


```

SSSSSSSS  CCCCCCCC  AAAAAA  NN  NN  TTTTTTTTTT
SSSSSSSS  CCCCCCCC  AAAAAA  NN  NN  TTTTTTTTTT
SS        CC        AA    AA  NN  NN  TT
SS        CC        AA    AA  NN  NN  TT
SS        CC        AA    AA  NNNN NN  TT
SS        CC        AA    AA  NNNN NN  TT
SSSSSS    CC        AA    AA  NN  NN  TT
SSSSSS    CC        AA    AA  NN  NN  TT
          SS        AAAAAAAAAA NN  NN  TT
          SS        AAAAAAAAAA NN  NN  TT
          SS        AA    AA  NN  NN  TT
          SS        AA    AA  NN  NN  TT
          SS        AA    AA  NN  NN  TT
          CC        CCCCCCCC AA    AA  NN  NN  TT
          CC        CCCCCCCC AA    AA  NN  NN  TT

```

```

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
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44

```

0001 0 %TITLE 'Scan and translate text into the MRA'
0002 0 MODULE SCANT (
0003 0 IDENT = 'V04-000'
P 0004 0 %BLISS32[
P 0005 0 ADDRESSING_MODE(EXTERNAL=LONG_RELATIVE, NONEXTERNAL=LONG_RELATIVE)
0006 0 ]
0007 0 ) =
0008 1 BEGIN
0009 1
0010 1 *****
0011 1 *
0012 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY *
0013 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS. *
0014 1 * ALL RIGHTS RESERVED. *
0015 1 *
0016 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED *
0017 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE *
0018 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER *
0019 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY *
0020 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY *
0021 1 * TRANSFERRED. *
0022 1 *
0023 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE *
0024 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT *
0025 1 * CORPORATION. *
0026 1 *
0027 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS *
0028 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL. *
0029 1 *
0030 1 *
0031 1 *****
0032 1
0033 1 ++
0034 1 FACILITY: DSR (Digital Standard RUNOFF) / DSRPLUS
0035 1
0036 1 ABSTRACT: Text scanner and translator, creating the MRA.
0037 1
0038 1
0039 1 ENVIRONMENT: Transportable
0040 1
0041 1 AUTHOR: R.W.Friday
0042 1
0043 1 CREATION DATE: April, 1978
0044 1

```

: 4
: 4
: 4
: 4
: 4

```

: 46 0045 1 %SBTTL 'Revision History'
: 47 0046 1
: 48 0047 1   MODIFIED BY:
: 49 0048 1
: 50 0049 1   009  REM00009  Ray Marshall  17-November-1983
: 51 0050 1      Modified the external definition of ATABLE to use the new
: 52 0051 1      macro ATABLE_DEFINITION defined in ATCODE.REQ.
: 53 0052 1
: 54 0053 1   008  REM00008  Ray Marshall  7-November-1983
: 55 0054 1      Added support for case changing within the DEC multinational
: 56 0055 1      character set.
: 57 0056 1
: 58 0057 1   007  KFA00007  Ken Alden    15-Mar-1983
: 59 0058 1      For DSRPLUS: Added recognition of an escape sequence
: 60 0059 1
: 61 0060 1   006  KAD00006  Keith Dawson 07-Mar-1983
: 62 0061 1      Global edit of all modules. Updated module names, idents,
: 63 0062 1      copyright dates. Changed require files to BLISS library.
: 64 0063 1
: 65 0064 1  --

```


SCANT
V04-000

Scan and translate text into the MRA
Module Level Declarations

K 2
16-Sep-1984 01:45:21
14-Sep-1984 13:08:03

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[RUNOFF.SRC]SCANT.BLI;1 (3) Page 4

**F1

:	124	0252	1	KHAR;
:	125	0253	1	
:	126	0254	1	EXTERNAL ROUTINE
:	127	0255	1	DOFLG,
:	128	0256	1	ENDCHR,
:	129	0257	1	ENDWRD,
:	130	0258	1	ERM,
:	131	0259	1	ERMS,
:	132	0260	1	FCIMRA,
:	133	0261	1	PUTMSG,
:	134	0262	1	RSKIPS,
:	135	0263	1	XTAB;
:	136	0264	1	

