```
(RPAQQ MACROAUXCOMS
       ((EXPORT (DECLARE%: DONTCOPY (MACROS NNLITATOM \NULL.OR.FIXP \CHECKTYPE CANONICAL.TIMERUNITS))
                (PROP DMACRO \MACRO.EVAL)
                (OPTIMIZERS \MACRO.MX))
        (COMS                                                  ; functions which help macro and compiler writers.
              (FNS LISPFORM.SIMPLIFY NO.SIDEEFFECTS.FNP CODE.SUBST CODE.SUBPAIR)
              (GLOBALRESOURCES \NSE.STRPTR))
        (COMS (FNS ARGS.COMMUTABLEP ARGS.COMMUTABLEP.LIST VAR.NOT.USED \VARNOTUSED \VARNOTUSED.LIST
                   EVALUABLE.CONSTANTP EVALUABLE.CONSTANT.FIXP)
              (MACROS EVALUABLE.CONSTANT.FIXP CARCDR.FNP))
        (FNS \DECL.COMNT.PROCESS)
        (COMS (FNS \WALKOVER.SPECIALFORMS \WALKOVER.SF.LIST \WALKOVER.FUNCTION)
              (DECLARE%: DONTCOPY (CONSTANTS \QUOTIFYING.NLS \WALKABLE.SPECIALFORMS)
                     (MACROS \WALKABLE.SPECIALFORMP))
              (ADDVARS (CONSTANTFOLDFNS IMIN IMAX IABS LOGOR LOGXOR LOGAND))
              (VARS NOSIDEFNS)
              (GLOBALVARS CLISPARRAY CONSTANTFOLDFNS))
        (PROP FILETYPE MACROAUX)))

;; FOLLOWING DEFINITIONS EXPORTED

(DECLARE%: DONTCOPY

(DECLARE%: EVAL@COMPILE

(PUTPROPS NNLITATOM MACRO (OPENLAMBDA (X)
                              (AND X (LITATOM X))))

(PUTPROPS \NULL.OR.FIXP MACRO (OPENLAMBDA (X)
                                  (OR (NULL X)
                                      (FIXP X))))

(PUTPROPS \CHECKTYPE MACRO [X (PROG ((VAR (CAR X))
                                     (PRED (CADR X)))
                                    (if [AND (LISTP PRED)
                                             (MEMB (CAR PRED)
                                                   ''FUNCTION]
                                        then (SETQ PRED (LIST (CADR PRED)
                                                              VAR)))
                                    (RETURN (SUBPAIR '(MSG VAR PRED)
                                                     (LIST (CONCAT "
                                                             is not a suitable value for the variable:  "
                                                           VAR)
                                                         VAR PRED)
                                                   '(until PRED do (SETQ VAR (ERROR VAR MSG])

(PUTPROPS CANONICAL.TIMERUNITS MACRO (OPENLAMBDA (X)          (* Checks for common abbreviations before calling
                                                             \CanonicalizeTimerUnits)
                                         (SELECTQ X
                                             ((TICKS MILLISECONDS SECONDS)
                                                             (* These are the canonical forms)
                                                 X)
                                             (NIL 'MILLISECONDS)
                                             (\CanonicalizeTimerUnits X))))
)
)

(PUTPROPS \MACRO.EVAL DMACRO [Z (PROG ((X (EXPANDMACRO (CAR Z)
                                                  T)))
                                      (if (EQ X (CAR Z))
                                          then (ERROR "No macro property -- \MACRO.EVAL" X)
                                        else (RETURN (EVAL X])

(DEFOPTIMIZER \MACRO.MX (FORM)
                        FORM)

;; END EXPORTED DEFINITIONS
;; functions which help macro and compiler writers.
```

(DEFINEQ

(**LISPFORM.SIMPLIFY**
  [LAMBDA (X EVALFLG)   (* lmm "11-Jul-85 02:46")

        (* Reduce some LISP code to its more primitive form. Currently, supporst macroexpansion, dwimmification, and evaluation
        of compile-time constants.)

    (**if** (LISTP X)
        then (LET ((FN (CAR X))
                  Y)
                 (COND
                   ((NOT (LITATOM FN))
                    X)
                   ((AND EVALFLG (GETD FN))
                    X)
                   ((SETQ Y (GETMACROPROP FN COMPILERMACROPROPS))
                    (**if** (EQ X (SETQ X (MACROEXPANSION X Y)))
                        then X
                      else (**LISPFORM.SIMPLIFY** X)))
                   ([AND (OR (SETQ Y (GETHASH X CLISPARRAY))
                             (DWIMIFY0? X X X NIL T "LISPFORM.SIMPLIFY")
                             (SETQ Y (GETHASH X CLISPARRAY]
                    (**LISPFORM.SIMPLIFY** Y))
                   ((SETQ Y (CONSTANTEXPRESSIONP X))
                    (KWOTE (CAR Y)))
                   (T X)))
      else (**if** EVALFLG
              then X
            else (LET ((CE (CONSTANTEXPRESSIONP X)))
                     (**if** CE
                         then (CAR CE)
                       else X])

(**NO.SIDEEFFECTS.FNP**
  [LAMBDA (X)   (* edited%: "14-May-86 15:12")
                                                                        (* Fast-case-test for simple memory access fns)

    (AND (NNLITATOM X)
         (OR (GETPROP X 'CROPS)
             (FMEMB X NOSIDEFNS]))

(**CODE.SUBST**
  [LAMBDA (X Y FORM)   (* JonL "21-NOV-82 14:24")
                                                                        (* Ho Hum, someday this ought to be made to work!)

