

RRRRRRRRRR		UUU		UUU	NNN		NNN	00000000		FFFFFFFFFFFF		FFFFFFFFFFFF
RRRRRRRRRR		UUU		UUU	NNN		NNN	00000000		FFFFFFFFFFFF		FFFFFFFFFFFF
RRRRRRRRRR		UUU		UUU	NNN		NNN	00000000		FFFFFFFFFFFF		FFFFFFFFFFFF
RRR	RRR	UUU		UUU	NNN		NNN	000	000	FFF		FFF
RRR	RRR	UUU		UUU	NNN		NNN	000	000	FFF		FFF
RRR	RRR	UUU		UUU	NNN		NNN	000	000	FFF		FFF
RRR	RRR	UUU		UUU	NNNNNN		NNN	000	000	FFF		FFF
RRR	RRR	UUU		UUU	NNNNNN		NNN	000	000	FFF		FFF
RRR	RRR	UUU		UUU	NNNNNN		NNN	000	000	FFF		FFF
RRRRRRRRRR		UUU		UUU	NNN	NNN	NNN	000	000	FFFFFFFFFFFF		FFFFFFFFFFFF
RRRRRRRRRR		UUU		UUU	NNN	NNN	NNN	000	000	FFFFFFFFFFFF		FFFFFFFFFFFF
RRRRRRRRRR		UUU		UUU	NNN	NNN	NNN	000	000	FFFFFFFFFFFF		FFFFFFFFFFFF
RRR	RRR	UUU		UUU	NNN	NNNNNN	NNN	000	000	FFF		FFF
RRR	RRR	UUU		UUU	NNN	NNNNNN	NNN	000	000	FFF		FFF
RRR	RRR	UUU		UUU	NNN	NNNNNN	NNN	000	000	FFF		FFF
RRR	RRR	UUU		UUU	NNN	NNN	NNN	000	000	FFF		FFF
RRR	RRR	UUU		UUU	NNN	NNN	NNN	000	000	FFF		FFF
RRR	RRR	UUU		UUU	NNN	NNN	NNN	000	000	FFF		FFF
RRR	RRR	UUU		UUU	NNN	NNN	NNN	000	000	FFF		FFF
RRR	RRR	UUUUUUUUUUUUUUUU		UUUUUUUUUUUUUUUU	NNN	NNN	NNN	00000000		FFF		FFF
RRR	RRR	UUUUUUUUUUUUUUUU		UUUUUUUUUUUUUUUU	NNN	NNN	NNN	00000000		FFF		FFF
RRR	RRR	UUUUUUUUUUUUUUUU		UUUUUUUUUUUUUUUU	NNN	NNN	NNN	00000000		FFF		FFF

```

RRRRRRRR      NN      NN      000000      VV      VV      MM      MM      SSSSSSSS
RRRRRRRR      NN      NN      000000      VV      VV      MM      MM      SSSSSSSS
RR      RR      NN      NN      00      00      VV      VV      MMMM      MMMM      SS
RR      RR      NN      NN      00      00      VV      VV      MMMM      MMMM      SS
RR      RR      NNNN      NN      00      00      VV      VV      MM      MM      SS
RR      RR      NNNN      NN      00      00      VV      VV      MM      MM      SS
RRRRRRRR      NN      NN      NN      00      00      VV      VV      MM      MM      SSSSSS
RRRRRRRR      NN      NN      NN      00      00      VV      VV      MM      MM      SSSSSS
RR      RR      NN      NNNN      00      00      VV      VV      MM      MM      SS
RR      RR      NN      NNNN      00      00      VV      VV      MM      MM      SS
RR      RR      NN      NN      00      00      VV      VV      MM      MM      SS
RR      RR      NN      NN      00      00      VV      VV      MM      MM      SS
RR      RR      NN      NN      00      00      VV      VV      MM      MM      SS
RR      RR      NN      NN      00      00      VV      VV      MM      MM      SS
RR      RR      NN      NN      000000      VV      MM      MM      SSSSSSSS
RR      RR      NN      NN      000000      VV      MM      MM      SSSSSSSS

```

```

LL      IIIIII      SSSSSSSS
LL      IIIIII      SSSSSSSS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SSSSSS
LL      II      SSSSSS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SS
LLLLLLLLLLLL      IIIIII      SSSSSSSS
LLLLLLLLLLLL      IIIIII      SSSSSSSS

```

```

1 0001 0 %TITLE 'RNOVMS - DSR VMS Command Line Interface'
2 0002 0 MODULE RNOVMS ( IDENT = 'V04-000'
3 0003 0 ADDRESSING_MODE( EXTERNAL=LONG_RELATIVE,
4 0004 0 NONEXTERNAL=LONG_RELATIVE )
5 0005 0 ) =
6 0006 1 BEGIN
7 0007 1
8 0008 1
9 0009 1 *****
10 0010 1 *
11 0011 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY *
12 0012 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS. *
13 0013 1 * ALL RIGHTS RESERVED. *
14 0014 1 *
15 0015 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED *
16 0016 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE *
17 0017 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER *
18 0018 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY *
19 0019 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY *
20 0020 1 * TRANSFERRED. *
21 0021 1 *
22 0022 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE *
23 0023 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT *
24 0024 1 * CORPORATION. *
25 0025 1 *
26 0026 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS *
27 0027 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL. *
28 0028 1 *
29 0029 1 *
30 0030 1 *****
31 0031 1
32 0032 1
33 0033 1 **
34 0034 1
35 0035 1 FACILITY: DSR (Digital Standard RUNOFF) / DSRPLUS
36 0036 1
37 0037 1 ABSTRACT:
38 0038 1
39 0039 1 This module is the DSR VMS Command Line interface module.
40 0040 1
41 0041 1 ENVIRONMENT: VAX/VMS User Mode
42 0042 1
43 0043 1 AUTHOR: Ray Marshall, CREATION DATE: 23 August 1981
44 0044 1

```

```
46 0045 1 %SBTTL 'Revision History'
47 0046 1
48 0047 1   MODIFIED BY:
49 0048 1
50 0049 1       026   REM00026   Ray Marshall   31-May-1984
51 0050 1           Added a dot to a reference to dbg2 where it gets bit 15 set on
52 0051 1           the user saying /DEC_INTERNAL=OUTPUT_LINE_NUMBER.  Dbg2 was
53 0052 1           being loaded with its own address!
54 0053 1
55 0054 1       025   REM00025   Ray Marshall   17-May-1984
56 0055 1           Added parameter OUTPUT_LINE_NUMBER to /DEC_INTERNAL.  This
57 0056 1           parameter sets bit 15 of debug flag word number 2.  The
58 0057 1           meaning of this bit is to force output line numbers to be put
59 0058 1           into the left margin of each line of text in the output file.
60 0059 1           Not controlled by any special parameter is but 14 which will
61 0060 1           prefix the line number with the internal output page number
62 0061 1           followed by a period.  This second bit will have to be set by
63 0062 1           the /DEC_INTERNAL=DBG2=XX4000 qualifier.
64 0063 1
65 0064 1       024   REM00024   Ray Marshall   11-April-1984
66 0065 1           Make /DEVICE return an error for invalid entries.
67 0066 1           Conditionalized the DIABLO, FLIP, & VT100 parameters of the
68 0067 1           /DEVICE qualifier for DSRPLUS only.  This way, with the
69 0068 1           change mentioned in the above entry, they will now yeald
70 0069 1           errors when given to DSR.
71 0070 1           Made some changes towards making the /FORM work better when
72 0071 1           /SIMULATE isn't specified.  These are incomplete and mostly
73 0072 1           commented out for now.  The variable to look for to get to
74 0073 1           this new work is FORM_QUALIFIER.
75 0074 1
76 0075 1       023   REM00023   Ray Marshall   24-March-1984
77 0076 1           Fixed definition of LIB$FIND_FILE_END.
78 0077 1
79 0078 1       022   REM00022   Ray Marshall   15-March-1984
80 0079 1           Added logic to initialize and cleanup for LIB$FIND_FILE.  This
81 0080 1           routine is now called from RUNOFF.BLI to support search-lists
82 0081 1           as implemented in VMS V4.
83 0082 1
84 0083 1       021   REM00021   Ray Marshall   5-December-1983
85 0084 1           Add [NO]LOAD parameter to the /DEVICE qualifier and make it
86 0085 1           related to the LN01[E] parameter.
87 0086 1
88 0087 1       020   REM00020   Ray Marshall   13-September-1983
89 0088 1           Increase the logical name buffer size from 64 to 256 to
90 0089 1           accommodate the larger file specifications for VMS V4.0.
91 0090 1           Add [NO]HEADER parameter to the /DEVICE qualifier and make it
92 0091 1           related to the LN01[E] parameter.
93 0092 1
94 0093 1       019   KAD00019   Keith Dawson   2-Jun-1983
95 0094 1           Set bits RNO$V_S_RIGHT and RNO$V_S_DOWN if the qualifiers
96 0095 1           were explicitly given.
97 0096 1
98 0097 1       018   REM00018   Ray Marshall   24-May-1983
99 0098 1           Modify exit logic so that if one of several calls to RUNOFF
100 0099 1           returns a failure, we will also return to VMS with a failure
101 0100 1           status code.
102 0101 1
```

103	0102	1	017	REM00017	Ray Marshall	11-May-1983	
104	0103	1		Removed the check for error returns from the call to RUNOFF.			
105	0104	1		This will allow a list of input files to continue being			
106	0105	1		processed if one in the middle has errors.			
107	0106	1					
108	0107	1	016	REM00016	Ray Marshall	10-May-1983	
109	0108	1		Added call to SYSSPURGWS to purge the user's working set.			
110	0109	1		Removed the .L32 extensions from all LIBRARY commands.			
111	0110	1		Deconditionalized the /DEBUG=SAVE RESTORE parameter logic to			
112	0111	1		allow it to be compiled into DSR as well as DSRPLUS.			
113	0112	1		Many small changes to the TPARSE tables to catch unprocessed			
114	0113	1		parameter text and report it as an error.			
115	0114	1		Single character parameter processing for all "character"			
116	0115	1		expecting qualifiers has been moved out of the TPARSE tables.			
117	0116	1		Now, if the parameter string is just one (1) character long,			
118	0117	1		that character will be put directly into it's proper resultant			
119	0118	1		field and TPARSE will not be called.			
120	0119	1		Removed %O, %D, & %X logic from /DEVICE TPARSE tables because			
121	0120	1		it was never used.			
122	0121	1					
123	0122	1	015	KAD00015	Keith Dawson	14-Apr-1983	
124	0123	1		Support /DEVICE=FLIP.			
125	0124	1					
126	0125	1	014	KAD00014	Keith Dawson	20-Mar-1983	
127	0126	1		Removed LN01 conditionals and all references to .BIX			
128	0127	1		and .BTC files.			
129	0128	1		Removed /OVERPRINT.			
130	0129	1		Changed GCA_FLIP bit to (.gca_op_dev EQL op_dev_flip).			
131	0130	1					
132	0131	1	013	KAD00013	Keith Dawson	07-Mar-1983	
133	0132	1		Global edit of all modules. Updated module names, idents,			
134	0133	1		copyright dates. Changed require files to BLISS library.			
135	0134	1					
136	0135	1	--				

```
138 0136 1 %SBTTL 'Module Level Declarations'
139 0137 1
140 0138 1 ! This literal prevents the new RNODAT from being compiled in. It also
141 0139 1 ! prevents any reference to it.
142 0140 1
143 0141 1 LITERAL
144 0142 1     NEW_RNODAT_MODULE = 1;           ! Allow the new RNODAT.
145 0143 1
146 0144 1
147 0145 1 ! TABLE OF CONTENTS:
148 0146 1 !
149 0147 1
150 0148 1 FORWARD ROUTINE
151 0149 1
152 0150 1     rnocli,                       ! DSR VMS command interface routine
153 0151 1     call_tparse,                   ! Causes TPARSE to put data into command table
154 0152 1     save_value,                   ! Used by TPARSE to store a DBG1 or DBG2 value
155 0153 1     set_output_device_code
156 0154 1
157 0155 1 %IF NEW_RNODAT_MODULE %THEN .
158 0156 1     rnodat : NOVALUE
159 0157 1 %FI
160 0158 1
161 0159 1 !
162 0160 1 ! INCLUDE FILES:
163 0161 1 !
164 0162 1
165 0163 1 LIBRARY 'SYSS$LIBRARY:STARLET'; ! System macros
166 0164 1 LIBRARY 'SYSS$LIBRARY:TPAMAC'; ! TPARSE macros
167 0165 1
168 0166 1 LIBRARY 'NXPORT:XPORT';           ! XPORT Library
169 0167 1 REQUIRE 'REQ:RNODEF';           ! RUNOFF variant definitions
170 0298 1
171 0299 1 undeclare %quote $descriptor;
172 0300 1
173 U 0301 1 %IF DSRPLUS %THEN
174 U 0302 1 LIBRARY 'REQ:DPLLIB';         ! DSRPLUS BLISS Library
175 0303 1 %ELSE
176 0304 1 LIBRARY 'REQ:DSRLIB';         ! DSR BLISS Library
177 0305 1 %FI
178 0306 1
179 0307 1 STRUCTURE
180 0308 1     BBLOCK [O, P, S, E; N] =
181 0309 1         [N]
182 0310 1         (BBLOCK + O) <P, S, E>;
183 0311 1
184 0312 1 ! MACROS:
185 0313 1 !
186 0314 1 !
187 0315 1 !     Define internal descriptors
188 0316 1 !
189 0317 1 MACRO ! (n,'text')
190 0318 1     CSTRING[] = %STRING( %CHAR(%CHARCOUNT(%REMAINING)),%REMAINING)%,
191 0319 1
192 M 0320 1     string_descr [] = (uplit (%charcount (%remaining),
193 0321 1         ch$ptr (uplit (%string (%remaining))))),
194 0322 1     swch_automatic = string_descr('AUTOMATIC') %,
```

```
195 0323 1 swch_backspace = string_descr('BACKSPACE') %,
196 0324 1 swch_bold = string_descr('BOLD') %,
197 0325 1 swch_change = string_descr('CHANGE BARS') %,
198 0326 1 swch_contents = string_descr('CONTENTS') %,
199 0327 1 swch_cross_reference = string_descr('CROSS REFERENCE') %,
200 0328 1 swch_dec_internal = string_descr('DEC_INTERNAL') %,
201 0329 1 swch_debug = string_descr('DEBUG') %,
202 0330 1 swch_device = string_descr('DEVICE') %,
203 0331 1 swch_down = string_descr('DOWN') %,
204 0332 1 swch_form_size = string_descr('FORM SIZE') %,
205 0333 1 swch_index = string_descr('INDEX') %,
206 0334 1 swch_input = string_descr('PI') %,
207 0335 1 swch_intermediate = string_descr('INTERMEDIATE') %,
208 0336 1 swch_log = string_descr('LOG') %,
209 0337 1 swch_messages = string_descr('MESSAGES') %,
210 0338 1 swch_output = string_descr('OUTPUT') %,
211 0339 1 swch_pages = string_descr('PAGES') %,
212 0340 1 swch_pause = string_descr('PAUSE') %,
213 0341 1 swch_right = string_descr('RIGHT') %,
214 0342 1 swch_separate = string_descr('SEPARATE UNDERLINE') %,
215 0343 1 swch_sequence = string_descr('SEQUENCE') %,
216 0344 1 swch_simulate = string_descr('SIMULATE') %,
217 0345 1 swch_spacing = string_descr('NONSPACING UNDERLINE') %,
218 0346 1 swch_underline = string_descr('UNDERLINE CHARACTER') %,
219 0347 1 swch_variant = string_descr('VARIANT') %;
220 0348 1
221 0349 1 !
222 0350 1 ! EQUATED SYMBOLS:
223 0351 1 !
224 0352 1
225 0353 1 LITERAL
226 0354 1 yes = 1, ! Used to turn indicators on
227 0355 1 no = 0, ! Used to turn indicators off
228 0356 1 logical_name_buffer_length = 256;
229 0357 1
230 0358 1 $LITERAL ! Literals used to dispatch within the SAVE_VALUE routine
231 0359 1 sv_byte_value = $DISTINCT,
232 0360 1 sv_dev_header = $DISTINCT,
233 0361 1 sv_dev_noheader = $DISTINCT,
234 0362 1 sv_dev_italic = $DISTINCT,
235 0363 1 sv_dev_landscape = $DISTINCT,
236 0364 1 sv_dev_load = $DISTINCT,
237 0365 1 sv_dev_noload = $DISTINCT,
238 0366 1 sv_dev_portrait = $DISTINCT,
239 0367 1 sv_dev_underline = $DISTINCT,
240 0368 1 sv_di_dbg1 = $DISTINCT,
241 0369 1 sv_di_dbg2 = $DISTINCT,
242 0370 1 sv_output_line_numbers = $DISTINCT;
243 0371 1
244 0372 1 $LITERAL
245 0373 1 swch_id_automatic = $DISTINCT,
246 0374 1 swch_id_backspace = $DISTINCT,
247 0375 1 swch_id_bold = $DISTINCT,
248 0376 1 swch_id_change = $DISTINCT,
249 0377 1 swch_id_contents = $DISTINCT,
250 0378 1 swch_id_cross_reference = $DISTINCT,
251 0379 1 swch_id_dec_internal = $DISTINCT,
```

```

252 0380 1 swch_id_debug = $DISTINCT,
253 0381 1 swch_id_device = $DISTINCT,
254 0382 1 swch_id_down = $DISTINCT,
255 0383 1 swch_id_form_size = $DISTINCT,
256 0384 1 swch_id_index = $DISTINCT,
257 0385 1 swch_id_input = $DISTINCT,
258 0386 1 swch_id_intermediate = $DISTINCT,
259 0387 1 swch_id_log = $DISTINCT,
260 0388 1 swch_id_messages = $DISTINCT,
261 0389 1 swch_id_output = $DISTINCT,
262 0390 1 swch_id_pages = $DISTINCT,
263 0391 1 swch_id_pause = $DISTINCT,
264 0392 1 swch_id_right = $DISTINCT,
265 0393 1 swch_id_separate = $DISTINCT,
266 0394 1 swch_id_sequence = $DISTINCT,
267 0395 1 swch_id_simulate = $DISTINCT,
268 0396 1 swch_id_spacing = $DISTINCT,
269 0397 1 swch_id_underline = $DISTINCT,
270 0398 1 swch_id_variant = $DISTINCT;
271 0399 1
272 0400 1 !
273 0401 1 !: OWN STORAGE:
274 0402 1 !:
275 0403 1 OWN
276 0404 1
277 0405 1 runoff_command : $RNO_CMD, ! DSR command information block
278 0406 1 swch_id, ! Will contain value assigned to the swch_xxx
279 0407 1 ! names assigned previously. This is used
280 0408 1 ! within common routines to know which
281 0409 1 ! qualifier is being processed.
282 0410 1 ln01_option_used, ! True if an LN01 option is specified
283 0411 1 ws_limits : VECTOR[2] INITIAL (0,%X'7FFFFFFF');
284 0412 1
285 0413 1 !
286 0414 1 !: EXTERNAL REFERENCES:
287 0415 1 !:
288 0416 1 EXTERNAL ROUTINE
289 0417 1 RUNOFF, ! Transportable DSR entry point
290 0418 1
291 0419 1 SYSSPURGWS : ADDRESSING_MODE(GENERAL), ! Purge working set
292 0420 1 %IF NEW RNO DAT_MODULE %THEN
293 0421 1 SYSSNUMTIM,
294 0422 1 %FI
295 0423 1 CLISGET VALUE : ADDRESSING_MODE(GENERAL), ! Gets switch value
296 0424 1 CLISPRESENT : ADDRESSING_MODE(GENERAL), ! Checks for switch
297 0425 1 LIB$CVT_DTB : ADDRESSING_MODE(GENERAL), ! Converts decimal ASCII string to integer
298 0426 1 LIB$TPARSE : ADDRESSING_MODE(GENERAL), ! Table driven parser
299 0427 1 LIB$LP_LINES : ADDRESSING_MODE(GENERAL), ! Get default printer paper length
300 0428 1 LIB$FIND_FILE END : ! NEEDED to free memory used by LIB$FIND_FILE.
301 0429 1 WEAK ADDRESSING_MODE(GENERAL);
302 0430 1
303 0431 1 EXTERNAL LITERAL
304 U 0432 1 %IF DSRPLUS %THEN
305 U 0433 1 dsrplus$_facility; ! Our facility number
306 0434 1 %ELSE
307 0435 1 runoff$_facility; ! Our facility number
308 0436 1 %FI

```



```

: 309      0437 1
: 310      0438 : ! The following system literals are also used, but defined in STARLET:
: 311      0439 1
: 312      0440 1      ss$_badparam,      ! "bad parameter value"
: 313      0441 1      sts$k_severe;      ! code for SEVERE (or fatal) error status
: 314      0442 1
: 315      0443 1 EXTERNAL LITERAL      ! Value returned by CLISPRESENT
: 316      0444 1      CLIS_ABSENT,      ! Qualifier totally absent
: 317      0445 1      CLIS_PRESENT,      ! Present and asserted (on verb)
: 318      0446 1      CLIS_DEFAULTED,      ! absent, but defaulted in CLD
: 319      0447 1      CLIS_NEGATED,      ! present and negated (on verb)
: 320      0448 1
: 321      0449 1      ! Additional values for VMS V3.1 or later
: 322      0450 1
: 323      0451 1      CLIS_LOCPRES : WEAK ,      ! present and asserted on parameter
: 324      0452 1      CLIS_LOCNEG : WEAK ;      ! present and negated on parameter
: 325      0453 1
: 326      0454 1 EXTERNAL
: 327      0455 1      lffctx;      ! LIB$FIND_FILE context cell
: 328      0456 1

