


```
RRRRRRRR      EEEEEEEEEEE PPPPPPPP      EEEEEEEEEEE      AAAAAA      TTTTTTTTTT
RRRRRRRR      EEEEEEEEEEE PPPPPPPP      EEEEEEEEEEE      AAAAAA      TTTTTTTTTT
RR      RR      EE      PP      PP      EE      AA      AA      TT
RR      RR      EE      PP      PP      EE      AA      AA      TT
RR      RR      EE      PP      PP      EE      AA      AA      TT
RR      RR      EE      PP      PP      EE      AA      AA      TT
RRRRRRRR      EEEEEEEEEEE PPPPPPPP      EEEEEEEEEEE      AA      AA      TT
RRRRRRRR      EEEEEEEEEEE PPPPPPPP      EEEEEEEEEEE      AA      AA      TT
RR      RR      EE      PP      PP      EE      AAAAAAAAAA      TT
RR      RR      EE      PP      PP      EE      AAAAAAAAAA      TT
RR      RR      EE      PP      PP      EE      AA      AA      TT
RR      RR      EE      PP      PP      EE      AA      AA      TT
RR      RR      EEEEEEEEEEE PP      EEEEEEEEEEE      AA      AA      TT
RR      RR      EEEEEEEEEEE PP      EEEEEEEEEEE      AA      AA      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
```



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


REPEAT
V04-000

Processes the .REPEAT command.
Revision History

C 14
16-Sep-1984 01:31:44
14-Sep-1984 13:07:50

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

Page 2
(2)

```

: 42 0041 1 %SBTTL 'Revision History'
: 43 0042 1
: 44 0043 1 MODIFIED BY:
: 45 0044 1
: 46 0045 1 006 REM00006 Ray Marshall 11-August-1983
: 47 0046 1 Previous edit was incomplete. There were several RETURN
: 48 0047 1 statements -- all of which needed return values.
: 49 0048 1
: 50 0049 1 005 REM00005 Ray Marshall July-1983
: 51 0050 1 Routine REPEAT was changed to cause it to return a value.
: 52 0051 1
: 53 0052 1 004 KAD00004 Keith Dawson 07-Mar-1983
: 54 0053 1 Global edit of all modules. Updated module names, idents,
: 55 0054 1 copyright dates. Changed require files to BLISS library.
: 56 0055 1
: 57 0056 1 !--

```

REPEAT
V04-000

Processes the .REPEAT command.
Module Level Declarations

D 14
16-Sep-1984 01:31:44
14-Sep-1984 13:07:50

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

Page 3
(3)

```

59      0057 1 %SBTTL 'Module Level Declarations'
60      0058 1
61      0059 1
62      0060 1 | TABLE OF CONTENTS:
63      0061 1
64      0062 1
65      0063 1 | INCLUDE FILES:
66      0064 1
67      0065 1
68      0066 1 LIBRARY 'NXPORT:XPORT';      ! XPORT Library
69      0067 1 REQUIRE 'REQ:RNODEF';      ! RUNOFF variant definitions
70      0198 1
71      U 0199 1 %IF DSRPLUS %THEN
72      U 0200 1 LIBRARY 'REQ:DPLLIB';      ! DSRPLUS BLISS Library
73      0201 1 %ELSE
74      0202 1 LIBRARY 'REQ:DSRLIB';      ! DSR BLISS Library
75      0203 1 %FI
76      0204 1
77      0205 1
78      0206 1 | MACROS:
79      0207 1
80      0208 1
81      0209 1 | EQUATED SYMBOLS:
82      0210 1
83      0211 1
84      0212 1 | OWN STORAGE:
85      0213 1
86      0214 1
87      0215 1 | EXTERNAL REFERENCES:
88      0216 1
89      0217 1
90      0218 1 EXTERNAL
91      0219 1     FLGT : FLAG TABLE,
92      0220 1     FS01 : FIXED_STRING,
93      0221 1     GCA : GCA_DEFINITION,
94      0222 1     IRA : FIXED_STRING,
95      0223 1     KHAR,
96      0224 1     NUMPRM : NUMPRM_DEFINE,
97      0225 1     SCA : SCA_DEFINITION;
98      0226 1
99      0227 1 EXTERNAL LITERAL      !Error messages
100     0228 1     RNFCM,
101     0229 1     RNFILC,
102     0230 1     RNFINM,
103     0231 1     RNFMNL,
104     0232 1     RNFMQS,
105     0233 1     RNFOST;
106     0234 1
107     0235 1 EXTERNAL ROUTINE
108     0236 1     ENDCMT,
109     0237 1     ERMA,
110     0238 1     GETQS,
111     0239 1     OUTJ,
112     0240 1     OUTNJ,
113     0241 1     PUS,
114     0242 1     SKPSEP;
```


