File created: 16-May-90 14:44:40 **{DSK}<usr>local>lde>lispcore>sources>CMLSTEP.;2**

changes to: (IL:VARS IL:CMLSTEPCOMS)

previous date: 10-Dec-87 15:11:29 {DSK}<usr>local>lde>lispcore>sources>CMLSTEP.;1

Read Table: XCL

Package: LISP

Format: XCCS


```
(IL:RPAQQ IL:CMLSTEPCOMS
          (

;;; CMLSTEP -- Single Stepper STEP

           (IL:FUNCTIONS STEP)
           (IL:FUNCTIONS STEP-COMMAND STEP-EVAL STEP-FORM STOP-STEPPING STEP-PRINT STEP-PRINT-VALUES)
           (IL:VARIABLES *STEP-IO* *STEP-INDENTATION-INCREMENT* *STEP-INDENTATION-LEVEL* *STEP-PRINT-LEVEL*
                  *STEP-STATE* *STEP-MAX-INDENTATION* *STEP-PRINT-LENGTH*)
           (IL:PROP (IL:FILETYPE IL:MAKEFILE-ENVIRONMENT)
                  IL:CMLSTEP)
           (IL:FUNCTIONS STEP-SLEEP)
           (IL:DECLARE\: IL:DONTEVAL@LOAD IL:DOEVAL@COMPILE IL:DONTCOPY IL:COMPILERVARS (IL:ADDVARS (IL:NLAMA)
                                                                                                   (IL:NLAML)
                                                                                                   (IL:LAMA)))))

;;; CMLSTEP -- Single Stepper STEP


(DEFMACRO STEP (FORM)
   "Evaluate FORM interactively"
   `(LET ((*EVALHOOK* #'STEP-COMMAND)
          (*STEP-STATE* T)
          (*STEP-INDENTATION-LEVEL* 0)
          (*STEP-IO* *QUERY-IO*))
         ,FORM))


(DEFUN STEP-COMMAND (FORM ENVIRONMENT)
   "This is the *EVALHOOK* when stepping.  It prints the form, and then reads a command.  The commands are
   single characters from the terminal.  If the stepper has subsequently been turned off, do the equivalent of
   the s command without printing."
   (COND
      ((NOT *STEP-STATE*)                                          ; If aborted, just eval it.
       (EVAL FORM ENVIRONMENT))
      ((NOT (OR (EQ *STEP-STATE* T)
                (AND (CONSP FORM)
                     (MEMBER (CAR FORM)
                             *STEP-STATE* :TEST 'EQ))))            ; Don't step this form, but keep on looking.
       (EVALHOOK FORM #'STEP-COMMAND NIL ENVIRONMENT))
      (T                                                           ; Otherwise, bind indent level, print form, and enter command
                                                                   ; loop.
         (COND
            ((OR (SYMBOLP FORM)
                 (CONSTANTP FORM)                                  ; Handles quote, not function.

                 )
             (STEP-PRINT FORM)
             (LET ((VALUE (EVAL FORM ENVIRONMENT)))
                  (STEP-PRINT-VALUES (LIST VALUE))
                  VALUE))
            (T (STEP-PRINT FORM)
               (ECASE (IL:ASKUSER NIL NIL ": " '((IL:\  "Step" IL:EXPLAINSTRING "<space> -- Step")
                                                 (IL:N "ext" IL:EXPAINSTRING "Next - Evaluate this expression
                                                       without stepping")
                                                 (IL:F "inish" IL:EXPAINSTRING "Finish - complete evaluation
                                                       without the stepper")
                                                 (IL:D "ebugger")
                                                 (IL:^ " abort")))
                  (IL:\                                            ; Space: step thru the evaluation of this form
                     (STEP-FORM FORM ENVIRONMENT))
                  (IL:N (STEP-EVAL FORM ENVIRONMENT))
