoid recursive interactions
357 0484 4          ! between LOUT and NEWPAG.
358 0485 4          HOLD_PAGING = .PHAN_PAGING;
359 0486 4          PHAN_PAGING = FALSE;
360 0487 4          ! Switch text pointers to saved title
361 0488 4          TSF = TITTSF;
362 0489 4          MRA = TITMRA;
363 0490 4          ! Save current status of title. The following code works by
364 0491 4          ! simply adding additional characters after the title, directly
365 0492 4          ! in the title buffer. The title status is restored later,
366 0493 4          ! so the title can be reused on subsequent pages.
367 0494 4          HOLD_TSF_INT_HL = .TSF_INT_HL;          ! Save current text status.
368 0495 4          HOLD_TSF_EXT_HL = .TSF_EXT_HL;
369 0496 4          HOLD_TSF_ADJUST = .TSF_ADJUST;
370 0497 4          HOLD_MRA_NEXT = .FS_NEXT (MRA);
371 0498 4          HOLD_MRA_LENGTH = .FS_LENGTH (MRA);
372 0499 4
373 0500 4      IF .HCT_NUMBER PAGE
374 0501 6          AND (.HCT_LAYOUT EQL LAYOUT_STANDARD)
375 0502 5          OR (.HCT_LAYOUT EQL LAYOUT_ROM_BOTC)
376 0503 4      THEN
377 0504 4          ! Put page number at end of title line.
378 0505 5          BEGIN
379 0506 5          PTR = CHSPTR (WORK AREA);
380 0507 5          ! Convert page number and get length.
381 0508 5          PAGE_LENGTH = PACPAG (PAGE, PTR);
382 0509 5          FILL_COUNT = (.GCA_LWIDTH - .TSF_EXT_HL) -
383 0510 5              (.PAGE_LENGTH + .page_sfring_length);
384 0511 5
385 0512 5          IF .FILL_COUNT LSS 0 OR
386 0513 6              ((.FS_MAXSIZE (MRA) - .FS_LENGTH (MRA)) LSS .FILL_COUNT)
387 0514 5          THEN
388 0515 5              ! No room at end of line for page number.
389 0516 5              ERM (RNFPWF, 0, 0)
390 0517 5          ELSE
391 0518 5              ! Put page number at end of title.
392 0519 6              BEGIN
393 0520 6
394 0521 6              INCR I FROM 1 TO .FILL_COUNT DO
395 0522 6                  FS_WCHAR (MRA, %C'-'); ! Position out with spaces.
396 0523 6
```

```

397      0524 6      ! Get 'page' in proper case.
398      0525 7      PTR = CH$PTR ((CASE .HCT_HD_CASE FROM 0 TO 2 OF
399      0526 7      SEI
400      U 0527 7 %IF german %THEN
401      UU 0528 7      [HCT_HD_LOWER] : UPLIT ('seite ');
402      UU 0529 7      [HCT_HD_UPPER] : UPLIT ('SEITE ');
403      U 0530 7      [HCT_HD_MIXED] : UPLIT ('Seite ');
404      0531 7 %ELSE
405      U 0532 7 %IF french %THEN
406      UU 0533 7      [HCT_HD_LOWER] : UPLIT ('page ');
407      UU 0534 7      [HCT_HD_UPPER] : UPLIT ('PAGE ');
408      U 0535 7      [HCT_HD_MIXED] : UPLIT ('Page ');
409      0536 7 %ELSE
410      U 0537 7 %IF italian %THEN
411      UU 0538 7      [HCT_HD_LOWER] : UPLIT ('page ');
412      UU 0539 7      [HCT_HD_UPPER] : UPLIT ('PAGE ');
413      U 0540 7      [HCT_HD_MIXED] : UPLIT ('Page ');
414      0541 7 %ELSE
415      0542 7      [HCT_HD_LOWER] : UPLIT ('page ');
416      0543 7      [HCT_HD_UPPER] : UPLIT ('PAGE ');
417      0544 7      [HCT_HD_MIXED] : UPLIT ('Page ');
418      0545 7 %FI %FI %FI
419      0546 6      TES));
420      0547 6
421      0548 6      ! Put word 'page' at end of line.
422      0549 6      INCR I FROM 1 TO .page_string_length DO
423      0550 6      FS_WCHAR (MRA, CH$RCHAR_A (PTR));
424      0551 6
425      0552 6      PTR = CH$PTR (WORK_AREA);
426      0553 6
427      0554 6      ! Output converted page number.
428      0555 6      INCR I FROM 1 TO .PAGE_LENGTH DO
429      0556 6      FS_WCHAR (MRA, CH$RCHAR_A (PTR));
430      0557 6
431      0558 6      ! Count characters added.
432      0559 6      TSF_INT_HL = .TSF_INT_HL + .FILL_COUNT +
433      0560 6      .PAGE_LENGTH + .page_string_length;
434      0561 6      TSF_EXT_HL = .TSF_EXT_HL + .FILL_COUNT +
435      0562 6      .PAGE_LENGTH + .page_string_length;
436      0563 5      END;
437      0564 5
438      0565 4      END;
439      0566 4
440      0567 4      IF .TSF_INT_HL NEQ 0
441      0568 4      THEN
442      0569 4      ! There is something to output
443      0570 5      BEGIN
444      0571 5
445      0572 5      ! If the user wants things centered, do it.
446      0573 5      IF .HCT_LAYOUT EQL LAYOUT_CENTERED
447      0574 5      THEN
448      0575 5      TSF_ADJUST = (.GCA_LWIDTH - .TSF_EXT_HL)/2
449      0576 5      ELSE
450      0577 5
451      0578 5      ! See if user wants the title/subtitle flipped
452      0579 5      IF (.HCT_LAYOUT EQL LAYOUT_FLIP_ODD) AND
453      0580 5      .HCT_ODD_EVEN

```

