

File created: 10-Mar-88 10:02:26 {ERIS}<SNOW>SUN>TEST>DO-TEST.;3

changes to: (FUNCTIONS XCL-USER::DO-ALL-TESTS)
(VARS DO-TESTCOMS)
(VARIABLES XCL-USER::*TEST-MODE* XCL-USER::*TEST-BATCH-RESULTS* XCL-USER::*TEST-FILE-PATTERN*)

previous date: 2-Mar-88 15:47:17 {ERIS}<SNOW>SUN>TEST>DO-TEST.;1

Read Table: XCL

Package: INTERLISP

Format: XCCS

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

(RPAQQ DO-TESTCOMS

```
((VARIABLES XCL-USER::*ANY-ERRORS* XCL-USER::*TEST-CLEANUP-FORMS* XCL-USER::*TEST-COMPILE*
  XCL-USER::*TEST-MODE* XCL-USER::*TEST-BATCH-RESULTS* XCL-USER::*TEST-FILE-PATTERN*
  XCL-USER::*TEST-FILE-NAME*)
(P (DEFPACKAGE "XCL-TEST" (:USE "LISP" "XCL")
  (:IMPORT XCL-USER::DO-TEST-FILE XCL-USER::DO-ALL-TESTS XCL-USER::DO-TEST
    XCL-USER::DO-TEST-GROUP XCL-USER::CL-READFILE XCL-USER::EXPECT-ERRORS
    XCL-USER::TEST-DEFUN XCL-USER::TEST-DEFMACRO XCL-USER::TEST-SETQ XCL-USER::*TEST-MODE*
    XCL-USER::*TEST-COMPILE* XCL-USER::*TEST-BATCH-RESULTS* XCL-USER::*TEST-FILE-PATTERN*
    XCL-USER::*TEST-FILE-NAME*))
(FUNCTIONS XCL-USER::DO-TEST XCL-USER::DO-TEST-GROUP XCL-USER::TEST-DEFMACRO XCL-USER::TEST-DEFUN
  XCL-USER::TEST-SETQ XCL-USER::WITHOUT-BATCH-MODE-ERRORS XCL-USER::EXPECT-ERRORS
  XCL-USER::DO-ALL-TESTS XCL-USER::CURRENT-FILE-NAME XCL-USER::CL-READFILE XCL-USER::DO-TEST-FILE
  XCL-USER::DO-TEST-LIST)
(PROP (MAKEFILE-ENVIRONMENT FILETYPE)
  DO-TEST)))
```

(CL:DEFVAR XCL-USER::*ANY-ERRORS* NIL)

(CL:DEFVAR XCL-USER::*TEST-CLEANUP-FORMS* NIL)

(CL:DEFVAR XCL-USER::*TEST-COMPILE* NIL)

(CL:DEFVAR XCL-USER::*TEST-MODE* :INTERACTIVE)

(CL:DEFVAR XCL-USER::*TEST-BATCH-RESULTS* "{dsk}</python0/lisp/test/auto/language>test-results")

