File created: 20-Feb-2024 09:28:38 **{DSK}<home>larry>il>medley>library>MSANALYZE.;2**

edit by: lmm

previous date: 17-Feb-2024 22:10:56 {DSK}<home>larry>il>medley>library>MSANALYZE.;3

Read Table: INTERLISP

Package: INTERLISP

Format: XCCS

```
(RPAQQ MSANALYZECOMS
       [(PROP FILETYPE MSANALYZE)
        (COMS (FNS VARS FREEVARS CALLS COLLECTFNDATA CALLS3)
              (VARS MSMACROPROPS (NOPACKCALLSFLG))
              (BLOCKS (CALLS CALLS COLLECTFNDATA CALLS3 (LOCALFREEVARS CALLSDATA)
                             (NOLINKFNS . T)
                             (GLOBALVARS INVISIBLEVARS COMPILEUSERFN NOPACKCALLSFLG LAMBDASPLST MSRECORDTRANFLG)
                             )
                      (NIL VARS FREEVARS (LOCALVARS . T)))
              (DECLARE%: EVAL@COMPILE (VARS MS.VERB.TO.NOTICED)
                     DONTCOPY
                     (MACROS MSVBNOTICED)))
        [COMS (FNS ALLCALLS MSINITFNDATA MSPRGE MSPRGMACRO MSPRGCALL MSBINDVAR MSPRGRECORD MSPRGERR
                   MSPRGTEMPLATE1 MSPRGTEMPLATE MSPRGLAMBDA MSPRGLST ADDTO NLAMBDAFNP MSPRGDWIM MSDWIMTRAN)
              (E (MAPC MSFNDATA (FUNCTION RPLACD)))
              (VARS MSFNDATA MSERRORFN (MSRECORDTRANFLG T))
              (ADDVARS (INVISIBLEVARS $$1 $$2 $$3 $$4 $$5 $$6 $$7 $$8 $$9 $$10 $$11 $$12 $$13 $$14 $$15 $$16
                               $$17))
              (DECLARE%: DONTCOPY (MACROS INCLISP LTEMPLATE))
              (BLOCKS (ALLCALLS ALLCALLS ADDTO MSBINDVAR MSDWIMTRAN MSPRGCALL MSPRGDWIM MSPRGE MSPRGMACRO
                                MSPRGERR MSPRGLAMBDA MSPRGLST MSPRGRECORD MSPRGTEMPLATE MSPRGTEMPLATE1 NLAMBDAFNP
                                (NOLINKFNS . T)
                                (LOCALFREEVARS FNNAME ERRORS FNDEF INCLISP ONLYRELS PARENTCONTEXT TOPVARS PARENT
                                               EACHTIME VARS)
                                (GLOBALVARS CLISPARRAY MSERRORFN MSRECORDTRANFLG MSFNDATA INVISIBLEVARS CLISPARRAY
                                               MSTEMPLATES USERTEMPLATES MSRECORDTRANFLG NLAMA NLAML DWIMFLG CLISPTRANFLG
                                               DWIMESSGAG)
                                (NOLINKFNS . T))
                      (NIL MSINITFNDATA NLAMBDAFNP MSPRGDWIM (LOCALVARS . T)
                           (GLOBALVARS NLAMA NLAML MSFNDATA DWIMFLG DWIMESSGAG)))
              (P (PUTDQ? MSWORDNAME (LAMBDA (X)
                                            X]
        (COMS (VARS (MSTEMPLATES (HASHARRAY 160))
                    (USERTEMPLATES (HASHARRAY 10)))
              (FILEVARS INITIALTEMPLATES)
              (GLOBALVARS MSTEMPLATES INITIALTEMPLATES)


;;; INITIALTEMPLATES is not needed after loading up

              [P (MAPC INITIALTEMPLATES (FUNCTION (LAMBDA (X)
                                                          (MAPC (CDR X)
                                                                (FUNCTION (LAMBDA (Y)
                                                                                  (PUTHASH Y (CAR X)
                                                                                           MSTEMPLATES]
              (DECLARE%: EVAL@COMPILE DONTCOPY (PROP MACRO LTEMPLATE)))
        (COMS (FNS MSFINDP)
              (BLOCKS (MSFINDP MSFINDP])

(PUTPROPS MSANALYZE FILETYPE :COMPILE-FILE)

(DEFINEQ

(VARS
  [LAMBDA (FN USEDATABASE)                                          (* lmm%: 29-DEC-75 23 22)
    (CDR (CALLS FN USEDATABASE T])


(FREEVARS
  [LAMBDA (FN USEDATABASE)                                          (* lmm%: 5-DEC-75 11 8)
    (CADDR (CALLS FN USEDATABASE 'FREEVARS])


(CALLS
  [LAMBDA (EXPR USEDATABASE VARSFLG)                                ; Edited 12-Jun-90 17:25 by teruuchi
                                                                    ; This FNS  is for the User Interface Function in
                                                                    ; MSANALYZE(MasterScope)
                                                                    ; Edited by Tomoru Teruuchi(12-June-90 : for AR#10020)
    (PROG (FREES (GLOBALS NIL)
                 FNDEF FLG)
          [COND
             ((AND USEDATABASE (LITATOM EXPR)
                   (GETD 'UPDATEFN))
              (UPDATEFN EXPR NIL 'ERROR)
              [SETQ FREES (GETRELATION EXPR '(USE FREELY]
              [SETQ FREES (SUBSET FREES (FUNCTION (LAMBDA (VAR)  ; This Function is The Predicate whether the variable is global or
```

```
                                                    ; not.
                                                (if (OR (FMEMB VAR GLOBALVARS)
                                                        (EQ (GETPROP VAR 'GLOBALVAR)
                                                            T))
                                                    then (pushnew GLOBALS VAR)
                                                         NIL
                                                  else T]     ; Edited by TT (Date : 8-May-1990)
            (RETURN (LIST [AND (NOT VARSFLG)
                               (GETRELATION EXPR '(CALL NOTERROR]
                          (AND (NEQ VARSFLG 'FREEVARS)
                               (GETRELATION EXPR 'BIND))
                          FREES GLOBALS]
        GETDLP
            (SETQ FNDEF (COND
                          [(LITATOM EXPR)
                           (OR (GETD (OR (GETP EXPR 'BROKEN)
                                         EXPR))
                               (GETP EXPR 'EXPR)
                               (AND (NEQ EXPR (SETQ EXPR (FNCHECK EXPR NIL NIL T)))
                                    (GO GETDLP]
                          (T EXPR)))
            (RETURN (COND
                       ((NULL FNDEF)
                        NIL)
                       ((SUBRP FNDEF)
                        NIL)
                       ((CCODEP FNDEF)
                        (SETQ FNDEF (CALLSCCODE FNDEF))
                        [OR NOPACKCALLSFLG (for X on (CAR FNDEF) do (FRPLACA X (PACK* '; (CAR X)
                                                                                      ';]
                        (FRPLACA (CDR FNDEF)
                                 (NCONC (CADR FNDEF)
                                        (CAR FNDEF)))
                        (SETQ FLG)
                        (CALLS3 (CDDR FNDEF))
                        (CALLS3 (CDDDR FNDEF))
                        (CDR FNDEF))
                       [(EXPRP FNDEF)
```

