



```

GGGGGGGG  CCCCCCCC  000000  DDDDDDDD  EEEEEEEEEE
GGGGGGGG  CCCCCCCC  000000  DDDDDDDD  EEEEEEEEEE
GG         CC         00         00  DD         DD  EE
GG         CC         00         00  DD         DD  EE
GG         CC         00         00  DD         DD  EE
GG         CC         00         00  DD         DD  EE
GG         CC         00         00  DD         DD  EE
GG         CC         00         00  DD         DD  EE
GG  GGGGGG  CC         00         00  DD         DD  EEEEEEEE
GG  GGGGGG  CC         00         00  DD         DD  EEEEEEEE
GG         GG  CC         00         00  DD         DD  EE
GG         GG  CC         00         00  DD         DD  EE
GG         GG  CC         00         00  DD         DD  EE
GG         GG  CC         00         00  DD         DD  EE
GGGGGG  CCCCCCCC  000000  DDDDDDDD  EEEEEEEEEE
GGGGGG  CCCCCCCC  000000  DDDDDDDD  EEEEEEEEEE

```

```

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

```

```

1 0001 0 MODULE gcode ( IDENT = 'V04-000'
2 P 0002 0 %BLISS32[, ADDRESSING_MODE (EXTERNAL = LONG_RELATIVE,
3 0003 0 NONEXTERNAL = LONG_RELATIVE)]
4 0004 0 ) =
5 0005 1 BEGIN
6 0006 1
7 0007 1 *****
8 0008 1 *
9 0009 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY *
10 0010 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS. *
11 0011 1 * ALL RIGHTS RESERVED. *
12 0012 1 *
13 0013 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED *
14 0014 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE *
15 0015 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER *
16 0016 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY *
17 0017 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY *
18 0018 1 * TRANSFERRED. *
19 0019 1 *
20 0020 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE *
21 0021 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT *
22 0022 1 * CORPORATION. *
23 0023 1 *
24 0024 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS *
25 0025 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL. *
26 0026 1 *
27 0027 1 *
28 0028 1 *****
29 0029 1
30 0030 1
31 0031 1 **
32 0032 1 FACILITY: DSR (Digital Standard RUNOFF) / DSRPLUS
33 0033 1
34 0034 1 ABSTRACT: Generates intermediate code for paper positioning.
35 0035 1
36 0036 1 ENVIRONMENT: Transportable
37 0037 1
38 0038 1 AUTHOR: R.W.Friday CREATION DATE: June, 1978
39 0039 1

```

GCODE  
V04-000

Revision History

G 16  
16-Sep-1984 00:37:45  
14-Sep-1984 13:06:30

VAX-11 Bliss-32 V4.0-742  
DISK\$VMSMASTER:[RUNOFF.SRC]GCODE.BLI;1 Page 2 (2)

41	0040	1	%SBTTL 'Revision History'
42	0041	1	MODIFIED BY:
43	0042	1	
44	0043	1	009 RER00009 Ron Randall 17-Mar-1983
45	0044	1	For DSRPLUS: Added code related to topnote tests.
46	0045	1	
47	0046	1	008 RER00008 Ron Randall 07-Mar-1983
48	0047	1	Global edit of all modules. Updated module names, idents,
49	0048	1	copyright dates. Changed require files to BLISS library.
50	0049	1	--
51	0050	1	

