

File created: 18-May-90 01:12:06 {DSK}<usr>local>lde>lispcore>sources>WTFIX.;2

changes to: (VARS WTFIXCOMS)

previous date: 24-Mar-88 10:35:24 {DSK}<usr>local>lde>lispcore>sources>WTFIX.;1

Read Table: XCL

Package: INTERLISP

Format: XCCS

; Copyright (c) 1982, 1983, 1985, 1986, 1988, 1990 by Venue & Xerox Corporation. All rights reserved.

```
(RPAQQ WTFIXCOMS ((FNS FINDATOM FIXAPPLY1 FIXLISPX/)
  (SPECVARS TYPE-IN? VARS EXPR PARENT BREAKFLG SUBPARENT FORMSFLG TAIL HELPFIXTAIL HELPFIXFLG
    TYPE-IN? VARS EXPR PARENT)
  (GLOBALVARS CLEARSTKLST SPELLINGS2 USERWORDS SPELLINGS3 LISPXHISTORY)))
```

(DEFINEQ

FINDATOM

(LAMBDA (ATOM POS BLIP) (* |J.Gibbons| " 9-Jun-83 23:27")

;; Returns TAIL for which atom is CAR. If found on *form* BLIP, also sets free variable PARENT to this form.

(PROG (XTAIL TEM NAME REALPOS)

(COND

((NULL BLIP))

((SETQ XTAIL (FMEMB ATOM (LISTP BLIP)))

(SETQ PARENT BLIP)

(SETQ SUBPARENT (AND (OR (EQ (CAR PARENT)

'SETQ)

(EQ (CAR PARENT)

'SAVESETQ))

(CDDR PARENT)))

(RETURN XTAIL)))

;; The EQ checks for no eval-blip on the stack, the NULL check checks for whether ATM is in that binding. In either case, must try something
;; else.

(COND

((EQ (SETQ NAME (STKNAME (SETQ REALPOS (SELECTQ (SYSTEMTYPE)

((D JERICO)

; Skip over internal interpreter frames

(REALSTKNTH -1 POS T))

POS))))

; Error of form (PROG (mumble (VAR U.B.A.) mumble) mumble)

'PROG)

(SOME (CADR (LISTP BLIP))

(FUNCTION (LAMBDA (X)

(AND (LISTP X)

(SETQ XTAIL (FMEMB ATOM (SETQ PARENT X))))))

(SETQ SUBPARENT XTAIL))

((EQ ATOM (CAR (SETQ TEM (BLIPVAL '*TAIL* POS))))

;; U.B.A. occurs at top level of COND clause but in consequent, e.g. (COND (& .. U.B.A. ..)), or at top level of SELECTQ clause,
;; e.g. (SELECTQ & .. (& .. U.B.A. ..)), or at top level of lambda expression.

(SETQ XTAIL TEM)

(COND

((EXPRP NAME)

(SETQ PARENT (COND

((LISTP NAME)

;; open lambda expression, e.g. error occurred at top level of lambda expression which was passed as
;; functional argument and then apply'ed.

NAME)

((LITATOM NAME)

(FGETD NAME)))

(T (HELP 'FINDATOM)))

(SETQ SUBPARENT (AND (EQ XTAIL (CDDR PARENT))

XTAIL)))

((SETQ FORMSFLG (EQ NAME 'COND))

(SETQ PARENT (CAR (SOME (STKARG 1 REALPOS)

(FUNCTION (LAMBDA (X)

(TAILP XTAIL X))))))

((AND (EQ NAME 'PROGN)

(EQ (STKNAME (SETQ REALPOS (FSTKNTH -2 REALPOS REALPOS)))

'SELECTQ)

(SETQ PARENT (CAR (SOME (STKARG 1 REALPOS)

(FUNCTION (LAMBDA (X)

(TAILP XTAIL X))))))

;; SELECTQs are a mess: e.g. (SELECTQ X (key ... atom (expr) ...)) in this case, need to find the tail of the SELECTQ and it
;; is not bound anywhere easily accessible on the stack. SELECTQs are implemented via a call to PROG from APPLY

(SETQ SUBPARENT (AND (EQ XTAIL (CDR PARENT))

XTAIL)))

((AND (LISTP TEM)

(EQ ATOM (CAAR TEM)))

; This is necessary for u.b.a.'s as predicate in COND clause, e.g.

