

```
;; This is a collection of tests from the Debugger.NoteFile. It tests the break package and IL
error system. The individual test files for each of the functions have been appended together in
this big file to share common code and gain diagnostic information by testing the functions in a
particular order.
```

```
;;
;; The source for the text file listing is the NoteCards database at
{Erinyes}<Test>Lisp>Lyric>Plans>Debugger.NoteFile. Changes are made only to the NoteFile. The
listings are
;; Filed As: {eris}<lispcore>test>Debugger>Debugger.u and
{eris}<lispcore>test>Debugger>BreakWindow.u
```

```
(do-test "setup user interaction window"
  (unless (fboundp 'do-test-menu-setup)
    (load "{Eris}<test>tools>do-test-menu.dfasl" t))
  (setq dtmw (do-test-menu-setup "Debugger"))))
(do-test "bring up exec"
  (do-test-menu-message dtmw 'low
    (concatenate 'string
      "From the background menu, bring up a new exec.

      "
      "In the exec
      "
      "(SETQ *PACKAGE* (FIND-PACKAGE 'XCL-TEST))"))))
(do-test-group "STEPPER"
  (do-test "call step"
    (do-test-menu-message dtmw 'low
      (concatenate 'string
        "Shift select the following into the exec window.

        "
        "(step (defun foo nil (print \"hello\")))

        "
        "You should be prompted by repeating the form and a colon.)))
    (do-test "step space"
      (do-test-menu-message dtmw 'low
        (concatenate 'string
          "Type a space in the exec window after the colon. "
          "You should see a long progn form followed by a colon."
          "

          Type another space. "
          "You should see the rest of the first part of the form.)))
    (do-test "step Next"
      (do-test-menu-message dtmw 'low
        (concatenate 'string "Type N in the exec window after the colon. "
          "You should see a long lambda form.)))
    (do-test "step Debugger (AR 7709)"
      (do-test-menu-message dtmw 'low
        (concatenate 'string "Type d in the exec window after the colon. "
          "You should see a break window.)))
    (do-test "step break window OK"
      (do-test-menu-message dtmw 'low
        (concatenate 'string "Type ok in the break window and hit return. "
          "The break window should go a way and"
          " you should see a colon.)))
    (do-test "step Finish"
      (do-test-menu-message dtmw 'low
        (concatenate 'string "Type f in the exec widnow after the colon. "
          " You should see the word foo three times "
          "followed by a new Exec promp.)))
    (do-test "step uparrow"
      (do-test-menu-message dtmw 'low
        (concatenate 'string
          "Shift select the following into the exec window.

          "
          "(step (defun foo nil (print \"hello\")))

          "
          "You should be prompted by repeating the form and a colon."
          " Type an uparrow.

          "
          "You should see the word abort and then an exec prompt.))))))
(do-test-group ("EVAL UB OK")
  (do-test "menued commands"
    (do-test-menu-message dtmw 'low
```

```

(concatenate 'string
  "In the exec, enter

  IL:*DEBUGGER-MENU-ITEMS*

  "
  "You should see (\"EVAL\" \"EDIT\" \"REVERT\" \"↑\" \"PROCEED\" \"OK\" \"BT\" \"BT!\" \"?=\"))
(do-test "eval create break window"
  (do-test-menu-message dtmw 'low
    (concatenate 'string "Shift select the following into the exec
  "
    " (progn (unbreak foo)
    " " (defun foo nil (print \"hello\"))
    "
    " (break-function 'foo))(foo)

  "
  "You should then see a break window containing Breakpoint at foo")))
(do-test "pre eval value"
  (do-test-menu-message dtmw 'low
    (concatenate 'string
      "In the break window, type the VALUE command.

  "
      "You should see \"Not yet evaluated.\""))
(do-test "type eval"
  (do-test-menu-message dtmw 'low
    (concatenate 'string
      "In the break window, type the EVAL command.

  "
      "You should see \"hello\""))
(do-test "menu eval"
  (do-test-menu-message dtmw 'low
    (concatenate 'string
      "Select EVAL from the middle button menu in the break window.

  "
      "You should see \"hello\""))
(do-test "type ub"
  (do-test-menu-message dtmw 'low
    (concatenate 'string
      "In the break window, type the UB command.

  "
      "You should see (foo)"))
(do-test "post eval value"
  (do-test-menu-message dtmw 'low
    (concatenate 'string
      "In the break window, type the VALUE command.

  "
      "You should see \"hello\""))
(do-test "menu OK"
  (do-test-menu-message dtmw 'low
    (concatenate 'string "Press down the middle button of the mouse "
      "in the break window, select the OK command.
  "
      "The break window should disappear and "
      "you should see \"hello\" in the exec.))))
(do-test-group ("proceed comands")
  (do-test "create break window"
    (defun foo nil
      (restart-case (break)
        (use-value (x)
          (or x (il:promptforward "Use a different value"))
          (nil nil :report "Just return NIL"))))
    (do-test-menu-message dtmw 'low
      (concatenate 'string "In the exec, type (foo)
  "
        "You should then see a break window containing "
        "Break"))
  (do-test "use value PROCEED command from menu"
    (do-test-menu-message dtmw 'low
      (concatenate 'string "Press down the middle button of the mouse "
        "in the break window, " "select the PROCEED command.
  "
        "You should then see a menu pop up.
  "
        " Select \"Use a different value\" from the menu.