```

138 0265 1 GLOBAL ROUTINE SCANT : NOVALUE =
139 0266 1
140 0267 1 !++
141 0268 1 FUNCTIONAL DESCRIPTION:
142 0269 1
143 0270 1
144 0271 1 FORMAL PARAMETERS: None
145 0272 1
146 0273 1 IMPLICIT INPUTS: None
147 0274 1
148 0275 1 IMPLICIT OUTPUTS: None
149 0276 1
150 0277 1 ROUTINE VALUE:
151 0278 1 COMPLETION CODES: None
152 0279 1
153 0280 1 SIDE EFFECTS: None
154 0281 1
155 0282 1 --
156 0283 1
157 0284 2 BEGIN
158 0285 2
159 0286 2 WHILE .FS_LENGTH (IRA) GEQ 0 DO
160 0287 3 BEGIN
161 0288 3
162 0289 3 LABEL
163 0290 3 PRESCAN;
164 0291 3
165 0292 3 PRESCAN :
166 0293 4 BEGIN !Start PRESCAN operation
167 0294 4
168 0295 4 IF .SCA_PRESCAN
169 0296 4 THEN
170 0297 4 !If SCA PRESCAN is TRUE, text scanning is stopped by a ';'.
171 0298 5 BEGIN
172 0299 5 !Make sure that a <QUOTE flag> is not forcing the character.
173 0300 5
174 0301 5 IF .SCA_FLAGS !
175 0302 5 AND .FLGT [QUO_FLAG, FLAG_ENABLED]
176 0303 5 THEN
177 0304 5
178 0305 5 IF .KHAR EQL .FLGT [QUO_FLAG, FLAG_CHARACTER]
179 0306 5 THEN
180 0307 5 !<QUOTE flag> forcing something
181 0308 5 !Normal processing will take care of it
182 0309 5 LEAVE PRESCAN;
183 0310 5
184 0311 5 !See if <OVERSTRIKE flag> forces next character
185 0312 5
186 0313 5 IF .SCA_FLAGS !
187 0314 5 AND .FLGT [OVR_FLAG, FLAG_ENABLED]
188 0315 5 THEN
189 0316 5
190 0317 5 IF .KHAR EQL .FLGT [OVR_FLAG, FLAG_CHARACTER]
191 0318 5 THEN
192 0319 5 !<OVERSTRIKE flag> forcing next character.
193 0320 5 !Normal processing will take care of two characters.
194 0321 5 LEAVE PRESCAN;

```

```

195 0322 5
196 0323 5
197 0324 5
198 0325 5
199 0326 5
200 0327 5
201 0328 4
202 0329 4
203 0330 3
204 0331 3
205 0332 3
206 0333 3
207 0334 3
208 0335 3
209 0336 4
210 0337 4
211 0338 5
212 0339 5
213 0340 4
214 0341 4
215 0342 5
216 0343 4
217 0344 4
218 0345 4
219 U 0346 4 %IF DSRPLUS %THEN
220 U 0347 4
221 0348 4 %FI
222 0349 4
223 0350 4
224 0351 4
225 0352 4
226 0353 4
227 0354 5
228 0355 5
229 0356 5
230 0357 5
231 0358 5
232 0359 5
233 0360 5
234 0361 5
235 0362 5
236 0363 4
237 0364 4
238 0365 3
239 0366 3
240 0367 3
241 0368 4
242 0369 4
243 0370 4
244 0371 3
245 0372 3
246 0373 3
247 0374 4
248 0375 4
249 0376 4
250 0377 4
251 0378 4

```

```

IF .KHAR EQL %C';'
THEN
!Found a ';' terminating the scan.
RETURN;

END;

END; !End of PRESCAN operation

CASE .ATABLE [.KHAR] FROM 1 TO ACTION_COUNT OF
SET
[A_CONTROL] :
BEGIN
IF (.KHAR EQL 0) !NULL?
OR (.KHAR EQL %O'177') !DEL?
THEN
!Ignore NULL and DEL
(0)
ELSE
IF .SCA_CC_OK
OR .SCA_PASS
THEN
!Control characters are ok or this is a pass through sequence.
ENDCHR (.KHAR)
ELSE
!Control characters are not ok.
BEGIN
LOCAL
X,
Y;
Y = CH$PTR (X);
CH$WCHAR (.KHAR + %C'@', .Y);
ERM (RNFIIF, .Y, 1);
ERM (RNFLOC, 0, 0);
END;
KCNS ();
END;