```

454 0581 5      THEN
455 0582 5          TSF_ADJUST = .GCA_LWIDTH - .TSF_EXT_HL;
456 0583 5      LOUT ();          ! Output title line.
457 0584 5      TSF_INT_HL = .HOLD_TSF_INT_HL;    ! Restore title to its previous status.
458 0585 5      TSF_EXT_HL = .HOLD_TSF_EXT_HL;
459 0586 5      TSF_ADJUST = .HOLD_TSF_ADJUST;
460 0587 5      FS_NEXT (MRA) = .HOLD_MRA_NEXT;
461 0588 5      FS_LENGTH (MRA) = .HOLD_MRA_LENGTH;
462 0589 5      END
463 0590 4      ELSE
464 0591 4          ! Even if there is no title reserve the title line so
465 0592 4          ! that all pages have the same layout.
466 0593 4      NO_BAR_SKIP (1);          ! End of title processing.
467 0594 4
468 0595 4      !+
469 0596 4      !- Output the subtitle.
470 0597 4
471 0598 4      IF .HCT_SUBTITLE
472 0599 4      THEN
473 0600 5          BEGIN
474 0601 5          LOCAL
475 0602 5              ! Where (what column) to truncate the subtitle.
476 0603 5              TRUNCATE_POINT;
477 0604 5
478 0605 5          TRUNCATE_POINT = 0;
479 0606 5          FS_INIT (FRA);
480 0607 5          FS_INIT (ADD_CHARS);
481 0608 5          FS_INIT (SAVE_CHARS);
482 0609 5          ! Set up pointers to subtitle and save status.
483 0610 5          TSF = SBTTSF;
484 0611 5          MRA = SBTMRA;
485 0612 5          HOLD_TSF_EXT_HL = .TSF_EXT_HL;
486 0613 5          HOLD_TSF_INT_HL = .TSF_INT_HL;
487 0614 5          HOLD_TSF_ADJUST = .TSF_ADJUST;
488 0615 5          HOLD_MRA_NEXT = .FS_NEXT (MRA);
489 0616 5          HOLD_MRA_LENGTH = .FS_LENGTH (MRA);
490 0617 5
491 0618 5          ! See if the user said .DATE.
492 0619 5          IF .HCT_DATE
493 0620 7              AND (.HCT_LAYOUT EQL LAYOUT_STANDARD)
494 0621 6              OR (.HCT_LAYOUT EQL LAYOUT_RUN_BOTC)
495 0622 5          THEN
496 0623 6          BEGIN
497 0624 6              ! Get date information in work area.
498 0625 6              PTR = CHSPTR (WORK_AREA);
499 0626 6              CNVDAT(DATTIM,PTR,DATE_LENGTH);
500 0627 6
501 0628 6              ! See if date will fit.
502 0629 6              ! If not, write "... <date>" to the ADD_CHARS buffer.
503 0630 7              IF .TSF_EXT_HL GTR (.GCA_LWIDTH - .DATE_LENGTH)
504 0631 6              THEN
505 0632 7                  BEGIN
506 0633 7                      FS_WCHAR (ADD_CHARS, %C' ');
507 0634 7
508 0635 7                      INCR I FROM 1 TO 3 DO
509 0636 7                          FS_WCHAR (ADD_CHARS, %C'.');
510 0637 7

```

```
511 0638 7 FS_WCHAR (ADD_CHARS, %C' ');
512 0639 7 PTR = CH$PTR (WORK_AREA);
513 0640 7
514 0641 7 INCR I FROM 1 TO .DATE_LENGTH DO
515 0642 7 FS_WCHAR (ADD_CHARS, CH$RCHAR_A (PTR));
516 0643 7
517 0644 6 END;
518 0645 6 ELSE END
519 0646 5
520 0647 5
521 0648 5 ! No date is called for. Check if subtitle will fit.
522 0649 5 ! If not, write "... " to the ADD_CHARS buffer.
523 0650 5 IF .TSF_EXT_HL GTR .GCA_LWIDTH
524 0651 5 THEN
525 0652 6 BEGIN
526 0653 6 FS_WCHAR (ADD_CHARS, %C' ');
527 0654 6
528 0655 6 INCR I FROM 1 TO 3 DO
529 0656 6 FS_WCHAR (ADD_CHARS, %C'.');
530 0657 6
531 0658 5 END;
532 0659 5
533 0660 5 ! Is truncation needed? If so, ADD_CHARS is nonzero.
534 0661 6 IF ((.FS_LENGTH (ADD_CHARS) EQL 0) OR
535 0662 6 (.FS_LENGTH (ADD_CHARS) GEQ .GCA_LWIDTH))
536 0663 5 THEN
537 0664 5 ! Subtitle will fit as is. Add padding and date if
538 0665 5 ! the user said .DATE.
539 0666 6 BEGIN
540 0667 6
541 0668 6 IF .HCT DATE
542 0669 8 AND (.HCT_LAYOUT EQL LAYOUT_STANDARD)
543 0670 7 OR (.HCT_LAYOUT EQL LAYOUT_RUN_BOTC))
544 0671 6 THEN
545 0672 7 BEGIN
546 0673 7 LOCAL
547 0674 7 FILL_COUNT;
548 0675 7
549 0676 7 ! Calculate the required padding before the date.
550 0677 7 FILL_COUNT = .GCA_LWIDTH - .TSF_EXT_HL - .DATE_LENGTH;
551 0678 7
552 0679 7 ! Put in the padding.
553 0680 7 INCR I FROM 1 TO .FILL_COUNT DO
554 0681 7 FS_WCHAR (MRA, %C'-');
555 0682 7
556 0683 7 ! Copy the date.
557 0684 7 PTR = CH$PTR (WORK_AREA);
558 0685 7
559 0686 7 INCR I FROM 1 TO .DATE_LENGTH DO
560 0687 7 FS_WCHAR(MRA, CH$RCHAR_A(PTR));
561 0688 7
562 0689 7 ! Update the appropriate counts.
563 0690 7 TSF_INT_HL = .TSF_INT_HL + .FILL_COUNT + .DATE_LENGTH;
564 0691 7 TSF_EXT_HL = .TSF_EXT_HL + .FILL_COUNT + .DATE_LENGTH;
565 0692 6 END;
566 0693 6 ELSE END
567 0694 5
```

```

568 0695 5 ! Subtitle must be truncated. We must add something;
569 0696 5 ! either "... " or "...<date>".
570 0697 6 BEGIN
571 0698 6 TRUNCATE_POINT = .GCA_LWIDTH - .FS_LENGTH (ADD_CHARS);
572 0699 6
573 0700 6 ! Now actually truncate the subtitle MRA/TSF.
574 0701 6 TRUNCATE (.TRUNCATE_POINT);
575 0702 6 ! Save the characters about to be overwritten on the MRA.
576 0703 6 PTR = .FS_NEXT (MRA);
577 0704 6
578 0705 6 INCR I FROM 1 TO .FS_LENGTH (ADD_CHARS) DO
579 0706 6     FS_WCHAR (SAVE_CHARS, CHRCHAR_A (PTR));
580 0707 6
581 0708 6 ! Copy the truncation text to the MRA and update pointers.
582 0709 6 PTR = .FS_START (ADD_CHARS);
583 0710 6
584 0711 6 INCR I FROM 1 TO .FS_LENGTH (ADD_CHARS) DO
585 0712 6     FS_WCHAR (MRA, CHRCHAR_A (PTR));
586 0713 6
587 0714 6 TSF_INT_HL = .TSF_INT_HL + .FS_LENGTH (ADD_CHARS);
588 0715 6 TSF_EXT_HL = .TSF_EXT_HL + .FS_LENGTH (ADD_CHARS);
589 0716 6 END;
590 0717 5
591 0718 5 ! Put out the subtitle.
592 0719 5 IF .TSF_INT_HL NEQ 0
593 0720 5 THEN
594 0721 6 BEGIN
595 0722 6
596 0723 6 ! Center the subtitle if the user wants it centered.
597 0724 6 IF .HCT_LAYOUT EQL LAYOUT_CENTERED
598 0725 6 THEN
599 0726 6     TSF_ADJUST = (.GCA_LWIDTH - .TSF_EXT_HL)/2
600 0727 6 ELSE
601 0728 6
602 0729 6 ! See if user wants subtitle flipped.
603 0730 6 IF (.HCT_LAYOUT EQL LAYOUT_FLIP_ODD)
604 0731 6 AND .HCT_ODD_EVEN ! Tests TRUE if odd
605 0732 6 THEN
606 0733 6     TSF_ADJUST = .GCA_LWIDTH - .TSF_EXT_HL;
607 0734 6
608 0735 6 ! Output the subtitle.
609 0736 6 LOUT ();
610 0737 6
611 0738 6 ! If we truncated the MRA, then restore its saved text.
612 0739 6 IF .FS_LENGTH (SAVE_CHARS) GTR 0
613 0740 6 THEN
614 0741 7 BEGIN
615 0742 7 ! Truncate again to make sure we write saved characters
616 0743 7 ! back to the same place they were overwritten from.
617 0744 7 TRUNCATE (.TRUNCATE_POINT);
618 0745 7 PTR = .FS_START (SAVE_CHARS);
619 0746 7
620 0747 7 INCR I FROM 1 TO .FS_LENGTH (SAVE_CHARS) DO
621 0748 7     FS_WCHAR (MRA, CHRCHAR_A (PTR));
622 0749 7
623 0750 6 END;
624 0751 6 END

```