(CL:DEFVAR XCL-USER::*TEST-FILE-PATTERN*

```
' ("{dsk}</python0/lisp/test/auto/language/4>*.TEST" "{dsk}</python0/lisp/test/auto/language/5>*.TEST"
  "{dsk}</python0/lisp/test/auto/language/6>*.TEST" "{dsk}</python0/lisp/test/auto/language/7>*.TEST"
  "{dsk}</python0/lisp/test/auto/language/8>*.TEST" "{dsk}</python0/lisp/test/auto/language/9>*.TEST"
  "{dsk}</python0/lisp/test/auto/language/10>*.TEST"
  "{dsk}</python0/lisp/test/auto/language/11>*.TEST"
  "{dsk}</python0/lisp/test/auto/language/12>*.TEST"
  "{dsk}</python0/lisp/test/auto/language/13>*.TEST"
  "{dsk}</python0/lisp/test/auto/language/14>*.TEST"
  "{dsk}</python0/lisp/test/auto/language/15>*.TEST"
  "{dsk}</python0/lisp/test/auto/language/16>*.TEST"
  "{dsk}</python0/lisp/test/auto/language/17>*.TEST"
  "{dsk}</python0/lisp/test/auto/language/18>*.TEST"
  "{dsk}</python0/lisp/test/auto/language/19>*.TEST"
  "{dsk}</python0/lisp/test/auto/language/20>*.TEST"
  "{dsk}</python0/lisp/test/auto/language/21>*.TEST"
  "{dsk}</python0/lisp/test/auto/language/22>*.TEST"
  "{dsk}</python0/lisp/test/auto/language/23>*.TEST"
  "{dsk}</python0/lisp/test/auto/language/24>*.TEST"
  "{dsk}</python0/lisp/test/auto/language/25>*.TEST"
  "{dsk}</python0/lisp/test/auto/language/ar>*.TEST"
  "{dsk}</python0/lisp/test/auto/language/other>*.TEST"))
```

(CL:DEFVAR XCL-USER::*TEST-FILE-NAME* "unknown")

(DEFPACKAGE "XCL-TEST" (:USE "LISP" "XCL")

```
(:IMPORT XCL-USER::DO-TEST-FILE XCL-USER::DO-ALL-TESTS XCL-USER::DO-TEST XCL-USER::DO-TEST-GROUP
  XCL-USER::CL-READFILE XCL-USER::EXPECT-ERRORS XCL-USER::TEST-DEFUN XCL-USER::TEST-DEFMACRO
  XCL-USER::TEST-SETQ XCL-USER::*TEST-MODE* XCL-USER::*TEST-COMPILE* XCL-USER::*TEST-BATCH-RESULTS*
  XCL-USER::*TEST-FILE-PATTERN* XCL-USER::*TEST-FILE-NAME*))
```

(DEFMACRO XCL-USER::DO-TEST (NAME-AND-OPTIONS &BODY BODY)

```
(LET ((NAME NIL)
  (OPTIONS NIL))
(COND
  ((CL:CONSP NAME-AND-OPTIONS)
  (CL:SETQ NAME (CAR NAME-AND-OPTIONS)))
```

```

      (CL:SETQ OPTIONS (CDR NAME-AND-OPTIONS)))
      (T (CL:SETQ NAME NAME-AND-OPTIONS)))
    (CL:IF (OR (EQ XCL-USER::*TEST-MODE* :INTERACTIVE)
              (EQ XCL-USER::*TEST-MODE* :BATCH-VERBOSE))
          (CL:FORMAT *ERROR-OUTPUT* "Testing... ~S~%" NAME))
    `(NOT (CL:WHEN (NULL (XCL-USER::WITHOUT-BATCH-MODE-ERRORS ,@BODY))
                  (CL:FORMAT *ERROR-OUTPUT* "Test \"~A\" failed in file \"~A\"~%" ',NAME (
                                                                XCL-USER::CURRENT-FILE-NAME
                                                                ))
                  (SETQ XCL-USER::*ANY-ERRORS* T))))))