[A_DIGIT, A_OTHER] :
BEGIN
ENDCHR (.KHAR);
KCNS ();
END;

[A_PUNCT] :
BEGIN
!A punctuation mark, maybe.
ENDCHR (.KHAR);
SCA WRD_LC_PUNCT = .SCA_PERIOD AND .SCA_FILL;
KCNS ();

```



```

252 0379 3      END;
253 0380 3
254 0381 3      [A_SPACE] :
255 0382 4      BEGIN
256 0383 4      IF .SCA_FC          !Was the previous character a space?
257 0384 4      THEN
258 0385 5      BEGIN
259 0386 5      !When filling, multiple spaces get replaced
260 0387 5      !by a single space.
261 0388 5      !In .NO FILL mode, all spaces are saved
262 0389 5      IF NOT .SCA_FILL
263 0390 5      THEN
264 0391 5      OUTXSP ()
265 0392 5      ELSE
266 0393 5      !Be sure user didn't say '& ', which
267 0394 5      !is an error in .FILL mode when there
268 0395 5      !is a string of spaces in a row
269 0396 5      IF .SCA_WRD_AC_UND    !Underline this space?
270 0397 5      THEN
271 0398 5      !User did say '& ', which is a problem for him.
272 0399 6      BEGIN
273 0400 6      SCA_WRD_AC_UND = FALSE;    !Fix it up.
274 0401 6      ERMS (RNFNA, .FS_START (IRA),
275 0402 6      CHSDIFF (.FS_NEXT (IRA), .FS_START (IRA)) - 1);
276 0403 5      END;
277 0404 5      END
278 0405 4      ELSE          !This space terminates a word.
279 0406 4      ENDWRD (TRUE, .SCA_JUSTIFY, FALSE),
280 0407 4
281 0408 4      KCNS ();          !Get next character.
282 0409 4
283 0410 4      IF .SCA_RSKIPS
284 0411 4      THEN
285 0412 4      RSKIPS (IRA);
286 0413 4
287 0414 3      END;
288 0415 3
289 0416 3      [A_TAB] :
290 0417 4      BEGIN
291 0418 4      IF NOT XTAB ()
292 0419 4      THEN
293 0420 5      BEGIN          !Couldn't expand tab, so handle like a space.
294 0421 5      IF .SCA_FC          !Was the previous character a space?
295 0422 5      THEN
296 0423 5      !When filling, multiple spaces get replaced a single
297 0424 5      !space. .NO FILL mode, all spaces are saved
298 0425 6      BEGIN
299 0426 6      IF NOT .SCA_FILL
300 0427 6      THEN
301 0428 6      OUTXSP ();
302 0429 6      END
303 0430 5      ELSE
304 0431 5      ENDWRD (TRUE, .SCA_JUSTIFY, FALSE);    !This space terminates a word.
305 0432 4      END;
306 0433 4
307 0434 4      KCNS ();          !Get next character
308 0435 4

```