```

```

"
"You should see a request for a value.
"
"Type a T to the request and hit return."
"The break window should go away and "
"you should see a T in the exec window.)))
(do-test "use value typed PROCEED command"
  (do-test-menu-message dtmw 'low
    (concatenate 'string "In the exec, type (foo)
"
  "in the break window, type the PROCEED command
"
  "You should then see a list.
"
  " Type the number by the item from the list and hit RETURN
"
  "Use a different value.

"
  "You should see a request for a value.

"
  "Type a T to the request and hit return."
  "The break window should go away and "
  "you should see a T in the exec window.)))
(do-test "use value typed PR command"
  (do-test-menu-message dtmw 'low
    (concatenate 'string "In the exec, type (foo)
"
  "in the break window, type the PR command
"
  "You should then see a list.
"
  " Type the number by the item from the list and hit RETURN
"
  "Use a different value.

"
  "You should see a request for a value.

"
  "Type a T to the request and hit return."
  "The break window should go away and "
  "you should see a T in the exec window.))))
(do-test-group ("RETURN")
  (do-test "RETURN T command"
    (do-test-menu-message dtmw 'low
      (concatenate 'string "In the exec type (foo). "
        "In the resulting break window, type the RETURN command.
"
      "The break window should go away and "
      "you should see a NIL result in the exec.)))
  (do-test "RETURN command"
    (do-test-menu-message dtmw 'low
      (concatenate 'string "In the exec type (foo). "
        "In the resulting break window, type the RETURN T command.
"
      "The break window should go away and "
      "you should see a T result in the exec.))))
(do-test-group ("?= uparrow")
  (do-test "=?= create break window"
    (do-test-menu-message dtmw 'low
      (concatenate 'string "Type (foo) to the exec
"
      "You should then see a break window ")))
  (do-test "type ?="
    (do-test-menu-message dtmw 'low
      (concatenate 'string
        "In the break window, type the ?= command.
"
      "You should see the two parameters: (break) and a Lexical Environment.)))
  (do-test "menu ?="
    (do-test-menu-message dtmw 'low
      (concatenate 'string "With the middle button in the break window, "
        "select the ?= command.

```

```

"
    "You should see the same two parameters.")))
(do-test "BREAKDELIMITER correct default"
  (do-test-menu-message dtmw 'low
    (concatenate 'string "Enter

IL:BREAKDELIMITER

"
    "You should see some representation of a "
    "carriage return character such as a quoted new line.")))
(do-test "menu ↑"
  (do-test-menu-message dtmw 'low
    (concatenate 'string "With the middle button in the break window, "
      "select the ↑ command.

"
      "The break window should disappear "
      "and the caret should be back in the exec "
      "with no value returned.")))
(do-test-group "BREAK WINDOW"
  (do-test "create break window"
    (do-test-menu-message dtmw 'low
      (concatenate 'string
        "In an exec type control B to cause a break.

"
        "You should see a break window pop up.")))
  (do-test "type BT"
    (do-test-menu-message dtmw 'low
      (concatenate 'string "Type the BT command.

"
        "You should see a backtrace scroll by.

"
        "(If the window turns black, just hit a space.)")))
  (do-test "type DBT"
    (do-test-menu-message dtmw 'low
      (concatenate 'string "Type the DBT command.

"
        "You should see a backtrace window "
        "appended to the side of the break window.")))
  (do-test "menu BT"
    (do-test-menu-message dtmw 'low
      (concatenate 'string "From the middle button in the break window, "
        "choose the BT command.

"
        "You should see a backtrace window "
        "appended to the side of the break window.")))
  (do-test "type BT!"
    (do-test-menu-message dtmw 'low
      (concatenate 'string "Type the BT! command.

"
        "You should see a longer backtrace scroll by.")))
  (do-test "menu DBT!"
    (do-test-menu-message dtmw 'low
      (concatenate 'string "Type the DBT! command.

"
        "You should see the detailed backtrace in a "
        "window appended to the side of the break window.")))
  (do-test "menu BT!"
    (do-test-menu-message dtmw 'low
      (concatenate 'string "From the middle button in the break window, "
        "choose the BT! command.

"
        "You should see a more detailed backtrace window appear "
        "appended to the side of the break window.

"
        ")))
  (do-test "scroll BT!"
    (do-test-menu-message dtmw 'low
      (concatenate 'string
        " You should be able to scroll up and down in the backtrace window.")))