```

```

330 0457 1 %SBTTL 'TPARSE state tables'
331 0458 1
332 0459 1 Build a (MASK,ADDR) pair for TPARSE bit-mask table entries
333 0460 1
334 0461 1 MACRO
335 0462 1     xmask [i,p,s,e] = (1^((p)+(s))) - (1^(p)) %,
336 0463 1     mask [ ] = (0 OR xmask (%REMAINING)) %,
337 0464 1     tpmsk (o,p,s,e,base) = mask (o,p,s,e) (base) + %UPVAL*(o) %,
338 0465 1     tpmskf_(fld,base) = tpmsk_(%FIELDEXPAND(fld), base) %;
339 0466 1
340 0467 1
341 0468 1 Tables to parse /DEBUG
342 0469 1
343 0470 1 $init_state(debug_states,debug_keys);
344 P 0471 1 $state(
345 P 0472 1     ('ALL', tpa$_exit,,tpmsk_(%FIELDEXPAND(rno$V_deb_cond,0),
346 P 0473 1     %FIELDEXPAND(rno$V_deb_cond,1),
347 P 0474 1     6,0, runoff_command)),
348 P 0475 1     ('CONDITIONALS', tpa$_exit,,tpmskf_(rno$V_deb_cond, runoff_command)),
349 P 0476 1     ('CONTENTS', tpa$_exit,,tpmskf_(rno$V_deb_cont, runoff_command)),
350 U 0477 1 %IF DSRPLUS %THEN
351 U 0478 1     ('CROSS_REFERENCES',tpa$_exit,,tpmskf_(rno$V_deb_cros, runoff_command)),
352 P 0479 1 %FI
353 P 0480 1     ('FILES', tpa$_exit,,tpmskf_(rno$V_deb_files, runoff_command)),
354 P 0481 1     ('INDEX', tpa$_exit,,tpmskf_(rno$V_deb_index, runoff_command)),
355 P 0482 1     ('SAVE_RESTORE', tpa$_exit,,tpmskf_(rno$V_deb_save, runoff_command))
356 0483 1 );
357 0484 1
358 0485 1 Tables to parse /MESSAGES
359 0486 1
360 0487 1 $init_state(messages_states,messages_keys);
361 P 0488 1 $state(
362 P 0489 1     ('OUTPUT', tpa$_exit,,tpmskf_(rno$V_msg_out, runoff_command)),
363 P 0490 1     ('USER', tpa$_exit,,tpmskf_(rno$V_msg_user, runoff_command)),
364 P 0491 1     (tpa$_lambda,tpa$_exit,)
365 0492 1 );
366 0493 1
367 0494 1 Tables to parse /DEVICE
368 0495 1
369 0496 1 $init_state(device_states,device_keys);
370 P 0497 1 $state(dev_options,
371 U 0498 1 %IF dsrplus %THEN
372 U 0499 1     ('DIABLO', tpa$_exit,set_output_device_code,..,op_dev_diablo),
373 P 0500 1 %FI
374 P 0501 1     ('LINEPRINTER', tpa$_exit,set_output_device_code,..,op_dev_line_printer),
375 U 0502 1 %IF dsrplus %THEN
376 U 0503 1     ('VT100', tpa$_exit,set_output_device_code,..,op_dev_vt100),
377 U 0504 1     ('FLIP', tpa$_exit,set_output_device_code,..,op_dev_flip),
378 P 0505 1 %FI
379 P 0506 1     ('LN01', tpa$_exit,set_output_device_code,..,op_dev_ln01),
380 P 0507 1     ('LN01E', tpa$_exit,set_output_device_code,..,op_dev_ln01e),
381 P 0508 1     ('UNDERLINE', tpa$_exit,save_value,..,sv_dev_underline),
382 P 0509 1     ('ITALIC', tpa$_exit,save_value,..,sv_dev_italic),
383 P 0510 1     ('PORTRAIT', tpa$_exit,save_value,..,sv_dev_portrait),
384 P 0511 1     ('LANDSCAPE', tpa$_exit,save_value,..,sv_dev_landscape),
385 P 0512 1     ('HEADER', tpa$_exit,save_value,..,sv_dev_header),
386 P 0513 1     ('NOHEADER', tpa$_exit,save_value,..,sv_dev_noheader),

```

```

: 387 P 0514 1 ('LOAD', tpa$_exit,save_value,,,sv_dev_load),
388 PP 0515 1 ('NOLOAD', tpa$_exit,save_value,,,sv_dev_noload),
389 P 0516 1 (tpa$_lambda, tpa$_fail,)
390 0517 1 );
391 0518 1 ;
392 0519 1 ; Tables to parse /DEC_INTERNAL
393 0520 1 ;
394 0521 1 $init_state(dec_internal_states,dec_internal_keys);
395 PP 0522 1 $state(options,
396 PP 0523 1 (options,
397 PP 0524 1 ('DBG1', dbg1,,,sv_di_dbg1),
398 PP 0525 1 ('DBG2', dbg1,,,sv_di_dbg2),
399 PP 0526 1 ('FLIP', options,set_output_device_code,,,op_dev_flip),
400 PP 0527 1 ('VT100', options,set_output_device_code,,,op_dev_vt100),
401 P 0528 1 ('OUTPUT_LINE_NUMBER', options,save_value,,,sv_output_line_numbers),
402 P 0529 1 (tpa$_lambda,tpa$_exit,)
403 0530 1 );
404 0531 1 ;
405 P 0532 1 $state(
406 PP 0533 1 ('%f' which_base),
407 P 0534 1 (tpa$_lambda,tpa$_exit,)
408 0535 1 );
409 0536 1 ;
410 P 0537 1 $state(dbg1,
411 PP 0538 1 ('='),
412 PP 0539 1 (':'),
413 P 0540 1 ('.', options)
414 0541 1 );
415 P 0542 1 $state(
416 PP 0543 1 ('%f' which_base),
417 PP 0544 1 (tpa$_hex, options,save_value),
418 P 0545 1 (tpa$_lambda,options)
419 0546 1 );
420 P 0547 1 $state(which_base,
421 PP 0548 1 ('X', hex_num),
422 PP 0549 1 ('O', oct_num),
423 PP 0550 1 ('D', dec_num),
424 P 0551 1 (tpa$_hex, options,save_value),
425 P 0552 1 (tpa$_lambda,options)
426 0553 1 );
427 P 0554 1 $state(hex_num,
428 0555 1 (tpa$_hex,options,save_value));
429 P 0556 1 $state(oct_num,
430 0557 1 (tpa$_octal,options,save_value));
431 P 0558 1 $state(dec_num,
432 0559 1 (tpa$_decimal,options,save_value));
433 0560 1 ;
434 0561 1 ;
435 0562 1 ; Tables to parse /CHANGE BARS, /UNDERLINE_CHARACTER,
436 0563 1 /NONSPACING_UNDERLINE, & /SEPARATE_UNDERLINE.
437 0564 1 ;
438 0565 1 ;
439 0566 1 ; For that matter, with the appropriate changes made to routine
440 0567 1 SAVE_VALUE, these tables will handle any qualifier that expects to
441 0568 1 be given a single character or the numeric value representing that
442 0569 1 character.
443 0570 1 $init_state(chars_states,chars_keys);

```

```
444 P 0571 1 $state(  
445 P 0572 1 ('%f' chars_which_base,...,sv_byte_value),  
446 P 0573 1 (tpa$_eos, tpa$_exit,);  
447 P 0574 1 (tpa$_lambda, tpa$_fail,);  
448 P 0575 1 );  
449 P 0576 1 $state(chars_which_base,  
450 P 0577 1 ('X' chars_hex_num),  
451 P 0578 1 ('O' chars_oct_num),  
452 P 0579 1 ('D' chars_dec_num),  
453 P 0580 1 (tpa$_decimal, chars_done,save_value),  
454 P 0581 1 (tpa$_hex, chars_done,save_value),  
455 P 0582 1 (tpa$_lambda, tpa$_fail,);  
456 P 0583 1 );  
457 P 0584 1 $state(chars_hex_num,  
458 P 0585 1 (tpa$_hex, chars_done,save_value));  
459 P 0586 1 $state(chars_oct_num,  
460 P 0587 1 (tpa$_octal, chars_done,save_value));  
461 P 0588 1 $state(chars_dec_num,  
462 P 0589 1 (tpa$_decimal, chars_done,save_value));  
463 P 0590 1 $state(chars_done,  
464 P 0591 1 (tpa$_eos, tpa$_exit,);  
465 P 0592 1 (tpa$_lambda, tpa$_fail,);  
466 P 0593 1 );
```

```

: 468 0594 1 %SBTTL 'RNOCLI -- Header and local storage'
: 469 0595 1 GLOBAL ROUTINE RNOCLI =
: 470 0596 1 ++
: 471 0597 1
: 472 0598 1 FUNCTIONAL DESCRIPTION:
: 473 0599 1
: 474 0600 1 This routine uses the VMS DCL CLE routines to obtain command
: 475 0601 1 line information which is in turn passed to the DSR application
: 476 0602 1 in a transportable manner.
: 477 0603 1
: 478 0604 1 FORMAL PARAMETERS: None
: 479 0605 1
: 480 0606 1 IMPLICIT INPUTS: None
: 481 0607 1
: 482 0608 1 IMPLICIT OUTPUTS: None
: 483 0609 1
: 484 0610 1 COMPLETION CODES:
: 485 0611 1
: 486 0612 1 completion code from DSR
: 487 0613 1
: 488 0614 1 SIDE EFFECTS: None
: 489 0615 1 --
: 490 0616 1
: 491 0617 2 BEGIN
: 492 0618 2
: 493 0619 2 LOCAL
: 494 0620 2 desc : ! Descriptor work area for translating #'s
: 495 0621 2 $STR_DESCRIPTOR(CLASS=DYNAMIC),
: 496 0622 2 flag_state, ! Presence/absence implicit/explicit of /FLAG
: 497 0623 2 status, ! Temporary routine completion code
: 498 0624 2 exit_status, ! Final status for return to VMS
: 499 0625 2 form_qualifier, ! Used to save the status of /FORM
: 500 0626 2 temp; ! Real temporary storage!

```

```

502 0627 2 %SBTTL 'RNOCLI -- Initialization'
503 0628 2
504 0629 2 Misc. internal initialization
505 0630 2
506 0631 2 Initialize the DSR command block.
507 0632 2
508 0633 2 $STR_DESC_INIT( DESCRIPTOR = desc, CLASS = DYNAMIC);
509 0634 2
510 0635 2 exit_status = %X'10000001';           ! Initialize to normal return status.
511 0636 2                                     ! Bit # 28 is also set to prevent VMS
512 0637 2                                     ! from outputting any exit message on
513 0638 2                                     ! our behalf.
514 0639 2
515 0640 2 lffctx = 0;           ! Initialize the LIB$FIND_FILE context cell
516 0641 2
517 0642 2 WHILE          CLISGET_VALUE(swch_input,desc)
518 0643 2 DO
519 0644 2   BEGIN
520 0645 2
521 0646 2     BEGIN
522 0647 2     MAP runoff_command : VECTOR;
523 0648 2
524 0649 2     INCR index FROM 0 TO RNOSC_CMD_LEN / %UPVAL - 1 DO
525 0650 2     runoff_command[index] = false;
526 0651 2     END;
527 0652 2
528 0653 2     runoff_command[RNOSC_UNDERLIN_] = %X'FF';
529 0654 2
530 0655 2
531 0656 2 Complete the DSR command block (RNOCMD) from the information returned by
532 0657 2 the DCL CLI.
533 0658 2
534 0659 2 NOTE : Some RNOCMD fields are set by qualifier action routines.
535 0660 2
536 0661 2
537 P 0662 2 $STR_DESC_INIT( DESCRIPTOR = runoff_command[RNOST_INPUT],
538 0663 2 CLASS = DYNAMIC);
539 0664 2
540 0665 2 $STR_COPY(STRING = desc, TARGET = runoff_command[RNOST_INPUT] );
541 0666 2
542 P 0667 2 $STR_DESC_INIT( DESCRIPTOR = runoff_command[RNOST_OUTPUT],
543 0668 2 CLASS = DYNAMIC);
544 0669 2
545 P 0670 2 $STR_DESC_INIT( DESCRIPTOR = runoff_command[RNOST_INTERMEDIATE],
546 0671 2 CLASS = DYNAMIC);
547 0672 2
548 P 0673 2 $STR_DESC_INIT( DESCRIPTOR = runoff_command[RNOST_PAGES],
549 0674 2 CLASS = DYNAMIC);
550 0675 2
551 P 0676 2 $STR_DESC_INIT( DESCRIPTOR = runoff_command[RNOST_VARIANT],
552 0677 2 CLASS = DYNAMIC);
553 0678 2
554 P 0679 2 $STR_DESC_INIT( DESCRIPTOR = runoff_command[rno$t_drop_rec],
555 0680 2 CLASS = DYNAMIC);
556 0681 2
557 P 0682 2 $STR_DESC_INIT( DESCRIPTOR = runoff_command[rno$t_program_name],
558 0683 2 CLASS = DYNAMIC);

```



```
564 0688 3 %SBTTL 'RNOCLI -- /AUTO /CROSS /BACK /PAUSE /SEQ /SEQ /SIM /LOG'
565 0689
566 0690 Pickup switch settings and setup command table accordingly
567 0691
568 0692 Process switches without values: /CROSSREF /AUTOMATIC
569 0693 /BACKSPACE /PAUSE /SEQUENCE
570 0694 /SIMULATE /LOG
571 0695
572 U 0696 %IF DSRPLUS %THEN
573 U 0697
574 U 0698 temp = cli$present(swch_cross_reference);
575 U 0699
576 U 0700 SELECT .temp OF
577 U 0701 SET
578 U 0702
579 U 0703 [CLIS_NEGATED, CLIS_PRESENT, CLIS_LOCPRES, CLIS_LOCNEG] :
580 U 0704 runoff_command[rno$v_s_cross_reference] = true;
581 U 0705
582 U 0706 [CLIS_ABSENT, CLIS_NEGATED, CLIS_LOCNEG] :
583 U 0707 runoff_command[rno$v_cross_reference] = false;
584 U 0708
585 U 0709 [CLIS_DEFAULTED, CLIS_PRESENT, CLIS_LOCPRES] :
586 U 0710 runoff_command[rno$v_cross_reference] = true;
587 U 0711
588 U 0712 TES;
589 U 0713
590 U 0714 temp = cli$present(swch_automatic);
591 U 0715
592 U 0716 SELECT .temp OF
593 U 0717 SET
594 U 0718
595 U 0719 [CLIS_NEGATED, CLIS_PRESENT, CLIS_LOCPRES, CLIS_LOCNEG] :
596 U 0720 runoff_command[rno$v_s_automatic] = true;
597 U 0721
598 U 0722 [CLIS_ABSENT, CLIS_NEGATED, CLIS_LOCNEG] :
599 U 0723 runoff_command[rno$v_automatic] = false;
600 U 0724
601 U 0725 [CLIS_DEFAULTED, CLIS_PRESENT, CLIS_LOCPRES] :
602 U 0726 runoff_command[rno$v_automatic] = true;
603 U 0727
604 U 0728 TES;
605 U 0729 %FI
606 0730 temp = cli$present(swch_backspace);
607 0731
608 0732 SELECT .temp OF
609 0733 SET
610 0734
611 0735 [CLIS_ABSENT, CLIS_DEFAULTED] :
612 0736 runoff_command[rno$v_s_backspace] = false;
613 0737
614 0738 [CLIS_NEGATED, CLIS_PRESENT, CLIS_LOCPRES, CLIS_LOCNEG] :
615 0739 runoff_command[rno$v_s_backspace] = true;
616 0740
617 0741 [CLIS_ABSENT, CLIS_NEGATED, CLIS_LOCNEG] :
618 0742 runoff_command[rno$v_backspace] = false;
619 0743
620 0744 [CLIS_DEFAULTED, CLIS_PRESENT, CLIS_LOCPRES] :
```


621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677

0745
0746
0747
0748
0749
0750
0751
0752
0753
0754
0755
0756
0757
0758
0759
0760
0761
0762
0763
0764
0765
0766
0767
0768
0769
0770
0771
0772
0773
0774
0775
0776
0777
0778
0779
0780
0781
0782
0783
0784
0785
0786
0787
0788
0789
0790
0791
0792
0793
0794
0795
0796
0797
0798
0799
0800
0801

```
runoff_command[rno$v_backspace] = true;
TES;
temp = cli$present(swch_log);
SELECT .temp OF
SET
[CLIS_ABSENT, CLIS_DEFAULTED] :
runoff_command[rno$v_s_log] = false;
[CLIS_NEGATED, CLIS_PRESENT, CLIS_LOCPRES, CLIS_LOCNEG] :
runoff_command[rno$v_s_log] = true;
[CLIS_ABSENT, CLIS_NEGATED, CLIS_LOCNEG] :
runoff_command[rno$v_log] = false;
[CLIS_DEFAULTED, CLIS_PRESENT, CLIS_LOCPRES] :
runoff_command[rno$v_log] = true;
TES;
temp = cli$present(swch_pause);
SELECT .temp OF
SET
[CLIS_ABSENT, CLIS_DEFAULTED] :
runoff_command[rno$v_s_pause] = false;
[CLIS_NEGATED, CLIS_PRESENT, CLIS_LOCPRES, CLIS_LOCNEG] :
runoff_command[rno$v_s_pause] = true;
[CLIS_ABSENT, CLIS_NEGATED, CLIS_LOCNEG] :
runoff_command[rno$v_pause] = false;
[CLIS_DEFAULTED, CLIS_PRESENT, CLIS_LOCPRES] :
runoff_command[rno$v_pause] = true;
TES;
temp = cli$present(swch_sequence);
SELECT .temp OF
SET
[CLIS_ABSENT, CLIS_DEFAULTED] :
runoff_command[rno$v_s_sequence] = false;
[CLIS_NEGATED, CLIS_PRESENT, CLIS_LOCPRES, CLIS_LOCNEG] :
runoff_command[rno$v_s_sequence] = true;
[CLIS_ABSENT, CLIS_NEGATED, CLIS_LOCNEG] :
runoff_command[rno$v_sequence] = false;
[CLIS_DEFAULTED, CLIS_PRESENT, CLIS_LOCPRES] :
```

.. 678
... 679
... 680
... 681
... 682
... 683
... 684
... 685
... 686
... 687
... 688
... 689
... 690
... 691
... 692
... 693
... 694
... 695
... 696
... 697
... 698
.. 699

```
0802          runoff_command[rno$V_sequence] = true;
0803
0804      TES;
0805
0806      temp = cli$present(swch_simulate);
0807
0808      SELECT .temp OF
0809      SET
0810
0811      [CLIS_ABSENT, CLIS_DEFAULTED] :
0812          runoff_command[rno$V_s_simulate] = false;
0813
0814      [CLIS_NEGATED, CLIS_PRESENT, CLIS_LOCPRES, CLIS_LOCNEG] :
0815          runoff_command[rno$V_s_simulate] = true;
0816
0817      [CLIS_ABSENT, CLIS_NEGATED, CLIS_LOCNEG] :
0818          runoff_command[rno$V_simulate] = false;
0819
0820      [CLIS_DEFAULTED, CLIS_PRESENT, CLIS_LOCPRES] :
0821          runoff_command[rno$V_simulate] = true;
0822
0823      TES;
```

```
0824 3 %SBTTL 'RNOCLI -- /OUTPUT /INTERMEDIATE'
0825
0826 Process switches specifying files: /OUTPUT /INTERMEDIATE
0827 The input file(s) [currently identified as INFIL] will be processed
0828 last.
0829
0830 temp = cli$present(swch_output);
0831
0832 ! First, initialize all flags affected by this switch:
0833
0834 runoff_command[rno$v_s_output] = false;
0835 runoff_command[rno$v_output] = false;
0836 runoff_command[rno$v_quick] = false;
0837
0838 SELECT .temp OF
0839 SET
0840
0841 [CLIS_NEGATED, CLIS_PRESENT, CLIS_LOCPRES, CLIS_LOCNEG] :
0842 runoff_command[rno$v_s_output] = true;
0843
0844 [CLIS_NEGATED, CLIS_LOCNEG] :
0845 runoff_command[rno$v_quick] = true;
0846
0847 [CLIS_PRESENT, CLIS_LOCPRES
0848 %IF NOT DSRPLUS %THEN , CLIS_DEFAULTED %FI ] :
0849 runoff_command[rno$v_output] = true;
0850
0851 [CLIS_DEFAULTED, CLIS_PRESENT, CLIS_LOCPRES] :
0852 c[cli$get_value(swch_output,runoff_command[rno$t_output]);
0853
0854 TES;
0855
0856 temp = cli$present(swch_intermediate);
0857
0858 SELECT .temp OF
0859 SET
0860
0861 [CLIS_NEGATED, CLIS_PRESENT, CLIS_LOCPRES, CLIS_LOCNEG] :
0862 runoff_command[rno$v_s_intermediate] = true;
0863
0864 [CLIS_DEFAULTED, CLIS_PRESENT, CLIS_LOCPRES] :
0865 runoff_command[rno$v_intermediate] = true;
0866
0867 [CLIS_DEFAULTED, CLIS_NEGATED, CLIS_PRESENT, CLIS_LOCPRES, CLIS_LOCNEG] :
0868 c[cli$get_value(swch_intermediate,runoff_command[rno$t_intermediate]);
0869
0870 TES;
```

```

: 749 0871 3 %SBTTL 'RNOCLI -- /CHANGE /UNDER /SEPARATE /NONSPACING'
: 750 0872
: 751 0873 Process switches passing ASCII characters: /CHANGE BARS
: 752 0874 /UNDERLINE_CHARACTER /SEPARATE_UNDERLINE /NONSPACING_UNDERLINE
: 753 0875
: 754 0876
: 755 0877 temp = cli$present(swch_change);
: 756 0878
: 757 0879 SELECT .temp OF
: 758 0880 SET
: 759 0881
: 760 0882 [CLIS_ABSENT, CLIS_DEFAULTED] :
: 761 0883 runoff_command[rno$v_s_change] = false;
: 762 0884
: 763 0885 [CLIS_NEGATED, CLIS_PRESENT, CLIS_LOCPRES, CLIS_LOCNEG] :
: 764 0886 runoff_command[rno$v_s_change] = true;
: 765 0887
: 766 0888 [CLIS_ABSENT] :
: 767 0889 runoff_command[rno$c_change] = %X'FF';
: 768 0890
: 769 0891 [CLIS_NEGATED, CLIS_LOCNEG] :
: 770 0892 BEGIN
: 771 0893 runoff_command[rno$c_change] = 0;
: 772 0894 runoff_command[rno$v_change] = false;
: 773 0895 END;
: 774 0896
: 775 0897 [CLIS_DEFAULTED, CLIS_PRESENT, CLIS_LOCPRES] :
: 776 0898 BEGIN
: 777 0899 runoff_command[rno$v_change] = true;
: 778 0900 swch_id = swch_id_change;
: 779 0901 IF cli$get_value(swch_change, desc) THEN
: 780 0902 IF .desc[str$h_length] EQL 1 THEN
: 781 0903 runoff_command[rno$c_change] = CHRCHAR(.desc[str$a_pointer])
: 782 0904 ELSE
: 783 0905
: 784 0906 ( IF NOT (status = call_tparse( desc
: 785 0907 ,chars_states
: 786 0908 ,chars_keys) )
: 787 U 0909 THEN RETURN (%IF DSRPLUS %THEN dsrplus$ facility
: 788 0910 %ELSE runoff$facility %FI ^16
: 789 0911 OR ss$ badparam
: 790 0912 OR sts$k_severe) )
: 791 0913
: 792 0914 ELSE
: 793 0915 runoff_command[rno$c_change] = %C'!';
: 794 0916 END;
: 795 0917
: 796 0918 [CLIS_DEFAULTED, CLIS_NEGATED, CLIS_PRESENT, CLIS_LOCPRES, CLIS_LOCNEG] :
: 797 0919 BEGIN
: 798 0920 runoff_command[rno$v_chng_char] = true;
: 799 0921 IF .runoff_command[rno$c_change] EQL 0 THEN
: 800 0922 runoff_command[rno$v_change] = false;
: 801 0923 END;
: 802 0924
: 803 0925 TES;
: 804 0926
: 805 0927 temp = cli$present(swch_underline);