Module Level Declarations

```

53 0051 1 %SBTTL 'Module Level Declarations'
54 0052 1
55 0053 1 : TABLE OF CONTENTS:
56 0054 1
57 0055 1 FORWARD ROUTINE
58 0056 1     gcpage      : NOVALUE,
59 0057 1     gcpcs     : NOVALUE,
60 0058 1     gcskip    : NOVALUE,
61 0059 1     gtpc      : NOVALUE,
62 0060 1     guskip    : NOVALUE;
63 0061 1
64 0062 1
65 0063 1 : INCLUDE FILES:
66 0064 1
67 0065 1 LIBRARY 'NXPOR:XPOR';           ! XPORT Library
68 0066 1 REQUIRE 'REQ:RNODEF';       ! RUNOFF variant definitions
69 0197 1
70 U 0198 1 %IF DSRPLUS %THEN
71 U 0199 1 LIBRARY 'REQ:DPLLIB';       ! DSRPLUS BLISS Library
72 0200 1 %ELSE
73 0201 1 LIBRARY 'REQ:DSRLIB';       ! DSR BLISS Library
74 0202 1 %FI
75 0203 1
76 0204 1
77 0205 1 : EXTERNAL REFERENCES:
78 0206 1
79 0207 1 EXTERNAL LITERAL
80 0208 1     rintes     : UNSIGNED (8);
81 0209 1
82 0210 1 EXTERNAL
83 0211 1     fnct       : fnct_definition,
84 0212 1     gca        : gca_definition,
85 0213 1     irac       : irac_definition,
86 0214 1     mra        : REF FIXED STRING,
87 0215 1     sca        : sca_definition,
88 0216 1     tsf        : tsf_definition;
89 0217 1
90 U 0218 1 %IF DSRPLUS %THEN
91 U 0219 1 EXTERNAL
92 U 0220 1     topnot    : tn_definition;
93 0221 1 %FI
94 0222 1
95 0223 1 EXTERNAL ROUTINE
96 0224 1     outcrg;
97 0225 1

```

```

99 0226 1 GLOBAL ROUTINE gcpage : NOVALUE =
100 0227 1
101 0228 1 |++
102 0229 1 | FUNCTIONAL DESCRIPTION:
103 0230 1 |
104 0231 1 |     Generates code for starting a new page, if not already
105 0232 1 |     at the top of a page.
106 0233 1 |
107 0234 1 | FORMAL PARAMETERS:     None
108 0235 1 |
109 0236 1 | IMPLICIT INPUTS:      None
110 0237 1 |
111 0238 1 | IMPLICIT OUTPUTS:    None
112 0239 1 |
113 0240 1 | ROUTINE VALUE:
114 0241 1 | COMPLETION CODES:    None
115 0242 1 |
116 0243 1 | SIDE EFFECTS.        None
117 0244 1 | --
118 0245 1 |
119 0246 2 | BEGIN
120 0247 2 |
121 0248 2 | IF .fnct_collecting
122 0249 2 | THEN
123 0250 2 |     RETURN;
124 0251 2 |
125 0252 2 | fs_wchar (mra, rintes);
126 0253 2 | fs_wchar (mra, %C'p');
127 0254 2 | fs_wchar (mra, %C' ');
128 0255 2 | tsf_int_vl = .tsf_int_vl + 3;
129 0256 1 | END;

```

```

! Don't start a new page if
! footnotes are being collected,
! since footnotes all belong on
! one page.

```

! End of GCPAGE

```

.TITLE GCODE
.IDENT  \V04-000\

.EXTRN  RINTES, FNCT, GCA
.EXTRN  IRAC, MRA, SCA, TSF
.EXTRN  OUTCRG

.PSECT  $CODE$,NOWRT,2

```

```

0000 00000
33 00000000G EF E8 00002
50 00000000G EF D0 00009
51 04 A0 9E 00010
00 B1 00G 8F 90 C0014
61 D6 00019
00 B1 0C A0 D6 0001B
70 8F 90 0001E
61 D6 00023
00 B1 0C A0 D6 00025
20 90 00028
61 D6 0002C
0C A0 D6 0002E
18 50 00000000G EF D0 00031
A0 03 C0 00038

```

```

.ENTRY  GCPAGE, Save nothing
BLBS   FNCT+20, 1$
KOVL   MRA, R0
MOVAB  4(R0), R1
MOVVB  #RINTES, @0(R1)
INCL   (R1)
INCL   12(R0)
MOVVB  #112, @0(R1)
INCL   (R1)
INCL   12(R0)
MOVVB  #32, @0(R1)
INCL   (R1)
INCL   12(R0)
MOVL   TSF, R0
ADDL2  #3, 24(R0)