```

```
(DEFMACRO XCL-USER::DO-TEST-GROUP (NAME-AND-OPTIONS &BODY BODY)
```

```

  (LET ((NAME NIL)
        (OPTIONS NIL))
    (COND
      ((CL:CONSP NAME-AND-OPTIONS)
       (CL:SETQ NAME (CAR NAME-AND-OPTIONS))
       (CL:SETQ OPTIONS (CDR NAME-AND-OPTIONS)))
      (T (CL:SETQ NAME NAME-AND-OPTIONS)))
    ;; Hack: find :BEFORE and :AFTER clauses in the body and move them out
    (CL:LOOP (CL:IF (AND (CL:SYMBOLP (CAR BODY))
                        (OR (EQ (CAR BODY)
                               :BEFORE)
                          (EQ (CAR BODY)
                               :AFTER))))
              (PROGN (SETQ OPTIONS (APPEND OPTIONS (LIST (CAR BODY)
                                                         (CADR BODY))))
                    (SETQ BODY (CDDR BODY)))
              (RETURN NIL)))
    `(LET ((XCL-USER::*TEST-CLEANUP-FORMS* NIL)
          (CL:BLOCK ,NAME
            , (CL:IF (OR (EQ XCL-USER::*TEST-MODE* :INTERACTIVE)
                        (EQ XCL-USER::*TEST-MODE* :BATCH-VERBOSE))
                    (CL:FORMAT *ERROR-OUTPUT* "Testing... ~S~%" NAME))
            , (LET ((BEFORE (IGNORE-ERRORS (CL:GETF OPTIONS :BEFORE))))
                (CL:IF BEFORE
                  `(CL:WHEN (NULL (XCL-USER::WITHOUT-BATCH-MODE-ERRORS ,BEFORE T))
                        (CL:FORMAT *ERROR-OUTPUT* ":BEFORE forms for test \"~A\" in file ~S failed."
                                  ',NAME
                                  (XCL-USER::CURRENT-FILE-NAME))
                        (SETQ XCL-USER::*ANY-ERRORS* T)
                        (CL:RETURN-FROM ,NAME))))
                  , @ (|for| B |in| BODY |join| (|if| (AND (CL:CONSP B)
                                                         (EQ (CAR B)
                                                             'XCL-USER::DO-TEST))
                                                         |then| (LIST B)
                                                         |else| (CL:FORMAT *ERROR-OUTPUT* "Non DO-TEST form in ~S in ~S~%~S~%"
                                                                    NAME (XCL-USER::CURRENT-FILE-NAME)
                                                                    B)))
                  , (LET ((AFTER (IGNORE-ERRORS (CL:GETF OPTIONS :AFTER))))
                      (CL:IF AFTER
                        `(CL:WHEN (NULL (XCL-USER::WITHOUT-BATCH-MODE-ERRORS ,AFTER T))
                              (CL:FORMAT *ERROR-OUTPUT* ":AFTER forms for test \"~A\" in file ~S failed."
                                        ',NAME
                                        (XCL-USER::CURRENT-FILE-NAME))
                              (CL:SETQ XCL-USER::*ANY-ERRORS* T))))
                        (CL:EVAL (CONS 'PROGN XCL-USER::*TEST-CLEANUP-FORMS*))
                        NIL)))

```

```
(DEFMACRO XCL-USER::TEST-DEFMACRO (NAME &REST STUFF)
```

```

  `(PROGN (CL:IF (CL:FBOUNDP ',NAME)
                 (CL:IF (CL:MACRO-FUNCTION ',NAME)
                       (CL:PUSH (LIST 'CL:SETF (LIST 'CL:SYMBOL-FUNCTION (LIST 'CL:MACRO-FUNCTION
                                                                                ',NAME)))
                                (LIST 'QUOTE (CL:SYMBOL-FUNCTION (CL:MACRO-FUNCTION ',NAME))))
                       XCL-USER::*TEST-CLEANUP-FORMS*)
                 (CL:ERROR "Please don't redefine ~A in a test form" ',NAME))
    (CL:PUSH (LIST 'REMPROP ',NAME 'MACRO-FN)
            XCL-USER::*TEST-CLEANUP-FORMS*))
  (DEFMACRO ,NAME ,@STUFF))

```

```
(DEFMACRO XCL-USER::TEST-DEFUN (NAME &REST STUFF)
```

```

  `(PROGN (CL:IF (CL:FBOUNDP ',NAME)
                 (CL:IF (OR (CL:MACRO-FUNCTION ',NAME)
                           (CL:SPECIAL-FORM-P ',NAME))
                       (CL:ERROR "Please don't redefine ~A in a test form" ',NAME)
                       (CL:PUSH (LIST 'CL:SETF (LIST 'CL:SYMBOL-FUNCTION ' ',NAME)
                                                    (LIST 'QUOTE (CL:SYMBOL-FUNCTION ',NAME)))
                                XCL-USER::*TEST-CLEANUP-FORMS*))
                 (CL:PUSH (LIST 'CL:FMAKUNBOUND ' ',NAME)
                          XCL-USER::*TEST-CLEANUP-FORMS*))
    (CL:DEFUN ,NAME ,@STUFF))