```

309 0436 4          IF .SCA_RSKIPS
310 0437 4          THEN
311 0438 4          !Skip multiple spaces.
312 0439 4          RSKIPS (IRA);
313 0440 4
314 0441 4          SCA_FC_CASE = TRUE;          !Set case rules
315 0442 4          END;
316 0443 3
317 0444 3  [A_U_LETTER] :
318 0445 4  BEGIN
319 0446 4  IF .SCA_FC_CASE
320 0447 4  THEN
321 0448 4  KHAR = .KHAR + .SCA_WRD_FC_UT
322 0449 4  ELSE
323 0450 4  KHAR = .KHAR + .SCA_WRD_OC_UT;
324 0451 4
325 0452 4  ENDCHR (.KHAR);
326 0453 4  KCNS ();
327 0454 3  END;
328 0455 3
329 0456 3  [A_L_LETTER] :
330 0457 4  BEGIN
331 0458 4  IF .SCA_FC_CASE
332 0459 4  THEN
333 0460 4  KHAR = .KHAR + .SCA_WRD_FC_LT
334 0461 4  ELSE
335 0462 4  KHAR = .KHAR + .SCA_WRD_OC_LT;
336 0463 4
337 0464 4  ENDCHR (.KHAR);
338 0465 4  KCNS ();
339 0466 3  END;
340 0467 3
341 0468 3  [A_U_ALPHA] :
342 0469 4  BEGIN
343 0470 4  IF .SCA_FC_CASE
344 0471 4  THEN
345 0472 4  KHAR = .KHAR + .SCA_MNWRD_FC_UT
346 0473 4  ELSE
347 0474 4  KHAR = .KHAR + .SCA_MNWRD_OC_UT;
348 0475 4
349 0476 4  ENDCHR (.KHAR);
350 0477 4  KCNS ();
351 0478 3  END;
352 0479 3
353 0480 3  [A_L_ALPHA] :
354 0481 4  BEGIN
355 0482 4  IF .SCA_FC_CASE
356 0483 4  THEN
357 0484 4  KHAR = .KHAR + .SCA_MNWRD_FC_LT
358 0485 4  ELSE
359 0486 4  KHAR = .KHAR + .SCA_MNWRD_OC_LT;
360 0487 4
361 0488 4  ENDCHR (.KHAR);
362 0489 4  KCNS ();
363 0490 3  END;
364 0491 3
365 0492 3  [A_FLAG] :

```

```

: 366      0493      3          DOFLG ();
: 367      0494      3
: 368      0495      3          [A_INT ESC] :
: 369      0496      4          BEGIN
: 370      0497      4          PUTMSG (RNFSSR, 0, 0);
: 371      0498      4          RETURN;
: 372      0499      3          END;
: 373      0500      3          TES;
: 374      0501      3
: 375      0502      2          END;
: 376      0503      1          END;

```

!End of SCANT

```

.TITLE SCANT Scan and translate text into the MRA
.IDENT \V04-000\

```

```

.EXTRN RINTES, FLGT, IRA
.EXTRN MRA, SCA, TSF, RNFFNA
.EXTRN RNFIBO, RNFIIF, RNFLOC
.EXTRN RNFSSR, ATABLE, KHAR
.EXTRN DOFLG, ENDCHR, ENDWRD
.EXTRN ERM, ERMS, FCIMRA
.EXTRN PUTMSG, RSKIPS, XTAB

```

.PSECT \$CODE\$,NOWRT,2

```

.ENTRY SCANT, Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11 : 0265
MOVAB ENDWRD, R11
MOVAB OUTXSP, R10
MOVAB ERM, R9
MOVZBL #RINTES, R8
MOVAB FLGT+12, R7
MOVAB ENDCHR, R6
MOVAB IRA+12, R5
MOVAB KHAR, R4
MOVAB SCA+208, R3
SUBL2 #4, SP
TSTL IRA+12 : 0286
BGEQ 2$
RET
BLBC SCA+172, 5$ : 0295
BLBC @SCA+144, 4$ : 0301
BLBC FLGT+12, 3$ : 0302
CPL KHAR, FLGT+84 : 0305
BEQL 5$
BLBC @SCA+144, 4$ : 0313
BLBC FLGT+48, 4$ : 0314
CPL KHAR, FLGT+120 : 0317
BEQL 5$
CPL KHAR, #59 : 0323
BNEQ 5$
RET
MOVL KHAR, R2 : 0332
CASEB ATABLE[R2], #1, #11
.WORD 12$-6$, -
       18$-6$, -
       37$-6$, -

```

```

OFFC 00000
5B 00000000G EF 9E 00002
5A 00000000V EF 9E 00009
59 00000000G EF 9E 00010
58 00000000G 8F 9A 00017
57 00000000G EF 9E 0001B
56 00000000G EF 9E 00022
55 00000000G EF 9E 00029
54 00000000G EF 9E 00030
53 00000000G EF 9E 00037
5E 04 C2 0003E
65 D5 00041 1$:
01 18 00043
04 00045
21 DC A3 E9 00046 2$:
17 CO B3 E9 0004A
06 67 E9 0004E
48 A7 64 D1 00051
14 13 00055
0A CO B3 E9 00057 3$:
06 24 A7 E9 0005B
6C A7 64 D1 0005F
06 13 00063
3B 64 D1 00065 4$:
01 12 00068
04 0006A
52 64 D0 0006B 5$:
01 00000000GEF42 8F 0006E
00C2 0062 00077 6$:
0173 010B 0007F
013E 0029 0050 00087

```