; (COND .. (U.B.A. ..))

(SETQ PARENT (SETQ XTAIL (CAR TEM)))

```

      (SETQ FORMSFLG T))
    (AND (LISTP EXPR)
      (SETQ XTAIL (FMEMB ATOM (SETQ PARENT EXPR))))
    ;; Desperation. Will work for case like SETQ (FOO 8CONS A B) where there is no higher PARENT containing ATM, but it is
    ;; contained in EXPR.
    (SETQ SUBPARENT (AND (EQ XTAIL (CDR PARENT))
      XTAIL))
    (SETQ FORMSFLG T) ; e.g. EDITFNS (' (FOO FIE) --) should go to ((QUOTE --)) not
                      ; (QUOTE ---)
  ))
  (SELECTQ (SYSTEMTYPE)
    ((D JERICHO)
      (RELSTK REALPOS))
    NIL)
  (RETURN (AND PARENT XTAIL))))

```

FIXAPPLY1

```

(LAMBDA (POS OLD NEW *FORM*BLIP) (* |Imm| "11-JUN-81 11:32")
  ;; Analogous to FINDATOM. Finds the XTAIL for which the u.d.f. OLD is CAR, and replaces it with NEW. Also changes places where OLD
  ;; appears on the stack to NEW. Sets PARENT to be the element containing OLD, and returns with the TAIL. Releases POS in all cases.

```

```

  (PROG (XTAIL LST)
    (PROG ((N (STKNARGS POS)))
      LP (COND
        ((ZEROP N)
          (RETURN))
        ((EQ (FSTKARG N POS)
          OLD)
          (SETSTKARG N POS NEW)))
        (SETQ N (SUB1 N))
        (GO LP))
    NX

```

```

  ;; Attempts to find the occurrence of the u.d.f. in the user's program, e.g. if (MAPC X (FUNCTION PRINTNT)) caused the error, this will change
  ;; (FUNCTION PRINTNT) to (FUNCTION PRINT)

```

```

  (COND
    ((EQ (CAAAAR LISPXHISTORY)
      OLD)
      (GO OUT)))
  (SETQ LST (COND
    (TYPE-IN? EXPR)
    (T (OR *FORM*BLIP (BLIPVAL '*FORM* POS)))))

```

```

  ;; POS0 is used because want to find the form from which the function before FAULTAPPLY, e.g. MAPC, was called.

```

```

  (COND
    ((SOME (CDR LST)
      (FUNCTION (LAMBDA (X)
        (AND (EQ (CAR X)
          'FUNCTION)
          (EQ (CADR X)
            OLD)
          (SETQ XTAIL (CDR X))))))
      (SETQ PARENT LST))
    ((AND (LISTP EXPR)
      (SETQ XTAIL (FMEMB OLD EXPR)))

```

```

  ; Second case corresponds to where MAPC ((A B C) PRINTNT) is
  ; typed in.

```

```

    )
    (T (GO OUT)))
  (/RPLNODE XTAIL NEW (CDR XTAIL))
  OUT (RELSTK POS)
  (RETURN XTAIL)))

```

FIXLISPX/

```

(LAMBDA (X FN) ; wt: 25-FEB-76 2 2
  (COND
    ((NULL TYPE-IN?)
      X)
    (T (LISPX/ X FN VARS)))))

```

```

)

```

```

(DECLARE\ : DOEVAL@COMPILE DONTCOPY

```

```

(SPECVARS TYPE-IN? VARS EXPR PARENT BREAKFLG SUBPARENT FORMSFLG TAIL HELPFIXTAIL HELPFIXFLG TYPE-IN? VARS EXPR
  PARENT)

```

```

)

```

```

(DECLARE\ : DOEVAL@COMPILE DONTCOPY

```

```

(GLOBALVARS CLEARSTKLST SPELLINGS2 USERWORDS SPELLINGS3 LISPXHISTORY)
)

```

{MEDLEY}<sources>WTFIX.;1

Page 3

(PUTPROPS **WTFIX COPYRIGHT** ("Venue & Xerox Corporation" 1982 1983 1985 1986 1988 1990))

FUNCTION INDEX

FINDATOM	1	FIXAPPLY1	2	FIXLISPX/	2
----------------	---	-----------------	---	-----------------	---