

File created: 16-May-90 14:21:56 {DSK}<usr>local>lde>lispcore>sources>CMLRAND.;2

changes to: (IL:VARS IL:CMLRANDCOMS)

previous date: 21-Jan-88 11:43:47 {DSK}<usr>local>lde>lispcore>sources>CMLRAND.;1

Read Table: XCL

Package: LISP

Format: XCCS

; Copyright (c) 1986, 1987, 1988, 1990 by Venue & Xerox Corporation. All rights reserved.

```
(IL:RPAQQ IL:CMLRANDCOMS ((IL:STRUCTURES RANDOM-STATE)
  (IL:VARIABLES %RANDOM-SIZE)
  (IL:FUNCTIONS %MAKE-RANDOM-ARRAY %PRINT-RANDOM-STATE %RANDOM MAKE-RANDOM-STATE RANDOM
  )
  (IL:VARIABLES *RANDOM-STATE*)
  (IL:PROP (IL:FILETYPE IL:MAKEFILE-ENVIRONMENT)
    IL:CMLRAND)
  (IL:DECLARE\ : IL:DONTCOPY IL:DOEVAL@COMPILE (IL:LOCALVARS . T))))
```

```
(DEFSTRUCT (RANDOM-STATE (:CONSTRUCTOR %MAKE-RANDOM-STATE)
  (:PRINT-FUNCTION %PRINT-RANDOM-STATE)
  (:COPIER NIL))
```

```
(I 0)
(J 30)
(ARRAY (%MAKE-RANDOM-ARRAY)))
```

```
(DEFCONSTANT %RANDOM-SIZE 55)
```

```
(DEFUN %MAKE-RANDOM-ARRAY (&OPTIONAL SEED1 SEED2)
  (MAKE-ARRAY %RANDOM-SIZE :INITIAL-CONTENTS
    (LET ((RANDOM-LIST ' (53375 47430 1274 55702 61592 27723 11236 16824 35838 62289 11525 37822 34676 105
      58750 27759 9988 4217 56951 30292 24550 1397 54588 54264 43300 3862 39006
      11386 52259 1055 955 16320 19910 58470 3263 64657 1704 17373 56820 17255
      51637 47962 26272 4464 2884 51773 39422 64835 57733 34919 5315 12110 15116
      10133 10816)))
      (RANDOM-CONST-A (OR SEED1 (IL:CLOCK)))
      (RANDOM-CONST-B (OR SEED2 (IL:IDATE))))
    (MAPCAR #' (LAMBDA (X)
      (SETQ RANDOM-CONST-A (LOGAND RANDOM-CONST-A MOST-POSITIVE-FIXNUM))
      (LOGXOR X (SETQ RANDOM-CONST-B (PROG1 (LOGAND (+ (* RANDOM-CONST-A 19869)
        RANDOM-CONST-A)
        MOST-POSITIVE-FIXNUM)
      (SETQ RANDOM-CONST-A RANDOM-CONST-B))))))
      RANDOM-LIST))))
```

```
(DEFUN %PRINT-RANDOM-STATE (STATE STREAM PRINT-LEVEL)
  (LET ((XCL:*PRINT-STRUCTURE* T)
    (*PRINT-ARRAY* T)
    (DEFAULT-STRUCTURE-PRINTER STATE STREAM PRINT-LEVEL)))
```

```
(DEFUN %RANDOM (STATE)
```

```
;; This function implements the XRAND subroutine described in Stanford memo STAN-CS-77-601, Analysis of Additive Random Number
;; Generators, by John F. Reiser, on p 28.0. The numbers are stored as 16 bit binary fractions (i.e. the decimal point is on the left of the 16-bit
;; quantity)
```

```
(LET ((I (RANDOM-STATE-I STATE))
  (J (RANDOM-STATE-J STATE))
  (ARRAY (RANDOM-STATE-ARRAY STATE))
  RV)
  (SETQ RV (LOGAND (- (AREF ARRAY I)
    (AREF ARRAY J))
    MOST-POSITIVE-FIXNUM))
  (SETF (AREF ARRAY I)
  RV)
  (SETQ I (1+ I))
  (IF (EQ I %RANDOM-SIZE)
    (SETQ I 0))
  (SETQ J (1+ J))
  (IF (EQ J %RANDOM-SIZE)
    (SETQ J 0))
  (SETF (RANDOM-STATE-I STATE)
  I)
  (SETF (RANDOM-STATE-J STATE)
  J)
  RV))
```

```
(DEFUN MAKE-RANDOM-STATE (&OPTIONAL (STATE *RANDOM-STATE*))
```

:: Make a random state object. If State is not supplied, return a copy of the default random state. If State is a random state, then return a copy of it.
:: If state is T then return a random state generated from the universal time.

```

(COND
  ((EQ STATE T)
   (%MAKE-RANDOM-STATE))
  ((RANDOM-STATE-P STATE)
   (%MAKE-RANDOM-STATE :I (RANDOM-STATE-I STATE)
    :J
    (RANDOM-STATE-J STATE)
    :ARRAY
    (XCL:COPY-ARRAY (RANDOM-STATE-ARRAY STATE))))
  (T (ERROR "Not a random-state: ~S" STATE)))

(DEFUN RANDOM (NUMBER &OPTIONAL (STATE *RANDOM-STATE*))
  (IF (NOT (> NUMBER 0))
      (ERROR "Not a positive number: ~s" NUMBER))
  (LET ((RV (%RANDOM STATE))
        (TYPECASE NUMBER
         (FIXNUM (IF (EQ NUMBER MOST-POSITIVE-FIXNUM)
                     RV
                     (IL:IREMAINDER RV NUMBER)))
         (FLOAT (LET ((FNUMBER (FLOAT NUMBER))
                     (DECLARE (TYPE FLOAT FNUMBER)
                      (SETQ FNUMBER (* FNUMBER (IL:FQUOTIENT (FLOAT RV)
                                                                (FLOAT (1+ MOST-POSITIVE-FIXNUM)))))))
          (INTEGER (DO ((TOT RV (+ (ASH TOT 16)
                                   (%RANDOM STATE)))
                       (END (ASH NUMBER -16)
                            (ASH END -16)))
                      ((EQ 0 END)
                       (REM TOT NUMBER))))))
      (T (ERROR 'XCL:TYPE-MISMATCH :EXPECTED-TYPE '(OR INTEGER FLOAT)
                :NAME NUMBER :VALUE NUMBER :MESSAGE "an integer or a float"))))

(DEFPARAMETER *RANDOM-STATE* (%MAKE-RANDOM-STATE))

(IL:PUTPROPS IL:CMLRAND IL:FILETYPE COMPILE-FILE)

(IL:PUTPROPS IL:CMLRAND IL:MAKEFILE-ENVIRONMENT (:READTABLE "XCL" :PACKAGE "LISP"))

(IL:DECLARE\ : IL:DONTCOPY IL:DOEVAL@COMPILE

(IL:DECLARE\ : IL:DOEVAL@COMPILE IL:DONTCOPY

(IL:LOCALVARS . T)
)
)

(IL:PUTPROPS IL:CMLRAND IL:COPYRIGHT ("Venue & Xerox Corporation" 1986 1987 1988 1990))

```

FUNCTION INDEX

%MAKE-RANDOM-ARRAY1 %RANDOM1 RANDOM2
%PRINT-RANDOM-STATE1 MAKE-RANDOM-STATE1

PROPERTY INDEX

IL:CMLRAND2

VARIABLE INDEX

RANDOM-STATE2

CONSTANT INDEX

%RANDOM-SIZE1

STRUCTURE INDEX

RANDOM-STATE1