(* Note that EXPR can be a piece of a function definition, and calls will still work.)

```
                       (RESETVARS ((MSRECORDTRANFLG T))
                                  (RETURN (PROG (CALLSDATA LAMFLG)
                                                [COND
                                                   ((FMEMB (CAR FNDEF)
                                                           LAMBDASPLST)
                                                    (SETQ LAMFLG T)
                                                    (COND
                                                       ((OR (AND (EQ (CAR (CADDR FNDEF))
                                                                     '*)
                                                                 (EQ (CADR (CADDR FNDEF))
                                                                     'DECLARATIONS%:))
                                                            (EQ (CAR (CADDR FNDEF))
                                                                'CLISP%:))
                                                        (MSPRGDWIM FNDEF EXPR FNDEF)))
                                                    (SELECTQ (CAR FNDEF)
                                                        ([LAMBDA NLAMBDA]
                                                            NIL)
                                                        (SETQ FNDEF (OR (AND COMPILEUSERFN (APPLY* COMPILEUSERFN
                                                                                                  NIL FNDEF))
                                                                        FNDEF]
                                                (SETQ CALLSDATA
                                                 (ALLCALLS
                                                  FNDEF LAMFLG
                                                  [UNION (CONSTANT (MSVBNOTICED 'USE 'FREELY))
                                                         (AND (NEQ VARSFLG 'FREEVARS)
                                                              (UNION (CONSTANT (MSVBNOTICED 'BIND))
                                                                     (AND (NULL VARSFLG)
                                                                          (CONSTANT (MSVBNOTICED 'CALL
                                                                                                 'NOTERROR]
                                                  EXPR T))
                                                [SETQ FREES (NCONC FREES (COLLECTFNDATA
                                                                          (CONSTANT (MSVBNOTICED 'USE
                                                                                                 'FREELY]
                                                [SETQ FREES
                                                 (SUBSET FREES (FUNCTION (LAMBDA (VAR)
                                                                           (if (OR (FMEMB VAR GLOBALVARS)
                                                                                   (EQ (GETPROP VAR 'GLOBALVAR)
                                                                                       T))
                                                                               then (pushnew GLOBALS VAR)
                                                                                    NIL
                                                                             else T]
                                                                    ; Edited by TT (Date : 8-May-1990)
                                                (RETURN (LIST [COLLECTFNDATA (CONSTANT (MSVBNOTICED
                                                                                        'CALL
                                                                                        'NOTERROR]
                                                              [COLLECTFNDATA (CONSTANT (MSVBNOTICED
```

```
                                                                    'BIND]
                                                          FREES GLOBALS]
                    (T '?])
```

(COLLECTFNDATA

```
  [LAMBDA (LST)                                                    (* lmm "21-DEC-78 22:56")
    (COND
       ((NLISTP LST)
        (CDR (FASSOC LST CALLSDATA)))
       (T (PROG (VAL)
                (for X in LST do (SETQ VAL (UNION (COLLECTFNDATA X)
                                                  VAL)))
                (RETURN VAL])
```