```

```

: 806 0928 3
: 807 0929 3
: 808 0930 3
: 809 0931 3
: 810 0932 3
: 811 0933 3
: 812 0934 3
: 813 0935 3
: 814 0936 3
: 815 0937 3
: 816 0938 3
: 817 0939 3
: 818 0940 3
: 819 0941 3
: 820 0942 4
: 821 0943 4
: 822 0944 4
: 823 0945 4
: 824 0946 4
: 825 0947 4
: 826 0948 4
: 827 0949 5
: 828 0950 5
: 829 0951 6
: 830 0952 6
: 831 0953 6
: 832 U 0954 6
: 833 0955 6
: 834 0956 6
: 835 0957 6
: 836 0958 5
: 837 0959 4
: 838 0960 4
: 839 0961 4
: 840 0962 3
: 841 0963 3
: 842 0964 3
: 843 0965 3
: 844 0966 3
: 845 0967 3
: 846 0968 3
: 847 0969 3
: 848 0970 3
: 849 0971 3
: 850 0972 3
: 851 0973 3
: 852 0974 3
: 853 0975 3
: 854 0976 3
: 855 0977 3
: 856 0978 3
: 857 0979 3
: 858 0980 3
: 859 0981 4
: 860 0982 4
: 861 0983 4
: 862 0984 4

! First, initialize all flags affected by this switch:
runoff_command[rno$v_s_underline] = false;
runoff_command[rno$v_underline] = false;
runoff_command[rno$v_und_char] = false;

SELECT .temp OF
SET
[CLIS_NEGATED, CLIS_PRESENT, CLIS_LOCPRES, CLIS_LOCNEG] :
runoff_command[rno$v_s_underline] = true;

[CLIS_DEFAULTED, CLIS_PRESENT, CLIS_LOCPRES] :
BEGIN
runoff_command[rno$v_underline] = true;
runoff_command[rno$v_und_char] = true;
IF cli$get_value(swch_underline, desc) THEN
IF .desc[st$length] EQL 1 THEN
runoff_command[rno$c_underline] = CHRCHAR(.desc[st$a_pointer])
ELSE
BEGIN
swch_id = swch_id_underline;
IF NOT (status = call_tparse( desc
,chars_states
,chars_keys) )
THEN RETURN (%IF DSRPLUS %THEN dsrplus$ facility
%ELSE runoff$_facility %FI ^16
OR ss$ badparam
OR sts$k_severe)
END
ELSE
runoff_command[rno$c_underline] = %C'_';

END;

TES;

temp = cli$present(swch_separate);

! First, initialize all flags affected by this switch except for the
! underline character flag:
runoff_command[rno$v_s_und_separ] = false;
runoff_command[rno$v_und_separ] = false;

SELECT .temp OF
SET
[CLIS_NEGATED, CLIS_PRESENT, CLIS_LOCPRES, CLIS_LOCNEG] :
runoff_command[rno$v_s_und_separ] = true;

[CLIS_DEFAULTED, CLIS_PRESENT, CLIS_LOCPRES] :
BEGIN
runoff_command[rno$v_und_separ] = true;
runoff_command[rno$v_underline] = true;
runoff_command[rno$v_und_char] = true;

```

```

863      0985      4      IF cli$get_value(swch_separate,desc) THEN
864      0986      4      IF .desc[stresh_length] EQL 1 THEN
865      0987      4      runoff_command[rno$c_underline] = CHRCHAR(.desc[str$a_pointer])
866      0988      4      ELSE
867      0989      5      BEGIN
868      0990      5      swch_id = swch_id_separate;
869      0991      6      IF NOT (status = call_tparse( desc
870      0992      6      ,chars_states
871      0993      6      ,chars_keys) )
872      U 0994      6      THEN RETURN (%IF DSRPLUS %THEN dsrplus$ facility
873      0995      6      %ELSE runoff$_facility %FI ^16
874      0996      6      OR ss$ badparam
875      0997      6      OR sts$k_severe)
876      0998      5      END
877      0999      4      ELSE
878      1000      4      runoff_command[rno$c_underline] = %C'-';
879      1001      4      END;
880      1002      3
881      1003      3      TES;
882      1004      3      temp = cli$present(swch_spacing);
883      1005      3      First, initialize all flags affected by this switch except for the
884      1006      3      underline character flag:
885      1007      3      !!
886      1008      3      !!
887      1009      3
888      1010      3      runoff_command[rno$v_s_und_nonsp] = false;
889      1011      3      runoff_command[rno$v_und_nonsp] = false;
890      1012      3
891      1013      3      SELECT .temp OF
892      1014      3      SET
893      1015      3
894      1016      3      [CLIS_NEGATED, CLIS_PRESENT, CLIS_LOCPRES, CLIS_LOCNEG] :
895      1017      3      runoff_command[rno$v_s_und_nonsp] = true;
896      1018      3
897      1019      3      [CLIS_DEFAULTED, CLIS_PRESENT, CLIS_LOCPRES] :
898      1020      4      BEGIN
899      1021      4      runoff_command[rno$v_und_nonsp] = true;
900      1022      4      runoff_command[rno$v_underline] = true;
901      1023      4      runoff_command[rno$v_und_char] = true;
902      1024      4      IF cli$get_value(swch_spacing,desc) THEN
903      1025      4      IF .desc[stresh_length] EQL 1 THEN
904      1026      4      runoff_command[rno$c_underline] = CHRCHAR(.desc[str$a_pointer])
905      1027      4      ELSE
906      1028      5      BEGIN
907      1029      5      swch_id = swch_id_underline;
908      1030      6      IF NOT (status = call_tparse( desc
909      1031      6      ,chars_states
910      1032      6      ,chars_keys) )
911      U 1033      6      THEN RETURN (%IF DSRPLUS %THEN dsrplus$ facility
912      1034      6      %ELSE runoff$_facility %FI ^16
913      1035      6      OR ss$ badparam
914      1036      6      OR sts$k_severe)
915      1037      5      END
916      1038      4      ELSE
917      1039      4      runoff_command[rno$c_underline] = 7;
918      1040      3      END;
919      1041      3

```

```
: 920      1042      3      TES;  
: 921      1043      3  
: 922      1044      3      IF .runoff_command[RNO$C_UNDERLINE] EQLU %X'FF'  
: 923      1045      3      THEN  
: 924      1046      3      runoff_command[RNO$C_UNDERLINE] = no  
: 925      1047      3      ELSE  
: 926      1048      3      runoff_command[RNO$V_UND_CHAR] = yes;
```

```

: 928      1049      3 %SBTTL 'RNOCLI -- /PAGES /VARIANT'
: 929      1050
: 930      1051      Process switches passing ASCII strings parsed by DSP:
: 931      1052
: 932      1053      /PAGES /VARIANT
: 933      1054
: 934      1055      temp = cli$present(swch_pages);
: 935      1056
: 936      1057      SELECT .temp OF
: 937      1058      SET
: 938      1059
: 939      1060      [CLIS_ABSENT, CLIS_NEGATED, CLIS_LOCNEG] : false;
: 940      1061      [CLIS_DEFAULTED, CLIS_PRESENT, CLIS_LOCPRES] :
: 941      1062      cli$get_value(swch_pages,runoff_command[rno$t_pages]);
: 942      1063
: 943      1064      TES;
: 944      1065
: 945      1066      temp = cli$present(swch_variant);
: 946      1067
: 947      1068      SELECT .temp OF
: 948      1069      SET
: 949      1070
: 950      1071      [CLIS_ABSENT, CLIS_NEGATED, CLIS_LOCNEG] : false;
: 951      1072      [CLIS_DEFAULTED, CLIS_PRESENT, CLIS_LOCPRES] :
: 952      1073      cli$get_value(swch_variant,runoff_command[rno$t_variant]);
: 953      1074
: 954      1075      TES;
```



```

: 956 1076 3 %SBTTL 'RNOCLI -- /BOLD /DOWN /RIGHT'
: 957 1077 3
: 958 1078 3
: 959 1079 3 Process switches passing numeric arguments: /BOLD /DOWN /RIGHT
: 960 1080 3
: 961 1081 3 temp = cli$present(swch_bold);
: 962 1082 3
: 963 1083 3 SELECT .temp OF
: 964 1084 3 SET
: 965 1085 3 [CLIS_ABSENT] :
: 966 1086 3 runoff_command[rno$h_bold] = 1;
: 967 1087 3
: 968 1088 3 [CLIS_NEGATED, CLIS_LOCNEG] :
: 969 1089 3 runoff_command[rno$h_bold] = 0;
: 970 1090 3
: 971 1091 3 [CLIS_DEFAULTED, CLIS_PRESENT, CLIS_LOCPRES] :
: 972 1092 3 IF cli$get_value(swch_bold,desc) THEN
: 973 1093 3 BEGIN
: 974 1094 3 status = lib$cvt_dtb(.desc[st$r$length],
: 975 1095 3 .desc[st$a_pointer],
: 976 1096 3 temp);
: 977 1097 3 runoff_command[rno$h_bold] = .temp
: 978 1098 3 END
: 979 1099 3 ELSE
: 980 1100 3 runoff_command[rno$h_bold] = 1;
: 981 1101 3
: 982 1102 3 TES:
: 983 1103 3
: 984 1104 3 temp = cli$present(swch_down);
: 985 1105 3
: 986 1106 3 SELECT .temp OF
: 987 1107 3 SET
: 988 1108 3 [CLIS_ABSENT, CLIS_NEGATED, CLIS_LOCNEG] :
: 989 1109 3 runoff_command[rno$h_down] = 0;
: 990 1110 3
: 991 1111 3 [CLIS_DEFAULTED, CLIS_PRESENT, CLIS_LOCPRES] :
: 992 1112 3 IF cli$get_value(swch_down,desc) THEN
: 993 1113 3 BEGIN
: 994 1114 3 status = lib$cvt_dtb(.desc[st$r$length],
: 995 1115 3 .desc[st$a_pointer],
: 996 1116 3 temp);
: 997 1117 3 runoff_command[rno$h_down] = .temp
: 998 1118 3 END
: 999 1119 3 ELSE
: 1000 1120 3 runoff_command[rno$h_down] = 5;
: 1001 1121 3
: 1002 1122 3 [CLIS_NEGATED, CLIS_LOCNEG, CLIS_PRESENT, CLIS_LOCPRES] :
: 1003 1123 3 runoff_command[rno$`v_s_down] = true;
: 1004 1124 3
: 1005 1125 3 TES:
: 1006 1126 3
: 1007 1127 3
: 1008 1128 3 form_qualifier = cli$present(swch_form_size);
: 1009 1129 3
: 1010 1130 3 SELECT .form_qualifier OF
: 1011 1131 3 SET
: 1012 1132 3

```

```

: 1013      1133      3
: 1014      1134      3
: 1015      1135      3
: 1016      1136      3
: 1017      1137      3
: 1018      1138      3
: 1019      1139      3
: 1020      1140      3
: 1021      1141      4
: 1022      1142      4
: 1023      1143      4
: 1024      1144      4
: 1025      1145      4
: 1026      1146      4
: 1027      1147      3
: 1028      1148      4
: 1029      1149      4
: 1030      1150      4
: 1031      1151      4
: 1032      1152      3
: 1033      1153      3
: 1034      1154      3
: 1035      1155      3
: 1036      1156      3
: 1037      1157      3
: 1038      1158      3
: 1039      1159      3
: 1040      1160      3
: 1041      1161      3
: 1042      1162      3
: 1043      1163      4
: 1044      1164      4
: 1045      1165      4
: 1046      1166      4
: 1047      1167      4
: 1048      1168      4
: 1049      1169      4
: 1050      1170      3
: 1051      1171      3
: 1052      1172      3
: 1053      1173      3
: 1054      1174      3
: 1055      1175      3
: 1056      1176      3
: 1057      1177      3

[CLIS_ABSENT] :
runoff_command[rno$h_form_size] = lib$lp_lines();

[CLIS_NEGATED, CLIS_LOCNEG] :
runoff_command[rno$h_form_size] = 0;

[CLIS_DEFAULTED, CLIS_PRESENT, CLIS_LOCPRES] :
IF cli$get_value(swch_form_size,desc) THEN
BEGIN
status = lib$cvt_dtb(.desc[st$r$length],
                    .desc[st$a$pointer],
                    temp);
runoff_command[rno$h_form_size] = .temp
END
ELSE
BEGIN
runoff_command[rno$h_form_size] = lib$lp_lines();
form_qualifier = CLIS_ABSENT ! so /DEV works right.
END;

TES;

temp = cli$present(swch_right);

SELECT .temp OF
SET

[CLIS_ABSENT, CLIS_NEGATED, CLIS_LOCNEG] :
runoff_command[rno$h_right] = 0;

[CLIS_DEFAULTED, CLIS_PRESENT, CLIS_LOCPRES] :
IF cli$get_value(swch_right,desc) THEN
BEGIN
status = lib$cvt_dtb(.desc[st$r$length],
                    .desc[st$a$pointer],
                    temp);
runoff_command[rno$h_right] = .temp
END
ELSE
runoff_command[rno$h_right] = 5;

[CLIS_NEGATED, CLIS_LOCNEG, CLIS_PRESENT, CLIS_LOCPRES] :
runoff_command[rno$v_s_right] = true;

TES;

```

```

: 1059      1178      3 %SBTTL 'RNOCLI -- /DEBUG /DEVICE /MESSAGES /DEC_INTERNAL'
: 1060      1179
: 1061      1180      Process switches passing lists of arguments: /DEBUG /DEVICE /MESSAGES
: 1062      1181      and /DEC_INTERNAL
: 1063      1182
: 1064      1183      temp = cli$present(swch_debug);
: 1065      1184
: 1066      1185      ! First initialize all bits that can be set by this qualifier:
: 1067      1186
: 1068      1187      runoff_command[rno$V_deb_cond] = false;
: 1069      1188      runoff_command[rno$V_deb_cont] = false;
: 1070      1189      runoff_command[rno$V_deb_files] = false;
: 1071      1190      runoff_command[rno$V_deb_index] = false;
: 1072      1191      runoff_command[rno$V_deb_cros] = false;
: 1073      1192      runoff_command[rno$V_deb_save] = false;
: 1074      1193
: 1075      1194      SELECT .temp OF
: 1076      1195      SET
: 1077      1196
: 1078      1197      [CLIS_DEFAULTED, CLIS_PRESENT, CLIS_LOCPRES] :
: 1079      1198      WHILE cli$get_value(swch_debug,desc)
: 1080      1199      DO call_tparse(desc,debug_states,debug_keys);
: 1081      1200
: 1082      1201      TES;
: 1083      1202
: 1084      1203      temp = cli$present(swch_device);
: 1085      1204
: 1086      1205      ! First, initialize all items that can be set by this qualifier:
: 1087      1206
: 1088      1207      runoff_command[rno$V_4_out_format] = op_dev_line_printer;
: 1089      1208      runoff_command[rno$V_ln01_port_land] = true;      ! 1 ==> portrait
: 1090      1209      runoff_command[rno$V_ln01_ital_under] = true;      ! 1 ==> italics
: 1091      1210      runoff_command[rno$V_ln01_header] = true;      ! 1 ==> O/P header info.
: 1092      1211      runoff_command[rno$V_ln01_load] = true;      ! 1 ==> O/P font load OCS
: 1093      1212
: 1094      1213      SELECT .temp OF
: 1095      1214      SET
: 1096      1215
: 1097      1216      [CLIS_DEFAULTED, CLIS_PRESENT, CLIS_LOCPRES] :
: 1098      1217      WHILE cli$get_value(swch_device,desc)
: 1099      1218      DO
: 1100      1219      IF NOT (status = call_tparse( desc
: 1101      1220      ,device_states
: 1102      1221      ,device_keys) )
: 1103      1222      THEN RETURN (%IF DSRPLUS %THEN dsrplus$ facility
: 1104      1223      %ELSE runoff$_facility %FI ^16
: 1105      1224      OR ss$ badparam
: 1106      1225      OR sts$k_severe);
: 1107      1226
: 1108      1227      TES;
: 1109      1228      %C
: 1110      1229      IF .form_qualifier EQL CLIS_ABSENT THEN ! If we used LIB$LP_LINES
: 1111      1230      IF .runoff_command[rno$V_4_out_format] EQL op_dev_ln01
: 1112      1231      OR
: 1113      1232      .runoff_command[rno$V_4_out_format] EQL op_dev_ln01e
: 1114      1233      THEN
: 1115      1234      runoff_command[rno$h_form_size] = 255
: 1115      1234      ELSE

```

```

: 1116      C 1235      3      runoff_command[rno$h_form_size] = .runoff_command[rno$h_form_size] - 6;
: 1117      1236      )%
: 1118      1237      temp = cli$present(swch_messages);
: 1119      1238      !
: 1120      1239      First initialize all bits that can be set by this qualifier:
: 1121      1240      !
: 1122      1241      runoff_command[rno$v_msg_out] = false;
: 1123      1242      runoff_command[rno$v_msg_user] = false;
: 1124      1243      !
: 1125      1244      SELECT .temp OF
: 1126      1245      SET
: 1127      1246      !
: 1128      1247      [CLIS DEFAULTED, CLIS_PRESENT, CLIS_LOCPRES] :
: 1129      1248      WHILE cli$get_value(swch_messages,desc)
: 1130      1249      DO IF NOT (status = call_tparse( desc
: 1131      1250      ,messages_states
: 1132      1251      ,messages_keys) )
: 1133      U 1252      4      THEN RETURN (%IF DSRPLUS %THEN dsrplus$ facility
: 1134      1253      4      %ELSE      runoff$ facility %FI ^16
: 1135      1254      4      OR ss$ badparam
: 1136      1255      4      OR sts$k_severe)
: 1137      1256      4
: 1138      1257      3      TES;
: 1139      1258      3
: 1140      1259      temp = cli$present(swch_dec_internal);
: 1141      1260      3
: 1142      1261      !      First, initialize all items that can be set by this qualifier:
: 1143      1262      !
: 1144      1263      3      RNO$V_4 OUT FORMAT has already been initialized by the processing
: 1145      1264      3      of the 7DEVICE qualifier.
: 1146      1265      3
: 1147      1266      runoff_command[rno$h_dbg1] = 0;
: 1148      1267      runoff_command[rno$h_dbg2] = 0;
: 1149      1268      3
: 1150      1269      SELECT .temp OF
: 1151      1270      SET
: 1152      1271      !
: 1153      1272      [CLIS DEFAULTED, CLIS_PRESENT, CLIS_LOCPRES] :
: 1154      1273      WHILE cli$get_value(swch_dec_internal,desc)
: 1155      1274      DO call_tparse(desc,dec_internal_states,dec_internal_keys);
: 1156      1275      3
: 1157      1276      3      TES;

```

```

: 1159      1277 3 %SBTTL 'RNOCLI -- final error checking'
: 1160      1278 3
: 1161      1279 3      ! If an LN01 option was used without specifying an LN01,
: 1162      1280 3      !      report the error and exit with a fatal status.
: 1163      1281 3
: 1164      1282 3      IF .ln01_option_used
: 1165      1283 3          AND NOT
: 1166      1284 4          (.runoff_command[rno$v_4_out_format] EQL op_dev_ln01
: 1167      1285 4          OR .runoff_command[rno$v_4_out_format] EQL op_dev_ln01e)
: 1168      1286 4
: 1169      1287 3      THEN
: 1170      1288 4          RETURN (
: 1171      1289 4              ! Our facility name
: 1172      1290 4              %IF DSRPLUS %THEN      dsrplus$_facility
: 1173      1291 3              %ELSE      runoff$_facility %FI * 16)
: 1174      1292 3              OR shr$_confqual      ! shared message from VMS-land
:              OR sts$E_severe;      ! force FATAL status

```

```
: 1176 1293 3 XSBTTL 'RNOCLI -- Point to dropped record file if present'
: 1177 1294 4
: 1178 1295 4 This logic is to support dropped statistical records from DSR and DSRPLUS.
: 1179 1296 4 It looks for the existance of the logical 'DSR$STATISTICAL_RECORDS' and, if
: 1180 1297 4 present, sets up the descriptor RNO$T_DROP_REC to be pointing to the fully
: 1181 1298 4 translated name.
: 1182 1299 4
: 1183 1300 4 If the logical does not exist, then the descriptor RNO$T_DROP_REC will be
: 1184 1301 4 left in its initialized state.
: 1185 1302 3
: 1186 1303 4 BEGIN
: 1187 1304 4 LOCAL
: 1188 1305 4   assign_status,
: 1189 1306 4   trnlog_status,
: 1190 1307 4   tt_buf_desc : BLOCK [8, BYTE],
: 1191 1308 4   tt_name_string : VECTOR [logical_name_buffer_length, BYTE],
: 1192 1309 4   tt_name_desc : BLOCK [8, BYTE],
: 1193 1310 4   logtab;
: 1194 1311 4
: 1195 1312 4   ! Put 'canned' logical name into buffer
: 1196 1313 4
: 1197 1314 4   CH$MOVE (23, UPLIT ('DSR$STATISTICAL_RECORDS'), tt_name_string);
: 1198 1315 4
: 1199 1316 4   ! Initialize descriptor to be pointing to the 'canned' logical
: 1200 1317 4
: 1201 1318 4   tt_name_desc [dsc$w_length] = 23;
: 1202 1319 4   tt_name_desc [dsc$b_dtype] = dsc$k_dtype_t;
: 1203 1320 4   tt_name_desc [dsc$b_class] = dsc$k_class_s;
: 1204 1321 4   tt_name_desc [dsc$a_pointer] = tt_name_string [0];
: 1205 1322 4
: 1206 1323 4   ! The same buffer will be receiving the system's translation of the logical
: 1207 1324 4   ! names. Initialize the destination descriptor to point to the same buffer,
: 1208 1325 4   ! but to indicate that buffer's full length.
: 1209 1326 4
: 1210 1327 4   tt_buf_desc [dsc$w_length] = logical_name_buffer_length;
: 1211 1328 4   tt_buf_desc [dsc$b_dtype] = dsc$k_dtype_t;
: 1212 1329 4   tt_buf_desc [dsc$b_class] = dsc$k_class_s;
: 1213 1330 4   tt_buf_desc [dsc$a_pointer] = tt_name_string [0];
: 1214 1331 4
: 1215 1332 4   ! Before we attempt to translate, we first determine just which logical
: 1216 1333 4   ! name table(s) we will look at (or which one(s) we will ignore).
: 1217 1334 4
: 1218 1335 5 BEGIN
: 1219 1336 5 LOCAL
: 1220 1337 5   foo;
: 1221 1338 5
: 1222 1339 5   foo = .runoff_command[rno$h_dbg1];
: 1223 1340 5
: 1224 1341 5   IF .foo<13,1> THEN
: 1225 1342 5     logtab = 0           ! This says we look at all tables
: 1226 1343 5   ELSE
: 1227 1344 5     logtab = 6           ! This says we only look at the system table.
: 1228 1345 4 END;
: 1229 1346 4
: 1230 1347 4   ! To see if the 'canned' logical name is defined, perform one initial
: 1231 1348 4   ! translation and look at the returned status.
: 1232 1349 4
```

```

: 1233 P 1350 4   trnlog_status = $trnlog (   lognam = tt_name_desc
: 1234 P 1351 4   ,rsllen = tt_name_desc [dsc$w_length]
: 1235 P 1352 4   ,rslbuf = tt_buf_desc
: 1236   1353 4   ,dsbmsk = .logtab);! Don't look at process or
: 1237   1354 4   ! group logical name tables
: 1238   1355 4
: 1239   1356 4   IF .trnlog_status EQL ss$_normal THEN ! If we are not told that no
: 1240   1357 5   BEGIN ! translation took place, then we
: 1241   1358 5   ! must assume (at least initially)
: 1242   1359 5   ! that a logical was defined, and
: 1243   1360 5   ! we must process it.
: 1244   1361 5
: 1245   1362 5   ! However, if the return status wasn't 'normal', then something went
: 1246   1363 5   ! wrong and we must abort processing of the logical.
: 1247   1364 5
: 1248   1365 5
: 1249   1366 5   ! Do the logical name translation up to 9 more times to be sure we have
: 1250   1367 5   ! the real destination.
: 1251   1368 5
: 1252   1369 5   INCR i FROM 1 TO 9 DO
: 1253   1370 5
: 1254   1371 6       IF (trnlog_status =
: 1255 P 1372 6         $trnlog ( lognam = tt_name_desc
: 1256 P 1373 6         ,rsllen = tt_name_desc [dsc$w_length]
: 1257 P 1374 6         ,rslbuf = tt_buf_desc
: 1258   1375 7         ,dsbmsk = 6 ) ! Don't look at process or
: 1259   1376 7         ! group logical name tables
: 1260   1377 7
: 1261   1378 5         ) EQL ss$_notran
: 1262   1379 5       THEN
: 1263   1380 5         EXITLOOP;
: 1264   1381 5
: 1265 P 1382 5       $STR_COPY( STRING = tt_name_desc
: 1266   1383 6         ,TARGET = runoff_command[rno$t_drop_rec] )
: 1267   1384 6
: 1268   1385 5       END
: 1269   1386 3       END;

```

41
41
40
40
45
45

```

: 1271 1387 3 %SBTTL 'RNOCLI -- DSR invocation and exit'
: 1272 1388
: 1273 1389 : Call the DSR application to process a single command line.
: 1274 1390
: 1275 1391
: 1276 1392 %IF NEW_RNODAT_MODULE %THEN
: 1277 1393     _rnodat( runoff_command );           ! Pickup current date & time
: 1278 1394 %FI
: 1279 1395
: 1280 1396     status = RUNOFF( runoff_command ); ! Pass the DSR application the address
: 1281 1397     ! of the complete command information block.
: 1282 1398
: 1283 1399     IF NOT .status                       ! If DSR returns a failure
: 1284 1400     THEN                               ! completion code, save it for
: 1285 1401         exit_status = .status;      ! final exit later.
: 1286 1402
: 1287 1403     END;                               ! end of WHILE --- DO against input filename list
: 1288 1404
: 1289 1405 Tell the RTL to dump any LIB$FIND_FILE context that it may still have
: 1290 1406 lying around.
: 1291 1407
: 1292 1408     IF LIB$FIND_FILE_END NEQA 0 THEN
: 1293 1409         LIB$FIND_FILE_END(lfctx);
: 1294 1410
: 1295 1411 Purge the working set to free up space for other users.
: 1296 1412
: 1297 1413     SYSS$PURGWS(ws_limits);
: 1298 1414
: 1299 1415 Return to the system.
: 1300 1416
: 1301 1417     RETURN .exit_status                 ! Return the final DSR completion code to the caller.
: 1302 1418
: 1303 1419 END;                               ! End of routine RNOCLI

```

```

.TITLE RNOVMS RNOVMS - DSR VMS Command Line Interface
.IDENT \V04-000\
.PSECT _LIB$KEY1$,NOWRT, SHR, PIC,1

```

```

00000 ;TPASKEYSTO
          U.2: .BLKB 0
          4C 4C 41 00000 ;TPASKEYST
          U.4: .ASCII \ALL\
          FF 00003 ;TPASKEYSTO
          U.9: .BLKB 0
          53 4C 41 4E 4F 49 54 49 44 4E 4F 43 00004 ;TPASKEYST
          U.11: .ASCII \CONDITIONALS\
          FF 00010 ;TPASKEYSTO
          U.16: .BLKB 0
          53 54 4E 45 54 4E 4F 43 00011 ;TPASKEYST
          U.18: .ASCII \CONTENTS\
          FF 00019 ;TPASKEYSTO
          U.23: .BLKB 0

```



```

53 45 4C 49 46 0001A ;TPASKEYST
                    U.25: .ASCII \FILES\
                    FF 0001F .BYTE -1
                    00020 ;TPASKEYSTO
                    U.30: .BLKB 0
58 45 44 4E 49 00020 ;TPASKEYST
                    U.32: .ASCII \INDEX\
                    FF 00025 .BYTE -1
                    00026 ;TPASKEYSTO
                    U.37: .BLKB 0
45 52 4F 54 53 45 52 5F 45 56 41 53 00026 ;TPASKEYST
                    U.39: .ASCII \SAVE_RESTORE\
                    FF 00032 .BYTE -1
                    FF 00033 ;TPASKEYFILL
                    U.44: .BYTE -1
                    00034 ;TPASKEYSTO
                    U.46: .BLKB 0
54 55 50 54 55 4F 00034 ;TPASKEYST
                    U.48: .ASCII \OUTPUT\
                    FF 0003A .BYTE -1
                    0003B ;TPASKEYSTO
                    U.53: .BLKB 0
52 45 53 55 0003B ;TPASKEYST
                    U.55: .ASCII \USER\
                    FF 0003F .BYTE -1
                    FF 00040 ;TPASKEYFILL
                    U.62: .BYTE -1
                    00041 ;TPASKEYSTO
                    U.64: .BLKB 0
52 45 54 4E 49 52 50 45 4E 49 4C 00041 ;TPASKEYST
                    U.66: .ASCII \LINEPRINTER\
                    FF 0004C .BYTE -1
                    0004D ;TPASKEYSTO
                    U.72: .BLKB 0
31 30 4E 4C 0004D ;TPASKEYST
                    U.74: .ASCII \LN01\
                    FF 00051 .BYTE -1
                    00052 ;TPASKEYSTO
                    U.80: .BLKB 0
45 31 30 4E 4C 00052 ;TPASKEYST
                    U.82: .ASCII \LN01E\
                    FF 00057 .BYTE -1
                    00058 ;TPASKEYSTO
                    U.88: .BLKB 0
45 4E 49 4C 52 45 44 4E 55 00058 ;TPASKEYST
                    U.90: .ASCII \UNDERLINE\
                    FF 00061 .BYTE -1
                    00062 ;TPASKEYSTO
                    U.96: .BLKB 0
43 49 4C 41 54 49 00062 ;TPASKEYST
                    U.98: .ASCII \ITALIC\
                    FF 00068 .BYTE -1
                    00069 ;TPASKEYSTO
                    U.104: .BLKB 0
54 49 41 52 54 52 4F 50 00069 ;TPASKEYST
                    U.106: .ASCII \PORTRAIT\
                    FF 00071 .BYTE -1

```

```

00072 ;TPASKEYSTO
      U.112: .BLKB 0
45 50 41 43 53 44 4E 41 4C 00072 ;TPASKEYST
      U.114: .ASCII \LANDSCAPE\
      FF 0007B .BYTE -1
0007C ;TPASKEYSTO
      U.120: .BLKB 0
      52 45 44 41 45 48 0007C ;TPASKEYST
      U.122: .ASCII \HEADER\
      FF 00082 .BYTE -1
00083 ;TPASKEYSTO
      U.128: .BLKB 0
      52 45 44 41 45 48 4F 4E 00083 ;TPASKEYST
      U.130: .ASCII \NOHEADER\
      FF 0008B .BYTE -1
0008C ;TPASKEYSTO
      U.136: .BLKB 0
      44 41 4F 4C 0008C ;TPASKEYST
      U.138: .ASCII \LOAD\
      FF 00090 .BYTE -1
00091 ;TPASKEYSTO
      U.144: .BLKB 0
      44 41 4F 4C 4F 4E 00091 ;TPASKEYST
      U.146: .ASCII \NOLOAD\
      FF 00097 .BYTE -1
      FF 00098 ;TPASKEYFILL
      U.154: .BYTE -1
00099 ;TPASKEYSTO
      U.158: .BLKB 0
      31 47 42 44 00099 ;TPASKEYST
      U.160: .ASCII \DBG1\
      FF 0009D .BYTE -1
0009E ;TPASKEYSTO
      U.166: .BLKB 0
      32 47 42 44 0009E ;TPASKEYST
      U.168: .ASCII \DBG2\
      FF 000A2 .BYTE -1
000A3 ;TPASKEYSTO
      U.173: .BLKB 0
      50 49 4C 46 000A3 ;TPASKEYST
      U.175: .ASCII \FLIP\
      FF 000A7 .BYTE -1
000A8 ;TPASKEYSTO
      U.181: .BLKB 0
      30 30 31 54 56 000A8 ;TPASKEYST
      U.183: .ASCII \VT100\
      FF 000AD .BYTE -1
000AE ;TPASKEYSTO
      U.189: .BLKB 0
4D 55 4E 5F 45 4E 49 4C 5F 54 55 50 54 55 4F 000AE ;TPASKEYST
      U.191: .ASCII \OUTPUT_LINE_NUMBER\
      52 45 42 000BD .BYTE -1
      FF 000C0 .BYTE -1
      FF 000C1 ;TPASKEYFILL
      U.199: .BYTE -1
      .PSECT _LIB$STATES, NOWRT, SHR, PIC, 1

```

```

J000 DEBUG_STATES::
      7100 00000 ;TPASTYPE .BLKB 0
      U.5: .WORD 28928
00000000* 00002 ;TPASADDR
      U.6: .LONG <<<RUNOFF_COMMAND+76>-U.6>-4>
00007E00 00006 ;TPASMASK
      U.7: .LONG 32256
      FFFF 0000A ;TPASTARGET
      U.8: .WORD -1
      7101 0000C ;TPASTYPE
      U.12: .WORD 28929
00000000* 0000E ;TPASADDR
      U.13: .LONG <<<RUNOFF_COMMAND+76>-U.13>-4>
00000200 00012 ;TPASMASK
      U.14: .LONG 512
      FFFF 00016 ;TPASTARGET
      U.15: .WORD -1
      7102 00018 ;TPASTYPE
      U.19: .WORD 28930
00000000* 0001A ;TPASADDR
      U.20: .LONG <<<RUNOFF_COMMAND+76>-U.20>-4>
00000400 0001E ;TPASMASK
      U.21: .LONG 1024
      FFFF 00022 ;TPASTARGET
      U.22: .WORD -1
      7103 00024 ;TPASTYPE
      U.26: .WORD 28931
00000000* 00026 ;TPASADDR
      U.27: .LONG <<<RUNOFF_COMMAND+76>-U.27>-4>
00001000 0002A ;TPASMASK
      U.28: .LONG 4096
      FFFF 0002E ;TPASTARGET
      U.29: .WORD -1
      7104 00030 ;TPASTYPE
      U.33: .WORD 28932
00000000* 00032 ;TPASADDR
      U.34: .LONG <<<RUNOFF_COMMAND+76>-U.34>-4>
00002000 00036 ;TPASMASK
      U.35: .LONG 8192
      FFFF 0003A ;TPASTARGET
      U.36: .WORD -1
      7505 0003C ;TPASTYPE
      U.40: .WORD 29957
00000000* 0003E ;TPASADDR
      U.41: .LONG <<<RUNOFF_COMMAND+76>-U.41>-4>
00004000 00042 ;TPASMASK
      U.42: .LONG 16384
      FFFF 00046 ;TPASTARGET
      U.43: .WORD -1
      00048 MESSAGES_STATES::
      .BLKB 0
      7100 00048 ;TPASTYPE
      U.49: .WORD 28928
00000000* 0004A ;TPASADDR
      U.50: .LONG <<<RUNOFF_COMMAND+76>-U.50>-4>

```

```

00100000 0004E ;TPASMASK
          U.51: .LONG 1048576
          FFFF 00052 ;TPASTARGET
          U.52: .WORD -1
          7101 00054 ;TPASTYPE
          U.56: .WORD 28929
00000000* 00056 ;TPASADDR
          U.57: .LONG <<<RUNOFF_COMMAND+76>-U.57>-4>
00200000 0005A ;TPASMASK
          U.58: .LONG 2097152
          FFFF 0005E ;TPASTARGET
          U.59: .WORD -1
          15F6 00060 ;TPASTYPE
          U.60: .WORD 5622
          FFFF 00062 ;TPASTARGET
          U.61: .WORD -1
          00064 DEVICE_STATES:
          .BLKB 0
          00064 DEV_OPTIONS:
          .BLKB 0
          9300 00064 ;TPASTYPE
          U.67: .WORD -27904
          01 00066 ;TPASFLAGS2
          U.68: .BYTE 1
00000000 00067 ;TPASPARAM
          U.69: .LONG 0
00000000V 0006B ;TPASACTION
          U.70: .LONG <<SET_OUTPUT_DEVICE_CODE-U.70>-4>
          FFFF 0006F ;TPASTARGET
          U.71: .WORD -1
          9301 00071 ;TPASTYPE
          U.75: .WORD -27903
          01 00073 ;TPASFLAGS2
          U.76: .BYTE 1
00000004 00074 ;TPASPARAM
          U.77: .LONG 4
00000000V 00078 ;TPASACTION
          U.78: .LONG <<SET_OUTPUT_DEVICE_CODE-U.78>-4>
          FFFF 0007C ;TPASTARGET
          U.79: .WORD -1
          9302 0007E ;TPASTYPE
          U.83: .WORD -27902
          01 00080 ;TPASFLAGS2
          U.84: .BYTE 1
00000005 00081 ;TPASPARAM
          U.85: .LONG 5
00000000V 00085 ;TPASACTION
          U.86: .LONG <<SET_OUTPUT_DEVICE_CODE-U.86>-4>
          FFFF 00089 ;TPASTARGET
          U.87: .WORD -1
          9303 0008B ;TPASTYPE
          U.91: .WORD -27901
          01 0008D ;TPASFLAGS2
          U.92: .BYTE 1
00000009 0008E ;TPASPARAM
          U.93: .LONG 9
00000000V 00092 ;TPASACTION

```

FFFF	00096	U.94: .LONG	<<SAVE_VALUE-U.94>-4>	:
		:TPASTARGET		:
9304	00098	U.95: .WORD	-1	:
		:TPASTYPE		:
01	0009A	U.99: .WORD	-27900	:
		:TPASFLAGS2		:
00000004	0009B	U.100: .BYTE	1	:
		:TPASPARAM		:
00000000V	0009F	U.101: .LONG	4	:
		:TPASACTION		:
FFFF	000A3	U.102: .LONG	<<SAVE_VALUE-U.102>-4>	:
		:TPASTARGET		:
9305	000A5	U.103: .WORD	-1	:
		:TPASTYPE		:
01	000A7	U.107: .WORD	-27899	:
		:TPASFLAGS2		:
00000008	000A8	U.108: .BYTE	1	:
		:TPASPARAM		:
00000000V	000AC	U.109: .LONG	8	:
		:TPASACTION		:
FFFF	000B0	U.110: .LONG	<<SAVE_VALUE-U.110>-4>	:
		:TPASTARGET		:
9306	000B2	U.111: .WORD	-1	:
		:TPASTYPE		:
01	000B4	U.115: .WORD	-27898	:
		:TPASFLAGS2		:
00000005	000B5	U.116: .BYTE	1	:
		:TPASPARAM		:
00000000V	000B9	U.117: .LONG	5	:
		:TPASACTION		:
FFFF	000BD	U.118: .LONG	<<SAVE_VALUE-U.118>-4>	:
		:TPASTARGET		:
9307	000BF	U.119: .WORD	-1	:
		:TPASTYPE		:
01	000C1	U.123: .WORD	-27897	:
		:TPASFLAGS2		:
00000002	000C2	U.124: .BYTE	1	:
		:TPASPARAM		:
00000000V	000C6	U.125: .LONG	2	:
		:TPASACTION		:
FFFF	000CA	U.126: .LONG	<<SAVE_VALUE-U.126>-4>	:
		:TPASTARGET		:
9308	000CC	U.127: .WORD	-1	:
		:TPASTYPE		:
01	000CE	U.131: .WORD	-27896	:
		:TPASFLAGS2		:
00000003	000CF	U.132: .BYTE	1	:
		:TPASPARAM		:
00000000V	000D3	U.133: .LONG	3	:
		:TPASACTION		:
FFFF	000D7	U.134: .LONG	<<SAVE_VALUE-U.134>-4>	:
		:TPASTARGET		:
9309	000D9	U.135: .WORD	-1	:
		:TPASTYPE		:
01	000DB	U.139: .WORD	-27895	:
		:TPASFLAGS2		:
		U.140: .BYTE	1	:

```
00000006 000DC ;TPASPARAM
              U.141: .LONG 6
00000000V 000E0 ;TPASACTION
              U.142: .LONG <<SAVE_VALUE-U.142>-4>
          FFFF 000E4 ;TPASTARGET
              U.143: .WORD -1
          930A 000E6 ;TPASTYPE
              U.147: .WORD -27894
              01 000E8 ;TPAS$FLAGS2
              U.148: .BYTE 1
00000007 000E9 ;TPASPARAM
              U.149: .LONG 7
00000000V 000ED ;TPASACTION
              U.150: .LONG <<SAVE_VALUE-U.150>-4>
          FFFF 000F1 ;TPASTARGET
              U.151: .WORD -1
          15F6 000F3 ;TPASTYPE
              U.152: .WORD 5622
          FFFE 000F5 ;TPASTARGET
              U.153: .WORD -2
              000F7 .BLKB 1
              000F8 DEC_INTERNAL STATES::
              .BLKB 0
          102C 000F8 OPTIONS: .BLKB 0
              U.156: .WORD 4140
          0000* 000FA ;TPASTARGET
              U.157: .WORD <<OPTIONS-U.157>-2>
          1300 000FC ;TPASTYPE
              U.161: .WORD 4864
              01 000FE ;TPAS$FLAGS2
              U.162: .BYTE 1
0000000A 000FF ;TPASPARAM
              U.163: .LONG 10
          0000* 00103 ;TPASTARGET
              U.165: .WORD <<U.164-U.165>-2>
          1301 00105 ;TPASTYPE
              U.169: .WORD 4865
              01 00107 ;TPAS$FLAGS2
              U.170: .BYTE 1
0000000B 00108 ;TPASPARAM
              U.171: .LONG 11
          0000* 0010C ;TPASTARGET
              U.172: .WORD <<U.164-U.172>-2>
          9302 0010E ;TPASTYPE
              U.176: .WORD -27902
              01 00110 ;TPAS$FLAGS2
              U.177: .BYTE 1
00000002 00111 ;TPASPARAM
              U.178: .LONG 2
00000000V 00115 ;TPASACTION
              U.179: .LONG <<SET_OUTPUT_DEVICE_CODE-U.179>-4>
          0000* 00119 ;TPASTARGET
              U.180: .WORD <<OPTIONS-U.180>-2>
          9303 0011B ;TPASTYPE
              U.184: .WORD -27901
              01 0011D ;TPAS$FLAGS2
```