```

```

(DEFMACRO XCL-USER::TEST-SETQ (&REST XCL-USER::STUFF)
  (LET (XCL-USER::UNBINDLIST)
    (CL:DO ((XCL-USER::X XCL-USER::STUFF (CDDR XCL-USER::X)))
      ((NULL XCL-USER::X))
      (CL:PUSH `(CL:IF (BOUNDP ',(CAR XCL-USER::X))
        (CL:PUSH (LIST 'CL:SETQ ',(CAR XCL-USER::X)
          (LIST 'QUOTE (CL:SYMBOL-VALUE ',(CAR XCL-USER::X))))
        XCL-USER::*TEST-CLEANUP-FORMS*)
      (CL:PUSH (LIST 'CL:MAKUNBOUND ',(CAR XCL-USER::X))
        XCL-USER::*TEST-CLEANUP-FORMS*))
      XCL-USER::UNBINDLIST)
    `(PROGN ,@XCL-USER::UNBINDLIST (CL:SETQ ,@XCL-USER::STUFF))))

(DEFMACRO XCL-USER::WITHOUT-BATCH-MODE-ERRORS (&BODY BODY)
  (COND
    ((EQ XCL-USER::*TEST-MODE* :INTERACTIVE)
      `(PROGN ,@BODY))
    (T `(IGNORE-ERRORS ,@BODY))))

(DEFMACRO XCL-USER::EXPECT-ERRORS (ERROR-TYPES &REST FORMS)
  `(CONDITION-CASE (PROGN ,@FORMS NIL)
    (,ERROR-TYPES (CONDITION)
      (CL:VALUES T CONDITION))))

(CL:DEFUN XCL-USER::DO-ALL-TESTS (&KEY (XCL-USER::RESULTS XCL-USER::*TEST-BATCH-RESULTS*)
  (XCL-USER::PATTERNS (CL:IF (CL:CONSP XCL-USER::*TEST-FILE-PATTERN*)
    XCL-USER::*TEST-FILE-PATTERN*
    (LIST XCL-USER::*TEST-FILE-PATTERN*)))
  (XCL-USER::SYSOUT-TYPE NIL)
  (XCL-USER::RESUME NIL))
  (LET ((NO-PROBLEMS T)
    (*DEFAULT-PATHNAME-DEFAULTS* (PATHNAME "{dsk}</python0/lisp/test/auto/language>"))
    (*ERROR-OUTPUT* (CL:IF (EQ XCL-USER::RESULTS T)
      *ERROR-OUTPUT*
      (OPEN XCL-USER::RESULTS :DIRECTION :OUTPUT :IF-EXISTS (CL:IF XCL-USER::RESUME
        :APPEND
        :NEW-VERSION))))))
    (CL:UNWIND-PROTECT
      (PROGN (CL:IF (NOT XCL-USER::RESUME)
        (PROGN (CL:FORMAT *ERROR-OUTPUT* ";;; Test results for sysout of ~A~%" MAKESYSDATE)
          (CL:IF XCL-USER::SYSOUT-TYPE (CL:FORMAT *ERROR-OUTPUT* ";;; Sysout type is ~A~%"
            XCL-USER::SYSOUT-TYPE))
          (CL:IF XCL-USER::*TEST-COMPILE* (CL:FORMAT *ERROR-OUTPUT* ";;; Tests are being