(CALLS3

```
  [LAMBDA (LST)                                                    (* lmm " 6-JUL-78 00:23")
                                                                   (* lmm%: 13-DEC-75 4 51)
    (PROG (FLG)
          [for X on (CAR LST) do (OR (NOT (FMEMB (CAR X)
                                                 INVISIBLEVARS))
                                     (SETQ FLG (FRPLACA X]
          (COND
             (FLG (FRPLACA LST (DREMOVE NIL (CAR LST])

)

(RPAQQ MSMACROPROPS (ALTOMACRO DMACRO BYTEMACRO MACRO MACRO-FN))

(RPAQQ NOPACKCALLSFLG NIL)

(DECLARE%: DONTEVAL@LOAD DOEVAL@COMPILE DONTCOPY

(BLOCK%: CALLS CALLS COLLECTFNDATA CALLS3 (LOCALFREEVARS CALLSDATA)
       (NOLINKFNS . T)
       (GLOBALVARS INVISIBLEVARS COMPILEUSERFN NOPACKCALLSFLG LAMBDASPLST MSRECORDTRANFLG))

(BLOCK%: NIL VARS FREEVARS (LOCALVARS . T))
)

(DECLARE%: EVAL@COMPILE

(RPAQQ MS.VERB.TO.NOTICED
       ((BIND (NIL BIND ARG))
        [CALL (DIRECTLY CALL EFFECT PREDICATE NLAMBDA)
              (EFFECT EFFECT)
              (INDIRECTLY APPLY STACK)
              (NIL CALL ERROR APPLY EFFECT PREDICATE NLAMBDA STACK)
              (NOTERROR APPLY CALL EFFECT PREDICATE NLAMBDA STACK)
              (PREDICATE PREDICATE)
              (TESTING PREDICATE)
              (VALUE CALL)
              (NLAMBDA NLAMBDA]
        (CREATE (NIL CREATE))
        (DECLARE (LOCALVARS LOCALVARS)
               (NIL LOCALVARS SPECVARS)
               (SPECVARS SPECVARS))
        (FETCH (NIL FETCH))
        (REFERENCE (FIELDS FETCH)
               (FREELY REFFREE)
               (CL:LOCALLY REF)
               (NIL REFFREE REF))
        (REPLACE (NIL REPLACE))
        (SET (FIELDS FETCH REPLACE)
             (FREELY SETFREE)
             (CL:LOCALLY SET)
             (NIL SETFREE SET))
        (SMASH (FIELDS FETCH REPLACE)
               (FREELY SMASHFREE)
               (CL:LOCALLY SMASH)
               (NIL SMASHFREE SMASH))
        (TEST (FREELY TESTFREE)
              (CL:LOCALLY TEST)
              (NIL TESTFREE TEST))
        (USE (FIELDS FETCH REPLACE)
             (FREELY SETFREE SMASHFREE REFFREE TESTFREE)
             (I.S.OPRS CLISP)
             (INDIRECTLY LOCALFREEVARS)
             (CL:LOCALLY SET SMASH REF TEST)
             (NIL SETFREE SET SMASHFREE SMASH REFFREE REF TESTFREE TEST)
             (PREDICATE TEST TESTFREE)
             (PROPNAMES PROP)
             (RECORDS RECORD CREATE)
             (TESTING TEST TESTFREE)
             (VALUE SMASH SMASHFREE REF REFFREE)
             (TYPE TYPE))))
```

```
(DECLARE%: EVAL@COMPILE

(PUTPROPS MSVBNOTICED MACRO [OPENLAMBDA (VERB MOD)
                              (CDR (ASSOC MOD (CDR (ASSOC VERB MS.VERB.TO.NOTICED])
)
)

(DEFINEQ

(ALLCALLS
  [LAMBDA (FNDEF LAMFLG ONLYRELS FNNAME INTERNALFLG EACHTIME)    ; Edited 21-Apr-88 16:31 by jrb:
    (LET ((VARS TOPVARS INCLISP ERRORS (PARENT FNDEF)
                (NOSPELLFLG T))
          (DECLARE (CL:SPECIAL NOSPELLFLG))
          (MSINITFNDATA)
          [COND
             (LAMFLG (MSPRGLAMBDA FNDEF 'ARG))
             (T (MSPRGE FNDEF NIL 'RETURN]
          (COND
             (INTERNALFLG MSFNDATA)
             (T (for X in MSFNDATA when (CDR X) collect (CONS (CAR X)
                                                              (CDR X])

(MSINITFNDATA
  [LAMBDA NIL

          (* MSFNDATA is an association list of the "noticed" types, e.g. CALL, BIND, etc.
          -- the FRPLACD resets the pointer)

    (for Y in MSFNDATA do (FRPLACD Y NIL])

(MSPRGE
  [LAMBDA (EXPR SUPEXPR EVALCONTEXT)                             ; Edited 17-Feb-2024 21:52 by lmm
                                                                 (* lmm "27-May-86 04:44")
    ;; analyzes EXPR;  SUPEXPR is the parent expression and is used in the SHOWUSE case where we are printing occurrances of various things
    ;; rather than updating data base;  EVALCONTEXT is a type of reference for this expression from the template: SMASH etc
    (PROG (TEM CALLED CLISP)
          [COND
             ((NLISTP EXPR)
              (RETURN (COND
                         ((AND (LITATOM EXPR)
                               EXPR
                               (NEQ EXPR T)
                               (NOT (FMEMB EXPR INVISIBLEVARS))) ; A variable reference
                          (COND
                             ([OR (FMEMB EXPR VARS)
                                  (SOME TOPVARS (FUNCTION (LAMBDA (Z)
                                                                 ; bound higher up in the function but but used in a functional
                                                                 ; argument
                                                          (COND
                                                             ((FMEMB EXPR Z)
                                                              (ADDTO 'LOCALFREEVARS EXPR SUPEXPR)
                                                              T]
                                                                 ; Things were added to VARS only if they were 'noticeable'
                              (SELECTQ EVALCONTEXT
                                   ((SMASH TEST SET)
                                        (ADDTO EVALCONTEXT EXPR SUPEXPR))
                                   (CHANGE (ADDTO 'SET EXPR SUPEXPR))
                                   (ADDTO 'REF EXPR SUPEXPR)))
                             (T (SELECTQ EVALCONTEXT
                                    (SMASH (ADDTO 'SMASHFREE EXPR SUPEXPR))
                                    (TEST (ADDTO 'TESTFREE EXPR SUPEXPR))
                                    ((SET CHANGE)
                                         (ADDTO 'SETFREE EXPR SUPEXPR))
                                    (ADDTO 'REFFREE EXPR SUPEXPR]
          (COND
             ((EQ EVALCONTEXT 'SET)                              ; in a 'SET' context, but not a variable
              (MSPRGERR PARENT)))
          (COND
             ((LISTP (SETQ CALLED (CAR EXPR)))
              (MSPRGLAMBDA CALLED NIL (SELECTQ EVALCONTEXT
                                          ((TEST EFFECT SMASH)
                                              EVALCONTEXT)
                                          NIL))
              (SELECTQ (CAR CALLED)
                   (LAMBDA (MSPRGLST (CDR EXPR)
                                     EXPR))
                   NIL)
              (RETURN)))
          [COND
             ((SETQ TEM (LTEMPLATE CALLED))
              (RETURN (MSPRGTEMPLATE EXPR TEM EVALCONTEXT]
```

```
          [COND
             ((NOT (FGETD (OR (GETP CALLED 'BROKEN)
                              CALLED)))
              (COND
                 [[AND DWIMFLG (SETQ TEM (GETPROP CALLED 'CLISPWORD]
                                                          ; E.G.  IF, FOR, etc.
                  (SETQ CLISP (MSDWIMTRAN EXPR))
                  (RETURN (COND
                             (CLISP [COND
                                       (TEM (SELECTQ (CAR TEM)
                                                (RECORDTRAN (OR (MSPRGRECORD EXPR EVALCONTEXT)
                                                                MSRECORDTRANFLG
                                                                (RETURN))
                                                                ; optionally also look at translation

                                                            )
                                                (IFWORD)
                                                (CHANGETRAN (MSPRGE (CADR EXPR)
                                                                    EXPR
                                                                    'CHANGE))
                                                (for X in EXPR
                                                   when [AND (LITATOM X)
                                                             (EQ (CAR TEM)
                                                                 (CAR (GETPROP X 'CLISPWORD]
                                                   do (ADDTO 'CLISP X EXPR]
                                                                ; Analyze the CLISP translation
                                     (PROG ((INCLISP (INCLISP EXPR)))
                                           (MSPRGE CLISP EXPR EVALCONTEXT)))
                                ((OR (NULL (GETPROP (CAR EXPR)
                                                    'CLISPWORD))
                                     (NEQ (CAR EXPR)
                                          CALLED))
                                 (RETURN (MSPRGE EXPR SUPEXPR EVALCONTEXT)))
                                (T (SELECTQ (CAR TEM)
                                       (RECORDTRAN (MSPRGRECORD EXPR EVALCONTEXT))
                                       (CHANGETRAN (MSPRGE (CADR EXPR)
                                                           EXPR
                                                           'CHANGE)
                                                   (MSPRGLST (CDDR EXPR)
                                                             EXPR))
                                       (PROGN                 ; CLISP word wouldn't DWIMIFY
                                              (MSPRGCALL CALLED EXPR EVALCONTEXT)
                                              (MSPRGERR EXPR)
                                              (MSPRGLST (CDR EXPR)
                                                        EXPR]
                   ((SETQ TEM (CL:MACRO-FUNCTION CALLED))
                    (LET ((ME (CL:MACROEXPAND-1 EXPR)))
                         (COND
                            ((AND (NOT (EQUAL ME EXPR))
                                  (NOT (EQUAL ME INCLISP)))
                             (MSPRGCALL (CAR EXPR)
                                    EXPR EVALCONTEXT)
                             (PROG ((INCLISP (INCLISP EXPR))
                                    (EXPR EXPR))
                                   (MSPRGE ME EXPR 'EVAL))
                             (RETURN T]
          (COND
             ((NLAMBDAFNP CALLED)
              (ADDTO 'NLAMBDA CALLED EXPR)
              (COND
                 ((AND MSMACROPROPS (SETQ TEM (GETMACROPROP CALLED MSMACROPROPS))
                       (MSPRGMACRO EXPR TEM EVALCONTEXT))
                  (RETURN)))
              (MSPRGCALL CALLED EXPR EVALCONTEXT))
             (T                                           ; normal lambda function call
                (MSPRGCALL CALLED EXPR EVALCONTEXT)
                (MSPRGLST (CDR EXPR)
                       EXPR
                       'EVAL])
```

(MSPRGMACRO

```
  [LAMBDA (FORM MACDEF CONTEXT)                           ; Edited 17-Feb-2024 21:43 by lmm
                                                          ; Edited 18-Aug-2021 11:23 by larry

    (PROG [(ME (if (LITATOM MACDEF)
                   then (CL:MACROEXPAND-1 FORM)
                   else (MACROEXPANSION FORM MACDEF]
          (COND
             ((AND (NOT (EQUAL ME FORM))
                   (NOT (EQUAL ME INCLISP)))
              (MSPRGCALL (CAR FORM)
                     FORM CONTEXT)
              (PROG ((INCLISP (INCLISP FORM))
                     (EXPR FORM))
                    (MSPRGE ME FORM 'EVAL))
              (RETURN T])
```

(MSPRGCALL

```
[LAMBDA (FN PRNT CONTEXT)                                   (* lmm "22-DEC-78 12:57")
  (ADDTO [COND
           (TOPVARS 'APPLY)
           (T (SELECTQ CONTEXT
                  (TEST 'PREDICATE)
                  (EFFECT 'EFFECT)
                  'CALL]
       FN PRNT])
```