```

```

: 0226
: 0248
: 0252
:
:
: 0253
:
: 0254
:
: 0255

```

GCODE  
V04-000

Module Level Declarations

J 16  
16-Sep-1984 00:37:45  
14-Sep-1984 13:06:30

VAX-11 Bliss-32 V4.0-742  
DISK\$VMSMASTER:[RUNOFF.SRC]GCODE.BLI;1 (4)

Page 5

04 0003C 1\$: RET

: 0256

; Routine Size: 61 bytes, Routine Base: \$CODE\$ + 0000

; 130 0257 1

```

132 0258 1 GLOBAL ROUTINE gcpos (position) : NOVALUE =
133 0259 1
134 0260 1 !++
135 0261 1 FUNCTIONAL DESCRIPTION:
136 0262 1
137 0263 1     Generates code to position to a particular line on a page.
138 0264 1
139 0265 1 FORMAL PARAMETERS:
140 0266 1
141 0267 1     position - Indicates which line is to be positioned to.
142 0268 1     Negative means from the bottom of the page;
143 0269 1     positive means from the top.
144 0270 1
145 0271 1 IMPLICIT INPUTS:      None
146 0272 1
147 0273 1 IMPLICIT OUTPUTS:     None
148 0274 1
149 0275 1 ROUTINE VALUE:
150 0276 1 COMPLETION CODES:     None
151 0277 1
152 0278 1 SIDE EFFECTS:         None
153 0279 1 --
154 0280 1
155 0281 2 BEGIN
156 0282 2 LOCAL
157 0283 2     tsf_phregs : REF VECTOR [tsf_nregs];
158 0284 2
159 0285 2
160 0286 2     ! Don't go anywhere if footnotes are being collected. Leave
161 0287 2     ! the paper positioned where it is.
162 0288 2
163 0289 2 IF .fnct_collecting
164 0290 2 THEN
165 0291 2     RETURN;
166 0292 2
167 0293 2     tsf_phregs = tsf__phregs;
168 0294 2
169 0295 2 IF .tsf_next_reg GEQ tsf_nregs
170 0296 2
171 0297 2     ! Be sure not to allocate too many "registers".
172 0298 2
173 0299 2 THEN
174 0300 2     outcrg ();
175 0301 2
176 0302 2     tsf_phregs [.tsf_next_reg] = .position;
177 0303 2     fs_wchar (mra, rintes);
178 0304 2     fs_wchar (mra, %C'g');
179 0305 2     fs_wchar (mra, .tsf_next_reg);
180 0306 2     tsf_int_vl = .tsf_int_vl + 3;
181 0307 2     tsf_next_reg = .tsf_next_reg + 1;
182 0308 2     tsf_bar_char = .sca_bar_char;      ! Propogate change bars.
183 0309 2     tsf_bar_s = .tsf_bar_s OR .irac_bars;
184 0310 1 END;      ! End of GCPOS