            compiled~%"))
          (CL:FORMAT *ERROR-OUTPUT* ";;; Tests run on ~A~%" (DATE))
          (CL:FORMAT *ERROR-OUTPUT* ";;; Running tests from ~A~2%" XCL-USER::PATTERNS)
          (CL:SETQ XCL-USER::*ALL-FILES-REMAINING* (FOR XCL-USER::DP IN XCL-USER::PATTERNS
            JOIN (DIRECTORY XCL-USER::DP))))
          (CL:FORMAT *ERROR-OUTPUT* ";;; Resuming after dying on file ~S~%" (CL:POP
            XCL-USER::*ALL-FILES-REMAINING*
            )))
        (|while| XCL-USER::*ALL-FILES-REMAINING* |do| (CL:FORMAT *STANDARD-OUTPUT* "Testing ~s..."
          (CAR XCL-USER::*ALL-FILES-REMAINING*))
          (CL:SETQ NO-PROBLEMS (AND (XCL-USER::DO-TEST-FILE
            (CAR
              XCL-USER::*ALL-FILES-REMAINING*
            )
            NO-PROBLEMS))
          (CL:FORMAT *STANDARD-OUTPUT* "done~%")
          (CL:POP XCL-USER::*ALL-FILES-REMAINING*)))
          (CL:FORMAT *ERROR-OUTPUT* "(END-OF-TESTS)"))
      (CL:UNLESS (EQ XCL-USER::RESULTS T)
        (CL:CLOSE *ERROR-OUTPUT*))
      NO-PROBLEMS))

(CL:DEFUN XCL-USER::CURRENT-FILE-NAME ()
  XCL-USER::*TEST-FILE-NAME*)

(CL:DEFUN XCL-USER::CL-READFILE (TEST-FILE &OPTIONAL (*READTABLE* CMLRDTBL)
  (ENDTOKEN "STOP"))
  (LET ((XCL-USER::TRUE-NAME (CL:PROBE-FILE TEST-FILE)))
    (CL:IF XCL-USER::TRUE-NAME
      (LET (TEM (*PACKAGE* (CL:FIND-PACKAGE 'XCL-USER::XCL-TEST))
        (*FEATURES* (CONS :NO-STACK-OVERFLOW *FEATURES*)))
        (CL:SETQ XCL-USER::*TEST-FILE-NAME* (CL:NAMESTRING XCL-USER::TRUE-NAME))
        (CL:WITH-OPEN-FILE (TEST-FILE TEST-FILE :DIRECTION :INPUT)
          (UNTIL (OR (NULL (IGNORE-ERRORS (CL:SETQ TEM (CL:READ TEST-FILE))))
            (AND (CL:SYMBOLP TEM)
              (CL:STRING= TEM ENDTOKEN))))
          COLLECT TEM)))
      (PROGN (CL:FORMAT *ERROR-OUTPUT* "~%Couldn't find file ~A~%" TEST-FILE)

