

File created: 16-May-90 18:23:44 {DSK}<usr>local>lde>lispcore>sources>INSPECT-CLOSURE.;2

changes to: (VARS INSPECT-CLOSURECOMS)

previous date: 3-Feb-88 15:15:04 {DSK}<usr>local>lde>lispcore>sources>INSPECT-CLOSURE.;1

Read Table: XCL

Package: INTERLISP

Format: XCCS

; Copyright (c) 1988, 1990 by Venue & Xerox Corporation. All rights reserved.

(RPAQQ **INSPECT-CLOSURECOMS** (

;;; A nicer inspector for lexical closures.

```
(FUNCTIONS INSPECT-CLOSURE CLOSURE-PROPERTIES CLOSURE-FETCHFN CLOSURE-STOREFN)
(ADDVARS (INSPECTMACROS ((FUNCTION CLOSURE-P) . INSPECT-CLOSURE))))
```

;;; A nicer inspector for lexical closures.

```
(CL:DEFUN INSPECT-CLOSURE (CLOSURE TYPE WHERE)
  (INSPECTW.CREATE CLOSURE (CLOSURE-PROPERTIES CLOSURE)
    'CLOSURE-FETCHFN
    'CLOSURE-STOREFN NIL NIL NIL NIL NIL NIL #'(CL:LAMBDA (PROP DATUM)
      (CL:IF (NULL (CDR PROP))
        NIL
        (CAR PROP))))))
```

```
(CL:DEFUN CLOSURE-PROPERTIES (CLOSURE)
  "Make up a property description for a closure."
  ;; Does not list fields that aren't present in the closure. Tags the fields present with a dummy field, which the inspect module is kind enough to
  ;; provide.
  (LIST* '("function" FUNCTION) ; The function in the closure.
    (CL:MAPCAN ; Here we compute the properties from the environment.
      #'(CL:LAMBDA (SUB-ENV-NAME SUB-ENV-GET &OPTIONAL (SUB-ENV (CL:FUNCALL SUB-ENV-GET (
        CLOSURE-ENVIRONMENT
        CLOSURE))))
        (CL:WHEN SUB-ENV ; Only display if there's something in this part of the environment.
          (LIST* `',(CL:STRING-DOWNCASE (CL:SYMBOL-NAME SUB-ENV-NAME)))
            ; Dummy field printed in middle.
            (CL:DO ((PLIST SUB-ENV (CDDR PLIST))
              (PROP-SPECS NIL)
              ( (NULL PLIST)
                PROP-SPECS)
              (CL:PUSH `',(CL:FIRST PLIST)
                ,SUB-ENV-NAME)
              PROP-SPECS))))
      ' (VARS FUNCTIONS BLOCKS TAGBODIES)
      ' (ENVIRONMENT-VARS ENVIRONMENT-FUNCTIONS ENVIRONMENT-BLOCKS ENVIRONMENT-TAGBODIES))))
```

```
(CL:DEFUN CLOSURE-FETCHFN (CLOSURE PROP)
  (COND
    ((NULL (CDR PROP))
      (CAR PROP))
    ((EQ (CADR PROP)
      'FUNCTION)
      (CLOSURE-FUNCTION CLOSURE))
    (T (LET (ACCESSOR)
      (CL:IF (SETQ ACCESSOR (CDR (CL:ASSOC (CADR PROP)
        ' ((VARS . ENVIRONMENT-VARS)
          (FUNCTIONS . ENVIRONMENT-FUNCTIONS)
          (BLOCKS . ENVIRONMENT-BLOCKS)
          (TAGBODIES . ENVIRONMENT-TAGBODIES))
        :TEST
        'EQ)))
        (CL:GETF (CL:FUNCALL ACCESSOR (CLOSURE-ENVIRONMENT CLOSURE))
          (CAR PROP)))))))
```

```
(CL:DEFUN CLOSURE-STOREFN (CLOSURE PROP VALUE)
  (COND
    ((NULL (CDR PROP))
      NIL)
    ((EQ (CADR PROP)
      'FUNCTION)
      (CL:SETF (CLOSURE-FUNCTION CLOSURE)
        VALUE))
    (T (LET (ACCESSOR)
      (CL:IF (SETQ ACCESSOR (CDR (CL:ASSOC (CADR PROP)
        ' ((VARS . ENVIRONMENT-VARS)
```

```
(FUNCTIONS . ENVIRONMENT-FUNCTIONS)
(BLOCKS . ENVIRONMENT-BLOCKS)
(TAGBODIES . ENVIRONMENT-TAGBODIES)
:TEST
'EQ))
(LET ((PLIST (CL:FUNCALL ACCESSOR (CLOSURE-ENVIRONMENT CLOSURE))))
  (CL:SETF (CL:GETF PLIST (CAR PROP))
    VALUE))))))

(ADDTOPVAR INSPECTMACROS ((FUNCTION CLOSURE-P) . INSPECT-CLOSURE))

(PUTPROPS INSPECT-CLOSURE COPYRIGHT ("Venue & Xerox Corporation" 1988 1990))
```

FUNCTION INDEX

CLOSURE-FETCHFN1	CLOSURE-PROPERTIES1	CLOSURE-STOREFN1	INSPECT-CLOSURE1
-----------------	--------	--------------------	--------	-----------------	--------	-----------------	--------

VARIABLE INDEX

INSPECTMACROS2
---------------	--------