```

625      0752      5      ELSE
626      0753      5      ! If the user said .SUBTITLE but the subtitle is empty
627      0754      5      ! reserve the line so all pages have the same layout.
628      0755      5      NO_BAR_SKIP (1);
629      0756      5
630      0757      5      ! Restore subtitle status
631      0758      5      TSF_INT_HL = .HOLD_TSF_INT_HL;
632      0759      5      TSF_EXT_HL = .HOLD_TSF_EXT_HL;
633      0760      5      TSF_ADJUST = .HOLD_TSF_ADJUST;
634      0761      5      FS_NEXT (MRA) = .HOLD_MRA_NEXT;
635      0762      5      FS_LENGTH (MRA) = .HOLD_MRA_LENGTH;
636      0763      4      END;
637      0764      4      ! End of subtitle processing
638      0765      4
639      0766      4      ! Restore status of paging.
640      0767      4      PHAN_PAGING = .HOLD_PAGING
641      0768      4      END
642      0769      3      ELSE
643      0770      3      ! Even if there was no header on this page, still reserve the
644      0771      3      ! lines so that all pages have the same layout.
645      0772      3      IF .HCT_SUBTITLE
646      0773      3      THEN
647      0774      4      NO_BAR_SKIP (2)
648      0775      3      ELSE
649      0776      3      NO_BAR_SKIP (1);
650      0777      3
651      0778      3      NO_BAR_SKIP (2);
652      0779      2      END;
653      0780      2      ! Generate two blank lines after titles.
654      0781      2      ! End processing of headers.
655      0782      2      !+
656      0783      2      Output topnotes.
657      U 0784      2      %IF DSRPLUS %THEN
658      UU 0785      2      !
659      UU 0786      2      ! See if topnotes are enabled.
660      UU 0787      2      !
661      UU 0788      2      IF (.tn_internal_enab OR .tn_user_enab)
662      UU 0789      2      AND
663      UU 0790      2      ((.tn_internal_int GTR 0) OR (.tn_user_int GTR 0))
664      UU 0791      2      THEN
665      U 0792      2      tnout ();
666      0793      2      %FI
667      0794      2
668      0795      2      ! The following block generates accumulated figures in a "safe" way. The
669      0796      2      ! need for a "safe" way of generating the accumulated figures was
670      0797      2      ! demonstrated by a file that contained a .figure deferred 58 command. What
671      0798      2      ! happened is that there was an access violation. The reason for that is
672      0799      2      ! that the NO_BAR_SKIP macro generates a call on USKIPL. And USKIPL checks
673      0800      2      ! for the end-of-page happening as it generates lines; if it detects that,
674      0801      2      ! it calls NEWPAG which calls this module. BUT, before this module finishes
675      0802      2      ! it generates the deferred figures, thereby leading to a loop that finishes
676      0803      2      ! only when the stack overflows. The fix is to clear the deferred figure
677      0804      2      ! count before issuing the macro. Then when we get called again, we will not
678      0805      2      ! see any more figures to be generated, and the loop will go away.
679      0806      3      BEGIN
680      0807      3      LOCAL
681      0808      3      X;

```

```

: 682 0809 3
: 683 0810 3 X = .PHAN FIGURE;
: 684 0811 3 PHAN FIGURE = 0;
: 685 0812 3 NO BAR_SKIP (.X);
: 686 0813 2 END;
: 687 0814 2
: 688 0815 2 NO BAR_SKIP (.PHAN FIGURE);
: 689 0816 2 PHAN_TOP_FIRST = FALSE;
: 690 0817 2 PHAN_HEADER = TRUE;
: 691 0818 2 MRA = .HOLD_MRA;
: 692 0819 2 TSF = .HOLD_TSF;
: 693 0820 2
: 694 0821 1 END;

```

```

! Generate deferred figures.
! No longer at top of first page.
! Enable headers for subsequent pages.
! Switch back to the usual
! text buffers.
! End of TOPPAG

```

```

.TITLE TOPPAG Top of the page processing
.IDENT \V04-000\

```

```

.PSECT $SPLITS$,NOWRT,NOEXE,2

```

```

00 00 00 20 65 67 61 70 00000 P.AAA: .ASCII \page \<0><0><0>
00 00 00 20 45 47 41 50 00008 P.AAB: .ASCII \PAGE \<0><0><0>
00 00 00 20 65 67 61 50 00010 P.AAC: .ASCII \Page \<0><0><0>

```

```

.PSECT $OWNS$,NOEXE,2

```

```

00000000 00000032 00000000 00000000 00000 ADD_CHARS:
                                .LONG 0, 0, 50, 0
00010 .BLKB 50
00042 .BLKB 2
00000000 00000032 00000000 00000000 00044 SAVE_CHARS:
                                .LONG 0, 0, 50, 0
00054 .BLKB 50

```

```

.EXTRN DATTIM, FRA, GCA
.EXTRN HCT, MRA, NPAGEN
.EXTRN PAGEN, SCA, SPAGER
.EXTRN TPAGER, PHAN, SBTMRA
.EXTRN SBTTSF, TITMRA, TITTSF
.EXTRN TSF, RNFPWF, RINTES
.EXTRN CNVDAT, ERM, ERML
.EXTRN ERMN, LOUT, PACPAG
.EXTRN PAGFND, PUTTPG, USKIPL

```

