

File created: 17-Jun-88 15:44:57 {QV}<PEDERSEN>LISP>TESTER.;5

changes to: (IL:VARS IL:TESTERCOMS)  
(IL:FUNCTIONS TEST-EQUAL DEFINE-VERIFIED-TEST MAKE-TEST-DEFUN TYPE-NUMBER GET-FORMS  
VERIFIED-TEST-TO-DO-TEST COMMENT-P)  
(IL:COMMANDS "COPY-TEST" "E-TEST")  
(FILE-ENVIRONMENTS "TESTER")

previous date: 14-Jun-88 14:56:12 {QV}<PEDERSEN>LISP>TESTER.;4

Read Table: XCL

Package: XCL-USER

Format: XCCS

; Copyright (c) 1988 by Xerox Corporation. All rights reserved.

```
(IL:RPAQQ IL:TESTERCOMS
  ((IL:DECLARE\ IL:DONTCOPY IL:DONTEVAL@LOAD IL:DOEVAL@COMPILE (IL:FILES IL:CMLARRAY-SUPPORT))
   (IL:FUNCTIONS TEST-EQUAL)
   (IL:COMS
    ;; tester definition
    (IL:DEFINE-TYPES VERIFIED-TESTS)
    (IL:FUNCTIONS DEFINE-VERIFIED-TEST MAKE-TEST-DEFUN))
    (IL:FUNCTIONS VERIFIED-TEST-TO-DO-TEST GET-FORMS COMMENT-P)
    (IL:FUNCTIONS TYPE-NUMBER MAKE-TEST-DEFUN)
    (IL:COMMANDS "COPY-TEST" "E-TEST")
    (FILE-ENVIRONMENTS "TESTER")))

(IL:DECLARE\ IL:DONTCOPY IL:DONTEVAL@LOAD IL:DOEVAL@COMPILE
  (IL:FILESLOAD IL:CMLARRAY-SUPPORT)
)

(DEFUN TEST-EQUAL (X Y)
  (EQUAL X Y))

;; tester definition

(DEF-DEFINE-TYPE VERIFIED-TESTS "verified regression test")

(DEFDEFINER DEFINE-VERIFIED-TEST VERIFIED-TESTS (NAME MESSAGE &BODY BODY)
  ;; Assumes the body is a form that returns a value or a list of values (comparable by equal) that may be computed at definition/compile time. NAME
  ;; is a symbol and MESSAGE is a string to printed at success/failure
  (LET ((VALUES (COMPILE-FORM `(PROGN ,@BODY)))
        `(EVAL-WHEN (LOAD)
          (FORMAT *ERROR-OUTPUT* "~&Test: ~a, " ,MESSAGE)
          (IF (TEST-EQUAL ',VALUES (PROGN ,@BODY))
              (FORMAT *ERROR-OUTPUT* "succeeded.~%")
              (FORMAT *ERROR-OUTPUT* "failed. *****~%")))))

(DEFMACRO MAKE-TEST-DEFUN (TEST-NAME)
  `(DEFUN ,TEST-NAME ()
    ,@(NTHCDR 3 (IL:GETDEF TEST-NAME 'VERIFIED-TESTS))))

(DEFUN VERIFIED-TEST-TO-DO-TEST (FILENAME PATHNAME &OPTIONAL (LINELENGTH 60))
  (LET* ((ROOT-NAME (INTERN (STRING FILENAME)
                            (FIND-PACKAGE "INTERLISP")))
         (MAKEFILE-ENVIRONMENT (GET ROOT-NAME 'IL:MAKEFILE-ENVIRONMENT)))
    (LET ((*PACKAGE* (FIND-PACKAGE (OR (SECOND (MEMBER :PACKAGE MAKEFILE-ENVIRONMENT :TEST #'EQ))
                                       "INTERLISP"))))
      (*READTABLE* (IL:FIND-READTABLE (OR (SECOND (MEMBER :READTABLE MAKEFILE-ENVIRONMENT :TEST
                                                           #'EQ))
                                          "INTERLISP"))))
      (*PRINT-BASE* (OR (SECOND (MEMBER :BASE MAKEFILE-ENVIRONMENT :TEST #'EQ))
                        10))
      (*PRINT-CASE* :DOWNCASE)
      (*PRINT-ARRAY* T)
      (*PRINT-LEVEL* NIL)
      (*PRINT-LENGTH* NIL)
      (*PRINT-STRUCTURE* T)
      ;; Interlisp gorp that controls pretty printing
      (IL:*PRINT-SEMICOLON-COMMENTS* T)
      (IL:FONTCHANGEFLG NIL)
      (IL:\#RPARS NIL)
      (IL:**COMMENT**FLG NIL))
    (DECLARE (GLOBAL IL:FILELINELENGTH IL:PRETTYFLG))
    (DECLARE (SPECIAL IL:FONTCHANGEFLG IL:\#RPARS IL:**COMMENT**FLG IL:*PRINT-SEMICOLON-COMMENTS*))
    (WITH-OPEN-FILE (STREAM (MAKE-PATHNAME :TYPE "TEST" :VERSION :NEWEST :DEFAULTS PATHNAME)
```

```

                :DIRECTION :OUTPUT)
(IL:RESETVARS
;; Interlisp gorp that controls pretty printing
                ((IL:FILELINELENGTH LINELENGTH)
                 (IL:PRETTYFLG T))

;; Identifier
                (FORMAT STREAM "~&;;; File converted on ~A from source ~A" (IL:DATE)
                  ROOT-NAME)
                (LET ((DATES (GET ROOT-NAME 'IL:FILEDATES)))
                  (WHEN DATES
                    (FORMAT STREAM "~&;;; Original source ~A created ~A" (CDAR DATES)
                      (CAAR DATES))))
                (TERPRI STREAM)
                (TERPRI STREAM)

;; Copyright notice
                (LET ((OWNER (GET ROOT-NAME 'IL:COPYRIGHT)))
                  (WHEN (AND OWNER (CONSP OWNER))
                    (FORMAT STREAM ";;; Copyright (c) "
                      (DO ((TAIL (CDR OWNER)
                                (CDR TAIL)))
                        ((NULL TAIL)
                         (FORMAT STREAM "~4d" (CAR TAIL))
                         (IF (CDR TAIL)
                             (PRINC ", " STREAM)))
                        (FORMAT STREAM " by ~a~%" (CAR OWNER))))
                    (TERPRI STREAM)
                    (DOLIST (COM (SYMBOL-VALUE (IL:FILECOMS ROOT-NAME)))
                      (DOLIST (FORM (GET-FORMS COM))
                        (PPRINT FORM STREAM)
                        (TERPRI STREAM)
                        (IL:BLOCK))))
                (NAMESTRING STREAM))))

