

CCCCCCCC CCCCCCCC 000000 000000 NN NN TTTTTTTTTT EEEEEEEEEE NN NN TTTTTTTTTT SSSSSSSS
CC CC 00 00 NN NN NN TT EE NN NN TTTTTTTTTT SSSSSSSS
CC CC 00 00 NNNN NN NN TT EE NNNN NN NN TT SS
CC CC 00 00 NNNN NN NN TT EE NNNN NN NN TT SS
CC CC 00 00 NN NN NN TT EEEEEEE NN NN NN TT SS
CC CC 00 00 NN NN NN TT EEEEEEE NN NN NN TT SSSSSS
CC CC 00 00 NN NNNN NN NN TT EE NN NNNN TT SS
CC CC 00 00 NN NNNN NN NN TT EE NN NNNN TT SS
CC CC 00 00 NN NN NN TT EE NN NN NN TT SS
CC CC 00 00 NN NN NN TT EE NN NN NN TT SS
CCCCCCCC CCCCCCCC 000000 000000 NN NN TT EEEEEEEEEE NN NN TT SSSSSSSS ...
CCCCCCCC CCCCCCCC 000000 000000 NN NN TT EEEEEEEEEE NN NN TT SSSSSSSS ...

LL L L IIIIIIII SSSSSSSS
LL L L IIIIIIII SSSSSSSS
LL L L IIIIIIII SS SS
LL L L IIIIIIII SS SS
LL L L IIIIIIII SSSSSS
LL L L IIIIIIII SSSSSS
LL L L IIIIIIII SS SS
LL L L IIIIIIII SS SS
LLLLLLLLLL L L L L L L IIIIIIII SSSSSSSS

```
0001 0 XTITLE 'CONTENTS - Table of Contents formatter program'  
0002 0 !<BLF/NOFORMAT>  
0003 0  
0004 0 MODULE contents (IDENT = 'V04-000'  
0005 0           %BLISS32[, ADDRESSING_MODE (EXTERNAL = LONG_RELATIVE, NONEXTERNAL = LONG_RELATIVE)]  
0006 0           ) =  
0007 1 BEGIN  
0008 1  
0009 1 !<BLF/FORMAT>  
0010 1 <BLF/LOWERCASE_USER>  
0011 1 <BLF/UPPERCASE_KEY>  
0012 1 <BLF/MACRO>  
0013 1  
0014 1  
0015 1 *****  
0016 1 *  
0017 1 *  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY  
0018 1 *  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.  
0019 1 *  ALL RIGHTS RESERVED.  
0020 1 *  
0021 1 *  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED  
0022 1 *  ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE  
0023 1 *  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER  
0024 1 *  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY  
0025 1 *  OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY  
0026 1 *  TRANSFERRED.  
0027 1 *  
0028 1 *  THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE  
0029 1 *  AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT  
0030 1 *  CORPORATION.  
0031 1 *  
0032 1 *  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS  
0033 1 *  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.  
0034 1 *  
0035 1 *  
0036 1 *****  
0037 1 .  
0038 1 .  
0039 1 ++  
0040 1 FACILITY:  
0041 1 DSR (Digital Standard RUNOFF) /DSRPLUS DSRTOC/CONTENTS Utility  
0042 1  
0043 1 ABSTRACT:  
0044 1 Read and process a RUNOFF binary table of contents file.  
0045 1  
0046 1 ENVIRONMENT: Transportable  
0047 1  
0048 1 AUTHOR: JPK  
0049 1  
0050 1 CREATION DATE: March 1982  
0051 1  
0052 1 MODIFIED BY:  
0053 1  
0054 1     013      JPK00011    20-May-1983  
0055 1     Modified CONTENTS to check status of $XPO_PARSE_SPEC to avoid  
0056 1     XPORT error messages.  
0057 1
```

58 0058 1 012 JPK00010 05-May-1983
59 0059 1 Modified CONTENTS to change text for index record group to all
60 0060 1 caps. Modified CAPTION to change the number of blank lines
61 0061 1 after the entity title line.
62 0062 1
63 0063 1 011 JPK00009 24-Mar-1983
64 0064 1 Modified CNTT20 to support new command line syntax.
65 0065 1 Modified CONTENTS adding routines PROCESS PAGE,
66 0066 1 PROCESS_ENTITY_INFO and PROCESS_ENTITY_TXT to remove
67 0067 1 code from routine TOC so that CONTENTS will compile on TOPS-20
68 0068 1 Modified CNTVMSREQ to remove conditional require of RNODEF.
69 0069 1
70 0070 1 010 JPK00008 09-Mar-1983
71 0071 1 Modified CONTENTS and CAPTION to support new BRN formats,
72 0072 1 support SEND CONTENTS, /DOUBLE_SPACE, page numbered chapters,
73 0073 1 guarantee space after section number and to write new
74 0074 1 prologue and epilog for RUNOFF output.
75 0075 1 Modified FORMAT to quote only the RUNOFF flags used by CONTENTS.
76 0076 1 Modified CNTVMS to fix default for /DOUBLE_SPACE and do more
77 0077 1 value checking.
78 0078 1
79 0079 1 009 JPK00007 14-Feb-1983
80 0080 1 Global edit of all sources for CONTENTS/DSRTOC:
81 0081 1 - module names are now consistant with file names
82 0082 1 - copyright dates have been updated
83 0083 1 - facility names have been updated
84 0084 1 - revision history was updated to be consistant with DSR/DSRPLUS
85 0085 1
86 0086 1 008 JPK00006 14-Feb-1983
87 0087 1 Modified CNTVMS, CONTENTS, FORMAT and CNTVMSMSG to generate
88 0088 1 error messages for DSRTOC or CONTENTS depending on the
89 0089 1 compiletime variant for DSRPLUS (/VARIANT:8192)
90 0090 1
91 0091 1 007 JPK00005 14-Feb-1983
92 0092 1 Changed the names of the temporary files used to store
93 0093 1 .SEND TOC, figure, table and example captions from XPORT.TMP;1,
94 0094 1 ;2, ;3, ;4 to OCNST.TMP, OCNTFG.TMP, OCNTTB.TMP, and
95 0095 1 OCNTEX.TMP respectively. Delete the output file if /NOOUTPUT
96 0096 1 was specified on the command line. Module CONTENTS modified.
97 0097 1
98 0098 1 006 JPK00004 11-Feb-1983
99 0099 1 Changed the global variable name INDENT to LINE_INDENT in
100 0100 1 modules CONTENTS, CAPTION, FORMAT and GBLDCL.
101 0101 1 Removed declarations of PDENTS in modules CNTVMS, CONTENTS,
102 0102 1 and CAPTION and replaced with a module wide BIND using the
103 0103 1 new name INDENTS.
104 0104 1 Changed handling of INDENTS [1]. It no longer represents the
105 0105 1 sum of the chapter and title indents.
106 0106 1
107 0107 1 005 KFA00003 04-Feb-1983
108 0108 1 Modified format of header level one to not include
109 0109 1 ".0" in the counter. Change was in OUTSEC in module CONTENTS.
110 0110 1
111 0111 1 004 KFA00002 14-Oct-1982
112 0112 1 Modified CNTVMS, CONTENTS, CAPTION and CNTCLI.REQ to handle
113 0113 1 new syntax for /INDENT qualifier. All indents are now
114 0114 1 stored in the vector CMDBLK [CONTENT\$AG_HL_INDENT []].

115 0115 1 | where 0 = chapter indent, 1 = header level one indent, etc.
116 0116 1 |
117 0117 1 | 003 KFA00001 13-Oct-1982
118 0118 1 | Modified CONTENTS and CAPTION to allow null captions for
119 0119 1 | headers, examples, figures, & tables.
120 0120 1 |
121 0121 1 | 002 JPK00001 17-Sep-1982
122 0122 1 | Modified CONTENTS for TOPS-20. This included splitting routine
123 0123 1 | toc () into toc () and tocfin () so that it can be compiled on
124 0124 1 | the 20, and calling \$XPO_GET_MEM and \$XPO_FREE_MEM to
125 0125 1 | temporarily store the prefix and suffix strings for HEADER
126 0126 1 | EXAMPLE, FIGURE and TABLE.
127 0127 1 |
128 0128 1 | ---
129 0129 1 |
130 0130 1 |
131 0131 1 | TABLE OF CONTENTS:
132 0132 1 |
133 0133 1 |
134 0134 1 | FORWARD ROUTINE
135 0135 1 | toc
136 0136 1 | tocini,
137 0137 1 | process_page : NOVALUE,
138 0138 1 | process_entity_info : NOVALUE,
139 0139 1 | process_entity_txt : NOVALUE,
140 0140 1 | tocfin,
141 0141 1 | vfycap,
142 0142 1 | outsec : NOVALUE;
143 0143 1 |
144 0144 1 |
145 0145 1 | INCLUDE FILES:
146 0146 1 |
147 0147 1 |
148 0148 1 | LIBRARY 'NXPORT:XPORT';
149 0149 1 |
150 L 0150 1 | %IF %BLISS (BLISS36)
151 U 0151 1 | %THEN
152 U 0152 1 |
153 U 0153 1 | REQUIRE 'REQ:RNODEF';
154 U 0154 1 |
155 0155 1 | %FI
156 0156 1 |
157 L 0157 1 | %IF %BLISS (BLISS32)
158 U 0158 1 | %THEN
159 U 0159 1 |
160 0160 1 | REQUIRE 'REQ:CNTVMSREQ'; ! VMS error messages
161 0400 1 |
162 0401 1 | %FI
163 0402 1 |
164 0403 1 | REQUIRE 'REQ:BRNRTY'; ! Binary RUNOFF file formats
165 0483 1 |
166 0484 1 | REQUIRE 'REQ:PAGEN'; ! Page reference format
167 0565 1 |
168 0566 1 | REQUIRE 'REQ:TOCRTY'; ! Table of Contents file formats
169 0676 1 |
170 0677 1 | REQUIRE 'REQ:CNTCLI'; ! Command line information block formats
171 0805 1 |

```
172 0806 1 MACROS:  
173 0807 1  
174 0808 1  
175 0809 1  
176 0810 1 MACRO  
177 0811 1  
178 0812 1 Write a character to output line  
179 0813 1  
180 M 0814 1 write_char (ch) [] =  
181 M 0815 1 BEGIN  
182 M 0816 1 CH$WCHAR_A (ch, lp);  
183 M 0817 1 intlin = .intlin + 1;  
184 M 0818 1  
185 M 0819 1 %IF NOT %NULL (%REMAINING)  
186 M 0820 1 %THEN  
187 M 0821 1 extlin = .extlin + 1;  
188 M 0822 1 %FI  
189 M 0823 1  
190 M 0824 1 END  
191 0825 1 %,  
192 0826 1  
193 0827 1 Write a text literal to the output line  
194 0828 1  
195 M 0829 1 literal_text (str) =  
196 M 0830 1 BEGIN  
197 M 0831 1 CH$MOVE (%CHARCOUNT (str), CH$PTR (UPLIT (str)), .lp);  
198 M 0832 1 lp = CH$PLUS (.lp, %CHARCOUNT (str));  
199 M 0833 1 intlin = .intlin + %CHARCOUNT (str);  
200 M 0834 1 END  
201 0835 1 %,  
202 0836 1  
203 0837 1 Pad the output line with blanks  
204 0838 1  
205 M 0839 1 pad (n_blanks) =  
206 M 0840 1 BEGIN  
207 M 0841 1  
208 M 0842 1 IF n_blanks GTR 0  
209 M 0843 1 THEN  
210 M 0844 1 BEGIN  
211 M 0845 1 CH$FILL (%C' ', n_blanks, .lp);  
212 M 0846 1 lp = CH$PLUS (.lp, n_blanks);  
213 M 0847 1 intlin = .intlin + n_blanks;  
214 M 0848 1 extlin = .extlin + n_blanks;  
215 M 0849 1 END;  
216 M 0850 1  
217 M 0851 1 END  
218 0852 1 %,  
219 0853 1  
220 0854 1 Clear the text lines being built up.  
221 0855 1  
222 M 0856 1 clr_line (_)=  
223 M 0857 1 BEGIN  
224 M 0858 1 lp = CH$PTR (line);  
225 M 0859 1 intlin = 0;  
226 M 0860 1 extlin = 0;  
227 M 0861 1 END  
228 0862 1 %.
```

```
229 0863 1
230 0864 1 | Insert specified character sequence into file, as is.
231 0865 1
232 M 0866 1
233 M 0867 1 put (str) =
234 M 0868 1 BEGIN
235 M 0869 1 $str_copy (string = str, target = tmpstr);
236 M 0870 1 chrout = .chrout + .tmpstr [str$h_length];
237 M 0871 1 $xpo_put (iob = tocoob, string = tmpstr);
238 M 0872 1 | For TMS output, split the output file if it gets too large
239 M 0873 1
240 M 0874 1
241 M 0875 1 IF .cmdblk [contents$v_tms11] THEN split ();
242 M 0876 1
243 M 0877 1 END
244 0878 1 %
245 0879 1
246 0880 1
247 0881 1 | EQUATED SYMBOLS:
248 0882 1 !
249 0883 1
250 0884 1 LITERAL
251 0885 1 true = 1,
252 0886 1 false = 0,
253 0887 1 new_captions = 1, ! Using HEADER FIGURE, etc.
254 0888 1 old_captions = -1, ! Using SEND TOC for captions
255 0889 1 tms_line_length = 600,
256 0890 1 infinity = 1*(%BPVAL - 1) - 1;
257 0891 1
258 0892 1
259 0893 1 | OWN STORAGE:
260 0894 1 !
261 0895 1
262 0896 1 OWN
263 0897 1 tociob : $xpo_iob (),
264 0898 1 file_type, ! IOB for binary index file.
265 0899 1 fgiob : $xpo_iob (), ! Input file type
266 0900 1 tbiob : $xpo_iob (), ! IOB for HEADER FIGURES
267 0901 1 exiob : $xpo_iob (), ! IOB for HEADER TABLEs
268 0902 1 hlcnt, ! IOB for HEADER EXAMPLEs
269 0903 1 lasthl, ! Maximum number of header levels
270 0904 1 toc_page : INITIAL (false), ! previous "n"
271 0905 1 captyp, ! True if a CHAPTER should have a page reference
272 0906 1 capmsg, ! To tell if SEND TOC or HEADER x was used
273 0907 1 errcnt; ! True if incompatible captions and message was written
274 0908 1 ! Count of number of CONFUSED messages.
275 0909 1
276 0910 1 | EXTERNAL REFERENCES:
277 0911 1 !
278 0912 1
279 0913 1 EXTERNAL
280 0914 1 cmdblk : $contents_cmd, ! Command line information block
281 0915 1 tocoob : $xpo_iob (?), ! IOB for the resulting .RNT file
282 0916 1 stciob : $xpo_iob (), ! IOB for send toc temp file
283 0917 1 cntvrl, ! Length of version number string
284 0918 1 cntvrp, ! CH$PTR to version number string
285 0919 1 chrout, ! Number of characters written to output file
```

```
; 286 0920 1 tmpstr : $str_descriptor (),          | For temporary strings
287 0921 1 txtlen,                                | Length of input text.
288 0922 1 hl_n,                                 | 'n' from latest .HL n command
289 0923 1 hl[ist : VECTOR [100],                | Header level counters
290 0924 1 hldsp : VECTOR [100],                 | Header level display characteristics
291 0925 1 major,                                | Major record type code
292 0926 1 minor,                                | Minor record type code
293 0927 1 pagen : page_definition,              | Current page number.
294 0928 1 nxtstc,                               | Next SEND TOC "n" value to process
295 0929 1 rno_ex : $str_descriptor (),          | String constant 'EXAMPLES'
296 0930 1 rno_fg : $str_descriptor (),          | String constant 'FIGURES'
297 0931 1 rno_tb : $str_descriptor (),          | String constant 'TABLES'
298 0932 1 tms_ex : $str_descriptor (),          | String constant 'Examples'
299 0933 1 tms_fg : $str_descriptor (),          | String constant 'Figures'
300 0934 1 tms_tb : $str_descriptor (),          | String constant 'Tables'
301 0935 1 prestr : $str_descriptor (),          | Caption prefix strings
302 0936 1 sfxstr : $str_descriptor (),          | Caption suffix strings
303 0937 1 lp,                                    | CH$PTR along line being built up
304 0938 1 intlin,                               | Number of characters needed to represent text
305 0939 1 extlin,                               | Number of resulting print positions
306 0940 1 line : VECTOR [CH$ALLOCATION (10000)],| Buffer in which line is being built up.
307 0941 1 lenpag,                               | Number of characters in the converted page number.
308 0942 1 txtpag : VECTOR [CH$ALLOCATION (50)],| The text (lots of room)
309 0943 1 ptrpag,                               | A CH$PTR (working pointer)
310 0944 1 lensct,                               | Number of characters in the converted section number
311 0945 1 txtsct : VECTOR [CH$ALLOCATION (150)],| The text (lots of room)
312 0946 1 ptrsct,                               | A CH$PTR (working pointer)
313 0947 1 rmargin,                             | Used by ENDWRD for controlling filling lines.
314 0948 1 wrap,                                | Wrap long lines around to here.
315 0949 1 line_indent;                         | Assume this standard indentation before the text.
316 0950 1
317 0951 1 BIND
318 0952 1     indents = cmdblk [contents$ag_hl_indent] : VECTOR;
319 0953 1
320 0954 1 EXTERNAL ROUTINE
321 0955 1     fmttxt : NOVALUE,                  | Generate formatted text
322 0956 1     insref : NOVALUE,                 | Insert a page reference
323 0957 1     hdr x : NOVALUE,                 | Process .HEADER x Captions
324 0958 1     stchdr : NOVALUE,                 | Process .SEND TOC n Captions
325 0959 1     insrnt : NOVALUE,                 | Insert text directly into RNT
326 0960 1     split : NOVALUE,                 | Start new output file for TMS
327 0961 1     pacbas,                            | Convert a number to ASCII
328 0962 1     pacpag,                            | Convert a PAGEN to a page number
329 0963 1     pacsec,                            | Format a section number
330 0964 1     pacxxx;                           | Convert number to ASCII using display codes
331 0965 1
332 L 0966 1 %IF %BLISS (BLISS32)
333 0967 1 %THEN
334 0968 1
335 0969 1 EXTERNAL ROUTINE
336 0970 1     open_error;                      | Action routine to handle file open failures
337 0971 1
338 0972 1 %FI
```

: 340 0973 1 %SBTTL 'TOC - Transportable entry point'
341 0974 1 GLOBAL ROUTINE toc =
342 0975 1 !++
343 0976 1 FUNCTIONAL DESCRIPTION:
344 0977 1
345 0978 1 Reads a binary table of contents file, as generated by RUNOFF
346 0979 1 and uses it to generate a .RNT file.
347 0980 1
348 0981 1 FORMAL PARAMETERS:
349 0982 1
350 0983 1 NONE
351 0984 1
352 0985 1 IMPLICIT INPUTS:
353 0986 1
354 0987 1 Implicit is the order in which records are written, and the
355 0988 1 format of the file. The format of the file is adequately
356 0989 1 documented by the code. It will not be further explained.
357 0990 1
358 0991 1 The order of information in the file is not obvious, although
359 0992 1 its use shows what is available when.
360 0993 1
361 0994 1 Briefly, there is an entry in the file (MAJ PAGE) for every page
362 0995 1 in the document, regardless of whether or not that page is
363 0996 1 mentioned in the table of contents. Further, the page number is
364 0997 1 guaranteed to occur before you get some text associated with
365 0998 1 that page. So, if header level text is found, the page number
366 0999 1 is already correct.
367 1000 1
368 1001 1 Similarly, header level numbers and display information are
369 1002 1 always written before the header level text to which it is
370 1003 1 applicable. Note that the header level information is *not*
371 1004 1 reset in the file at the start of a chapter; you have to wait
372 1005 1 for the next header level number record to be input to be sure
373 1006 1 where you are. The page number and header level information is
374 1007 1 as valid for text from the .SEND command as it is for text from
375 1008 1 a .HL command.
376 1009 1
377 1010 1 IMPLICIT OUTPUTS:
378 1011 1
379 1012 1 NONE
380 1013 1
381 1014 1 ROUTINE VALUE:
382 1015 1 COMPLETION CODES:
383 1016 1
384 1017 1 NONE
385 1018 1
386 1019 1 SIDE EFFECTS:
387 1020 1
388 1021 1 NONE
389 1022 1
390 1023 1 !--
391 1024 1
392 1025 2 BEGIN
393 1026 2
394 1027 2 Do initialization
395 1028 2
396 1029 2 IF NOT tocini () THEN RETURN false;

```
: 397      1030 2
: 398      1031 2
: 399      1032 2 | Read record-groups until end of file. Process each record group
: 400      1033 2 | in turn. Return at end-of-file.
: 401      1034 2
: 402      1035 2 WHILE $xpo_get (iob = tociob, fullwords = 2) EQL xpo$_normal DO
: 403      1036 3   BEGIN
: 404      1037 3
: 405      1038 3   BIND
: 406      1039 3     codes = .tociob [iob$a_data] : VECTOR;
: 407      1040 3
: 408      1041 4     IF (.file_type EQL brn_file) AND (.codes [0] NEQ brn_contents)
: 409      1042 3   THEN
: 410      1043 4     BEGIN
: 411      1044 4       Not a contents record group. Skip it.
: 412      1045 4
: 413      1046 4
: 414      1047 4     $xpo_get (iob = tociob, fullwords = .codes [1]);
: 415      1048 4   END
: 416      1049 3 ELSE
: 417      1050 4   BEGIN
: 418      1051 4       A contents record group or not a .BRN file
: 419      1052 4
: 420      1053 4
: 421      1054 4     IF .file_type EQL brn_file THEN $xpo_get (iob = tociob, fullwords = 2);
: 422      1055 4
: 423      1056 4     major = .codes [0];
: 424      1057 4     minor = .codes [1];
: 425      1058 4
: 426      1059 4     SELECTONE true OF
: 427      1060 4       SET
: 428      1061 4
: 429      1062 4       [.major EQL maj_new_toc] :
: 430      1063 5   BEGIN
: 431      1064 5       File identification record in a BRN file
: 432      1065 5
: 433      1066 5
: 434      1067 6     IF (.file_type NEQ brn_file)
: 435      1068 6     OR (.minor NEQ toc_format)
: 436      1069 5   THEN
: 437      1070 6   BEGIN
: 438      1071 6       Report bad file format and quit
: 439      1072 6
: 440      1073 6
: 441      1074 6
: L 1075 6 %IF %BLISS (BLISS32)
: 443      1076 6 %THEN
: 444      1077 6     SIGNAL (contents$_invinput, 1, tociob [iob$t_resultant]);
: 445      1078 6 %ELSE
: 446      1079 6     $xpo_put_msg (severity = warning,
: 447      1080 6       string = $str_concat ('invalid file format ''',
: 448      1081 6       tociob [iob$t_resultant], '''));
: 449      1082 6 %FI
: 450      1083 5     END;
: 451      1084 5
: 452      1085 4     END;
: 453      1086 4
```

```
; 454      1087 4      [(.major EQL maj_runoff) AND (.minor EQL min_page)] :  
; 455      1088 4      | Process a page number.  
; 456      1089 4      |  
; 457      1090 4      | process_page ();  
; 458      1091 4      |  
; 459      1092 4      |  
; 460      1093 4      [(.major EQL maj_runoff) AND (.minor EQL min_index)] :  
; 461      1094 5      | BEGIN  
; 462      1095 5      |  
; 463      1096 5      | Make an entry in the table of contents for the index  
; 464      1097 5      |  
; 465      1098 5      | errcnt = 0; ! Reset count of CONFUSED messages.  
; 466      1099 5      |  
; 467      1100 5      IF .cmdblk [contents$v_tms11]  
; 468      1101 5      THEN  
; 469      1102 6      | BEGIN  
; 470      1103 6      | put ('');  
; 471      1104 6      |  
; 472      1105 6      | IF .hl_n NEQ 0 THEN put ('[va15]');  
; 473      1106 6      |  
; 474      1107 6      | put ('*cOn*Index@A');  
; 475      1108 6      | END  
; 476      1109 5      | ELSE  
; 477      1110 6      | BEGIN  
; 478      1111 6      | IF .hl_n NEQ 0 THEN put ('.SK2') ELSE put ('.SK1');  
; 479      1112 6      |  
; 480      1113 6      | IF .indents [0] NEQ 0  
; 481      1114 6      | THEN  
; 482      1115 6      | put ($str_concat ('.I-', $str_ascii (.indents [0])));  
; 483      1116 6      |  
; 484      1117 6      | put ('INDEX');  
; 485      1118 6      | put ('.BLANK');  
; 486      1119 5      | END;  
; 487      1120 5      |  
; 488      1121 5      | hl_n = 0; ! Set .BLANK info  
; 489      1122 5      | lasthl = 0;  
; 490      1123 4      | END;  
; 491      1124 4      |  
; 492      1125 5      [(.major EQL maj_runoff)  
; 493      1126 5      | AND (  
; 494      1127 6      |   (.minor EQL min_exampl_inf)  
; 495      1128 6      |   OR (.minor EQL min_table_inf)  
; 496      1129 6      |   OR (.minor EQL min_figur_inf)  
; 497      1130 5      |   )  
; 498      1131 4      | ] :  
; 499      1132 4      |  
; 500      1133 4      | Example, Table, or Figure information  
; 501      1134 4      |  
; 502      1135 4      | process_entity_info ();  
; 503      1136 4      |  
; 504      1137 5      [(.major EQL maj_runoff)  
; 505      1138 5      | AND (  
; 506      1139 6      |   (.minor EQL min_exampl_txt)  
; 507      1140 6      |   OR (.minor EQL min_exampl_txt_u)  
; 508      1141 6      |   OR (.minor EQL min_figur_txt)  
; 509      1142 6      |   OR (.minor EQL min_figur_txt_u)  
; 510      1143 6      |   OR (.minor EQL min_table_txt)
```

```
511      1144 6          OR (.minor EQL min_table_txt_u)
512      1145 5          )
513      1146 4          ]
514      1147 4          | Figure, Table or Example text
515      1148 4          |
516      1149 4          process_entity_txt ();
517      1150 4          [
518      1151 4          | (.major EQL maj_runoff) AND (.minor EQL min_hl_inf) :
519      1152 4          BEGIN
520      1153 5          | Read header level counters and display descriptors
521      1154 5          |
522      1155 5          errcnt = 0;           ! Reset count of CONFUSED messages.
523      1156 5          |
524      1157 5          First get count of header level counters.
525      1158 5          |
526      1159 5          $xpo_get (iob = tociob, fullwords = 1);
527      1160 5          hlcnt = .(tociob [iob$a_data]);
528      1161 5          |
529      1162 5          Next, get the "n" value for this ".HL n" command
530      1163 5          |
531      1164 5          $xpo_get (iob = tociob, fullwords = 1);
532      1165 5          hl_n = .(tociob [iob$a_data]);
533      1166 5          |
534      1167 5          Now read the complete set of header level counters,
535      1168 5          not all of which are important
536      1169 5          |
537      1170 5          $xpo_get (iob = tociob, fullwords = .hlcnt);
538      1171 5          BEGIN
539      1172 5          |
540      1173 6          BIND
541      1174 6          |   x = .tociob [iob$a_data] : VECTOR;
542      1175 6          |
543      1176 6          INCR i FROM 1 TO .hlcnt DO
544      1177 6          |   BEGIN
545      1178 6          |       Copy the header level counters to a safe place
546      1179 7          |       hllist [.i - 1] = .x [.i - 1];
547      1180 7          |   END;
548      1181 7          |
549      1182 7          |
550      1183 7          |
551      1184 6          |
552      1185 6          |
553      1186 5          |
554      1187 5          |
555      1188 5          Finally, read the header level display descriptors.
556      1189 5          |
557      1190 5          $xpo_get (iob = tociob, fullwords = .hlcnt);
558      1191 6          BEGIN
559      1192 6          |
560      1193 6          BIND
561      1194 6          |   x = .tociob [iob$a_data] : VECTOR;
562      1195 6          |
563      1196 6          INCR i FROM 1 TO .hlcnt DO
564      1197 7          |   BEGIN
565      1198 7          |       Copy the header level descriptors to a safe place
566      1199 7          |
567      1200 7          |
```

```
; 568      1201 7          hldsp [.i - 1] = .x [.i - 1];  
569      1202 6          END;  
570      1203 6  
571      1204 5          END;  
572      1205 4          END;  
573      1206 4  
574      1207 5          [(.major EQL maj_runoff)  
575      1208 5          AND (  
576      1209 6          (.minor EQL min_hl_txt)  
577      1210 6          OR (.minor EQL min_hl_txt_u)  
578      1211 5          )]  
579      1212 4          ] : BEGIN  
580      1213 5          | Process the text part of a header level.  
581      1214 5          LOCAL  
582      1215 5          | needed_fill;  
583      1216 5  
584      1217 5  
585      1218 5  
586      1219 5  
587      1220 5          errcnt = 0;           ! Reset count of CONFUSED messages.  
588      1221 5  
589      1222 5          | First read the number of characters needed to  
590      1223 5          | represent the header level  
591      1224 5  
592      1225 5          $xpo_get (iob = tociob, fullwords = 1);  
593      1226 5          txtlen = .(tociob [iob$a_data]);   ! Save text length.  
594      1227 5  
595      1228 5          | Generate a table of contents with or without text  
596      1229 5          | The header level may or may not have text associated with it  
597      1230 5          | Get the text, if there is any  
598      1231 5  
599      1232 5          IF .txtlen NEQ 0  
600      1233 5          THEN  
601      1234 5          | $xpo_get (iob = tociob, fullwords = CH$ALLOCATION (.txtlen));  
602      1235 5  
603      1236 5  
604      1237 5          | Generate the associated TOC text only if this header  
605      1238 5          | level is within the depth specified by the user.  
606      1239 5  
607      1240 5          IF .hl_n LEQ .cmdblk [contents$g_headers]  
608      1241 5          THEN  
609      1242 6          BEGIN  
610      1243 6  
611      1244 7          IF (.hl_n LEQ .cmdblk [contents$g_double_space])  
612      1245 6          AND NOT .cmdblk [contents$v_tms11]  
613      1246 6          THEN  
614      1247 6          put ('.BLANK');  
615      1248 6  
616      1249 6  
617      1250 6          | Initialize output line.  
618      1251 6          clr_line ();  
619      1252 6  
620      1253 6  
621      1254 6          | All header level entries are generated under the  
622      1255 6          | assumption of a .LMn command in the file.  
623      1256 6          | Therefore, even though no text has been generated  
624      1257 6          | yet we have to act as if 'n' characters have been
```

```
: 625      1258 6          | taken up.
: 626      1259 6          extlin = .indents [0];
: 627      1260 6          | Set standard indentation in front of this header level line
: 628      1261 6          line_indent = .indents [0];
: 629      1262 6          | Generate the header level numbers if the header is
: 630      1263 6          numbered. If unnumbered, just put out the text.
: 631      1264 6          outsec ();
: 632      1265 6          |
: 633      1266 6          wrap = .extlin;
: 634      1267 6          | Generate the page number so as to get its length.
: 635      1268 6          This is needed so that the proper number of dots
: 636      1269 6          can be supplied.
: 637      1270 6          |
: 638      1271 6          Set up line-wrap point
: 639      1272 6          |
: 640      1273 6          wrap = .extlin;
: 641      1274 6          |
: 642      1275 6          IF NOT .cmdblk [contents$v_standard_page]
: 643      1276 6          THEN
: 644      1277 6          BEGIN
: 645      1278 6          |
: 646      1279 6          Use the running page counter instead of a
: 647      1280 6          real page number.
: 648      1281 6          lenpag = pacbas (.pagen [sct_run_page], ptrpag, 10);
: 649      1282 6          END
: 650      1283 7          ELSE
: 651      1284 7          BEGIN
: 652      1285 7          |
: 653      1286 7          Supply the usual page number.
: 654      1287 7          lenpag = pacpag (pagen, ptrpag);
: 655      1288 7          END;
: 656      1289 7          |
: 657      1290 6          Set limit beyond which words must be wrapped
: 658      1291 7          rmargin = .cmdblk [contents$g_page_width] - .lenpag - 1;
: 659      1292 7          |
: 660      1293 7          Scan text and put it onto the line.
: 661      1294 7          fmttxt (.txtlen, CH$PTR (.tociob [iob$a_data]));
: 662      1295 7          |
: 663      1296 6          Insert page reference into line if desired.
: 664      1297 6          |
: 665      1298 6          insref ();
: 666      1299 6          |
: 667      1300 6          Now actually put this line of text into the file
: 668      1301 6          put ((.intlin, CH$PTR (line)));
: 669      1302 6          clr_line ();
: 670      1303 6
: 671      1304 6
: 672      1305 6
: 673      1306 6
: 674      1307 6
: 675      1308 6
: 676      1309 6
: 677      1310 6
: 678      1311 6
: 679      1312 6
: 680      1313 6
: 681      1314 6
```