```

0018      0B
0173      0169
013E      0029
          012D
          0029
          010B
          0050

```

: 1
: 1
: 1
: 1
: 1
: 1

					7\$-6\$,-		
					25\$-6\$,-		
					27\$-6\$,-		
					8\$-6\$,-		
					39\$-6\$,-		
					10\$-6\$,-		
					8\$-6\$,-		
					29\$-6\$,-		
					31\$-6\$		
			52	D5 0008F 7\$:	TSTL	R2	0338
			43	13 00091	BEQL	11\$	
	0000007F	8F	52	D1 00093	CMPL	R2, #127	0339
			3A	13 0009A	BEQL	11\$	
		05	B3	E9 0009C	BLBC	@SCA+108, 9\$	0345
			52	DD 000A0 8\$:	PUSHL	R2	0351
			0121	31 000A2	BRW	34\$	
	60	50	6E	9E 000A5 9\$:	MOVAB	X, Y	0359
		52	8F	81 000A8	ADDB3	#64, R2, (Y)	0360
			01	DD 000AD	PUSHL	#1	0361
			50	DD 000AF	PUSHL	Y	
			8F	DD 000B1	PUSHL	#RNFIF	
		69	03	FB 000B7	CALLS	#3, ERM	
			7E	7C 000BA	CLRQ	-(SP)	0362
			8F	DD 000BC	PUSHL	#RNFLOC	
		69	03	FB 000C2	CALLS	#3, ERM	
			0F	11 000C5	BRB	11\$	0364
			52	DD 000C7 10\$:	PUSHL	R2	0376
		66	01	FB 000C9	CALLS	#1, ENDCHR	
		50	B3	D2 000CC	MCOML	@SCA+104, R0	0377
	78	A3	B0	50	CB 000D0	BICL3	R0, @SCA+128, SCA+328
				00F0	31 000D6 11\$:	BRW	35\$
			2D	C4	A3 E9 000D9 12\$:	BLBC	SCA+148, 14\$
			05	98	B3 E8 000DD	BLBS	@SCA+104, 13\$
			6A		00 FB 000E1	CALLS	#0, OUTXSP
				2E	11 000E4	BRB	15\$
	29	F8	A3		01 E1 000E6 13\$:	BBC	#1, SCA+200, 15\$
		F8	A3		02 8A 000EB	BICB2	#2, SCA+200
	50	F8	A5	F4	A5 C3 000EF	SUBL3	IRA, IRA+4, R0
				FF	A0 9F 000F5	PUSHAB	-1(R0)
				F4	A5 DD 000F8	PUSHL	IRA
			00000000G	EF	00000000G 8F DD 000FB	PUSHL	#RNFNA
					03 FB 00101	CALLS	#3, ERMS
					0A 11 00108	BRB	15\$
					7E D4 0010A 14\$:	CLRL	-(SP)
			94	B3	DD 0010C	PUSHL	@SCA+100
				01	DD 0010F	PUSHL	#1
			68	03	FB 00111	CALLS	#3, ENDWRD
				65	D5 00114 15\$:	TSTL	IRA+12
				08	14 00116	BGTR	16\$
			64	58	9A 00118	MOVZBL	R8, KHAR
			65	01	CE 0011B	MNEGL	#1, IRA+12
				09	11 0011E	BRB	17\$
			64	F8	B5 9A 00120 16\$:	MOVZBL	@IRA+4, KHAR
				F8	A5 D6 00124	INCL	IRA+4
					65 D7 00127	DECL	IRA+12
			53	FC	A3 E9 00129 17\$:	BLBC	SCA+204, 24\$
				F4	A5 9F 0012D	PUSHAB	IRA
							0410
							0412

00000000G	EF		01	FB	00130	CALLS	#1, RSKIPS	:	
			47	11	00137	BRB	24\$:	0332
00000000G	EF		00	FB	00139	18\$:	CALLS	#0, XTAB	0418
	17		50	E8	00140		BLBS	R0, 20\$	
	09	C4	A3	E9	00143		BLBC	SCA+148, 19\$	0421
	0F	98	B3	E8	00147		BLBS	@SCA+104, 20\$	0426
	6A		00	FB	00148		CALLS	#0, OUTXSP	0428