00000003	0011E	U.185: .BYTE	1	:
		:TPASPARAM		:
00000000V	00122	U.186: .LONG	3	:
		:TPASACTION		:
0000*	00126	U.187: .LONG	<<SET_OUTPUT_DEVICE_CODE-U.187>-4>	:
		:TPASTARGET		:
9304	00128	U.188: .WORD	<<OPTIONS-U.188>-2>	:
		:TPASTYPE		:
01	0012A	U.192: .WORD	-27900	:
		:TPASFLAGS2		:
0000000C	0012B	U.193: .BYTE	1	:
		:TPASPARAM		:
00000000V	0012F	U.194: .LONG	12	:
		:TPASACTION		:
0000*	00133	U.195: .LONG	<<SAVE_VALUE-U.195>-4>	:
		:TPASTARGET		:
15F6	00135	U.196: .WORD	<<OPTIONS-U.196>-2>	:
		:TPASTYPE		:
FFFF	00137	U.197: .WORD	5622	:
		:TPASTARGET		:
1025	00139	U.198: .WORD	-1	:
		:TPASTYPE		:
0000*	0013B	U.200: .WORD	4133	:
		:TPASTARGET		:
15F6	0013D	U.202: .WORD	<<U.201-U.202>-2>	:
		:TPASTYPE		:
FFFF	0013F	U.203: .WORD	5622	:
		:TPASTARGET		:
	00141	U.204: .WORD	-1	:
		:DBG1		:
003D	00141	U.164: .BLKB	0	:
		:TPASTYPE		:
003A	00143	U.205: .WORD	61	:
		:TPASTYPE		:
142C	00145	U.206: .WORD	58	:
		:TPASTYPE		:
0000*	00147	U.207: .WORD	5164	:
		:TPASTARGET		:
1025	00149	U.208: .WORD	<<OPTIONS-U.208>-2>	:
		:TPASTYPE		:
0000*	0014B	U.209: .WORD	4133	:
		:TPASTARGET		:
91F5	0014D	U.210: .WORD	<<U.201-U.210>-2>	:
		:TPASTYPE		:
00000000V	0014F	U.211: .WORD	-28171	:
		:TPASACTION		:
0000*	00153	U.212: .LONG	<<SAVE_VALUE-U.212>-4>	:
		:TPASTARGET		:
15F6	00155	U.213: .WORD	<<OPTIONS-U.213>-2>	:
		:TPASTYPE		:
0000*	00157	U.214: .WORD	5622	:
		:TPASTARGET		:
	00159	U.215: .WORD	<<OPTIONS-U.215>-2>	:
		:WHICH_BASE		:
1058	00159	U.201: .BLKB	0	:
		:TPASTYPE		:
		U.216: .WORD	4184	:

```
0000* 0015B ;TPASTARGET  
          U.218: .WORD <<U.217-U.218>-2> ;  
104F 0015D ;TPASTYPE ;  
          U.219: .WORD 4175 ;  
0000* 0015F ;TPASTARGET ;  
          U.221: .WORD <<U.220-U.221>-2> ;  
1044 00161 ;TPASTYPE ;  
          U.222: .WORD 4164 ;  
0000* 00163 ;TPASTARGET ;  
          U.224: .WORD <<U.223-U.224>-2> ;  
91F5 00165 ;TPASTYPE ;  
          U.225: .WORD -28171 ;  
00000000V 00167 ;TPASACTION ;  
          U.226: .LONG <<SAVE_VALUE-U.226>-4> ;  
0000* 0016B ;TPASTARGET ;  
          U.227: .WORD <<OPTIONS-U.227>-2> ;  
15F6 0016D ;TPASTYPE ;  
          U.228: .WORD 5622 ;  
0000* 0016F ;TPASTARGET ;  
          U.229: .WORD <<OPTIONS-U.229>-2> ;  
          00171 ;HEX_NUM ;  
          U.217: .BLKB 0 ;  
95F5 00171 ;TPASTYPE ;  
          U.230: .WORD -27147 ;  
00000000V 00173 ;TPASACTION ;  
          U.231: .LONG <<SAVE_VALUE-U.231>-4> ;  
0000* 00177 ;TPASTARGET ;  
          U.232: .WORD <<OPTIONS-U.232>-2> ;  
          00179 ;OCT_NUM ;  
          U.220: .BLKB 0 ;  
95F4 00179 ;TPASTYPE ;  
          U.233: .WORD -27148 ;  
00000000V 0017B ;TPASACTION ;  
          U.234: .LONG <<SAVE_VALUE-U.234>-4> ;  
0000* 0017F ;TPASTARGET ;  
          U.235: .WORD <<OPTIONS-U.235>-2> ;  
          00181 ;DEC_NUM ;  
          U.223: .BLKB 0 ;  
95F3 00181 ;TPASTYPE ;  
          U.236: .WORD -27149 ;  
00000000V 00183 ;TPASACTION ;  
          U.237: .LONG <<SAVE_VALUE-U.237>-4> ;  
0000* 00187 ;TPASTARGET ;  
          U.238: .WORD <<OPTIONS-U.238>-2> ;  
          00189 ;BLKB 3 ;  
          0018C CHARS_STATES.: ;  
          ;BLKB 0 ;  
1225 0018C ;TPASTYPE ;  
          U.240: .WORD 4645 ;  
          01 0018E ;TPASFLAGS2 ;  
          U.241: .BYTE 1 ;  
00000001 0018F ;TPASPARAM ;  
          U.242: .LONG 1 ;  
0000* 00193 ;TPASTARGET ;  
          U.244: .WORD <<U.243-U.244>-2> ;  
11F7 00195 ;TPASTYPE ;  
          U.245: .WORD 4599 ;
```


FFFF	00197	;TPASTARGET				
		U.246:	.WORD	-1		:
15F6	00199	;TPASTYPE				:
		U.247:	.WORD	5622		:
FFFE	0019B	;TPASTARGET				:
		U.248:	.WORD	-2		:
	0019D	;CHARS_WHICH BASE				:
		U.243:	.BLKB	0		:
1058	0019D	;TPASTYPE				:
		U.249:	.WORD	4184		:
0000*	0019F	;TPASTARGET				:
		U.251:	.WORD	<<U.250-U.251>-2>		:
104F	001A1	;TPASTYPE				:
		U.252:	.WORD	4175		:
0000*	001A3	;TPASTARGET				:
		U.254:	.WORD	<<U.253-U.254>-2>		:
1044	001A5	;TPASTYPE				:
		U.255:	.WORD	4164		:
0000*	001A7	;TPASTARGET				:
		U.257:	.WORD	<<U.256-U.257>-2>		:
91F3	001A9	;TPASTYPE				:
		U.258:	.WORD	-28173		:
00000000V	001AB	;TPASACTION				:
		U.259:	.LONG	<<SAVE_VALUE-U.259>-4>		:
0000*	001AF	;TPASTARGET				:
		U.261:	.WORD	<<U.260-U.261>-2>		:
91F5	001B1	;TPASTYPE				:
		U.262:	.WORD	-28171		:
00000000V	001B3	;TPASACTION				:
		U.263:	.LONG	<<SAVE_VALUE-U.263>-4>		:
0000*	001B7	;TPASTARGET				:
		U.264:	.WORD	<<U.260-U.264>-2>		:
15F6	001B9	;TPASTYPE				:
		U.265:	.WORD	5622		:
FFFE	001BB	;TPASTARGET				:
		U.266:	.WORD	-2		:
	001BD	;CHARS_HEX NUM				:
		U.250:	.BLKB	0		:
95F5	001BD	;TPASTYPE				:
		U.267:	.WORD	-27147		:
00000000V	001BF	;TPASACTION				:
		U.268:	.LONG	<<SAVE_VALUE-U.268>-4>		:
0000*	001C3	;TPASTARGET				:
		U.269:	.WORD	<<U.260-U.269>-2>		:
	001C5	;CHARS_OCT NUM				:
		U.253:	.BLKB	0		:
95F4	001C5	;TPASTYPE				:
		U.270:	.WORD	-27148		:
00000000V	001C7	;TPASACTION				:
		U.271:	.LONG	<<SAVE_VALUE-U.271>-4>		:
0000*	001CB	;TPASTARGET				:
		U.272:	.WORD	<<U.260-U.272>-2>		:
	001CD	;CHARS_DEC NUM				:
		U.256:	.BLKB	0		:
95F3	001CD	;TPASTYPE				:
		U.273:	.WORD	-27149		:
00000000V	001CF	;TPASACTION				:

```
0000* 001D3 U.274: .LONG <<SAVE_VALUE-U.274>-4> ;
;TPASTARGET ;
U.275: .WORD <<U.260-U.275>-2> ;
001D5 ;CHARS_DONE 0 ;
U.260: .BLKB 0 ;
11F7 001D5 ;TPASTYPE 4599 ;
U.276: .WORD 4599 ;
FFFF 001D7 ;TPASTARGET -1 ;
U.277: .WORD -1 ;
15F6 001D9 ;TPASTYPE 5622 ;
U.278: .WORD 5622 ;
FFFE 001DB ;TPASTARGET -2 ;
U.279: .WORD -2 ;

.PSECT _LIB$KEY0$,NOWRT, SHR, PIC,1

00000 DEBUG_KEYS::
00000 ;TPASKEY0 .BLKB 0
U.1: .BLKB 0
0000* 00000 ;TPASKEY <U.2-U.1> ;
U.3: .WORD <U.2-U.1> ;
0000* 00002 ;TPASKEY <U.9-U.1> ;
U.10: .WORD <U.9-U.1> ;
0000* 00004 ;TPASKEY <U.16-U.1> ;
U.17: .WORD <U.16-U.1> ;
0000* 00006 ;TPASKEY <U.23-U.1> ;
U.24: .WORD <U.23-U.1> ;
0000* 00008 ;TPASKEY <U.30-U.1> ;
U.31: .WORD <U.30-U.1> ;
0000* 0000A ;TPASKEY <U.37-U.1> ;
U.38: .WORD <U.37-U.1> ;
0000C MESSAGES_KEYS::
0000C ;TPASKEY0 .BLKB 0
U.45: .BLKB 0
0000* 0000C ;TPASKEY <U.46-U.45> ;
U.47: .WORD <U.46-U.45> ;
0000* 0000E ;TPASKEY <U.53-U.45> ;
U.54: .WORD <U.53-U.45> ;
00010 DEVICE_KEYS::
00010 ;TPASKEY0 .BLKB 0
U.63: .BLKB 0
0000* 00010 ;TPASKEY <U.64-U.63> ;
U.65: .WORD <U.64-U.63> ;
0000* 00012 ;TPASKEY <U.72-U.63> ;
U.73: .WORD <U.72-U.63> ;
0000* 00014 ;TPASKEY <U.80-U.63> ;
U.81: .WORD <U.80-U.63> ;
0000* 00016 ;TPASKEY <U.88-U.63> ;
U.89: .WORD <U.88-U.63> ;
0000* 00018 ;TPASKEY <U.96-U.63> ;
U.97: .WORD <U.96-U.63> ;
0000* 0001A ;TPASKEY <U.104-U.63> ;
U.105: .WORD <U.104-U.63> ;
0000* 0001C ;TPASKEY
```

```

0000* 0001E U.113: .WORD <U.112-U.63>
;TPASKEY
0000* 00020 U.121: .WORD <U.120-U.63>
;TPASKEY
0000* 00022 U.129: .WORD <U.128-U.63>
;TPASKEY
0000* 00024 U.137: .WORD <U.136-U.63>
;TPASKEY
00026 U.145: .WORD <U.144-U.63>
.BLKB 2
00028 DEC_INTERNAL_KEYS::
.BLKB 0
00028 ;TPASKEY0
U.155: .BLKB 0
0000* 00028 ;TPASKEY
U.159: .WORD <U.158-U.155>
0000* 0002A ;TPASKEY
U.167: .WORD <U.166-U.155>
0000* 0002C ;TPASKEY
U.174: .WORD <U.173-U.155>
0000* 0002E ;TPASKEY
U.182: .WORD <U.181-U.155>
0000* 00030 ;TPASKEY
U.190: .WORD <U.189-U.155>
00032 .BLKB 2
00034 CHARS_KEYS::
.BLKB 0
00034 ;TPASKEY0
U.239: .BLKB 0

```

.PSECT \$SPLITS,NOWRT,NOEXE,2

```

00 00 31 50 00000 P.AAB: .ASCII \P1\<0><0>
00000002 00004 P.AAA: .LONG 2
00000000' 00008 .ADDRESS P.AAB
00 00 00 45 43 41 50 53 48 43 41 42 0000C P.AAD: .ASCII \BACKSPACE\<0><0><0>
00000009 00018 P.AAC: .LONG 9
00000000' 0001C .ADDRESS P.AAD
00 47 4F 4C 00020 P.AAF: .ASCII \LOG\<0>
00000003 00024 P.AAE: .LONG 3
00000000' 00028 .ADDRESS P.AAF
00 00 00 45 53 55 41 50 0002C P.AAH: .ASCII \PAUSE\<0><0><0>
00000005 00034 P.AAG: .LONG 5
00000000' 00038 .ADDRESS P.AAH
45 43 4E 45 55 51 45 53 0003C P.AAJ: .ASCII \SEQUENCE\
00000008 00044 P.AAI: .LONG 8
00000000' 00048 .ADDRESS P.AAJ
45 54 41 4C 55 4D 49 53 0004C P.AAL: .ASCII \SIMULATE\
00000008 00054 P.AAK: .LONG 8
00000000' 00058 .ADDRESS P.AAL
00 00 54 55 50 54 55 4F 0005C P.AAN: .ASCII \OUTPUT\<0><0>
00000006 00064 P.AAM: .LONG 6
00000000' 00068 .ADDRESS P.AAN
00 00 54 55 50 54 55 4F 0006C P.AAP: .ASCII \OUTPUT\<0><0>
00000006 00074 P.AAO: .LONG 6
00000000' 00078 .ADDRESS P.AAP
45 54 41 49 44 45 4D 52 45 54 4E 49 0007C P.AAR: .ASCII \INTERMEDIATE\

```

											0000000C	00088	P.AAQ:	.LONG 12	
											00000000'	0008C		.ADDRESS P.AAR	
	45	54	41	49	44	45	4D	52	45	54	4E 49	00090	P.AAT:	.ASCII \INTERMEDIATE\	
											0000000C	0009C	P.AAS:	.LONG 12	
	00	53	52	41	42	5F	45	47	4E	41	48 43	000A0		.ADDRESS P.AAT	
											0000000B	000A4	P.AAV:	.ASCII \CHANGE_BARS\<0>	
											00000000'	000B0	P.AAU:	.LONG 11	
	00	53	52	41	42	5F	45	47	4E	41	48 43	000B4		.ADDRESS P.AAV	
											0000000B	000B8	P.AAX:	.ASCII \CHANGE_BARS\<0>	
											00000000'	000C4	P.AAW:	.LONG 11	
41	52	41	48	43	5F	45	4E	49	4C	52	45 44 4E 55	000C8		.ADDRESS P.AAX	
										00	52 45 54 43	000CC	P.AAZ:	.ASCII \UNDERLINE_CHARACTER\<0>	
											00000013	000DB			
											00000000'	000E0	P.AAY:	.LONG 19	
41	52	41	48	43	5F	45	4E	49	4C	52	45 44 4E 55	000E4		.ADDRESS P.AAZ	
										00	52 45 54 43	000F7	P.ABB:	.ASCII \UNDERLINE_CHARACTER\<0>	
											00000013	000FC			
4C	52	45	44	4E	55	5F	45	54	41	52	41 50 45 53	00100	P.ABA:	.LONG 19	
										00	00 45 4E 49	00104	P.ABD:	.ADDRESS P.ABB	
											00000012	00113		.ASCII \SEPARATE_UNDERLINE\<0><0>	
4C	52	45	44	4E	55	5F	45	54	41	52	41 50 45 53	00118	P.ABC:	.LONG 18	
										00	00 45 4E 49	0011C		.ADDRESS P.ABD	
											00000012	00120	P.ABF:	.ASCII \SEPARATE_UNDERLINE\<0><0>	
											00000000'	0012F			
45	44	4E	55	5F	47	4E	49	43	41	50	53 4E 4F 4E	00134	P.ABE:	.LONG 18	
										45	4E 49 4C 52	00138		.ADDRESS P.ABF	
											00000014	0013C	P.ABH:	.ASCII \NONSPACING_UNDERLINE\<0>	
											00000000'	0014B			
45	44	4E	55	5F	47	4E	49	43	41	50	53 4E 4F 4E	00150	P.ABG:	.LONG 20	
										45	4E 49 4C 52	00154		.ADDRESS P.ABH	
											00000014	00158	P.ABJ:	.ASCII \NONSPACING_UNDERLINE\<0>	
											00000000'	00167			
											00000014	0016C	P.ABI:	.LONG 20	
											00000000'	00170		.ADDRESS P.ABJ	
											00000005	00174	P.ABL:	.ASCII \PAGES\<0><0><0>	
											00000000'	0017C	P.ABK:	.LONG 5	
											00000000'	00180		.ADDRESS P.ABL	
											00000005	00184	P.ABN:	.ASCII \PAGES\<0><0><0>	
											00000000'	0018C	P.ABM:	.LONG 5	
											00000000'	00190		.ADDRESS P.ABN	
											00000007	00194	P.ABP:	.ASCII \VARIANT\<0>	
											00000000'	0019C	P.ABO:	.LONG 7	
											00000000'	001A0		.ADDRESS P.ABP	
											00000007	001A4	P.ABR:	.ASCII \VARIANT\<0>	
											00000000'	001AC	P.ABQ:	.LONG 7	
											00000000'	001B0		.ADDRESS P.ABR	
											44 4C 4F 42	001B4	P.ABT:	.ASCII \BOLD\<0>	
											00000004	001B8	P.ABS:	.LONG 4	
											00000000'	001BC		.ADDRESS P.ABT	
											44 4C 4F 42	001C0	P.ABV:	.ASCII \BOLD\<0>	
											00000004	001C4	P.ABU:	.LONG 4	
											00000000'	001C8		.ADDRESS P.ABV	
											4E 57 4F 44	001CC	P.ABX:	.ASCII \DOWN\<0>	
											00000004	001D0	P.ABW:	.LONG 4	
											00000000'	001D4		.ADDRESS P.ABX	
											4E 57 4F 44	001D8	P.ABZ:	.ASCII \DOWN\<0>	

										00000004	001DC	P.ABY:	.LONG	4					
										00000000	001E0		.ADDRESS	P.ABZ					
00	00	00	45	5A	49	53	5F	4D	52	4F	46	001E4	P.ACB:	.ASCII	\FORM_SIZE\<0><0><0>				
										00000009	001F0	P.ACA:	.LONG	9					
										00000000	001F4		.ADDRESS	P.ACB					
00	00	00	45	5A	49	53	5F	4D	52	4F	46	001F8	P.ACD:	.ASCII	\FORM_SIZE\<0><0><0>				
										00000009	00204	P.ACC:	.LONG	9					
										00000000	00208		.ADDRESS	P.ACD					
				00	00	00	54	48	47	49	52	0020C	P.ACF:	.ASCII	\RIGHT\<0><0><0>				
										00000005	00214	P.ACE:	.LONG	5					
										00000000	00218		.ADDRESS	P.ACF					
				00	00	00	54	48	47	49	52	0021C	P.ACH:	.ASCII	\RIGHT\<0><0><0>				
										00000005	00224	P.ACG:	.LONG	5					
										00000000	00228		.ADDRESS	P.ACH					
				00	00	00	47	55	42	45	44	0022C	P.ACJ:	.ASCII	\DEBUG\<0><0><0>				
										00000005	00234	P.ACI:	.LONG	5					
										00000000	00238		.ADDRESS	P.ACJ					
				00	00	00	47	55	42	45	44	0023C	P.ACL:	.ASCII	\DEBUG\<0><0><0>				
										00000005	00244	P.ACK:	.LONG	5					
										00000000	00248		.ADDRESS	P.ACL					
				00	00	45	43	49	56	45	44	0024C	P.ACN:	.ASCII	\DEVICE\<0><0>				
										00000006	00254	P.ACM:	.LONG	6					
										00000000	00258		.ADDRESS	P.ACN					
				00	00	45	43	49	56	45	44	0025C	P.ACP:	.ASCII	\DEVICE\<0><0>				
										00000006	00264	P.ACO:	.LONG	6					
										00000000	00268		.ADDRESS	P.ACP					
				53	45	47	41	53	53	45	4D	0026C	P.ACR:	.ASCII	\MESSAGES\<0><0>				
										00000008	00274	P.ACQ:	.LONG	8					
										00000000	00278		.ADDRESS	P.ACR					
				53	45	47	41	53	53	45	4D	0027C	P.ACT:	.ASCII	\MESSAGES\<0><0>				
										00000008	00284	P.ACS:	.LONG	8					
										00000000	00288		.ADDRESS	P.ACT					
4C	41	4E	52	45	54	4E	49	5F	43	45	44	0028C	P.ACV:	.ASCII	\DEC_INTERNAL\<0><0>				
										0000000C	00298	P.ACU:	.LONG	12					
										00000000	0029C		.ADDRESS	P.ACV					
				4C	41	4E	52	45	54	4E	49	5F	43	45	44	002A0	P.ACX:	.ASCII	\DEC_INTERNAL\<0><0>
										0000000C	002AC	P.ACW:	.LONG	12					
										00000000	002B0		.ADDRESS	P.ACX					
4C	41	43	49	54	53	49	54	41	54	53	24	52	53	44	002B4	P.ACY:	.ASCII	\DSR\$STATISTICAL_RECORDS\<0>	
				00	53	44	52	4F	43	45	52	5F	002C3						

.PSECT \$OWNS,NOEXE,2

00000 RUNOFF_COMMAND:
 .BLKB 100
 00064 SWCH_ID: .BLKB 4
 00068 LN01_OPTION_USED:
 .BLRB 4
 7FFFFFFF 00000000 0006C WS_LIMITS:
 .LONG 0, 2147483647

\$STR\$DESC= RUNOFF_COMMAND
 \$STR\$BIN_DESC= RUNOFF_COMMAND
 \$STR\$TARGET= RUNOFF_COMMAND
 \$STR\$DESC= RUNOFF_COMMAND+8
 \$STR\$BIN_DESC= RUNOFF_COMMAND+8
 \$STR\$DESC= RUNOFF_COMMAND+16

.....

```

$STR$BIN_DESC= RUNOFF_COMMAND+16
$STR$DESC= RUNOFF_COMMAND+48
$STR$BIN_DESC= RUNOFF_COMMAND+48
$STR$DESC= RUNOFF_COMMAND+56
$STR$BIN_DESC= RUNOFF_COMMAND+56
$STR$DESC= RUNOFF_COMMAND+24
$STR$BIN_DESC= RUNOFF_COMMAND+24
$STR$DESC= RUNOFF_COMMAND+32
$STR$BIN_DESC= RUNOFF_COMMAND+32
$STR$DESC= RUNOFF_COMMAND+40
$STR$BIN_DESC= RUNOFF_COMMAND+40
$STR$TARGET= RUNOFF_COMMAND+24

```

```

.EXTRN RUNOFF, SYSSPURGWS
.EXTRN SYSSNUMTIM, CLISGET_VALUE
.EXTRN CLISPRESENT, LIBSCVT_DTB
.EXTRN LIB$PARSE, LIB$LP_LINES
.EXTRN RUNOFF$FACILITY
.EXTRN CLIS_ABSENT, CLIS_PRESENT
.EXTRN CLIS_DEFAULTED, CLIS_NEGATED
.EXTRN LFFCTX, XST$COPY
.EXTRN STR$FAILURE, SYS$STRNLOG
.WEAK LIB$FIND_FILE_END
.WEAK CLIS_LOCPRES, CLIS_LOCNEG