682 1315 5
683 1316 5
684 1317 5
685 1318 4
686 1319 4
687 1320 5
688 1321 5
689 1322 6
690 1323 6
691 1324 5
692 1325 4
693 1326 5
694 1327 5
695 1328 5
696 1329 5
697 1330 5
698 1331 5
699 1332 5
700 1333 5
701 1334 5
702 1335 5
703 1336 5
704 1337 5
705 1338 5
706 1339 5
707 1340 5
708 1341 5
709 1342 5
P 1343 5
710 1344 5
711 1345 5
712 1346 5
713 1347 5
714 1348 5
715 1349 5
716 1350 5
717 1351 5
718 1352 5
P 1353 5
721 1354 5
722 1355 5
723 1356 4
724 1357 4
725 1358 5
726 1359 5
727 1360 6
728 1361 6
729 1362 6
730 1363 6
731 1364 6
732 1365 6
733 1366 5
734 1367 4
735 1368 5
736 1369 5
737 1370 5
738 1371 5

END;
lasthl = .hl_n;
END;
[(.major EQL maj_runoff)
AND (
(.minor EQL min_chapt_inf)
OR (.minor EQL min_append_inf)
)
]
:
BEGIN
| Process Chapter/Appendix info record
LOCAL
len;
\$xpo_get (iob = tociob, fullwords = 1); ! Get TOC page
toc_page = ..tociob [iob\$a_data];
\$xpo_get (iob = tociob, fullwords = 1); ! Skip unused word
\$xpc_get (iob = tociob, fullwords = 1); ! Get prefix length
len = ..tociob [iob\$a_data];
IF .len GTR 0
THEN ! Get prefix string
\$xpo_get (iob = tociob, fullwords = CH\$ALLOCATION (.len));
\$str_copy (string = (.len, CH\$PTR (.tociob [iob\$a_data])),
target = prestr);
\$xpo_get (iob = tociob, fullwords = 1); ! Get suffix length
len = ..tociob [iob\$a_data];
IF .len GTR 0
THEN ! Get suffix string
\$xpo_get (iob = tociob, fullwords = CH\$ALLOCATION (.len));
\$str_copy (string = (.len, CH\$PTR (.tociob [iob\$a_data])),
target = sfxstr);
END;
[(.major EQL maj_runoff)
AND (
(.minor EQL min_chapt)
OR (.minor EQL min_chapt_txt)
OR (.minor EQL min_chapt_txt_u)
OR (.minor EQL min_append)
OR (.minor EQL min_append_txt)
OR (.minor EQL min_append_txt_u)
)
]
:
BEGIN
| Process the start of a chapter or appendix

```
: 739      1372 5          LOCAL
: 740      1373 5          ptr;
: 741      1374 5
: 742      1375 5          errcnt = 0;           ! Reset count of CONFUSED messages.
: 743      1376 5          clr_line();        ! Clear the text buffer
: 744      1377 5
: 745      1378 5          IF .cmdblk [contents$v_tms11]
: 746      1379 5          THEN
: 747      1380 6          BEGIN
: 748      1381 6          | TMS output
: 749      1382 6          put ('');
: 750      1383 6
: 751      1384 6
: 752      1385 6
: 753      1386 6          IF .hl_n NEQ 0 THEN put ('[va15]');
: 754      1387 6
: 755      1388 6          IF NOT .toc_page
: 756      1389 6          THEN
: 757      1390 7          literal_text ('*cOn*')
: 758      1391 6          ELSE
: 759      1392 6          literal_text ('*cOnp*');
: 760      1393 6
: 761      1394 6          END
: 762      1395 5          ELSE
: 763      1396 6          BEGIN
: 764      1397 6          | RUNOFF output
: 765      1398 6          put ('.TEST PAGE 5');
: 766      1399 6
: 767      1400 6
: 768      1401 6
: 769      1402 6          IF .cmdblk [contents$g_headers] NEQ 0
: 770      1403 6          THEN
: 771      1404 7          BEGIN
: 772      1405 7
: 773      1406 7          IF .hl_n NEQ 0 THEN put ('.SK2') ELSE put ('.SK1');
: 774      1407 7
: 775      1408 6
: 776      1409 6
: 777      1410 6          IF .indents [0] NEQ 0
: 778      1411 6          THEN
: 779      1412 6          put ($str_concat ('.I-', $str_ascii (.indents [0])));
: 780      1413 6
: 781      1414 6          line_indent = .indents [0];
: 782      1415 6          rmargin = .cmdblk [contents$g_page_width];
: 783      1416 5          END;
: 784      1417 5
: 785      1418 5          |
: 786      1419 5          Now reset the .BLANK information
: 787      1420 5          lasthl = 0;
: 788      1421 5          hl_n = 0;
: 789      1422 5
: 790      1423 5
: 791      1424 6          IF (.minor NEQ min_chapt_txt_u)
: 792      1425 6          AND (.minor NEQ min_append_txu)
: 793      1426 5          THEN
: 794      1427 6          BEGIN
: 795      1428 6          !
```

CONTENTS - Table of Contents formatter program J 8
TOC - Transportable entry point 16-Sep-1984 00:01:05 VAX-11 Bliss-32 v4.0-742
14-Sep-1984 13:05:48 [RUNOFF.SRC]CONTENTS.BLI;1

```
: 796      1429 6           | Insert the word CHAPTER or APPENDIX
: 797      1430 6
: 798      1431 7
: 799      1432 7
: 800      1433 6
: 801      1434 7
: 802      1435 7
: 803      1436 7
: 804      1437 7
: 805      1438 7
: 806      1439 7
: 807      1440 7
: 808      1441 7
: 809      1442 7
: 810      1443 7
: 811      1444 7
: 812      1445 6
: 813      1446 7
: 814      1447 7
: 815      1448 7
: 816      1449 7
: 817      1450 7
: 818      1451 7
: 819      1452 7
: 820      1453 7
: 821      1454 7
: 822      1455 7
: 823      1456 6
: 824      1457 6
: 825      1458 6
: 826      1459 6
: 827      1460 6
: 828      1461 6
: 829      1462 6
: 830      1463 6
: 831      1464 6
: 832      1465 6
: 833      1466 6
: 834      1467 6
: 835      1468 6
: 836      1469 6
: 837      1470 6
: 838      1471 6
: 839      1472 6
: 840      1473 6
: 841      1474 6
: 842      1475 6
: 843      1476 6
: 844      1477 6
: 845      1478 6
: 846      1479 6
: 847      1480 5
: 848      1481 5
: 849      1482 5
: 850      1483 5
: 851      1484 5
: 852      1485 5

:           | Insert the word CHAPTER or APPENDIX
:           IF (.minor EQL min_chapt)
:           OR (.minor EQL min_chapt_txt)
:           THEN
:               BEGIN
:                   | This is the start of a chapter
:                   IF .cmdblk [contents$v_tms11]
:                   THEN
:                       fmttxt (8, CH$PTR (UPLIT ('Chapter ')))
:                   ELSE
:                       fmttxt (8, CH$PTR (UPLIT ('(CHAPTER '))));
:               END
:           ELSE
:               BEGIN
:                   | This is the start of an appendix
:                   IF .cmdblk [contents$v_tms11]
:                   THEN
:                       fmttxt (9, CH$PTR (UPLIT ('Appendix ')))
:                   ELSE
:                       fmttxt (9, CH$PTR (UPLIT ('APPENDIX ')));
:               END;
:           | Insert the prefix string if any
:           IF .prestr [str$h_length] NEQ 0
:           THEN
:               fmttxt (.prestr [str$h_length], .prestr [str$a_pointer]);
:           | Convert the "number" of this chapter/appendix to its visual form.
:           ptrsct = CH$PTR (txtsct);
:           lensct = pacsec (pagen, ptrsct);
:           fmttxt (.lensct, CH$PTR (txtsct));
:           | Insert the suffix string if there is one
:           IF .sfxstr [str$h_length] NEQ 0
:           THEN
:               fmttxt (.sfxstr [str$h_length], .sfxstr [str$a_pointer]);
:           IF NOT .cmdblk [contents$v_tms11] THEN write_char (%C' ', counts_visually);
:           END;
:           IF .cmdblk [contents$v_tms11] THEN write_char (%C'a');
:           | Determine the number of characters in the chapter/appendix title
```

```
: 853      1486  5           | if any.  
: 854      1487  5  
: 855      1488  5  
: 856      1489  5           $xpo_get (iob = tociov, fullwords = 1);  
: 857      1490  5           txtlen = .(tociov [iob$a_data]); !Save text length.  
: 858      1491  5           Now read the chapter/appendix text if  
: 859      1492  5           any exists.  
: 860      1493  5  
: 861      1494  5           IF .txtlen GTR 0  
: 862      1495  5           THEN  
: 863      1496  5           $xpo_get (iob = tociov, fullwords = CH$ALLOCATION (.txtlen));  
: 864      1497  5  
: 865      1498  5  
: 866      1499  5           IF NOT .cmdblk [contents$v_tms11]  
: 867      1500  5           THEN  
: 868      1501  5           pad ((.indents [0] + .indents [1] - .extlin));  
: 869      1502  5  
: 870      1503  5           | Be sure that long chapter titles get wrapped around correctly  
: 871      1504  5           wrap = .extlin;  
: 872      1505  5  
: 873      1506  5  
: 874      1507  5  
: 875      1508  5           IF .toc_page  
: 876      1509  6           THEN  
: 877      1510  6           BEGIN  
: 878      1511  6           | Generate a page number.  
: 879      1512  6           ptrpag = CH$PTR (txtpag);  
: 880      1513  6           IF NOT .cmdblk [contents$v_standard_page]  
: 881      1514  6           THEN  
: 882      1515  6           lenpag = pacbas (.pagen [sct_run_page], ptrpag, 10)  
: 883      1516  6           ELSE  
: 884      1517  6           lenpag = pacpag (pagen, ptrpag);  
: 885      1518  6  
: 886      1519  6  
: 887      1520  6  
: 888      1521  6  
: 889      1522  5           rmargin = .cmdblk [contents$g_page_width] - .lenpag - 1;  
: 890      1523  5           END;  
: 891      1524  5  
: 892      1525  5           | Scan text and put it onto the line.  
: 893      1526  5  
: 894      1527  5           fmetxt (.txtlen, CH$PTR (.tociov [iob$a_data]));  
: 895      1528  5  
: 896      1529  5           IF .toc_page  
: 897      1530  5           THEN  
: 898      1531  6           BEGIN  
: 899      1532  6           | Insert page reference into line  
: 900      1533  6           insref ();  
: 901      1534  6           toc_page = false;  
: 902      1535  6           END  
: 903      1536  6           ELSE  
: 904      1537  6           BEGIN  
: 905      1538  5           IF .cmdblk [contents$v_tms11] THEN write_char (%('a'));  
: 906      1539  6           END;  
: 907      1540  6  
: 908      1541  6  
: 909      1542  5
```

```
: 910      1543 5
: 911      1544 5
: 912      1545 5
: 913      1546 5
: 914      1547 5
: 915      1548 5
: 916      1549 5
: 917      1550 5
: 918      1551 5
: 919      1552 5
: 920      1553 4
: 921      1554 4
: 922      1555 4
: 923      1556 5
: 924      1557 5
: 925      1558 5
: 926      1559 5
: 927      1560 5
: 928      1561 5
: 929      1562 5
: 930      1563 5
: 931      1564 5
: 932      1565 5
: 933      1566 5
: 934      1567 5
: 935      1568 5
: 936      1569 5
: 937      1570 5
: 938      1571 5
: 939      1572 5
: 940      1573 5
: 941      1574 5
: 942      1575 5
: 943      1576 5
: 944      1577 5
: 945      1578 5
: 946      1579 6
: 947      1580 6
: 948      1581 6
: 949      1582 6
: 950      1583 6
: 951      1584 6
: 952      1585 6
: 953      1586 6
: 954      1587 7
: 955      1588 7
: 956      1589 7
: 957      1590 7
: 958      1591 7
: 959      1592 7
: 960      1593 6
: 961      1594 7
: 962      1595 7
: 963      1596 7
: 964      1597 7
: 965      1598 7
: 966      1599 7

Now actually put this line of text into the file
put ((.intlin, CH$PTR (line)));
clr_line ();

Standard skipping after starting chapter/appendix entry
IF NOT .cmdblk [contents$v_tms11] THEN put ('.BLANK');
END;

[.major EQL maj_send_contents] :
BEGIN

    A SEND CONTENTS record

LOCAL
    type,
    position,
    len,
    iob_ptr : REF $xpo_iob ();

type = .minor<0, %BPVAL/2>;
position = .minor<%BPVAL/2, %BPVAL/2>;
$xpo_get (iob = tociob, fullwords = 1);      ! Get text length
len = ..tociob [iob$a_data];

IF .len GTR 0
THEN
    $xpo_get (iob = tociob, fullwords = CH$ALLOCATION (.len));
SELECTONE .position OF
    SET

    [scon_lev]:
    BEGIN

        Insert in main part of table of contents
        iob_ptr = 0;

        IF .type EQL scon_comm
        THEN
            BEGIN
                A command. Insert directly into .RNT file
                put ((.len, CH$PTR (.tociob [iob$a_data])));
            END
        ELSE
            BEGIN
                Text to be inserted into the RNT
                hl_n = 1;          ! Treat as HL 1
            END
    END

```

```
: 967      1600 7           IF .type EQL scon_txt THEN toc_page = true;
: 968      1601 7           Set minor code to unnumbered HL so OUTSEC
: 969      1602 7           won't generate a section number.
: 970      1603 7           minor = min_hl_txt_u;
: 971          7
: 972      1604
: 973      1605
: 974      1606
: 975      1607 7           Initialize formatting parameters
: 976      1608 7           clr_line ();
: 977      1609 7           ext[in = .indents [0];
: 978      1610 7           line_indent = .indents [0];
: 979      1611 7           Call OUTSEC to indent line to proper point
: 980      1612 7           outsec ();
: 981      1613 7           wrap = .extlin;
: 982      1614 7
: 983      1615 7
: 984      1616 7
: 985      1617 7
: 986      1618 7
: 987      1619 7
: 988      1620 8           IF .toc_page
: 989      1621 8           THEN BEGIN
: 990      1622 8               Generate page number to get its length
: 991      1623 8               ptrpag = CH$PTR (txtpag);
: 992      1624 8
: 993      1625 8
: 994      1626 8
: 995      1627 8
: 996      1628 8
: 997      1629 8
: 998      1630 8
: 999      1631 8
: 1000     1632 8
: 1001     1633 7
: 1002     1634 7
: 1003     1635 7
: 1004     1636 7
: 1005     1637 7
: 1006     1638 7
: 1007     1639 7
: 1008     1640 7
: 1009     1641 8
: 1010     1642 8
: 1011     1643 8
: 1012     1644 8
: 1013     1645 7
: 1014     1646 8
: 1015     1647 8
: 1016     1648 8
: 1017     1649 7
: 1018     1650 7
: 1019     1651 7
: 1020     1652 7
: 1021     1653 7
: 1022     1654 6
: 1023     1655 6
: 1024     1656 5           END;
: 1025
: 1026
: 1027
: 1028
: 1029
: 1030
: 1031
: 1032
: 1033
: 1034
: 1035
: 1036
: 1037
: 1038
: 1039
: 1040
: 1041
: 1042
: 1043
: 1044
: 1045
: 1046
: 1047
: 1048
: 1049
: 1050
: 1051
: 1052
: 1053
: 1054
: 1055
: 1056
: 1057
: 1058
: 1059
: 1060
: 1061
: 1062
: 1063
: 1064
: 1065
: 1066
: 1067
: 1068
: 1069
: 1070
: 1071
: 1072
: 1073
: 1074
: 1075
: 1076
: 1077
: 1078
: 1079
: 1080
: 1081
: 1082
: 1083
: 1084
: 1085
: 1086
: 1087
: 1088
: 1089
: 1090
: 1091
: 1092
: 1093
: 1094
: 1095
: 1096
: 1097
: 1098
: 1099
: 1100
: 1101
: 1102
: 1103
: 1104
: 1105
: 1106
: 1107
: 1108
: 1109
: 1110
: 1111
: 1112
: 1113
: 1114
: 1115
: 1116
: 1117
: 1118
: 1119
: 1120
: 1121
: 1122
: 1123
: 1124
: 1125
: 1126
: 1127
: 1128
: 1129
: 1130
: 1131
: 1132
: 1133
: 1134
: 1135
: 1136
: 1137
: 1138
: 1139
: 1140
: 1141
: 1142
: 1143
: 1144
: 1145
: 1146
: 1147
: 1148
: 1149
: 1150
: 1151
: 1152
: 1153
: 1154
: 1155
: 1156
: 1157
: 1158
: 1159
: 1160
: 1161
: 1162
: 1163
: 1164
: 1165
: 1166
: 1167
: 1168
: 1169
: 1170
: 1171
: 1172
: 1173
: 1174
: 1175
: 1176
: 1177
: 1178
: 1179
: 1180
: 1181
: 1182
: 1183
: 1184
: 1185
: 1186
: 1187
: 1188
: 1189
: 1190
: 1191
: 1192
: 1193
: 1194
: 1195
: 1196
: 1197
: 1198
: 1199
: 1200
: 1201
: 1202
: 1203
: 1204
: 1205
: 1206
: 1207
: 1208
: 1209
: 1210
: 1211
: 1212
: 1213
: 1214
: 1215
: 1216
: 1217
: 1218
: 1219
: 1220
: 1221
: 1222
: 1223
: 1224
: 1225
: 1226
: 1227
: 1228
: 1229
: 1230
: 1231
: 1232
: 1233
: 1234
: 1235
: 1236
: 1237
: 1238
: 1239
: 1240
: 1241
: 1242
: 1243
: 1244
: 1245
: 1246
: 1247
: 1248
: 1249
: 1250
: 1251
: 1252
: 1253
: 1254
: 1255
: 1256
: 1257
: 1258
: 1259
: 1260
: 1261
: 1262
: 1263
: 1264
: 1265
: 1266
: 1267
: 1268
: 1269
: 1270
: 1271
: 1272
: 1273
: 1274
: 1275
: 1276
: 1277
: 1278
: 1279
: 1280
: 1281
: 1282
: 1283
: 1284
: 1285
: 1286
: 1287
: 1288
: 1289
: 1290
: 1291
: 1292
: 1293
: 1294
: 1295
: 1296
: 1297
: 1298
: 1299
: 1300
: 1301
: 1302
: 1303
: 1304
: 1305
: 1306
: 1307
: 1308
: 1309
: 1310
: 1311
: 1312
: 1313
: 1314
: 1315
: 1316
: 1317
: 1318
: 1319
: 1320
: 1321
: 1322
: 1323
: 1324
: 1325
: 1326
: 1327
: 1328
: 1329
: 1330
: 1331
: 1332
: 1333
: 1334
: 1335
: 1336
: 1337
: 1338
: 1339
: 1340
: 1341
: 1342
: 1343
: 1344
: 1345
: 1346
: 1347
: 1348
: 1349
: 1350
: 1351
: 1352
: 1353
: 1354
: 1355
: 1356
: 1357
: 1358
: 1359
: 1360
: 1361
: 1362
: 1363
: 1364
: 1365
: 1366
: 1367
: 1368
: 1369
: 1370
: 1371
: 1372
: 1373
: 1374
: 1375
: 1376
: 1377
: 1378
: 1379
: 1380
: 1381
: 1382
: 1383
: 1384
: 1385
: 1386
: 1387
: 1388
: 1389
: 1390
: 1391
: 1392
: 1393
: 1394
: 1395
: 1396
: 1397
: 1398
: 1399
: 1400
: 1401
: 1402
: 1403
: 1404
: 1405
: 1406
: 1407
: 1408
: 1409
: 1410
: 1411
: 1412
: 1413
: 1414
: 1415
: 1416
: 1417
: 1418
: 1419
: 1420
: 1421
: 1422
: 1423
: 1424
: 1425
: 1426
: 1427
: 1428
: 1429
: 1430
: 1431
: 1432
: 1433
: 1434
: 1435
: 1436
: 1437
: 1438
: 1439
: 1440
: 1441
: 1442
: 1443
: 1444
: 1445
: 1446
: 1447
: 1448
: 1449
: 1450
: 1451
: 1452
: 1453
: 1454
: 1455
: 1456
: 1457
: 1458
: 1459
: 1460
: 1461
: 1462
: 1463
: 1464
: 1465
: 1466
: 1467
: 1468
: 1469
: 1470
: 1471
: 1472
: 1473
: 1474
: 1475
: 1476
: 1477
: 1478
: 1479
: 1480
: 1481
: 1482
: 1483
: 1484
: 1485
: 1486
: 1487
: 1488
: 1489
: 1490
: 1491
: 1492
: 1493
: 1494
: 1495
: 1496
: 1497
: 1498
: 1499
: 1500
: 1501
: 1502
: 1503
: 1504
: 1505
: 1506
: 1507
: 1508
: 1509
: 1510
: 1511
: 1512
: 1513
: 1514
: 1515
: 1516
: 1517
: 1518
: 1519
: 1520
: 1521
: 1522
: 1523
: 1524
: 1525
: 1526
: 1527
: 1528
: 1529
: 1530
: 1531
: 1532
: 1533
: 1534
: 1535
: 1536
: 1537
: 1538
: 1539
: 1540
: 1541
: 1542
: 1543
: 1544
: 1545
: 1546
: 1547
: 1548
: 1549
: 1550
: 1551
: 1552
: 1553
: 1554
: 1555
: 1556
: 1557
: 1558
: 1559
: 1560
: 1561
: 1562
: 1563
: 1564
: 1565
: 1566
: 1567
: 1568
: 1569
: 1570
: 1571
: 1572
: 1573
: 1574
: 1575
: 1576
: 1577
: 1578
: 1579
: 1580
: 1581
: 1582
: 1583
: 1584
: 1585
: 1586
: 1587
: 1588
: 1589
: 1590
: 1591
: 1592
: 1593
: 1594
: 1595
: 1596
: 1597
: 1598
: 1599
: 1600
: 1601
: 1602
: 1603
: 1604
: 1605
: 1606
: 1607
: 1608
: 1609
: 1610
: 1611
: 1612
: 1613
: 1614
: 1615
: 1616
: 1617
: 1618
: 1619
: 1620
: 1621
: 1622
: 1623
: 1624
: 1625
: 1626
: 1627
: 1628
: 1629
: 1630
: 1631
: 1632
: 1633
: 1634
: 1635
: 1636
: 1637
: 1638
: 1639
: 1640
: 1641
: 1642
: 1643
: 1644
: 1645
: 1646
: 1647
: 1648
: 1649
: 1650
: 1651
: 1652
: 1653
: 1654
: 1655
: 1656
```

```
1024      1657 5
1025      1658 5
1026      1659 5
1027      1660 5
1028      1661 5
1029      1662 5
1030      1663 5
1031      1664 5
1032      1665 5
1033      1666 5
1034      1667 5
1035      1668 5
1036      1669 6
1037      1670 5
1038      1671 5
1039      1672 6
1040      1673 6
1041      1674 6
1042      1675 6
1043      1676 6
1044      1677 6
1045      1678 6
1046      1679 6
1047      1680 6
1048      1681 6
1049      1682 6
1050      1683 6
1051      1684 6
1052      1685 6
1053      1686 6
P 1054      1687 6
1055      1688 6
1056      1689 6
1057      1690 5
1058      1691 5
1059      1692 4
1060      1693 4
1061      1694 4
1062      1695 5
1063      1696 5
1064      1697 5
1065      1698 5
1066      1699 5
1067      1700 5
1068      1701 5
1069      1702 5
1070      1703 5
1071      1704 5
1072      1705 5
1073      1706 5
1074      1707 5
1075      1708 5
1076      1709 5
1077      1710 5
1078      1711 5
1079      1712 5
1080      1713 5

[.scon_exa]:
    iob_ptr = exiob;           ! Insert with examples

[.scon_fig]:
    iob_ptr = fgiob;           ! Insert with figures

[.scon_tab]:
    iob_ptr = tbiob;           ! Insert with tables

    TES;

    IF (.iob_ptr NEQ 0)
    AND vfycap (new_captions)
    THEN
        BEGIN
            | SEND CONTENTS is to be inserted with FIGURES, TABLES
            | or EXAMPLES. Write record to appropriate temp file.

        LOCAL
            record_header : VECTOR [3];
            record_header [0] = .major;
            record_header [1] = .minor;
            record_header [2] = .len;
            $xpo_put (iob = .iob_ptr, binary_data = (3, record_header));
            IF .len NEQ 0
            THEN
                $xpo_put (iob = .iob_ptr,
                           binary_data = (CH$ALLOCATION (.len), .tociob [iob$a_data]));
            END;

        END;

    [.major EQL maj_send] :
    BEGIN
        Process text given on a .SEND command
        NOTE: When the user said .SEND TOC n, text the value of "n"
        was written to the file too. It is now available as
        the value saved in minor. If the user didn't
        specify n, then 0 was passed.

        NOTE: The value of "n" has the following meanings:
              0   insert directly into .RNT file
              1   Example caption
              -1  insert directly into .RNT file with
                  example captions.
              2   Figure caption
              -2  insert directly into .RNT file with
```

```
: 1081    1714 5
: 1082    1715 5
: 1083    1716 5
: 1084    1717 5
: 1085    1718 5
: 1086    1719 5
: 1087    1720 5
: 1088    1721 5
: 1089    1722 5
: 1090    1723 5
: 1091    1724 5
: 1092    1725 5
: 1093    1726 5
: 1094    1727 5
: 1095    1728 5
: 1096    1729 5
: 1097    1730 5
: 1098    1731 5
: 1099    1732 5
: 1100    1733 5
: 1101    1734 5
: 1102    1735 5
: 1103    1736 5
: 1104    1737 5
: 1105    1738 5
: 1106    1739 5
: 1107    1740 5
: 1108    1741 5
: 1109    1742 5
: 1110    1743 6
: 1111    1744 6
: 1112    1745 6
: 1113    1746 6
: 1114    1747 6
: 1115    1748 6
: 1116    1749 6
: 1117    1750 6
: 1118    1751 6
: 1119    1752 6
: 1120    1753 6
: 1121    1754 6
: 1122    1755 6
: 1123    1756 6
: 1124    1757 6
: 1125    1758 7
: 1126    1759 7
: 1127    1760 7
: 1128    1761 7
: 1129    1762 7
: 1130    1763 7
: 1131    1764 6
: 1132    1765 7
: 1133    1766 7
: 1134    1767 7
: 1135    1768 7
: 1136    1769 8
: 1137    1770 7

example captions.

      3   Table caption
      -3   insert directly into .RNT file with
            table captions

      n>3  Caption with no title or lead-in
      -n   insert directly into .RNT file with
            +n captions

Captions are parsed for a leading word followed by a
comma. This word is used for the figure number.
Captions are treated like header levels in that they
are followed by dots and the chapter-page number.

LOCAL
ptr;

errcnt = 0;           ! Reset count of CONFUSED messages.

Determine the number of characters specified on the .SEND command

$xpo_get (iob = tociob, fullwords = 1);
txtlen = .(tociob [iob$sa_data]); !Save text length.

If there was text given process it.

IF .txtlen GTR 0
THEN
BEGIN
| There was text given.
| Now actually read the text

LOCAL
record_header : VECTOR [3];

$xpo_get (iob = tociob, fullwords = CH$ALLOCATION (.txtlen));
record_header [0] = maj_send;
record_header [1] = .minor;
record_header [2] = .txtlen;

IF .minor EQL 0
THEN
BEGIN
| Insert text directly into .RNT file
insrnt (.txtlen, CH$PTR (.tociob [iob$sa_data]));
END
ELSE
BEGIN
| Miscellaneous caption or send
IF (ABS (.minor) GTR 3) OR ((ABS (.minor) LEQ 3) AND vfycap (old_captions))
THEN
```

```
: 1138      1771  8
: 1139      1772  8
: 1140      1773  8
: 1141      1774  8
: 1142      1775  8
: 1143      1776  8
: 1144      1777  8
: 1145      1778  8
: P 1779  8
: 1146      1780  8
: 1147      1781  7
: 1148      1782  7
: 1149      1783  6
: 1150      1784  6
: 1151      1785  5
: 1152      1786  5
: 1153      1787  4
: 1154      1788  4
: 1155      1789  4
: 1156      1790  5
: 1157      1791  5
: 1158      1792  5
: 1159      1793  5
: 1160      1794  5
: 1161      L 1795  5 %IF %BLISS (BLISS32)
: 1162      1796  5 %THEN
: 1163      1797  5           SIGNAL (contents$_invrecord, 1, tociob [iob$t_resultant]);
: 1164      U 1798  5 %ELSE
: 1165      U 1799  5           $xpo_put_msg (severity = warning,
: 1166      U 1800  5             string = $str_concat ('invalid record group in file ''',
: 1167      U 1801  5             tociob [iob$t_resultant], '''));
: 1168      1802  5 %FI
: 1169      1803  5
: 1170      1804  4           END;
: 1171      1805  4           TES:
: 1172      1806  4
: 1173      1807  3           END;
: 1174      1808  3
: 1175      1809  2           END;
: 1176      1810  2
: 1177      1811  2
: 1178      1812  2           | Finished processing file.
: 1179      1813  2
: 1180      1814  2
: 1181      1815  2           IF .cmdblk [contents$v_log]
: 1182      1816  2           THEN
: 1183      L 1817  2 %IF %BLISS (BLISS32)
: 1184      1818  2 %THEN
: 1185      1819  2           SIGNAL (contents$_complete, 1, tociob [iob$t_resultant]);
: 1186      U 1820  2 %ELSE
: 1187      U 1821  2           $xpo_put_msg (severity = success,
: 1188      U 1822  2             string = $str_concat ('processing complete ''', tociob [iob$t_resultant], '''));
: 1189      1823  2 %FI
: 1190      1824  2
: 1191      1825  2           $xpo_close (iob = tociob);
: 1192      1826  2           RETURN true;
: 1193      1827  1           END;
```

```
          ! Close input file
          ! Successful completion (by definition)
          ! End of toc
```

```

:
.TITLE CONTENTS CONTENTS - Table of Contents formatter
program
.IDENT \V04-000\
.PSECT $PLIT$,NOWRT,NOEXE,2

40 40 78 65 64 6E 5D 35 31 61 76 5B 00000 P.AAA: .BLKB 0
      49 2A 6E 30 63 2A 00000 P.AAB: .ASCII \[va15]\
      32 4B 53 2E 00006 P.AAC: .ASCII \*cOn*Indexaa\
      31 4B 53 2E 00012 P.AAD: .ASCII \.SK2\
      2D 49 2E 00016 P.AAE: .ASCII \.SK1\
      4B 58 45 44 4E 49 0001A P.AAG: .ASCII \.I-\
      4B 4E 41 4C 42 2E 0001D P.AAI: .ASCII \INDEX\
      4B 4E 41 4C 42 2E 00022 P.AAJ: .ASCII \.BLANK\
      00 00 00 2A 6E 30 63 2A 00028 P.AAK: .ASCII \.BLANK\
      5D 35 31 61 76 5B 0002E P.AAL: .BLKB 0
      00 00 00 2A 70 6E 30 63 2A 00034 P.AAM: .ASCII \[va15]\
      00 00 00 2A 70 6E 30 63 2A 0003C P.AAN: .ASCII \*c0n*\<0><0><0>
      35 20 45 47 41 50 20 54 53 45 54 2E 00044 P.AAO: .ASCII \*c0np*\<0><0>
      32 4B 53 2E 00050 P.AAP: .ASCII \.TEST PAGE 5\
      31 4B 53 2E 00054 P.AAQ: .ASCII \.SK2\
      2D 49 2E 00058 P.AAR: .ASCII \.SK1\
      0005B P.AAT: .ASCII \.I-\
      .BLKB 1
      20 72 65 74 70 61 68 43 0005C P.AAV: .ASCII \Chapter \
      20 52 45 54 50 41 48 43 00064 P.AAW: .ASCII \CHAPTER \
      00 00 00 20 78 69 64 6E 65 70 70 41 0006C P.AAX: .ASCII \Appendix \<0><0><0>
      00 00 00 20 58 49 44 4E 45 50 50 41 00078 P.AAY: .ASCII \APPENDIX \<0><0><0>
      4B 4E 41 4C 42 2E 00084 P.AAZ: .ASCII \.BLANK\

.PSECT $OWN$,NOEXE,2

00000 TOCIOB: .BLKB 244
000F4 FILE_TYPE:
      .BLKB 4
      000F8 FGIOB: .BLKB 244
      001EC TBIOB: .BLKB 244
      002E0 EXIOB: .BLKB 244
      003D4 HLCNT: .BLKB 4
      003D8 LASTHL: .BLKB 4
      00000000 003DC TOC_PAGE:
      .LONG 0
      003E0 CAPTYP: .BLKB 4
      003E4 CAPMSG: .BLKB 4
      003E8 ERRCNT: .BLKB 4
      0000 003EC $STR$STRING:
      .WORD 0
      01 0E 003EE .BYTE 14, 1
      00000000 003F0 .ADDRESS P.AAA
      0006 003F4 $STR$STRING:
      .WORD 6
      01 0E 003F6 .BYTE 14, 1
      00000000 003F8 .ADDRESS P.AAB
      000C 003FC $STR$STRING:
      .WORD 12
:

```

01 0E, 003FE .BYTE 14, 1
00000000, 00400 .ADDRESS P.AAC
0004 00404 \$STR\$STRING:
.WORD 4
01 0E, 00406 .BYTE 14, 1
00000000, 00408 .ADDRESS P.AAD
0004 0040C \$STR\$STRING:
.WORD 4
01 0E, 0040E .BYTE 14, 1
00000000, 00410 .ADDRESS P.AAE
0003 00414 \$STR\$STRING0:
.WORD 3
01 0E, 00416 .BYTE 14, 1
00000000, 00418 .ADDRESS P.AAG
0005 0041C \$STR\$STRING:
.WORD 5
01 0E, 0041E .BYTE 14, 1
00000000, 00420 .ADDRESS P.AAI
0006 00424 \$STR\$STRING:
.WORD 6
01 0E, 00426 .BYTE 14, 1
00000000, 00428 .ADDRESS P.AAJ
0006 0042C \$STR\$STRING:
.WORD 6
01 0E, 0042E .BYTE 14, 1
00000000, 00430 .ADDRESS P.AAK
0000 00434 \$STR\$STRING:
.WORD 0
01 0E, 00436 .BYTE 14, 1
00000000, 00438 .ADDRESS P.AAL
0006 0043C \$STR\$STRING:
.WORD 6
01 0E, 0043E .BYTE 14, 1
00000000, 00440 .ADDRESS P.AAM
000C 00444 \$STR\$STRING:
.WORD 12
01 0E, 00446 .BYTE 14, 1
00000000, 00448 .ADDRESS P.AAP
0004 0044C \$STR\$STRING:
.WORD 4
01 0E, 0044E .BYTE 14, 1
00000000, 00450 .ADDRESS P.AAQ
0004 00454 \$STR\$STRING:
.WORD 4
01 0E, 00456 .BYTE 14, 1
00000000, 00458 .ADDRESS P.AAR
0003 0045C \$STR\$STRING0:
.WORD 3
01 0E, 0045E .BYTE 14, 1
00000000, 00460 .ADDRESS P.AAT
0006 00464 \$STR\$STRING:
.WORD 6
01 0E, 00466 .BYTE 14, 1
00000000, 00468 .ADDRESS P.AAZ

IOB\$=
IOB\$=TOCIOB
TOCIOB

IOB\$= TOCIOB
IOB\$= TOCIOB
IOB\$= TOCIOB
IOB\$= TOCIOB
IOB\$= TOCIOB
IOB\$= TOCIOB
IOB\$= TOCIOB
IOB\$= TOCIOB
IOB\$= TOCIOB
IOB\$= TOCIOB
IOB\$= TOCIOB
IOB\$= TOCIOB
IOB\$= TOCIOB
IOB\$= TOCIOB
IOB\$= TOCIOB
IOB\$= TOCIOB
IOB\$= TOCIOB
IOB\$= TOCIOB
IOB\$= TOCIOB
IOB\$= TOCIOB
IOB\$= TOCIOB
IOB\$= TOCIOB
IOB\$= TOCIOB
EXTRN DSRTOC\$_BADVALUE
EXTRN DSRTOC\$_OPENIN, DSRTOC\$_OPENOUT
EXTRN DSRTOC\$_VALERR, DSRTOC\$_CAPTIONS
EXTRN DSRTOC\$_CLOSEQUOT
EXTRN DSRTOC\$_CONFQUAL
EXTRN DSRTOC\$_CTRLCHAR
EXTRN DSRTOC\$_EMPTYIN
EXTRN DSRTOC\$_IGNORED
EXTRN DSRTOC\$_INVINPUT
EXTRN DSRTOC\$_INVRECORD
EXTRN DSRTOC\$_OVERSTRK
EXTRN DSRTOC\$_COMPLETE
EXTRN DSRTOC\$_CREATED
EXTRN DSRTOC\$_IGNORENEW
EXTRN DSRTOC\$_IGNOREOLD
EXTRN DSRTOC\$_PROCFILE
EXTRN DSRTOC\$_TEXTD, DSRTOC\$_TMS11
EXTRN DSRTOC\$_IDENT, CMDBLK
EXTRN TOCOOB, STCIOB, CNTVRL
EXTRN CNTVRP, CHROUT, TMPSTR
EXTRN TXTLEN, HL_N, HLLIST
EXTRN HLDSP, MAJOR, MINOR
EXTRN PAGEN, NXTSTC, RNO_EX
EXTRN RNO_FG, RNO_TB, TMS_EX
EXTRN TMS_FG, TMS_TB, PRESTR
EXTRN SFXSTR, LP, INTLIN
EXTRN EXTLIN, LINE, LENPAG
EXTRN TXTPAG, PTRPAG, LENSCT
EXTRN TXTSCT, PTRSCT, RMARGIN
EXTRN WRAP, LINE INDENT
EXTRN FMTXT, INSREF, HDR_X
EXTRN STCHDR, INSRNT, SPLIT
EXTRN PACBAS, PACPAG, PACSEC
EXTRN PACXXX, OPEN ERROR
EXTRN XPOSGET, XPOSFAILURE
EXTRN XSTSOPY, STR\$FAILURE
EXTRN XPOSPUT, XSTSJOIN

6 9

```

.EXTRN XST$ASCII, XPO$CLOSE
.PSECT SCODE$,NOWRT,2

      OFFC 00000
      5B 00000000G EF 9E 00002 .ENTRY TOC, Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11 0974
      5A 00000000G EF 9E 00009 MOVAB XPO$FAILURE, R11
      59 00000000G EF 9E 00010 MOVAB CMDBLK, R10
      58 00000000G EF 9E 00017 MOVAB IOBS+68, R9
      57 00000000' EF 9E 0001E MOVAB $STR$TARGET, R8
      5E 14 C2 00025 MOVAB IOBS+52, R7
      EF 00 FB 00028 SUBL2 #20, SP
      03 50 E8 0002F CALLS #0, TOCINI
      10C6 31 00032 BLBS R0, 1$
      67 08 B0 00035 1$: MOVW #8, IOBS+52
      02 A7 02 90 00038 MOVB #2, IOBS+54
      F8 A7 06 90 0003C MOVB #6, IOBS+44
      5B DD 00040 PUSHL R11
      7E D4 00042 CLRL -(SP)
      CC A7 9F 00044 PUSHAB IOBS
      00000000G EF 03 FB 00047 CALLS #3, XPO$GET
      00208001 8F 50 D1 0004E CMPL R0, #2129921
      03 13 00055 BEQL 2$
      1075 31 00057 BRW 136$
      52 04 A7 D0 0005A 2$: MOVL TOCI0B+56, R2
      50 D4 0005E CLRL R0
      FFFFFFFF 8F 00C0 C7 D1 00060 CMPL FILE_TYPE, #-1
      24 12 00069 BNEQ 3$
      02 62 D1 0006D INCL R0
      1D 13 00070 CMPL (R2), #2
      BEQL 3$
      67 04 A2 04 A5 00072 MULW3 #4, 4(R2), IOBS+52
      02 A7 02 90 00077 MOVB #2, IOBS+54
      F8 A7 06 90 0007B MOVB #6, IOBS+44
      5B DD 0007F PUSHL R11
      7E D4 00081 CLRL -(SP)
      CC A7 9F 00083 PUSHAB IOBS
      00000000G EF 03 FB 00086 CALLS #3, XPO$GET
      19 50 E9 0008F 3$: BRB 1$
      67 08 B0 00092 BLBC R0, 4$
      02 A7 02 90 00095 MOVW #8, IOBS+52
      F8 A7 06 90 00099 MOVB #2, IOBS+54
      5B DD 0009D PUSHL R11
      7E D4 0009F CLRL -(SP)
      CC A7 9F 000A1 PUSHAB IOBS
      00000000G EF 03 FB 000A4 CALLS #3, XPO$GET
      00000000G EF 62 D0 000AB 4$: MOVL (R2), MAJOR
      00000000G EF 04 A2 D0 000B2 MOVL 4(R2), MINOR
      54 00000000G EF 00 000BA MOVL MAJOR, R4
      01 54 D1 000C1 CMPL R4, #1
      22 12 000C4 BNEQ 6$
      FFFFFFFF 8F 00C0 C7 D1 000C6 CMPL FILE_TYPE, #-1
      03 00000000G EF 09 12 000CF BNEQ 5$
      E8 A7 9F 000DA 5$: CMPL MINOR, #3
      36 13 000D8 BEQL 9$
      PUSHAB TOCI0B+28
      1077
  
```

00000000G	01	DD	000DD	PUSHL	#1		
	8F	DD	000DF	PUSHL	#DSRTOCS_INVINPUT		
	OF	DD	31 000E5	BRW	134\$		
	52	D4	000E8	CLRL	R2		
02	54	D1	000EA	CMPL	R4, #2		
	02	12	000ED	BNEQ	7\$		
	52	D6	000EF	INCL	R2		
01 00000000G	50	D4	000F1	CLRL	R0		
	EF	D1	000F3	CMPL	MINOR, #1		
	02	12	000FA	BNEQ	8\$		
	50	D6	000FC	INCL	R0		
51	52	D2	000FE	MCOML	R2, R1		
50	51	CA	00101	BICL2	R1, R0		
01	50	D1	00104	CMPL	R0, #1		
	0A	12	00107	BNEQ	10\$		
00000000V	EF	00	FB 00109	CALLS	#0, PROCESS_PAGE		
	FF	22	31 00110	BRW	1\$		
05 00000000G	50	D4	00113	CLRL	R0		
	EF	D1	00115	CMPL	MINOR, #5		
	02	12	0011C	BNEQ	11\$		
	50	D6	0011E	INCL	R0		
51	52	D2	00120	MCOML	R2, R1		
50	51	CA	00123	BICL2	R1, R0		
01	50	D1	00126	CMPL	R0, #1		
	03	13	00129	BEQL	12\$		
	01EF	31	0012B	BRW	24\$		
03	6A	03B4	C7 D4 0012E	CLRL	ERRCNT		
	01	E0	00132	BBS	#1, CMDBLK, 13\$		
	009B	31	00136	BRW	16\$		
00000000G	EF	9F	00139	13\$:	PUSHAB	STR\$FAILURE	
	7E	D4	0013F	CLRL	-(SP)		
	58	DD	00141	PUSHL	R8		
03B8	C7	9F	00143	PUSHAB	\$STR\$STRING		
	7E	D4	00147	CLRL	-(SP)		
00000000G	EF	05	FB 00149	CALLS	#5, XST\$COPY		
50	68	3C	00150	MOVZWL	TMPSTR, R0		
00000000G	EF	50	CO 00153	ADDL2	R0, CHROUT		
69	68	9E	0015A	MOVAB	\$I0BS\$OUTPUT, IOBS+68		
E8 A9	07	90	0015D	MOVB	#7, IOBS+44		
	5B	DD	00161	PUSHL	R11		
	7E	D4	00163	CLRL	-(SP)		
BC	A9	9F	00165	PUSHAB	IOBS		
00000000G	EF	03	FB 00168	CALLS	#3, XPOS\$PUT		
00000000G	6A	01	E1 0016F	BBC	#1, CMDBLK, 14\$		
EF	00	FB	00173	CALLS	#0, SPLIT		
00000000G	EF	D5	0017A	14\$:	TSTL	HL N	
	41	13	00180	BEQL	155		
00000000G	EF	9F	00182	PUSHL	STR\$FAILURE		
	7E	D4	00188	CLRL	-(SP)		
	58	DD	0018A	PUSHL	R8		
03C0	C7	9F	0018C	PUSHAB	\$STR\$STRING		
	7E	D4	00190	CLRL	-(SP)		
00000000G	EF	05	FB 00192	CALLS	#5, XST\$COPY		
50	68	3C	00199	MOVZWL	TMPSTR, R0		
00000000G	EF	50	CO 0019C	ADDL2	R0, CHROUT		
69	68	9E	001A3	MOVAB	\$I0BS\$OUTPUT, IOBS+68		
E8 A9	07	90	001A6	MOVB	#7, IOBS+44		

07	00000000G	EF	BC	5B	DD	001AA	PUSHL	R11
				7E	D4	001AC	CLRL	-(SP)
				A9	9F	001AE	PUSHAB	I0B\$
				03	FB	001B1	CALLS	#3, XPOS\$PUT
	00000000G	6A		01	E1	001B8	BBC	#1, CMDBLK, 15\$
				00	FB	001BC	CALLS	#0, SPLIT
	00000000G	EF		FF	9F	001C3	PUSHAB	STR\$FAILURE
				7E	D4	001C9	CLRL	-(SP)
			03C8	58	DD	001CB	PUSHL	R8
				C7	9F	001CD	PUSHAB	\$STR\$STRING
			010A	31	001D1	BRW	22\$	
	00000000G			10	13	001DA	TSTL	HL_N
				EF	D5	001D4	BEQL	17\$
	00000000G			10	13	001DA	PUSHAB	STR\$FAILURE
				EF	9F	001DC	CLRL	-(SP)
				7E	D4	001E2	PUSHL	R8
			03D0	58	DD	001E4	PUSHAB	\$STR\$STRING
				C7	9F	001E6	BRB	18\$
	00000000G			0E	11	001EA	PUSHAB	STR\$FAILURE
				EF	9F	001EC	CLRL	-(SP)
			03D8	7E	D4	001F2	PUSHL	R8
				58	DD	001F4	PUSHAB	\$STR\$STRING
				C7	9F	001F6	CLRL	-(SP)
	00000000G	EF		7E	D4	001FA	PUSHL	R11
				05	FB	001FC	CALLS	#5, XST\$COPY
	00000000G	50		68	3C	00203	MOVZWL	TMPSTR, R0
				EF	50	CO	ADDL2	R0, CHROUT
			E8	69	68	0020D	MOVAB	\$I0B\$OUTPUT, I0B\$+68
		A9		07	90	00210	MOVB	#7, I0B\$+44
				5B	DD	00214	PUSHL	R11
				7E	D4	00216	CLRL	-(SP)
				BC	A9	9F	PUSHAB	I0B\$
	00000000G	EF		03	FB	0021B	CALLS	#3, XPOS\$PUT
		6A		01	E1	00222	BBC	#1, CMDBLK, 19\$
	00000000G	EF		00	FB	00226	CALLS	#0, SPLIT
		50	20	AA	D0	0022D	MOVL	INDENTS, R0
				5C	13	00231	BEQL	20\$
				7E	D4	00233	CLRL	-(SP)
	00000000G	7E	0903	50	DD	00235	PUSHL	R0
				8F	3C	00237	MOVZWL	#2307, -(SP)
		EF		03	FB	0023C	CALLS	#3, XST\$ASCII
				50	DD	00243	PUSHL	R0
	00000000G		03E0	C7	9F	00245	PUSHAB	\$STR\$STRING0
				02	FB	00249	CALLS	#2, XST\$JOIN
	00000000G	EF	00000000G	EF	9F	00250	PUSHAB	STR\$FAILURE
				7E	D4	00256	CLRL	-(SP)
			0101	8F	BB	00258	PUSHR	#^M<R0,R8>
				7E	D4	0025C	CLRL	-(SP)
	00000000G	EF		05	FB	0025E	CALLS	#5, XST\$COPY
		50		68	3C	00265	MOVZWL	TMPSTR, R0
	00000000G	EF		50	CO	00268	ADDL2	R0, CHROUT
		69		68	9E	0026F	MOVAB	\$I0B\$OUTPUT, I0B\$+68
		E8	A9	07	90	00272	MOVB	#7, I0B\$+44
				5B	DD	00276	PUSHL	R11
				7E	D4	00278	CLRL	-(SP)
				BC	A9	9F	PUSHAB	I0B\$
	00000000G	EF		03	FB	0027D	CALLS	#3, XPOS\$PUT
		6A		01	E1	00284	BBC	#1, CMDBLK, 20\$

00000000G	EF	00000000G	00	FB	00288	20\$:	CALLS	#0, SPLIT		1117
			EF	9F	0028F		PUSHAB	STR\$FAILURE		
			7E	D4	00295		CLRL	-(SP)		
			58	DD	00297		PUSHL	R8		
		03E8	C7	9F	00299		PUSHAB	\$STR\$STRING		
			7E	D4	0029D		CLRL	-(SP)		
00000000G	EF		05	FB	0029F		CALLS	#5, XST\$COPY		
50			68	3C	002A6		MOVZWL	TMPSTR, R0		
00000000G	EF		50	CO	002A9		ADDL2	R0, CHROUT		
69			68	9E	002B0		MOVAB	\$IOBS\$OUTPUT, IOBS\$+68		
E8	A9		07	90	002B3		MOVB	#7, IOBS\$+44		
			5B	DD	002B7		PUSHL	R11		
			7E	D4	002B9		CLRL	-(SP)		
		BC	A9	9F	002BB		PUSHAB	IOBS		
07	00000000G	EF	03	FB	002BE		CALLS	#3, XPOSPUT		
	6A		01	E1	002C5		BBC	#1, CMDBLK, 21\$		
00000000G	EF	00000000G	EF	9F	002D0	21\$:	CALLS	#0, SPLIT		1118
			7E	D4	002D6		PUSHAB	STR\$FAILURE		
			58	DD	002D8		CLRL	-(SP)		
		03F0	C7	9F	002DA		PUSHL	R8		
			7E	D4	002DE	22\$:	PUSHAB	\$STR\$STRING		
00000000G	EF		05	FB	002E0		CLRL	-(SP)		
50			68	3C	002E7		CALLS	#5, XST\$COPY		
00000000G	EF		50	CO	002EA		MOVZWL	TMPSTR, R0		
69			68	9E	002F1		ADDL2	R0, CHROUT		
E8	A9		07	90	002F4		MOVAB	\$IOBS\$OUTPUT, IOBS\$+68		
			5B	DD	002F8		MOVB	#7, IOBS\$+44		
			7E	D4	002FA		PUSHL	R11		
		BC	A9	9F	002FC		CLRL	-(SP)		
07	00000000G	EF	03	FB	002FF		PUSHAB	IOBS		
	6A		01	E1	00306		CALLS	#3, XPOSPUT		
00000000G	EF	00000000G	EF	D4	00311	23\$:	BBC	#1, CMDBLK, 23\$		
		03A4	C7	D4	00317		CALLS	#0, SPLIT		
			3A	11	0031B		CLRL	HL_N		1121
53	00000000G	EF	D0	0031D	24\$:		CLRL	LA5THL		1122
			50	D4	00324		BRB	28\$		1059
07			53	D1	00326		MOVL	MINOR, R3		1127
			02	12	00329		CLRL	R0		
			50	D6	0032B		CMPL	R3, #7		
			51	D4	0032D	25\$:	BNEQ	25\$		
	OD		53	D1	0032F		INCL	R0		1128
			02	12	00332		CLRL	R1		
			51	D6	00334		CMPL	R3, #13		
51			50	C8	00336	26\$:	BNEQ	26\$		
			50	D4	00339		INCL	R1		
0A			53	D1	0033B		CLRL	R0, R1		1129
			02	12	0033E		CMPL	R0		
			50	D6	00340		BNEQ	R3, #10		
			51	C8	00342	27\$:	INCL	27\$		
50			52	D2	00345		CLRL	R0		
55			55	CA	00348		BISL2	R1, R0		1126
50			50	D1	0034B		MCOML	R2, R5		
01			09	12	0034E		BICL2	R5, R0		
			00	FB	00350		CMPL	R0, #1		
00000000V	EF		5E	11	00357	28\$:	BNEQ	29\$		
							CALLS	#0, PROCESS_ENTITY_INFO		1135
							BRB	36\$		

51	00000000G	EF	D0	00359	29\$:	MOVL	MINOR, R1	1139
08		53	D4	00360		CLRL	R3	
		51	D1	00362		CMPL	R1, #8	
		02	12	00365		BNEQ	30\$	
		53	D6	00367		INCL	R3	
09		50	D4	00369	30\$:	CLRL	R0	1140
		51	D1	0036B		CMPL	R1, #9	
		02	12	0036E		BNEQ	31\$	
		50	D6	00370		INCL	R0	
50		53	C8	00372	31\$:	BISL2	R3, R0	1141
		53	D4	00375		CLRL	R3	
08		51	D1	00377		CMPL	R1, #11	
		02	12	0037A		BNEQ	32\$	
		53	D6	0037C		INCL	R3	
53		50	C8	0037E	32\$:	BISL2	R0, R3	1142
		50	D4	00381		CLRL	R0	
0C		51	D1	00383		CMPL	R1, #12	
		02	12	00386		BNEQ	33\$	
		50	D6	00388		INCL	R0	
50		53	C8	0038A	33\$:	BISL2	R3, R0	1143
		53	D4	0038D		CLRL	R3	
0E		51	D1	0038F		CMPL	R1, #14	
		02	12	00392		BNEQ	34\$	
		53	D6	00394		INCL	R3	
53		50	C8	00396	34\$:	BISL2	R0, R3	1144
		50	D4	00399		CLRL	R0	
0F		51	D1	0039B		CMPL	R1, #15	
		02	12	0039E		BNEQ	35\$	
		50	D6	003A0		INCL	R0	
50		53	C8	003A2	35\$:	BISL2	R3, R0	1138
55		52	D2	003A5		MCOML	R2, R5	
50		55	CA	003A8		BICL2	R5, R0	
01		50	D1	003AB		CMPL	R0, #1	
		0A	12	003AE		BNEQ	37\$	
00000000V	EF	00	FB	003B0		CALLS	#0, PROCESS_ENTITY_TXT	1150
		FC	7B	31	003B7	36\$:	BRW	1\$
		50	D4	003BA	37\$:	CLRL	R0	1152
00000000G		EF	D5	003BC		TSTL	MINOR	
		02	12	003C2		BNEQ	38\$	
		50	D6	003C4		INCL	R0	
51		52	D2	003C6	38\$:	MCOML	R2, R1	
50		51	CA	003C9		BICL2	R1, R0	
01		50	D1	003CC		CMPL	R0, #1	
		03	13	003CF		BEQL	39\$	
		00AF	31	003D1		BRW	44\$	
		03B4	C7	D4	003D4	39\$:	CLRL	ERRCNT
		67	04	B0	003D8		MOVW	#4, IOBS+52
02	A7	02	90	003DB		MOVW	#2, IOBS+54	1157
F8	A7	06	90	003DF		MOVW	#6, IOBS+44	1161
		5B	DD	003E3		PUSHL	R11	
		7E	D4	003E5		CLRL	-(SP)	
00000000G	EF	CC	A7	9F	003E7	PUSHAB	IOBS	
03A0	C7	03	FB	003EA		CALLS	#3, XPOS\$GET	1162
	04	B7	D0	003F1		MOVL	@TOCIOB+56, HLCNT	
	67	04	B0	003F7		MOVW	#4, IOBS+52	1166
02	A7	02	90	003FA		MOVW	#2, IOBS+54	
F8	A7	06	90	003FE		MOVW	#6, IOBS+44	

CONTENTS V04-000

CONTENTS - Table of Contents formatter program 16-Sep-1984 00:01:05 VAX-11 Bliss-32 V4.0-742
TOC - Transportable entry point 14-Sep-1984 13:05:48 [RUNOFF.SRC]CONTENTS.BLI;1

L 9
16-Sep-1984 00:01:0
14-Sep-1984 13:05:4

VAX-11 Bliss-32 V4.0-742
[RUNOFF.SRC]CONTENTS.BLI:1

Page 30
(2)

67	02	50	04	A5	004E2	MULW3	#4, R0, IOBS\$+52		
	F8	A7	02	90	004E6	MOVB	#2, IOBS\$+54		
		A7	06	90	004EA	MOVB	#6, IOBS\$+44		
			5B	DD	004EE	PUSHL	R11		
			7E	D4	004F0	CLRL	-(SP)		
			A7	9F	004F2	PUSHAB	IOBS\$		
	00000000G	EF	03	FB	004F5	CALLS	#3, XPOSGET		
	04	AA	00000000G	FF	D1	004FC	CMPL	HL N, CMDBLK+4	
			03	15	00504	BLEQ	495	1240	
			0160	31	00506	BRW	54S		
	1C	AA	00000000G	EF	D1	00509	49\$: CMPL	HL N, CMDBLK+28	
			45	14	00511	BGTR	50S	1244	
41	6A	00000000G	EF	01	E0	00513	BBS	#1, CMDBLK, 50\$	
			EF	9F	00517	PUSHAB	STR\$FAILURE	1245	
			7E	D4	0051D	CLRL	-(SP)	1247	
			58	DD	0051F	PUSHL	R8		
			03F8	C7	9F	00521	PUSHAB	\$STR\$STRING	
				7E	D4	00525	CLRL	-(SP)	
	00000000G	EF	05	FB	00527	CALLS	#5, XSTS\$COPY		
	50		68	3C	0052E	MOVZWL	TMPSTR, R0		
	00000000G	EF	50	CO	00531	ADDL2	R0, CHROUT		
	69		68	9E	00538	MOVAB	\$IOBS\$OUTPUT, IOBS\$+68		
	E8	A9	07	90	0053B	MOVB	#7, IOBS\$+44		
			5B	DD	0053F	PUSHL	R11		
			7E	D4	00541	CLRL	-(SP)		
			BC	A9	9F	00543	PUSHAB	IOBS\$	
07	00000000G	EF	03	FB	00546	CALLS	#3, XPOSPUT		
	6A		01	E1	0054D	BBC	#1, CMDBLK, 50\$		
	00000000G	EF	00	FB	00551	CALLS	#0, SPLIT		
	00000000G	EF	00000000G	EF	9E	00558	50\$: MOVAB	LÍNE, LP	
			00000000G	EF	D4	00563	CLRL	INTLÍN	1252
			00000000G	EF	D4	00569	CLRL	EXTLÍN	
	00000000G	EF	20	AA	DO	0056F	MOVL	INDENTS, EXTLÍN	1260
	00000000G	EF	20	AA	DO	00577	MOVL	INDENTS, LÍNE_INDENT	1264
	00000000V	EF	00	FB	0057F	CALLS	#0, OUTSEC	1269	
	00000000G	EF	00000000G	EF	DO	00586	MOVL	EXTLÍN, WRAP	1273
	00000000G	EF	00000000G	EF	9E	00591	MOVAB	TXTPAG, PTRPAG	1279
18	6A		03	E0	0059C	BBS	#3, CMDBLK, 51\$		
			0A	DD	005A0	PUSHL	#10	1281	
			EF	9F	005A2	PUSHAB	PTRPAG	1288	
	7E	00000000G	EF	3C	005A8	MOVZWL	PAGEN+14, -(SP)		
	00000000G	EF	03	FB	005AF	CALLS	#3, PACBÁS		
			13	11	005B6	BRB	52\$		
			00000000G	EF	9F	005B8	51\$: PUSHAB	PTRPAG	
			00000000G	EF	9F	005BE	PUSHAB	PAGEN	1295
	00000000G	EF	02	FB	005C4	CALLS	#2, PACPAG		
	00000000G	EF	50	DO	005CB	MOVL	R0, LENPAG		
50	18	AA	00000000G	EF	C3	005D2	52\$: SUB!3	LENPAG, CMDBLK+24, R0	1301
	0G000000G	EF	FF	A0	9E	005DB	MOVAB	-1(R0), RMARGIN	
			04	A7	DD	005E3	PUSHL	TOCIOB+56	1305
			00000000G	EF	DD	005E6	PUSHL	TXTLEN	
	00000000G	EF	02	FB	005EC	CALLS	#2, FMTXT		
	00000000G	EF	00	FB	005F3	CALLS	#0, INSREF		
	OC	AE	00000000G	EF	B0	005FA	MOVW	INFLÍN, \$STR\$STRING	1309
	OE	AE	00000000G	EF	90	00602	MOVB	#14, \$STR\$STRING+2	1313
	OF	AE	01	90	00606	MOVB	#1, \$STR\$STRING+3		
	10	AE	00000000G	EF	9E	0060A	MOVAB	LÍNF, \$STR\$STRING+4	

		00000000G	EF	9F	00612	PUSHAB	STR\$FAILURE			
			7E	D4	00618	CLRL	-(SP)			
			58	DD	0061A	PUSHL	R8			
			AE	9F	0061C	PUSHAB	\$STR\$STRING			
			7E	D4	0061F	CLRL	-(SP)			
		00000000G	EF	05	FB	00621	CALLS	#5, XST\$COPY		
			68	3C	00628	MOVZWL	TMP\$TR, R0			
		00000000G	EF	50	CO	0062B	ADDL2	R0, CHROUT		
			69	68	9E	00632	MOVAB	\$I0B\$OUTPUT, IOB\$+68		
		E8	A9	07	90	00635	MOVB	#7 IOB\$+44		
			58	DD	00639	PUSHL	R11			
			7E	D4	0063B	CLRL	-(SP)			
			A9	9F	0063D	PUSHAB	IOB\$			
07		00000000G	EF	03	FB	00640	CALLS	#3, XPOS\$PUT		
			6A	01	E1	00647	BBC	#1, CMDBLK, 53\$		
		00000000G	EF	00	FB	0064B	CALLS	#0, SPLIT		
		00000000G	EF	9E	00652	53\$:	MOVAB	LINE, LP		
		00000000G	EF	D4	0065D	CLRL	INTLIN	1314		
		00000000G	EF	D4	00663	CLRL	EXTLIN			
	03A4	C7 00000000G	EF	D0	00669	54\$:	MOVL	HL_N, LASTHL		
			F9C0	31	00672	BRW	1\$	1317		
			51	D4	00675	55\$:	CLRL	R1		
		10 00000000G	EF	D1	00677	CMPL	MINOR, #16	1059		
			02	12	0067E	BNEQ	56\$	1322		
			51	D6	00680	INCL	R1			
			50	D4	00682	56\$:	CLRL	RO		
		13 00000000G	EF	D1	00684	CMPL	MINOR, #19	1323		
			02	12	0068B	BNEQ	57\$			
			50	D6	0068D	INCL	RO			
			51	C8	0068F	57\$:	BISL2	R1, RO		
			53	D2	00692	MCOML	R2, R3	1321		
			50	CA	00695	BICL2	R3, RO			
			01	D1	00698	CMPL	RO, #1			
			03	13	0069B	BEQL	58\$			
			0111	31	0069D	BRW	61\$			
	02	67		04	B0	006A0	58\$:	MOVW	#4, IOB\$+52	
		A7		02	90	006A3	MOVB	#2, IOB\$+54	1333	
	F8	A7		06	90	006A7	MOVB	#6, IOB\$+44		
			5B	DD	006AB	PUSHL	R11			
			7E	D4	006AD	CLRL	-(SP)			
			CC	A7	9F	006AF	PUSHAB	IOB\$		
		00000000G	EF	03	FB	006B2	CALLS	#3, XPOS\$GET		
	03A8	C7	04	B7	D0	006B9	MOVL	@TOC\$IOB\$+56, TOC_PAGE	1334	
			67	04	B0	006BF	MOVW	#4, IOB\$+52	1335	
			02	A7	02	90	006C2	MOVB	#2, IOB\$+54	
		F8	A7	06	90	006C6	MOVB	#6, IOB\$+44		
			5B	DD	006CA	PUSHL	R11			
			7E	D4	006CC	CLRL	-(SP)			
			CC	A7	9F	006CE	PUSHAB	IOB\$		
		00000000G	EF	03	FB	006D1	CALLS	#3, XPOS\$GET		
			67	04	B0	006D8	MOVW	#4, IOB\$+52	1336	
			02	A7	02	90	006DB	MOVB	#2, IOB\$+54	
		F8	A7	06	90	006DF	MOVB	#6, IOB\$+44		
			5B	DD	006E3	PUSHL	R11			
			7E	D4	006E5	CLRL	-(SP)			
			CC	A7	9F	006E7	PUSHAB	IOB\$		
		00000000G	EF	03	FB	006EA	CALLS	#3, XPOS\$GET		