(MSBINDVAR

```
[LAMBDA (VAR TYPE EXPR)                                     (* lmm " 6-JUL-78 00:23")
  (COND
     ((AND VAR (LITATOM VAR)
           (NEQ VAR T))
      [COND
         ((NOT (FMEMB VAR INVISIBLEVARS))
          (ADDTO (OR TYPE 'BIND)
                 VAR
                 (OR EXPR PARENT]
      (SETQ VARS (CONS VAR VARS)))
     (T (MSPRGERR (COND
                    ((LITATOM VAR)
                     (OR EXPR PARENT))
                    (T VAR])
```

(MSPRGRECORD

```
[LAMBDA (PRNT CNTXT)                                        ; Edited  8-Apr-88 14:49 by jrb:
                                                            (* ANALYZE RECORD EXPRESSION PRNT -
                                                            RETURN NIL IF ANALYZED SUCCESSFULLY)

  (PROG (Z)
        (MSPRGTEMPLATE
         PRNT
         (SELECTQ (CAR PRNT)
              ((create CREATE)
                   (ADDTO 'CREATE (CADR PRNT)
                          PRNT)
                   (SETQ Z (CDDR PRNT))
                   [while Z do (COND
                                 ([EQ 'RECORDTRAN (CAR (GETPROP (CAR Z)
                                                                'CLISPWORD]
                                                            (* e.g. USING or COPYING)
                                  (MSPRGE (CADR Z)
                                          PRNT
                                          (SELECTQ (CAR Z)
                                               ((smashing SMASHING)
                                                    'SMASH)
                                               NIL))
                                  (SETQ Z (CDDR Z)))
                                 ((EQ (CADR Z)
                                      '_)                   (* If dwimified correctly, the fields should be separated by _'s)
                                  (ADDTO 'REPLACE (CAR Z)
                                         PRNT)
                                  (MSPRGE (CADDR Z)
                                          PRNT)
                                  (SETQ Z (CDDDR Z)))
                                 ((EQ (CAAR Z)
                                      'SETQ)                (* partially dwimified)
                                  (ADDTO 'REPLACE (CADAR Z)
                                         PRNT)
                                  (MSPRGE (CADDAR Z)
                                          PRNT)
                                  (SETQ Z (CDR Z)))
                                 (T                         (* shouldn't happen, but)
                                    (MSPRGE (CAR Z)
                                            PRNT)
                                    (SETQ Z (CDR Z]
                   (RETURN))
              ((fetch FETCH ffetch FFETCH)
                   [LET [(OF? (OR (EQ (CL:THIRD PRNT)
                                      'OF)
                                  (EQ (CL:THIRD PRNT)
                                      'of]
                        (COND
                           [(EQ CNTXT 'CHANGE)
                            `(NIL (IF LISTP (BOTH (NIL |..| FETCH (BOTH FETCH REPLACE))
                                                  (|..| RECORD NIL))
                                      (BOTH FETCH REPLACE))
                                  ,@(if OF?
                                        then '(NIL EVAL |..| PPE)
                                      else '(EVAL |..| PPE]
                           (T `(NIL (IF LISTP (BOTH (NIL |..| FETCH)
                                                    (|..| RECORD NIL))
                                            FETCH)
```

```
                                ,@(if OF?
                                      then '(NIL EVAL |..| PPE)
                                    else '(EVAL |..| PPE])
                ((REPLACE /REPLACE replace /replace freplace FREPLACE)
                    (LET* [[OF? (OR (EQ (CL:THIRD PRNT)
                                        'OF)
                                    (EQ (CL:THIRD PRNT)
                                        'of]
                                (WITHSLOT (if OF?
                                              then (CL:FIFTH PRNT)
                                            else (CL:FOURTH PRNT)))
                                (WITH? (OR (EQ WITHSLOT 'WITH)
                                           (EQ WITHSLOT 'with]
                               `(NIL (IF LISTP (BOTH (NIL |..| FETCH REPLACE)
                                                     (|..| RECORD NIL))
                                         REPLACE)
                                     ,@(if OF?
                                           then '(NIL)
                                         else NIL)
                                     SMASH
                                     ,@(if WITH?
                                           then '(NIL)
                                         else NIL)
                                     EVAL)))
                ((type? TYPE?)
                    '(CLISP RECORD EVAL . PPE))
                ((initrecord INITRECORD)
                    '(CLISP RECORD . PPE))
                ((WITH with)
                    [COND
                       ((SETQ Z (RECORDFIELDNAMES (CADR PRNT)))
                        (ADDTO 'RECORD (CADR PRNT)
                               PRNT)
                        (MSPRGE (CADDR PRNT)
                               PRNT)
                        (for X
                           in
                           (PROG1 [for X on MSFNDATA when (FMEMB (CAAR X)
                                                                 '(SETFREE TESTFREE REFFREE))
                                      collect (LIST (CAR X)
                                                    (RPLACA X (LIST (CAAR X]
                                  (PROG [ONLYRELS
                                         (EACHTIME (AND EACHTIME
                                                        (for X inside (PROGN EACHTIME)
                                                           when [NOT (FMEMB X '(SETFREE TESTFREE REFFREE]
                                                           collect X]
                                                   (MSPRGLST (CDDDR PRNT)
                                                             PRNT)))
                           do (for Y in (PROG1 (CDR (CAADR X))
                                               (RPLACA (CADR X)
                                                       (CAR X)))
                                 do (ADDTO (COND
                                              ((FMEMB Y Z)
                                               (SELECTQ (CAAR X)
                                                    (SETFREE 'REPLACE)
                                                    'FETCH))
                                              (T (CAAR X)))
                                           Y PRNT)))
                        (RETURN))
                       (T '(RECORD |..| EVAL])
                (RETURN T])
```

(MSPRGERR

```
[LAMBDA (EXPR)                                          (* lmm "21-DEC-78 22:44")
  (SETQ ERRORS T)
  (ADDTO 'ERROR MSERRORFN EXPR])
```

(MSPRGTEMPLATE1

```
[LAMBDA (X TEMPLATE)                                    ; Edited 19-Feb-88 16:56 by jrb:
  (COND
     ((NLISTP TEMPLATE)
      (SELECTQ TEMPLATE
           ((EVAL SMASH TEST EFFECT SET)
                (MSPRGE X PARENT TEMPLATE))
           ((FUNCTION FUNCTIONAL)
```