```

.PSECT \$CODE\$,NOWRT,2

OFFC 00000

```

.ENTRY RNOCLI, Save R2,R3,R4,R5,R6,R7,R8,R9,R10,- R11 ; 0595
      MOVL #CLIS_PRESENT, R11
      MOVAB P.AAA, R10
      MOVAB RUNOFF_COMMAND+76, R9
      MOVAB -284(SP), SP
      MOVL #34471936, $STR$DESC ; 0633
      CLRL $STR$DESC+4
      MOVL #268435457, EXIT_STATUS ; 0635
      CLRL LFFCTX ; 0640
      PUSHAB DESC ; 0642
      PUSHL R10
      CALLS #2, CLISGET_VALUE
      BLBS R0, 2$
      BRW 137$
      CLRL INDEX ; 0649
      CLRL RUNOFF_COMMAND[INDEX] ; 0650
      AOBLEQ #5, INDEX, 3$
      MNEGB #1, RUNOFF_COMMAND+85 ; 0653
      MOVL #34471936, $STR$DESC ; 0663
      CLRL $STR$DESC+4
      PUSHAB STR$FAILURE ; 0665
      CLRL -(SP)
      PUSHAB $STR$TARGET
      PUSHAB $STR$STRING
      CLRL -(SP)
      CALLS #5, XST$COPY
      MOVL #34471936, $STR$DESC ; 0668
      CLRL $STR$DESC+4
      MOVL #34471936, $STR$DESC ; 0671
      CLRL $STR$DESC+4

```

```

      5B 00000000G 8F D0 00002
      5A 00000000' EF 9E 00009
      59 00000000' EF 9E 00010
      5E FEE4 CE 9E 00017
      F8 AD 020E0000 8F D0 0001C
           FC AD D4 00024
      57 10000001 8F D0 00027
           00000000G EF D4 0002E
           F8 AD 9F 00034 1$:
           SA DD 00037
           00 02 FB 00039
           03 50 E8 00040
           OBC3 31 00043
           50 D4 00046 2$:
           B4 A940 D4 00048 3$:
      F8 50 05 F3 0004C
           09 A9 01 8E 00050
           B4 A9 020E0000 8F D0 00054
           B8 A9 D4 0005C
           00000000G EF 9F 0005F
           7E D4 00065
           B4 A9 9F 00067
           F8 AD 9F 0006A
           7E D4 0006D
           00000000G EF 05 FB 0006F
           BC A9 020E0000 8F D0 00076
           C0 A9 D4 0007E
           C4 A9 020E0000 8F D0 00081
           C8 A9 D4 00089

```

: R

02	A9	08	88	00183	14\$:	BISB2	#8,	RUNOFF_COMMAND+78	0758	
00000000G	8F	50	D1	00187	15\$:	CMPL	RO,	#CLIS_ABSENT	0760	
		12	13	0018E		BEQL	16\$			
00000000G	8F	50	D1	00190		CMPL	RO,	#CLIS_NEGATED		
		09	13	00197		BEQL	16\$			
00000000G	8F	50	D1	00199		CMPL	RO,	#CLIS_LOCNEG		
		04	12	001A0		BNEQ	17\$			
02	A9	04	8A	001A2	16\$:	BICB2	#4,	RUNOFF_COMMAND+78	0761	
00000000G	8F	50	D1	001A6	17\$:	CMPL	RO,	#CLIS_DEFAULTED	0763	
		0E	13	001AD		BEQL	18\$			
	5B	50	D1	001AF		CMPL	RO,	R11		
		09	13	001B2		BEQL	18\$			
00000000G	8F	50	D1	001B4		CMPL	RO,	#CLIS_LOCPRES		
		04	12	001BB		BNEQ	19\$			
02	A9	04	88	001BD	18\$:	BISB2	#4,	RUNOFF_COMMAND+78	0764	
00000000G	00	30	AA	9F	001C1	19\$:	PUSHAB	P.AAG	0768	
	6E	01	FB	001C4		CALLS	#1,	CLISPRESENT		
00000000G	8F	50	D0	001CB		MOVL	RO,	TEMP		
		50	D1	001CE		CMPL	RO,	#CLIS_ABSENT	0773	
00000000G	8F	09	13	001D5		BEQL	20\$			
		50	D1	001D7		CMPL	RO,	#CLIS_DEFAULTED		
		05	12	001DE		BNEQ	21\$			
03	A9	80	8F	8A	001E0	20\$:	BICB2	#128,	RUNOFF_COMMAND+79	0774
00000000G	8F	50	D1	001E5	21\$:	CMPL	RO,	#CLIS_NEGATED	0776	
		17	13	001EC		BEQL	22\$			
	5B	50	D1	001EE		CMPL	RO,	R11		
		12	13	001F1		BEQL	22\$			
00000000G	8F	50	D1	001F3		CMPL	RO,	#CLIS_LOCPRES		
		09	13	001FA		BEQL	22\$			
00000000G	8F	50	D1	001FC		CMPL	RO,	#CLIS_LOCNEG		
		05	12	00203		BNEQ	23\$			
03	A9	80	8F	88	00205	22\$:	BISB2	#128,	RUNOFF_COMMAND+79	0777
00000000G	8F	50	D1	0020A	23\$:	CMPL	RO,	#CLIS_ABSENT	0779	
		12	13	00211		BEQL	24\$			
00000000G	8F	50	D1	00213		CMPL	RO,	#CLIS_NEGATED		
		09	13	0021A		BEQL	24\$			
00000000G	8F	50	D1	0021C		CMPL	RO,	#CLIS_LOCNEG		
		05	12	00223		BNEQ	25\$			
03	A9	40	8F	8A	00225	24\$:	BICB2	#64,	RUNOFF_COMMAND+79	0780
00000000G	8F	50	D1	0022A	25\$:	CMPL	RO,	#CLIS_DEFAULTED	0782	
		0E	13	00231		BEQL	26\$			
	5B	50	D1	00233		CMPL	RO,	R11		
		09	13	00236		BEQL	26\$			
00000000G	8F	50	D1	00238		CMPL	RO,	#CLIS_LOCPRES		
		05	12	0023F		BNEQ	27\$			
03	A9	40	8F	88	00241	26\$:	BISB2	#64,	RUNOFF_COMMAND+79	0783
00000000G	00	40	AA	9F	00246	27\$:	PUSHAB	P.AAI	0787	
	6E	01	FB	00249		CALLS	#1,	CLISPRESENT		
00000000G	8F	50	D0	00250		MOVL	RO,	TEMP		
		50	D1	00253		CMPL	RO,	#CLIS_ABSENT	0792	
00000000G	8F	09	13	0025A		BEQL	28\$			
		50	D1	0025C		CMPL	RO,	#CLIS_DEFAULTED		
		04	12	00263		BNEQ	29\$			
04	A9	04	8A	00265	28\$:	BICB2	#4,	RUNOFF_COMMAND+80	0793	
00000000G	8F	50	D1	00269	29\$:	CMPL	RO,	#CLIS_NEGATED	0795	
		17	13	00270		BEQL	30\$			
	5B	50	D1	00272		CMPL	RO,	R11		

			12	13	00275	BEQL	30\$			
00000000G	8F		50	D1	00277	CMPL	RO,	#CLIS_LOCPRES		
			09	13	0027E	BEQL	30\$			
00000000G	8F		50	D1	00280	CMPL	RO,	#CLIS_LOCNEG		
			04	12	00287	BNEQ	31\$			
	04	A9	04	88	00289	BISB2	#4,	RUNOFF_COMMAND+80	0796	
00000000G	8F		50	D1	0028D	CMPL	RO,	#CLIS_ABSENT	0798	
			12	13	00294	BEQL	32\$			
00000000G	8F		50	D1	00296	CMPL	RO,	#CLIS_NEGATED		
			09	13	0029D	BEQL	32\$			
00000000G	8F		50	D1	0029F	CMPL	RO,	#CLIS_LOCNEG		
			04	12	002A6	BNEQ	33\$			
	04	A9	02	8A	002A8	BICB2	#2,	RUNOFF_COMMAND+80	0799	
00000000G	8F		50	D1	002AC	CMPL	RO,	#CLIS_DEFAULTED	0801	
			0E	13	002B3	BEQL	34\$			
		5B	50	D1	002B5	CMPL	RO,	R11		
			09	13	002B8	BEQL	34\$			
00000000G	8F		50	D1	002BA	CMPL	RO,	#CLIS_LOCPRES		
			04	12	002C1	BNEQ	35\$			
	04	A9	02	88	002C3	BISB2	#2,	RUNOFF_COMMAND+80	0802	
			AA	9F	002C7	PUSHAB	P.AAK		0806	
			01	FB	002CA	CALLS	#1,	CLISPRESENT		
00000000G	00		50	D0	002D1	MOVL	RO,	TEMP		
	6E		50	D1	002D4	CMPL	RO,	#CLIS_ABSENT	0811	
00000000G	8F		09	13	002DB	BEQL	36\$			
			50	D1	002DD	CMPL	RO,	#CLIS_DEFAULTED		
00000000G	8F		04	12	002E4	BNEQ	37\$			
			10	8A	002E6	BICB2	#16,	RUNOFF_COMMAND+80	0812	
00000000G	8F		50	D1	002EA	CMPL	RO,	#CLIS_NEGATED	0814	
			17	13	002F1	BEQL	38\$			
		5B	50	D1	002F3	CMPL	RO,	R11		
			12	13	002F6	BEQL	38\$			
00000000G	8F		50	D1	002F8	CMPL	RO,	#CLIS_LOCPRES		
			09	13	002FF	BEQL	38\$			
00000000G	8F		50	D1	00301	CMPL	RO,	#CLIS_LOCNEG		
			04	12	00308	BNEQ	39\$			
	04	A9	10	88	0030A	BISB2	#16,	RUNOFF_COMMAND+80	0815	
00000000G	8F		50	D1	0030E	CMPL	RO,	#CLIS_ABSENT	0817	
			12	13	00315	BEQL	40\$			
00000000G	8F		50	D1	00317	CMPL	RO,	#CLIS_NEGATED		
			09	13	0031E	BEQL	40\$			
00000000G	8F		50	D1	00320	CMPL	RO,	#CLIS_LOCNEG		
			04	12	00327	BNEQ	41\$			
			08	8A	00329	BICB2	#8,	RUNOFF_COMMAND+80	0818	
00000000G	8F		50	D1	0032D	CMPL	RO,	#CLIS_DEFAULTED	0820	
			0E	13	00334	BEQL	42\$			
		5B	50	D1	00336	CMPL	RO,	R11		
			09	13	00339	BEQL	42\$			
00000000G	8F		50	D1	0033B	CMPL	RO,	#CLIS_LOCPRES		
			04	12	00342	BNEQ	43\$			
	04	A9	08	88	00344	BISB2	#8,	RUNOFF_COMMAND+80	0821	
			AA	9F	00348	PUSHAB	P.AAM		0830	
			01	FB	0034B	CALLS	#1,	CLISPRESENT		
00000000G	00		50	D0	00352	MOVL	RO,	TEMP		
	6E		8F	AA	00355	BICW2	#268,	RUNOFF_COMMAND+79	0836	
	03	A9	6E	D0	0035B	MOVL	TEMP,	RO	0838	
00000000G	8F		50	D1	0035E	CMPL	RO,	#CLIS_NEGATED	0841	

			17	13	00365	BEQL	44\$				
	5B		50	D1	00367	CMPL	RO,	R11			
			12	13	0036A	BEQL	44\$				
00000000G	8F		50	D1	0036C	CMPL	RO,	#CLIS_LOCPRES			
			09	13	00373	BEQL	44\$				
00000000G	8F		50	D1	00375	CMPL	RO,	#CLIS_LOCNEG			
			04	12	0037C	BNEQ	45\$				
03	A9		08	88	0037E	BISB2	#8,	RUNOFF_COMMAND+79		0842	
00000000G	8F		50	D1	00382	CMPL	RO,	#CLIS_NEGATED		0844	
			09	13	00389	BEQL	46\$				
00000000G	8F		50	D1	0038B	CMPL	RO,	#CLIS_LOCNEG			
			04	12	00392	BNEQ	47\$				
04	A9		01	88	00394	BISB2	#1,	RUNOFF_COMMAND+80		0845	
	5B		50	D1	00398	CMPL	RO,	R11		0847	
			12	13	0039B	BEQL	48\$				
00000000G	8F		50	D1	0039D	CMPL	RO,	#CLIS_LOCPRES			
			09	13	003A4	BEQL	48\$				
00000000G	8F		50	D1	003A6	CMPL	RO,	#CLIS_DEFAULTED			
			04	12	003AD	BNEQ	49\$				
03	A9		04	88	003AF	BISB2	#4,	RUNOFF_COMMAND+79		0849	
00000000G	8F		50	D1	003B3	CMPL	RO,	#CLIS_DEFAULTED		0851	
			0E	13	003BA	BEQL	50\$				
	5B		50	D1	003BC	CMPL	RO,	R11			
			09	13	003BF	BEQL	50\$				
00000000G	8F		50	D1	003C1	CMPL	RO,	#CLIS_LOCPRES			
			0D	12	003C8	BNEQ	51\$				
		BC	A9	9F	003CA	PUSHAB	RUNOFF_COMMAND+8			0852	
		70	AA	9F	003CD	PUSHAB	P.AAO				
00000000G	00		02	FB	003D0	CALLS	#2,	CLISGET_VALUE			
		0084	CA	9F	003D7	PUSHAB	P.AAQ			0856	
00000000G	00		01	FB	003DB	CALLS	#1,	CLISPRESENT			
	6E		50	D0	003E2	MOVL	RO,	TEMP			
00000000G	8F		50	D1	003E5	CMPL	RO,	#CLIS_NEGATED		0861	
			17	13	003EC	BEQL	52\$				
	5B		50	D1	003EE	CMPL	RO,	R11			
			12	13	003F1	BEQL	52\$				
00000000G	8F		50	D1	003F3	CMPL	RO,	#CLIS_LOCPRES			
			09	13	003FA	BEQL	52\$				
00000000G	8F		50	D1	003FC	CMPL	RO,	#CLIS_LOCNEG			
			04	12	00403	BNEQ	53\$				
02	A9		02	88	00405	BISB2	#2,	RUNOFF_COMMAND+78		0862	
00000000G	8F		50	D1	00409	CMPL	RO,	#CLIS_DEFAULTED		0864	
			0E	13	00410	BEQL	54\$				
	5B		50	D1	00412	CMPL	RO,	R11			
			09	13	00415	BEQL	54\$				
00000000G	8F		50	D1	00417	CMPL	RO,	#CLIS_LOCPRES			
			04	12	0041E	BNEQ	55\$				
02	A9		01	88	00420	BISB2	#1,	RUNOFF_COMMAND+78		0865	
00000000G	8F		50	D1	00424	CMPL	RO,	#CLIS_DEFAULTED		0867	
			20	13	0042B	BEQL	56\$				
00000000G	8F		50	D1	0042D	CMPL	RO,	#CLIS_NEGATED			
			17	13	00434	BEQL	56\$				
	5B		50	D1	00436	CMPL	RO,	R11			
			12	13	00439	BEQL	56\$				
00000000G	8F		50	D1	0043B	CMPL	RO,	#CLIS_LOCPRES			
			09	13	00442	BEQL	56\$				
00000000G	8F		50	D1	00444	CMPL	RO,	#CLIS_LOCNEG			

00000000G	00	C4	0E	12	0044B	BNEQ	57\$				
		0098	A9	9F	0044D	PUSHAB	RUNOFF_COMMAND+16				0868
			CA	9F	00450	PUSHAB	P.AAS				
00000000G	00		02	FB	00454	CALLS	#2, CLISGET_VALUE				
		00AC	CA	9F	0045B	PUSHAB	P.AAU				0877
00000000G	00		01	FB	0045F	CALLS	#1, CLISPRESENT				
	6E		50	D0	00466	MOVL	R0, TEMP				
	52		6E	D0	00469	MOVL	TEMP, R2				0879
00000000G	8F		52	D1	0046C	CMPL	R2, #CLIS_ABSENT				0882
			09	13	00473	BEQL	58\$				
00000000G	8F		52	D1	00475	CMPL	R2, #CLIS_DEFAULTED				
			03	12	0047C	BNEQ	59\$				
	69		20	8A	0047E	BICB2	#32, RUNOFF_COMMAND+76				0883
00000000G	8F		52	D1	00481	CMPL	R2, #CLIS_NEGATED				0885
			17	13	00488	BEQL	60\$				
	5B		52	D1	0048A	CMPL	R2, R11				
			12	13	0048D	BEQL	60\$				
00000000G	8F		52	D1	0048F	CMPL	R2, #CLIS_LOCPRES				
			09	13	00496	BEQL	60\$				
00000000G	8F		52	D1	00498	CMPL	R2, #CLIS_LOCNEG				
			03	12	0049F	BNEQ	61\$				
	69		20	88	004A1	BISB2	#32, RUNOFF_COMMAND+76				0886
00000000G	8F		52	D1	004A4	CMPL	R2, #CLIS_ABSENT				0888
			04	12	004AB	BNEQ	62\$				
	08	A9	01	8E	004AD	MNEGB	#1, RUNOFF_COMMAND+84				0889
00000000G	8F		52	D1	004B1	CMPL	R2, #CLIS_NEGATED				0891
			09	13	004B8	BEQL	63\$				
00000000G	8F		52	D1	004BA	CMPL	R2, #CLIS_LOCNEG				
			06	12	004C1	BNEQ	64\$				
		08	A9	94	004C3	CLRB	RUNOFF_COMMAND+84				0893
	69		10	8A	004C6	BICB2	#16, RUNOFF_COMMAND+76				0894
00000000G	8F		52	D1	004C9	CMPL	R2, #CLIS_DEFAULTED				0897
			0E	13	004D0	BEQL	65\$				
	5B		52	D1	004D2	CMPL	R2, R11				
			09	13	004D5	BEQL	65\$				
00000000G	8F		52	D1	004D7	CMPL	R2, #CLIS_LOCPRES				
			49	12	004DE	BNEQ	68\$				
	69		10	88	004E0	BISB2	#16, RUNOFF_COMMAND+76				0899
	18	A9	04	D0	004E3	MOVL	#4, SWCH_ID				0900
		F8	AD	9F	004E7	PUSHAB	DESC				0901
		00CO	CA	9F	004EA	PUSHAB	P.AAW				
00000000G	00		02	FB	004EE	CALLS	#2, CLISGET_VALUE				
	2C		50	E9	004F5	BLBC	R0, 67\$				
	01	F8	AD	B1	004F8	CMPL	DESC, #1				0902
			07	12	004FC	BNEQ	66\$				
	08	A9	BD	90	004FE	MOVB	@DESC+4, RUNOFF_COMMAND+84				0903
			24	11	00503	BRB	68\$				
		00000000'	EF	9F	00505	PUSHAB	CHARS_KEYS				0906
		00000000'	EF	9F	0050B	PUSHAB	CHARS_STATES				
		F8	AD	9F	00511	PUSHAB	DESC				
00000000V	EF		03	FB	00514	CALLS	#3, CALL_TPARSE				
	56		50	D0	0051B	MOVL	R0, STATUS				
	08		56	E8	0051E	BLBS	STATUS, 68\$				
			05C9	31	00521	BRW	126\$				0909
	08	A9	7C	8F	90	00524	MOVB	#124, RUNOFF_COMMAND+84			0915
00000000G	8F		52	D1	00529	CMPL	R2, #CLIS_DEFAULTED				0918
			20	13	00530	BEQL	69\$				

00000000G	8F		52	D1	00532	CMPL	R2	#CLIS_NEGATED		
			17	13	00539	BEQL	69\$			
	5B		52	D1	0053B	CMPL	R2	R11		
			12	13	0053E	BEQL	69\$			
00000000G	8F		52	D1	00540	CMPL	R2	#CLIS_LOCPRES		
			09	13	00547	BEQL	69\$			
00000000G	8F		52	D1	00549	CMPL	R2	#CLIS_LOCNEG		
			0C	12	00550	BNEQ	70\$			
	69	40	8F	88	00552	BISB2	#64	RUNOFF_COMMAND+76	0920	
		08	A9	95	00556	TSTB		RUNOFF_COMMAND+84	0921	
			03	12	00559	BNEQ	70\$			
	69		10	8A	0055B	BICB2	#16	RUNOFF_COMMAND+76	0922	
		00DC	CA	9F	0055E	PUSHAB	P.AAY		0927	
00000000G	00		01	FB	00562	CALLS	#1	CLISPRESENT		
	6E		50	D0	00569	MOVL	RO	TEMP		
	04	0860	8F	AA	0056C	BICW2	#2144	RUNOFF_COMMAND+80	0933	
			6E	D0	00572	MOVL	TEMP	RO	0935	
00000000G	8F		50	D1	00575	CMPL	RO	#CLIS_NEGATED	0938	
			17	13	0057C	BEQL	71\$			
	5B		50	D1	0057E	CMPL	RO	R11		
			12	13	00581	BEQL	71\$			
00000000G	8F		50	D1	00583	CMPL	RO	#CLIS_LOCPRES		
			09	13	0058A	BEQL	71\$			
00000000G	8F		50	D1	0058C	CMPL	RO	#CLIS_LOCNEG		
			05	12	00593	BNEQ	72\$			
	04	40	8F	88	00595	BISB2	#64	RUNOFF_COMMAND+80	0939	
00000000G	8F		50	D1	0059A	CMPL	RO	#CLIS_DEFAULTED	0941	
			0E	13	005A1	BEQL	73\$			
			5B	50	D1	005A3	CMPL	RO	R11	
			09	13	005A6	BEQL	73\$			
00000000G	8F		50	D1	005A8	CMPL	PO	#CLIS_LOCPRES		
			4C	12	005AF	BNEQ	76\$			
	04	0820	8F	A8	005B1	BISW2	#2080	RUNOFF_COMMAND+80	0944	
		F8	AD	9F	005B7	PUSHAB	DESC		0945	
		00F8	CA	9F	005BA	PUSHAB	P.ABA			
00000000G	00		02	FB	005BE	CALLS	#2	CLISGET_VALUE		
	30		50	E9	005C5	BLBC	RO	75\$		
	01	F8	AD	B1	005C8	CMPL	DESC	#1	0946	
			07	12	005CC	BNEQ	74\$			
	09	A9	BD	90	005CE	MOVB	@DESC+4	RUNOFF_COMMAND+85	0947	
			28	11	005D3	BRB	76\$			
	18	A9	19	D0	005D5	MOVL	#25	SWCH_ID	0950	
		00000000'	EF	9F	005D9	PUSHAB	CHARS_KEYS		0951	
		00000000'	EF	9F	005DF	PUSHAB	CHARS_STATES			
		F8	AD	9F	005E5	PUSHAB	DESC			
00000000V	EF		03	FB	005E8	CALLS	#3	CALL_TPARSE		
	56		50	D0	005EF	MOVL	RO	STATUS		
	08		56	E8	005F2	BLBS	STATUS	76\$		
			04F5	31	005F5	BRW	126\$		0954	
	09	A9	8F	90	005F8	MOVB	#95	RUNOFF_COMMAND+85	0960	
		0114	CA	9F	005FD	PUSHAB	P.ABC		0966	
00000000G	00		01	FB	00601	CALLS	#1	CLISPRESENT		
	6E		50	D0	00608	MOVL	RO	TEMP		
	04	0180	8F	AA	0060B	BICW2	#384	RUNOFF_COMMAND+80	0971	
			6E	D0	00611	MOVL	TEMP	RO	0974	
00000000G	8F		50	D1	00614	CMPL	RO	#CLIS_NEGATED	0977	
			17	13	0061B	BEQL	77\$			