                  (IL:F
                     (STOP-STEPPING)
                     (EVAL FORM ENVIRONMENT))
                  (IL:D                                            ; Enter the debugger with this form, but then continue stepping
                                                                   ; afterward.
                     (XCL:DEBUGGER :FORM FORM :ENVIRONMENT ENVIRONMENT :CONDITION (XCL:MAKE-CONDITION
                                                                                  'SI::BREAKPOINT :FUNCTION
                                                                                  FORM))
                     (STEP-COMMAND FORM ENVIRONMENT))
                  (IL:^ (IL:ERROR!)))))))))
```

```
(DEFUN STEP-EVAL (FORM ENVIRONMENT)
   "Evaluate this form (without stepping) and print values"
   (LET ((RESULTS (MULTIPLE-VALUE-LIST (EVAL FORM ENVIRONMENT))))
        (STEP-PRINT-VALUES RESULTS)
        (VALUES-LIST RESULTS)))


(DEFUN STEP-FORM (FORM ENVIRONMENT)
   "Evaluate this form by stepping and print values."
   (LET ((RESULTS (LET ((*STEP-INDENTATION-LEVEL* (+ *STEP-INDENTATION-INCREMENT* *STEP-INDENTATION-LEVEL*)))
                    (MULTIPLE-VALUE-LIST (EVALHOOK FORM #'STEP-COMMAND NIL ENVIRONMENT)))))
        (STEP-PRINT-VALUES RESULTS)
        (VALUES-LIST RESULTS)))


(DEFUN STOP-STEPPING ()
   (SETQ *STEP-STATE* NIL)
   (SETQ *EVALHOOK* NIL))


(DEFUN STEP-PRINT (FORM)
   "Print form according to the current indentation level, and according to *STEP-PRINT-LEVEL* and
   *STEP-PRINT-LENGTH*"
   (LET ((*PRINT-LEVEL* *STEP-PRINT-LEVEL*)
         (*PRINT-LENGTH* *STEP-PRINT-LENGTH*))
        (FORMAT *STEP-IO* "~&~vT~S " (MIN *STEP-INDENTATION-LEVEL* *STEP-MAX-INDENTATION*)
               FORM)))


(DEFUN STEP-PRINT-VALUES (VALUE-LIST)
   "PRINT-VALUES is called to print a list of values which were returned from an evaluation."
   (WHEN VALUE-LIST
       (LET ((*PRINT-LEVEL* *STEP-PRINT-LEVEL*)
             (*PRINT-LENGTH* *STEP-PRINT-LENGTH*))
            (DOLIST (VALUE VALUE-LIST)
                (FORMAT *STEP-IO* "~&~vT= ~S " (MIN *STEP-INDENTATION-LEVEL* *STEP-MAX-INDENTATION*)
                       VALUE))))
   (TERPRI *STEP-IO*))


(DEFVAR *STEP-IO* NIL
   "Stream to which step I/O is done, bound by CL:STEP.")


(DEFVAR *STEP-INDENTATION-INCREMENT* 2
   "Number of spaces to increase indenting.")


(DEFVAR *STEP-INDENTATION-LEVEL* 0)


(DEFVAR *STEP-PRINT-LEVEL* 2
   "Local value")


(DEFVAR *STEP-STATE* NIL
   "EG, enabled")


(DEFVAR *STEP-MAX-INDENTATION* 40)


(DEFVAR *STEP-PRINT-LENGTH* 5
   "Local value")

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

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


(DEFUN STEP-SLEEP (FUNCTIONS)
   "Sleep until the given functions are reached."
```

;;; This is an interesting bit of functionality for step that needs a better interface. It currently exists but will not be documented for the release.

```
   (WHEN (NULL FUNCTIONS)
         (SETQ FUNCTIONS 0))
   (SETQ *STEP-STATE* FUNCTIONS)
   (SETQ *EVALHOOK* #'STEP-COMMAND)
   (SETQ *STEP-INDENTATION-LEVEL* 0))

(IL:DECLARE\: IL:DONTEVAL@LOAD IL:DOEVAL@COMPILE IL:DONTCOPY IL:COMPILERVARS

(IL:ADDTOVAR IL:NLAMA )

(IL:ADDTOVAR IL:NLAML )
```

```
(IL:ADDTOVAR IL:LAMA )
)

(IL:PUTPROPS IL:CMLSTEP IL:COPYRIGHT ("Venue & Xerox Corporation" 1986 1987 1990))
```

FUNCTION INDEX

VARIABLE INDEX

PROPERTY INDEX

MACRO INDEX