(* This is a functional arg to something -
the marker FUNCTIONAL means that it will be a separate function while FUNCTION is reserved for those things which compile open -
e.g. MAPC is marked (EVAL FUNCTION FUNCTION . PPE) while SORT is marked
(EVAL FUNCTIONAL . PPE))

```
            [OR (COND
                   ((AND (LISTP X)
                         (NULL (CDDR X)))
```

```
                        (COND
                           ((EQ (CAR X)
                                'F/L)
                            (MSPRGDWIM X FNNAME FNDEF)))
                        (SELECTQ (CAR X)
                            ((FUNCTION QUOTE)
                                 (MSPRGTEMPLATE (CADR X)
                                        (COND
                                           ((LISTP (CADR X))
                                            (SELECTQ TEMPLATE
                                                 (FUNCTIONAL '(REMOTE LAMBDA))
                                                 'LAMBDA))
                                           ((OR (NEQ (CAR X)
                                                     'FUNCTION)
                                                (EQ TEMPLATE 'FUNCTIONAL))
                                            '(REMOTE CALL))
                                           (T 'CALL))
                                        X)
                                 T)
                            NIL)))
                    (EQ X T)
                    (NULL X)
                    (PROGN                                          (* arbitrary expression as functional argument)
                           (ADDTO 'ERROR 'apply PARENT)
                           (MSPRGE X PARENT 'FUNCTIONAL])
               (STACK                                               (* arg to stack fn, e.g. RETFROM)
                      [OR (AND (LISTP X)
                               (SELECTQ (CAR X)
                                    ((FUNCTION QUOTE)
                                         (ADDTO 'STACK (CADR X)
                                                PARENT)
                                         T)
                                    NIL))
                          (PROGN (ADDTO 'ERROR 'stackfn PARENT)
                                 (MSPRGE X PARENT 'EVAL])
               (PROP (COND
                        ((AND (LISTP X)
                              (EQ (CAR X)
                                  'QUOTE))
                         (for Y inside (CADR X) do (ADDTO 'PROP Y PARENT)))
                        (T (MSPRGE X PARENT TEMPLATE))))
               (NIL                                                 (* not used)
                    NIL)
               (RETURN                                              (* this is sometimes the value of PARENT expression)
                       (MSPRGTEMPLATE1 X (SELECTQ PARENTCONTEXT
                                                  ((TEST EFFECT)
                                                       PARENTCONTEXT)
                                                  'EVAL)))
               (TESTRETURN                                          (* if PARENT is tested, then so is this)
                           (MSPRGTEMPLATE1 X (SELECTQ PARENTCONTEXT
                                                      (TEST PARENTCONTEXT)
                                                      'EVAL)))
               (BIND (MSBINDVAR X))
               (LAMBDA (MSPRGLAMBDA X))
               (PPE                                                 (* paren error if not NIL)
                    (COND
                       (X (MSPRGERR PARENT)
                          (MSPRGLST X PARENT))))
               (CALL (MSPRGCALL X PARENT PARENTCONTEXT))
               (EVALQT [COND
                          ((EQ (CAR X)
                               'QUOTE)
                           (MSPRGTEMPLATE (CADR X)
                                  '(REMOTE EVAL)
                                  PARENT))
                          (T (MSPRGE X PARENT 'EVAL])
               (ADDTO TEMPLATE X PARENT T)))
          (T (SELECTQ (CAR TEMPLATE)
                 (IF [PROG ((EXPR X))
                           (DECLARE (SPECVARS EXPR))
                           (MSPRGTEMPLATE1 X (COND
                                                ((COND
                                                    ((LISTP (CADR TEMPLATE))
                                                                    (* ASSERT%: ((REMOTE EVAL) EXPR))
                                                     (EVAL (CADR TEMPLATE)))
                                                    (T (APPLY* (CADR TEMPLATE)
                                                               X)))
                                                 (CADDR TEMPLATE))
                                                (T (CADDDR TEMPLATE])
                 (|..| [COND
                          [(AND (CADR TEMPLATE)
                                (NULL (CDDR TEMPLATE)))           (* Special case to handle most common cases)
                           (MAPC X (FUNCTION (LAMBDA (X)
                                             (MSPRGTEMPLATE1 X (CADR TEMPLATE]
                          (T (FRPTQ (IDIFFERENCE (LENGTH X)
                                                 (LENGTH (CDDR TEMPLATE)))
                                    (MSPRGTEMPLATE1 (CAR X)
```

```
                                    (CADR TEMPLATE))
                                (SETQ X (CDR X)))
                            (MSPRGTEMPLATE1 X (CDDR TEMPLATE])
                (MACRO (ADDTO 'CALL (CAR X)
                              PARENT)
                       (MSPRGMACRO X (CDR TEMPLATE)))
                (BOTH (MSPRGTEMPLATE1 X (CADR TEMPLATE))
                      (MSPRGTEMPLATE1 X (CADDR TEMPLATE)))
                (@ [PROG ((EXPR X))
                         (DECLARE (SPECVARS EXPR))
                         (MSPRGTEMPLATE1 (EVAL (CADR TEMPLATE))
                                         (EVAL (CADDR TEMPLATE])
                (REMOTE (PROG (VARS (TOPVARS (CONS VARS TOPVARS)))
                              (MSPRGTEMPLATE1 X (CADR TEMPLATE))))
                (KEYWORDS ;; KEYWORDS list of keys...
                          ;; Specifies list of legal keywords
                          ;; (FOO (LAMBDA... (BAR :BAZ DREK))) is recorded in the database as
                          ;; (ADDTO 'KEYCALLS '(BAR . :BAZ) '(BAR :BAZ DREK))
                          ;; i.e. there is a table for each keyword relating functions that call functions specifying them.
                          [LET [(LEGAL-KEYS (OR (CDR TEMPLATE)
                                                (GETRELATION (CAR PARENT)
                                                             'KEYACCEPT]
                                            (while X bind (ALLOW-OTHER-KEYS _ (MEMQ '&ALLOW-OTHER-KEYS LEGAL-KEYS))
                                                  KEYSUSED?
                                               do (if (AND (CL:KEYWORDP (CAR X))
                                                           (OR ALLOW-OTHER-KEYS (MEMQ (CAR X)
                                                                                      LEGAL-KEYS)))
                                                      then (ADDTO 'KEYSPECIFY (CAR X)
                                                                  PARENT)
                                                           (SETQ KEYSUSED? T)
                                                    else                          ; log bogus keyword as ppe
                                                         (MSPRGERR PARENT))
                                                  (pop X)
                                                  (if X
                                                      then (MSPRGTEMPLATE1 (CAR X)
                                                                           'EVAL)
                                                           (pop X)
                                                    else (MSPRGERR PARENT)     ; log no value for keyword as ppe
                                                         )
                                               finally (if KEYSUSED?
                                                           then (ADDTO 'KEYCALL (CAR PARENT)
                                                                       PARENT])
                (COND
                   ((LISTP X)
                    (MSPRGTEMPLATE1 (CAR X)
                                    (CAR TEMPLATE))
                    (MSPRGTEMPLATE1 (CDR X)
                                    (CDR TEMPLATE])
```

(MSPRGTEMPLATE

```
[LAMBDA (PARENT TEMPLATE PARENTCONTEXT)                    ; Edited 26-Dec-2021 10:09 by larry
                                                           (* lmm "23-Jul-86 00:15")
  (PROG ((VARS VARS)
         TEM)
        (COND
           [(EQ TEMPLATE 'MACRO)
            (COND
               ((SETQ TEM (GETMACROPROP (CAR PARENT)
                                        MSMACROPROPS))
                (MSPRGMACRO PARENT TEM))
               (T (MSPRGTEMPLATE1 PARENT '(CALL |..| EVAL]
           (T (MSPRGTEMPLATE1 PARENT TEMPLATE])
```