```

.PSECT $CODE$,NOWRT,2

```

```

                                OFFC 00000
                                .ENTRY TOPPAG, Save R2,R3,R4,R5,R6,R7,R8,R9,R10,- : 0297
                                R11
                                SE          CO      AE 9E 00002 MOVAB -64(SP), SP
                                56          05 00 00006 MOVL #5, PAGE_STRING_LENGTH
                                00000000' EF D4 00009 CLRL SAVE_CHARS+12
                                00000000' EF 9E 0000F MOVAB SAVE_CHARS+16, SAVE_CHARS
                                00000000' EF D0 0001A MOVL SAVE_CHARS, SAVE_CHARS+4
                                00000000G EF DD 00025 PUSHL MRA
                                00000000G EF DD 0002B PUSHL TSF
                                00000000G EF D4 00031 CLRL FRA+12
                                00000000G EF 9E 00037 MOVAB FRA+16, FRA

```


		00000000G	EF	00000000G	EF	D0	00042	MOVL	FRA, FRA+4			
			14	00000000G	EF	E9	0004D	BLBC	PHAN+52, 1\$		0358	
7E		00 00000000G	EF		01	7A	00054	EMUL	#1, PHAN+56, #0, -(SP)		0360	
50		50	8E	00000000G	EF	7B	0005D	EDIV	PHAN+8, (SP)+, R0, R0			
					12	11	00066	BRB	2\$			
7E		00 00000000G	EF		01	7A	00068	1\$: EMUL	#1, PHAN+56, #0, -(SP)		0362	
50		50	8E	00000000G	EF	7B	00071	EDIV	PHAN+36, (SP)+, R0, R0			
					50	D5	0007A	2\$: TSTL	R0			
		00000000G	EF		09	13	0007C	BEQL	3\$			
					0C	D0	0007E	MOVL	#12, PHAN+32		0365	
					06	11	00085	BRB	4\$			
					EF	D4	00087	3\$: CLRL	PHAN+32		0367	
		45 00000000G	EF		EF	E8	0008D	4\$: BLBS	PHAN+24, 8\$		0370	
					50	D4	00094	CLRL	I		0380	
		00000000G	EF		40	D0	00096	5\$: MOVL	NPAGEN[I], PAGEN[I]		0381	
		EF 50			03	F3	000A3	AOBLEQ	#3, I, 5\$			
					EF	B5	000A7	TSTW	NPAGEN+2		0386	
					08	12	000AD	BNEQ	6\$			
					EF	D6	000AF	INCL	NPAGEN+8		0388	
					06	11	000B5	BRB	7\$			
					EF	B6	000B7	6\$: INCW	NPAGEN+2		0390	
					EF	B6	000BD	7\$: INCW	NPAGEN+14		0393	
		00000000G	EF		EF	7D	000C3	MOVQ	HCT+36, HCT+28		0396	
		00000000G	FF		FF	D0	000CE	MOVL	@HCT+44, @HCT+12		0398	
		10 00000000G	EF		01	E1	000D9	8\$: BBC	#1, GCA+124, 9\$		0403	
			7E		01	CE	000E1	MNEGL	#1, -(SP)		0416	
					EF	9F	000E4	PUSHAB	PAGEN			
		00000000G	EF		02	FB	000EA	CALLS	#2, PUTTPG			
		0D 00000000G	EF		EF	E8	000F1	9\$: BLBS	GCA+112, 10\$		0420	
		06 00000000G	EF		EF	E8	000F8	BLBS	PHAN+24, 10\$		0421	
					EF	D6	000FF	INCL	GCA+88		0424	
					EF	D4	00105	10\$: CLRL	PHAN		0433	
					EF	D4	0010B	CLRL	PHAN+12		0434	
		00000000G	EF		EF	D2	00111	MCOML	HCT+16, HCT+16		0438	
		31 00000000G	EF		EF	E9	0011C	BLBC	GCA+112, 12\$		0447	
		2A 00000000G	EF		EF	E8	00123	BLBS	GCA+208, 12\$		0449	
					7E	D4	0012A	CLRL	-(SP)		0452	
					EF	DD	0012C	PUSHL	GCA+108			
					EF	9F	00132	PUSHAB	SPAGER			
					EF	9F	00138	PUSHAB	PAGEN			
		00000000G	EF		04	FB	0013E	CALLS	#4, PAGFND			
					51	D4	00145	CLRL	R1			
					50	D5	00147	TSTL	R0			
					02	13	00149	BEQL	11\$			
					51	D6	0014B	INCL	R1			
		00000000G	EF		51	D2	0014D	11\$: MCOML	R1, GCA+112		0451	
		00000000G	EF		01	88	00154	12\$: BISB2	#1, GCA+180		0460	
					EF	D0	0015B	MOVL	PHAN+48, R1		0462	
					27	15	00162	BLEQ	13\$			
					EF	D0	00164	MOVL	TSF, R0		0464	
52		7C A0			00	EF	0016B	EXTZV	#0, #1, 124(R0), HOLD_BARS			
					7C A0	01	8A	00171	BICB2	#1, 124(R0)		
					51	DD	00175	PUSHL	R1			
		00000000G	EF		01	FB	00177	CALLS	#1, USKIPL			
					50	D0	0017E	MOVL	TSF, R0			
7C	A0	01			52	F0	00185	INSV	HOLD_BARS, #0, #1, 124(R0)			
		00000000G	EF		01	8A	0018B	13\$: BICB2	#1, GCA+180		0468	

	03	00000000G	FF	E8	00192	BLBS	@HCT+8, 14\$	0470			
			056F	31	00199	BRW	71\$				
	03	00000000G	FF	E8	0019C	14\$:	BLBS	@PHAN+28, 15\$	0477		
			0502	31	001A3	BRW	67\$				
	08	AE	00000000G	FF	D0	001A6	15\$:	MOVL	@PHAN+40, HOLD_PAGING	0485	
			00000000G	FF	D4	001AE	CLRL	@PHAN+40	0486		
	00000000G	EF	00000000G	EF	9E	001B4	MOVAB	TITTSF, TSF	0488		
	00000000G	EF	00000000G	EF	9E	001BF	MOVAB	TITMRA, MRA	0489		
	50	00000000G	EF	D0	001CA	MOVL	TSF, R0	0494			
	57		60	7D	001D1	MOVQ	(R0), HOLD_TSF_INT_HL				
	59	28	A0	D0	001D4	MOVL	40(R0), HOLD_TSF_ADJUST	0496			
	50	00000000G	EF	D0	001D8	MOVL	MRA, R0	0497			
	5A	04	A0	D0	001DF	MOVL	4(R0), HOLD_MRA_NEXT				
	5B	0C	A0	D0	001E3	MOVL	12(R0), HOLD_MRA_LENGTH	0498			
	61	00000000G	FF	E9	001E7	BLBC	@HCT+12, 18\$	0500			
	50	00000000G	EF	D0	001EE	MOVL	HCT+28, R0	0501			
			05	13	001F5	BEQL	16\$				
	03		50	D1	001F7	CPL	R0, #3	0502			
			53	12	001FA	BNEQ	18\$				
	10	AE	14	AE	9E	001FC	16\$:	MOVAB	WORK_AREA, PTR	0506	
			10	AE	9F	00201	PUSHAB	PTR	0508		
		00000000G	EF	9F	00204	PUSHAB	PAGEN				
	00000000G	EF	02	FB	0020A	CALLS	#2, PACPAG				
	54		50	D0	00211	MOVL	R0, PAGE_LENGTH				
	52	00000000G	EF	D0	00214	MOVL	TSF, R2	0509			
	51	00000000G	FF	A2	C3	0021B	SUBL3	4(R2), @GCA+140, R1			
	50		54	C1	00224	ADDL3	PAGE_STRING_LENGTH, PAGE_LENGTH, R0	0510			
	55		51	C3	00228	SUBL3	R0, R1, FILE_COUNT				
			12	19	0022C	BLSS	17\$	0512			
	50	08	50	00000000G	EF	D0	0022E	MOVL	MRA, R0	0513	
			A0	C3	00235	SUBL3	12(R0), 8(R0), R0				
			55	D1	0023B	CPL	R0, FILL_COUNT				
				12	18	0023E	BGEQ	19\$			
				7E	7C	00240	17\$:	CLRQ	-(SP)	0516	
		00000000G	EF	8F	DD	00242	PUSHL	#RNFPPWF			
			03	FB	00248	CALLS	#3, ERM				
			0094	31	0024F	18\$:	BRW	31\$			
			50	00000000G	EF	D0	00252	19\$:	MOVL	MRA, R0	0522
			50	04	C0	00259	ADDL2	#4, R0			
			51	00000000G	EF	D0	0025C	MOVL	MRA, R1		
			>1	0C	C0	00263	ADDL2	#12, R1			
				53	D4	00266	CLRL	I			
				08	11	00268	BRB	21\$			
			00	B0	20	90	0026A	20\$:	MOVAB	#32, @0(R0)	
				60	D6	0026E	INCL	(R0)			
				61	D6	C0270	INCL	(R1)			
	F4		53	00000000G	55	F3	00272	21\$:	AOBLEQ	FILL_COUNT, I, 20\$	0521
	02		00	00000000G	FF	CF	00276	CASEL	@HCT+20, #0, #2	0546	
	0018		000F	0006	0027E	22\$:	.WORD	23\$-22\$,-			
								24\$-22\$,-			
								25\$-22\$			
