

BBBBBBBBBBBB		AAAAAAA		SSSSSSSSSS		RRRRRRRRRR		TTTTTTTTTTTT		LLL
BBBBBBBBBBBB		AAAAAAA		SSSSSSSSSS		RRRRRRRRRR		TTTTTTTTTTTT		LLL
BBBBBBBBBBBB		AAAAAAA		SSSSSSSSSS		RRRRRRRRRR		TTTTTTTTTTTT		LLL
BBB	BBB	AAA	AAA	SSS		RRR	RRR	TTT		LLL
BBB	BBB	AAA	AAA	SSS		RRR	RRR	TTT		LLL
BBB	BBB	AAA	AAA	SSS		RRR	RRR	TTT		LLL
BBB	BBB	AAA	AAA	SSS		RRR	RRR	TTT		LLL
BBB	BBB	AAA	AAA	SSS		RRR	RRR	TTT		LLL
BBB	BBB	AAA	AAA	SSS		RRR	RRR	TTT		LLL
BBBBBBBBBBBB		AAA	AAA	SSSSSSSS		RRRRRRRRRR		TTT		LLL
BBBBBBBBBBBB		AAA	AAA	SSSSSSSS		RRRRRRRRRR		TTT		LLL
BBBBBBBBBBBB		AAA	AAA	SSSSSSSS		RRRRRRRRRR		TTT		LLL
BBB	BBB	AAAAAAAAAAAA			SSS	RRR	RRR	TTT		LLL
BBB	BBB	AAAAAAAAAAAA			SSS	RRR	RRR	TTT		LLL
BBB	BBB	AAAAAAAAAAAA			SSS	RRR	RRR	TTT		LLL
BBB	BBB	AAA	AAA		SSS	RRR	RRR	TTT		LLL
BBB	BBB	AAA	AAA		SSS	RRR	RRR	TTT		LLL
BBB	BBB	AAA	AAA		SSS	RRR	RRR	TTT		LLL
BBB	BBB	AAA	AAA		SSS	RRR	RRR	TTT		LLL
BBB	BBB	AAA	AAA		SSS	RRR	RRR	TTT		LLL
BBBBBBBBBBBB		AAA	AAA	SSSSSSSSSS		RRR	RRR	TTT		LLLLLLLLLLLL
BBBBBBBBBBBB		AAA	AAA	SSSSSSSSSS		RRR	RRR	TTT		LLLLLLLLLLLL
BBBBBBBBBBBB		AAA	AAA	SSSSSSSSSS		RRR	RRR	TTT		LLLLLLLLLLLL

```
BBBBBBBBB      AAAAAA      SSSSSSSS      RRRRRRRR      AAAAAA      NN      NN      DDDDDDDD      000000      MM      MM
BBBBBBBBB      AAAAAA      SSSSSSSS      RRRRRRRR      AAAAAA      NN      NN      DDDDDDDD      000000      MM      MM
BB      BB      AA      AA      SS      RR      RR      AA      AA      NN      NN      DD      DD      00      00      MMMM      MMMM
BB      BB      AA      AA      SS      RR      RR      AA      AA      NN      NN      DD      DD      00      00      MMMM      MMMM
BB      BB      AA      AA      SS      RR      RR      AA      AA      NNNN      NN      DD      DD      00      00      MM      MM
BB      BB      AA      AA      SS      RR      RR      AA      AA      NNNN      NN      DD      DD      00      00      MM      MM
BBBBBBBBB      AA      AA      SSSSSS      RRRRRRRR      AA      AA      NN      NN      DD      DD      00      00      MM      MM
BBBBBBBBB      AA      AA      SSSSSS      RRRRRRRR      AA      AA      NN      NN      DD      DD      00      00      MM      MM
BB      BB      AAAAAAAAAA      SS      RR      RR      AAAAAAAAAA      NN      NNNN      DD      DD      00      00      MM      MM
BB      BB      AAAAAAAAAA      SS      RR      RR      AAAAAAAAAA      NN      NNNN      DD      DD      00      00      MM      MM
BB      BB      AA      AA      SS      RR      RR      AA      AA      NN      NN      DD      DD      00      00      MM      MM
BB      BB      AA      AA      SS      RR      RR      AA      AA      NN      NN      DD      DD      00      00      MM      MM
BBBBBBBBB      AA      AA      SSSSSSSS      RR      RR      AA      AA      NN      NN      DDDDDDDD      000000      MM      MM
BBBBBBBBB      AA      AA      SSSSSSSS      RR      RR      AA      AA      NN      NN      DDDDDDDD      000000      MM      MM

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
```


(2) 62
(3) 93
(4) 142
(5) 185

DECLARATIONS
BASSRANDOMIZE - Initialize random number seed
BASS\$RAND_INIT - Initialize random number seed
BASS\$RND_F_R1 - BASIC Random Number Generator

```
0000 1      .TITLE BAS$RANDOM - BASIC Random Number Support
0000 2      .IDENT /1-005/ ; File: BASRANDOM.MAR
0000 3
0000 4
0000 5 *****
0000 6
0000 7 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0000 8 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0000 9 * ALL RIGHTS RESERVED.
0000 10
0000 11 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0000 12 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0000 13 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0000 14 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0000 15 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0000 16 * TRANSFERRED.
0000 17
0000 18 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0000 19 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0000 20 * CORPORATION.
0000 21
0000 22 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0000 23 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0000 24
0000 25 *****
0000 26
0000 27
0000 28
0000 29 ++
0000 30 FACILITY: BASIC Support Library
0000 31
0000 32 ABSTRACT:
0000 33
0000 34 This module provides support for all BASIC random number
0000 35 functions. The seed is kept in OWN storage in this module.
0000 36 There are 2 entry points. The first initializes the seed
0000 37 based on the time of day to prevent getting the same sequence
0000 38 each time a program is run. The second picks up the seed
0000 39 and generates a random number.
0000 40 This is a general random number generator. It is
0000 41 of the multiplicative congruential type, and hence
0000 42 is fast, although prone to certain classes of
0000 43 non-random sequences.
0000 44
0000 45 ENVIRONMENT: any access mode, normally user mode
0000 46 modular, non-ASf reentrant procedure
0000 47 --
0000 48 +
0000 49 AUTHOR: R. Will, CREATION DATE: 29-Nov-78
0000 50
0000 51 MODIFIED BY:
0000 52
0000 53 R. Will, 29-Nov-78: VERSION 01
0000 54 01 - Original creation.
0000 55 1-002 - Add " to the PSECT directives. JBS 22-DEC-78
0000 56 1-003 - Add BAS$$RAND INIT. JBS 04-JUN-1979
0000 57 1-004 - Don't cause integer overflow. JBS 05-AUG-1979
```


BAS\$RANDOM
1-005

- BASIC Random Number Support

L 3

16-SEP-1984 00:01:20 VAX/VMS Macro V04-00
6-SEP-1984 10:35:11 [BASRTL.SRC]BASRANDOM.MAR;1

Page 2
(1)

```
0000 58 ; 1-005 - Start the generator at 1C5983F7, so it doesn't give 0 the
0000 59 ; first time it is called. JBS 01-SEP-1979
0000 60 ;-
```

```

0000 62      .SBTTL  DECLARATIONS
0000 63
0000 64 :
0000 65 : INCLUDE FILES:
0000 66 :
0000 67 :     NONE
0000 68 :
0000 69 : EXTERNAL DECLARATIONS:
0000 70 :
0000 71 :     .DSABL  GBL                ; Prevents undeclared symbols
0000 72 :                                     ; from being automatically global
0000 73 :     .EXTRN  SYS$GETTIM           ; system services, current time
0000 74 :
0000 75 : MACROS:
0000 76 :
0000 77 :     NONE
0000 78 :
0000 79 : EQUATED SYMBOLS:
0000 80 :
0000 81 :
0000 82 : OWN STORAGE:
0000 83 :
00000000 84 :     .PSECT  _BAS$DATA,PIC,NOSHR,WRT,LONG,NOEXE
0000 85
1C5983F7 0000 86 SEED:  .LONG  ^X1C5983F7        ; Initial value of seed
0004 87 :
0004 88 :
0004 89 : PSECT DECLARATIONS:
0004 90 :
00000000 91 :     .PSECT  _BAS$CODE,PIC,USR,CON,REL,LCL,RD,SHR,NOWRT,LONG,EXE

```



```
0000 93 .SBTTL BASSRANDOMIZE - Initialize random number seed
0000 94 :++
0000 95 : FUNCTIONAL DESCRIPTION:
0000 96 :
0000 97 : This routine initializes the seed by getting the current system
0000 98 : time and adding the low 32 bits to the high 32 bits to get the seed.
0000 99 :
0000 100 : CALLING SEQUENCE:
0000 101 :
0000 102 : CALL BASSRANDOMIZE ( )
0000 103 :
0000 104 : INPUT PARAMETERS:
0000 105 :
0000 106 : NONE
0000 107 :
0000 108 : IMPLICIT INPUTS:
0000 109 :
0000 110 : NONE
0000 111 :
0000 112 : OUTPUT PARAMETERS:
0000 113 :
0000 114 : NONE
0000 115 :
0000 116 : IMPLICIT OUTPUTS:
0000 117 :
0000 118 : SEED.wlu
0000 119 :
0000 120 : FUNCTION VALUE:
0000 121 : COMPLETION CODES:
0000 122 :
0000 123 : NONE
0000 124 :
0000 125 : SIDE EFFECTS:
0000 126 :
0000 127 : SEED is altered
0000 128 :
0000 129 :--
0000 130 :
0000 131 .ENTRY BASSRANDOMIZE , ^M<> ; Entry point
0002 132 ;
7E 7C 0002 133 CLRQ -(SP) ; space on stack for time
0004 134 $GETTIM_S (SP) ; get current time
000D 135 ; as top 2 words on stack
0000 136 ADDL3 (SP)+,(SP)+,SEED ; add the two words of
0015 137 ; current time to get
0015 138 ; randomized seed
0015 139 ; and clean up stack
04 0015 140 RET
```



```
0016 142 .SBTTL BASS$RAND_INIT - Initialize random number seed
0016 143 :++
0016 144 : FUNCTIONAL DESCRIPTION:
0016 145 :
0016 146 : This routine sets the seed to 1C5983F7. It is used by the RUN
0016 147 : command when the environment is initialized.
0016 148 :
0016 149 : CALLING SEQUENCE:
0016 150 :
0016 151 : CALL BASS$RAND_INIT ( )
0016 152 :
0016 153 : INPUT PARAMETERS:
0016 154 :
0016 155 : NONE
0016 156 :
0016 157 : IMPLICIT INPUTS:
0016 158 :
0016 159 : NONE
0016 160 :
0016 161 : OUTPUT PARAMETERS:
0016 162 :
0016 163 : NONE
0016 164 :
0016 165 : IMPLICIT OUTPUTS:
0016 166 :
0016 167 : SEED.wlu
0016 168 :
0016 169 : FUNCTION VALUE:
0016 170 : COMPLETION CODES:
0016 171 :
0016 172 : NONE
0016 173 :
0016 174 : SIDE EFFECTS:
0016 175 :
0016 176 : SEED is altered
0016 177 :
0016 178 :--
0016 179 :
0000 0016 180 .ENTRY BASS$RAND_INIT , ^M<> ; Entry point
0018 181
00000000'EF 1C5983F7 8F D0 0018 182 MOVL #^X1C5983F7, SEED ; Initialize the seed
04 0023 183 RET ; and return
```



```
0024 185 .SBTTL BAS$RND_F_R1 - BASIC Random Number Generator
0024 186 :++
0024 187 : FUNCTIONAL DESCRIPTION:
0024 188 :
0024 189 : RANDOM - BASIC Pseudo Random Number Generator
0024 190 :
0024 191 : Call the function BAS$RND_F ( ) to obtain the
0024 192 : next pseudo-random number. The seed is updated by
0024 193 : the function automatically as a side effect. The
0024 194 : result is a floating point number that is uniformly
0024 195 : distributed in the range 0.0 inclusive to 1.0 exclusive.
0024 196 : There are no restrictions on the seed, although
0024 197 : it should be initialized to different values on
0024 198 : separate runs.
0024 199 :
0024 200 : The algorithm used is to update the seed as:
0024 201 :
0024 202 :     seed = 69069. * seed + 1      (mod 2**32)
0024 203 :
0024 204 : and then to convert the seed to floating point.
0024 205 :
0024 206 : Note, because the result is never 1.0, a simple
0024 207 : way to get a uniform random integer selector is
0024 208 : to multiply by the number of cases. For example
0024 209 : if a uniform choice among 5 situations is to be
0024 210 : made, then the following BASIC statement will
0024 211 : work:
0024 212 :     ON 1+INT(5.0*RND) 1,2,3,4,5
0024 213 :
0024 214 : Note that the explicit INT is necessary before
0024 215 : adding 1 in order to avoid a possible rounding
0024 216 : during the normalization after the floating add.
0024 217 :
0024 218 : This is a general random number generator. It is
0024 219 : of the multiplicative congruential type, and hence
0024 220 : is fast, although prone to certain classes of
0024 221 : non-random sequences. This non-random behavior
0024 222 : typically arises when considering triples of
0024 223 : numbers generated by this method.
0024 224 :
0024 225 : For more information on congruential generators,
0024 226 : see:
0024 227 :     Random Number Generation (pp. 1192-1197)
0024 228 :     by G. Marsaglia
0024 229 :
0024 230 : in: Encyclopedia of Computer Science
0024 231 :     edited by Anthony Ralston
0024 232 :     Petrocelli (New York, 1976)
0024 233 :
0024 234 : CALLING SEQUENCE:
0024 235 :
0024 236 :     JSB result.wf.v = BAS$RND_F_R1 ( )
0024 237 :
0024 238 : INPUT PARAMETERS:
0024 239 :
0024 240 :     NONE
0024 241 :
```

```
0024 242 : IMPLICIT INPUTS:
0024 243 :
0024 244 :     SEED.mlu
0024 245 :
0024 246 : OUTPUT PARAMETERS:
0024 247 :
0024 248 :     NONE
0024 249 :
0024 250 : IMPLICIT OUTPUTS:
0024 251 :
0024 252 :     SEED.mlu
0024 253 :
0024 254 : FUNCTION VALUE
0024 255 :
0024 256 :     Returns in R0 a single-precision floating point value between
0024 257 :     0.0 inclusive and 1.0 exclusive.
0024 258 :
0024 259 : SIDE EFFECTS:
0024 260 :
0024 261 :     The value of SEED is altered
0024 262 :
0024 263 : --
0024 264 : BAS$RND_F_R1:: ;JSB entry
0024 265 : +
0024 266 : Do the multiply in a way that will not cause an integer
0024 267 : overflow.
0024 268 : -
01 00010DCD 8F 00000000'EF 7A 0024 269 EMUL SEED,#69069,#1,R0 ; compute product in R0/R1
0030 270
0031 271 MOVL R0,SEED ; ignore high-order bits of product
0038 272
0038 273 : +
0038 274 : The next instructions convert the seed from unsigned integer
0038 275 : to floating point in the range 0.0 to 1.0 exclusive.
0038 276 : -
50 00000000'EF 18 08 EF 0038 277 EXTZV #8,#24,SEED,R0 ;Get the most significant bits
0041 278 ; of the seed in the range
0041 279 ; 0 .. (2**24)-1
50 50 4E 0041 280 CVTLF R0,R0 ;Convert to floating without
0044 281 ; rounding. The result is
0044 282 ; positive and in the range
0044 283 ; 0.0 .. (2.0**24)-1.0
0044 284
0044 285 : +
0044 286 : If this were to be placed as an inline expansion, then
0044 287 : MULF #^X00003480,R0 could replace the next two instructions.
0044 288 : -
0044 289
0044 290 BEQL 10$ ;If zero, already correct
50 0C00 05 13 0044 291 SUBW #24a7,R0 ;DIVF #^f2.0**24
0046 292 ; the result is now in the
004B 293 ; range 0.0 .. 1.0 exclusive
004B 294
004B 295 10$: RSB
004C 296
004C 297 .END
```


BAS\$RANDOM
Symbol table

- BASIC Random Number Support

E 4

16-SEP-1984 00:01:20
6-SEP-1984 10:35:11

VAX/VMS Macro V04-00
[BASRTL.SRC]BASRANDOM.MAR;1

Page 8
(5)

BAS\$RAND_INIT	00000016	RG	02
BAS\$RANDOMIZE	00000000	RG	02
BAS\$RND_F_R1	00000024	RG	02
SEED	00000000	R	01
SY\$GETTIM	*****	GX	00

+-----+
! Psect synopsis !
+-----+

PSECT name	Allocation	PSECT No.	Attributes													
. ABS	00000000 (0.)	00 (0.)	NOPIC	USR	CON	ABS	LCL	NOSHR	NOEXE	NORD	NOWRT	NOVEC	BYTE			
_BAS\$DATA	00000004 (4.)	01 (1.)	PIC	USR	CON	REL	LCL	NOSHR	NOEXE	RD	WRT	NOVEC	LONG			
_BAS\$CODE	0000004C (76.)	02 (2.)	PIC	USR	CON	REL	LCL	SHR	EXE	RD	NOWRT	NOVEC	LONG			

+-----+
! Performance indicators !
+-----+

Phase	Page faults	CPU Time	Elapsed Time
Initialization	32	00:00:00.08	00:00:00.65
Command processing	113	00:00:00.45	00:00:03.67
Pass 1	102	00:00:00.64	00:00:02.68
Symbol table sort	0	00:00:00.00	00:00:00.00
Pass 2	61	00:00:00.56	00:00:01.19
Symbol table output	2	00:00:00.01	00:00:00.01
Psect synopsis output	3	00:00:00.02	00:00:00.02
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	315	00:00:01.76	00:00:08.22

The working set limit was 1050 pages.
3063 bytes (6 pages) of virtual memory were used to buffer the intermediate code.
There were 10 pages of symbol table space allocated to hold 5 non-local and 1 local symbols.
297 source lines were read in Pass 1, producing 16 object records in Pass 2.
2 pages of virtual memory were used to define 2 macros.

+-----+
! Macro library statistics !
+-----+

Macro library name	Macros defined
_\$255\$DUA28:[SYSLIB]STARLET.MLB;2	2

12 GETS were required to define 2 macros.

There were no errors, warnings or information messages.

MACRO/ENABLE=SUPPRESSION/DISABLE=(GLOBAL,TRACEBACK)/LIS=LIS\$:BASRANDOM/OBJ=OBJ\$:BASRANDOM MSRC\$:BASRANDOM/UPDATE=(ENH\$:BASRANDOM)

0030 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