(MSPRGLAMBDA

```
[LAMBDA (EXPR FLG TYPE)                                    ; Edited  3-Jun-88 10:23 by jrb:
  (SELECTQ (CAR (LISTP EXPR))
      (CL:LAMBDA [LET
                  ((PARENT EXPR)
                   (PARENTCONTEXT TYPE)
                   (VARS VARS))
                  [bind (ARGS _ (CADR EXPR))
                        ARG
                        (EVALCOUNT _ 0)
                        KEYS? KEYLIST while ARGS
                     do (SETQ ARG (pop ARGS))
                        ;; We can be in one of two states - keyword scanning or not
                        [COND
                           [(CL:SYMBOLP ARG)
                            ;; Check and see if it's a keyword thingy
                            (COND
```

```
                ((EQ ARG '&KEY)
                 (SETQ KEYS? T))
                [(FMEMB ARG CL:LAMBDA-LIST-KEYWORDS)
                 (SETQ KEYS? NIL)
                 (if (EQ ARG '&ALLOW-OTHER-KEYS)
                     then (ADDTO 'KEYACCEPT (CAR (push KEYLIST '&ALLOW-OTHER-KEYS))
                                 (CADR EXPR]
                (T (if KEYS?
                       then (ADDTO 'KEYACCEPT (CAR (push KEYLIST (MAKE-KEYWORD ARG)))
                                   (CADR EXPR))
                     else (CL:INCF EVALCOUNT))
                   (MSBINDVAR ARG]
           [(CL:CONSP ARG)                          ; Strangely enough they all EVAL their CADRs.
            (MSPRGTEMPLATE1 (CADR ARG)
                   'EVAL)
            (if KEYS?
                then (if (CL:SYMBOLP (CAR ARG))
                         then (MSBINDVAR (CAR ARG))
                              (ADDTO 'KEYACCEPT [CAR (push KEYLIST (MAKE-KEYWORD (CAR ARG]
                                     (CADR EXPR))
                       else (ADDTO 'KEYACCEPT [CAR (push KEYLIST (MAKE-KEYWORD (CAAR ARG]
                                   (CADR EXPR))
                            (MSBINDVAR (CADAR ARG)))
              else (CL:IF (CL:SYMBOLP (CAR ARG))
                          (MSBINDVAR (CAR ARG))
                          (MSBINDVAR (CADAR ARG)))
                   (OR (NULL (CDDR ARG))
                       (NOT (CL:SYMBOLP (CADDR ARG)))
                       (MSBINDVAR ARG]
           (T (MSPRGTEMPLATE1 ARG 'PPE]
finally (if KEYLIST
            then ;; Look at old template; if there isn't one or it looks like one we put out
                 ;; (of the form (EVAL* KEYWORDS keys...))
                 ;; replace it with a new  template (and somehow mark everything that calls FNNAME for reanalysis
                 [LET ((OLDTEMP (GETTEMPLATE FNNAME))
                       (EC EVALCOUNT))
                      ;; First pop off all the EVALs at the front and count them
                      (while (EQ (CAR OLDTEMP)
                                 'EVAL)
                         do (CL:DECF EC)
                            (pop OLDTEMP))
                      (if (OR (NULL OLDTEMP)
                              (EQ (CAR OLDTEMP)
                                  'KEYWORDS))
                          then (pop OLDTEMP)
                               (if (AND (CL:ZEROP EC)
                                        (NULL (CL:SET-DIFFERENCE OLDTEMP KEYLIST))
                                        (NULL (CL:SET-DIFFERENCE KEYLIST OLDTEMP)))
                                   then ; it matches, don't replace it
                                        NIL
                                 else ; It looks like one of ours; replace it
                                      (SETQ KEYLIST (CONS 'KEYWORDS (CL:NREVERSE KEYLIST)))
                                      (while (CL:PLUSP EVALCOUNT)
                                         do (push KEYLIST 'EVAL)
                                            (CL:DECF EVALCOUNT))
                                      (SETTEMPLATE FNNAME KEYLIST)
                                      ;; These templates shouldn't show up in FILES?, since they're solely for
                                      ;; Masterscope internal use...
                                      (UNMARKASCHANGED FNNAME 'TEMPLATES)
                                      ;; The SETTEMPLATE call marked everyone who calls FNNAME to be
                                      ;; reanalyzed; it also marked FNNAME - this gets rid of that
                                      (PUTHASH FNNAME NIL MSCHANGEDARRAY]
                 else ;; It's possible that we created an old template for this function and it no longer has keywords, so we
                      ;; MAY need to delete it
                      (LET ((OLDTEMP (GETTEMPLATE FNNAME)))
                           (while (EQ (CAR OLDTEMP)
                                      'EVAL)
                              do (pop OLDTEMP))
                           (if (EQ (CAR OLDTEMP)
                                   'KEYWORDS)
                               then (SETTEMPLATE FNNAME NIL)
                                    ;; These templates shouldn't show up in FILES?, since they're solely for Masterscope
                                    ;; internal use...
                                    (UNMARKASCHANGED FNNAME 'TEMPLATES)
                                    ;; The SETTEMPLATE call marked everyone who calls FNNAME to be reanalyzed; it
                                    ;; also marked FNNAME - this gets rid of that
                                    (PUTHASH FNNAME NIL MSCHANGEDARRAY]
(MSPRGTEMPLATE1 (CDDR EXPR)
       '(|..| EVAL RETURN])
```

```
        ([LAMBDA NLAMBDA OPENLAMBDA]
             (MSPRGTEMPLATE EXPR '(NIL (IF LISTP (|..| BIND)
                                           (IF (PROGN EXPR) BIND))
                                       |..| EFFECT RETURN)
                    TYPE))
        (PROG (CLISP TEM)
              (COND
                 ((AND (SETQ TEM (ASSOC (CAR EXPR)
                                        LAMBDATRANFNS))
                       (SETQ CLISP (CL:FUNCALL (CADR TEM)
                                               EXPR)))
                  (PROG ((INCLISP (INCLISP EXPR)))
                        (MSPRGLAMBDA CLISP FLG T)))
                 ((AND DWIMFLG (SETQ CLISP (MSDWIMTRAN EXPR)))   (* has a CLISP translation (e.g. DLAMBDA))
                  (PROG ((INCLISP (INCLISP EXPR)))               (* rebind INCLISP, and try again on the translation)
                        (MSPRGLAMBDA CLISP FLG TYPE)))
                 (T (MSPRGERR EXPR)
                    (MSPRGE EXPR])
```

(MSPRGLST

```
  [LAMBDA (L PARNT CNTX)                                          (* lmm "27-JUN-78 01:57")
    (for X in L do (MSPRGE X PARNT CNTX])
```

(ADDTO

```
  [LAMBDA (RELATION WHAT PRNT FLG)                                (* lmm "24-DEC-78 11:51")
    (PROG ((PTR (FASSOC RELATION MSFNDATA)))
          [OR PTR (COND
                     (FLG (RETURN))
                     (T (SHOULDNT]
          (OR (NULL ONLYRELS)
              (FMEMB RELATION ONLYRELS)
              (RETURN))
          (AND EACHTIME (EQMEMB RELATION (CAR EACHTIME))
               (APPLY* (CADR EACHTIME)
                       WHAT
                       (CADDR EACHTIME)
                       (CADDDR EACHTIME)
                       PRNT INCLISP))
      LP  (COND
             ((NULL (CDR PTR))
              (FRPLACD PTR (LIST WHAT)))
             ((EQ (CAR (SETQ PTR (CDR PTR)))
                  WHAT)
              (RETURN))
             (T (GO LP])
```

(NLAMBDAFNP

```
  [LAMBDA (FN)                                                    (* lmm "26-Mar-85 12:37")
    (AND [NOT (EQMEMB 'EVAL (GETPROP FN 'INFO]
         (COND
            ([OR (FGETD (SETQ FN (OR (GETPROP FN 'BROKEN)
                                     FN)))
                 (SETQ FN (GETLIS FN '(EXPR CODE]
```

(* if the function is defined, check its argtype to tell you whether it is NLAMBDA or LAMBDA)

```
            (SELECTQ (ARGTYPE FN)
                 ((1 3)
                      T)
                 NIL))
            (T                                                    (* otherwise, rely on NLAMA or NLAML)
               (OR (FMEMB FN NLAMA)
                   (FMEMB FN NLAML])
```