REPEAT
V04-000

Processes the .REPEAT command.
Module Level Declarations

E 14
16-Sep-1984 01:31:44
14-Sep-1984 13:07:50

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

Page 4
(4)

```
: 116 0243 1 GLOBAL ROUTINE repeat (handler_code) =
: 117 0244 1
: 118 0245 1 ++
: 119 0246 1 FUNCTIONAL DESCRIPTION:
: 120 0247 1
: 121 0248 1 See the ABSTRACT, above.
: 122 0249 1
: 123 0250 1 FORMAL PARAMETERS:
: 124 0251 1
: 125 0252 1 HANDLER_CODE is a dummy argument, used only for conformance
: 126 0253 1 with other routines.
: 127 0254 1
: 128 0255 1 IMPLICIT INPUTS: None
: 129 0256 1
: 130 0257 1 IMPLICIT OUTPUTS: None
: 131 0258 1
: 132 0259 1 ROUTINE VALUE:
: 133 0260 1 COMPLETION CODES: None
: 134 0261 1
: 135 0262 1 SIDE EFFECTS: None
: 136 0263 1
: 137 0264 1 --
: 138 0265 1
: 139 0266 2 BEGIN
: 140 0267 2
: 141 0268 2 LOCAL
: 142 0269 2 hold_err_count,
: 143 0270 2 hold_ira: VECTOR [4],
: 144 0271 2 rpt_string: fixed_string [150],
: 145 0272 2 hold_khar,
: 146 0273 2 repeat_count,
: 147 0274 2 result;
: 148 0275 2
: 149 0276 2 IF NOT .num_result
: 150 0277 2 THEN
: 151 0278 2 BEGIN
: 152 0279 2 endcmt ();
: 153 0280 2 RETURN true;
: 154 0281 2 END;
: 155 0282 2
: 156 0283 2 IF .num_length EQL 0
: 157 0284 2 THEN
: 158 0285 2 BEGIN
: 159 0286 2 erma (rnfmnl, true);
: 160 0287 2 RETURN true;
: 161 0288 2 END;
: 162 0289 2
: 163 0290 2 IF .num_value LEQ 0
: 164 0291 2 THEN
: 165 0292 2 BEGIN
: 166 0293 2 erma (rnfinm, true);
: 167 0294 2 RETURN true;
: 168 0295 2 END;
: 169 0296 2
: 170 0297 2 skipsep (ira);
: 171 0298 2
: 172 0299 2 !Set up area to hold the repeat text
```

!Skip spaces, tabs, and a comma.