			0A	11	0014E		BRB	20\$	0421
			7E	D4	00150	19\$:	CLRL	-(SP)	0431
		94	B3	DD	00152		PUSHL	@SCA+100	
			01	DD	00155		PUSHL	#1	
	6B		03	FB	00157		CALLS	#3, ENDWRD	
			65	D5	0015A	20\$:	TSTL	IRA+12	0434
			08	14	0015C		BGTR	21\$	
	64		58	9A	0015E		MOVZBL	R8, KHAR	
	65		01	CE	00161		MNEGL	#1, IRA+12	
			09	11	00164		BRB	22\$	
	64	F8	B5	9A	00166	21\$:	MOVZBL	@IRA+4, KHAR	
		F8	A5	D6	0016A		INCL	IRA+4	
			65	D7	0016D		DECL	IRA+12	
	0A	FC	A3	E9	0016F	22\$:	BLBC	SCA+204, 23\$	0436
		F4	A5	9F	00173		PUSHAB	IRA	0439
00000000G	EF		01	FB	00176		CALLS	#1, RSKIPS	
	63		01	D0	0017D	23\$:	MOVL	#1, SCA+208	0441
			65	11	00180	24\$:	BRB	38\$	0332
	07		63	E9	00182	25\$:	BLBC	SCA+208, 26\$	0446
	64	FF40	C3	C0	00185		ADDL2	SCA+16, KHAR	0448
			38	11	0018A		BRB	33\$	
	64	FF48	C3	C0	0018C	26\$:	ADDL2	SCA+24, KHAR	0450
			31	11	00191		BRB	33\$	0452
	07		63	E9	00193	27\$:	BLBC	SCA+208, 28\$	0458
	64	FF44	C3	C0	00196		ADDL2	SCA+20, KHAR	0460
			27	11	0019B		BRB	33\$	
	64	FF4C	C3	C0	0019D	28\$:	ADDL2	SCA+28, KHAR	0462
			20	11	001A2		BRB	33\$	0464
	07		63	E9	001A4	29\$:	BLBC	SCA+208, 30\$	0470
	64	FF70	C3	C0	001A7		ADDL2	SCA+64, KHAR	0472
			16	11	001AC		BRB	33\$	
	64	FF78	C3	C0	001AE	30\$:	ADDL2	SCA+72, KHAR	0474
			0F	11	001B3		BRB	33\$	0476
	07		63	E9	001B5	31\$:	BLBC	SCA+208, 32\$	0482
	64	FF74	C3	C0	001B8		ADDL2	SCA+68, KHAR	0484
			05	11	001BD		BRB	33\$	
	64	FF7C	C3	C0	001BF	32\$:	ADDL2	SCA+76, KHAR	0486
			64	DD	001C4	33\$:	PUSHL	KHAR	0488
	66		01	FB	001C6	34\$:	CALLS	#1, ENDCHR	
			65	D5	001C9	35\$:	TSTL	IRA+12	0489
			08	14	001CB		BGTR	36\$	
	64		58	9A	001CD		MOVZBL	R8, KHAR	
	65		01	CE	001D0		MNEGL	#1, IRA+12	
			12	11	001D3		BRB	38\$	
	64	F8	B5	9A	001D5	36\$:	MOVZBL	@IRA+4, KHAR	
		F8	A5	D6	001D9		INCL	IRA+4	
			65	D7	001DC		DECL	IRA+12	
			07	11	001DE		BRB	38\$	0332
00000000G	EF		00	FB	001E0	37\$:	CALLS	#0, DOFLG	0493
			FE57	31	001E7	38\$:	BRW	1\$	

SCANT
V04-000

Scan and translate text into the MRA
Module Level Declarations

F 3
16-Sep-1984 01:45:21
14-Sep-1984 13:08:03

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

Page 12
(4)

00000000G	EF	00000000G	7E	7C	001EA	39\$:	CLRQ	-(SP)	:	0497
			8F	DD	001EC		PUSHL	#RNFSSR	:	
			03	FB	001F2		CALLS	#3, PUTMSG	:	
				04	001F9		RET		:	0503

; Routine Size: 506 bytes, Routine Base: \$CODE\$ + 0000