```



			003C 00000	.ENTRY	GCPOS, Save R2,R3,R4,R5		: 0258
	55	00000000G	EF 9E 00002	MOVAB	TSF, R5		
	74	00000000G	EF EB 00009	BLBS	FNC†+20, 2\$		: 0289
	50		65 D0 00010	MOVL	TSF, R0		: 0293
	52	008C	C0 9E 00013	MOVAB	140(R0), TSF_PHREGS		
	05	0088	C0 D1 00018	CMP	136(R0), #5		: 0295
			07 19 0001D	BLSS	1\$		
	00000000G	EF	00 FB 0001F	CALLS	#0, OUTCRG		: 0300
	53		65 D0 00026	MOVL	TSF, R3		: 0302
	54	0088	C3 9E 00029	MOVAB	136(R3), R4		
	50		64 D0 0002E	MOVL	(R4), R0		
	6240	04	AC D0 00031	MOVL	POSITION, (TSF_PHREGS, [R0])		
	52	00000000G	EF D0 00036	MOVL	MRA, R2		: 0303
	51	04	A2 9E 0003D	MOVAB	4(R2), R1		
	00	B1	00G 8F 90 00041	MOVB	#RINTES, @0(R1)		
			61 D6 00046	INCL	(R1)		
		0C	A2 D6 00048	INCL	12(R2)		
	00	B1	67 8F 90 0004B	MOVB	#103, @0(R1)		: 0304
			61 D6 00050	INCL	(R1)		
		0C	A2 D6 00052	INCL	12(R2)		
	00	B1	50 90 00055	MOVB	R0, @0(R1)		: 0305
			61 D6 00059	INCL	(R1)		
		0C	A2 D6 0005B	INCL	12(R2)		
	18	A3	03 C0 0005E	ADDL2	#3, 24(R3)		: 0306
			64 D6 00062	INCL	(R4)		: 0307
	1C	A3 00000000G	FF D0 00064	MOVL	@SCA+136, 28(R3)		: 0308
	50	7C	01 00 0006C	EXTZV	#0, #1, 124(R3), R0		: 0309
	51	00000000G	01 00 00072	EXTZV	#0, #1, IRAC, R1		
	50		51 88 0007B	BISB2	R1, R0		
	7C	A3	01 50 0007E	INSV	R0, #0, #1, 124(R3)		
			00 50 00084	RET			: 0310
			04 00084	2\$:			

: Routine Size: 133 bytes, Routine Base: \$CODE\$ + 003D

: 185 0311 1

```

187 0312 1 GLOBAL ROUTINE gcskip (spacing) : NOVALUE =
188 0313 1
189 0314 1 |++
190 0315 1 | FUNCTIONAL DESCRIPTION:
191 0316 1 |
192 0317 1 |     Generate code to skip lines that don't occur at the top
193 0318 1 |     of a page.
194 0319 1 |
195 0320 1 | FORMAL PARAMETERS:
196 0321 1 |
197 0322 1 |     spacing - Indicates how many lines are to be skipped.
198 0323 1 |
199 0324 1 | IMPLICIT INPUTS:      None
200 0325 1 |
201 0326 1 | IMPLICIT OUTPUTS:    None
202 0327 1 |
203 0328 1 | ROUTINE VALUE:
204 0329 1 | COMPLETION CODES:    None
205 0330 1 |
206 0331 1 | SIDE EFFECTS:        None
207 0332 1 | --
208 0333 1 |
209 0334 2 | BEGIN
210 0335 2 | LOCAL
211 0336 2 |     tsf_phregs : REF VECTOR [tsf_nregs];
212 0337 2 |
213 0338 2 |     tsf_phregs = tsf__phregs;
214 0339 2 |
215 0340 2 | IF .spacing LEQ 0
216 0341 2 | THEN
217 0342 2 |     RETURN;                ! Don't generate code for single spacing.
218 0343 2 |
219 0344 2 | IF .tsf_next_reg GEQ tsf_nregs    ! Don't allocate too many 'registers'.
220 0345 2 | THEN
221 0346 2 |     outcrg ();
222 0347 2 |
223 0348 2 |     tsf_phregs [.tsf_next_reg] = .spacing;
224 0349 2 |     fs_wchar (mra, rintes);
225 0350 2 |
226 U 0351 2 | %IF DSPPLUS %THEN
227 UU 0352 2 | |
228 UU 0353 2 | |     If collecting topnotes, make the lines unconditional and count them.
229 UU 0354 2 | |
230 UU 0355 2 | | IF .tn_collecting
231 UU 0356 2 | | THEN
232 UU 0357 2 | |     BEGIN
233 UU 0358 2 | |         fs_wchar (mra, %(u));
234 UU 0359 2 | |         tsf_lines = .tsf_lines + .spacing;
235 UU 0360 2 | |     END
236 UU 0361 2 | | ELSE
237 U 0362 2 | |     BEGIN
238 0363 2 | | %FI
239 0364 2 | |
240 0365 2 | |     If collecting footnotes, make the lines unconditional and count them.
241 0366 2 | |
242 0367 2 | | IF .fnr_collecting
243 0368 2 | | THEN