```

```

(DEFUN GET-FORMS (COMMAND)
  (LET
    ((UNSUPPORTED-TYPES ' (IL:FNS IL:SPECVARS IL:GLOBALVARS IL:LOCALVARS IL:INITVARS IL:ALISTS IL:DEFS
                          IL:INITRECORDS IL:LISPMACROS IL:MACROS IL:PROPS IL:RECORDS IL:SYSRECORDS
                          IL:USERMACROS IL:VARS IL:CONSTANTS EXPORT IL:RESOURCES IL:INITRESOURCES
                          IL:GLOBALRESOURCES IL:I.S.OPRS IL:HORRIBLEVARS IL:UGLYVARS IL:BITMAPS IL:Cursors
                          IL:ADVICE IL:ADVISE IL:COURIERPROGRAMS IL:TEMPLATES IL:PROP IL:FILES
                          IL:DECLARE:))
     (FILEPKGTYPE (CAR COMMAND)))
    (IF (MEMBER FILEPKGTYPE UNSUPPORTED-TYPES :TEST #'EQ)
        (PROGN (WARN "Filepkg type ~s not supported: ~s" FILEPKGTYPE COMMAND)
              NIL)
        (CASE FILEPKGTYPE
          (IL:P (CDR COMMAND))
          (IL:COMS
            ;; Recurse
            (MAPCAN #' (LAMBDA (X)
                       (GET-FORMS X))
                    (CDR COMMAND)))
            ((EVAL-WHEN IL:EVAL-WHEN) `( (EVAL-WHEN , (MAPCAR #' (LAMBDA (SYM)
                                                                    (INTERN (STRING SYM)
                                                                    (FIND-PACKAGE "LISP"))))
                                         (SECOND COMMAND))
                                         ,@(GET-FORMS (THIRD COMMAND))))))
          ((IL:*)
            ;; Comment
            (LIST COMMAND))
          (T ;; Should the filepkgtype of a definer
            (LET
              ((IGNORED-DEFINERS ' (FILE-ENVIRONMENTS IL:DEFINE-TYPES OPTIMIZERS IL:SEdit-FORMATS
                                    ADVISED-FUNCTIONS IL:COMMANDS IL:SPECIAL-FORMS PROFILES
                                    XCL::WALKER-TEMPLATES))
               (DEFINER-TYPE (IL:GETFILEPKGTYPE FILEPKGTYPE 'IL:COMMANDS T)))
              (IF (MEMBER DEFINER-TYPE IGNORED-DEFINERS :TEST #'EQ)
                  (UNLESS (EQ DEFINER-TYPE 'FILE-ENVIRONMENTS)
                    (PROGN (WARN "Ignoring definer coms: ~s" COMMAND)
                          NIL))
                  (LET*
                    ((GET-DEF-METHOD (AND DEFINER-TYPE (GET DEFINER-TYPE :DEFINED-BY)
                                           (GET DEFINER-TYPE 'IL:GETDEF)))
                     (DEFS (AND GET-DEF-METHOD (MAPCAR #' (LAMBDA (NAME)
                                                             (IF (COMMENT-P NAME)
                                                                 NAME
                                                                 (FUNCCALL GET-DEF-METHOD NAME DEFINER-TYPE))
                                                             )
                           (CDR COMMAND))))
                     (CASE DEFINER-TYPE