```

378 0504 1 ROUTINE OUTXSP : NOVALUE =
379 0505 1
380 0506 1 ++
381 0507 1 FUNCTIONAL DESCRIPTION:
382 0508 1
383 0509 1 This routine is called in .NO FILL mode, when extra spaces are detected
384 0510 1 between words. It appends them to any other such spaces until either a new
385 0511 1 word is started, or the current output line is filled. In the latter case,
386 0512 1 such extra spaces get dropped.
387 0513 1
388 0514 1 FORMAL PARAMETERS: None
389 0515 1
390 0516 1 IMPLICIT INPUTS: None
391 0517 1
392 0518 1 IMPLICIT OUTPUTS: None
393 0519 1
394 0520 1 ROUTINE VALUE:
395 0521 1 COMPLETION CODES: None
396 0522 1
397 0523 1 SIDE EFFECTS: None
398 0524 1
399 0525 1 --
400 0526 1
401 0527 2 BEGIN
402 0528 2
403 0529 2 IF CH$DIFF(.FS_NEXT(MRA),.FS_START(MRA)) EQL .TSF_INT_VL!
404 0530 2 THEN
405 0531 2 FCIMRA (); !Will be first character going into MRA.
406 0532 2
407 0533 2 !Make sure that an extremely long string of spaces doesn't cause MRA
408 0534 2 !to overflow. Stop 10 short so that underlining, bolding, etc can be
409 0535 2 !forced out.
410 0536 2 IF (.FS_MAXSIZE (MRA) - .FS_LENGTH (MRA)) LSS 10
411 0537 2 THEN
412 0538 2 BEGIN !Character might not fit in storage
413 0539 2
414 0540 2 EXTERNAL
415 0541 2 IRA : FIXED_STRING; !needed only for error handling.
416 0542 2
417 0543 2 ERM (RNFIBO, .FS_NEXT (IRA), .FS_LENGTH (IRA));
418 0544 2 ERM (RNFLOC, 0, 0);
419 0545 2 ENDWRD (FALSE, FALSE, FALSE);
420 0546 2 WHILE .KHAR NEQ RINTES DO !Skip to end of input line.
421 0547 2 KCNS ();
422 0548 2 RETURN
423 0549 2 END;
424 0550 2
425 0551 2 IF .SCA_WRD_AC_UND
426 0552 2 THEN
427 0553 2 BEGIN !forced underlining for this space
428 0554 2 FS_WCHAR (MRA, RINTES);
429 0555 2 FS_WCHAR (MRA, %C'U');
430 0556 2 FS_WCHAR (MRA, %C' ');
431 0557 2 SCA_WRD_LST_UND = .SCA_WRD_LST_UND + 1;
432 0558 2 SCA_WRD_AC_UND = FALSE;
433 0559 2 END;
434 0560 2

```

```
: 435 0561 2 FS WCHAR (MRA, %C' '); !Put out the actual space.
: 436 0562 2 SCA_WRD_LST_SP = .SCA_WRD_LST_SP + 1;
: 437 0563 2 !Update pointer to where next word will start.
: 438 0564 2 SCA_WRD_PNTR = .FS_NEXT (MRA);
: 439 0565 1 END; !End of OUTXSP
```