	52	04	B7	D0	006F1	MOVL	@TOCIOB+56, LEN	1337	
	50	03	A2	15	006F5	BLEQ	59\$	1339	
	50	04	C6	006F7	MOVAB	3(R2), R0		1341	
	50	04	A5	006FB	DIVL2	#4, R0			
	50	02	90	00702	MULW3	#4, R0, IOBS\$+52			
67	02	A7	06	90	00706	MOVB	#2, IOBS\$+54		
	F8	A7	5B	DD	0070A	MOVB	#6, IOBS\$+44		
			7E	D4	0070C	PUSHL	R11		
		CC	A7	9F	0070E	CLRL	-(SP)		
	00000000G	EF	03	FB	00711	PUSHAB	IOBS\$		
	OC	AE	52	B0	00718	CALLS	#3, XPOS\$GET		
	OE	AE	0E	90	0071C	MOVW	LEN, \$STR\$STRING		
	OF	AE	01	90	00720	MOVB	#14, \$STR\$STRING+2		
	10	AE	04	A7	00724	MOVL	#1, \$STR\$STRING+3		
	00000000G	EF	9F	00729	PUSHAB	TOCIOB+56, \$STR\$STRING+4			
			7E	D4	0072F	STR\$FAILURE			
	00000000G	EF	9F	00731	CLRL	-(SP)			
		18	AE	9F	00737	PUSHAB	\$STR\$TARGET		
	00000000G	EF	7E	D4	0073A	PUSHAB	\$STR\$STRING		
	67	67	05	FB	0073C	CLRL	-(SP)		
	02	A7	04	B0	00743	CALLS	#5, XST\$COPY	1344	
	F8	A7	02	90	00746	MOVW	#4, IOBS\$+52		
			06	90	0074A	MOVB	#2, IOBS\$+54		
			5B	DD	0074E	MOVB	#6, IOBS\$+44		
			7E	D4	00750	PUSHL	R11		
	00000000G	EF	CC	A7	9F	00752	CLRL	-(SP)	
	52	04	03	FB	00755	PUSHAB	IOBS\$		
			B7	D0	0075C	CALLS	#3, XPOS\$GET		
			21	15	00760	MOVL	ATOCIOB+56, LEN	1347	
		50	03	A2	9E	00762	BLEQ	60\$	1349
		50	04	C6	00766	MOVAB	3(R2), R0		
		50	04	A5	00769	DIVL2	#4, R0	1351	
67	02	A7	02	90	0076D	MULW3	#4, R0, IOBS\$+52		
	F8	A7	06	90	00771	MOVB	#2, IOBS\$+54		
			5B	DD	00775	MOVB	#6, IOBS\$+44		
			7E	D4	00777	PUSHL	R11		
	00000000G	EF	CC	A7	9F	00779	CLRL	-(SP)	
			03	FB	0077C	PUSHAB	IOBS\$		
	OC	AE	52	B0	00783	CALLS	#3, XPOS\$GET		
	OE	AE	0E	90	00787	MOVW	LEN, \$STR\$STRING	1354	
	OF	AE	01	90	0078B	MOVB	#14, \$STR\$STRING+2		
	10	AE	04	A7	0078F	MOVL	#1, \$STR\$STRING+3		
	00000000G	EF	9F	00794	PUSHAB	TOCIOB+56, \$STR\$STRING+4			
			7E	D4	0079A	STR\$FAILURE			
	00000000G	EF	9F	0079C	CLRL	-(SP)			
		18	AE	9F	007A2	PUSHAB	\$STR\$TARGET		
	00000000G	EF	7E	D4	007A5	PUSHAB	\$STR\$STRING		
			05	FB	007A7	CLRL	-(SP)		
		F884	31	007AE	CALLS	#5, XST\$COPY			
	51	00000000G	EF	D0	007B1	BRW	1\$	1059	
			53	D4	007B8	MOVL	MINOR R1	1360	
	03		51	D1	007BA	CLRL	R3		
			02	12	007BD	CMPL	R1, #3		
			53	D6	007BF	BNEQ	62\$		
			50	D4	007C1	INCL	R3		
	11		51	D1	007C3	CLRL	R0		
						CMPL	R1, #17	1361	

07	00000000G	EF	BC	A9	9F	0097F	PUSHAB	I0BS		
	6A			03	FB	00982	CALLS	#3, XPOSPUT		
	EF			01	E1	00989	BBC	#1, CMDBLK, 78\$		
	50		20	00	FB	0098D	CALLS	#0, SPLIT		
				AA	D0	00994	78\$: MOVL	INDENTS, R0	1410	
				5C	13	00998	BEQL	79\$		
				7E	D4	0099A	CLRL	-(SP)	1412	
				50	DD	0099C	PUSHL	R0		
	7E		0903	8F	3C	0099E	MOVZWL	#2307, -(SP)		
	EF			03	FB	009A3	CALLS	#3, XSTSASCII		
				50	DD	009AA	PUSHL	R0		
			0428	C7	9F	009AC	PUSHAB	\$STR\$STRING0		
	00000000G	EF		02	FB	009B0	CALLS	#2, XSTSJOIN		
				EF	9F	009B7	PUSHAB	STR\$FAILURE		
			00000000G	7E	D4	009BD	CLRL	-(SP)		
				8F	BB	009BF	PUSHR	#^M<R0,R8>		
			0101	7E	D4	009C3	CLRL	-(SP)		
	00000000G	EF		05	FB	009C5	CALLS	#5, XSTS\$COPY		
	50			68	3C	009CC	MOVZWL	TMPSTR, R0		
	00000000G	EF		50	C0	009CF	ADDL2	R0, CHROUT		
	69			68	9E	009D6	MOVAB	\$I0BS\$OUTPUT, I0B\$+68		
	E8	A9		07	90	009D9	MOVB	#7, I0B\$+44		
				5B	DD	009DD	PUSHL	R11		
				7E	D4	009DF	CLRL	-(SP)		
			BC	A9	9F	009E1	PUSHAB	I0BS		
07	00000000G	EF		03	FB	009E4	CALLS	#3, XPOSPUT		
	6A			01	E1	009EB	BBC	#1, CMDBLK, 79\$		
	00000000G	EF		00	FB	009EF	CALLS	#0, SPLIT		
	00000000G	EF	20	AA	D0	009F6	79\$: MOVL	INDENTS, LINE INDENT	1414	
	00000000G	EF	18	AA	D0	009FE	MOVBL	CMDBLK+24, RMARGIN	1415	
			03A4	C7	D4	00A06	80\$: CLRL	LASTHL	1421	
	00000000G	EF		D4	00A0A		CLRL	HL N	1422	
	50	00000000G		EF	D0	00A10	MOVL	MINOR, R0	1424	
	12			50	D1	00A17	CMPL	R0, #18		
				03	13	00A1A	BEQL	81\$		
	15			50	D1	00A1C	CMPL	R0, #21	1425	
				03	12	00A1F	81\$: BNEQ	82\$		
			00C0	31	00A21	BRW	92\$			
	03			50	D1	00A24	82\$: CMPL	R0, #3	1431	
				05	13	00A27	BEQL	83\$		
	11			50	D1	00A29	CMPL	R0, #17	1432	
				16	12	00A2C	BNEQ	86\$		
08	6A		01	E1	00A2E	83\$: BBC	#1, CMDBLK, 84\$		1438	
	00000000'	EF		9F	00A32	PUSHAB	P.AAV		1440	
	00000000'	EF	06	11	00A38	BRB	85\$			
	00000000'	EF	08	9F	00A3A	84\$: PUSHAB	P.AAW		1442	
				DD	00A40	85\$: PUSHBL	#8			
			14	11	00A42	BRB	89\$			
08	6A		01	E1	00A44	86\$: BBC	#1, CMDBLK, 87\$		1450	
	00000000'	EF		9F	00A48	PUSHAB	P.AAX		1452	
	00000000'	EF	06	11	00A4E	BRB	88\$			
	00000000'	EF	09	9F	00A50	87\$: PUSHAB	P.AAY		1454	
	00000000G	EF		DD	00A56	88\$: PUSHBL	#9			
	50	00000000G		EF	02	FB	00A58	CALLS	#2, FMTXT	1461
				EF	3C	00A5F	MOVZWL	PRESTR, R0		
				0F	13	00A66	BEQL	90\$		
				EF	DD	00A68	PUSHL	PRESTR+4	1463	

					0A	DD	00B9E	PUSHL	#10				1517		
					EF	9F	00BA0	PUSHAB	PTRPAG						
					EF	3C	00BA6	MOVZWL	PAGEN+14, -(SP)						
					03	FB	00BAD	CALLS	#3, PACBÁS						
					13	11	00BB4	BRB	98\$						
					EF	9F	00BB6	97\$:	PUSHAB	PTRPAG			1519		
					EF	9F	00BBC	PUSHAB	PAGEN						
					02	FB	00BC2	CALLS	#2, PACPAG						
					50	DD	00BC9	98\$:	MOVL	R0, LENPAG					
					EF	C3	00BD0	SUBL3	LENPAG, CMDBLK+24, R0				1521		
					A0	9E	00BD9	MOVAB	-1(R0), RMARGIN						
					04	A7	DD	00BE1	99\$:	PUSHL	TOCIOB+56				
					EF	DD	00BE4	PUSHL	TXTLEN				1527		
					02	FB	00BEA	CALLS	#2, FMTXT						
					C7	E9	00BF1	BLBC	TOC_PAGE, 100\$				1529		
					OD	03A8	00	FB	00BF6	CALLS	#0, INSRÉF		1535		
					EF	03A8	C7	D4	00bfd	CLRL	TOC PAGE		1536		
							18	11	00C01	BRB	101\$		1529		
					6A	01	E1	00C03	100\$:	BBC	#1, CMDBLK, 101\$		1541		
					FF	8F	90	00C07		MOVAB	#64, ALP				
										INCL	LP				
										INCL	INTLIN				
					OC	AE	00000000G	EF	80	00C1B	101\$:	MOVW	INTLIN, \$STR\$STRING		1547
					OE	AE	0E	90	00C23		MOVAB	#14, \$STR\$STRING+2			
					OF	AE	01	90	00C27		MOVAB	#1, \$STR\$STRING+3			
					10	AE	EF	9E	00C2B		MOVAB	LINE, \$STR\$STRING+4			
										PUSHAB	STR\$FAILURE				
										CLRL	-(SP)				
										PUSHL	R8				
										PUSHAB	\$STR\$STRING				
										CLRL	-(SP)				
										CALLS	#5, XST\$COPY				
										MOVZWL	TMPSTR, R0				
										ADDL2	RO, CHROUT				
										MOVAB	\$IOB\$OUTPUT, IOBS+68				
										MOVAB	#7, IOBS+44				
										CLRL	R11				
										PUSHL	-(SP)				
										CLRL	IOBS				
										PUSHAB	#3, XPO\$PUT				
										CALLS	#1, CMDBLK, 102\$				
										CALLS	#0, SPLIT				
										MOVAB	LINE, LP				
										CLRL	INTLIN				
										CLRL	EXTLIN				
										BBS	#1, CMDBLK, 103\$				
										PUSHAB	STR\$FAILURE				
										CLRL	-(SP)				
										PUSHL	R8				
										PUSHAB	\$STR\$STRING				
										CLRL	-(SP)				
										CALLS	#5, XST\$COPY				
										MOVZWL	TMPSTR, R0				
										ADDL2	RO, CHROUT				
										MOVAB	\$IOB\$OUTPUT, IOBS+68				
										MOVAB	#7, IOBS+44				
										CLRL	R11				
										PUSHL					

07	00000000G	EF	BC	7E	D4	00C88	CLRL	-(SP)		
	6A			A9	9F	00CBA	PUSHAB	I0B\$		
		01		03	FB	00CBD	CALLS	#3, XPOSPUT		
				01	E1	00CC4	BBC	#1, CMDBLK, 103\$		
	00000000G	EF		00	FB	00CC8	CALLS	#0, SPLIT		
		04	F363	31	00CCF	103\$:	BRW	1\$	1059	
				54	D1	00CD2	104\$:	CMPL	1555	
				03	13	00CD5	BEQL	R4, #4		
			02AB	31	00CD7		BRW	105\$		
	55	00000000G	EF	3C	00CDA	105\$:	MOVZWL	124\$		
	54	00000000G	EF	3C	00CE1		MOVZWL	MINOR, TYPE	1566	
	67			04	B0	00CE8	MOVW	MINOR+2, POSITION	1567	
	02	A7		02	90	00CEB	MOVB	#4, I0B\$+52	1568	
	F8	A7		06	90	00CEF	MOVB	#2, I0B\$+54		
				5B	DD	00CF3	PUSHL	#6, I0B\$+44		
				7E	D4	00CF5	CLRL	R11		
	00000000G	EF	CC	A7	9F	00CF7	PUSHAB	-(SP)		
	52			03	FB	00CFA	CALLS	I0B\$		
		04		B7	D0	00D01	MOVW	#3, XPOSGET		
				21	15	00D05	BLEQ	@TOCIOB+56, LEN	1569	
	50			03	A2	9E	00D07	MOVAB	106\$	1571
	50			04	C6	00D0B	DIVL2	3(R2), R0	1573	
	02	A7		04	A5	00D0E	MULW3	#4, R0		
	F8	A7		02	90	00D12	MOVB	#4, R0, I0B\$+52		
				06	90	00D16	MOVB	#2, I0B\$+54		
				5B	DD	00D1A	PUSHL	#6, I0B\$+44		
	00000000G	EF	CC	A7	9F	00D1E	CLRL	R11		
	01			03	FB	00D21	PUSHAB	-(SP)		
				54	D1	00D28	CALLS	I0B\$		
				03	13	00D2B	CMPL	#3, XPOSGET		
			01BB	31	00D2D	106\$:	POSITION, #1		1578	
				53	D4	00D30	BEQL	107\$		
		01		55	D1	00D32	BRW	108\$		
				54	12	00D35	CLRL	IOB PTR	1583	
	OC	AE		52	B0	00D37	CMPL	TYPE, #1	1585	
	OE	AE		0E	90	00D3B	BNEQ	109\$		
	OF	AE		01	90	00D3F	MOVW	LEN, \$STR\$STRING	1591	
	10	AE		04	A7	00D43	MOVB	#14, \$STR\$STRING+2		
		00000000G		EF	9F	00D48	MOVB	#1, \$STR\$STRING+3		
				7E	D4	00D4E	MOVL	TOCIOB+56, \$STR\$STRING+4		
				58	DD	00D50	PUSHL	STR\$FAILURE		
			18	AE	9F	00D52	CLRL	-(SP)		
	00000000G	EF		7E	D4	00D55	PUSHAB	R8		
				05	FB	00D57	CLRL	\$STR\$STRING		
				68	3C	00D5E	CALLS	-(SP)		
	00000000G	EF		50	C0	00D61	MOVZWL	#5, XST\$COPY		
				68	9E	00D68	ADDL2	TMPSTR, R0		
		E8	A9	07	90	00D6B	MOVAB	R0, CHROUT		
				5B	DD	00D6F	MOVB	\$I0B\$OUTPUT, I0B\$+68		
				7E	D4	00D71	PUSHL	#7, I0B\$+44		
			BC	A9	9F	00D73	CLRL	R11		
	00000000G	EF		03	FB	00D76	PUSHAB	-(SP)		
				01	E1	00D7D	CALLS	I0B\$		
	6A			00	FB	00D81	BBC	#3, XPOSPUT		
	00000000G	EF		0182	31	00D88	CALLS	#1, CMDBLK, 108\$		
				01	D0	00D8B	BRW	#0, SPLIT		
	00000000G	EF				109\$:	121\$		1585	
							MOVL	#1, HL_N	1598	

			02	55	D1	00D92	CMP	TYPE, #2	1600
			03AB	C7	05	12	00D75	BNEQ	110\$
			00000000G	EF	01	D0	00D97	MOVL	#1, TOC PAGE
			00000000G	EF	06	D0	00D9C	110\$:	MOV
			00000000G		EF	9E	00DA3	MOVAB	#6, MINOR
			00000000G		EF	D4	00DAE	CLRL	LINE, LP
			00000000G		EF	D4	00DB4	CLRL	EXTLIN
			00000000G	EF	20	AA	00DBA	MOVL	INDENTS, EXTLIN
			00000000G	EF	20	AA	00DC2	MOVL	INDENTS, LINE_INDENT
			00000000V	EF	00	FB	00DCA	CALLS	#0, OUTSEC
			00000000G	EF	EF	D0	00DD1	MOVL	EXTLIN, WRAP
			03AB	C7	E9	00DDC	BLBC	TOC PAGE, 113\$	
18			00000000G	43	EF	9E	00DE1	MOVAB	TXTPAG, PTRPAG
			00000000G	EF	00	E0	00DEC	BBS	#3, CMDBLK, 111\$
			00000000G	6A	03	DD	00DF0	PUSHL	#10
			00000000G		EF	9F	00DF2	PUSHAB	PTRPAG
			00000000G	7E	3C	00DF8	MOVZWL	PAGEN+14, -(SP)	
			00000000G	EF	03	FB	00DFF	CALLS	#3, PACBAS
			00000000G		13	11	00E06	BRB	112\$
			00000000G		EF	9F	00E08	111\$:	PUSHAB
			00000000G		EF	9F	00E0E	PUSHAB	PTRPAG
			00000000G	EF	02	FB	00E14	CALLS	PAGEN
			00000000G	EF	50	D0	00E1B	112\$:	MOVL
			00000000G		06	11	00E22	BRB	#2, PACPAG
50	18	AA	00000000G	EF	D4	00E24	113\$:	CLRL	LENPAG
			00000000G	EF	EF	C3	00E2A	114\$:	SUBL3
					A0	9E	00E33	MOVAB	LENPAG, CMDBLK+24, R0
					04	A7	DD	PUSHL	-1(R0), RMARGIN
			00000000G		52	DD	00E3E	PUSHL	TOCIOB+56
			00000000G	EF	02	FB	00E40	CALLS	LEN
			03AB	C7	E9	00E47	BLBC	#2, FMTXT	
			00000000G	EF	00	FB	00E4C	CALLS	TOC PAGE, 115\$
			03AB	C7	D4	00E53	CLRL	#0, INSRREF	
			00000000G		18	11	00E57	TOC PAGE	1642
14		6A	00000000G	EF	01	E1	00E59	115\$:	BRB
			40	FF	8F	90	00E5D	BBC	116\$, CMDBLK, 116\$
			00000000G		EF	D6	00E65	MOVB	#64, ALP
			00000000G		EF	D6	00E6B	INCL	LP
			00000000G	OC	EF	B0	00E71	116\$:	INCL
			00000000G	OE	EF	90	00E79	MOVW	INTLIN, \$STR\$STRING
			00000000G	OF	01	90	00E7D	MOVB	#14, \$STR\$STRING+2
			00000000G	10	AE	9E	00E81	MOVAB	#1, \$STR\$STRING+3
			00000000G		EF	9F	00E89	PUSHAB	LINE, \$STR\$STRING+4
			00000000G		7E	D4	00E8F	CLRL	STR\$FAILURE
					58	DD	00E91	PUSHL	-(SP)
					18	AE	9F	PUSHAB	R8
					7E	D4	00E93	CLRL	\$STR\$STRING
					05	FB	00E96	PUSHL	-(SP)
					50	FB	00E98	CALLS	#5, XST\$COPY
					68	3C	00E9F	MOVZWL	TMPSTR, R0
					68	C0	00EA2	ADDL2	RO, CHROUT
					69	9E	00EA9	MOVAB	\$IOBS\$OUTPUT, IOBS+68
					E8	07	90	MOVB	#7, IOBS+44
					A9	5B	00EAC	PUSHL	R11
					A9	DD	00EB0	CLRL	-(SP)
					7E	D4	00EB2	PUSHAB	IOBS
07			00000000G	EF	BC	A9	9F	CALLS	#3, XPOS\$PUT
				6A	03	FB	00EB4	BBC	#1, CMDBLK, 117\$
					01	E1	00EBE		

000000000G	EF	000000000G	00	FB	00EC2		CALLS	#0, SPLIT	1652	
		000000000G	EF	9E	00EC9	117\$:	MOVAB	LINE, LP		
		000000000G	EF	D4	00ED4		CLRL	INTLIN		
		000000000G	EF	D4	00EDA		CLRL	EXTLIN		
03A4	C7	000000000G	EF	D0	00EE0		MOVL	HL N, LASTHL	1653	
			22	11	00EE9		BRB	12T\$	1575	
	02		54	D1	00EEB	118\$:	CMPL	POSITION, #2	1658	
		02AC	07	12	00EEE		BNEQ	119\$		
	53		C7	9E	00EOF		MOVAB	EXIOB, IOB_PTR	1659	
			16	11	00EF5		BRB	121\$		
	03		54	D1	00EF7	119\$:	CMPL	POSITION, #3	1661	
		00C4	07	12	00EFA		BNEQ	120\$		
	53		C7	9E	00EFC		MOVAB	FGIOB, IOB_PTR	1662	
			OA	11	00F01		BRB	121\$		
	04		54	D1	00F03	120\$:	CMPL	POSITION, #4	1664	
		01B8	05	12	00F06		BNEQ	121\$		
	53		C7	9E	00F08		MOVAB	TBIOB, IOB_PTR	1665	
			53	D5	00F0D	121\$:	TSTL	IOB_PTR	1669	
			OC	13	00F0F		BEQL	122\$		
			01	DD	00F11		PUSHL	#1	1670	
000000000V	EF		01	FB	00F13		CALLS	#1, VFYCAP		
	03		50	E8	00F1A		BLBS	R0, 123\$		
		F115	31	00F1D	122\$:		BRW	1\$		
	08	AE	000000000G	EF	D0	00F20	123\$:	MOVL	MAJOR, RECORD_HEADER	1680
	0C	AE	000000000G	EF	D0	00F28		MOVL	MINOR, RECORD_HEADER+4	1681
	10	AE		52	DO	00F30		MOVL	LEN, RECORD HEADER+8	1682
	6E			0C	B0	00F34		MOVW	#12, \$IOB\$OUTPUT	1683
	02	AE		02	90	00F37		MOVB	#2, \$IOB\$OUTPUT+2	
	03	AE		01	90	00F3B		MOVB	#1, \$IOB\$OUTPUT+3	
	04	AE	08	AE	9E	00F3F		MOVAB	RECORD HEADER, \$IOB\$OUTPUT+4	
	44	A3		6E	9E	00F44		MOVAB	\$IOB\$OUTPUT, 68(IOB_PTR)	
	2C	A3		07	90	00F48		MOVB	#7, 44(IOB_PTR)	
				5B	DD	00F4C		PUSHL	R11	
				7E	D4	00F4E		CLRL	-(SP)	
				53	DD	00F50		PUSHL	IOB_PTR	
000000000G	EF		03	FB	00F52		CALLS	#3, XPOSPUT	1685	
			52	D5	00F59		TSTL	LEN		
			C0	13	00F5B		BEQL	122\$		
	52		03	C0	00F5D		ADDL2	#3, R2	1688	
	52		04	C6	00F60		DIVL2	#4, R2		
	02	AE		04	A5	00F63		MULW3	#4, R2, \$IOB\$OUTPUT	
	03	AE		02	90	00F67		MOVB	#2, \$IOB\$OUTPUT+2	
	04	AE	04	01	90	00F6B		MOVB	#1, \$IOB\$OUTPUT+3	
	44	A3	04	A7	DO	00F6F		MOVL	TO CIOB+56, \$IOB\$OUTPUT+4	
	2C	A3	04	6E	9E	00F74		MOVAB	\$IOB\$OUTPUT, 68(IOB_PTR)	
			07	90	00F78		MOVB	#7, 44(IOB_PTR)		
			5B	DD	00F7C		PUSHL	R11		
			7E	D4	00F7E		CLRL	-(SP)		
			53	DD	00F80		PUSHL	IOB_PTR		
			0121	31	00F82		BRW	132\$		
	03		54	D1	00F85	124\$:	CMPL	R4, #3	1694	
			03	13	00F88		BEQL	125\$		
		03B4	0122	31	00F8A		BRW	133\$		
	67		C7	D4	00F8D	125\$:	CLRL	ERRCNT	1732	
	02	A7	04	B0	00F91		MOVW	#4, IOB\$+52	1736	
	F8	A7	02	90	00F94		MOVB	#2, IOB\$+54		
			06	90	00F98		MOVB	#6, IOB\$+44		

			5B	DD 00F9C	PUSHL	R11			
			7E	D4 00F9E	CLRL	-(SP)			
			A7	9F 00FA0	PUSHAB	I0BS\$			
		CC	03	FB 00FA3	CALLS	#3, XPOSGET			
	00000000G	EF	B7	DO 00FAA	MOVL	ATOCIOB+56, TXTLEN		1737	
	00000000G	EF	50	00000000G	EF	DO 00FB2	MOVL	TXTLEN, R0	1741
			4B	15 00FB9	BLEQ	126\$			
			03	C0 00FB8	ADDL2	#3, R0			
			50	04 C6 00FB8	DIVL2	#4, R0			
			50	04 A5 00FC1	MULW3	#4, R0, I0BS\$+52		1751	
67			50	02 90 00FC5	MOVB	#2, I0BS\$+54			
			50	06 90 00FC9	MOVB	#6, I0BS\$+44			
			5B	DD 00FC0	PUSHL	R11			
			7E	D4 00FCF	CLRL	-(SP)			
		CC	A7	9F 00FD1	PUSHAB	I0BS\$			
	00000000G	EF	03	FB 00FD4	CALLS	#3, XPOSGET			
	08	AE	03	DO 00FDB	MOVL	#3, RECORD_HEADER		1752	
	50	00000000G	EF	DO 00FDF	MOVL	MINOR, R0		1753	
	0C	AE	50	DO 00FE6	MOVL	R0, RECORD_HEADER+4			
	10	AE	00000000G	EF	DO 00FEA	MOVL	TXTLEN, RECORD_HEADER+8		
			50	D5 00FF2	TSTL	R0		1754	
			13	12 00FF4	BNEQ	127\$		1756	
			A7	DD 00FF6	PUSHL	TOCIOB+56		1762	
		04	EF	00000000G	EF	DD 00FF9	PUSHL	TXTLEN	
			F02C	31 01006 126\$:	CALLS	#2, INSRNT			
			03	18 01009 127\$:	BRW	1\$		1756	
			50	CE 0100B	BGEQ	128\$		1769	
			03	50 D1 0100E 128\$:	MNEGL	R0, R0			
			OD	14 01011	CMPL	R0, #3			
			7E	01 CE 01013	BGTR	129\$			
	00000000V	EF	01	FB 01016	MNEGL	#1, -(SP)			
			E6	50 E9 0101D	CALLS	#1, VFYCAP			
			50	DO 01020 129\$:	BLBC	R0, 126\$			
			03	18 01027	MOVL	MINOR, R0		1777	
			50	CE 01029	BGEQ	130\$			
	00000000G	EF	50	D1 0102C 130\$:	MNEGL	R0, R0			
			07	15 01033	CMPL	R0, NXTSTC			
			00000000G	EF	EF 01035	BLEQ	131\$		
			50	DO 0103C 131\$:	MOVL	NXTSTC, R0			
			6E	50 DO 01043	MOVL	R0, NXTSTC		1778	
			02	OC B0 01043	MOVW	#12, \$I0BSOUTPUT			
			AE	02 90 01046	MOVB	#2, \$I0BSOUTPUT+2			
			03	01 90 0104A	MOVB	#1, \$I0BSOUTPUT+3			
			04	AE 9E 0104E	MOVAB	RECORD HEADER, \$I0BSOUTPUT+4			
	00000000G	EF	08	6E 9E 01053	MOVAB	\$I0BSOUTPUT, I0BS\$+68			
	00000000G	EF	00000000G	EF	07 90 0105A	MOVB	#7, I0BS\$+44		
			5B	DD 01061	PUSHL	R11			
			7E	D4 01063	CLRL	-(SP)			
			EF	9F 01065	PUSHAB	I0BS\$			
	00000000G	EF	03	FB 0106B	CALLS	#3, XPOSPUT			
	50 00000000G	EF	03	C1 01072	ADDL3	#3, TXTLEN, R0		1780	
			50	04 C6 0107A	DIVL2	#4, R0			
	6E		04	A5 0107D	MULW3	#4, R0, \$I0BSOUTPUT			
			02	90 01081	MOVB	#2, \$I0BSOUTPUT+2			
			03	01 90 01085	MOVB	#1, \$I0BSOUTPUT+3			
			04	AE A7 DO 01089	MOVL	TOCIOB+56, \$I0BSOUTPUT+4			
	00000000G	EF	04	6E 9E 0108E	MOVAB	\$I0BSOUTPUT, I0BS\$+68			

CONTENTS V04-000

CONTENTS - Table of Contents formatter program 16-Sep-1984 00:01:05 VAX-11 Bliss-32 V4.0-742
TOC - Transportable entry point 14-Sep-1984 13:05:48 [RUNOFF.SRC]CONTENTS.BLI;1

Page 43
(2)

	00000000G	EF	07	90	01095		MOVBL	#7, IOBS+44	
			5B	DD	0109C		PUSHL	R11	
			7E	D4	0109E		CLRL	-(SP)	
	00000000G	EF	03	FB	010A0	132\$:	PUSHAB	IOBS	
			1D	11	010AD		CALLS	#3, XPOSPUT	
			C7	D6	010AF	133\$:	BRB	135\$	1741
	05	03B4	C7	D1	010B3		INCL	ERRCNT	1791
		03B4	12	14	010B8		CMPL	ERRCNT, #5	1793
			E8	A7	9F 010BA		BGTR	135\$	
				01	DD 010BD		PUSHAB	TOCI0B+28	1797
	00000000G	00	8F	DD	010BF		PUSHL	#1	
			03	FB	010C5	134\$:	PUSHL	#DSRTOCS INVRECORD	
			EF66	31	010CC	135\$:	CALLS	#3, LIB\$SIGNAL	
12		6A	05	E1	010CF	136\$:	BRW	1\$	1035
			E8	A7	9F 010D3		BBC	#5, CMDBLK, 137\$	1815
				01	DD 010D6		PUSHAB	TOCI0B+28	1819
	00000000G	00	8F	DD	010D8		PUSHL	#1	
			03	FB	010DE		PUSHL	#DSRTOCS COMPLETE	
	F8	A7	02	90	010E5	137\$:	CALLS	#3, LIB\$SIGNAL	
			5B	DD	010E9		MOVBL	#2, IOBS+44	1825
			7E	D4	010EB		PUSHL	R11	
	00000000G	EF	CC	A7	9F 010ED		CLRL	-(SP)	
		50	03	FB	010FO		PUSHAB	IOBS	
			01	DO	010F7		CALLS	#3, XPOSCLOSE	
			04	010FA			MOVL	#1, RO	1826
			50	D4	010FB	138\$:	RET		
			04	010FD			CLRL	RO	1827