REPEAT
V04-000

Processes the .REPEAT command.
Module Level Declarations

F 14
16-Sep-1984 01:31:44
14-Sep-1984 13:07:50

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

Page 5
(4)

```

173 0300 2 fs_maxsize (rpt_string) = 150;
174 0301 2 fs_init (rpt_string);
175 0302 2
176 0303 2 result = getqs (rpt_string);          !Try to get what's to be repeated.
177 0304 2
178 0305 2 IF .result NEQ getqs_normal
179 0306 2 THEN
180 0307 2 !Analyze error a bit more closely.
181 0308 2 IF .result EQL getqs_too_long
182 0309 2 THEN
183 0310 2 BEGIN
184 0311 2   erma (rnfqst, false);
185 0312 2   RETURN true;
186 0313 2 END
187 0314 2 ELSE
188 0315 2 !Missing or improperly delimited quoted string.
189 0316 2 BEGIN
190 0317 2   erma (rnfmqs, true);
191 0318 2   RETURN true;
192 0319 2 END;
193 0320 2
194 0321 2 !
195 0322 2 !
196 C 0323 2 !
197 C 0324 2 !
198 C 0325 2 !
199 C 0326 2 !
200 C 0327 2 !
201 C 0328 2 !
202 C 0329 2 !
203 C 0330 2 !
204 C 0331 2 !
205 C 0332 2 !
206 C 0333 2 !
207 C 0334 2 !
208 C 0335 2 !
209 C 0336 2 !
210 C 0337 2 !
211 C 0338 2 !
212 C 0339 2 !
213 C 0340 2 !
214 C 0341 2 !
215 C 0342 2 !
216 C 0343 2 !
217 C 0344 2 !
218 C 0345 2 !
219 C 0346 2 !
220 0347 2 !
221 0348 2 !
222 0349 2 !
223 0350 2 !
224 0351 2 !
225 0352 2 !
226 0353 2 !
227 0354 2 !
228 0355 2 !
229 0356 2 !

XXXXXXXXXXXXXXXXXXXXXXXXXXXX
!Check to see if user is trying to repeat a command. If
!so, tell him it won't work and refuse to even try it.
!NOTE: This restriction may be lifted later, when instead
!of calling SCANT, PUS gets called instead.
BEGIN
LOCAL
  ptr,
  x;

ptr = .fs_start (rpt_string);          !Cop a peek at first character.
x = CH$RCHAR (.ptr);                  !...

IF .x EQL .flgt [con_flag, flag_character] !<CONTROL flag> ?
THEN
!User is trying to do something not yet allowed.
BEGIN
  erma (rnfilc, false);
  RETURN true;
END;

! ***** The above code will eventually change when we have fully
! ***** implemented repeated commands
END;

XXXXXXXXXXXXXXXXXXXXXXXXXXXX
!Save status of current input string, and
!substitute what the user wants repeated instead.
BEGIN
MAP
  ira : VECTOR;

```


REPEAT
V04-000

Processes the .REPEAT command.
Module Level Declarations

G 14
16-Sep-1984 01:31:44
14-Sep-1984 13:07:50

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

Page 6
(4)

```
: 230      0357 3      INCR i FROM 0 TO 3 DO
: 231      0358      hold_ira [.i] = .ira [.i];
: 232      0359
: 233      0360      hold_khar = .khar;      ! Preserve next character.
: 234      0361      hold_err_count = .gca_errcnt;      ! Remember error status.
: 235      0362      END;
: 236      0363
: 237      0364      repeat_count = .num_value;      ! Pick up repeat count.
: 238      0365
: 239      0366      INCR i FROM 1 TO .repeat_count DO
: 240      0367      BEGIN
: 241      0368          fs_maxsize (ira) = .fs_maxsize (rpt_string);      ! Point IRA to user's
: 242      0369          fs_length (ira) = .fs_length (rpt_string);      ! string for use by
: 243      0370          fs_start (ira) = .fs_start (rpt_string);      ! KCNS and PUS
: 244      0371          fs_next (ira) = .fs_start (rpt_string);
: 245      0372          kcns ();      ! Initialize scan.
: 246      0373
: 247      0374          pus (false);      ! Process 'uneated' string without using ENDWRD.
: 248      0375
: 249      0376      !
: 250      0377      !-----
: 251      0378      !
: 252      0379      scant ();      ! Scan user's string as if it were text.
: 253      0380      IF NOT .sca_fill      ! End of line means end of record if not filling.
: 254      0381      THEN
: 255      0382          IF .sca_justify
: 256      0383          THEN
: 257      0384              outj ()      ! No fill, justify
: 258      0385          ELSE
: 259      0386              outnj ();      ! No fill, no justify
: 260      0387      !
: 261      0388      !-----
: 262      0389      !
: 263      0390      IF .hold_err_count NEQ .gca_errcnt
: 264      0391      THEN
: 265      0392          EXITLOOP;      ! Abort processing if an error occurs.
: 266      0393      END;
: 267      0394
: 268      0395      ! Restore previous status of IRA, FS01 and KHAR
: 269      0396      BEGIN
: 270      0397      MAP
: 271      0398          ira : VECTOR;
: 272      0399      INCR i FROM 0 TO 3 DO
: 273      0400          ira [.i] = .hold_ira [.i];      ! Restore IRA
: 274      0401      khar = .hold_khar
: 275      0402      END
: 276      0403
: 277      0404      END;      ! End of REPEAT
```