			53	00000000'	EF	9E	00284	23\$:	MOVAB	P.AAA, R3	0542
					10	11	0028B	BRB	26\$	0546	
			53	00000000'	EF	9E	0028D	24\$:	MOVAB	P.AAB, R3	0543
					07	11	00294	BRB	26\$	0546	
			53	00000000'	EF	9E	00296	25\$:	MOVAB	P.AAC, R3	0544
	10	AE	53	D0	0029D	26\$:	MOVL	R3, PTR	0546		

				53	D4	002A1		CLRL	I		0549	
				0C	11	002A3		BRB	28\$			
		00	B0	10	BE	90	002A5	27\$:	MOVB	@PTR, @0(R0)	0550	
				10	AE	D6	002AA		INCL	PTR		
				60	D6	002AD		INCL	(R0)			
				61	D6	002AF		INCL	(R1)			
	F0			56	F3	002B1		28\$:	AOBLEQ	PAGE_STRING_LENGTH, I, 27\$	0549	
		10	AE	14	AE	9E	002B5		MOVAB	WORK_AREA, PTR	0552	
				53	D4	002BA			CLRL	I	0555	
				0C	11	002BC			BRB	30\$		
		00	B0	10	BE	90	002BE	29\$:	MOVB	@PTR, @0(R0)	0556	
				10	AE	D6	002C3		INCL	PTR		
				60	D6	002C6		INCL	(R0)			
				61	D6	002C8		INCL	(R1)			
	F0			54	F3	002CA		30\$:	AOBLEQ	PAGE_LENGTH, I, 29\$	0555	
	50			55	C1	002CE			ADDL3	FILL_COUNT, (R2), R0	0559	
				50	C0	002D2			ADDL2	PAGE_LENGTH, R0		
	62			56	C1	002D5			ADDL3	PAGE_STRING_LENGTH, R0, (R2)	0560	
	50			55	A2	002D9	04		ADDL3	4(R2), FILL_COUNT, R0	0561	
				50	C0	002DE			ADDL2	PAGE_LENGTH, R0		
04	A2			56	C1	002E1			ADDL3	PAGE_STRING_LENGTH, R0, 4(R2)	0562	
				50	00000000G	EF	D0	002E6	31\$:	MOVL	TSF, R0	0567
				60	D5	002ED			TSTL	(R0)		
				59	13	002EF			BEQL	34\$		
				01	00000000G	EF	D1	002F1		CMPL	HCT+28, #1	0573
				10	12	002F8			BNEQ	32\$		
28	51	00000000G	FF	04	A0	C3	002FA		SUBL3	4(R0), @GCA+140, R1	0575	
	A0		51	02	C7	00303			DIVL3	#2, R1, 40(R0)		
				1A	11	00308			BRB	33\$		
				02	00000000G	EF	D1	0030A	32\$:	CMPL	HCT+28, #2	0579
				11	12	00311			BNEQ	33\$		
				0A	00000000G	EF	E9	00313		BLBC	HCT+16, 33\$	0580
28	A0	00000000G	FF	04	A0	C3	0031A		SUBL3	4(R0), @GCA+140, 40(R0)	0582	
		00000000G	EF		00	FB	00324	33\$:	CALLS	#0, LOUT	0583	
				50	00000000G	EF	D0	0032B		MOVL	TSF, R0	
				60	57	7D	00332		MOVQ	HOLD_TSF_INT_HL, (R0)	0584	
	28			59	D0	00335			MOVL	HOLD_TSF_ADJUST, 40(R0)	0586	
				50	00000000G	EF	D0	00339		MOVL	MRA, R0	0587
				04	A0	D0	00340		MOVL	HOLD_MRA_NEXT, 4(R0)		
				0C	A0	D0	00344		MOVL	HOLD_MRA_LENGTH, 12(R0)	0588	
				20	11	00348			BRB	35\$	0567	
52	7C	A0		00	EF	0034A		34\$:	EXTZV	#0, #1, 124(R0), HOLD_BARS	0593	
				7C	A0	01	8A	00350		BICB2	#1, 124(R0)	
				01	0D	00354			PUSHL	#1		
				00000000G	EF	01	FB	00356		CALLS	#1, USKIPL	
				50	00000000G	EF	D0	0035D		MOVL	TSF, R0	
				00	52	F0	00364		INSV	HOLD_BARS, #0, #1, 124(R0)		
7C	A0			01	00	F0	00364					
				03	00000000G	FF	E8	0036A	35\$:	BLBS	@HCT+24, 36\$	0598
				032A	31	00371			BRW	66\$		
				55	D4	00374		36\$:	CLRL	TRUNCATE_POINT	0605	
				00000000G	EF	D4	00376		CLRL	FRA+12	0606	
				00000000G	EF	9E	0037C		MOVAB	FRA+16, FRA		
				00000000G	EF	D0	00387		MOVL	FRA, FRA+4		
				00000000'	EF	D4	00392		CLRL	ADD_CHARS+12	0607	
				00000000'	EF	9E	00398		MOVAB	ADD_CHARS+16, ADD_CHARS		
				00000000'	EF	D0	003A3		MOVL	ADD_CHARS, ADD_CHARS+4		
				00000000'	EF	D4	003AE		CLRL	SAVE_CHARS+12	0608	

	00000000'	EF	00000000'	EF	9E	003B4	MOVAB	SAVE_CHARS+16, SAVE_CHARS				
	00000000'	EF	00000000'	EF	D0	003BF	MOVL	SAVE_CHARS, SAVE_CHARS+4				
	00000000G	EF	00000000G	EF	9E	003CA	MOVAB	SBTTSF, TSF		0610		
	00000000G	EF	00000000G	EF	9E	003D5	MOVAB	SBTMRA, MRA		0611		
		50	00000000G	EF	D0	003E0	MOVL	TSF, R0		0612		
		57		60	7D	003E7	MOVQ	(R0), HOLD TSF INT HL		0613		
		59	28	A0	D0	003EA	MOVL	40(R0), HOLD_TSF_ADJUST		0614		
		51	00000000G	EF	D0	003EE	MOVL	MRA, R1		0615		
		5A	04	A1	D0	003F5	MOVL	4(R1), HOLD MRA NEXT				
		5B	0C	A1	D0	003F9	MOVL	12(R1), HOLD_MRA_LENGTH		0616		
		03	00000000G	FF	E8	003FD	BLBS	@HCT+4, 38\$		0619		
				00A6	31	00404	BRW	43\$				
			51	00000000G	EF	D0	00407	38\$:	MOVL	HCT+28, R1	0620	
					05	13	0040E		BEQL	39\$		
			03		51	D1	00410		CMP	R1, #3	0621	
					EF	12	00413		BNEQ	37\$		
	10	AE	14	AE	9E	00415	39\$:	MOVAB	WORK_AREA, PTR	0625		
			0C	AE	9F	0041A		PUSHAB	DATE_LENGTH	0626		
			14	AE	9F	0041D		PUSHAB	PTR			
				00000000G	EF	9F	00420		PUSHAB	DATTIM		
			00000000G	EF	03	FB	00426		CALLS	#3, CNVDAT		
			50	00000000G	EF	D0	0042D		MOVL	TSF, R0	0630	
			52	00000000G	FF	D0	00434		MOVL	@GCA+140, R2		
	51		52	0C	AE	C3	0043B		SUBL3	DATE_LENGTH, R2, R1		
			51	04	A0	D1	00440		CMP	4(R0), R1		
					72	15	00444		BLEQ	44\$		
			00000000'	FF	20	90	00446		MOV	#32, @ADD_CHARS+4	0633	
			00000000'	EF	D6	0044D		INCL	ADD_CHARS+4			
			00000000'	EF	D6	00453		INCL	ADD_CHARS+12			
			50		01	D0	00459		MOVL	#1, I	0635	
			00000000'	FF	2E	90	0045C	40\$:	MOV	#46, @ADD_CHARS+4	0636	
			00000000'	EF	D6	00463		INCL	ADD_CHARS+4			
			00000000'	EF	D6	00469		INCL	ADD_CHARS+12			
	E9		50		03	F3	0046F		AOBLEQ	#3, I, 40\$	0635	
			00000000'	FF	20	90	00473		MOV	#32, @ADD_CHARS+4	0638	
			00000000'	EF	D6	0047A		INCL	ADD_CHARS+4			
			00000000'	EF	D6	00480		INCL	ADD_CHARS+12			
			10	AE	9E	00486		MOVAB	WORK_AREA, PTR	0639		
					50	D4	0048B		CLRL	I	0641	
					17	11	0048D		BRB	42\$		
			00000000'	FF	10	BE	90	0048F	41\$:	MOV	@PTR, @ADD_CHARS+4	0642
					10	AE	D6	00497		INCL	PTR	
			00000000'	EF	D6	0049A		INCL	ADD_CHARS+4			
			00000000'	EF	D6	004A0		INCL	ADD_CHARS+12			
	E4		50	0C	AE	F3	004A6	42\$:	AOBLEQ	DATE_LENGTH, I, 41\$	0641	
					3A	11	004AB		BRB	46\$	0619	
			52	00000000G	FF	D0	004AD	43\$:	MOVL	@GCA+140, R2	0650	
			52	04	A0	D1	004B4		CMP	4(R0), R2		
					2D	15	004B8	44\$:	BLEQ	46\$		
			00000000'	FF	20	90	004BA		MOV	#32, @ADD_CHARS+4	0653	
			00000000'	EF	D6	004C1		INCL	ADD_CHARS+4			
			00000000'	EF	D6	004C7		INCL	ADD_CHARS+12			
			50		01	D0	004CD		MOVL	#1, I	0655	
			00000000'	FF	2E	90	004D0	45\$:	MOV	#46, @ADD_CHARS+4	0656	
			00000000'	EF	D6	004D7		INCL	ADD_CHARS+4			
			00000000'	EF	D6	004DD		INCL	ADD_CHARS+12			
	E9		50		03	F3	004E3		AOBLEQ	#3, I, 45\$	0655	

	50	00000000'	EF	D0	004E7	46\$:	MOVL	ADD_CHARS+12, R0	:	0661	:	
			05	13	004EE		BEQL	47\$:		:	