(MSPRGDWIM

```
  [LAMBDA (X FN DEF)                                              ; Edited  8-Apr-88 11:55 by jrb:
    (AND DWIMFLG (LET ((NOSPELLFLG T)
                       FILEPKGFLG)
                      (DECLARE (SPECVARS NOSPELLFLG))
                      (RESETVARS ((DWIMESSGAG T))
```

;; JRB Made these RESETVARS to placate the PavCompiler

```
                                 (PROG (LISPXHIST)                (* ASSERT%: ((REMOTE EVAL) DWIMESSGAG FILEPKGFLG
                                  NOSPELLFLG))
                                       (DWIMIFY0 X (OR (AND (LITATOM FN)
                                                            FN)
                                                       '?)
                                             VARS DEF])
```

(MSDWIMTRAN

```
  [LAMBDA (EXPR)                                                  (* DD%: "28-DEC-81 13:46")
    (AND DWIMFLG (COND
```

```
                    ((AND CLISPARRAY (GETHASH EXPR CLISPARRAY)))
                    ((AND CLISPTRANFLG (EQ (CAR (LISTP EXPR))
                                           CLISPTRANFLG))
                     (CADR EXPR))
                    (T (MSPRGDWIM EXPR FNNAME FNDEF)
                       (OR (AND CLISPARRAY (GETHASH EXPR CLISPARRAY))
                           (AND CLISPTRANFLG (EQ (CAR (LISTP EXPR))
                                                 CLISPTRANFLG)
                                (CADR EXPR])

)

(RPAQQ MSFNDATA
       ((BIND)
        (CALL)
        (EFFECT)
        (PREDICATE)
        (CLISP)
        (PROP)
        (SETFREE)
        (SET)
        (SMASHFREE)
        (SMASH)
        (REFFREE)
        (REF)
        (FETCH)
        (REPLACE)
        (RECORD)
        (ERROR)
        (ARG)
        (CREATE)
        (LOCALVARS)
        (SPECVARS)
        (APPLY)
        (TESTFREE)
        (TEST)
        (LOCALFREEVARS)
        [NLAMBDA]
        (TYPE)
        (STACK)
        (KEYACCEPT)
        (KEYSPECIFY)
        (KEYCALL)
        (FLET)
        (LABEL)
        (MACROLET)
        (COMPILER-LET)))

(RPAQQ MSERRORFN ppe)

(RPAQQ MSRECORDTRANFLG T)

(ADDTOVAR INVISIBLEVARS $$1 $$2 $$3 $$4 $$5 $$6 $$7 $$8 $$9 $$10 $$11 $$12 $$13 $$14 $$15 $$16 $$17)

(DECLARE%: DONTCOPY

(DECLARE%: EVAL@COMPILE

(PUTPROPS INCLISP MACRO ((.X.)
                         (COND
                            ((AND INCLISP EACHTIME (NOT (MSFINDP INCLISP .X.)))
                             INCLISP)
                            (T .X.))))

(PUTPROPS LTEMPLATE MACRO [LAMBDA (Y)
                           (DECLARE (LOCALVARS Y))
                           (AND [NEQ T (SETQ Y (OR (GETHASH Y USERTEMPLATES)
                                                   (GETHASH Y MSTEMPLATES]
                                Y])
)
)

(DECLARE%: DONTEVAL@LOAD DOEVAL@COMPILE DONTCOPY

(BLOCK%: ALLCALLS ALLCALLS ADDTO MSBINDVAR MSDWIMTRAN MSPRGCALL MSPRGDWIM MSPRGE MSPRGMACRO MSPRGERR MSPRGLAMBDA
       MSPRGLST MSPRGRECORD MSPRGTEMPLATE MSPRGTEMPLATE1 NLAMBDAFNP (NOLINKFNS . T)
       (LOCALFREEVARS FNNAME ERRORS FNDEF INCLISP ONLYRELS PARENTCONTEXT TOPVARS PARENT EACHTIME VARS)
       (GLOBALVARS CLISPARRAY MSERRORFN MSRECORDTRANFLG MSFNDATA INVISIBLEVARS CLISPARRAY MSTEMPLATES
              USERTEMPLATES MSRECORDTRANFLG NLAMA NLAML DWIMFLG CLISPTRANFLG DWIMESSGAG)
       (NOLINKFNS . T))

(BLOCK%: NIL MSINITFNDATA NLAMBDAFNP MSPRGDWIM (LOCALVARS . T)
       (GLOBALVARS NLAMA NLAML MSFNDATA DWIMFLG DWIMESSGAG))
)

[PUTDQ? MSWORDNAME (LAMBDA (X)
                    X]
```

```
(RPAQ MSTEMPLATES (HASHARRAY 160))

(RPAQ USERTEMPLATES (HASHARRAY 10))

(RPAQQ INITIALTEMPLATES
       (((CALL (REMOTE (IF LITATOM CALL LAMBDA))
               (IF LITATOM EVAL NIL))
         FUNCTION)
        ((NIL NIL |..| EVAL RETURN)
         CL:BLOCK)
        ((CALL |..| EVAL)
         CL:CATCH CL:THROW)
        ((NIL NIL EVAL)
         CL:RETURN-FROM)
        ([IF (EQ (CADR EXPR)
                 'ASSERT%:)
             (NIL NIL |..| (IF LISTP (@ (CDR EXPR)
                                        (LIST '|..| (MSWORDNAME (CAR EXPR]
         *)
        ((NIL (BOTH (|..| (IF LISTP (NIL EVAL |..| EFFECT)
                              NIL))
                    (|..| (IF LISTP (BIND)
                              BIND)))
              |..| EFFECT RETURN)
         LET CL:COMPILER-LET)
        ((NIL (|..| (IF LISTP (BIND EVAL . PPE)
                        BIND))
              |..| EFFECT RETURN)
         LET*)
        ((NIL |..| (IF LISTP EFFECT))
         CL:TAGBODY)
        ((NIL (BOTH (|..| (IF LISTP (NIL EVAL |..| EFFECT)
                              NIL))
                    (|..| (IF LISTP (BIND)
                              BIND)))
              |..|
              (IF LISTP EFFECT))
         PROG)
        (MACRO RESETVARS)
        ((CALL EVAL)
         XNLSETQ NLSETQ ERSETQ)
        ((CALL |..| EVAL)
         RESETFORM FRPTQ)
        ((CALL EVAL EVAL FUNCTIONAL FUNCTIONAL . PPE)
         MAP2C)
        ((CALL EVAL EVAL SMASH . PPE)
         /DSUBST DSUBST)
        ((CALL EVAL FUNCTION FUNCTION . PPE)
         MAPCAR MAPCON MAPCONC MAPLIST SUBSET EVERY NOTEVERY ANY NOTANY SOME MAPC MAP)
        ((CALL EVAL FUNCTIONAL . PPE)
         MAPHASH)
        ((CALL EVAL PROP . PPE)
         GETP GETLIS GET GETPROP LISTGET LISTGET1 REMPROP /REMPROP)
        ((CALL EVAL PROP EVAL . PPE)
         PUT /PUT PUTPROP /PUTPROP LISTPUT LISTPUT1)
        ((CALL EVAL SMASH . PPE)
         /ATTACH ATTACH)
        ((CALL FUNCTIONAL . PPE)
         MAPATOMS)
        ((CALL FUNCTIONAL |..| EVAL)
         APPLY* BLKAPPLY* APPLY BLKAPPLY)
        ((CALL EVAL SMASH . PPE)
         DREMOVE /DREMOVE)
        ((CALL SET EVAL EVAL . PPE)
         RESETVAR)
        ((CALL SET EVAL . PPE)
         SETN)
        ((CALL SMASH . PPE)
         DREVERSE)
        ((CALL SMASH EVAL . PPE)
         RPLACD /RPLACD RPLACA /RPLACA RPLNODE2 /RPLNODE2 FRPLACD FRPLNODE2 TCONC /TCONC LCONC /LCONC NCONC1
         /NCONC1 FRPLACA)
        ((CALL SMASH EVAL EVAL . PPE)
         RPLNODE FRPLNODE /RPLNODE)
        ((CALL SMASH FUNCTIONAL . PPE)
         SORT)
        ((CALL (BOTH SET EVAL) . PPE)
         ADD1VAR SUB1VAR)
        ((CALL (IF NULL NIL (IF ATOM SET EVAL))
               EVAL . PPE)
         RESETSAVE)
        ((CALL (IF (EQ (CAR (LISTP EXPR))
                       'QUOTE)
                   (NIL SET)
                   EVAL)
               EVAL . PPE)
```

```
 SET /SET SETTOPVAL /SETTOPVAL SETATOMVAL /SETATOMVAL)
((CALL (IF (EQ (CAR (LISTP EXPR))
               'QUOTE)
           (NIL SET)
           EVAL)
       EVAL EVAL EVAL . PPE)
 SAVESET)
((CALL (IF (EQ (CAR (LISTP EXPR))
               'QUOTE)
           (NIL EVAL)
           EVAL)
       |..| EVAL)
 GETATOMVAL EVAL EVALV)
((NIL |..| TESTRETURN RETURN)
 OR)
((NIL |..| TEST RETURN)
 AND)
((NIL |..| EFFECT RETURN)
 PROGN)
((NIL TEST RETURN RETURN)
 CL:IF)
((NIL |..| (IF CDR (TEST |..| EFFECT RETURN)
               (TESTRETURN . PPE)))
 COND)
([CALL |..| (@ EXPR (CONS NIL (SELECTQ (CAR (LISTP EXPR))
                                       (LOCALVARS '(IF LISTP (|..| LOCALVARS)
                                                       LOCALVARS))
                                       (SPECVARS '(IF LISTP (|..| SPECVARS)
                                                      SPECVARS))
                                       NIL]
 DECLARE)
((NIL RETURN)
 CLISP% )
((NIL EVAL . PPE)
 LISTP NLISTP RETURN)
((NIL TEST . PPE)
 NOT NULL)
((CALL EVAL |..| (NIL |..| EFFECT RETURN)
       RETURN)
 SELECTQ SELCHARQ)
((CALL EVAL |..| (EVAL |..| EFFECT RETURN)
       RETURN)
 SELECTC)
((CALL EVAL |..| ((IF LISTP (|..| EVAL)
                      EVAL)
                  |..| EFFECT RETURN)
       RETURN)
 SELECT)
((NIL EVAL EVAL . PPE)
 EQ NEQ)
((NIL NIL . PPE)
 QUOTE GO)
((NIL EVAL . PPE)
 CAR CDR CAAR CADR CDAR CDDR CAAAR CAADR CADAR CDAAR CADDR CDADR CDDAR CDDDR CAAAAR CAAADR CAADAR CAADDR
 CADAAR CADADR CADDAR CADDDR CDAAAR CDAADR CDADAR CDADDR CDDAAR CDDADR CDDDAR CDDDDR)
((NIL RETURN |..| EFFECT)
 PROG1)
((NIL SET NIL . PPE)
 SETQQ)
((NIL SET EVAL . PPE)
 SETQ ADV-SETQ SAVESETQ)
([@ EXPR (CONS NIL (MAPCON (CDR EXPR)
                           [FUNCTION (LAMBDA (X)
                                             (if (LITATOM (CAR X))
                                                 then
                                                 (LIST 'SET 'EVAL)
                                                 else
                                                 (LIST 'SMASH 'EVAL]
                           (FUNCTION (LAMBDA (X)
                                             (CDDR X]
 CL:SETQ CL:SETF)
((CALL EVAL (BOTH (@ 'RPTN 'BIND)
                  RETURN) . PPE)
 RPTQ)
((CALL EVALQT |..| EVAL)
 EVAL ERRORSET)
((BOTH [IF (EQ (CAR (LISTP (CADDR EXPR)))
               'QUOTE)
           (NIL NIL (NIL (|..| (BIND]
       (CALL EVALQT EVAL . PPE))
 EVALA)
((CALL EVALQT STACK STACK EVAL EVAL . PPE)
 ENVEVAL)
((CALL FUNCTIONAL EVALQT STACK STACK EVAL EVAL . PPE)
 ENVAPPLY)
((CALL STACK EVAL EVAL EVAL . PPE)
 STKAPPLY)
```

```
        ((CALL STACK EVALQT EVAL EVAL . PPE)
         RETEVAL STKEVAL)
        ((CALL STACK EVAL EVAL . PPE)
         RETFROM RETTO)
        ((NIL NIL RETURN)
         THE)))

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS MSTEMPLATES INITIALTEMPLATES)
)
```