.TITLE REPEAT Processes the .REPEAT command.
.IDENT \V04-000\

.EXTRN FLGT, FS01, GCA
.EXTRN IRA, KHAR, NUMPRM
.EXTRN SCA, RNFCM, RNFILC
.EXTRN RNFINM, RNFMNL, RNFMQS


```
.EXTRN RNFQST, ENDCMT, ERMA
.EXTRN GETQS, OUTJ, OUTNJ
.EXTRN PUS, SKPSEP, RINTES

.PSECT $CODE$,NOWRT,2

.ENTRY REPEAT, Save R2,R3,R4,R5,R6,R7,R8,R9 : 0243
MOVAB GCA+196, R9
MOVAB NUMPRM+4, R8
MOVAB KHAR, R7
MOVAB IRA, R6
MOVAB -184(SP), SP
BLBS NUMPRM, 1$ : 0276
CALLS #0, ENDCMT : 0279
BRB 6$ : 0280
TSTL NUMPRM+12 : 0283
BNEQ 2$
PUSHL #1 : 0286
PUSHL #RNF MNL
BRB 5$
TSTL NUMPRM+4 : 0290
BGTR 3$
PUSHL #1 : 0293
PUSHL #RNF INM
BRB 5$
PUSHL R6 : 0297
CALLS #1, SKPSEP
MOVZBL #150, RPT_STRING+8 : 0300
CLRL RPT_STRING+12 : 0301
MOVAB RPT_STRING+16, RPT_STRING
MOVL RPT_STRING, RPT_STRING+4
PUSHL SP : 0303
CALLS #1, GETQS : 0305
TSTL RESULT : 0308
BEQL 7$
CMPL RESULT, #2 : 0311
BNEQ 4$
CLRL -(SP)
PUSHL #RNFQST : 0317
BRB 5$
PUSHL #1
PUSHL #RNF MQS
CALLS #2, ERMA : 0318
MOVL #1, R0 : 0316
RET : 0357
CLRL I : 0358
MOVL IRA[I], HOLD_IRA[I]
AOBLEQ #3, I, 8$
MOVL KHAR, HOLD_KHAR : 0360
MOVL GCA+196, HOLD_ERR_COUNT : 0361
MOVL NUMPRM+4, REPEAT_COUNT : 0364
CLRL I : 0390
BRB 12$
MOVQ RPT_STRING+8, IRA+8 : 0368
MOVL RPT_STRING, IRA : 0370
MOVL RPT_STRING, IRA+4 : 0371
TSTL IRA+12 : 0372
```