; Routine Size: 4350 bytes, Routine Base: \$CODE\$ + 0000

```
: 1196 1828 1 %SBTTL 'TOCINI - Initialization'  
.: 1197 1829 1 ROUTINE tocini =  
.: 1198 1830 1 ++  
.: 1199 1831 1  
.: 1200 1832 1 FUNCTIONAL DESCRIPTION:  
.: 1201 1833 1  
.: 1202 1834 1 This routine is called by toc () to do initialization.  
.: 1203 1835 1  
.: 1204 1836 1 It opens the input file and determines the input file type (BRN,  
.: 1205 1837 1 BTC or invalid).  
.: 1206 1838 1  
.: 1207 1839 1 The output file, and several temporary files used to store  
.: 1208 1840 1 figure, table and example information are opened if they are not  
.: 1209 1841 1 already open. If they are not already open, the command line  
.: 1210 1842 1 text is prettied up for output to the output file and the output  
.: 1211 1843 1 file prologue is written.  
.: 1212 1844 1  
.: 1213 1845 1 FORMAL PARAMETERS:  
.: 1214 1846 1  
.: 1215 1847 1 None  
.: 1216 1848 1  
.: 1217 1849 1 IMPLICIT INPUTS:  
.: 1218 1850 1  
.: 1219 1851 1 CMDBLK - Command line information block  
.: 1220 1852 1  
.: 1221 1853 1 IMPLICIT OUTPUTS:  
.: 1222 1854 1  
.: 1223 1855 1 CMDBLK - Command line is reformatted  
.: 1224 1856 1 FILE_TYPE - Set to BRN_FILE if .BRN or MAJ_NEWTOC if .BTC  
.: 1225 1857 1 CAPTOP - Set to zero if output files not opened  
.: 1226 1858 1 CAPMSG - Set to zero if output files not opened  
.: 1227 1859 1 NXTSTC - Set to INFINITY if output files not opened  
.: 1228 1860 1 CHROUT - Set to zero if output files not opened  
.: 1229 1861 1 LASTHL - Set to zero if output files not opened  
.: 1230 1862 1 TOCIOB - Input file IOB  
.: 1231 1863 1 TOCOOB - Output file IOB  
.: 1232 1864 1 STCIOB - Send TOC temp file IOB  
.: 1233 1865 1 FGIQB - Header Figure temp file IOB  
.: 1234 1866 1 TBIQB - Header Table temp file IOB  
.: 1235 1867 1 EXIQB - Header Example temp file IOB  
.: 1236 1868 1 LP - CH$PTR (LINE)  
.: 1237 1869 1 INTLIN - Set to zero  
.: 1238 1870 1 EXTLIN - Set to zero  
.: 1239 1871 1  
.: 1240 1872 1 ROUTINE VALUE:  
.: 1241 1873 1 COMPLETION CODES:  
.: 1242 1874 1 None  
.: 1243 1875 1  
.: 1244 1876 1  
.: 1245 1877 1 SIDE EFFECTS:  
.: 1246 1878 1  
.: 1247 1879 1 None  
.: 1248 1880 1 --  
.: 1249 1881 2 BEGIN  
.: 1250 1882 2  
.: 1251 1883 2 LOCAL  
.: 1252 1884 2 parse_block : $xpo_spec_block;
```

```
: 1253      1885 2
.: 1254      1886 2
.: 1255      1887 2
.: 1256      1888 2
.: 1257      1889 2
.: 1258      1890 2
.: 1259      1891 2
.: 1260      1892 2
.: 1261      1893 2
.: 1262      1894 2
P 1263      1895 2
.: 1264      1896 3
.: 1265      1897 2
.: 1266      1898 2
.: 1267      1899 2
.: 1268      1900 2
.: 1269      1901 2
.: 1270      1902 2
.: 1271      1903 2
.: 1272      1904 2
.: 1273      1905 2
.: 1274      1906 3
.: 1275      1907 3
.: 1276      1908 3
.: 1277      1909 3
P 1278      1910 3
P 1279      1911 3
.: 1280      1912 3
.: 1281      1913 3
.: 1282      1914 2
.: 1283      1915 3
.: 1284      1916 3
.: 1285      1917 3
.: 1286      1918 3
P 1287      1919 3
.: 1288      1920 3
.: 1289      1921 3
.: 1290      1922 4
.: 1291      1923 4
.: 1292      1924 4
.: 1293      1925 4
P 1294      1926 4
P 1295      1927 4
P 1296      1928 4
.: 1297      1929 4
.: 1298      1930 3
.: 1299      1931 3
.: 1300      1932 2
.: 1301      1933 2
.: 1302      1934 2
.: 1303      1935 2
L 1304      1936 2
.: 1305      1937 2
.: 1306      1938 2
U 1307      1939 2
U 1308      1940 2
U 1309      1941 2

      | Initialization
      | clr_line ();
      | IF .cmdblk [contents$v_tms11] THEN cmdblk [contents$g_page_width] = tms_line_length;
      | $xpo_iob_init (iob = tociob);
      | IF NOT $xpo_parse_spec (spec_block = parse_block, failure = 0,
      |                         file_spec = cmdblk [contents$t_input_file])
      | THEN
      |   | Parse failed. Set parse_block [xpo$h_file_type] non-zero to force
      |   | quick open failure message.
      |   | parse_block [xpo$h_file_type] = 1;
      | IF .parse_block [xpo$h_file_type] NEQ 0
      | THEN
      |   BEGIN
      |     | User specified a file type. Open file with no defaults.
      |     $xpo_open (iob = tociob, file_spec = cmdblk [contents$t_input_file], attributes = BINARY
      |                %IF %BLISS (BLISS32) %THEN , failure = open_error %FI
      | );
      |   END
      | ELSE
      |   BEGIN
      |     | User did not specify a file type. Try .BRN first.
      |     IF $xpo_open (iob = tociob, file_spec = cmdblk [contents$t_input_file], default = ('.BRN'),
      |                   attributes = BINARY, failure = 0) NEQ xpo$normal
      |     THEN
      |       BEGIN
      |         | No .BRN file. Try .BTC
      |         $xpo_open (iob = tociob, file_spec = cmdblk [contents$t_input_file], default = '.BTC',
      |                    attributes = BINARY
      |                    %IF %BLISS (BLISS32) %THEN , failure = open_error %FI
      | );
      |       END;
      |     END;
      |   END;
      | IF .cmdblk [contents$v_log]
      | THEN
      | %IF %BLISS (BLISS32)
      | %THEN
      |   SIGNAL (contents$procfile, 1, tociob [iob$t_resultant]);
      | ELSE
      |   $xpo_put_msg (severity = success,
      |                 string = $str_concat ('processing file ''', tociob [iob$t_resultant], '''));
```

```
: 1310      1942 2 %FI
: 1311      1943 2
: 1312      1944 2 IF NOT .tocoob [iob$v_open]
: 1313      1945 2 THEN
: 1314          BEGIN
: 1315          1946 3
: 1316          1947 3 Output file not open.
: 1317          1948 3 Open it and do other once only initialization.
: 1318          1949 3
: 1319          1950 3
: 1320          1951 3 LOCAL
: 1321          1952 3     file_type,
: 1322          1953 3     cmd_line : REF $str_descriptor (),
: 1323          1954 3     cmd_len,
: 1324          1955 3     cmd_ptr,
: 1325          1956 3     ptr;
: 1326          1957 3
: 1327          1958 3
: 1328          1959 3
: 1329          1960 3
: 1330          1961 3     captyp = 0;           ! No captions seen yet
: 1331          1962 3     capmsg = false;        ! No message written
: 1332          1963 3     nxtstc = infinity;    ! No .SEND TOC n's seen yet
: 1333          1964 3     chroutr = 0;          ! No characters written to output
: 1334          1965 3     lasthl = 0;           ! Previous HL level
: 1335          1966 3
: 1336          1967 3 IF .cmdblk [contents$v_tms11]
: 1337          1968 3 THEN
: 1338          1969 4     BEGIN
: 1339          1970 4     hl_n = 0;           ! Current HL level
: 1340          1971 4     file_type = CH$PTR (UPLIT ('.TMS')); ! Output file type
: 1341          1972 4     END
: 1342          1973 3 ELSE
: 1343          1974 4     BEGIN
: 1344          1975 4     hl_n = -1;          ! Current HL level
: 1345          1976 4     file_type = CH$PTR (UPLIT ('.RNT')); ! Output file type
: 1346          1977 3     END;
: 1347          1978 3
: 1348          1979 3
: 1349          1980 3
: 1350          1981 3
: 1351          1982 3
: P 1352          1983 3     $xpo_iob_init (iob = tocoob);
: P 1353          1984 3     $xpo_open (iob = tocoob, file_spec = cmdblk [contents$t_output_file], default = (4, .file_type),
: 1354          1985 3             related = tociob [iob$t_resultant], options = output
: 1355          1986 3             %IF %BLISS (BLISS32) %THEN , failure = open_error %FI
: 1356          1987 3
: 1357          1988 3
: 1358          1989 3
: 1359          1990 3
: 1360          1991 3
: 1361          1992 3
: 1362          1993 3
: 1363          1994 3
: 1364          1995 3
: 1365          1996 3
: 1366          1997 3
: 1367          1998 3
: 1368          1999 3
: 1369          2000 3
: 1370          2001 3
: 1371          2002 3
: 1372          2003 3
: 1373          2004 3
: 1374          2005 3
: 1375          2006 3
: 1376          2007 3
: 1377          2008 3
: 1378          2009 3
: 1379          2010 3
: 1380          2011 3
: 1381          2012 3
: 1382          2013 3
: 1383          2014 3
: 1384          2015 3
: 1385          2016 3
: 1386          2017 3
: 1387          2018 3
: 1388          2019 3
: 1389          2020 3
: 1390          2021 3
: 1391          2022 3
: 1392          2023 3
: 1393          2024 3
: 1394          2025 3
: 1395          2026 3
: 1396          2027 3
: 1397          2028 3
: 1398          2029 3
: 1399          2030 3
: 1400          2031 3
: 1401          2032 3
: 1402          2033 3
: 1403          2034 3
: 1404          2035 3
: 1405          2036 3
: 1406          2037 3
: 1407          2038 3
: 1408          2039 3
: 1409          2040 3
: 1410          2041 3
: 1411          2042 3
: 1412          2043 3
: 1413          2044 3
: 1414          2045 3
: 1415          2046 3
: 1416          2047 3
: 1417          2048 3
: 1418          2049 3
: 1419          2050 3
: 1420          2051 3
: 1421          2052 3
: 1422          2053 3
: 1423          2054 3
: 1424          2055 3
: 1425          2056 3
: 1426          2057 3
: 1427          2058 3
: 1428          2059 3
: 1429          2060 3
: 1430          2061 3
: 1431          2062 3
: 1432          2063 3
: 1433          2064 3
: 1434          2065 3
: 1435          2066 3
: 1436          2067 3
: 1437          2068 3
: 1438          2069 3
: 1439          2070 3
: 1440          2071 3
: 1441          2072 3
: 1442          2073 3
: 1443          2074 3
: 1444          2075 3
: 1445          2076 3
: 1446          2077 3
: 1447          2078 3
: 1448          2079 3
: 1449          2080 3
: 1450          2081 3
: 1451          2082 3
: 1452          2083 3
: 1453          2084 3
: 1454          2085 3
: 1455          2086 3
: 1456          2087 3
: 1457          2088 3
: 1458          2089 3
: 1459          2090 3
: 1460          2091 3
: 1461          2092 3
: 1462          2093 3
: 1463          2094 3
: 1464          2095 3
: 1465          2096 3
: 1466          2097 3
: 1467          2098 3
: 1468          2099 3
: 1469          2100 3
: 1470          2101 3
: 1471          2102 3
: 1472          2103 3
: 1473          2104 3
: 1474          2105 3
: 1475          2106 3
: 1476          2107 3
: 1477          2108 3
: 1478          2109 3
: 1479          2110 3
: 1480          2111 3
: 1481          2112 3
: 1482          2113 3
: 1483          2114 3
: 1484          2115 3
: 1485          2116 3
: 1486          2117 3
: 1487          2118 3
: 1488          2119 3
: 1489          2120 3
: 1490          2121 3
: 1491          2122 3
: 1492          2123 3
: 1493          2124 3
: 1494          2125 3
: 1495          2126 3
: 1496          2127 3
: 1497          2128 3
: 1498          2129 3
: 1499          2130 3
: 1500          2131 3
: 1501          2132 3
: 1502          2133 3
: 1503          2134 3
: 1504          2135 3
: 1505          2136 3
: 1506          2137 3
: 1507          2138 3
: 1508          2139 3
: 1509          2140 3
: 1510          2141 3
: 1511          2142 3
: 1512          2143 3
: 1513          2144 3
: 1514          2145 3
: 1515          2146 3
: 1516          2147 3
: 1517          2148 3
: 1518          2149 3
: 1519          2150 3
: 1520          2151 3
: 1521          2152 3
: 1522          2153 3
: 1523          2154 3
: 1524          2155 3
: 1525          2156 3
: 1526          2157 3
: 1527          2158 3
: 1528          2159 3
: 1529          2160 3
: 1530          2161 3
: 1531          2162 3
: 1532          2163 3
: 1533          2164 3
: 1534          2165 3
: 1535          2166 3
: 1536          2167 3
: 1537          2168 3
: 1538          2169 3
: 1539          2170 3
: 1540          2171 3
: 1541          2172 3
: 1542          2173 3
: 1543          2174 3
: 1544          2175 3
: 1545          2176 3
: 1546          2177 3
: 1547          2178 3
: 1548          2179 3
: 1549          2180 3
: 1550          2181 3
: 1551          2182 3
: 1552          2183 3
: 1553          2184 3
: 1554          2185 3
: 1555          2186 3
: 1556          2187 3
: 1557          2188 3
: 1558          2189 3
: 1559          2190 3
: 1560          2191 3
: 1561          2192 3
: 1562          2193 3
: 1563          2194 3
: 1564          2195 3
: 1565          2196 3
: 1566          2197 3
: 1567          2198 3
: 1568          2199 3
: 1569          2200 3
: 1570          2201 3
: 1571          2202 3
: 1572          2203 3
: 1573          2204 3
: 1574          2205 3
: 1575          2206 3
: 1576          2207 3
: 1577          2208 3
: 1578          2209 3
: 1579          2210 3
: 1580          2211 3
: 1581          2212 3
: 1582          2213 3
: 1583          2214 3
: 1584          2215 3
: 1585          2216 3
: 1586          2217 3
: 1587          2218 3
: 1588          2219 3
: 1589          2220 3
: 1590          2221 3
: 1591          2222 3
: 1592          2223 3
: 1593          2224 3
: 1594          2225 3
: 1595          2226 3
: 1596          2227 3
: 1597          2228 3
: 1598          2229 3
: 1599          2230 3
: 1600          2231 3
: 1601          2232 3
: 1602          2233 3
: 1603          2234 3
: 1604          2235 3
: 1605          2236 3
: 1606          2237 3
: 1607          2238 3
: 1608          2239 3
: 1609          2240 3
: 1610          2241 3
: 1611          2242 3
: 1612          2243 3
: 1613          2244 3
: 1614          2245 3
: 1615          2246 3
: 1616          2247 3
: 1617          2248 3
: 1618          2249 3
: 1619          2250 3
: 1620          2251 3
: 1621          2252 3
: 1622          2253 3
: 1623          2254 3
: 1624          2255 3
: 1625          2256 3
: 1626          2257 3
: 1627          2258 3
: 1628          2259 3
: 1629          2260 3
: 1630          2261 3
: 1631          2262 3
: 1632          2263 3
: 1633          2264 3
: 1634          2265 3
: 1635          2266 3
: 1636          2267 3
: 1637          2268 3
: 1638          2269 3
: 1639          2270 3
: 1640          2271 3
: 1641          2272 3
: 1642          2273 3
: 1643          2274 3
: 1644          2275 3
: 1645          2276 3
: 1646          2277 3
: 1647          2278 3
: 1648          2279 3
: 1649          2280 3
: 1650          2281 3
: 1651          2282 3
: 1652          2283 3
: 1653          2284 3
: 1654          2285 3
: 1655          2286 3
: 1656          2287 3
: 1657          2288 3
: 1658          2289 3
: 1659          2290 3
: 1660          2291 3
: 1661          2292 3
: 1662          2293 3
: 1663          2294 3
: 1664          2295 3
: 1665          2296 3
: 1666          2297 3
: 1667          2298 3
: 1668          2299 3
: 1669          2300 3
: 1670          2301 3
: 1671          2302 3
: 1672          2303 3
: 1673          2304 3
: 1674          2305 3
: 1675          2306 3
: 1676          2307 3
: 1677          2308 3
: 1678          2309 3
: 1679          2310 3
: 1680          2311 3
: 1681          2312 3
: 1682          2313 3
: 1683          2314 3
: 1684          2315 3
: 1685          2316 3
: 1686          2317 3
: 1687          2318 3
: 1688          2319 3
: 1689          2320 3
: 1690          2321 3
: 1691          2322 3
: 1692          2323 3
: 1693          2324 3
: 1694          2325 3
: 1695          2326 3
: 1696          2327 3
: 1697          2328 3
: 1698          2329 3
: 1699          2330 3
: 1700          2331 3
: 1701          2332 3
: 1702          2333 3
: 1703          2334 3
: 1704          2335 3
: 1705          2336 3
: 1706          2337 3
: 1707          2338 3
: 1708          2339 3
: 1709          2340 3
: 1710          2341 3
: 1711          2342 3
: 1712          2343 3
: 1713          2344 3
: 1714          2345 3
: 1715          2346 3
: 1716          2347 3
: 1717          2348 3
: 1718          2349 3
: 1719          2350 3
: 1720          2351 3
: 1721          2352 3
: 1722          2353 3
: 1723          2354 3
: 1724          2355 3
: 1725          2356 3
: 1726          2357 3
: 1727          2358 3
: 1728          2359 3
: 1729          2360 3
: 1730          2361 3
: 1731          2362 3
: 1732          2363 3
: 1733          2364 3
: 1734          2365 3
: 1735          2366 3
: 1736          2367 3
: 1737          2368 3
: 1738          2369 3
: 1739          2370 3
: 1740          2371 3
: 1741          2372 3
: 1742          2373 3
: 1743          2374 3
: 1744          2375 3
: 1745          2376 3
: 1746          2377 3
: 1747          2378 3
: 1748          2379 3
: 1749          2380 3
: 1750          2381 3
: 1751          2382 3
: 1752          2383 3
: 1753          2384 3
: 1754          2385 3
: 1755          2386 3
: 1756          2387 3
: 1757          2388 3
: 1758          2389 3
: 1759          2390 3
: 1760          2391 3
: 1761          2392 3
: 1762          2393 3
: 1763          2394 3
: 1764          2395 3
: 1765          2396 3
: 1766          2397 3
: 1767          2398 3
: 1768          2399 3
: 1769          2400 3
: 1770          2401 3
: 1771          2402 3
: 1772          2403 3
: 1773          2404 3
: 1774          2405 3
: 1775          2406 3
: 1776          2407 3
: 1777          2408 3
: 1778          2409 3
: 1779          2410 3
: 1780          2411 3
: 1781          2412 3
: 1782          2413 3
: 1783          2414 3
: 1784          2415 3
: 1785          2416 3
: 1786          2417 3
: 1787          2418 3
: 1788          2419 3
: 1789          2420 3
: 1790          2421 3
: 1791          2422 3
: 1792          2423 3
```

CONTENTS - Table of Contents formatter program C 11
TOCINI - Initialization 16-Sep-1984 00:01:05 VAX-11 Bliss-32 V4.0-742
14-Sep-1984 13:05:48 [RUNOFF.SRC]CONTENTS.BLI;1

```
: 1367      1999 3
: 1368      2000 3
: 1369      2001 3
: 1370      2002 3
: 1371      2003 3
: 1372      2004 3
: 1373      2005 4
: 1374      2006 4
: 1375      2007 4
: 1376      2008 4
: 1377      L 2009 4
: 1378      U 2010 4
: 1379      U 2011 4
: 1380      U 2012 4
: 1381      U 2013 4
: 1382      U 2014 4
: 1383      U 2015 4
: 1384      U 2016 4
: 1385      U 2017 4
: 1386      U 2018 4
: 1387      U 2019 4
: 1388      U 2020 4
: 1389      U 2021 4
: 1390      U 2022 4
: 1391      U 2023 4
: 1392      U 2024 3
: 1393      U 2025 4
: 1394      U 2026 4
: 1395      U 2027 4
: 1396      U 2028 4
: 1397      L 2029 4
: 1398      U 2030 4
: 1399      U 2031 4
: 1400      U 2032 4
: 1401      U 2033 4
: 1402      U 2034 4
: 1403      U 2035 4
: 1404      U 2036 4
: 1405      U 2037 4
: 1406      U 2038 4
: 1407      U 2039 4
: 1408      U 2040 4
: 1409      U 2041 4
: 1410      U 2042 4
: 1411      U 2043 4
: 1412      U 2044 4
: 1413      U 2045 4
: 1414      U 2046 4
: 1415      U 2047 4
: 1416      U 2048 4
: 1417      U 2049 4
: 1418      U 2050 4
: 1419      P 2051 4
: 1420      P 2052 4
: 1421      P 2053 4
: 1422      P 2054 4
: 1423      P 2055 4

    WHILE NOT CH$FAIL (ptr = CH$FIND_CH (.cmd_len, .cmd_ptr, %C';')) DO
        CH$WCHAR (%C'.', .ptr);
        !Change ';' to ','

        IF .cmdblk [contents$v_tms11]
        THEN
            BEGIN
                | Write the TMS prologue

                L %IF DSRPLUS
                U %THEN
                U put ($str_concat ('< CONTENTS version ', (.cntvrl, .cntvrp), ' >'));
                U
                U %ELSE
                U put ($str_concat ('< DSRTOC version ', (.cntvrl, .cntvrp), ' >'));
                U
                U %FI
                U
                put ($str_concat ('< ', cmdblk [contents$t_command_line], ' >'));
                put ('*start*');
                put ('*cinit*');
                END
            ELSE
                BEGIN
                    | Write RUNOFF prologue

                    L %IF DSRPLUS
                    U %THEN
                    U put ($str_concat ('..! CONTENTS version ', (.cntvrl, .cntvrp)));
                    U
                    U %ELSE
                    U put ($str_concat ('..! DSRTOC version ', (.cntvrl, .cntvrp)));
                    U
                    U %FI
                    U
                    put ($str_concat ('..! ', cmdblk [contents$t_command_line]));
                    put ('.SAVE');
                    put ('.NO FLAGS ALL');
                    put ('.NO FLAGS BREAK' NO FLAGS CAPITALIZE NO FLAGS ENDFOOTNOTE');
                    put ('.NO FLAGS HYPHENATE .NO FLAGS INDEX NO FLAGS PERIOD');
                    put ('.NO FLAGS SPACE NO FLAGS SUBSTITUTE');
                    put ('.FLAGS ACCEPT .FLAGS BOLD * FLAGS COMMENT !');
                    put ('.FLAGS LOWERCASE \', .FLAGS OVERSTRIKE % .FLAGS UNDERLINE &');
                    put ('.FLAGS UPPERCASE ^); .NO JUSTIFY');

                    P $str_concat (
                        '.LEFT MARGIN',
                        $str_ascii (.indents [0]),
                        '.RIGHT MARGIN',

```

```
: 1424 P 2056 4           $str_ascii (.cmdblk [contents$g_page_width]),  
: 1425 P 2057 4           .PAGE_SIZE  
: 1426 P 2058 4           $str_ascii (.cmdblk [contents$g_page_width])  
: 1427 P 2059 4           )  
: 1428 P 2060 4           );  
: 1429 P 2061 4           END:  
: 1430 P 2062 3           | Open SEND TOC temp file  
: 1431 P 2063 3           $xpo_iob_init (iob = stciob);  
: 1432 P 2064 3           $xpo_open (iob = stciob, file_spec = 'OCNTST.TMP', attributes = BINARY, options = output  
: 1433 P 2065 3           %IF #BLISS (BLISS32) %THEN , failure = open_error %FI  
: 1434 P 2066 3           );  
: 1435 P 2067 3           | Open figures temp file  
: 1436 P 2068 3           $xpo_iob_init (iob = fgiob);  
: 1437 P 2069 3           $xpo_open (iob = fgiob, file_spec = 'OCNTFG.TMP', attributes = BINARY, options = output  
: 1438 P 2070 3           %IF #BLISS (BLISS32) %THEN , failure = open_error %FI  
: 1439 P 2071 3           );  
: 1440 P 2072 3           | Open tables temp file  
: 1441 P 2073 3           $xpo_iob_init (iob = tbiob);  
: 1442 P 2074 3           $xpo_open (iob = tbiob, file_spec = 'OCNTTB.TMP', attributes = BINARY, options = output  
: 1443 P 2075 3           %IF #BLISS (BLISS32) %THEN , failure = open_error %FI  
: 1444 P 2076 3           );  
: 1445 P 2077 3           | Open examples temp file  
: 1446 P 2078 3           $xpo_iob_init (iob = exiob);  
: 1447 P 2079 3           $xpo_open (iob = exiob, file_spec = 'OCNTEX.TMP', attributes = BINARY, options = output  
: 1448 P 2080 3           %IF #BLISS (BLISS32) %THEN , failure = open_error %FI  
: 1449 P 2081 3           );  
: 1450 P 2082 3           | IF .cmdblk [contents$v_require]  
: 1451 P 2083 3           THEN  
: 1452 P 2084 3           BEGIN  
: 1453 P 2085 3           | Include contents of require file now  
: 1454 P 2086 3           LOCAL  
: 1455 P 2087 3           require_iob : $xpo_iob ();  
: 1456 P 2088 3           $xpo_iob_init (iob = require_iob);  
: 1457 P 2089 3           $xpo_open (iob = require_iob,  
: 1458 P 2090 3           file_spec = cmdblk [contents$t_require_file]  
: 1459 P 2091 3           %IF #BLISS (BLISS32) %THEN , failure = open_error %FI  
: 1460 P 2092 3           );  
: 1461 P 2093 3           IF .cmdblk [contents$v_require]  
: 1462 P 2094 3           THEN  
: 1463 P 2095 4           BEGIN  
: 1464 P 2096 4           | Include contents of require file now  
: 1465 P 2097 4           LOCAL  
: 1466 P 2098 4           require_iob : $xpo_iob ();  
: 1467 P 2099 4           $xpo_iob_init (iob = require_iob);  
: 1468 P 2100 4           $xpo_open (iob = require_iob,  
: 1469 P 2101 4           file_spec = cmdblk [contents$t_require_file]  
: 1470 P 2102 4           %IF #BLISS (BLISS32) %THEN , failure = open_error %FI  
: 1471 P 2103 4           );  
: 1472 P 2104 4           WHILE $xpo_get (iob = require_iob) EQL xpo$_normal DO  
: 1473 P 2105 4           put (require_iob [iob$t_string]);  
: 1474 P 2106 4           $xpo_close (iob = require_iob);  
: 1475 P 2107 4           END  
: 1476 P 2108 4           .  
: 1477 P 2109 4           put (require_iob [iob$t_string]);  
: 1478 P 2110 4           $xpo_close (iob = require_iob);  
: 1479 P 2111 4           END  
: 1480 P 2112 4           END
```

```
: 1481      2113 3      ELSE
: 1482      2114 4      BEGIN
: 1483      2115 4      | No require file - generate standard heading
: 1484      2116 4
: 1485      2117 4
: 1486      2118 4      IF NOT .cmdblk [contents$v_tms11]
: 1487      2119 4      THEN
: 1488      2120 5      BEGIN
: 1489      2121 5      put ('.CENTER;CONTENTS');
: 1490      2122 5      put ('.BLANK');
: 1491      2123 4      END;
: 1492      2124 4
: 1493      2125 3      END;
: 1494      2126 3
: 1495      2127 2      END;
: 1496      2128 2
: 1497      2129 2
: 1498      2130 2      | Read file information record
: 1499      2131 2
: 1500      2132 2      IF $xpo_get (iob = tociob, fullwords = 2) NEQ xpo$normal
: 1501      2133 2      THEN
: 1502      2134 3      BEGIN
: 1503      2135 3      | Empty input file
: 1504      2136 3
: 1505      2137 3
: L 2133 3      %IF %BLISS (BLISS32)
: 2139 3      %THEN
: 2140 3      %ELSE SIGNAL (contents$_emptyin, 1, tociob [iob$t_resultant]);
: 2141 3
: U 2142 3      %ELSE $xpo_put_msg (severity = warning,
: 2143 3      string = $str_concat ('empty input file ''',
: 2144 3      tociob [iob$t_resultant], '''));
: 2145 3      %FI
: 2146 3
: 2147 3      RETURN false;
: 2148 3      END
: 2149 2      ELSE
: 2150 3      BEGIN
: 2151 3      | Determine file type
: 2152 3
: 2153 3      BIND
: 2154 3      codes = .toclob [iob$a_data] : VECTOR;
: 2155 3
: 2156 3      LOCAL
: 2157 3      correct_format;
: 2158 3
: 2159 3      correct_format = true;           ! Assume file format is correct
: 2160 3
: 2161 3      SELECTONE .codes [0] OF
: 2162 3      SET
: 2163 3
: 2164 3      [brn_file] :
: 2165 3      BEGIN
: 2166 4      | Validate .BRN format
: 2167 4
: 2168 4
: 2169 4
```

```
: 1538      2170 4           file_type = brn_file;
: 1539      2171 4
: 1540      2172 4           IF .codes [1] NEQ brn_ident THEN correct_format = false;
: 1541      2173 3           END;
: 1542      2174 3
: 1543      2175 3           [maj_new_toc] :
: 1544      2176 4           BEGIN
: 1545      2177 4           | Validate .BTC format
: 1546      2178 4
: 1547      2179 4
: 1548      2180 4           file_type = maj_new_toc;
: 1549      2181 4
: 1550      2182 4           IF .codes [1] NEQ toc_format THEN correct_format = false;
: 1551      2183 3           END;
: 1552      2184 3
: 1553      2185 3           [OTHERWISE] :
: 1554      2186 3
: 1555      2187 3           | Invalid file format
: 1556      2188 3
: 1557      2189 3           correct_format = false;
: 1558      2190 3
: 1559      2191 3           TES;
: 1560      2192 3
: 1561      2193 3           IF NOT .correct_format
: 1562      2194 3           THEN
: 1563      2195 4           BEGIN
: 1564      2196 4
: 1565      2197 4           | Report bad file format and quit
: 1566      2198 4
: L 1567      2199 4           %IF %BLISS (BLISS32)
: 1568      2200 4           %THEN
: 1569      2201 4           SIGNAL (contents$_invinput, 1, tociob [iob$t_resultant]);
: 1570      U 2202 4           %ELSE
: 1571      U 2203 4           $xpo_put_msg (severity = warning,
: 1572      U 2204 4           string = $str_concat ('invalid file format ''',
: 1573      U 2205 4           tociob [iob$t_resultant], '''));
: 1574      2206 4           %FI
: 1575      2207 4
: 1576      2208 4           RETURN false;
: 1577      2209 3           END;
: 1578      2210 3
: 1579      2211 2           END;
: 1580      2212 2
: 1581      2213 2           RETURN true;
: 1582      2214 1           END;
```

.PSECT \$PLIT\$,NOWRT,NOEXE,2

4E	52	42	2E	0008A	P.ABA:	.ASCII	\.BRN\
43	54	42	2E	0008E	P.ABB:	.ASCII	\.BTC\
53	4D	54	2E	00092		.BLKB	2
54	4E	52	2E	00094	P.ABC:	.ASCII	\.TMS\
54	4E	52	2E	00098	P.ABD:	.ASCII	\.RNT\
53	44	20	3C	0009C	P.ABG:	.ASCII	\< DSRTOC version \
20	6E	000AB					

6F 69 73 72 65 76 20 43 4F 54 52 53 44 20 3C 0009C P.ABG: .ASCII \< DSRTOC version \

0011 0047C \$STR\$STRING0:
01 0E 0047E .WORD 17
00000000' 00480 .BYTE 14, 1
0002 00484 \$STR\$STRING2:
01 0E 00486 .WORD 2
00000000' 00488 .BYTE 14, 1
0002 0048C \$STR\$STRING0:
01 0E 0048E .WORD 2
00000000' 00490 .BYTE 14, 1
0002 00494 \$STR\$STRING2:
01 0E 00496 .WORD 2
00000000' 00498 .BYTE 14, 1
0007 0049C \$STR\$STRING:
01 0E 0049E .WORD 7
00000000' 004A0 .BYTE 14, 1
0007 004A4 \$STR\$STRING:
01 0E 004A6 .WORD 7
00000000' 004A8 .BYTE 14, 1
0012 004AC \$STR\$STRING0:
01 0E 004AE .WORD 18
00000000' 004B0 .BYTE 14, 1
0003 004B4 \$STR\$STRING0:
01 0E 004B6 .WORD 3
00000000' 004B8 .BYTE 14, 1
0005 004BC \$STR\$STRING:
01 0E 004BE .WORD 5
00000000' 004C0 .BYTE 14, 1
000D 004C4 \$STR\$STRING:
01 0E 004C6 .WORD 13
00000000' 004C8 .BYTE 14, 1
003B 004CC \$STR\$STRING:
01 0E 004CE .WORD 59
00000000' 004D0 .BYTE 14, 1
0035 004D4 \$STR\$STRING:
01 0E 004D6 .WORD 53
00000000' 004D8 .BYTE 14, 1
0025 004DC \$STR\$STRING:
01 0E 004DE .WORD 37
00000000' 004E0 .BYTE 14, 1
0030 004E4 \$STR\$STRING:
01 0E 004E6 .WORD 48
00000000' 004E8 .BYTE 14, 1
0039 004EC \$STR\$STRING:

01 0E 004EE .WORD 57
00000000' 004F0 .BYTE 14, 1
0012 004F4 \$STR\$STRING:
01 0E 004F6 .WORD 18
00000000' 004F8 .BYTE 14, 1
000A 004FC \$STR\$STRING:
01 0E 004FE .WORD 10
00000000' 00500 .BYTE 14, 1
0015 00504 \$STR\$STRING:
01 0E 00506 .WORD 21
00000000' 00508 .BYTE 14, 1
000D 0050C \$STR\$STRING0:
01 0E 0050E .WORD 13
00000000' 00510 .BYTE 14, 1
0010 00514 \$STR\$STRING2:
01 0E 00516 .WORD 16
00000000' 00518 .BYTE 14, 1
000E 0051C \$STR\$STRING4:
01 0E 0051E .WORD 14
00000000' 00520 .BYTE 14, 1
000A 00524 \$IOB\$FILE SPEC:
01 0E 00526 .WORD 10
00000000' 00528 .BYTE 14, 1
000A 0052C \$IOB\$FILE SPEC:
01 0E 0052E .WORD 10
00000000' 00530 .BYTE 14, 1
000A 00534 \$IOB\$FILE SPEC:
01 0E 00536 .WORD 10
00000000' 00538 .BYTE 14, 1
000A 0053C \$IOB\$FILE SPEC:
01 0E 0053E .WORD 10
00000000' 00540 .BYTE 14, 1
0010 00544 \$STR\$STRING:
01 0E 00546 .WORD 16
00000000' 00548 .BYTE 14, 1
0006 0054C \$STR\$STRING:
01 0E 0054E .WORD 6
00000000' 00550 .BYTE 14, 1
.ADDRESS P.ACW

IOB\$= TOCJOB
IOB\$RESULTANT= TOCJOB+28
IOB\$= TOCJOB
IOB\$= TOCJOB
IOB\$= TOCJOB

```
$IOB$RELATED=      TOCIOB+28
IOB$=              FGIOB
IOB$RESULTANT=    FGIOB+28
IOB$=              FGIOB
IOB$=              TBIOB
IOB$RESULTANT=    TBIOB+28
IOB$=              TBIOB
IOB$=              EXIOB
IOB$RESULTANT=    EXIOB+28
IOB$=              EXIOB
IOB$=              TOCIOB
.EXTRN XPOS$PARSE SPEC, XPOS$OPEN
.EXTRN XST$FORMAT
.PSECT SCODE$, NOWRT, 2
```