```

NIL)))

```

(CL:DEFUN XCL-USER::DO-TEST-FILE (FILENAME)
  (LET* ((*PACKAGE* (CL:FIND-PACKAGE 'XCL-USER::XCL-TEST))
         (XCL-USER::*TEST-FILE-NAME* NIL)
         (TEST-FORMS (XCL-USER::CL-READFILE FILENAME CMLRDTBL))
         (XCL-USER::*ANY-ERRORS* NIL))
    (XCL-USER::DO-TEST-LIST TEST-FORMS)
    (|if| XCL-USER::*ANY-ERRORS*
      |then| (CL:TERPRI *ERROR-OUTPUT*))
    (NOT XCL-USER::*ANY-ERRORS*)))

(CL:DEFUN XCL-USER::DO-TEST-LIST (XCL-USER::TEST-FORMS &OPTIONAL XCL-USER::OPTIONS XCL-USER::NAME)
  (LET
    ((DFNFLG NIL))
    (DECLARE (CL:SPECIAL DFNFLG))
    (|if| (NULL XCL-USER::TEST-FORMS)
      |then| (CL:FORMAT *ERROR-OUTPUT* "~%(Trouble reading ~A)~%" (XCL-USER::CURRENT-FILE-NAME))
      (CL:SETQ XCL-USER::*ANY-ERRORS* T)
      |else| (|for| XCL-USER::FORM |in| XCL-USER::TEST-FORMS
        |do| (BLOCK 0)
          (CL:IF (AND (CL:CONSP XCL-USER::FORM)
                     (OR (EQ (CAR XCL-USER::FORM)
                           'XCL-USER::DO-TEST)
                         (EQ (CAR XCL-USER::FORM)
                           'XCL-USER::DO-TEST-GROUP))))
            (CL:IF XCL-USER::*TEST-COMPILE*
              (CL:BLOCK XCL-USER::COMPILER-PUNT
                (LET ((XCL-USER::COMPILED-FORM (CL:IF (EQ XCL-USER::*TEST-MODE* :INTERACTIVE)
                                                         (CL:COMPILE NIL `(CL:LAMBDA NIL
                                                                    ,XCL-USER::FORM))
                                                         (IGNORE-ERRORS
                                                           (CL:COMPILE NIL `(CL:LAMBDA NIL
                                                                    ,XCL-USER::FORM))))))
                  (CL:IF (NULL (CL:COMPILED-FUNCTION-P XCL-USER::COMPILED-FORM))
                    (LET ((*PRINT-LEVEL* 3)
                          (*PRINT-LENGTH* 3))
                      (CL:FORMAT *ERROR-OUTPUT* "Compilation of this form in file ~S
failed:~% ~S~%" (XCL-USER::CURRENT-FILE-NAME)
                  XCL-USER::FORM)
                      (CL:RETURN-FROM XCL-USER::COMPILER-PUNT)))
                    (CL:IF (NULL (CL:IF (EQ XCL-USER::*TEST-MODE* :INTERACTIVE)
                                         (PROGN (CL:FUNCALL XCL-USER::COMPILED-FORM)
                                                    T)
                                         (IGNORE-ERRORS (PROGN (CL:FUNCALL
                                                                    XCL-USER::COMPILED-FORM)
                                                                    T))))
                      (LET ((*PRINT-LEVEL* 3)
                            (*PRINT-LENGTH* 3))
                        (CL:FORMAT *ERROR-OUTPUT* "Compiled code failed for this form in
file ~S :~%~S~%" (XCL-USER::CURRENT-FILE-NAME)
                              XCL-USER::FORM))))))
                    (CL:EVAL XCL-USER::FORM))
                    (CL:FORMAT *ERROR-OUTPUT* "Non DO-TEST form at top level in ~S~%~S~%" (
                                                                    XCL-USER::CURRENT-FILE-NAME
                                                                    )
                              XCL-USER::FORM))))))
          (PUTPROPS DO-TEST FILETYPE :COMPILE-FILE)
          (PUTPROPS DO-TEST COPYRIGHT ("Xerox Corporation" 1986 1987 1988))

```

FUNCTION INDEX

XCL-USER::CL-READFILE	3	XCL-USER::DO-ALL-TESTS	3	XCL-USER::DO-TEST-LIST	4
XCL-USER::CURRENT-FILE-NAME	3	XCL-USER::DO-TEST-FILE	4		

MACRO INDEX

XCL-USER::DO-TEST	1	XCL-USER::TEST-DEFUN	2
XCL-USER::DO-TEST-GROUP	2	XCL-USER::TEST-SETQ	3
XCL-USER::EXPECT-ERRORS	3	XCL-USER::WITHOUT-BATCH-MODE-ERRORS	3
XCL-USER::TEST-DEFMACRO	2		

VARIABLE INDEX

XCL-USER::*ANY-ERRORS*	1	XCL-USER::*TEST-COMPILE*	1	XCL-USER::*TEST-MODE*	1
XCL-USER::*TEST-BATCH-RESULTS* ...	1	XCL-USER::*TEST-FILE-NAME*	1	DO-TESTCOMS	1
XCL-USER::*TEST-CLEANUP-FORMS* ...	1	XCL-USER::*TEST-FILE-PATTERN*	1		

PROPERTY INDEX

DO-TEST	4
---------------	---