59	00000000G	EF	9E	00002		
58	00000000G	EF	9E	00009		
57	00000000G	EF	9E	00010		
56	00000000G	EF	9E	00017		
5E	FF48	CE	9E	0001E		
09	FC	A8	E8	00023		
00000000G	EF	00	FB	00027		
		61	11	0002E		
	08	A8	D5	00030	1\$:	
		0A	12	00033		
		01	DD	00035		
	00000000G	8F	DD	00037		
		4B	11	0003D		
		68	D5	0003F	2\$:	
		0A	14	00041		
		01	DD	00043		
	00000000G	8F	DD	00045		
		3D	11	0004B		
		56	DD	0004D	3\$:	
00000000G	EF	01	FB	0004F		
08	AE	96	8F	9A	00056	
		0C	AE	D4	0005B	
	6E	10	AE	9E	0005E	
04	AE		6E	D0	00062	
			5E	DD	00066	
00000000G	EF		01	FB	00068	
			50	D5	0006F	
			22	13	00071	
	02		50	D1	00073	
			0A	12	00076	
			7E	D4	00078	
	00000000G		8F	DD	0007A	
			08	11	00080	
			01	DD	00082	4\$:
	00000000G		8F	DD	00084	
00000000G	EF		02	FB	0008A	5\$:
	50		01	D0	00091	6\$:
			04	00094		
			50	D4	00095	7\$:
F6	F0 AD40	6640	D0	00097	8\$:	
	50		03	F3	0009D	
	53		67	D0	000A1	
	55		69	D0	000A4	
	54		68	D0	000A7	
			52	D4	000AA	
			33	11	000AC	
08	A6	08	AE	7D	000AE	9\$:
	66		6E	D0	000B3	
04	A6		6E	D0	000B6	
		0C	A6	D5	000BA	

REPEAT
V04-000

Processes the .REPEAT command.
Module Level Declarations

I 14
16-Sep-1984 01:31:44
14-Sep-1984 13:07:50

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

Page 8
(4)

		0A	14	000BD	BGTR	10\$	
		8F	9A	000BF	MOVZBL	#RINTES, KHAR	
	OC	A6	01	CE 000C3	MNEGL	#1, IRA+12	
		67	0A	11 000C7	BRB	11\$	
			04	B6 9A 000C9	MOVZBL	@IRA+4, KHAR	
			04	A6 D6 000CD	INCL	IRA+4	
			OC	A6 D7 000D0	DECL	IRA+12	
			7E	D4 000D3	CLRL	-(SP)	
	00000000G	EF	01	FB 000D5	CALLS	#1, PUS	0374
		69	55	D1 000DC	CML	HOLD_ERR_COUNT, GCA+196	0390
			04	12 000DF	BNEQ	13\$	
	C9	52	54	F3 000E1	AOBLEQ	REPEAT_COUNT, I, 9\$	0366
			50	D4 000E5	CLRL	I	0399
		6640	FO AD40	D0 000E7	MOVL	HOLD IRA[I], IRA[I]	0400
	F6	50	03	F3 000ED	AOBLEQ	#3, I, 14\$	
		67	53	D0 000F1	MOVL	HOLD_KHAR, KHAR	0401
		50	53	D0 000F4	MOVL	HOLD_KHAR, RO	0404
			04	000F7	RET		

; Routine Size: 248 bytes, Routine Base: \$CODE\$ + 0000

; 278 0405 1
; 279 0406 1 END
; 280 0407 0 ELUDOM

!End of module

PSECT SUMMARY

Name	Bytes	Attributes
\$CODE\$	248	NOVEC,NOWRT, RD , EXE,NOSHR, LCL, REL, CON,NOPI,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	26	2	86	00:00.3

COMMAND QUALIFIERS

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

REPEAT
V04-000

Processes the .REPEAT command.
Module Level Declarations

J 14
16-Sep-1984 01:31:44
14-Sep-1984 13:07:50

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

Page 9
(4)

; Size: 248 code + 0 data bytes
; Run Time: 00:06.5
; Elapsed Time: 00:16.6
; Lines/CPU Min: 3768
; Lexemes/CPU-Min: 15046
; Memory Used: 78 pages
; Compilation Complete

0347 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY