File created: 24-Jul-85 17:42:37 **{DSK}<LISPFILES>TESTER>SOURCES>RANDOM-GENERATOR.;2**

changes to: (FNS TEST.GENERATE-RANDOM)

previous date: 19-Jul-85 11:24:20 {DSK}<LISPFILES>TESTER>SOURCES>RANDOM-GENERATOR.;1

Read Table: OLD-INTERLISP-FILE

Package: INTERLISP

Format: MCCS

(* * Copyright (c) 1985 by Xerox Corporation. All rights reserved.)

```
(RPAQQ RANDOM-GENERATORCOMS
       ((FNS TEST.GENERATE-RANDOM TEST.GENERATE-LIST-OF-ITEMS TEST.RANDOM-SELECTION)
        (INITVARS (TEST.TYPES-OF-RANDOM-OBJECTS (QUOTE (INTEGER SHORT-SIMPLE-LIST SHORT-LIST CHARACTER)))
                  (TEST.MIN-INTEGER MIN.FIXP)
                  (TEST.MAX-INTEGER MAX.FIXP)
                  (TEST.VERY-SMALL 4)
                  (TEST.SMALLP MAX.SMALLP)
                  (TEST.SMALL 1000)
                  (TEST.LARGE (IDIFFERENCE MAX.FIXP 1000))
                  (TEST.MAX-DEPTH 4)
                  (TEST.MAX-CHARACTER-CODE 255))))

(DEFINEQ

(TEST.GENERATE-RANDOM
  [LAMBDA (OBJECT-SPECIFICATIONS)                    (* sm "24-Jul-85 15:57")
    (SELECTQ (if (LISTP OBJECT-SPECIFICATIONS)
                 then (CAR OBJECT-SPECIFICATIONS)
                 else OBJECT-SPECIFICATIONS)
        (INTEGER (RAND TEST.MIN-INTEGER TEST.MAX-INTEGER))
        (SPECIAL-INTEGER
            (TEST.RANDOM-SELECTION (QUOTE (0 1 -1))))
        (BOUND-INTEGER
            (RAND (CADR OBJECT-SPECIFICATIONS)
                  (CADDR OBJECT-SPECIFICATIONS)))
        (LARGE-INTEGER
            (TEST.RANDOM-SELECTION (LIST (RAND (IMINUS TEST.LARGE)
                                               TEST.MIN-INTEGER)
                                         (RAND TEST.LARGE TEST.MAX-INTEGER))))
        (SMALL-INTEGER
            (TEST.RANDOM-SELECTION (LIST (RAND (IMINUS TEST.SMALL)
                                               0)
                                         (RAND 0 TEST.SMALL))))
        (SMALL-POSITIVE-INTEGER
            (RAND 1 TEST.SMALL))
        (SMALL-NON-NEGATIVE-INTEGER
            (RAND 0 TEST.SMALL))
        (SMALL-NEGATIVE-INTEGER
            (RAND (IMINUS TEST.SMALL)
                  -1))
        (SMALL-NON-POSITIVE-INTEGER
            (RAND (IMINUS TEST.SMALL)
                  0))
        (VERY-SMALL-POSITIVE-INTEGER
            (RAND 1 TEST.VERY-SMALL))
        (VERY-SMALL-NON-NEGATIVE-INTEGER
            (RAND 0 TEST.VERY-SMALL))
        (VERY-SMALL-NEGATIVE-INTEGER
            (RAND (IMINUS TEST.VERY-SMALL)
                  -1))
        (VERY-SMALL-NON-POSITIVE-INTEGER
            (RAND (IMINUS TEST.VERY-SMALL)
                  0))
        (VERY-LARGE-POSITIVE-INTEGER
            (RAND (IDIFFERENCE TEST.MAX-INTEGER 1000)
                  TEST.MAX-INTEGER))
        (VERY-LARGE-NEGATIVE-INTEGER
            (RAND TEST.MIN-INTEGER (IDIFFERENCE TEST.MIN-INTEGER 1000)))
        (POSITIVE-INTEGER
            (RAND 1 TEST.MAX-INTEGER))
```

```
        (NON-NEGATIVE-INTEGER
            (RAND 0 TEST.MAX-INTEGER))
        (NEGATIVE-INTEGER
            (RAND TEST.MIN-INTEGER -1))
        (NON-POSITIVE-INTEGER
            (RAND TEST.MIN-INTEGER 0))
        (BIGNUM [PACK (CONS (TEST.RANDOM-SELECTION (QUOTE (- "")))
                            (CONS (RAND 0 9)
                                  (for I from 1 to (RAND 20 99) collect (RAND 0 9])
        (POSITIVE-BIGNUM
            [PACK (CONS (RAND 0 9)
                        (for I from 1 to (RAND 20 99) collect (RAND 0 9])
        (SPECIAL-BIGNUM
            (TEST.RANDOM-SELECTION (QUOTE (16383 16382 16384 32767 32766 32768 65535 65534 65536
                                           16777215 16777214 16777216 134217727 134217726
                                           134217728))))
        (POSITIVE-POWEROF10-BIGNUM
            (PACK (CONS 1 (for I from 1 to (RAND 20 99) collect 0))))
        (WINDOW (CREATEW (TEST.GENERATE-RANDOM (QUOTE REGION))
                         (TEST.RANDOM-SELECTION (LIST NIL "DUMMY"))
                         (RAND 0 20)))
        (REGION [PROG (LEFT BOTTOM)
                      (SETQ LEFT (RAND 0 SCREENWIDTH))
                      (SETQ BOTTOM (RAND 0 SCREENHEIGHT))
                      (RETURN (CREATEREGION LEFT BOTTOM (RAND 10 (IDIFFERENCE SCREENWIDTH LEFT))
                                            (RAND 10 (IDIFFERENCE SCREENHEIGHT BOTTOM])
        (LIST-OF-ITEMS
            (TEST.GENERATE-LIST-OF-ITEMS (if (LISTP OBJECT-SPECIFICATIONS)