	52		50	D1	004F0		CMPL	R0, R2	:	0662	:	
			79	19	004F3		BLSS	54\$:		:	
	70	00000000G	FF	E9	004F5	47\$:	BLBC	@HCT+4, 53\$:	0668	:	
	50	00000000G	EF	D0	004FC		MOVL	HCT+28, R0	:	0669	:	
			05	13	00503		BEQL	48\$:		:	
	03		50	D1	00505		CMPL	R0, #3	:	0670	:	
			62	12	00508		BNEQ	53\$:		:	
	51	00000000G	EF	D0	0050A	48\$:	MOVL	TSF, R1	:	0677	:	
	52	04	A1	C2	00511		SUBL2	4(R1), R2	:		:	
54	52	0C	AE	C3	00515		SUBL3	DATE_LENGTH R2, FILL_COUNT	:		:	
	50	00000000G	EF	D0	0051A		MOVL	MRA, R0	:	0681	:	
	52	04	A0	9E	00521		MOVAB	4(R0), R2	:		:	
	50	000CJ000G	EF	D0	00525		MOVL	MRA, R0	:		:	
			53	D4	0052C		CLRL	I	:		:	
			09	11	0052E		BRB	50\$:		:	
	00	B2	20	90	00530	49\$:	MOVAB	#32, @0(R2)	:		:	
			62	D6	00534		INCL	(R2)	:		:	
		0C	A0	D6	00536		INCL	12(R0)	:		:	
F3	53		54	F3	00539	50\$:	AOBLEQ	FILL_COUNT, I, 49\$:	0680	:	
	10	AE	14	AE	9E	0053D	MOVAB	WORK_AREA, PTR	:	0684	:	
			53	D4	00542		CLRL	I	:	0686	:	
			0D	11	00544		BRB	52\$:		:	
	00	B2	10	BE	90	00546	51\$:	MOVAB	@PTR, @0(R2)	:	0687	:
			10	AE	D6	0054B	INCL	PTR	:		:	
			62	D6	0054E		INCL	(R2)	:		:	
		0C	A0	D6	00550		INCL	12(R0)	:		:	
EE	53	0C	AE	F3	00553	52\$:	AOBLEQ	DATE_LENGTH, I, 51\$:	0686	:	
50	61		54	C1	00558		ADDL3	FILL_COUNT, (R1), R0	:	0690	:	
	61	0C	BE40	9E	0055C		MOVAB	@DATE_LENGTH[R0], (R1)	:		:	
50	54	04	A1	C1	00561		ADDL3	4(R1), FILL_COUNT, R0	:	0691	:	
	04	A1	0C	BE40	9E	00566	MOVAB	@DATE_LENGTH[R0], 4(R1)	:		:	
			6B	11	0056C	53\$:	BRB	59\$:	0661	:	
55	52		50	C3	0056E	54\$:	SUBL3	R0, R2, TRUNCATE_POINT	:	0698	:	
			55	DD	00572		PUSHL	TRUNCATE_POINT	:	0701	:	
	00000000V		EF	01	FB	00574	CALLS	#1, TRUNCATE	:		:	
			50	D0	0057B		MOVL	MRA, R0	:	0703	:	
	10	AE	04	A0	D0	00582	MOVL	4(R0), PTR	:		:	
			51	D0	00587		MOVL	ADD_CHARS+12, R1	:	0705	:	
			52	D4	0058E		CLRL	I	:		:	
			17	11	00590		BRB	56\$:		:	
	00000000'		FF	10	BE	90	55\$:	MOVAB	@PTR, @SAVE_CHARS+4	:	0706	:
				10	AE	D6		INCL	PTR	:		:
		00000000'		00000000'	EF	D6		INCL	SAVE_CHARS+4	:		:
		00000000'		00000000'	EF	D6		INCL	SAVE_CHARS+12	:		:
E5	52		51	F3	005A9	56\$:	AOBLEQ	R1, I, 55\$:	0705	:	
	10	AE	00000000'	EF	D0	005AD	MOVL	ADD_CHARS, PTR	:	0709	:	
			52	D4	005B5		CLRL	I	:	0712	:	
			0E	11	005B7		BRB	58\$:		:	
	04	B0	10	BE	90	005B9	57\$:	MOVAB	@PTR, @4(R0)	:		:
			10	AE	D6	005BE	INCL	PTR	:		:	
			04	A0	D6	005C1	INCL	4(R0)	:		:	
			0C	A0	D6	005C4	INCL	12(R0)	:		:	
EE	52		51	F3	005C7	58\$:	AOBLEQ	R1, I, 57\$:	0711	:	
	50	00000000G	EF	D0	005CB		MOVL	TSF, R0	:	0712	:	
	60		51	C0	005D2		ADDL2	R1, (R0)	:	0714	:	

		04	A0	51	C0	005D5	ADDL2	R1, 4(R0)	0715
			50	00000000G	EF	D0 005D9	59\$:	MOVL TSF, R0	0719
					60	D5 005E0	TSTL	(R0)	
					7D	13 005E2	BEQL	64\$	
				01	00000000G	EF	D1 005E4	CMLP	HCT+28, #1
						10	12 005EB	BNEQ	60\$
28		51	00000000G	FF	04	A0 C3 005ED	SUBL3	4(R0), @GCA+140, R1	0726
		A0		51		02 C7 005F6	DIVL3	#2, R1, 40(R0)	
						1A	11 005FB	BRB	61\$
				02	00000000G	EF	D1 005FD	60\$:	CMLP
						11	12 00604	BNEQ	61\$
				0A	00000000G	EF	E9 00606	BLBC	HCT+16, 61\$
28		A0	00000000G	FF	04	A0 C3 0060D	SUBL3	4(R0), @GCA+140, 40(R0)	0731
			00000000G	EF		00 FB 00617	61\$:	CALLS	#0, LOUT
						5B	15 00624	TSTL	SAVE_CHARS+12
						55	DD 00626	BLEQ	65\$
			00000000V	EF		01 FB 00628	PUSHL	TRUNCATE POINT	0744
			10	AE	00000000'	EF	D0 0062F	CALLS	#1, TRUNCATE
				50	00000000G	EF	D0 00637	MOVL	SAVE_CHARS, PTR
				51	00000000G	EF	D0 0063E	MOVL	MRA, R0
						52	D4 00645	MOVL	MRA, R1
						0E	11 00647	CLRL	I
				04	B0	10	BE	90 00649	62\$:
						10	AE	D6 0064E	BRB
						04	A0	D6 00651	MOVVB
						0C	A1	D6 00654	@PTR, @4(R0)
			EA	52	00000000'	EF	F3 00657	INCL	PTR
						20	11 0065F	INCL	4(R0)
						00	EF	00661	INCL
52		7C	A0	01		01	8A	00667	12(R1)
						01	DD	0066B	AOBLEQ
						01	FB	0066D	SAVE_CHARS+12, I, 62\$
			00000000G	EF		50	D0	00674	63\$:
				50	00000000G	EF	D0	00674	BRB
				50	00000000G	EF	D0	0067B	65\$:
				60		57	7D	00688	EXTZV
				28	A0	59	D0	00688	#0, #1, 124(R0), HOLD_BARS
				50	00000000G	EF	D0	0068F	64\$:
						57	7D	00688	BICB2
						59	D0	00688	#1, 124(R0)
						5A	D0	00696	PUSHL
				04	A0	5A	D0	00696	#1
				0C	A0	5B	D0	0069A	CALLS
			00000000G	FF	08	AE	D0	0069E	#1, USKIPL
						3C	11	006A6	TSF, R0
				50	00000000G	EF	D0	006A8	MOVL
				0E	00000000G	FF	E9	006AF	INSV
						00	EF	006B6	HOLD_BARS, #0, #1, 124(R0)
52		7C	A0	01		01	8A	006BC	65\$:
						01	DD	006C0	MOVL
						02	DD	006C0	HOLD_TSF_INT_HL, (R0)
						0C	11	006C2	0758
				52	7C	A0	01	EF	006C4
						01	8A	006CA	0760
						01	DD	006CE	0761
						01	FB	006D0	69\$:
						01	FB	006D0	MOVL
				50	00000000G	EF	D0	006D7	MOVL
				00		52	F0	006DE	INSV
7C		A0	01	50	00000000G	EF	D0	006E4	70\$:
						00	EF	006EB	MOVL
				52	7C	A0	01	EF	006EB
									#0, #1, 124(R0), HOLD_BARS

			7C	A0		01	8A	006F1		BICB2	#1, 124(R0)		
						02	DD	006F5		PUSHL	#2		
			00000000G			01	FB	006F7		CALLS	#1, USKIPL		
						50	00000000G			MOVL	TSF, R0		
7C	A0			01		00				INSV	HOLD_BARS, #0, #1, 124(R0)		
						51	00000000G		71\$:	MOVL	PHAN+16, X		0810
							00000000G			CLRL	PHAN+16		0811
						50	00000000G			MOVL	TSF, R0		0812
						01				EXTZV	#0, #1, 124(R0), HOLD_BARS		
						00				BICB2	#1, 124(R0)		
						01	8A	00725		PUSHL	X		
						51	DD	00729		CALLS	#1, USKIPL		
			00000000G			01	FB	0072B		MOVL	TSF, R0		
						50	00000000G			INSV	HOLD_BARS, #0, #1, 124(R0)		
7C	A0			01		00				EXTZV	#0, #1, 124(R0), HOLD_BARS		0815
						01				BICB2	#1, 124(R0)		
						00				PUSHL	PHAN+16		
						01	8A	00745		CALLS	#1, USKIPL		
			00000000G			01	FB	0074F		MOVL	TSF, R0		
						50	00000000G			INSV	HOLD_BARS, #0, #1, 124(R0)		
7C	A0			01		00				CLRL	PHAN+24		0816
							00000000G			MOVL	#1, @PHAN+28		0817
										MOVL	HOLD_MRA, MRA		0818
							04	AE		MOVL	HOLD_TSF, TSF		0819
									6E	RET			0821
									04				

: Routine Size: 1920 bytes. Routine Base: \$CODE\$ + 0000

: 695 0822 1