12	00000000G	EF	6A	57	DD 000C3	PUSHL	R7		1934
				03	FB 000C5	CALLS	#3, XPOSOPEN		1938
				05	E1 000CC	4\$:	#5, CMDBLK, 5\$		
				A7	9F 000D0	PUSHAB	TOCIOB+28		
			1C	01	DD 000D3	PUSHL	#1		
	00000000G	00	00000000G	8F	DD 000D5	PUSHL	#DSRTOCS, PROCFILE		
		03	32	03	FB 000DB	CALLS	#3, LIB\$SIGNAL		1944
				A8	E9 000E2	5\$:	BLBC TOCOOB+50, 6\$		
				07F9	31 000E6	BRW	41\$		
				C7	7C 000E9	6\$:	CLRQ CAPTYP		1961
	00000000G	EF	7FFFFFFF	8F	D0 000ED	MOVL	#2147483647, NXTSTC		1963
			03E0	EF	D4 000F8	CLRL	CHROUT		1964
			00000000G	EF	D4 000FE	CLRL	LASTHL		1965
			03D8	C7	D4 00102	BBC	#1, CMDBLK, 7\$		1967
		OF	6A	01	E1 00106	CLRL	HL_N		1970
			00000000G	EF	D4 00106	MOVAB	P.ABC, FILE_TYPE		1971
			56 00000000'	EF	9E 0010C	BRB	8\$		1967
				0E	11 00113	MNEG	#1, HL_N		1975
	00000000G	EF	01	CE 00115	7\$:	MOVAB	P.ABD, FILE_TYPE		1976
		00	56 00000000'	EF	9E 0011C	MOVAB	#0, (SP), #0, #244, IOBS		1982
			6E	00	2C 00123	8\$:	MOVCS		
				68	0012A				
				68	0301003D	MOVL	#50397245, IOBS		
				A8	020E	MOVW	#526, IOB\$RESULTANT+2		
				B0	AD	04	MOVW #4, \$STR\$STRING		1986
				B2	AD	0E	MOVW #14, \$STR\$STRING+2		
				B3	AD	01	MOVW #1, \$STR\$STRING+3		
				B4	AD	56	MOVL FILE_TYPE, \$STR\$STRING+4		
						CLRL -(SP)			
						PUSHAB \$STR\$STRING			
						CLRL -(SP)			
	00000000G	EF		03	FB 0014F	CALLS	#3, XSTSFORMAT		
		04	A8	44	AA 00156	MOVAB	\$IOBSFILE SPEC, IOBS\$+4		
		08	A8	50	D0 0015B	MOVL	R0, IOBS\$+8		
		OC	A8	1C	A7 0015F	MOVAB	\$IOBSRELATED, IOBS\$+12		
		2E	A8	02	88 00164	BISB2	#2, IOBS\$+46		
		2C	A8	01	90 00168	MOVAB	#1, IOBS\$+44		
				00000000G	EF 9F 0016C	PUSHAB	OPEN_ERROR		
					7E D4 00172	CLRL	-(SP)		
					58 DD 00174	PUSHL	R8		
	00000000G	EF		03	FB 00176	CALLS	#3, XPOSOPEN		
		50	54	AA	9E 0017D	MOVAB	CMDBLK+84, CMD LINE		1990
		53		60	3C 00181	MOVZWL	(CMD_LINES), CMD LEN		1991
		52	04	AC	D0 00184	MOVL	4(CMD_LINES), CMD PTR		1992
		62	53	3C	3A 00188	9\$:	#60, CMD_LEN, (CMD_PTR)		1994
				02	12 0018C	BNEQ	10\$		
				51	D4 0018E	CLRL	R1		
				51	D5 00190	10\$:	TSTL PTR		
				06	13 00192	BEQL	11\$		
				61	5B	8F 90 00194	MOVB #91, (PTR)		1995
				EE	11 00198	BRB	9\$		
		62	53	3E	3A 0019A	11\$:	LOCC #62, CMD_LEN, (CMD_PTR)		1997
				02	12 0019E	BNEQ	12\$		
				51	D4 001A0	CLRL	R1		
				51	D5 001A2	12\$:	TSTL PTR		
				06	13 001A4	BEQL	13\$		
				61	5D	8F 90 001A6	MOVB #93, (PTR)		1998
				EE	11 001AA	BRB	11\$		

CONTENTS V04-000

CONTENTS - Table of Contents formatter program 16-Sep-1984 00:01:05 VAX-11 Bliss-32 V4.0-742
TOCINI - Initialization 14-Sep-1984 13:05:48 [RUNOFF.SRC]CONTENTS.BLI;1

L 1

14-Sep-1984 13:05:4

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

Page 56
(3)

				MOVZWL	TMPSTR, R0
				ADDL2	R0, CHROUT
				MOVAB	\$I0B\$OUTPUT, IOBS+68
				MOVB	#7, IOBS+44
				PUSHL	R11
				CLRL	-(SP)
				PUSHL	R8
				CALLS	#3, XPOSPUT
				BBC	#1, CMDBLK, 19\$
				CALLS	#0, SPLIT
				PUSHAB	STR\$FAILURE
				CLRL	-(SP)
				PUSHL	R9
				PUSHAB	\$STR\$STRING
				BRW	33\$
				MOVW	CNTVRL, \$STR\$STRING1
				MOVB	#14, \$STR\$STRING1+2
				MOVB	#1, \$STR\$STRING1+3
				MOVL	CNTVRP, \$STR\$STRING1+4
				PUSHAB	\$STR\$STRING1
				PUSHAB	\$STR\$STRING0
				CALLS	#2, XSTSJOIN
				PUSHAB	STR\$FAILURE
				CLRL	-(SP)
				PUSHR	#^M<R0,R9>
				CLRL	-(SP)
				CALLS	#5, XST\$COPY
				MOVZWL	TMPSTR, R0
				ADDL2	R0, CHROUT
				MOVAB	\$I0B\$OUTPUT, IOBS+68
				MOVB	#7, IOBS+44
				PUSHL	R11
				CLRL	-(SP)
				PUSHL	R8
				CALLS	#3, XPOSPUT
				BBC	#1, CMDBLK, 21\$
				CALLS	#0, SPLIT
				PUSHAB	\$STR\$STRING1
				PUSHAB	\$STR\$STRING0
				CALLS	#2, XSTSJOIN
				PUSHAB	STR\$FAILURE
				CLRL	-(SP)
				PUSHR	#^M<R0,R9>
				CLRL	-(SP)
				CALLS	#5, XST\$COPY
				MOVZWL	TMPSTR, R0
				ADDL2	R0, CHROUT
				MOVAB	\$I0B\$OUTPUT, IOBS+68
				MOVB	#7, IOBS+44
				PUSHL	R11
				CLRL	-(SP)
				PUSHL	R8
				CALLS	#3, XPOSPUT
				BBC	#1, CMDBLK, 22\$
				CALLS	#0, SPLIT
				PUSHAB	STR\$FAILURE
				CLRL	-(SP)

2022

2036

2040

2041

		04BC	59	DD 0038A	PUSHL R9	
			C7	9F 0038C	PUSHAB \$STR\$STRING	
			7E	D4 00390	CLRL -(SP)	
			05	FB 00392	CALLS #5, XST\$COPY	
			69	3C 00399	MOVZWL TMPSTR, R0	
			50	C0 0039C	ADDL2 R0, CHROUT	
			69	9E 003A3	MOVAB \$IOBSOUTPUT, IOBS+68	
			2C	A8	07 90 003A7	MOVB #7 IOBS+44
					5B DD 003AB	PUSHL R11
					7E D4 003AD	CLRL -(SP)
					58 DD 003AF	PUSHL R8
					03 FB 003B1	CALLS #3, XPOSPUT
07					01 E1 003B8	BBC #1, CMDBLK, 23\$
					00 FB 003BC	CALLS #0, SPLIT
				00000000G	EF 9F 003C3	23\$: PUSHAB STR\$FAILURE
					7E D4 003C9	CLRL -(SP)
		04C4	59	DD 003CB	PUSHL R9	
			C7	9F 003CD	PUSHAB \$STR\$STRING	
			7E	D4 003D1	CLRL -(SP)	
			05	FB 003D3	CALLS #5, XST\$COPY	
			69	3C 003DA	MOVZWL TMPSTR, R0	
			50	C0 003DD	ADDL2 R0, CHROUT	
			44	A8	69 9E 003E4	MOVAB \$IOBSOUTPUT, IOBS+68
			2C	A8	07 90 003E8	MOVB #7 IOBS+44
					5B DD 003EC	PUSHL R11
					7E D4 003EE	CLRL -(SP)
					58 DD 003FO	PUSHL R8
07					03 FB 003F2	CALLS #3, XPOSPUT
					01 E1 003F9	BBC #1, CMDBLK, 24\$
				00000000G	EF 9F 00404	24\$: CALLS #0, SPLIT
					7E D4 0040A	PUSHAB STR\$FAILURE
		04CC	59	DD 0040C	CLRL -(SP)	
			C7	9F 0040E	PUSHL R9	
			7E	D4 00412	PUSHAB \$STR\$STRING	
			05	FB 00414	CLRL -(SP)	
			69	3C 0041B	CALLS #5, XST\$COPY	
			50	C0 0041E	MOVZWL TMPSTR, R0	
			44	A8	69 9E 00425	ADDL2 R0, CHROUT
			2C	A8	07 90 00429	MOVAB \$IOBSOUTPUT, IOBS+68
					5B DD 0042D	MOVB #7 IOBS+44
					7E D4 0042F	PUSHL R11
					58 DD 00431	CLRL -(SP)
					03 FB 00433	PUSHL R8
07					01 E1 0043A	CALLS #3, XPOSPUT
				00000000G	EF 9F 00445	25\$: BBC #1, CMDBLK, 25\$
					7E D4 0044B	CALLS #0, SPLIT
		04D4	59	DD 0044D	PUSHAB STR\$FAILURE	
			C7	9F 0044F	CLRL -(SP)	
			05	FB 00455	PUSHL R9	
			69	3C 0045C	PUSHAB \$STR\$STRING	
			50	C0 0045F	CLRL -(SP)	
			44	A8	69 9E 00466	CALLS #5, XST\$COPY
			2C	A8	07 90 0046A	MOVZWL TMPSTR, R0
					5B DD 0046E	ADDL2 R0, CHROUT
						MOVAB \$IOBSOUTPUT, IOBS+68
						MOVB #7 IOBS+44
						PUSHL R11

2042

2043

2044

				7E	D4	00470		CLRL	- (SP)
				58	DD	00472		PUSHL	R8
				03	FB	00474		CALLS	#3, XPOSPUT
				01	E1	00478		BBC	#1, CMDBLK, 26\$
				00	FB	0047F		CALLS	#0, SPLIT
07	00000000G	EF	00000000G	FF	9F	00486	26\$:	PUSHAB	STR\$FAILURE
				7E	D4	0048C		CLRL	- (SP)
			04DC	59	DD	0048E		PUSHL	R9
				C7	9F	00490		PUSHAB	\$STR\$STRING
				7E	D4	00494		CLRL	- (SP)
	00000000G	EF		05	FB	00496		CALLS	#5, XST\$COPY
	50			69	3C	0049D		MOVZWL	TMPSTR, R0
	00000000G	EF		50	CO	004A0		ADDL2	R0, CHROUT
	44	A8		69	9E	004A7		MOVAB	\$I0B\$OUTPUT, IOBS+68
	2C	A8		07	90	004AB		MOVB	#7, IOBS+44
				5B	DD	004AF		PUSHL	R11
				7E	D4	004B1		CLRL	- (SP)
				58	DD	004B3		PUSHL	R8
07	0000C000G	EF		03	FB	004B5		CALLS	#3, XPOSPUT
	6A			01	E1	004BC		BBC	#1, CMDBLK, 27\$
	00000000G	EF	00000000G	00	FB	004C0		CALLS	#0, SPLIT
				FF	9F	004C7	27\$:	PUSHAB	STR\$FAILURE
				7E	D4	004CD		CLRL	- (SP)
			04E4	59	DD	004CF		PUSHL	R9
				C7	9F	004D1		PUSHAB	\$STR\$STRING
				7E	D4	004D5		CLRL	- (SP)
	00000000G	EF		05	FB	004D7		CALLS	#5, XST\$COPY
	50			69	3C	004DE		MOVZWL	TMPSTR, R0
	00000000G	EF		50	CO	004E1		ADDL2	R0, CHROUT
	44	A8		69	9E	004E8		MOVAB	\$I0B\$OUTPUT, IOBS+68
	2C	A8		07	90	004EC		MOVB	#7, IOBS+44
				5B	DD	004FO		PUSHL	R11
				7E	D4	004F2		CLRL	- (SP)
				58	DD	004F4		PUSHL	R8
07	00000000G	EF		03	FB	004F6		CALLS	#3, XPOSPUT
	6A			01	E1	004FD		BBC	#1, CMDBLK, 28\$
	00000000G	EF	00000000G	00	FB	00501		CALLS	#0, SPLIT
				FF	9F	00508	28\$:	PUSHAB	STR\$FAILURE
				7E	D4	0050E		CLRL	- (SP)
			04EC	59	DD	00510		PUSHL	R9
				C7	9F	00512		PUSHAB	\$STR\$STRING
				7E	D4	00516		CLRL	- (SP)
	00000000G	EF		05	FB	00518		CALLS	#5, XST\$COPY
	50			69	3C	0051F		MOVZWL	TMPSTR, R0
	00000000G	EF		50	CO	00522		ADDL2	R0, CHROUT
	44	A8		69	9E	00529		MOVAB	\$I0B\$OUTPUT, IOBS+68
	2C	A8		07	90	0052D		MOVB	#7, IOBS+44
				5B	DD	00531		PUSHL	R11
				7E	D4	00533		CLRL	- (SP)
				58	DD	00535		PUSHL	R8
07	00000000G	EF		03	FB	00537		CALLS	#3, XPOSPUT
	6A			01	E1	0053E		BBC	#1, CMDBLK, 29\$
	00000000G	EF	00000000G	00	FB	00542		CALLS	#0, SPLIT
				FF	9F	00549	29\$:	PUSHAB	STR\$FAILURE
				7E	D4	0054F		CLRL	- (SP)
			04F4	59	DD	00551		PUSHL	R9
				C7	9F	00553		PUSHL	\$STR\$STRJNG

			7E D4 00557	CLRL -(SP)		
			05 FB 00559	CALLS #5, XST\$COPY		
			69 3C 00560	MOVZWL TMPSTR, R0		
			50 C0 00563	ADDL2 R0, CHROUT		
			69 9E 0056A	MOVAB SIOB\$OUTPUT, IOB\$+68		
			07 90 0056E	MOVB #7 IOB\$+44		
			5B DD 00572	PUSHL R11		
			7E D4 00574	CLRL -(SP)		
			58 DD 00576	PUSHL R8		
07	00000000G	EF	03 FB 00578	CALLS #3, XPO\$PUT		
			01 E1 0057F	BBC #1, CMDBLK, 30\$		
	00000000G	EF	00 FB 00583	CALLS #0, SPLIT		
			EF 9F 0058A	00000000G	30\$: PUSHAB STR\$FAILURE	2049
			7E D4 00590	CLRL -(SP)		
			59 DD 00592	PUSHL R9		
		04FC	C7 9F 00594	PUSHAB \$STR\$STRING		
			7E D4 00598	CLRL -(SP)		
	00000000G	EF	05 FB 0059A	CALLS #5, XST\$COPY		
			69 3C 005A1	MOVZWL TMPSTR, R0		
	00000000G	EF	50 C0 005A4	ADDL2 R0, CHROUT		
			69 9E 005AB	MOVAB SIOB\$OUTPUT, IOB\$+68		
			07 90 005AF	MOVB #7 IOB\$+44		
			5B DD 005B3	PUSHL R11		
			7E D4 005B5	CLRL -(SP)		
			58 DD 005B7	PUSHL R8		
07	00000000G	EF	03 FB 005B9	CALLS #3, XPO\$PUT		
			01 E1 005C0	BBC #1, CMDBLK, 31\$		
	00000000G	EF	00 FB 005C4	CALLS #0, SPLIT		
			EF 9F 005CB	00000000G	31\$: PUSHAB STR\$FAILURE	2050
			7E D4 005D1	CLRL -(SP)		
			59 DD 005D3	PUSHL R9		
		0504	C7 9F 005D5	PUSHAB \$STR\$STRING		
			7E D4 005D9	CLRL -(SP)		
	00000000G	EF	05 FB 005DB	CALLS #5, XST\$COPY		
			69 3C 005E2	MOVZWL TMPSTR, R0		
	00000000G	EF	50 C0 005E5	ADDL2 R0, CHROUT		
			69 9E 005EC	MOVAB SIOB\$OUTPUT, IOB\$+68		
			07 90 005F0	MOVB #7 IOB\$+44		
			5B DD 005F4	PUSHL R11		
			7E D4 005F6	CLRL -(SP)		
			58 DD 005F8	PUSHL R8		
07	00000000G	EF	03 FB 005FA	CALLS #3, XPO\$PUT		
			01 E1 00601	BBC #1, CMDBLK, 32\$		
	00000000G	EF	00 FB 00605	CALLS #0, SPLIT		
			7E D4 0060C	32\$: CLRL -(SP)		
		20	AA DD 0060E	PUSHL INDENTS		
	00000000G	EF	0903 8F 3C 00611	MOVZWL #2307, -(SP)		
			03 FB 00616	CALLS #3, XST\$ASCII		
			50 D0 0061D	MOVL R0, R3		
			7E D4 00620	CLRL -(SP)		
		18	AA DD 00622	PUSHL CMDBLK+24		
	00000000G	EF	0903 8F 3C 00625	MOVZWL #2307, -(SP)		
			03 FB 0062A	CALLS #3, XST\$ASCII		
			50 D0 00631	MOVL R0, R2		
			7E D4 00634	CLRL -(SP)		
		18	AA DD 00636	PUSHL CMDBLK+24		
	00000000G	EF	0903 8F 3C 00639	MOVZWL #2307, -(SP)		

00000000G	EF	03	FB	0063E	CALLS #3, XSTS\$ASCII		2067
		50	DD	00645	PUSHL	R0	
	051C	C7	9F	00647	PUSHAB	\$STR\$STRING4	
	0514	C7	9F	0064B	PUSHL	R2	
	0514	C7	9F	0064D	PUSHAB	\$STR\$STRING2	
	050C	C7	9F	00651	PUSHL	R3	
00000000G	EF	050C	C7	9F	00653	PUSHAB	\$STR\$STRING0
	00000000G	EF	06	FB	00657	CALLS	#6, XSTS\$JOIN
	0201	7E	D4	00664	PUSHAB	STR\$FAILURE	
00000000G	EF	00000000G	EF	9F	0065E	CLRL	-(SP)
	0201	8F	BB	00666	PUSHR	#^M<R0,R9>	
00000000G	EF	00000000G	EF	7E	D4	0066A	33\$: CLRL
	00000000G	EF	05	FB	0066C	CALLS	#5, XSTS\$COPY
	50	69	3C	00673	MOVZWL	TMPSTR, R0	
00000000G	EF	50	CO	00676	ADDL2	R0, CHROUT	
44	A8	69	9E	0067D	MOVAB	SIOB\$OUTPUT, IOBS+68	
2C	A8	07	90	00681	MOVB	#7, IOBS+44	
		5B	DD	00685	PUSHL	R11	
		7E	D4	00687	CLRL	-(SP)	
		58	DD	00689	PUSHL	R8	
07	00000000G	EF	03	FB	0068B	CALLS	#3, XPOS\$PUT
	00000000G	EF	6A	01	E1	BBC	#1, CMDBLK, 34\$
	00000000G	EF	00	FB	00696	CALLS	#0, SPLIT
00F4	8F	00	6E	00	2C	0069D	34\$: MOVCS #0, (SP), #0, #244, IOBS
		00000000G	EF	006A4	MOVL	#50397245, IOBS	
		00000000G	EF	0301003D	MOVW	#526, IOB\$RESULTANT+2	
		00000000G	EF	020E	MOVAB	SIOB\$FILE SPEC, IOBS+4	2070
		00000000G	EF	0524	BISL2	#65538, IOBS+46	
		00000000G	EF	00010002	MOVB	#1, IOBS+44	
		00000000G	EF	00000000G	PUSHAB	OPEN ERROR	
		00000000G	EF	006A9	CLRL	-(SP)	
		00000000G	EF	8F	PUSHAB	IOBS	
00F4	8F	00	6E	00	2C	006ED	CALLS #3, XPOS\$OPEN
		00000000G	EF	006E6	MOVCS	#0, (SP), #0, #244, IOBS	2074
		00000000G	EF	006F4	MOVL	#50397245, IOBS	
		00000000G	EF	0301003D	MOVW	#526, IOB\$RESULTANT+2	
		00000000G	EF	020E	MOVAB	SIOB\$FILE SPEC, IOBS+4	2077
		00000000G	EF	052C	BISL2	#65538, IOBS+46	
		00000000G	EF	00010002	MOVB	#1, IOBS+44	
		00000000G	EF	00000000G	PUSHAB	OPEN ERROR	
		00000000G	EF	006F8	CLRL	-(SP)	
		00000000G	EF	8F	PUSHAB	IOBS	
00F4	8F	00	6E	00	2C	006ED	CALLS #3, XPOS\$OPEN
		00000000G	EF	006E6	MOVCS	#0, (SP), #0, #244, IOBS	2081
		00000000G	EF	006F4	MOVL	#50397245, IOBS	
		00000000G	EF	0301003D	MOVW	#526, IOB\$RESULTANT+2	
		00000000G	EF	020E	MOVAB	SIOB\$FILE SPEC, IOBS+4	2084
		00000000G	EF	052C	BISL2	#65538, IOBS+46	
		00000000G	EF	00010002	MOVB	#1, IOBS+44	
		00000000G	EF	00000000G	PUSHAB	OPEN ERROR	
		00000000G	EF	006F8	CLRL	-(SP)	
		00000000G	EF	8F	PUSHAB	IOBS	
00F4	8F	00	6E	00	2C	0072F	CALLS #3, XPOS\$OPEN
		00000000G	EF	00728	MOVCS	#0, (SP), #0, #244, IOBS	2081
		00000000G	EF	00736	MOVL	#50397245, IOBS	
		00000000G	EF	01EC	MOVW	#526, IOB\$RESULTANT+2	
		00000000G	EF	C7 0301003D	MOVAB	SIOB\$FILE SPEC, IOBS+4	2084
		00000000G	EF	020E	BISL2	#65538, IOBS+46	
		00000000G	EF	0534	MOVB	#1, IOBS+44	
		00000000G	EF	00010002	PUSHAB	OPEN ERROR	
		00000000G	EF	00000000G	CLRL	-(SP)	
		00000000G	EF	01EC	PUSHAB	IOBS	
		00000000G	EF	00	2C	00771	CALLS #3, XPOS\$OPEN
		00000000G	EF	0076A	MOVCS	#0, (SP), #0, #244, IOBS	2088

00F4	8F	02E0	C7 0301003D	C7 00778	MOVL #50397245, IOBS			
		02FE	C7 020E	8F DO 0077B	MOVW #526, IOB\$RESULTANT+2			
		02E4	C7 053C	8F BO 00784	MOVAB \$IOB\$FILE SPEC, IOBS+4			
		030E	C7 00010002	C7 00788	BISL2 #65538, IOB\$+46			
		030C	C7 0000000G	8F C8 00792	MOV B #1, IOB\$+44			
				01 90 0079B	PUSHAB OPEN ERROR			
				EF 9F 007A0	CLRL -(SP)			
				7E D4 007A6	PUSHAB IOBS\$			
				C7 9F 007A8	CALLS #3, XPOSOPEN			
		03	0000000G	03 FB 007AC	BBS #2, CMDBLK, 35\$			
				6A 02 E0 007B3	BRW 39\$			
		00		00A2 31 007B7	MOVCS #0, (SP), #0, #244, IOBS\$			
				00 2C 007BA	35\$:			
				6E 007C1	MOVL #50397245, IOBS			
		1E	6E 0301003D	8F DO 007C2	MOVW #526, IOB\$RESULTANT+2			
		04	A5 020E	8F BO 007C9	MOVAB \$IOB\$FILE SPEC, IOBS+4			
		2C	AE 4C	AA 9E 007CF	MOV B #1, IOB\$+44			
				01 90 007D4	PUSHAB OPEN ERROR			
				EF 9F 007D8	CLRL -(SP)			
				7E D4 007DE	PUSHAB IOBS\$			
				AE 9F 007E0	CALLS #3, XPOSOPEN			
		0000000G	EF 0000000G	03 FB 007E3	MOV B #6, IOB\$+44			
		2C	AE	06 90 007EA	5B DD 007EE	PUSHL R11		
				7E D4 007FO	CLRL -(SP)			
				AE 9F 007F2	PUSHAB IOBS\$			
		0000000G	EF 0000000G	03 FB 007F5	CALLS #3, XPOSGET			
		00208001	8F	50 D1 007FC	CMPL R0, #2129921			
				42 12 00803	BNEQ 37\$			
				EF 9F 00805	FUSHAB STR\$FAILURE			
				7E D4 0080B	CLRL -(SP)			
				59 DD 0080D	PUSHL R9			
				AE 9F 0080F	PUSHAB \$STR\$STRING			
				7E D4 00812	CLRL -(SP)			
		0000000G	EF	05 FB 00814	CALLS #5, XST\$COPY			
		0000000G	50	69 3C 0081B	MOVZWL TMPSTR, R0			
		44	EF	50 C0 0081E	ADDL2 R0, CHROUT			
		2C	A8	69 9E 00825	MOVAB \$IOB\$OUTPUT, IOB\$+68			
				07 90 00829	MOV B #7, IOB\$+44			
				5B DD 0082D	PUSHL R11			
				7E D4 0082F	CLRL -(SP)			
				58 DD 00831	PUSHL R8			
		AC	0000000G	EF 0000000G	03 FB 00833	CALLS #3, XPOSPUT		
				6A 01 E1 0083A	BBC #1, CMDBLK, 36\$			
				00 FB 0083E	CALLS #0, SPLIT			
		2C	AE	A5 11 00845	BRB 36\$			
				02 90 00847	37\$: MOV B #2, IOB\$+44			
				5B DD 0084B	PUSHL R11			
				7E D4 0084D	CLRL -(SP)			
				AE 9F 0084F	PUSHAB IOBS\$			
		0000000G	EF	03 FB 00852	CALLS #3, XPOSCLOSE			
		F9	0000000G	0086 31 00859	BRW 41\$			
				01 E0 0085C	38\$: BBS #1, CMDBLK, 38\$			
				EF 9F 00860	PUSHAB STR\$FAILURE			
				7E D4 00866	CLRL -(SP)			
				59 DD 00868	PUSHL R9			
				0544 C7 9F 0086A	PUSHAB \$STR\$STRING			

			7E	D4	0086E		CLRL	- (SP)		
			05	FB	00870		CALLS	#5, XST\$COPY		
			69	3C	00877		MOVZWL	TMPSTR, R0		
			50	C0	0087A		ADDL2	R0, CHROUT		
			69	9E	00881		MOVAB	\$IOBS\$OUTPUT, IOBS\$+68		
			07	90	00885		MOVB	#7, IOBS\$+44		
			5B	DD	00889		PUSHL	R11		
			7E	D4	0088B		CLRL	- (SP)		
			58	DD	0088D		PUSHL	R8		
			03	FB	0088F		CALLS	#3, XPOS\$PUT		
			01	E1	00896		BBC	#1, CMD\$BLK, 40\$		
			00	FB	0089A		CALLS	#0, SPLIT		
07	00000000G	EF	00000000G	EF	9F	008A1	40\$:	PUSHAB	STR\$FAILURE	
			7E	D4	008A7		CLRL	- (SP)		
			59	DD	008A9		PUSHL	R9		
			C7	9F	008AB		PUSHAB	\$STR\$STRING		
			7E	D4	008AF		CLRL	- (SP)		
	00000000G	EF		05	FB	008B1		CALLS	#5, XST\$COPY	
			69	3C	008B8		MOVZWL	TMPSTR, R0		
	00000000G	EF		50	C0	008BB		ADDL2	R0, CHROUT	
			69	9E	008C2		MOVAB	\$IOBS\$OUTPUT, IOBS\$+68		
			07	90	008C6		MOVB	#7, IOBS\$+44		
			5B	DD	008CA		PUSHL	R11		
			7E	D4	008CC		CLRL	- (SP)		
			58	DD	008CE		PUSHL	R8		
	00000000G	EF		03	FB	008D0		CALLS	#3, XPOS\$PUT	
			01	E1	008D7		BBC	#1, CMD\$BLK, 41\$		
	00000000G	EF		00	FB	008DB		CALLS	#0, SPLIT	
			08	B0	008E2	41\$:	MOVW	#8, IOBS\$+52		
			02	90	008E6		MOVB	#2, IOBS\$+54		
			06	90	008EA		MOVB	#6, IOBS\$+44		
			5B	DD	008EE		PUSHL	R11		
			7E	D4	008F0		CLRL	- (SP)		
			57	DD	008F2		PUSHL	R7		
	00000000G	EF		03	FB	008F4		CALLS	#3, XPOS\$GET	
	00208001	8F		50	D1	008FB		CMPL	R0, #2129921	
			0D	13	00902		BEQL	42\$		
			1C	A7	9F	00904		PUSHAB	TOCIOB+28	
			01	DD	00907		PUSHL	#1		
			00000000G	8F	DD	00909		PUSHL	#DSRTOCS_EMPTYIN	
			3B	11	0090F		BRB	47\$		
			50	38	A7	DO	00911	42\$:	TOCIOB+56, R0	
			51		01	DO	00915		MOVL	#1, CORRECT_FORMAT
	FFFFFFFFFF	8F		60	D1	00918		CMPL	(R0), #-1	
			0B	12	0091F		BNEQ	43\$		
	00F4	C7		01	CE	00921		MNEG	#1, FILE TYPE	
		02		A0	D1	00926		CMPL	4(R0), #2	
			0E	11	0092A		BRB	44\$		
			01	60	D1	0092C	43\$:	CMPL	(R0), #1	
			0B	12	0092F		BNEQ	45\$		
	00F4	C7		01	D0	00931		MOVl	#1, FILE TYPE	
		03		A0	D1	00936		CMPL	4(R0), #3	
			02	13	0093A	44\$:	BEQL	46\$		
			51	D4	0093C	45\$:	CLRL	CORRECT_FORMAT		
			14	51	E8	0093E	46\$:	BLBS	CORRECT_FORMAT, 48\$	
			1C	A7	9F	00941		PUSHAB	TOCIOB+28	
			01	DD	00944		PUSHL	#1		

CONTENTS
VO4-0CO

CONTENTS - Table of Contents formatter program G 12
TOCINI - Initialization 16-Sep-1984 00:01:05 VAX-11 Bliss-32 V4.0-742
14-Sep-1984 13:05:48 [RUNOFF.SRC]CONTENTS.BLI;1

Page 64
(3)

00000000G	00	00000000G	8F	DD	00946	47\$:	PUSHL	#DSRTOCS	INVINPUT
			03	FB	0094C		CALLS	#3	LIBSSIGNAL
	50		04	11	00953		BRB	49\$	
			01	D0	00955	48\$:	MOVL	#1,	RO
			04	00958			RET		
			50	D4	00959	49\$:	CLRL	RO	
			04	0095B			RET		

2208
2213
2214

; Routine Size: 2396 bytes, Routine Base: \$CODE\$ + 10FE

```
1584    2215 1 %SBTTL 'PROCESS_PAGE - Process page reference record'  
1585    2216 1 ROUTINE process_page : NOVALUE =  
1586    2217 1 ++  
1587    2218 1  
1588    2219 1 FUNCTIONAL DESCRIPTION:  
1589    2220 1  
1590    2221 1 This routine is called by TOC () to process a page reference record.  
1591    2222 1  
1592    2223 1 FORMAL PARAMETERS:  
1593    2224 1  
1594    2225 1 None  
1595    2226 1  
1596    2227 1 IMPLICIT INPUTS:  
1597    2228 1  
1598    2229 1 None  
1599    2230 1  
1600    2231 1 IMPLICIT OUTPUTS:  
1601    2232 1  
1602    2233 1 pagen - contains the new page reference  
1603    2234 1 The page reference record is written to the four temporary files.  
1604    2235 1  
1605    2236 1 ROUTINE VALUE:  
1606    2237 1 COMPLETION CODES:  
1607    2238 1  
1608    2239 1 None  
1609    2240 1  
1610    2241 1 SIDE EFFECTS:  
1611    2242 1  
1612    2243 1 None  
1613    2244 1 --  
1614    2245 2 BEGIN  
1615    2246 2 errcnt = 0; ! Reset count of CONFUSED messages.  
1616    2247 2  
1617    2248 2  
1618    2249 2 | Read a page number.  
1619    2250 2  
1620    2251 2 $xpo_get (iob = tociob, fullwords = page_sct_size);  
1621    2252 2  
1622    2253 3 BEGIN  
1623    2254 3  
1624    2255 3 BIND  
1625    2256 3 page_ref = .tociob [iob$a_data] : VECTOR [page_sct_size];  
1626    2257 3  
1627    2258 3 MAP  
1628    2259 3 pagen : VECTOR [page_sct_size];  
1629    2260 3  
1630    2261 3 INCR i FROM 1 TO page_sct_size DO  
1631    2262 4 BEGIN  
1632    2263 4  
1633    2264 4 | Copy page number  
1634    2265 4  
1635    2266 4 pagen [.i - 1] = .page_ref [.i - 1];  
1636    2267 3 END;  
1637    2268 3  
1638    2269 2 END;  
1639    2270 2 BEGIN  
1640    2271 3
```

```

: 1641 2272 3
: 1642 2273 3
: 1643 2274 3
: 1644 2275 3
: 1645 2276 3
: 1646 2277 3
: 1647 2278 3
: 1648 2279 3
: 1649 2280 3
: 1650 2281 3
: 1651 2282 3
: 1652 2283 3
: 1653 2284 3
: 1654 2285 3
: 1655 2286 3
: 1656 2287 3
: 1657 2288 2
: 1658 2289 2
: 1659 2290 1

| Write page information to temp files

LOCAL
    record_header : VECTOR [2];

record_header [0] = maj_runoff;
record_header [1] = min_page;
$xo_put (job = stciob, binary_data = (2, record_header));
$xo_put (job = stciob, binary_data = (page_sct_size, pagen));
$xo_put (job = fgiob, binary_data = (2, record_header));
$xo_put (job = fgiob, binary_data = (page_sct_size, pagen));
$xo_put (job = tbiob, binary_data = (2, record_header));
$xo_put (job = tbiob, binary_data = (page_sct_size, pagen));
$xo_put (job = exiob, binary_data = (2, record_header));
$xo_put (job = exiob, binary_data = (page_sct_size, pagen));
END;
END;