                                             then (CDR OBJECT-SPECIFICATIONS)
                                           else NIL)))
        (SHORT-SIMPLE-LIST
            (for I from 1 to (TEST.GENERATE-RANDOM (QUOTE VERY-SMALL-NON-NEGATIVE-INTEGER))
               collect (PACK* (QUOTE A)
                              I)))
        (SHORT-SIMPLE-NON-NULL-LIST
            (for I from 1 to (TEST.GENERATE-RANDOM (QUOTE VERY-SMALL-POSITIVE-INTEGER)) collect (PACK*
                                                                                                 (QUOTE A)
                                                                                                 I)))
        (SHORT-LIST [PROG (DEPTH)
                          (SETQ DEPTH (if (AND OBJECT-SPECIFICATIONS (LISTP OBJECT-SPECIFICATIONS))
                                          then (CADR OBJECT-SPECIFICATIONS)
                                        else TEST.MAX-DEPTH))
                          (if (EQ DEPTH 1)
                              then (RETURN (TEST.GENERATE-RANDOM (QUOTE SHORT-SIMPLE-LIST)))
                            else (RETURN (for I from 1 to (TEST.GENERATE-RANDOM (QUOTE
                                                                                VERY-SMALL-NON-NEGATIVE-INTEGER
                                                                                ))
                                         collect (TEST.GENERATE-RANDOM (LIST (QUOTE SHORT-LIST)
                                                                             (RAND 1 (SUB1 DEPTH])
        (LIST-OF-CHARACTERS
            [PROG (NUM-OF-CHARACTERS)
                  [SETQ NUM-OF-CHARACTERS (if (AND OBJECT-SPECIFICATIONS (LISTP
                                                                         OBJECT-SPECIFICATIONS
                                                                         ))
                                              then (CADR OBJECT-SPECIFICATIONS)
                                            else (TEST.GENERATE-RANDOM (QUOTE
                                                                       LARGE-POSITIVE-INTEGER
                                                                       ]
                  (RETURN (for I from 1 to NUM-OF-CHARACTERS collect (TEST.GENERATE-RANDOM (QUOTE
                                                                                           CHARACTER
                                                                                           ])
        (CHARACTER
            (CHARACTER (RAND 0 TEST.MAX-CHARACTER-CODE)))
        (PRINTOUT T OBJECT-TYPE " CAN NOT BE GENERATED."])


(TEST.GENERATE-LIST-OF-ITEMS
  [LAMBDA (SPEC-LIST)                                    (* sm "17-Jun-85 00:27")
    (PROG (ITEM-TYPE MIN-ITEMS MAX-ITEMS)
```

```
(SETQ ITEM-TYPE (if SPEC-LIST
                    then (if (LISTP (CAR SPEC-LIST))
                             then (CAR SPEC-LIST)
                           else (LIST (CAR SPEC-LIST)))
                  else TEST.TYPES-OF-RANDOM-OBJECTS))
(SETQ MIN-ITEMS (if (CDR SPEC-LIST)
                    then (CADR SPEC-LIST)
                  else 0))
(SETQ MAX-ITEMS (if (CDDR SPEC-LIST)
                    then (CADDR SPEC-LIST)
                  else (RAND 0 20)))
(RETURN (for I from 1 to (RAND MIN-ITEMS MAX-ITEMS) collect (TEST.GENERATE-RANDOM (
                                                                TEST.RANDOM-SELECTION
                                                                  ITEM-TYPE])
```

```
(TEST.RANDOM-SELECTION
  [LAMBDA (L)                                          (* sm "15-Jun-85 17:30")
    (CAR (NTH L (RAND 1 (LENGTH L])

)

(RPAQ? TEST.TYPES-OF-RANDOM-OBJECTS (QUOTE (INTEGER SHORT-SIMPLE-LIST SHORT-LIST CHARACTER)))

(RPAQ? TEST.MIN-INTEGER MIN.FIXP)

(RPAQ? TEST.MAX-INTEGER MAX.FIXP)

(RPAQ? TEST.VERY-SMALL 4)

(RPAQ? TEST.SMALLP MAX.SMALLP)

(RPAQ? TEST.SMALL 1000)

(RPAQ? TEST.LARGE (IDIFFERENCE MAX.FIXP 1000))

(RPAQ? TEST.MAX-DEPTH 4)

(RPAQ? TEST.MAX-CHARACTER-CODE 255)

(PUTPROPS RANDOM-GENERATOR COPYRIGHT ("Xerox Corporation" 1985))
```

---

## FUNCTION INDEX

TEST.GENERATE-LIST-OF-ITEMS .....2
TEST.GENERATE-RANDOM .....1
TEST.RANDOM-SELECTION .....3

---

## VARIABLE INDEX

TEST.LARGE .....3
TEST.MAX-CHARACTER-CODE .....3
TEST.MAX-DEPTH .....3
TEST.MAX-INTEGER .....3
TEST.MIN-INTEGER .....3
TEST.SMALL .....3
TEST.SMALLP .....3
TEST.TYPES-OF-RANDOM-OBJECTS .....3
TEST.VERY-SMALL .....3

---