```
00FC 00000 OUTXSP: .WORD Save R2,R3,R4,R5,R6,R7
57 00G 8F 9A 00002 MOVZBL #RINTES, R7 0504
56 00000000G EF 9E 00006 MOVAB ERM, R6
55 00000000G EF 9E 0000D MOVAB KHAR, R5
54 00000000G EF 9E 00014 MOVAB MRA, R4
53 00000000G EF 9E 0001B MOVAB SCA+200, R3
52 00000000G EF 9E 00022 MOVAB IRA+12, R2
50 0: A0 64 D0 00029 MOVL MRA, R0 0529
51 00000000G EF D0 00031 SUBL3 (R0), 4(R0), R0
18 A1 50 D1 00038 MOVL TSF, R1
00000000G EF 07 12 0003C CMPL R0, 24(R1)
51 0C A0 00 FB 0003E BNEQ 1$ 0531
51 08 A0 64 D0 00045 1$: MOVL MRA, R0 0536
51 08 A0 0A C1 00048 ADDL3 #10, 12(R0), R1
40 18 00051 CMPL 8(R0), R1
62 DD 00053 BGEQ 4$ 0543
0000U000G EF A2 DD 00055 PUSHL IRA+12
66 0000U000G 8F DD 00058 PUSHL IRA+4
03 FB 0005E PUSHL #RNFIBO
7E 7C 00061 CALLS #3, ERM 0544
00000000G EF 8F C' 00063 CLRQ -(SP)
03 FB 00069 CALLS #3, ERM 0545
7E 7C 0006C CLRQ -(SP)
7E D4 0006E CLRL -(SP)
00000000G EF 03 FB 00070 CALLS #3, ENDWRD
57 57 D1 00077 2$: CMPL KHAR, R7 0546
5B 13 0007A BEQL 6$
62 D5 0007C TSTL IRA+12 0547
08 14 0007E BGTR 3$
65 57 9A 00080 MOVZBL R7, KHAR
62 01 CE 00083 MNEGL #1, IRA+12
EF 11 00086 BRB 2$
65 F8 B2 9A 00088 3$: MOVZBL @IRA+4, KHAR
F8 A2 D6 0008C INCL IRA+4
62 D7 0008F DECL IRA+12
E4 11 00091 BRB 2$
2A 63 01 E1 00093 4$: BBC #1, SCA+200, 5$ 0551
50 54 D0 00097 MOVL MRA, R0 0554
51 04 A0 9E 0009A MOVAB 4(R0), R1
00 B1 57 90 0009E MOVAB R7, @0(R1)
61 D6 000A2 INCL (R1)
00 0C A0 D6 000A4 INCL 12(R0)
00 B1 55 8F 90 000A7 MOVAB #85, @0(R1) 0555
61 D6 000AC INCL (R1)
0C A0 D6 000AE INCL 12(R0)
```


SCANT
V04-000

Scan and translate text into the MRA
Module Level Declarations

1 3
16-Sep-1984 01:45:21
14-Sep-1984 13:08:03

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

Page 15
(5)

SETD
V04-

```

00 B1          20 90 000B1      MOVB   #32, @0(R1)      : 0556
                61 D6 000B5      INCL   (R1)            :
                0C A0 D6 000B7      INCL   12(R0)         :
008C          C3 D6 000BA      INCL   SCA+340        : 0557
                02 8A 000BE      BICB2  #2, SCA+200    : 0558
04 63          64 D0 000C1 5$:    MOVL   MRA, R0        : 0561
04 50          20 90 000C4      MOVB   #32, @4(R0)    :
04 B0          04 A0 D6 0C0C8      INCL   4(R0)         :
                0C A0 D6 000CB      INCL   12(R0)         :
0084          C3 D6 000CE      INCL   SCA+33?       : 0562
04 30 A3       04 A0 D0 000D2      MOVL   4(R0), SCA+248 : 0564
                04 000D7 6$:      RET                    : 0565

```

; Routine Size: 216 bytes, Routine Base: \$CODE\$ + 01FA

```

: 440          0566 1
: 441          0567 1 END           !End of module
: 442          0568 0 ELUDOM

```

PSECT SUMMARY

Name	Bytes	Attributes
\$CODE\$	722	NOVEC, NOWRT, RD, EXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)

Library Statistics

File	Symbols		Pages Mapped	Processing Time
	Total	Loaded		
\$255\$DUA28:[SYSLIB]XPORT.L32;1	590	0	252	00:00.2
\$255\$DUA28:[RUNOFF.SRC]DSRLIB.L32;1	1248	63	86	00:00.3

COMMAND QUALIFIERS

```

:
: BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS$:SCANT/OBJ=OBJ$:SCANT MSRC$:SCANT/UPDATE=(ENH$:SCANT)
:
: Size:          722 code + 0 data bytes
: Run Time:      00:16.6
: Elapsed Time: 00:33.3
: Lines/CPU Min: 2051
: Lexemes/CPU-Min: 20785

```

```

: 1
: 1
: 1
: 1
: 1
: 1
: 1
: 1
: 1
: 1
: 1
: 1
: 1
: 1
: 1

```