```

IOB\$=	TOCJOB
IOB\$=	FGIJOB
IOB\$=	FGIJOB
IOB\$=	TBIJOB
IOB\$=	TBIJOB
IOB\$=	EXIJOB
IOB\$=	EXIJOB

007C 00000 PROCESS_PAGE:						
56	00000000G	EF	9E	00002	.WORD	Save R2,R3,R4,R5,R6
55	00000000G	EF	9E	00009	MOVAB	PAGEN, R6
54	00000000G	EF	9E	00010	MOVAB	IOB\$+68, R5
53	00000000G	EF	9E	00017	MOVAB	XPOSPUT, R4
52	00000000	EF	9E	0001E	MOVAB	XPOSFAILURE, R3
5E		10	C2	00025	SUBL2	IOB\$+52, R2
		03B4	C2	D4 00028	CLRL	#16, SP
02	62		10	B0 0002C	MOVW	ERRCNT
F8	A2		02	90 0002F	MOVB	#16, IOB\$+52
			06	90 00033	MOVB	#2, IOB\$+54
			53	DD 00037	PUSHL	#6, IOB\$+44
			7E	D4 00039	CLRL	R3
			CC	A2 9F 0003B	PUSHAB	-(SP)
	00000000G	EF	03	FB 0003E	CALLS	IOB\$, XPOSGET
		51	01	D0 00045	MOVL	#3, I
		50	04	B241 DE 00048	1\$: MOVAL	at(OI0B+56[I], R0
F1	FC A641	FC	A0	D0 0004D	MOVL	-4(R0), PAGEN-4[I]
		51	04	F3 00053	AOBLEQ	#4, I, 1\$
		08	02	D0 00057	MOVL	#2, RECORD_HEADER
		0C	AE	01 D0 0005B	MOVL	#1, RECORD_HEADER+4
		02	AE	08 B0 0005F	MOVW	#8, \$IOBSOUTPUT
		03	AE	02 90 00062	MOVB	#2, \$IOBSOUTPUT+2
		04	AE	01 90 00066	MOVB	#1, \$IOBSOUTPUT+3
		08	AE	9E 0006A	MOVAB	RECORD_HEADER, \$IOBSOUTPUT+4

E8	A5		6E	9E	0006F	MOVAB	\$10B\$OUTPUT, IOBS+68
			07	90	00072	MOVB	#7, IOBS+44
			53	DD	00076	PUSHL	R3
			7E	D4	00078	CLRL	-(SP)
		BC	A5	9F	0007A	PUSHAB	IOBS
64			03	FB	0007D	CALLS	#3, XPOSPUT
6E			10	B0	00080	MOVW	#16, \$10B\$OUTPUT
02	AE		02	90	00083	MOVB	#2, \$10B\$OUTPUT+2
03	AE		01	90	00087	MOVB	#1, \$10B\$OUTPUT+3
04	AE		66	9E	0008B	MOVAB	PAGEN, \$10B\$OUTPUT+4
E8	A5		6E	9E	0008F	MOVAB	\$10B\$OUTPUT, IOBS+68
			07	90	00092	MOVB	#7, IOBS+44
			53	DD	00096	PUSHL	R3
			7E	D4	00098	CLRL	-(SP)
64		BC	A5	9F	0009A	PUSHAB	IOBS
6E			03	FB	0009D	CALLS	#3, XPOSPUT
02	AE		08	B0	000A0	MOVW	#8, \$10B\$OUTPUT
03	AE		02	90	000A3	MOVB	#2, \$10B\$OUTPUT+2
04	AE		01	90	000A7	MOVB	#1, \$10B\$OUTPUT+3
0108	C2	08	AE	9E	000AB	MOVAB	RECORD HEADER, \$10B\$OUTPUT+4
00F0	C2		6E	9E	000B0	MOVAB	\$10B\$OUTPUT, IOBS+68
			07	90	000B5	MOVB	#7, IOBS+44
			53	DD	000BA	PUSHL	R3
			7E	D4	000BC	CLRL	-(SP)
64		00C4	C2	9F	000BE	PUSHAB	IOBS
6E			03	FB	000C2	CALLS	#3, XPOSPUT
02	AE		10	B0	000C5	MOVW	#16, \$10B\$OUTPUT
03	AE		02	90	000C8	MOVB	#2, \$10B\$OUTPUT+2
04	AE		01	90	000CC	MOVB	#1, \$10B\$OUTPUT+3
0108	C2		66	9E	000D0	MOVAB	PAGEN, \$10B\$OUTPUT+4
00F0	C2		6E	9E	000D4	MOVAB	\$10B\$OUTPUT, IOBS+68
			07	90	000D9	MOVB	#7, IOBS+44
			53	DD	000DE	PUSHL	R3
			7E	D4	000E0	CLRL	-(SP)
64		00C4	C2	9F	000E2	PUSHAB	IOBS
6E			03	FB	000E6	CALLS	#3, XPOSPUT
02	AE		08	B0	000E9	MOVW	#8, \$10B\$OUTPUT
03	AE		02	90	000EC	MOVB	#2, \$10B\$OUTPUT+2
04	AE		01	90	000F0	MOVB	#1, \$10B\$OUTPUT+3
01FC	C2	08	AE	9E	000F4	MOVAB	RECORD HEADER, \$10B\$OUTPUT+4
01E4	C2		6E	9E	000F9	MOVAB	\$10B\$OUTPUT, IOBS+68
			07	90	000FE	MOVB	#7, IOBS+44
			53	DD	00103	PUSHL	R3
			7E	D4	00105	CLRL	-(SP)
64		01B8	C2	9F	00107	PUSHAB	IOBS
6E			03	FB	0010B	CALLS	#3, XPOSPUT
02	AE		10	B0	0010E	MOVW	#16, \$10B\$OUTPUT
03	AE		02	90	00111	MOVB	#2, \$10B\$OUTPUT+2
04	AE		01	90	00115	MOVB	#1, \$10B\$OUTPUT+3
01FC	C2		66	9E	00119	MOVAB	PAGEN, \$10B\$OUTPUT+4
01E4	C2		6E	9E	0011D	MOVAB	\$10B\$OUTPUT, IOBS+68
			07	90	00122	MOVB	#7, IOBS+44
			53	DD	00127	PUSHL	R3
			7E	D4	00129	CLRL	-(SP)
64		01B8	C2	9F	0012B	PUSHAB	IOBS
6E			03	FB	0012F	CALLS	#3, XPOSPUT
			08	B0	00132	MOVW	#8, \$10B\$OUTPUT

CONTENTS
V04-000

CONTENTS - Table of Contents formatter program K 12
PROCESS_PAGE - Process page reference record 16-Sep-1984 00:01:05 VAX-11 Bliss-32 V4.0-742
14-Sep-1984 13:05:48 [RUNOFF.SRC]CONTENTS.BLI;1

Page 68
(4)

02 AE	02 90 00135	MOV B #2, \$IOB\$OUTPUT+2
03 AE	01 90 00139	MOV B #1, \$IOB\$OUTPUT+3
04 AE	AE 9E 0013D	MOVAB RECORD HEADER, \$IOB\$OUTPUT+4
02F0 C2	6E 9E 00142	MOVAB \$IOB\$OUTPUT, IOB\$+68
02D8 C2	07 90 00147	MOV B #7, IOB\$+44
	53 DD 0014C	PUSHL R3
	7E D4 0014E	CLRL -(SP)
	C2 9F 00150	PUSHAB IOB\$
64	03 FB 00154	CALLS #3, XPOSPUT
6E	10 B0 00157	MOVW #16, \$IOB\$OUTPUT
02 AE	02 90 0015A	MOV B #2, \$IOB\$OUTPUT+2
03 AE	01 90 0015E	MOV B #1, \$IOB\$OUTPUT+3
04 AE	66 9E 00162	MOVAB PAGEN, \$IOB\$OUTPUT+4
02F0 C2	6E 9E 00166	MOVAB \$IOB\$OUTPUT, IOB\$+68
02D8 C2	07 90 0016B	MOV B #7, IOB\$+44
	53 DD 00170	PUSHL R3
	7E D4 00172	CLRL -(SP)
	C2 9F 00174	PUSHAB IOB\$
64	03 FB 00178	CALLS #3, XPOSPUT
	04 0017B	RET

; Routine Size: 380 bytes, Routine Base: \$CODE\$ + 1A5A

2287

; 2290

```
: 1661 2291 1 %SBTTL 'PROCESS_ENTITY_INFO - Process figure/table/example info record'  
: 1662 2292 1 ROUTINE process_entity_info : NOVALUE =  
: 1663 2293 1 ++  
: 1664 2294 1  
: 1665 2295 1 FUNCTIONAL DESCRIPTION:  
: 1666 2296 1  
: 1667 2297 1 This routine is called by TOC () to process a figure/table/example  
: 1668 2298 1 information record.  
: 1669 2299 1  
: 1670 2300 1 FORMAL PARAMETERS:  
: 1671 2301 1 None  
: 1672 2302 1  
: 1673 2303 1 IMPLICIT INPUTS:  
: 1674 2304 1  
: 1675 2305 1  
: 1676 2306 1 major - major record code  
: 1677 2307 1 minor - minor record code  
: 1678 2308 1 exiob - IOB for examples temp file  
: 1679 2309 1 fgiob - IOB for figures temp file  
: 1680 2310 1 tbiob - IOB for tables temp file  
: 1681 2311 1  
: 1682 2312 1 IMPLICIT OUTPUTS:  
: 1683 2313 1 The record is written to the appropriate temp file if new captions  
: 1684 2314 1 are being processed.  
: 1685 2315 1  
: 1686 2316 1 ROUTINE VALUE:  
: 1687 2317 1 COMPLETION CODES:  
: 1688 2318 1 None  
: 1689 2319 1  
: 1690 2320 1  
: 1691 2321 1 SIDE EFFECTS:  
: 1692 2322 1 None  
: 1693 2323 1  
: 1694 2324 1  
: 1695 2325 1 --  
: 1696 2326 2 BEGIN  
: 1697 2327 2  
: 1698 2328 2 LOCAL  
: 1699 2329 2 prefix_addr,  
: 1700 2330 2 prefix_fullwords,  
: 1701 2331 2 suffix_addr,  
: 1702 2332 2 suffix_fullwords,  
: 1703 2333 2 iob_ptr : REF $xpo_iob (),  
: 1704 2334 2 info_list : VECTOR[6];  
: 1705 2335 2  
: 1706 2336 2 errcnt = 0; ! Reset count of CONFUSED messages.  
: 1707 2337 3 iob_ptr = (SELECTONE .minor OF  
: 1708 2338 3 SET  
: 1709 2339 3 [min_exampl_inf] : exiob;  
: 1710 2340 3 [min_table_inf] : tbiob;  
: 1711 2341 3 [min_figur_inf] : fgiob;  
: 1712 2342 2 TES);  
: 1713 2343 2  
: 1714 2344 2 info_list [0] = .major;  
: 1715 2345 2 info_list [1] = .minor;  
: 1716 2346 2  
: 1717 2347 2 ! Get counter, display descriptor and prefix text length
```

```
: 1718 2348 2      !
: 1719 2349 2      $xpo_get (iob = tociob, fullwords = 3);
: 1720 2350 2      BEGIN
: 1721 2351 3      BIND
: 1722 2352 3      counter_info = .tociob [iob$A_data] : VECTOR;
: 1723 2353 3
: 1724 2354 3      info_list [2] = .counter_info [0]; ! Counter value
: 1725 2355 3      info_list [3] = .counter_info [1]; ! Counter display code
: 1726 2356 3      info_list [4] = .counter_info [2]; ! Prefix text length
: 1727 2357 3      END;
: 1728 2358 2
: 1729 2359 2
: 1730 2360 2      IF .info_list [4] NEQ 0
: 1731 2361 2      THEN
: 1732 2362 3      BEGIN
: 1733 2363 3      | There is prefix text
: 1734 2364 3
: 1735 2365 3
: 1736 2366 3      prefix_fullwords = CH$ALLOCATION (.info_list [4]);
: 1737 2367 3      $xpo_get (iob = tociob, fullwords = .prefix_fullwords);
: 1738 2368 3
: 1739 2369 3      | Get some memory to store the prefix string in.
: 1740 2370 3
: 1741 2371 3      $xpo_get_mem (fullwords = .prefix_fullwords, result = prefix_addr);
: 1742 2372 3
: 1743 2373 3      | Copy the prefix string to the memory just allocated
: 1744 2374 3
: 1745 2375 4      BEGIN
: 1746 2376 4      BIND
: 1747 2377 4      input_vector = .tociob [iob$A_data] : VECTOR,
: 1748 2378 4      prefix_vector = .prefix_addr : VECTOR;
: 1749 2379 4
: 1750 2380 4      INCR i FROM 0 TO .prefix_fullwords - 1 DO
: 1751 2381 4      | prefix_vector [.i] = .input_vector [.i];
: 1752 2382 4
: 1753 2383 3
: 1754 2384 2
: 1755 2385 2
: 1756 2386 2
: 1757 2387 2      | Get length of suffix text
: 1758 2388 2
: 1759 2389 2      $xpo_get (iob = tociob, fullwords = 1);
: 1760 2390 2      info_list [5] = .(tociob [iob$A_data]);
: 1761 2391 2
: 1762 2392 2      IF .info_list [5] NEQ 0
: 1763 2393 2      THEN
: 1764 2394 3      BEGIN
: 1765 2395 3      | There is suffix text
: 1766 2396 3
: 1767 2397 3
: 1768 2398 3      suffix_fullwords = CH$ALLOCATION (.info_list [5]);
: 1769 2399 3      $xpo_get (iob = tociob, fullwords = .suffix_fullwords);
: 1770 2400 3
: 1771 2401 3
: 1772 2402 3
: 1773 2403 3
: 1774 2404 3      | Get some memory to store the suffix string in.
:                  $xpo_get_mem (fullwords = .suffix_fullwords, result = suffix_addr);
```

```
1775      2405 3      | Copy the suffix string to the dynamic memory area
1776      2406 3
1777      2407 4      BEGIN
1778      2408 4      BIND
1779      2409 4      input_vector = .tociob [iob$a_data] : VECTOR,
1780      2410 4      suffix_vector = .suffix_addr : VECTOR;
1781      2411 4
1782      2412 4      INCR i FROM 0 TO .suffix_fullwords - 1 DO
1783      2413 4      suffix_vector [.i] = .input_vector [.i];
1784      2414 4
1785      2415 3      END;
1786      2416 2      END;
1787      2417 2
1788      2418 2      IF vfycap (new_captions)
1789      2419 2      THEN
1790      2420 3      BEGIN
1791      2421 3      Write major-code, minor-code, counter,
1792      2422 3      display-descriptor and prefix text length.
1793      2423 3
1794      2424 3
1795      2425 3      $xpo_put (iob = .iob_ptr, binary_data = (5, info_list));
1796      2426 3
1797      2427 3      Write prefix text if any
1798      2428 3
1799      2429 3
1800      2430 3
P 1801      2431 3      IF .info_list [4] NEQ 0
1802      2432 3      THEN
1803      2433 3      $xpo_put (iob = .iob_ptr,
1804      2434 3      binary_data = (.prefix_fullwords, .prefix_addr));
1805      2435 3
1806      2436 3
1807      2437 3      Write suffix text length
1808      2438 3      $xpo_put (iob = .iob_ptr, binary_data = (1, info_list [5]));
1809      2439 3      Write suffix text if any
1810      2440 3
1811      2441 3      IF .info_list [5] NEQ 0
1812      2442 3      THEN
P 1813      2443 3      $xpo_put (iob = .iob_ptr,
1814      2444 3      binary_data = (.suffix_fullwords, .suffix_addr));
1815      2445 3
1816      2446 2      END;
1817      2447 2
1818      2448 2      IF .info_list [4] NEQ 0
1819      2449 2      THEN
1820      2450 3      BEGIN
1821      2451 3      Deallocate the memory we allocated for the prefix string
1822      2452 3      $xpo_free_mem (binary_data = (.prefix_fullwords, .prefix_addr));
1823      2453 3
1824      2454 3
1825      2455 2      END;
1826      2456 2
1827      2457 2      IF .info_list [5] NEQ 0
1828      2458 2      THEN
1829      2459 3      BEGIN
1830      2460 3      Deallocate the memory allocated for the suffix string
1831      2461 3
```

```
: 1832      2462 3      !
: 1833      2463 3      $xpo_free_mem (binary_data = (.suffix_fullwords, .suffix_addr));
: 1834      2464 2      END;
: 1835      2465 2      END;
: 1836      2466 1      END;
```

```
IOB$=          TOCIOB
IOB$=          TOCIOB
IOB$=          TOCIOB
IOB$=          TOCIOB
.EXTRN XPOS$ALLOC_MEM, XPOS$FREE_MEM
```

OFFC 00000 PROCESS_ENTITY_INFO:

5B	00000000G	EF	9E	00002	.WORD	Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11						2292
5A	00000000G	EF	9E	00009	MOVAB	XPO\$PUT, R11						
59	00000000G	EF	9E	00010	MOVAB	XPO\$GET, R10						
58	00000000	EF	9E	00017	MOVAB	XPO\$FAILURE, R9						
5E		20	C2	0001E	SUBL2	I0BS+52, R8						
	03B4	C8	D4	00021	CLRL	#32, SP						2336
50	00000000G	EF	D0	00025	MOVL	ERRCNT						2337
07		50	D1	0002C	CMPL	MINOR, R0						2339
		07	12	0002F	BNEQ	R0, #?						
54	02AC	C8	9E	00031	1\$: MOVAB	1\$						
		1B	11	00036	BRB	EXIOB, IOB_PTR						
0D		50	D1	00038	1\$: CMPL	4\$						2340
		07	12	0003B	BNEQ	R0, #13						
54	01B8	C8	9E	0003D	2\$: MOVAB	2\$						
		0F	11	00042	BRB	TBIOB, IOB_PTR						
0A		50	D1	00044	2\$: CMPL	4\$						2341
		05	13	00047	BEQL	R0, #10						
54		01	CE	00049	MNEGL	3\$						
		05	11	0004C	BRB	#1, IOB_PTR						
54	00C4	C8	9E	0004E	3\$: MOVAB	4\$						
08	AE 00000000G	EF	D0	00053	4\$: MOVL	FGIOB, IOB_PTR						2344
0C	AE	50	D0	0005B	MOVW	MAJOR, INFO_LIST						2345
68		0C	B0	0005F	MOVW	R0, INFO_LIST+4						2349
02	A8	02	90	00062	MOVW	#12, IOB\$+52						
F8	A8	06	90	00066	MOVW	#2, IOB\$+54						
		59	DD	0006A	MOVW	#6, IOB\$+44						
		7E	D4	0006C	PUSHL	PUSHAB	R9					
	CC	A8	9F	0006E	CLRL	- (SP)						
6A		03	FB	00071	PUSHAB	IOBS						
50	04	A8	D0	00074	CALLS	#3, XPO\$GET						
10	AE	60	7D	00078	MOVL	TO CIOB+56, R0						2353
18	AE	08	A0	0007C	MOVQ	(R0), INFO_LIST+8						2355
		55	13	00081	MOVL	8(R0), INFO_LIST+16						2357
					BEQL	8\$						2360
18	AE	03	C1	00083	ADDL3	#3, INFO_LIST+16, R0						2366
50	04	C7	00088	DIVL3	#4, R0, PREFIX FULLWORDS							
53	02	78	0008C	ASHL	#2, PREFIX FULLWORDS, R2							2367
68	52	B0	00090	MOVW	R2, IOB\$+52							
02	A8	02	90	00093	MOVW	#2, IOB\$+54						
F8	A8	06	90	00097	MOVW	#6, IOB\$+44						
		59	DD	0009B	PUSHL	PUSHAB	R9					
		7E	D4	0009D	CLRL	- (SP)						

		CC	A8	9F	0009F	PUSHAB	I0B\$	
	6A	02020000	03	FB	000A2	CALLS	#3, XPOS\$GET	
	6E		8F	DO	000A5	MOVL	#33685504, \$XPOSDESC	
		04	AE	D4	000AC	CLRL	\$XPOSDESC+4	
			59	DD	000AF	PUSHL	R9	
	7E		7E	7C	000B1	CLRQ	-(SP)	
			01	CE	000B3	MNEGL	#1, -(SP)	
			10	AE	9F	PUSHAB	\$XPOSDESC	
			52	DD	000B9	PUSHL	R2	
00000000G	EF		06	FB	000BB	CALLS	#6, XPOS\$ALLOC MEM	
	04		50	E9	000C2	BLBC	\$XPOSSTATUS, 5\$	
	56		04	AE	DO	000C5	MOVL	\$XPOSDESC+4, PREFIX_ADDR
	50		01	CE	000C9	5\$: MNEGL	#1, I	
	6640		06	11	000CC	BRB	7\$	
F6	50		04	B840	DO	000CE	6\$: MOVL	@TOC10B+56[1], (PREFIX_ADDR)[1]
	68		53	F2	000D4	7\$: AOBLS	PREFIX_FULLWORDS, I, 6\$	
02	A8		04	BO	000D8	MOVW	#4, IOB\$+52	
F8	A8		02	90	000DB	MOVB	#2, IOB\$+54	
			06	90	000DF	MOVB	#6, IOB\$+44	
			59	DD	000E3	PUSHL	R9	
			7E	D4	000E5	CLRL	-(SP)	
		CC	A8	9F	000E7	PUSHAB	I0B\$	
1C	6A		03	FB	000EA	CALLS	#3, XPOS\$GET	
	AE		04	B8	DO	000ED	MOVL	@TOC10B+56, INFO_LIST+20
			57	D4	000F2	CLRL	R7	
		1C	AE	D5	000F4	TSTL	INFO_LIST+20	
			57	13	000F7	BEQL	12\$	
			57	D6	000F9	INCL	R7	
50	1C	AE	03	C1	000FB	ADDL3	#3, INFO_LIST+20, R0	
52	50		04	C7	00100	DIVL3	#4, R0, SUFFIX_FULLWORDS	
55	52		02	78	00104	ASHL	#2, SUFFIX_FUL[WORDS], R5	
	68		55	BO	00108	MOVW	R5, IOB\$+52	
02	A8		02	90	0010B	MOVB	#2, IOB\$+54	
F8	A8		06	90	0010F	MOVB	#6, IOB\$+44	
			59	DD	00113	PUSHL	R9	
			7E	D4	00115	CLRL	-(SP)	
	6A	CC	A8	9F	00117	PUSHAB	I0B\$	
6E	02020000		03	FB	0011A	CALLS	#3, XPOS\$GET	
			8F	DO	0011D	MOVL	#33685504, \$XPOSDESC	
			04	AE	D4	CLRL	\$XPOSDESC+4	
			59	DD	00124	PUSHL	R9	
	7E		7E	7C	00129	CLRQ	-(SP)	
			01	CE	0012B	MNEGL	#1, -(SP)	
			10	AE	9F	PUSHAB	\$XPOSDESC	
			55	DD	00131	PUSHL	R5	
00000000G	EF		06	FB	00133	CALLS	#6, XPOS\$ALLOC MEM	
	04		50	E9	0013A	BLBC	\$XPOSSTATUS, 9\$	
	55		04	AE	DO	0013D	MOVL	\$XPOSDESC+4, SUFFIX_ADDR
	50		01	CE	00141	9\$: MNEGL	#1, I	
	6540		06	11	00144	BRB	11\$	
F6	50		04	B840	DO	00146	10\$: MOVL	@TOC10B+56[1], (SUFFIX_ADDR)[1]
			52	F2	0014C	11\$: AOBLS	SUFFIX_FULLWORDS, I, 10\$	
			01	DD	00150	12\$: PUSHL	#1	
00000000V	EF		01	FB	00152	CALLS	#1, VFYCAP	
	03		50	E8	00159	BLBS	R0, 13\$	
			008C	31	0015C	BRW	15\$	
	6E		14	BO	0015F	13\$: MOVW	#20, \$IOB\$OUTPUT	

CONTENTS
V04-000

CONTENTS - Table of Contents formatter program E 13
PROCESS_ENTITY_INFO - Process figure/table/exam 16-Sep-1984 00:01:05
14-Sep-1984 13:05:48 VAX-11 Bliss-32 V4.0-742
[RUNOFF.SRC]CONTENTS.BLI;1

Page 75
(5)

04 AE	55 D0 00220	MOVL SUFFIX_ADDR, \$XPOSDESC+4
	59 DD 00224	PUSHL R9
7E	7E 7C 00226	CLRQ -(SP)
	02 CE 00228	MNEG L #2, -(SP)
0000000G EF	10 AE 9F 0022B	PUSHAB \$XPOSDESC
	05 FB 0022E	CALLS #5, XPOSFREE_MEM
	04 00235 17\$:	RET

; 2466

; Routine Size: 566 bytes, Routine Base: \$CODE\$ + 1BD6

```
1838 2467 1 %SBTTL 'PROCESS_ENTITY_TXT - Process figure/table/example text record'
1839 2468 1 ROUTINE process_entity_txt : NOVALUE =
1840 2469 1 ++
1841 2470 1
1842 2471 1 FUNCTIONAL DESCRIPTION:
1843 2472 1
1844 2473 1 This routine is called by TOC () to process a figure/table/example
1845 2474 1 text record.
1846 2475 1
1847 2476 1 FORMAL PARAMETERS:
1848 2477 1
1849 2478 1 None
1850 2479 1
1851 2480 1 IMPLICIT INPUTS:
1852 2481 1
1853 2482 1 major - major record code
1854 2483 1 minor - minor record code
1855 2484 1 exiob - IOB for examples temp file
1856 2485 1 fgiob - IOB for figures temp file
1857 2486 1 tbiob - IOB for tables temp file
1858 2487 1
1859 2488 1 IMPLICIT OUTPUTS:
1860 2489 1
1861 2490 1 txtlen - length of caption text
1862 2491 1
1863 2492 1 The record is written to the appropriate temp file if new captions
1864 2493 1 are being processed.
1865 2494 1
1866 2495 1 ROUTINE VALUE:
1867 2496 1 COMPLETION CODES:
1868 2497 1
1869 2498 1 None
1870 2499 1
1871 2500 1 SIDE EFFECTS:
1872 2501 1
1873 2502 1 None
1874 2503 1 --
1875 2504 2 BEGIN
1876 2505 2
1877 2506 2 LOCAL
1878 2507 2     iob_ptr : REF $xpo_iob ();
1879 2508 2     record_header : VECTOR [3];
1880 2509 2
1881 2510 2     errcnt = 0;           ! Reset count of CONFUSED messages.
1882 2511 2
1883 2512 3     iob_ptr = (SELECTONE .minor OF
1884 2513 3         SET
1885 2514 3         [min_figur_txt] : fgiob;
1886 2515 3         [min_figur_txt_u] : fgiob;
1887 2516 3         [min_table_txt] : tbiob;
1888 2517 3         [min_table_txt_u] : tbiob;
1889 2518 3         [min_exampl_txt] : exiob;
1890 2519 3         [min_exampl_txt_u] : exiob;
1891 2520 2         TES);
1892 2521 2
1893 2522 2
1894 2523 2     ! Get length of text
```

```

1895    2524 2      !
1896    2525 2      $xpo_get (iob = .tociob, fullwords = 1);
1897    2526 2      txtlen = .(tociob [iob$@_data]);
1898    2527 2
1899    2528 2      record_header [0] = .major;
1900    2529 2      record_header [1] = .minor;
1901    2530 2      record_header [2] = .txtlen;
1902    2531 2
1903    2532 2      | Get the text, if there is any
1904    2533 2
1905    2534 2      IF .txtlen NEQ 0
1906    2535 2      THEN
1907    2536 2          $xpo_get (iob = .tociob, fullwords = CH$ALLOCATION (.txtlen));
1908    2537 2
1909    2538 2      IF vfycap (new_captions)
1910    2539 2      THEN
1911    2540 3          BEGIN
1912    2541 3
1913    2542 3          | Write record to temp file
1914    2543 3
1915    2544 3          $xpo_put (iob = .iob_ptr, binary_data = (3, record_header));
1916    2545 3          $xpo_put (iob = .iob_ptr,
1917    2546 3              binary_data = (CR$ALLOCATION (.txtlen), .tociob [iob$@_data]));
1918    2547 3
1919    2548 2
1920    2549 2
1921    2550 1      END;
END;

```

IOB\$= TOCIOB
IOB\$= TOCIOB

01FC 00000 PROCESS_ENTITY_TXT:

58 00000000G	EF 9E 00002	.WORD	Save R2,R3,R4,R5,R6,R7,R8	2468
57 00000000G	EF 9E 00009	MOVAB	XPO\$PUT, R8	
56 00000000G	EF 9E 00010	MOVAB	XPO\$GET, R7	
55 00000000G	EF 9E 00017	MOVAB	MINOR, R6	
54 00000000G	EF 9E 0001E	MOVAB	TXTLEN, R5	
53 00000000'	EF 9E 00025	MOVAB	XPO\$FAILURE, R4	
5E	14 C2 0002C	MOVAB	IOB\$+52, R3	
	03B4 C3 D4 0002F	SUBL2	#20, SP	
50	66 D0 00033	CLRL	ERRCNT	2510
0B	50 D1 00036	MOVL	MINOR, R0	2512
	05 13 00039	CMPL	R0, #11	2514
0C	50 D1 0003B	BEQL	1S	
	07 12 0003E	CMPL	R0, #12	2515
52	00C4 C3 9E 00040	BNEQ	2S	
	1\$: 25 11 00045	MOVAB	FGIOB, IOB_PTR	
0E	50 D1 00047	BRB	6S	2516
	2\$: 05 13 0004A	CMPL	R0, #14	
0F	50 D1 0004C	BEQL	3S	2517
	07 12 0004F	CMPL	R0, #15	
52	01B8 C3 9E 00051	BNEQ	4S	
	3\$: 14 11 00056	MOVAB	TBIOB, IOB_PTR	
		BRB	6S	

CONTENTS		H 13 CONTENTS - Table of Contents formatter program 16-Sep-1984 00:01:05 PROCESS_ENTITY_TXT - Process figure/table/examp 14-Sep-1984 13:05:48						VAX-11 Bliss-32 V4.0-742 [RUNOFF.SRC]CONTENTS.BLI;1	Page 78 (6)		
CONTENTS	V04-000	08	50	D1	00058	4\$:	CMPL	R0, #8	2518		
		09	0A	13	0005B		BEQL	5\$	2519		
		52	50	D1	0005D		CMPL	R0, #9			
			05	13	00060		BEQL	5\$			
		52	01	CE	00062		MNEGL	#1, IOB_PTR			
			05	11	00065		BRB	6\$			
		52	C3	9E	00067	5\$:	MOVAB	EXIOB, IOB_PTR			
			04	B0	0006C	6\$:	MOVW	#4, IOB\$+52	2525		
		02	A3	02	90	0006F	MOVB	#2, IOB\$+54			
		F8	A3	06	90	00073	MOVB	#6, IOB\$+44			
				54	DD	00077	PUSHL	R4			
				7E	D4	00079	CLRL	-(SP)			
			CC	A3	9F	0007B	PUSHAB	IOB\$			
		67	03	FB	0007E		CALLS	#3, XPOSGET			
		65	04	B3	DO	00081	MOVL	@TOCIOB+56, TXTLEN	2526		
		08	AE	00000000G	EF	DO	00085	MOVL	MAJOR, RECORD_HEADER	2528	
		0C	AE		66	DO	0008D	MOVL	MINOR, RECORD_HEADER+4	2529	
		50			65	DO	00091	MOVL	TXTLEN, R0	2530	
		10	AE		50	DO	00094	MOVL	R0, RECORD_HEADER+8		
				1C	13	00098	BEQL	7\$	2534		
		50			03	C0	0009A	ADDL2	#3, R0	2536	
		50			04	C6	0009D	DIVL2	#4, R0		
		63	02	A3	04	A5	000A0	MULW3	#4, R0, IOB\$+52		
			F8	A3	02	90	000A4	MOVB	#2, IOB\$+54		
				06	90	000A8	MOVB	#6, IOB\$+44			
				54	DD	000AC	PUSHL	R4			
				7E	D4	000AE	CLRL	-(SP)			
			CC	A3	9F	000B0	PUSHAB	IOB\$			
		67	03	FB	000B3		CALLS	#3, XPOSGET			
				01	DD	000B6	PUSHL	#1	2538		
		00000000V	EF		01	FB	000B8	CALLS	#1, VFYCAP		
			4A		50	E9	000BF	BLBC	R0, 8\$		
			6E		0C	B0	000C2	MOVW	#12, \$IOB\$OUTPUT	2544	
		02	AE		02	90	000C5	MOVB	#2, \$IOB\$OUTPUT+2		
		03	AE		01	90	000C9	MOVB	#1, \$IOB\$OUTPUT+3		
		04	AE		08	AE	000CD	MOVAB	RECORD_HEADER, \$IOB\$OUTPUT+4		
		44	A2		6E	9E	000D2	MOVAB	\$IOB\$OUTPUT, 68(IOB_PTR)		
		2C	A2		07	90	000D6	MOVB	#7, 44(IOB_PTR)		
				54	DD	000DA	PUSHL	R4			
				7E	D4	000DC	CLRL	-(SP)			
				52	DD	000DE	PUSHL	IOB_PTR			
		50	68		03	FB	000EO	CALLS	#3, XPO\$PUT		
			65		03	C1	000E3	ADDL3	#3, TXILEN, R0	2546	
		6E	50		04	C6	000E7	DIVL2	#4, R0		
			02	AE	04	A5	000EA	MULW3	#4, R0, \$IOB\$OUTPUT		
			03	AE	02	90	000EE	MOVB	#2, \$IOB\$OUTPUT+2		
			04	AE	01	90	000F2	MOVB	#1, \$IOB\$OUTPUT+3		
			44	A2	04	A3	DO	000F6	MOVL	TOCIOB+56, \$IOB\$OUTPUT+4	
			2C	A2	6E	9E	000FB	MOVAB	\$IOB\$OUTPUT, 68(IOB_PTR)		
				07	90	000FF	MOVB	#7, 44(IOB_PTR)			
				54	DD	00103	PUSHL	R4			
				7E	D4	00105	CLRL	-(SP)			
				52	DD	00107	PUSHL	IOB_PTR			
		68			03	FB	00109	CALLS	#3, XPO\$PUT		
					04	0010C	8\$:	RET	2550		