```
697 0823 1 %SBTTL 'TRUNCATE --'
698 0824 1 ROUTINE truncate (where) : NOVALUE =
699 0825 1
700 0826 1 ++
701 0827 1 FUNCTIONAL DESCRIPTION:
702 0828 1
703 0829 1 This routine TRUNCATES the text represented by a TSF/MRA pair.
704 0830 1 It does this by scanning the MRA counting printing
705 0831 1 characters (that is, non-escape-sequences).
706 0832 1
707 0833 1 FORMAL PARAMETERS:
708 0834 1
709 0835 1 where - The column at which to truncate the TSF/MRA.
710 0836 1
711 0837 1 IMPLICIT INPUTS: None
712 0838 1
713 0839 1 IMPLICIT OUTPUTS:
714 0840 1
715 0841 1 The current TSF/MRA is modified to effect the desired truncation.
716 0842 1
717 0843 1 ROUTINE VALUE:
718 0844 1 COMPLETION CODES: None
719 0845 1
720 0846 1 SIDE EFFECTS: None
721 0847 1 --
722 0848 1
723 0849 2 BEGIN
724 0850 2 LOCAL
725 0851 2 EXT_LEN,
726 0852 2 INT_LEN,
727 0853 2 PTR;
728 0854 2
729 0855 2 EXT_LEN = 0;
730 0856 2 INT_LEN = 0;
731 0857 2 PTR = .FS_START (MRA);
732 0858 2
733 0859 2 ! First skip over any vertical-motion escape sequences.
734 0860 2 IF .TSF_INT_VL GTR 0
735 0861 2 THEN
736 0862 2
737 0863 2 INCR I FROM 1 TO .TSF_INT_VL DO
738 0864 2 CH$RCHAR_A (PTR);
739 0865 2
740 0866 2 ! Now we are in the horizontal-motion part of the MRA. Scan the MRA, counting
741 0867 2 ! printing characters, until we reach the specified truncation point.
742 0868 2 WHILE .EXT_LEN NEQ .WHERE DO
743 0869 2 BEGIN
744 0870 2
745 0871 2 IF CH$RCHAR_A (PTR) NEQ RINTES
746 0872 2 THEN
747 0873 2 BEGIN
748 0874 2 ! Normal text. Increment both internal-length and external-length counts.
749 0875 2 EXT_LEN = .EXT_LEN + 1;
750 0876 2 INT_LEN = .INT_LEN + 1;
751 0877 2 END
752 0878 2 ELSE
753 0879 2 ! An escape sequence. Add it to the internal-length count and advance
```