```

```

(do-test "BTV command"
  (do-test-menu-message dtmw 'low
    (concatenate 'string "Type the BTV command.

"
      "You should see an even more detailed "
      "backtrace scroll by.

"
      "Terminate scrolling by hitting CTRLc.

"
    )))
(do-test "BTV! command"
  (do-test-menu-message dtmw 'low
    (concatenate 'string "Type the BTV! command.

"
      "You should see an even more detailed backtrace window appear "
      "scroll by. Terminate scrolling by hitting CTRLc.

"
    )))
(do-test-group ("@ REVERT STOP")
  (do-test "@ "
    (do-test-menu-message dtmw 'low
      (concatenate 'string "Enter an at-sign command.

" "@

"
      "You should see @ = il:interrupt.")))
  (do-test "@ foo"
    (do-test-menu-message dtmw 'low
      (concatenate 'string "Enter

" " @ foo

"
      "[Note: space between the @ and foo.]

"
      "You should see FOO not found.")))
  (do-test "@ exec-read"
    (do-test-menu-message dtmw 'low
      (concatenate 'string "Enter

" " @ exec-read

"
      "You should see @ = il:exec-read.")))
  (do-test "@ exec-read / 1 (AR 8062)"
    (do-test-menu-message dtmw 'low
      (concatenate 'string "Enter

" " @ exec-read / 1

"
      "You should see @ = il:exec-read again.")))
  (do-test "@ = exec-read (AR 8062)"
    (do-test-menu-message dtmw 'low
      (concatenate 'string "Enter

" " @ = exec-read

"
      "You should see @ = il:exec-read again.")))
  (do-test "@ number (AR 8062)"
    (do-test-menu-message dtmw 'low
      (concatenate 'string "Enter

" " @ 3

"
      "You should see the name of the third stack item.")))
  (do-test "frame window"
    (do-test-menu-message dtmw 'low
      (concatenate 'string
        "In the backtrace window attached to the break window "
        "find IL:TTYIN and select it with the left mouse button.

"
      )))

```

```

        "You should see the frame window for ttyin pop up. ")))
(do-test "menu revert"
  (do-test-menu-message dtmw 'low
    (concatenate 'string "With the middle button In the break window "
      "(not the frame window) , "
      "select REVERT from the menu..
      "
      "You should see the break window re-initialize as "
      "Breakpoint at IL:TTYIN. ")))
(do-test "STOP"
  (do-test-menu-message dtmw 'low
    (concatenate 'string "Enter
    STOP
    "
    "The break window should disappear "
    "and the carot should be back in the exec.")))
(do-test-group ("*short-backtrace-filter*")
  (do-test "*short-backtrace-filter*"
    (do-test-menu-message dtmw 'low
      (concatenate 'string
        "In the exec enter:
        il:*short-backtrace-filter*
        "
        "You should see an appropriate predicate name "
        "such as xcl::interesting-frame-p."))))
(do-test-group ("EDIT")
  (do-test "EDIT search for editable fn (AR 8137)"
    (do-test-menu-message dtmw 'low
      (concatenate 'string "In the exec type (foo). "
        "Enter EDIT in the resulting break window.
        "
        "An editor window for foo should appear. "
        "Close the editor window.")))
  (do-test "EDIT selected fn (AR 6231)"
    (do-test-menu-message dtmw 'low
      (concatenate 'string
        "In the break window, type the DBT command.
        "
        "Select the foo frame from the BT menu window with the mouse. "
        "Type EDIT in the break window. "
        "An editor window should appear.")))
  (do-test "EDIT OK (AR 8139)"
    (do-test-menu-message dtmw 'low
      (concatenate 'string
        "In the editor window, change the (BREAK) to \"hi\".
        "
        "Close the edit window.
        "
        "Type OK in the break window.
        "
        "\"hi\" should be returned in the exec.")))
  (do-test "AR 7957 Spurious WITH-PROFILE call in DEBUGGER"
    (do-test-menu-message dtmw 'low
      (concatenate 'string
        "Open a new INTERLISP exec from the background menu.
        "
        "In that exec type control B to cause a break.
        "
        "See that *PACKAGE* is bound to #<package INTERLISP>
        "
        "Uparrow out of the break window and close the exec.")))
  (do-test "close user interaction window" (do-test-menu-cleanup dtmw) t)

```