```

Module Level Declarations

```

: 244 0369 3 BEGIN
: 245 0370 3 fs_wchar (mra, %C'u');
: 246 0371 3 tsf_lines = .tsf_lines + .spacing;
: 247 0372 3 END
: 248 0373 2 ELSE
: 249 0374 2 fs_wchar (mra, %C's');
: 250 0375 2
: 251 U 0376 2 %IF DSRPLUS %THEN
: 252 U 0377 2 END;
: 253 0378 2 %FI
: 254 0379 2
: 255 0380 2 fs_wchar (mra, .tsf_next_reg);
: 256 0381 2 tsf_int_vl = .tsf_int_vl + 3;
: 257 0382 2 tsf_next_reg = .tsf_next_reg + 1;
: 258 0383 2 tsf_bar_char = .sca_bar_char;
: 259 0384 2 tsf_bars = .tsf_bars OR .irac_bars;
: 260 0385 1 END;

```

! Propagate change bars.  
! End of GCSKIP

				003C 00000	.ENTRY GCSKIP, Save R2,R3,R4,R5	0312
	55	00000000G	EF	9E 00002	MOVAB TSF, R5	0338
	50		65	D0 00009	MOVL TSF, R0	0340
	52	008C	C0	9E 0000C	MOVAB 140(R0), TSF_PHREGS	
	54	04	AC	D0 00011	MOVL SPACING, R4	
			01	14 00015	BGTR 1\$	
				04 00017	RET	
	05	0088	C0	D1 00018 1\$:	CMPL 136(R0), #5	0344
			07	19 0001D	BLSS 2\$	
	00000000G		00	FB 0001F	CALLS #0, OUTCRG	0346
	50		65	D0 00026 2\$:	MOVL TSF, R0	0348
	53	0088	C0	9E 00029	MOVAB 136(R0), R3	
	51		63	D0 0002E	MOVL (R3), R1	
	6241		54	D0 00031	MOVL R4, (TSF_PHREGS)[R1]	
	52	00000000G	EF	D0 00035	MOVL MRA, R2	0349
	51	04	A2	9E 0003C	MOVAB 4(R2), R1	
	00	B1 00G	8F	90 00040	MOVB #RINTES, @0(R1)	
			61	D6 00045	INCL (R1)	
			A2	D6 00047	INCL 12(R2)	
	00	10 00000000G	EF	E9 0004A	BLBC FNCT+20, 3\$	0367
	B1	75	8F	90 00051	MOVB #117, @0(R1)	0370
			61	D6 00056	INCL (R1)	
			A2	D6 00058	INCL 12(R2)	
	34	A0	54	C0 0005B	ADDL2 R4, 52(R0)	0371
			0A	11 0005F	BRB 4\$	0367
	00	B1 73	8F	90 00061 3\$:	MOVB #115, @0(R1)	0374
			61	D6 00066	INCL (R1)	
			A2	D6 00068	INCL 12(R2)	
	00	B1 0C	63	90 0006B 4\$:	MOVB (R3), @0(R1)	0380
			61	D6 0006F	INCL (R1)	
			A2	D6 00071	INCL 12(R2)	
	18	A0	03	C0 00074	ADDL2 #3, 24(R0)	0381
			63	D6 00078	INCL (R3)	0382
	51	7C A0	1C	A0 0007A	MOVL @SCA+136, 28(R0)	0383
			01	00 EF 00082	EXTZV #0, #1, 124(R0), R1	0384