000

: R

	5B		50	D1	0061D		CPL	R0	R11		
			12	13	00620		BEQL	77\$			
00000000G	8F		50	D1	00622		CPL	R0	#CLIS_LOCPRES		
			09	13	00629		BEQL	77\$			
00000000G	8F		50	D1	0062B		CPL	R0	#CLIS_LOCNEG		
			04	12	00632		BNEQ	78\$			
05	A9		01	88	00634	77\$:	BISB2	#1,	RUNOFF_COMMAND+81		0978
00000000G	8F		50	D1	00638	78\$:	CPL	R0	#CLIS_DEFAULTED		0980
			0E	13	0063F		BEQL	79\$			
	5B		50	D1	00641		CPL	R0	R11		
			09	13	00644		BEQL	79\$			
00000000G	8F		50	D1	00646		CPL	R0	#CLIS_LOCPRES		
			4B	12	0064D		BNEQ	82\$			
04	A9	08A0	8F	A8	0064F	79\$:	BISW2	#2208,	RUNOFF_COMMAND+80		0984
		F8	AD	9F	00655		PUSHAB	DESC			0985
		0130	CA	9F	00658		PUSHAB	P.ABE			
00000000G	00		02	FB	0065C		CALLS	#2,	CLISGET_VALUE		
	30		50	E9	00663		BLBC	R0,	81\$		
	01	F8	AD	B1	00666		CMPW	DESC,	#1		0986
			07	12	0066A		BNEQ	80\$			
09	A9	FC	BD	90	0066C		MOVB	@DESC+4,	RUNOFF_COMMAND+85		0987
			27	11	00671		BRB	82\$			
18	A9		15	D0	00673	80\$:	MOVL	#21,	SWCH_ID		0990
		00000000'	EF	9F	00677		PUSHAB	CHARS_KEYS			0991
		00000000'	EF	9F	0067D		PUSHAB	CHARS_STATES			
		F8	AD	9F	00683		PUSHAB	DESC			
00000000V	EF		03	FB	00686		CALLS	#3,	CALL_TPARSE		
	56		50	D0	0068D		MOVL	R0,	STATOS		
	07		56	E8	00690		BLBS	STATUS,	82\$		
			0457	31	00693		BRW	126\$			0994
09	A9		2D	90	00696	81\$:	MOVW	#45,	RUNOFF_COMMAND+85		1000
			CA	9F	0069A	82\$:	PUSHAB	P.ABG			1005
00000000G	00	014C	01	FB	0069E		CALLS	#1,	CLISPRESENT		
	6E		50	D0	006A5		MOVL	R0,	TEMP		
05	A9		06	8A	006A8		BICB2	#6,	RUNOFF_COMMAND+81		1011
	50		6E	D0	006AC		MOVL	TEMP,	R0		1013
00000000G	8F		50	D1	006AF		CPL	R0	#CLIS_NEGATED		1016
			17	13	006B6		BEQL	83\$			
	5B		50	D1	006B8		CPL	R0	R11		
			12	13	006BB		BEQL	83\$			
00000000G	8F		50	D1	006BD		CPL	R0	#CLIS_LOCPRES		
			09	13	006C4		BEQL	83\$			
00000000G	8F		50	D1	006C6		CPL	R0	#CLIS_LOCNEG		
			04	12	006CD		BNEQ	84\$			
05	A9		04	88	006CF	83\$:	BISB2	#4,	RUNOFF_COMMAND+81		1017
00000000G	8F		50	D1	006D3	84\$:	CPL	R0	#CLIS_DEFAULTED		1019
			0E	13	006DA		BEQL	85\$			
	5B		50	D1	006DC		CPL	R0	R11		
			09	13	006DF		BEQL	85\$			
00000000G	8F		50	D1	006E1		CPL	R0	#CLIS_LOCPRES		
			4B	12	006E8		BNEQ	88\$			
04	A9	0A20	8F	AB	006EA	85\$:	BISW2	#2592,	RUNOFF_COMMAND+80		1023
		F8	AD	9F	006F0		PUSHAB	DESC			1024
		0168	CA	9F	006F3		PUSHAB	P.ABI			
00000000G	00		02	FB	006F7		CALLS	#2,	CLISGET_VALUE		
	30		50	E9	006FE		BLBC	R0,	87\$		
	01	F8	AD	B1	00701		CMPW	DESC,	#1		1025

09	A9	FC	07	12	00705	BNEQ	86\$		
			BD	90	00707	MOVB	@DESC+4, RUNOFF_COMMAND+85		1026
18	A9		27	11	0070C	BRB	88\$		
			19	D0	0070E	MOVL	#25, SWCH_ID		1029
		00000000'	EF	9F	00712	PUSHAB	CHARS_KEYS		1030
		00000000'	EF	9F	00718	PUSHAB	CHARS_STATES		
		F8	AD	9F	0071E	PUSHAB	DESC		
00000000V	EF		03	FB	00721	CALLS	#3, CALL_TPARSE		
	56		50	D0	00728	MOVL	R0, STATUS		
	07		56	E8	0072B	BLBS	STATUS, 88\$		
			03BC	31	0072E	BRW	126\$		1033
09	A9		07	90	00731	MOVB	#7, RUNOFF_COMMAND+85		1039
FF	8F		A9	91	00735	CMPB	RUNOFF_COMMAND+85, #255		1044
			05	12	0073A	BNEQ	89\$		
			09	A9	94	0073C	CLRB	RUNOFF_COMMAND+85	1046
			04	11	0073F	BRB	90\$		
05	A9		08	88	00741	BISB2	#8, RUNOFF_COMMAND+81		1048
		0178	CA	9F	00745	PUSHAB	P.ABK		1055
00000000G	00		01	FB	00749	CALLS	#1, CLISPRESNT		
	6E		50	D0	00750	MOVL	R0, TEMP		
00000000G	8F		50	D1	00753	CMPL	R0, #CLIS_DEFAULTED		1061
			0E	13	0075A	BEQL	91\$		
	5B		50	D1	0075C	CMPL	R0, R11		
00000000G	8F		09	13	0075F	BEQL	91\$		
			50	D1	00761	CMPL	R0, #CLIS_LOCPRES		
			0E	12	00768	BNEQ	92\$		
		E4	A9	9F	0076A	PUSHAB	RUNOFF_COMMAND+48		1062
00000000G	00	0188	CA	9F	0076D	PUSHAB	P.ABM		
			02	FB	00771	CALLS	#2, CLISGET_VALUE		
00000000G	00		01	FB	0077C	PUSHAB	P.ABO		1066
	6E		50	D0	00783	CALLS	#1, CLISPRESNT		
00000000G	8F		50	D1	00786	MOVL	R0, TEMP		
			50	D1	00788	CMPL	R0, #CLIS_DEFAULTED		1072
	5B		0E	13	0078D	BEQL	93\$		
00000000G	8F		50	D1	0078F	CMPL	R0, R11		
			09	13	00792	BEQL	93\$		
			50	D1	00794	CMPL	R0, #CLIS_LOCPRES		
			0E	12	0079B	BNEQ	94\$		
		EC	A9	9F	0079D	PUSHAB	RUNOFF_COMMAND+56		1073
00000000G	00	01A8	CA	9F	007A0	PUSHAB	P.ABQ		
			02	FB	007A4	CALLS	#2, CLISGET_VALUE		
00000000G	00		01	FB	007AB	PUSHAB	P.ABS		1080
	6E		50	D0	007AF	CALLS	#1, CLISPRESNT		
00000000G	8F		50	D1	007B6	MOVL	R0, TEMP		
			50	D1	007B9	CMPL	R0, #CLIS_ABSENT		1085
			04	12	007C0	BNEQ	95\$		
0A	A9		01	B0	007C2	MOVW	#1, RUNOFF_COMMAND+86		1086
00000000G	8F		50	D1	007C6	CMPL	R0, #CLIS_NEGATED		1088
			09	13	007CD	BEQL	96\$		
00000000G	8F		50	D1	007CF	CMPL	R0, #CLIS_LOCNEG		
			03	12	007D6	BNEQ	97\$		
		0A	A9	B4	007D8	CLRW	RUNOFF_COMMAND+86		1089
00000000G	8F		50	D1	007DB	CMPL	R0, #CLIS_DEFAULTED		1091
			0E	13	007E2	BEQL	98\$		
	5B		50	D1	007E4	CMPL	R0, R11		
00000000G	8F		09	13	007E7	BEQL	98\$		
			50	D1	007E9	CMPL	R0, #CLIS_LOCPRES		

0000000G	00		00	FB	008D2	CALLS	#0, LIB\$LP_LINES	1134
14	A9		50	B0	008D9	MOVW	R0, RUNOFF_COMMAND+96	
0000000G	8F		52	D1	008DD	108\$:	C MPL R2, #CLIS_NEGATED	1136
			09	13	008E4		BEQL 109\$	
0000000G	8F		52	D1	008E6		C MPL R2, #CLIS_LOCNEG	
			03	12	008ED		BNEQ 110\$	
		14	A9	B4	008EF	109\$:	CLR W RUNOFF_COMMAND+96	1137
0000000G	8F		52	D1	008F2	110\$:	C MPL R2, #CCIS_DEFAULTED	1139
			0E	13	008F9		BEQL 111\$	
			52	D1	008FB		C MPL R2, R11	
			09	13	0C8FE		BEQL 111\$	
0000000G	8F		52	D1	00900		C MPL R2, #CLIS_LOCPRES	
			3C	12	00907		BNEQ 113\$	
		F8	AD	9F	00909	111\$:	PUSHAB DESC	1140
		0200	CA	9F	0090C		PUSHAB P.ACC	
0000000G	00		02	FB	00910		CALLS #2, CLISGET_VALUE	
	19		50	E9	00917		BLBC R0, 112\$	
			5E	DD	0091A		PUSHL SP	1142
		FC	AD	DD	0091C		PUSHL DESC+4	1143
		F8	AD	3C	0091F		MOVZWL DESC, -(SP)	1142
0000000G	00		03	FB	00923		CALLS #3, LIB\$CVT_DTB	
	56		50	D0	0092A		MOVL R0, STATUS	
14	A9		6E	B0	0092D		MOVW TEMP, RUNOFF_COMMAND+96	1145
			12	11	00931		BRB 113\$	
0000000G	00		00	FB	00933	112\$:	CALLS #0, LIB\$LP_LINES	1149
	14		50	B0	0093A		MOVW R0, RUNOFF_COMMAND+96	
			58	D0	0093E		MOVL #CLIS_ABSENT, FORM_QUALIFIER	1150
		00000000G	CA	9F	00945	113\$:	PUSHAB P.ACC	1154
		0210	01	FB	00949		CALLS #1, CLISPRESENT	
0000000G	00		50	D0	00950		MOVL R0, TEMP	
	6E		6E	D0	00953		MOVL TEMP, R2	1156
0000000G	8F		52	D1	00956		C MPL R2, #CLIS_ABSENT	1159
			12	13	0095D		BEQL 114\$	
0000000G	8F		52	D1	0095F		C MPL R2, #CLIS_NEGATED	
			09	13	00966		BEQL 114\$	
0000000G	8F		52	D1	00968		C MPL R2, #CLIS_LOCNEG	
			03	12	0096F		BNEQ 115\$	
		12	A9	B4	00971	114\$:	CLR W RUNOFF_COMMAND+94	1160
0000000G	8F		52	D1	00974	115\$:	C MPL R2, #CCIS_DEFAULTED	1162
			0E	13	0097B		BEQL 116\$	
			52	D1	0097D		C MPL R2, R11	
			09	13	00980		BEQL 116\$	
0000000G	8F		52	D1	00982		C MPL R2, #CLIS_LOCPRES	
			2E	12	00989		BNEQ 118\$	
		F8	AD	9F	0098B	116\$:	PUSHAB DESC	1163
		0220	CA	9F	0098E		PUSHAB P.ACC	
0000000G	00		02	FB	00992		CALLS #2, CLISGET_VALUE	
	19		50	E9	00999		BLBC R0, 117\$	
			5E	DD	0099C		PUSHL SP	1165
		FC	AD	DD	0099E		PUSHL DESC+4	1166
		F8	AD	3C	009A1		MOVZWL DESC, -(SP)	1165
0000000G	00		03	FB	009A5		CALLS #3, LIB\$CVT_DTB	
	56		50	D0	009AC		MOVL R0, STATUS	
12	A9		6E	B0	009AF		MOVW TEMP, RUNOFF_COMMAND+94	1168
			04	11	009B3		BRB 118\$	
			05	B0	009B5	117\$:	MOVW #5, RUNOFF_COMMAND+94	1171
0000000G	8F		52	D1	009B9	118\$:	C MPL R2, #CLIS_NEGATED	1173

00000000G	8F		17	13	009C0	BEQL	119\$		
			52	D1	009C2	CMPL	R2, #CLIS_LOCNEG		
	5B		0E	13	009C9	BEQL	119\$		
			52	D1	009CB	CMPL	R2, R11		
			09	13	009CE	BEQL	119\$		
00000000G	8F		52	D1	009D0	CMPL	R2, #CLIS_LOCPRES		
	06	A9	04	12	009D7	BNEQ	120\$		
			20	88	009D9	BISB2	#32, RUNOFF_COMMAND+82	1174	
		0230	CA	9F	009DD	PUSHAB	P.ACI	1183	
00000000G	00		01	FB	009E1	CALLS	#1, CLISPRESENT		
	6E		50	D0	009E8	MOVL	RO, TEMP		
	01	A9	7E	8F	8A	BICB2	#126, RUNOFF_COMMAND+77	1192	
			50	D0	009F0	MOVL	TEMP, RO	1194	
00000000G	8F		50	D1	009F3	CMPL	RO, #CLIS_DEFAULTED	1197	
			0E	13	009FA	BEQL	121\$		
	5B		50	D1	009FC	CMPL	RO, R11		
			09	13	009FF	BEQL	121\$		
00000000G	8F		50	D1	00A01	CMPL	RO, #CLIS_LOCPRES		
			29	12	00A08	BNEQ	122\$		
		F8	AD	9F	00A0A	PUSHAB	DESC	1198	
		0240	CA	9F	00A0D	PUSHAB	P.ACK		
00000000G	00		02	FB	00A11	CALLS	#2, CLISGET_VALUE		
	18		50	E9	00A18	BLBC	RO, 122\$		
		00000000'	EF	9F	00A1B	PUSHAB	DEBUG_KEYS	1199	
		00000000'	EF	9F	00A21	PUSHAB	DEBUG_STATES		
		F8	AD	9F	00A27	PUSHAB	DESC		
00000000V	EF		03	FB	00A2A	CALLS	#3, CALL_TPARSE		
			D7	11	00A31	BRB	121\$		
		0250	CA	9F	00A33	PUSHAB	P.ACM	1203	
00000000G	00		01	FB	00A37	CALLS	#1, CLISPRESENT		
	6E		50	D0	00A3E	MOVL	RO, TEMP		
	02	A9	03C0	8F	AA	BICW2	#960, RUNOFF_COMMAND+78	1207	
	06	A9		0F	88	BISB2	#15, RUNOFF_COMMAND+82	1211	
			50	D0	00A4B	MOVL	TEMP, RO	1213	
00000000G	8F		50	D1	00A4E	CMPL	RO, #CLIS_DEFAULTED	1216	
			0E	13	00A55	BEQL	123\$		
	5B		50	D1	00A57	CMPL	RO, R11		
			09	13	00A5A	BEQL	123\$		
00000000G	8F		50	D1	00A5C	CMPL	RO, #CLIS_LOCPRES		
			2F	12	00A63	BNEQ	124\$		
		F8	AD	9F	00A65	PUSHAB	DESC	1217	
		0260	CA	9F	00A68	PUSHAB	P.ACO		
00000000G	00		02	FB	00A6C	CALLS	#2, CLISGET_VALUE		
	1E		50	E9	00A73	BLBC	RO, 124\$		
		00000000'	EF	9F	00A76	PUSHAB	DEVICE_KEYS	1219	
		00000000'	EF	9F	00A7C	PUSHAB	DEVICE_STATES		
		F8	AD	9F	00A82	PUSHAB	DESC		
00000000V	EF		03	FB	00A85	CALLS	#3, CALL_TPARSE		
	56		50	D0	00A8C	MOVL	RO, STATUS		
	D3		56	E8	00A8F	BLBS	STATUS, 123\$		
			59	11	00A92	BRB	126\$	1222	
		0270	CA	9F	00A94	PUSHAB	P.ACO	1237	
00000000G	00		01	FB	00A98	CALLS	#1, CLISPRESENT		
	6E		50	D0	00A9F	MOVL	RO, TEMP		
	02	A9	30	8A	00AA2	BICB2	#48, RUNOFF_COMMAND+78	1242	
			50	D0	00AA6	MOVL	TEMP, RO	1244	
00000000G	8F		50	D1	00AA9	CMPL	RO, #CLIS_DEFAULTED	1247	

				0E 13 00AB0	BEQL	125\$		
			5B	50 D1 00AB2	CMPL	R0, R11		
				09 13 00AB5	BEQL	125\$		
			00000000G	8F 50 D1 00AB7	CMPL	R0, #CLIS_LOCPRES		
				35 12 00ABE	BNEQ	127\$		
				F8 AD 9F 00AC0 125\$:	PUSHAB	DESC	1248	
				0280 CA 9F 00AC3	PUSHAB	P.ACS		
			00000000G	00 02 FB 00AC7	CALLS	#2, CLISGET_VALUE		
				24 50 E9 00ACE	BLBC	R0, 127\$		
				00000000' EF 9F 00AD1	PUSHAB	MESSAGES_KEYS	1249	
				00000000' EF 9F 00AD7	PUSHAB	MESSAGES_STATES		
				F8 AD 9F 00ADD	PUSHAB	DESC		
			00000000V	EF 03 FB 00AE0	CALLS	#3, CALL_TPARSE		
				56 50 D0 00AE7	MOVL	R0, STATUS		
				D3 56 E8 00AEA	BLBS	STATUS, 125\$		
				50 00000000* 8F D0 00AED 126\$:	MOVL	#<<<RUNOFF\$_FACILITY@16>!20>!4>, R0	1252	
					04 00AF4	RET		
				0294 CA 9F 00AF5 127\$:	PUSHAB	P.ACU	1259	
			00000000G	00 01 FB 00AF9	CALLS	#1, CLISPRESENT		
				6E 50 D0 00B00	MOVL	R0, TEMP		
				0C A9 D4 00B03	CLRL	RUNOFF_COMMAND+88	1266	
				50 6E D0 00B06	MOVL	TEMP, R0	1269	
			00000000G	8F 50 D1 00B09	CMPL	R0, #CLIS_DEFAULTED	1272	
					0E 13 00B10	BEQL	128\$	
				5B 50 D1 00B12	CMPL	R0, R11		
					09 13 00B15	BEQL	128\$	
			00000000G	8F 50 D1 00B17	CMPL	R0, #CLIS_LOCPRES		
					29 12 00B1E	BNEQ	129\$	
				F8 AD 9F 00B20 128\$:	PUSHAB	DESC	1273	
				02AB CA 9F 00B23	PUSHAB	P.ACW		
			00000000G	00 02 FB 00B27	CALLS	#2, CLISGET_VALUE		
				18 50 E9 00B2E	BLBC	R0, 129\$		
				00000000' EF 9F 00B31	PUSHAB	DEC_INTERNAL_KEYS	1274	
				00000000' EF 9F 00B37	PUSHAB	DEC_INTERNAL_STATES		
				F8 AD 9F 00B3D	PUSHAB	DESC		
			00000000V	EF 03 FB 00B40	CALLS	#3, CALL_TPARSE		
					D7 11 00B47	BRB	128\$	
				18 1C A9 E9 00B49 129\$:	BLBC	LN01 OPTION USED, 130\$	1282	
04	02	A9		04 06 ED 00B4D	CMPZV	#6, #4, RUNOFF_COMMAND+78, #4	1284	
					10 13 00B53	BEQL	130\$	
05	02	A9		04 06 ED 00B55	CMPZV	#6, #4, RUNOFF_COMMAND+78, #5	1285	
					08 13 00B58	BEQL	130\$	
				50 00000000* 8F D0 00B5D	MOVL	#<<<RUNOFF\$_FACILITY@16>!4832>!4>, R0	1292	
					04 00B64	RET		
				0C AE 02B0 CA 17 28 00B65 130\$:	MOV3	#23, P.ACY, TT_NAME_STRING	1314	
				04 AE 010E0017 8F D0 00B6C	MOVL	#17694743, TT_NAME_DESC	1318	
				08 AE 0C AE 9E 00B74	MOVAB	TT_NAME_STRING, TT_NAME_DESC+4	1321	
				F0 AD 010E0100 8F D0 00B79	MOVL	#17694976, TT_BUF_DESC	1327	
				F4 AD 0C AE 9E 00B81	MOVAB	TT_NAME_STRING, TT_BUF_DESC+4	1330	
				50 0C A9 32 00B86	CVTDL	RUNOFF_COMMAND+88, FOO	1339	
04				50 0D E1 00B8A	BBC	#13, FOO, 131\$	1341	
					50 D4 00B8E	CLRL	LOGTAB	1342
					03 11 00B90	BRB	132\$	
				50 06 D0 00B92 131\$:	MOVL	#6, LOGTAB	1344	
					50 DD 00B95 132\$:	PUSHL	LOGTAB	1353
					7E 7C 00B97	CLRL	-(SP)	
				F0 AD 9F 00B99	PUSHAB	TT_BUF_DESC		