```

```

(VERIFIED-TESTS
 (SETQ
  DEFS
  (MAPCAR
   #' (LAMBDA
        (DEF)
        (DESTRUCTURING-BIND
         (TAG NAME MESSAGE &BODY BODY)
         DEF
         (LET ((VALUES (COMPILE-FORM (REMOVE-COMMENTS `(PROGN ,@BODY))))))
              `(DO-TEST ,MESSAGE (EQUAL ',VALUES
                                          ,@(IF (EQ 1 (LENGTH BODY))
                                                BODY
                                                `((PROGN ,@BODY))))))))
         DEFS))))
 (OR DEFS (PROGN (WARN "Can't parse: ~s" COMMAND)
                 NIL))))))

```

```

(DEFUN COMMENT-P (FORM)
 (AND (CONSP FORM)
      (EQ (CAR FORM)
          'IL:*)
      (CONSP (CDR FORM))
      (MEMBER (CADR FORM)
              '(IL:\; IL:|;;| IL:|;;;|)
              :TEST
              #'EQ)
      T))

```

```

(DEFUN TYPE-NUMBER (TYPE)
 (IL:%CML-TYPE-TO-TYPENUMBER-EXPANDER TYPE))

```

```

(DEFMACRO MAKE-TEST-DEFUN (TEST-NAME)
 `(DEFUN ,TEST-NAME ()
  ,@(NTHCDR 3 (IL:GETDEF TEST-NAME 'VERIFIED-TESTS))))

```

```

(DEFCOMMAND "COPY-TEST" (FROM TO) (IL:COPYDEF FROM TO 'VERIFIED-TESTS))

```

```

(DEFCOMMAND "E-TEST" (NAME) (ED NAME '(:DONTWAIT VERIFIED-TESTS)))

```

```

(DEFINE-FILE-ENVIRONMENT "TESTER" :PACKAGE "XCL-USER"
 :READTABLE "XCL"
 :COMPILER :COMPILE-FILE)

```

```

(IL:PUTPROPS IL:TESTER IL:COPYRIGHT ("Xerox Corporation" 1988))

```

---

**FUNCTION INDEX**

COMMENT-P .....3      TEST-EQUAL .....1      VERIFIED-TEST-TO-DO-TEST .....1  
GET-FORMS .....2      TYPE-NUMBER .....3

---

**COMMAND INDEX**

"COPY-TEST" .....3      "E-TEST" .....3

---

**FILE-ENVIRONMENT INDEX**

"TESTER" .....3

---

**DEFINE-TYPE INDEX**

VERIFIED-TESTS .....1

---

**MACRO INDEX**

MAKE-TEST-DEFUN .....1,3

---

**DEFINER INDEX**

DEFINE-VERIFIED-TEST .....1

---