GCODE  
V04-000

Module Level Declarations

C 1  
16-Sep-1984 00:37:45  
14-Sep-1984 13:06:30

VAX-11 Bliss-32 V4.0-742  
DISK\$VMMASTER:[RUNOFF.SRC]GCODE.BLI;1

Page 10  
(6)

GE  
VO

52	00000000G	EF	01	00	FF	00088	EXTZV	#0, #1, IRAC, R2	:
7C	A0	01	51	52	88	00091	BISB2	R2, R1	:
			00	51	F0	00094	INSV	R1, #0, #1, 124(R0)	:
					04	0009A	RET		; 0385

; Routine Size: 155 bytes, Routine Base: \$CODE\$ + 00C2

; 261 0386 1

00

:  
:  
:

```

: 263 0387 1 GLOBAL ROUTINE gtpc (count) : NOVALUE =
: 264 0388 1
: 265 0389 1 ++
: 266 0390 1 FUNCTIONAL DESCRIPTION:
: 267 0391 1
: 268 0392 1     Generates intermediate code for a .TEST PAGE command.
: 269 0393 1
: 270 0394 1 FORMAL PARAMETERS:
: 271 0395 1
: 272 0396 1     count - Specifies how many free lines should be tested for.
: 273 0397 1
: 274 0398 1 IMPLICIT INPUTS:      None
: 275 0399 1
: 276 0400 1 IMPLICIT OUTPUTS:     None
: 277 0401 1
: 278 0402 1 ROUTINE VALUE:
: 279 0403 1 COMPLETION CODES:      None
: 280 0404 1
: 281 0405 1 SIDE EFFECTS:          None
: 282 0406 1 --
: 283 0407 1
: 284 0408 2     BEGIN
: 285 0409 2     LOCAL
: 286 0410 2         tsf_phregs : REF VECTOR [tsf_nregs];
: 287 0411 2
: 288 0412 2         tsf_phregs = tsf__phregs;
: 289 0413 2
: 290 0414 2         |
: 291 0415 2         | If collecting a footnote, don't bother to do a test page, since
: 292 0416 2         | the text will fit by definition.
: 293 0417 2         |
: 294 0418 2         IF .fnct_collecting
: 295 0419 2         THEN
: 296 0420 2             RETURN;
: 297 0421 2
: 298 0422 2 %IF DSRPLUS %THEN
: 299 0423 2
: 300 0424 2         | If collecting a topnote, don't bother to do a test page.
: 301 0425 2         |
: 302 0426 2         IF .tn_collecting
: 303 0427 2         THEN
: 304 0428 2             RETURN;
: 305 0429 2 %FI
: 306 0430 2
: 307 0431 2     IF .tsf_next_reg GEQ tsf_nregs      ! Don't allocate too many "registers".
: 308 0432 2     THEN
: 309 0433 2         outcrg ();
: 310 0434 2
: 311 0435 2         tsf_phregs [.tsf_next_reg] = .count;
: 312 0436 2         fs_wchar (mra, rintes);           ! If (test page..)
: 313 0437 2         's_wchar (mra, %C't');
: 314 0438 2         fs_wchar (mra, .tsf_next_reg);
: 315 0439 2         fs_wchar (mra, rintes);           ! end THEN
: 316 0440 2         fs_wchar (mra, %C'.');
: 317 0441 2         fs_wchar (mra, %C' ');
: 318 0442 2         fs_wchar (mra, rintes);           ! else (page..)
: 319 0443 2         fs_wchar (mra, %C'p');

```

U  
U  
U  
U  
U  
U



GCODE  
V04-000

Module Level Declarations

F 1  
16-Sep-1984 00:37:45  
14-Sep-1984 13:06:30

VAX-11 Bliss-32 V4.0-742  
DISK\$VMSMASTER:[RUNOFF.SRC]GCODE.BLI;1  
Page 13  
(7)