; Routine Size: 269 bytes, Routine Base: \$CODE\$ + 1E0C

CONTENTS
V04-000

CONTENTS - Table of Contents formatter program 16-Sep-1984 00:01:05
PROCESS_ENTITY_TXT - Process figure/table/examp 14-Sep-1984 13:05:48
I 13
[RUNOFF.SRC]CONTENTS.BLI;1

Page 79
(6)

```
1923      2551 1 %SBTTL 'TOCFIN - Finish table of contents processing'  
1924      2552 1  
1925      2553 1 GLOBAL ROUTINE tocfin =  
1926      2554 1 ++  
1927      2555 1  
1928      2556 1  
1929      2557 1  
1930      2558 1  
1931      2559 1  
1932      2560 1  
1933      2561 1  
1934      2562 1  
1935      2563 1  
1936      2564 1  
1937      2565 1  
1938      2566 1  
1939      2567 1  
1940      2568 1  
1941      2569 1  
1942      2570 1  
1943      2571 1  
1944      2572 1  
1945      2573 1  
1946      2574 1  
1947      2575 1  
1948      2576 1  
1949      2577 1  
1950      2578 1  
1951      2579 1  
1952      2580 1  
1953      2581 1  
1954      2582 1  
1955      2583 1  
1956      2584 2  
1957      2585 2  
1958      2586 2  
1959      2587 2  
1960      2588 2  
1961      2589 2 $xpo_close (iob = exiob, options = remember);  
1962      2590 2 $xpo_close (iob = fgiob, options = remember);  
1963      2591 2 $xpo_close (iob = tbiob, options = remember);  
1964      2592 2 $xpo_close (iob = stciob, options = remember);  
1965      2593 2  
1966      2594 2  
1967      2595 2 IF .captyp EQL new_captions  
1968      2596 2 THEN  
1969      2597 3 BEGIN  
1970      2598 3  
1971      2599 3  
1972      2600 3  
1973      2601 3  
1974      2602 3 IF .cmdblk [contents$v_tms11]  
1975      2603 4 THEN  
1976      2604 4 BEGIN  
1977      2605 4  
1978      2606 4  
1979      2607 4  
          1 For TMS...  
          1  
          1 hdr_x (exiob, tms_ex, min_examp_inf, min_examp_txt, min_examp_txt_u);
```

```
1980 2608 4      hdr_x (fgiob, tms_fg, min_figur_inf, min_figur_txt, min_figur_txt_u);  
1981 2609 4      hdr_x (tbiob, tms_tb, min_table_inf, min_table_txt, min_table_txt_u);  
1982 2610 4      END  
1983 2611 3      ELSE  
1984 2612 4      BEGIN  
1985 2613 4      | For RUNOFF...  
1986 2614 4      |  
1987 2615 4      |  
1988 2616 4      |  
1989 2617 4      |  
1990 2618 4      |  
1991 2619 3      |  
1992 2620 3      |  
1993 2621 2      |  
1994 2622 2      |  
1995 2623 2      |  
1996 2624 2      | Process the .SEND TOC temp file  
1997 2625 2      |  
1998 2626 2      stchdr ();  
1999 2627 2      |  
2000 2628 2      | Delete temp files  
2001 2629 2      |  
2002 2630 2      $xpo_delete (iob = exiob);  
2003 2631 2      $xpo_delete (iob = fgiob);  
2004 2632 2      $xpo_delete (iob = tbiob);  
2005 2633 2      $xpo_delete (iob = stciob);  
2006 2634 2      |  
2007 2635 2      | Write appropriate epilogue  
2008 2636 2      |  
2009 2637 2      IF .cmdbl [contents$v_tms1i]  
2010 2638 2      THEN  
2011 2639 3      | put ('*cfini*')  
2012 2640 2      ELSE  
2013 2641 2      | put ('.RESTORE');  
2014 2642 2      |  
2015 2643 2      |  
2016 2644 2      | Finally, close the output file and call it quits  
2017 2645 2      |  
2018 2646 2      IF .cmdbl [contents$v_log]  
2019 2647 2      THEN  
2020 L 2648 2      %IF %BLISS (BLISS32)  
2021 2649 2      %THEN  
2022 2650 2      | SIGNAL (contents$_created, 1, tocoob [iob$t_resultant]);  
2023 U 2651 2      %ELSE  
2024 U 2652 2      | $xpo_put_msg (severity = success,  
2025 U 2653 2      |   string = $str_concat ('"', tocoob [iob$t_resultant], '" created'));  
2026 2654 2      %FI  
2027 2655 2      |  
2028 2656 2      IF .cmdbl [contents$v_output]  
2029 2657 2      THEN  
2030 2658 3      | $xpo_close (iob = tocoob)  
2031 2659 2      ELSE  
2032 2660 3      | BEGIN  
2033 2661 3      |  
2034 2662 3      | User specified /NOOUTPUT  
2035 2663 3      | Close and delete the output file.  
2036 2664 3      |
```

CONTENTS
V04-000

CONTENTS - Table of Contents formatter program L 13
TOCFIN - Finish table of contents processing 16-Sep-1984 00:01:05
14-Sep-1984 13:05:48 VAX-11 Bliss-32 v4.0-742
[RUNOFF.SRC]CONTENTS.BLI;1

Page 82
(7)

: 2037 2665 3 \$xpo_close (iob = tocoob, options = remember);
: 2038 2666 3 \$xpo_delete (iob = tocoob);
: 2039 2667 2 END;
: 2040 2668 2
: 2041 2669 2 RETURN true;
: 2042 2670 1 END;

.PSECT \$PLIT\$,NOWRT,NOEXE,2
45 52 69 6E 69 66 63 2A 00280 P.ACX: .ASCII *cfini*\br/>45 52 4F 54 53 45 52 2E 00287 PACY: .ASCII \.RESTORE\

.PSECT \$OWNS\$,NOEXE,2
0007 00554 \$STR\$STRING:
01 0E 00556 .WORD 7
00000000' 00558 .BYTE 14, 1
0008 0055C \$STR\$STRING:
01 0E 0055E .WORD 8
00000000' 00560 .BYTE 14, 1
.ADDRESS P.ACX
.ADDRESS PACY

IOB\$= EXIJOB
IOB\$= FGJOB
IOB\$= TBIJOB
IOB\$= EXIJOB
IOB\$= FGJOB
IOB\$= TBIJOB

.EXTRN XPOSDELETE

.PSECT \$CODE\$,NOWRT,2

OFFC 00000

5B 00000000G EF 9E 00002
5A 00000000G EF 9E 00009
59 00000000G EF 9E 00010
58 00000000G EF 9E 00017
57 00000000G EF 9E 0001E
56 00000000G EF 9E 00025
55 00000000G EF 9E 0002C
54 00000000G EF 9E 00033
53 00000000G EF 9E 0003A
52 00000000' EF 9E 00041
2E A2 10 88 00048
2C A2 02 90 0004C
53 DD 00050
7E D4 00052
52 DD 00054
65 03 FB 00056
FE46 C2 10 88 00059
FE44 C2 02 90 0005E
53 DD 00063
7E D4 00065

.ENTRY TOCFIN, Save R2,R3,R4,R5,R6,R7,R8,R9,R10,- R11 2552
MOVAB STR\$FAILURE, R11
MOVAB \$STR\$TARGET, R10
MOVAB XPOSDELETE, R9
MOVAB HDR X, R8
MOVAB CMDBLK, R7
MOVAB IOB\$+44, R6
MOVAB XPOSCLOSE, R5
MOVAB IOB\$+44, R4
MOVAB XPOSFAILURE, R3
MOVAB IOB\$, R2
BISB2 #16, IOB\$+46
MOVAB #2, IOB\$+44
PUSHL R3
CLRL -(SP)
PUSHL R2
CALLS #3, XPOSCLOSE
BISB2 #16, IOB\$+46
MOVAB #2, IOB\$+44
PUSHL R3
CLRL -(SP) 2588
2589

		FE18	C2	9F 00067	PUSHAB	I0BS	
FF3A	65		03	FB 0006B	CALLS	#3, XPOS\$CLOSE	
FF38	C2		10	88 0006E	BISB2	#16, I0BS+46	
			02	90 00073	MOVB	#2, I0BS+44	
			53	DD 00078	PUSHL	R3	
			7E	D4 0007A	CLRL	-(SP)	
		FF0C	C2	9F 0007C	PUSHAB	I0BS	
02	65		03	FB 00080	CALLS	#3, XPOS\$CLOSE	
	A6		10	88 00083	BISB2	#16, I0BS+46	
	66		02	90 00087	MOVB	#2, I0BS+44	
			53	DD 0008A	PUSHL	R3	
			7E	D4 0008C	CLRL	-(SP)	
		D4	A6	9F 0008E	PUSHAB	I0BS	
65			03	FB 00091	CALLS	#3, XPOS\$CLOSE	
01	0100		C2	D1 00094	CMPL	CAPTYPE, #1	
			6D	12 00099	BNEQ	3\$	
67			01	E1 0009B	BBC	#1, CMDBLK, 1\$	
			09	DD 0009F	PUSHL	#9	
			08	DD 000A1	PUSHL	#8	
			07	DD 000A3	PUSHL	#7	
	00000000G		EF	9F 000A5	PUSHAB	TMS_EX	
68			52	DD 000AB	PUSHL	R2	
			05	FB 000AD	CALLS	#5, HDR_X	
			0C	DD 000B0	PUSHL	#12	
			0B	DD 000B2	PUSHL	#11	
			0A	DD 000B4	PUSHL	#10	
	00000000G		EF	9F 000B6	PUSHAB	TMS FG	
68	FE18		C2	9F 000BC	PUSHAB	FGI0B	
68			05	FB 000C0	CALLS	#5, HDR_X	
			0F	DD 000C3	PUSHL	#15	
			0E	DD 000C5	PUSHL	#14	
			0D	DD 000C7	PUSHL	#13	
	00000000G		EF	9F 000C9	PUSHAB	TMS_TB	
			30	11 000CF	BRB	2\$	
			09	DD 000D1	1\$:	PUSHL	#9
			08	DD 000D3	PUSHL	#8	
			07	DD 000D5	PUSHL	#7	
	00000000G		EF	9F 000D7	PUSHAB	RNO_EX	
68			52	DD 000DD	PUSHL	R2	
68			05	FB 000DF	CALLS	#5, HDR_X	
			0C	DD 000E2	PUSHL	#12	
			0B	DD 000E4	PUSHL	#11	
			0A	DD 000E6	PUSHL	#10	
	00000000G		EF	9F 000E8	PUSHAB	RNO FG	
68	FE18		C2	9F 000EE	PUSHAB	FGI0B	
68			05	FB 000F2	CALLS	#5, HDR_X	
			0F	DD 000F5	PUSHL	#15	
			0E	DD 000F7	PUSHL	#14	
			0D	DD 000F9	PUSHL	#13	
	00000000G		EF	9F 000FB	PUSHAB	RNO TB	
	FF0C		C2	9F 00101	2\$:	PUSHAB	TBI0B
68			05	FB 00105	CALLS	#5, HDR_X	
2C	EF		00	FB 00108	3\$:	CALLS	#0, STCADR
	A2		03	90 0010F	MOVB	#3, I0BS+44	
			53	DD 00113	PUSHL	R3	
			7E	D4 00115	CLRL	-(SP)	
			52	DD 00117	PUSHL	R2	

CONTENTS V04-000

CONTENTS - Table of Contents formatter program 16-Sep-1984 00:01:05 VAX-11 Bliss-32 v4.0-742
TOCFIN - Finish table of contents processing 14-Sep-1984 13:05:48 [RUNOFF.SRC]CONTENTS.BLI;1

N 1

16-Sep-1984 00:01:05

VAX-11 Bliss-32 V4.0-742

Page 84
(7)

FE44	69	C2	03	FB	00119	CALLS	#3, XPOSDELETE						2631
			03	90	0011C	MOVB	#3, IOBS+44						
			53	DD	00121	PUSHL	R3						
			7E	D4	00123	CLRL	-(SP)						
FF38	69	C2	C2	9F	00125	PUSHAB	IOBS						2632
			03	FB	00129	CALLS	#3, XPOSDELETE						
			03	90	0012C	MOVB	#3, IOBS+44						
			53	DD	00131	PUSHL	R3						
			7E	D4	00133	CLRL	-(SP)						
	69	FF0C	C2	9F	00135	PUSHAB	IOBS						2633
			03	FB	00139	CALLS	#3, XPOSDELETE						
			03	90	0013C	MOVB	#3, IOBS+44						
			53	DD	0013F	PUSHL	R3						
			7E	D4	00141	CLRL	-(SP)						
OC	69	D4	A6	9F	00143	PUSHAB	IOBS						2637
	67		03	FB	00146	CALLS	#3, XPOSDELETE						2639
			01	E1	00149	BBC	#1, CMDBLK, 4\$						
			5B	DD	0014D	PUSHL	R11						
			7E	D4	0014F	CLRL	-(SP)						
			5A	DD	00151	PUSHL	R10						
		0274	C2	9F	00153	PUSHAB	\$STR\$STRING						2641
			0A	11	00157	BRB	5\$						
			5B	DD	00159	4\$:	PUSHL	R11					
			7E	D4	0015B	CLRL	-(SP)						
			5A	DD	0015D	PUSHL	R10						
		027C	C2	9F	0015F	PUSHAB	\$STR\$STRING						
			7E	D4	00163	CLRL	-(SP)						
00000000G	EF		05	FB	00165	CALLS	#5, XST\$COPY						
	50		6A	3C	0016C	MOVZWL	TMPSTR, R0						
00000000G	EF		50	C0	0016F	ADDL2	R0, CHROUT						
18	A4		6A	9E	00176	MOVAB	\$10B\$OUTPUT, IOBS+68						
	64		07	90	0017A	MOVB	#7, IOBS+44						
			53	DD	0017D	PUSHL	R3						
			7E	D4	0017F	CLRL	-(SP)						
00000000G	EF	D4	A4	9F	00181	PUSHAB	IOBS						
	67		03	FB	00184	CALLS	#3, XPOSPUT						
00000000G	EF		01	E1	0018B	BBC	#1, CMDBLK, 6\$						
12	67		00	FB	0018F	CALLS	#0, SPLIT						2646
		05	E1	00196	6\$:	BBC	#5, CMDBLK, 7\$						2650
		F0	A4	9F	0019A	PUSHAB	TOCOOB+28						
			01	DD	0019D	PUSHL	#1						
00000000G	00	00000000G	8F	DD	0019F	PUSHL	#DSRTOCS CREATED						
	0F		03	FB	001A5	CALLS	#3, LIB\$SIGNAL						2656
	64		67	E9	001AC	7\$:	BLBC	CMDBLK, 8\$					2658
			02	90	001AF	MOVB	#2, IOBS+44						
			53	DD	001B2	PUSHL	R3						
			7E	D4	001B4	CLRL	-(SP)						
			A4	9F	001B6	PUSHAB	IOBS						
02	65	D4	03	FB	001B9	CALLS	#3, XPOSCLOSE						
			1E	11	001BC	BRB	9\$						
			10	88	001BE	8\$:	BISB2	#16, IOBS+46					2665
			0C	90	001C2	MOVB	#2, IOBS+44						
			53	DD	001C5	PUSHL	R3						
			7E	D4	001C7	CLRL	-(SP)						
			A4	9F	001C9	PUSHAB	IOBS						
			03	FB	001CC	CALLS	#3, XPOSCLOSE						
			03	90	001CF	MOVB	#3, IOBS+44						2666

CONTENTS
VO4-000

CONTENTS - Table of Contents formatter program B 14
TOCFIN - Finish table of contents processing 16-Sep-1984 00:01:05
 14-Sep-1984 13:05:48
 VAX-11 Bliss-32 v4.0-742
 [RUNOFF.SRC]CONTENTS.BLI;1

Page 85
(7)

	53	DD	001D2	PUSHL	R3
	7E	D4	001D4	CLRL	-(SP)
69	D4	A4	9F 001D6	PUSHAB	I0B\$
		03	FB 001D9	CALLS	#3, XPO\$DELETE
50		01	DO 001DC 9\$:	MOVL	#1, R0
		04	001DF	RET	

: 2669
: 2670

: Routine Size: 480 bytes, Routine Base: \$CODE\$ + 1F19

```
: 2044
: 2045   2671 1 %SBTTL 'VFYCAP - Check caption type'
: 2046   2672 1 ROUTINE vfycap (type) =
: 2047   2673 1 ++
: 2048   2674 1
: 2049   2675 1 FUNCTIONAL DESCRIPTION:
: 2050   2676 1
: 2051   2677 1 This routine checks the caption type being processed against the
: 2052   2678 1 argument caption type. If they are the same, a value of true is
: 2053   2679 1 returned.
: 2054   2680 1
: 2055   2681 1 If they are different, a message is written if no message has been
: 2056   2682 1 written already, and false is returned.
: 2057   2683 1
: 2058   2684 1 If no caption type has been set yet, the caption type is set from
: 2059   2685 1 the argument type and true is returned.
: 2060   2686 1
: 2061   2687 1 FORMAL PARAMETERS:
: 2062   2688 1
: 2063   2689 1     type      - argument caption type
: 2064   2690 1
: 2065   2691 1 IMPLICIT INPUTS:
: 2066   2692 1
: 2067   2693 1     captyp    - caption type being processed
: 2068   2694 1     capmsg   - true if a message has been written
: 2069   2695 1
: 2070   2696 1 IMPLICIT OUTPUTS:
: 2071   2697 1
: 2072   2698 1     captyp    - will be set if it is zero
: 2073   2699 1     capmsg   - will be set to true if a message is written
: 2074   2700 1
: 2075   2701 1 ROUTINE VALUE:
: 2076   2702 1 COMPLETION CODES:
: 2077   2703 1
: 2078   2704 1     See functional description above.
: 2079   2705 1
: 2080   2706 1 SIDE EFFECTS:
: 2081   2707 1
: 2082   2708 1     None
: 2083   2709 1 ---+
: 2084   2710 1
: 2085   2711 2 BEGIN
: 2086   2712 2
: 2087   2713 2 SELECTONE .captyp OF
: 2088   2714 2     SET
: 2089   2715 2
: 2090   2716 2     [old_captions] :
: 2091   2717 2
: 2092   2718 2     IF .type EQL new_captions
: 2093   2719 2     THEN
: 2094   2720 3         BEGIN
: 2095   2721 3
: 2096   2722 3         IF NOT .capmsg
: 2097   2723 3         THEN
: 2098   2724 4             BEGIN
: 2099   2725 4                 capmsg = true;
: 2100   2726 4
: L 2727 4 %IF %BLISS (BLISS32)
```

```

2101      2728 4 %THEN
2102      2729 4
2103      U 2730 4 %ELSE
2104      U 2731 4
2105      U 2732 4 SIGNAL (contents$_captions, 0, contents$_ignorenew);
2106      U 2733 4 $xpo_put_msg (severity = warning,
2107      U 2734 4 string = 'both .SEND TOC captions and ".HEADER x" captions encountered',
2108      U 2735 4 string = '".HEADER x" captions will be ignored');
2109      U 2736 3 %
2110      U 2737 3 END;
2111      U 2738 3 RETURN false;
2112      U 2739 2 END;
2113      U 2740 2
2114      U 2741 2 [new_captions] :
2115      U 2742 2
2116      U 2743 2 IF .type EQL old_captions
2117      U 2744 2 THEN
2118      U 2745 3 BEGIN
2119      U 2746 3
2120      U 2747 3 IF NOT .capmsg
2121      U 2748 3 THEN
2122      U 2749 4 BEGIN
2123      U 2750 4 capmsg = true;
2124      U 2751 4
2125      L 2752 4 %IF %BLISS (BLISS32)
2126      L 2753 4 %THEN
2127      L 2754 4 SIGNAL (contents$_captions, 0, contents$_ignoreold);
2128      L 2755 4 %ELSE
2129      L 2756 4
2130      L 2757 4 $xpo_put_msg (severity = warning,
2131      L 2758 4 string = 'both .SEND TOC captions and ".HEADER x" captions encountered',
2132      L 2759 4 string = '.SEND TOC captions will be ignored');
2133      L 2760 4 %FI
2134      L 2761 3 %
2135      L 2762 3
2136      L 2763 3 RETURN false;
2137      L 2764 2 END;
2138      L 2765 2
2139      L 2766 2 [OTHERWISE] :
2140      L 2767 2 captyp = .type;
2141      L 2768 2
2142      L 2769 2 TES;
2143      L 2770 2
2144      L 2771 2 RETURN true;
2145      L 2772 1 END;

```

		0004	00000	VFYCAP:	.WORD	Save R2	: 2672
	52	00000000'	EF	9E	00002	MOVAB	CAPMSG, R2
	50	FC	A2	D0	00009	MOVL	CAPTYP, R0
FFFFFFFFFF	8F		50	D1	0000D	CMPL	R0, #-1
			14	12	00014	BNEQ	1\$
	01	04	AC	D1	00016	CMPL	TYPE, #1
			3F	12	0001A	BNEQ	4\$

CONTENTS
V04-000CONTENTS - Table of Contents formatter program E 14
VFYCAP - Check caption type 16-Sep-1984 00:01:05 VAX-11 Bliss-32 v4.0-742
[RUNOFF.SRC]CONTENTS.BLI;1Page 88
(8)

40		62	E8 0001C	BLBS	CAPMSG, 5\$	2722
62		01	D0 0001F	MOVL	#1, CAPMSG	2725
	00000000G	8F	DD 00022	PUSHL	#D\$RTOC\$_IGNORENEW	2729
		1B	11 00028	BRB	2\$	
		01	50 D1 0002A	1\$: CMPL	R0, #1	2741
FFFFFFF	8F	04	AC D1 0002F	BNEQ	3\$	
		27	12 0002D	CMPL	TYPE, #-1	2743
		22	12 00037	BNEQ	4\$	
	23	62	E8 00039	BLBS	CAPMSG, 5\$	2747
	62	01	D0 0003C	MOVL	#1, CAPMSG	2750
	00000000G	8F	DD 0003F	PUSHL	#D\$RTOC\$_IGNOREOLD	2754
		7E	D4 00045	2\$: CLR	-(SP)	
00000000G	00	00	03 FB 00047	PUSHL	#D\$RTOC\$_CAPTIONS	
		09	11 00054	CALLS	#3, LIB\$SIGNAL	
	FC	A2	04 AC D0 00056	3\$: BRB	5\$	2763
	50	04	0005B	4\$: MOVL	TYPE, CAPTYP	2767
		50	04 0005E	RET	#1, R0	2771
		04	00061	5\$: CLR	R0	2772
				RET		

; Routine Size: 98 bytes, Routine Base: \$CODE\$ + 20F9

```
: 2147 2773 1 %SBTTL 'OUTSEC - Format a section number'
2148 2774 1 ROUTINE outsec : NOVALUE =
2149 2775 1 ++
2150 2776 1 FUNCTIONAL DESCRIPTION:
2151 2777 1
2152 2778 1 OUTSEC generates a section number of the form 1.2.3.....
2153 2779 1
2154 2780 1 FORMAL PARAMETERS:
2155 2781 1
2156 2782 1 None
2157 2783 1
2158 2784 1 IMPLICIT INPUTS:
2159 2785 1
2160 2786 1 cmdblk - command line information block
2161 2787 1 lasthl - previous HL,"n"
2162 2788 1 hl_n - current HL,"n"
2163 2789 1 minor - minor record type code
2164 2790 1 pagen - current page number
2165 2791 1 hllist - header level counters
2166 2792 1 hldsp - header level display codes
2167 2793 1 - output line variables
2168 2794 1
2169 2795 1 IMPLICIT OUTPUTS:
2170 2796 1
2171 2797 1 ptrsct - modified
2172 2798 1 lensct - length of section number
2173 2799 1 txtsct - contains formatted section number
2174 2800 1 - output line variables modified (since section number is
2175 2801 1 inserted into the output line)
2176 2802 1
2177 2803 1 ROUTINE VALUE:
2178 2804 1 COMPLETION CODES:
2179 2805 1
2180 2806 1 None
2181 2807 1
2182 2808 1 SIDE EFFECTS:
2183 2809 1
2184 2810 1 None
2185 2811 1
2186 2812 1 --
2187 2813 1
2188 2814 2 BEGIN
2189 2815 2
2190 2816 2 LOCAL
2191 2817 2     hl_width; ! Number of columns reserved for header level.
2192 2818 2
2193 2819 2     lensct = 0;
2194 2820 2     ptrsct = CH$PTR (txtsct);
2195 2821 2
2196 2822 2     IF .cmdblk [contents$v_tms11]
2197 2823 2     THEN
2198 2824 3         BEGIN
2199 2825 3             ! Write out the HL level for TMS
2200 2826 3
2201 2827 3
2202 2828 3     IF (.lasthl NEQ .hl_n) AND (.lasthl NEQ 0) THEN put ('[va5]');
2203 2829 3
```

```
2204      2830 3    literal_text ('*c');
2205      2831 3    lensct = pacbas (.hl_n, ptrsct, 10);
2206      2832 3    ptrsct = CH$PTR (txt$ct);
2207      2833 3    CH$MOVE (.lensct, .ptrsct, .lp);
2208      2834 3    lp = CH$PLUS (.lp, .lensct);
2209      2835 3    intlin = .intlin + .lensct;
2210      2836 3    lensct = 0;
2211      2837 3    literal_text ('n*');
2212      2838 2    END;
2213      2839 2
2214      2840 2    IF .cmdblk [contents$v_include_sections]
2215      2841 3    AND (
2216      2842 4      (.cmdblk [contents$g_sections] GEQ .hl_n)
2217      2843 4      OR (
2218      2844 5          (.cmdblk [contents$g_sections] EQL -1)
2219      2845 5          AND (.minor NEQ min_hl_txt_u)
2220      2846 4          )
2221      2847 3      )
2222      2848 2    THEN BEGIN
2223      2849 3
2224      2850 3
2225      2851 3    Don't write the number if user said heads are unnumbered.
2226      2852 3    The user must have requested numbered heads and
2227      2853 3    the header level must be less than the largest one the user
2228      2854 3    specified or the heads must be displayed as input and the
2229      2855 3    text is not for an unnumbered header.
2230      2856 3
2231      2857 4    IF (.pagen [sct_typ] EQL 0) AND (.hl_n EQL 1)
2232      2858 3    THEN BEGIN
2233      2859 4
2234      2860 4    This is a HL 1 for a document not divided into sections
2235      2861 4
2236      2862 4
2237      2863 4    lensct = pacxxx (.hllist [0], ptrsct, .hldsp [0]);
2238      2864 4    END
2239      2865 3    ELSE BEGIN
2240      2866 4
2241      2867 4    IF .pagen [sct_typ] NEQ 0
2242      2868 4    THEN BEGIN
2243      2869 4    Document is divided into sections
2244      2870 5
2245      2871 5    lensct = pacsec (pagen, p$rsct);
2246      2872 5    CH$WCHAR_A (%C:, ptrsct);
2247      2873 5    lensct = .lensct + 1;
2248      2874 5    END;
2249      2875 5
2250      2876 5    INCR i FROM 1 TO .hl_n DO
2251      2877 4    BEGIN
2252      2878 4    lensct = .lensct + pacxxx (.hllist [.i - 1], ptrsct, .hldsp [.i - 1]);
2253      2879 4    CH$WCHAR_A (%C:, ptrsct);
2254      2880 5    lensct = .lensct + 1;
2255      2881 5    END;
2256      2882 5
2257      2883 5
2258      2884 4
2259      2885 4
2260      2886 4    !
```

CONTENTS - Table of Contents formatter program H 14
OUTSEC - Format a section number 16-Sep-1984 00:01:05 VAX-11 Bliss-32 v4.0-742
14-Sep-1984 13:05:48 [RUNOFF.SRC]CONTENTS.BLI:1

: 2261 2887 4 | Always get an extra dot, so uncount it.
: 2262 2888 4
: 2263 2889 4 lensct = .lensct - 1;
: 2264 2890 3 END;
: 2265 2891 3
: 2266 2892 3
: 2267 2893 3 Now copy the converted section number into the line that's being built up.
: 2268 2894 3
: 2269 2895 3
: 2270 2896 3
: 2271 2897 3
: 2272 2898 2 IF NOT .cmdblk [contents\$v_tms11] THEN write_char (%C' ', counts_visually);
: 2273 2899 2 END;
: 2274 2900 2
: 2275 2901 2 IF .cmdblk [contents\$v_tms11]
: 2276 2902 2 THEN write_char (%C'a')
: 2277 2903 2 ELSE BEGIN
: 2278 2904 2
: 2279 2905 2
: 2280 2906 3 For RUNOFF...
: 2281 2907 3
: 2282 2908 3 Compute the width of the region into which the header level
: 2283 2909 3 number was supposed to fit.
: 2284 2910 3
: 2285 2911 3
: 2286 2912 3
: 2287 2913 3
: 2288 2914 3 hl_width = 0;
: 2289 2915 3
: 2290 2916 3
: 2291 2917 3 Pad out this .HL number to fill the columns allocated.
: 2292 2918 3
: 2293 2919 3 pad ((.hl_width - .extlin));
: 2294 2920 2
: 2295 2921 2
: 2296 2922 1 END; ! End of outsec

.PSECT \$PLIT\$,NOWRT,NOEXE,2

5D 35 61 76 5B 0028F P.ACZ: .ASCII \[va5]\
00 00 63 2A 00294 P.ADA: .ASCII *c\<0><0>
00 00 2A 6E 00298 P.ADB: .ASCII \n*\<0><0>

.PSECT \$OWN\$,NOEXE,2

0005 00564 \$STR\$STRING:
01 0E 00566 .WORD 5
00000000 00568 .BYTE 14, 1
.ADDRESS P.ACZ

.PSECT \$CODE\$,NOWRT,2

0FFC 00000 OUTSEC: .WORD Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11

CONTENTS V04-000

CONTENTS - Table of Contents formatter program
OUTSEC - Format a section number

J 14
16-Sep-1984 00:01:0
14-Sep-1984 13:05:4

VAX-11 Bliss-32 v4.0-742
[RUNOFF.SRC]CONTENTS.BLI:1

Page 93
(9)

CONTENTS K 14
V04-000 CONTENTS - Table of Contents formatter program 16-Sep-1984 00:01:05
OUTSEC - Format a section number 14-Sep-1984 13:05:48 VAX-11 Bliss-32 V4.0-742
[RUNOFF.SRC]CONTENTS.BLI;1 Page 94
(9)

; Routine Size: 507 bytes, Routine Base: \$CODE\$ + 215B

; 2297 2923 1
; 2298 2924 1 END
; 2299 2925 0 ELUDGM

! End of module

.EXTRN LIB\$SIGNAL

PSECT SUMMARY

Name	Bytes	Attributes
\$OWN\$	1388	NOVEC, WRT, RD ,NOEXE,NOSHR, LCL, REL, CON,NOPIC,ALIGN(2)
SPLIT\$	668	NOVEC,NOWRT, RD ,NOEXE,NOSHR, LCL, REL, CON,NOPIC,ALIGN(2)
\$CODE\$	9046	NOVEC,NOWRT, RD , EXE,NOSHR, LCL, REL, CON,NOPIC,ALIGN(2)

Library Statistics

File	Symbols			Pages Mapped	Processing Time
	Total	Loaded	Percent		
\$_\$255\$DUA28:[SYSLIB]XPORT.L32;1	590	219	37	252	00:00.1

COMMAND QUALIFIERS

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

; Size: 9046 code + 2056 data bytes
; Run Time: 04:38.3
; Elapsed Time: 08:52.8
; Lines/CPU Min: 630
; Lexemes/CPU-Min: 91165
; Memory Used: 1086 pages
; Compilation Complete

0338 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