;;; INITIALTEMPLATES is not needed after loading up

```
[MAPC INITIALTEMPLATES (FUNCTION (LAMBDA (X)
                                   (MAPC (CDR X)
                                         (FUNCTION (LAMBDA (Y)
                                                     (PUTHASH Y (CAR X)
                                                            MSTEMPLATES]

(DECLARE%: EVAL@COMPILE DONTCOPY

(PUTPROPS LTEMPLATE MACRO [LAMBDA (Y)
                            (DECLARE (LOCALVARS Y))
                            (AND [NEQ T (SETQ Y (OR (GETHASH Y USERTEMPLATES)
                                                    (GETHASH Y MSTEMPLATES]
                                 Y])
)

(DEFINEQ

(MSFINDP
  [LAMBDA (STRUC SUBL)                                    ; Edited 18-Aug-2021 10:56 by larry
    (PROG NIL
      LP  (RETURN (OR (EQ SUBL STRUC)
                      (AND (LISTP STRUC)
                           (OR (MSFINDP (CAR STRUC)
                                        SUBL)
                               (PROGN (SETQ STRUC (CDR STRUC))
                                      (GO LP])

)

(DECLARE%: DONTEVAL@LOAD DOEVAL@COMPILE DONTCOPY

(BLOCK%: MSFINDP MSFINDP)
)
```

FUNCTION INDEX

ADDTO 11
ALLCALLS 4
CALLS 1
CALLS3 3
COLLECTFNDATA 3
FREEVARS 1
MSBINDVAR 6
MSDWIMTRAN 11
MSFINDP 15
MSINITFNDATA 4
MSPRGCALL 6
MSPRGDWIM 11
MSPRGE 4
MSPRGERR 7
MSPRGLAMBDA 9
MSPRGLST 11
MSPRGMACRO 5
MSPRGRECORD 6
MSPRGTEMPLATE 9
MSPRGTEMPLATE1 7
NLAMBDAFNP 11
VARS 1

VARIABLE INDEX

INITIALTEMPLATES 13
INVISIBLEVARS 12
MS.VERB.TO.NOTICED 3
MSERRORFN 12
MSFNDATA 12
MSMACROPROPS 3
MSRECORDTRANFLG 12
MSTEMPLATES 13
NOPACKCALLSFLG 3
USERTEMPLATES 13

MACRO INDEX

INCLISP 12
LTEMPLATE 12,15
MSVBNOTICED 4

PROPERTY INDEX

MSANALYZE 1