	00	B2	00G	8F	90	00092	MOVB	#RINTES, @0(R2)	:	0445
				62	D6	00097	INCL	(R2)	:	
				61	D6	00099	INCL	(R1)	:	
	00	B2		2E	90	0009B	MOVB	#46, @0(R2)	:	0446
				62	D6	0009F	INCL	(R2)	:	
				61	D6	000A1	INCL	(R1)	:	
	00	B2		20	90	000A3	MOVB	#32, @0(R2)	:	0447
				62	D6	000A7	INCL	(R2)	:	
				61	D6	000A9	INCL	(R1)	:	
	18	A0		0C	C0	000AB	ADDL2	#12, 24(R0)	:	0448
				64	D6	000AF	INCL	(R4)	:	0449
	1C	A0	00000000G	FF	D0	000B1	MOVL	@SCA+136, 28(R0)	:	0450
		51		00	EF	000B9	EXTZV	#0, #1, 124(R0), R1	:	0451
		52		00	EF	020BF	EXTZV	#0, #1, IRAC, R2	:	
		51		52	88	000C8	BISB2	R2, R1	:	
7C	A0			00	F0	000CB	INSV	R1, #0, #1, 124(R0)	:	
				04	00	000D1	RET		:	0452

; Routine Size: 210 bytes, Routine Base: \$CODE\$ + 0150

; 329 0453 1

GE

Module Level Declarations

```

331 0454 1 GLOBAL ROUTINE guskip (spacing) : NOVALUE =
332 0455 1
333 0456 1 !++
334 0457 1 FUNCTIONAL DESCRIPTION:
335 0458 1
336 0459 1     Generates code to skip unconditionally a number of lines.
337 0460 1
338 0461 1 FORMAL PARAMETERS:
339 0462 1
340 0463 1     spacing - Indicates how many lines should be skipped.
341 0464 1
342 0465 1 IMPLICIT INPUTS:      None
343 0466 1
344 0467 1 IMPLICIT OUTPUTS:     None
345 0468 1
346 0469 1 ROUTINE VALUE:
347 0470 1 COMPLETION CODES:     None
348 0471 1
349 0472 1 SIDE EFFECTS:         None
350 0473 1 !--
351 0474 1
352 0475 2 BEGIN
353 0476 2 LOCAL
354 0477 2     tsf_phregs : REF VECTOR [tsf_nregs];
355 0478 2
356 0479 2     tsf_phregs = tsf__phregs;
357 0480 2
358 0481 2     IF .spacing LEQ 0
359 0482 2     THEN
360 0483 2         RETURN;                ! Don't generate code for single spacing.
361 0484 2
362 0485 2     IF .tsf_next_reg GEQ tsf_nregs    ! Don't allocate too many "registers".
363 0486 2     THEN
364 0487 2         outcrg ();
365 0488 2
366 0489 2     tsf_phregs [.tsf_next_reg] = .spacing;
367 0490 2     fs_wchar (mra, rintes);
368 0491 2     fs_wchar (mra, %C'u');
369 0492 2     fs_wchar (mra, .tsf_next_reg);
370 0493 2     tsf_int_vl = .tsf_int_vl + 3;
371 0494 2     tsf_next_reg = .tsf_next_reg + 1;
372 0495 2     tsf_bar_char = .sca_bar_char;                ! Propagate change bars.
373 0496 2     tsf_bar_s   = .tsf_bar_s OR .irac_bar_s;
374 0497 2
375 0498 2
376 0499 2     ! If collecting a footnote, count the number of lines to be generated.
377 0500 2
378 0501 2     IF .fnct_collecting
379 0502 2     THEN
380 0503 2         tsf_lines = .tsf_lines + .spacing;
381 0504 2
382 0505 2     %IF DSRPLUS %THEN
383 0506 2
384 0507 2     ! If collecting a topnote, count the number of lines to be generated.
385 0508 2
386 0509 2     IF .tn_collecting
387 0510 2     THEN

```



```

: 388      U 0511 2      tsf_lines = .tsf_lines + .spacing;
: 389      0512 2 %FI
: 390      0513 2
: 391      0514 1      END;