		14	AE	9F	00B9C	PUSHAB	TT_NAME_DESC	
		18	AE	9F	00B9F	PUSHAB	TT_NAME_DESC	
00000000G	00		06	FB	00BA2	CALLS	#6, SYS\$TRNLOG	
	01		50	D1	00BA9	CMPL	TRNLOG_STATUS, #1	1356
			3B	12	00BAC	BNEQ	135\$	
	52		01	D0	00BAE	MOVL	#1, I	1369
			06	DD	00BB1	PUSHL	#6	1375
			7E	7C	00BB3	CLRQ	-(SP)	
		F0	AD	9F	00BB5	PUSHAB	TT_BUF_DESC	
		14	AE	9F	00BB8	PUSHAB	TT_NAME_DESC	
		18	AE	9F	00BBB	PUSHAB	TT_NAME_DESC	
00000000G	00		06	FB	00BBE	CALLS	#6, SYS\$TRNLOG	
00000629	8F		50	D1	00BC5	CMPL	TRNLOG_STATUS, #1577	1378
			04	13	00BCC	BEQL	134\$	
DF	52		09	F3	00BCE	AOBLEQ	#9, I, 133\$	1371
		00000000G	EF	9F	00BD2	PUSHAB	STR\$FAILURE	1383
			7E	D4	00BD8	CLRL	-(SP)	
		CC	A9	9F	00BDA	PUSHAB	\$STR\$TARGET	
		10	AE	9F	00BDD	PUSHAB	\$STR\$STRING	
			7E	D4	00BE0	CLRL	-(SP)	
00000000G	EF		05	FB	00BE2	CALLS	#5, XST\$COPY	
		B4	A9	9F	00BE9	PUSHAB	RUNOFF_COMMAND	1393
00000000V	EF		01	FB	00BEC	CALLS	#1, RNODAT	
		B4	A9	9F	00BF3	PUSHAB	RUNOFF_COMMAND	1396
00000000G	EF		01	FB	00BF6	CALLS	#1, RUNOFF	
	56		50	D0	00BFD	MOVL	R0, STATUS	
	03		56	E8	00C00	BLBS	STATUS, 136\$	1399
	57		56	D0	00C03	MOVL	STATUS, EXIT_STATUS	1401
		F4	2B	31	00C06	BRW	1\$	0642
	50	00000000G	00	9E	00C09	MOVAB	LIB\$FIND_FILE_END, R0	1408
			0D	13	00C10	BEQL	138\$	
		00000000G	EF	9F	00C12	PUSHAB	LFFCTX	1409
00000000G	00		01	FB	00C18	CALLS	#1, LIB\$FIND_FILE_END	
		20	A9	9F	00C1F	PUSHAB	WS_LIMITS	1413
00000000G	00		01	FB	00C22	CALLS	#1, SY\$PURGWS	
	50		57	D0	00C29	MOVL	EXIT_STATUS, R0	1417
			04	00C2C	RET			1419

; Routine Size: 3117 bytes, Routine Base: \$CODE\$ + 0000

```

: 1305 1420 1 %SBTTL 'CALL_TPARSE -- Routine to call LIB$TPARSE'
: 1306 1421 1 ROUTINE call_tparse(string,states,keys) =
: 1307 1422 1
: 1308 1423 1 ++
: 1309 1424 1
: 1310 1425 1 FUNCTIONAL DESCRIPTION:
: 1311 1426 1
: 1312 1427 1 This routine calls tparse given the string, states, and keys
: 1313 1428 1
: 1314 1429 1 FORMAL PARAMETERS:
: 1315 1430 1
: 1316 1431 1 string Address of string descriptor for string to parse
: 1317 1432 1 states Address of tparse states table
: 1318 1433 1 keys Address of tparse keys table
: 1319 1434 1
: 1320 1435 1 IMPLICIT INPUTS: none
: 1321 1436 1
: 1322 1437 1 IMPLICIT OUTPUTS: none
: 1323 1438 1
: 1324 1439 1 ROUTINE VALUE:
: 1325 1440 1 COMPLETION CODES: return code of LIB$TPARSE
: 1326 1441 1
: 1327 1442 1 SIDE EFFECTS: none known
: 1328 1443 1
: 1329 1444 1 --
: 1330 1445 2 BEGIN
: 1331 1446 2
: 1332 1447 2 MAP string : REF BBLOCK;
: 1333 1448 2 LOCAL
: 1334 1449 2 status,
: 1335 1450 2 tpablock : BBLOCK[tpa$k_length0];
: 1336 1451 2
: 1337 1452 2 Initialize the TPARSE block:
: 1338 1453 2
: 1339 1454 2 CH$FILL(0,tpa$k_length0,tpablock);
: 1340 1455 2 tpablock[tpa$l_count] = tpa$k_count0;
: 1341 1456 2 tpablock[tpa$l_options] = tpa$m_abbrev;
: 1342 1457 2 tpablock[tpa$l_stringcnt] = .string[dsc$w_length];
: 1343 1458 2 tpablock[tpa$l_stringptr] = .string[dsc$a_pointer];
: 1344 1459 2
: 1345 1460 2 Invoke TPARSE and return with its return code:
: 1346 1461 2
: 1347 1462 2 RETURN LIB$TPARSE(tpablock,.states,.keys)
: 1348 1463 1 END;

```

				003C 0000 CALL_TPARSE:		
				.WORD	Save R2,R3,R4,R5	: 1421
				SUBL2	#36, SP	
24	00	5E	24 C2 00002	MOVCS	#0, (SP), #0, #36, TPABLOCK	: 1454
		6E	00 2C 00005			
			6E 0000A			
		6E	08 D0 0000B	MOVL	#8, TPABLOCK	: 1455
	04	AE	02 D0 0000E	MOVL	#2, TPABLOCK+4	: 1456
		50	04 AC D0 00012	MOVL	STRING, R0	: 1457

RNOVMS
V04-000

RNOVMS - DSR VMS Command Line Interface
CALL_TPARSE -- Routine to call LIB\$TPARSE

E 8
16-Sep-1984 01:36:17
14-Sep-1984 13:07:54

VAX-11 Bliss-32 V4.0-742
[RUNOFF.SRC]RNOVMS.B32;1

Page 59
(16)

RN
V3

08	AE	60	3C	00016
0C	AE	A0	D0	0001A
	7E	08	AC	7D 0001F
		08	AE	9F 00023
00000000G	00	03	FB	00026
		04	00	0002D

MOVZWL	(R0),	TPABLOCK+8
MOVL	4(R0),	TPABLOCK+12
MOVQ	STATES,	-(SP)
PUSHAB	TPABLOCK	
CALLS	#3,	LIB\$TPARSE
RET		

:
: 1458
: 1462
:
: 1463

: Routine Size: 46 bytes, Routine Base: \$CODE\$ + 0C2D

: 1349 1464 1

```

: 1351 1465 1 %SBTTL 'SAVE_VALUE -- store values decided by $TPARSE'
: 1352 1466 1 ROUTINE save_value =
: 1353 1467 2 BEGIN
: 1354 1468 2 BUILTIN AP:
: 1355 1469 2 MAP AP : REF BBLOCK;
: 1356 1470 2
: 1357 1471 2 SELECTONE .ap[tpa$l_param] OF
: 1358 1472 2 SET
: 1359 1473 2
: 1360 1474 2 [sv_di_dbg1] : runoff_command[rno$h_dbg1] = .AP[TPA$L_NUMBER];
: 1361 1475 2
: 1362 1476 2 [sv_di_dbg2] : runoff_command[rno$h_dbg2] = .AP[TPA$L_NUMBER];
: 1363 1477 2
: 1364 1478 2 [sv_byte_value] : BEGIN
: 1365 1479 2
: 1366 1480 2 SELECTONE .swch_id OF
: 1367 1481 2 SET
: 1368 1482 2
: 1369 1483 2 [swch_id_change] :
: 1370 1484 2 runoff_command[rno$c_change] = .AP[TPA$L_NUMBER];
: 1371 1485 2
: 1372 1486 2 [swch_id_underline,swch_id_spacing,swch_id_separate] :
: 1373 1487 2 runoff_command[rno$c_underline] = .AP[TPA$L_NUMBER];
: 1374 1488 2
: 1375 1489 2 [OTHERWISE] : RETURN 0;
: 1376 1490 2
: 1377 1491 2 TES
: 1378 1492 2 END;
: 1379 1493 2
: 1380 1494 2 [sv_dev_landscape] :
: 1381 1495 2 BEGIN
: 1382 1496 2 runoff_command[rno$sv_ln01_port_land] = 0;
: 1383 1497 2 ln01_option_used = true
: 1384 1498 2 END;
: 1385 1499 2
: 1386 1500 2 [sv_dev_portrait] :
: 1387 1501 2 BEGIN
: 1388 1502 2 runoff_command[rno$sv_ln01_port_land] = 1;
: 1389 1503 2 ln01_option_used = true
: 1390 1504 2 END;
: 1391 1505 2
: 1392 1506 2 [sv_dev_underline] :
: 1393 1507 2 BEGIN
: 1394 1508 2 runoff_command[rno$sv_ln01_ital_under] = 0;
: 1395 1509 2 ln01_option_used = true
: 1396 1510 2 END;
: 1397 1511 2
: 1398 1512 2 [sv_dev_italic] :
: 1399 1513 2 BEGIN
: 1400 1514 2 runoff_command[rno$sv_ln01_ital_under] = 1;
: 1401 1515 2 ln01_option_used = true
: 1402 1516 2 END;
: 1403 1517 2
: 1404 1518 2 [sv_dev_header] :
: 1405 1519 2 BEGIN
: 1406 1520 2 runoff_command[rno$sv_ln01_header] = true;
: 1407 1521 2 ln01_option_used = true

```

```

: 1408      1522      2
: 1409      1523      2
: 1410      1524      2
: 1411      1525      2
: 1412      1526      2
: 1413      1527      2
: 1414      1528      2
: 1415      1529      2
: 1416      1530      2
: 1417      1531      2
: 1418      1532      2
: 1419      1533      2
: 1420      1534      2
: 1421      1535      2
: 1422      1536      2
: 1423      1537      2
: 1424      1538      2
: 1425      1539      2
: 1426      1540      2
: 1427      1541      2
: 1428      1542      2
: 1429      1543      2
: 1430      1544      2
: 1431      1545      2
: 1432      1546      2
: 1433      1547      2
: 1434      1548      2
: 1435      1549      2
: 1436      1550      2
: 1437      1551      2
: 1438      1552      2
: 1439      1553      1

```

```

      END;
[sv_dev_noheader] :
      BEGIN
      runoff_command[rno$vn01_header] = false;
      ln01_option_used = true
      END;
[sv_dev_load] :
      BEGIN
      runoff_command[rno$vn01_load] = true;
      ln01_option_used = true
      END;
[sv_dev_noload] :
      BEGIN
      runoff_command[rno$vn01_load] = false;
      ln01_option_used = true
      END;
[sv_output_line_numbers] : ! generate output line numbers in the
                           ! output file.
      runoff_command[rno$h_dbg2] =
      .runoff_command[rno$h_dbg2] + 1^15 ;
[OTHERWISE] : RETURN 0;
TES:
1          ! If we get back to here, return "SUCCESS"
END:      ! End of routine SAVE_VALUE

```

```

0004 0000 SAVE_VALUE:
      .WORD      Save R2
52 00000000' EF 9E 00002      MOVAB      RUNOFF_COMMAND+80, R2      : 1466
51          20 AC D0 00009      MOVL      32(AP), R1      : 1471
0A          51 D1 0000D      CMPL      R1, #10      : 1474
08 A2      1C AC B0 00012      MOVW      28(AP), RUNOFF_COMMAND+88
      35 11 00017      BRB
0B          51 D1 00019 1$:      CMPL      R1, #11      : 1476
      07 12 0001C      BNEQ      2$
0A A2      1C AC B0 0001E      MOVW      28(AP), RUNOFF_COMMAND+90
      29 11 00023      BRB
01          51 D1 00025 2$:      CMPL      R1, #1      : 1478
      26 12 00028      BNEQ      6$
50          14 A2 D0 0002A      MOVL      SWCH_ID, R0      : 1480
04          50 D1 0002E      CMPL      R0, #4      : 1483
      07 12 00031      BNEQ      3$
04 A2      1C AC 90 00033      MOVB      28(AP), RUNOFF_COMMAND+84      : 1484
      7C 11 00038      BRB      16$
15          50 D1 0003A 3$:      CMPL      R0, #21      : 1486

```

			0A 13 0003D	BEQL	4\$		
18			50 D1 0003F	CMPL	R0	#24	
			74 19 00042	BLSS	17\$		
19			50 D1 00044	CMPL	R0	#25	
			6F 14 00047	BGTR	17\$		
05	A2	1C	AC 90 00049	4\$: MOVB	28(AP),	RUNOFF_COMMAND+85	1487
			6B 11 0004E	5\$: BRB	18\$		
05			51 D1 00050	6\$: CMPL	R1,	#5	1494
			06 12 00053	BNEQ	7\$		
02	A2		02 8A 00055	BICB2	#2,	RUNOFF_COMMAND+82	1496
			4B 11 00059	BRB	14\$		1497
08			51 D1 0005B	7\$: CMPL	R1,	#8	1500
			06 12 0005E	BNEQ	8\$		
02	A2		02 88 00060	BISB2	#2,	RUNOFF_COMMAND+82	1502
			40 11 00064	BRB	14\$		1503
09			51 D1 00066	8\$: CMPL	R1,	#9	1506
			06 12 00069	BNEQ	9\$		
02	A2		01 8A 0006B	BICB2	#1,	RUNOFF_COMMAND+82	1508
			35 11 0006F	BRB	14\$		1509
04			51 D1 00071	9\$: CMPL	R1,	#4	1512
			06 12 00074	BNEQ	10\$		
02	A2		01 88 00076	BISB2	#1,	RUNOFF_COMMAND+82	1514
			2A 11 0007A	BRB	14\$		1515
02			51 D1 0007C	10\$: CMPL	R1,	#2	1518
			06 12 0007F	BNEQ	11\$		
02	A2		04 88 00081	BISB2	#4,	RUNOFF_COMMAND+82	1520
			1F 11 00085	BRB	14\$		1521
03			51 D1 00087	11\$: CMPL	R1,	#3	1524
			06 12 0008A	BNEQ	12\$		
02	A2		04 8A 0008C	BICB2	#4,	RUNOFF_COMMAND+82	1526
			14 11 00090	BRB	14\$		1527
06			51 D1 00092	12\$: CMPL	R1,	#6	1530
			06 12 00095	BNEQ	13\$		
02	A2		08 88 00097	BISB2	#8,	RUNOFF_COMMAND+82	1532
			09 11 0009B	BRB	14\$		1533
07			51 D1 0009D	13\$: CMPL	R1,	#7	1536
			0A 12 000A0	BNEQ	15\$		
02	A2		08 8A 000A2	BICB2	#8,	RUNOFF_COMMAND+82	1538
18	A2		01 D0 000A6	14\$: MOVL	#1,	LN01_OPTION_USED	1539
			0F 11 000AA	BRB	18\$		
0C			51 D1 000AC	15\$: CMPL	R1,	#12	1542
			07 12 000AF	BNEQ	17\$		
0B	A2	80	8F 80 000B1	ADDB2	#128,	RUNOFF_COMMAND+90	1545
			03 11 000B6	BRB	18\$		1544
			50 D4 000B8	17\$: CLRL	R0		1547
			04 000BA	RET			
50			01 D0 000BB	18\$: MOVL	#1,	R0	1553
			04 000BE	RET			

; Routine Size: 191 bytes, Routine Base: \$CODE\$ + 0C5B


```

: 1441 1554 1 %SBTTL 'SET_OUTPUT_DEVICE_CODE -- store O/P device type codes determined by $TPARSE'
: 1442 1555 1 ROUTINE set_output_device_code =
: 1443 1556 2 BEGIN
: 1444 1557 2 BUILTIN AP;
: 1445 1558 2 MAP AP : REF BBLOCK;
: 1446 1559 2
: 1447 1560 2 runoff_command[rno$v_4_out_format] = .ap[tpa$l_param];
: 1448 1561 2
: 1449 1562 2 1 ! Make it return 'success'
: 1450 1563 2
: 1451 1564 1 END; ! End of routine SET_OUTPUT_DEVICE_CODE

```

```

00000000' EF 04 06 20 AC FO 0002 .WORD Save nothing
01 DO 000C INSV 32(AP), #6, #4, RUNOFF_COMMAND+78
04 000F RET MOVL #1, R0

```

: 1555
: 1560
: 1564
:

: Routine Size: 16 bytes, Routine Base: \$CODE\$ + 0D1A

```
1453 1565 1 %SBTTL 'RNODAT - Get current system date and time'
1454 1566 1 %IF NEW_RNODAT_MODULE %THEN
1455 1567 1 ROUTINE=rnodat-(runoff_command) : NOVALUE =
1456 1568 1
1457 1569 1 ++
1458 1570 1
1459 1571 1 ABSTRACT: Obtains the system date and time in a standard format.
1460 1572 1
1461 1573 1 This routine has been made a part of RNOVMS.B32 to move all system
1462 1574 1 related logic into one place. It was extracted from RNODAT.BLI which
1463 1575 1 was at IDENT 007 at the time. This work was done by Ray Marshall on
1464 1576 1 August 6, 1982.
1465 1577 1
1466 1578 1 FUNCTIONAL DESCRIPTION:
1467 1579 1
1468 1580 1 This routine obtains the system date and time, breaks it
1469 1581 1 up into its constituents, and returns the results.
1470 1582 1
1471 1583 1 FORMAL PARAMETERS:
1472 1584 1
1473 1585 1 RUNOFF_COMMAND -- the command tables into which the current system
1474 1586 1 date and time will be stuffed.
1475 1587 1
1476 1588 1 IMPLICIT INPUTS: None
1477 1589 1
1478 1590 1 IMPLICIT OUTPUTS: None
1479 1591 1
1480 1592 1 ROUTINE VALUE:
1481 1593 1 COMPLETION CODES: None
1482 1594 1
1483 1595 1 SIDE EFFECTS: None
1484 1596 1
1485 1597 1 --
1486 1598 2 BEGIN
1487 1599 2
1488 1600 2 MAP
1489 1601 2 runoff_command : REF $RNO_CMD; ! Define control block data structure
1490 1602 2
1491 1603 2 LOCAL
1492 1604 2 TIME_32 : VECTOR [7,2];
1493 1605 2
1494 1606 2 SYSS$NUMTIM (TIME_32, 0); !Get binary date as a short-word vector.
1495 1607 2
1496 1608 2 !
1497 1609 2 ! Convert the system's date and time to the structure setup
1498 1610 2 ! within our command tables:
1499 1611 2 runoff_command[RNO$H_YEAR] = .TIME_32[0];
1500 1612 2 runoff_command[RNO$H_MONTH] = .TIME_32[1];
1501 1613 2 runoff_command[RNO$H_MONTHDAY] = .TIME_32[2];
1502 1614 2 runoff_command[RNO$H_HOURS] = .TIME_32[3];
1503 1615 2 runoff_command[RNO$H_MINUTES] = .TIME_32[4];
1504 1616 2 runoff_command[RNO$H_SECONDS] = .TIME_32[5];
1505 1617 2
1506 1618 1 END; ! End of RNODAT
```

	SE		0000	00000	RNODAT: .WORD	Save nothing	: 1567
			10	C2 00002	SUBL2	#16, SP	: 1606
			7E	D4 00005	CLRL	-(SP)	: 1611
		04	AE	9F 00007	PUSHAB	TIME_32	: 1615
00000000G	EF		02	FB 0000A	CALLS	#2, SYSS\$NUMTIM	: 1618
	50	04	AC	D0 00011	MOVL	RUNOFF_COMMAND, R0	
40	A0		6E	7D 00015	MOVQ	TIME_32, 64(R0)	
48	A0	08	AE	D0 00019	MOVL	TIME_32+8, 72(R0)	
			04	0001E	RET		

: Routine Size: 31 bytes, Routine Base: \$CODE\$ + 0D2A

```

: 1507          1619 1 %FI
: 1508          1620 1
: 1509          1621 1 END
: 1510          1622 0 ELUDOM
! End of module

```

PSECT SUMMARY

Name	Bytes	Attributes
\$OWNS	116	NOVEC, WRT, RD, NOEXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)
_LIB\$KEYOS	52	NOVEC, NOWRT, RD, EXE, SHR, LCL, REL, CON, PIC, ALIGN(1)
_LIB\$STATES	477	NOVEC, NOWRT, RD, EXE, SHR, LCL, REL, CON, PIC, ALIGN(1)
_LIB\$KEY1\$	194	NOVEC, NOWRT, RD, EXE, SHR, LCL, REL, CON, PIC, ALIGN(1)
\$PLITS	716	NOVEC, NOWRT, RD, NOEXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)
\$CODE\$	3401	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]STARLET.L32;1	9776	24 0	581	00:01.1
-\$255\$DUA28:[SYSLIB]TPAMAC.L32;1	42	29 69	14	00:00.1
-\$255\$DUA28:[SYSLIB]XPORT.L32;1	590	36 6	252	00:00.6
-\$255\$DUA28:[RUNOFF.SRC]DSRLIB.L32;1	1248	103 8	86	00:01.0

COMMAND QUALIFIERS

0348

AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION CONFIDENTIAL AND PROPRIETARY

A grid of 25 columns and 12 rows of terminal windows. Each window displays a different VAX/VMS utility or command-line interface. The windows are arranged in a regular grid pattern across the page.

Some of the visible window titles include:

- RNODEF LIS
- RNODET LIS
- RNOUMS LIS
- RNFERM LIS
- RNOURS LIS
- RSKIPS LIS
- RUNOFF LIS
- SAURES LIS
- RSDERM LIS
- RTERM LIS

The content within each window consists of text-based data, command prompts, and status information, typical of a VAX/VMS terminal session.