```

: 754      0880 3      ! the pointer beyond it.
: 755      0881 4      BEGIN
: 756      0882 4      INT_LEN = .INT_LEN + 3;
: 757      0883 4      CH$RCHAR_A (PTR);
: 758      0884 4      CH$RCHAR_A (PTR);
: 759      0885 3      END;
: 760      0886 3
: 761      0887 2      END;
: 762      0888 2
: 763      0889 2      ! Update the TSF and MRA to effect the truncation.
: 764      0890 2      TSF_EXT_HL = .EXT_LEN;
: 765      0891 2      TSF_INT_HL = .INT_LEN;
: 766      0892 2      FS_LENGTH (MRA) = .INT_LEN + .TSF INT VL;
: 767      0893 2      FS_NEXT (MRA) = CH$PLUS ( .FS_START (MRA), .FS_LENGTH (MRA) );
: 768      0894 2
: 769      0895 1      END;

```

! End of TRUNCATE

003C 0000 TRUNCATE:

```

: 0824      .WORD      Save R2,R3,R4,R5
: 0855      CLRL      EXT_LEN
: 0856      CLRL      INT_LEN
: 0857      MOVL      MRA, R0
: 0860      MOVL      (R0), PTR
: 0860      MOVL      TSF, R1
: 0863      TSTL      24(R1)
: 0863      BLEQ      3$
: 0864      CLRL      I
: 0864      BRB      2$
: 0864      INCL      PTR
: 0868      AOBLEQ   24(R1), I, 1$
: 0868      CMPL      EXT_LEN, WHERE
: 0871      BEQL      5$
: 0871      MOVZBL   (PTR)+, R2
: 0875      CMPL      R2, #RINTES
: 0876      BEQL      4$
: 0876      INCL      EXT_LEN
: 0876      INCL      INT_LEN
: 0877      BRB      3$
: 0882      ADDL2   #3, INT_LEN
: 0884      ADDL2   #2, PTR
: 0888      BRB      3$
: 0890      MOVL      EXT_LEN, 4(R1)
: 0891      MOVL      INT_LEN, (R1)
: 0892      MOVAB   @24(R1)[INT_LEN], 12(R0)
: 0893      ADDL3   12(R0), (R0), 4(R0)
: 0895      RET

```

: Routine Size: 91 bytes, Routine Base: \$CODE\$ + 0780

```

: 770      0896 1
: 771      0897 1 END
: 772      0898 0 ELUDOM

```

! End of module TOPPAG

PSECT SUMMARY

Name	Bytes	Attributes
\$OWNS	134	NOVEC, WRT, RD, NOEXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)
\$PLITS	24	NOVEC, NOWRT, RD, NOEXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)
\$CODE\$	2011	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	0	0	252	00:00.2
_\$255\$DUA28:[RUNOFF.SRC]DSRLIB.L32;1	1248	72	5	86	00:00.3

COMMAND QUALIFIERS

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

: Size: 2011 code + 158 data bytes
 : Run Time: 00:38.4
 : Elapsed Time: 01:17.8
 : Lines/CPU Min: 1403
 : Lexemes/CPU-Min: 17704
 : Memory Used: 382 pages
 : Compilation Complete

The image displays a dense grid of approximately 150 small, overlapping screenshots of VAX/VMS system output. Each screenshot shows various data lists and command-line interfaces. Several titles are clearly legible across the grid:

- STD LIS
- SETDAT LIS
- STC LIS
- STYLE LIS
- SCANT LIS
- STKFRM LIS
- SKIPL LIS
- SCL LIS
- SKPSEP LIS
- STH LIS
- SETTIM LIS
- SUBST LIS
- TITLES LIS
- TOPPAG LIS
- SUBPAG LIS

The individual screenshots contain detailed text, including headers, data columns, and status information, though the text is too small to read in detail. The overall appearance is that of a comprehensive reference manual for system utilities.