```

! End of GUSKIP

				007C 00000	.ENTRY GUSKIP, Save R2,R3,R4,R5,R6		0454
	56	00000000G	EF	9E 00002	MOVAB TSF, R6		
	50		66	D0 00009	MOVL TSF, R0		0479
	52	008C	C0	9E 0000C	MOVAB 140(R0), TSF_PHREGS		
	55	04	AC	D0 00011	MOVL SPACING, R5		0481
			76	15 00015	BLEQ 2\$		
	05	0088	C0	D1 00017	CMPL 136(R0), #5		0485
			07	19 0001C	BLSS 1\$		
		00000000G	EF	00 FB 0001E	CALLS #0, OUTCRG		0487
	51		66	D0 00025	MOVL TSF, R1		0489
	54	0088	C1	9E 00028	MOVAB 136(R1), R4		
	53		64	D0 0002D	MOVL (R4), R3		
	6243		55	D0 00030	MOVL R5, (TSF_PHREGS)[R3]		
	50	00000000G	EF	D0 00034	MOVL MRA, R0		0490
	52	04	A0	9E 0003B	MOVAB 4(R0), R2		
	00	B2	00G	8F 90 0003F	MOVB #RINTES, @0(R2)		
			62	D6 00044	INCL (R2)		
			0C	A0 D6 00046	INCL 12(R0)		
	00	B2	75	8F 90 00049	MOVB #117, @0(R2)		0491
			62	D6 0004E	INCL (R2)		
			0C	A0 D6 00050	INCL 12(R0)		
	00	B2	53	90 00053	MOVB R3, @0(R2)		0492
			62	D6 00057	INCL (R2)		
			0C	A0 D6 00059	INCL 12(R0)		
	18	A1	03	C0 0005C	ADDL2 #3, 24(R1)		0493
			64	D6 00060	INCL (R4)		0494
	1C	A1	00000000G	FF D0 00062	MOVL @SCA+136, 28(R1)		0495
	50	7C	A1	00 EF 0006A	EXTZV #0, #1, 124(R1), R0		0496
	52	00000000G	EF	00 EF 00070	EXTZV #0, #1, IRAC, R2		
	50		52	88 00079	BISB2 R2, R0		
	7C	A1	01	50 F0 0007C	INSV R0, #0, #1, 124(R1)		
			00	50 F0 0007C	INSV R0, #0, #1, 124(R1)		
			04	00000000G	BLBC FNCT+20, 2\$		0501
	34	A1	55	C0 00089	ADDL2 R5, 52(R1)		0503
			04	0008D	RET		0514

; Routine Size: 142 bytes, Routine Base: \$CODE\$ + 022F

```

: 392      0515 1
: 393      0516 1 END
: 394      0517 0 ELUDOM

```

! End of module

PSECT SUMMARY

```

: Name                Bytes                Attributes
: $CODE$              701 NOVEC,NOWRT, RD , EXE,NOSHR, LCL, REL, CON,NOPIC,ALIGN(2)

```

Library Statistics

File	----- Total	Symbols Loaded	----- Percent	Pages Mapped	Processing Time
\$255\$DUA28:[SYSLIB]XPORT.L32;1	590	0	0	252	00:00.1
_\$255\$DUA28:[RUNOFF.SRC]DSRLIB.L32;1	1248	28	2	86	00:00.3

COMMAND QUALIFIERS

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

```

: Size:                701 code + 0 data bytes
: Run Time:            00:15.6
: Elapsed Time:       00:38.3
: Lines/CPU Min:      1994
: Lexemes/CPU-Min:    26477
: Memory Used:        103 pages
: Compilation Complete

```

ENDWRD LIS	ERROR LIS	FIGURE LIS	FLGSEM LIS	FOOFIL LIS	GCODE LIS
FCTMRA LIS	FENONLY LIS	FJFNFI LIS	FOOBOT LIS	GBLDCL LIS	
FNDPLG LIS	FOOOUT LIS	FORMAT LIS			