    (SUBST X Y FORM])

(**CODE.SUBPAIR**
  [LAMBDA (L1 L2 FORM)   (* JonL "21-NOV-82 14:24")
                                                                        (* Ho Hum, someday this ought to be made to work!)

    (SUBPAIR L1 L2 FORM])
)

(DECLARE%: DONTCOPY

(DECLARE%: EVAL@COMPILE

[PUTDEF '\NSE.STRPTR 'RESOURCES '(NEW (ALLOCSTRING 0]
)
)

(/SETTOPVAL '\\NSE.STRPTR.GLOBALRESOURCE NIL)

(DEFINEQ

(**ARGS.COMMUTABLEP**
  [LAMBDA (X Y)   (* lmm "11-Jul-85 02:48")

        (* non-NIL iff the evaluation of X and Y can be done in either order without any change in effects or value.)

    (PROG (FN)
          [**if** (NLISTP Y)
              then (**if** (NLISTP X)
                          then

        (* If both args are atoms, then we can just punt out here with the answer.)

                          (RETURN T))

        (* Switch args so that we don't have to handle the case of Y an atom)

                  (SETQ X (PROG1 Y (SETQ Y X]
          (**if** (**if** (LISTP X)
                  then   (* Fast check for quoted frobs. Remember, Y can't be an atom.)

```
                           (MEMB (CAR X)
                                 \QUOTIFYING.NLS)
                    else                                        (* Cases like random, non-variable atoms)
                           (NOT (NNLITATOM X)))
                  then (RETURN T))
           (SETQ Y (LISPFORM.SIMPLIFY Y T))
           (RETURN (if (LISTP (SETQ FN (CAR Y)))
                     then (if (EQ (CAR FN)
                                  'LAMBDA)
                             then (ARGS.COMMUTABLEP.LIST Y (LISPFORM.SIMPLIFY X T)))
                   elseif (MEMB FN \QUOTIFYING.NLS)
                     then 'T
                   elseif (EQ FN 'SETQ)
                     then (AND (\VARNOTUSED X (CADR Y))
                               (ARGS.COMMUTABLEP.LIST (CDDR Y)
                                     (LISPFORM.SIMPLIFY X T)))
                   elseif (\WALKABLE.SPECIALFORMP FN)
                     then (\WALKOVER.SPECIALFORMS (FUNCTION ARGS.COMMUTABLEP)
                                  Y
                                  (LISPFORM.SIMPLIFY X T))
                     else (AND (NO.SIDEEFFECTS.FNP FN)
                               (ARGS.COMMUTABLEP.LIST (CDR Y)
                                     (LISPFORM.SIMPLIFY X T])
```

(**ARGS.COMMUTABLEP.LIST**
```
  [LAMBDA (L Y)                                                 (* JonL "21-NOV-82 15:07")
    (EVERY L (FUNCTION (LAMBDA (X)
                        (ARGS.COMMUTABLEP X Y])
```

(**VAR.NOT.USED**
```
  [LAMBDA (FORM VAR SETQONLY?)                                  (* JonL "21-NOV-82 14:01")
    (PROG NIL
      A   (if (NOT (LITATOM VAR))
             then (SETERRORN 14 VAR)
                  (SETQ VAR (ERRORX))
                  (GO A))
          (if (MEMB VAR '(NIL T))
             then (SETERRORN 27 VAR)
                  (SETQ VAR (ERRORX))
                  (GO A))
          (RETURN (\VARNOTUSED FORM VAR SETQONLY?])
```

(**VARNOTUSED**
```
  [LAMBDA (FORM VAR SETQONLY?)                                  (* JonL "21-NOV-82 16:10")
                                                                (* Look for free occurances of a variable VAR which may be
                                                                evaluable in FORM)

    (if (NLISTP FORM)
       then (AND (NOT SETQONLY?)
                 (NEQ VAR FORM))
     elseif (LISTP (CAR FORM))
       then (\VARNOTUSED.LIST FORM VAR SETQONLY?)
     elseif (EQ (CAR FORM)
                'LAMBDA)
       then

           (* Note that if a LAMBDA form bind a var X, then VAR can't be "used inside" the form.)

            (OR (MEMB VAR (CADR FORM))
                (\VARNOTUSED (CDDR FORM)
                    VAR SETQONLY?))
     elseif (MEMB (CAR FORM)
                  \QUOTIFYING.NLS)
       then T
     elseif (MEMB (CAR FORM)
                  '(SETQ))
       then                                                     (* Stupid Interlisp SETQ format --
                                                                You really wound't believe it!)
            (AND (NEQ VAR (CADR FORM))
                 (\VARNOTUSED.LIST FORM VAR SETQONLY?))
     elseif (\WALKABLE.SPECIALFORMP (CAR FORM))
       then (\WALKOVER.SPECIALFORMS (FUNCTION \VARNOTUSED)
                   FORM VAR SETQONLY?)
     elseif (NO.SIDEEFFECTS.FNP (CAR FORM))
       then (\VARNOTUSED.LIST (CDR FORM)
                   VAR SETQONLY?])
```

(**VARNOTUSED.LIST**
```
  [LAMBDA (L X SETQONLY?)                                       (* JonL "21-NOV-82 15:06")
    (EVERY L (FUNCTION (LAMBDA (FORM)
                        (\VARNOTUSED FORM X SETQONLY?])
```

(**EVALUABLE.CONSTANTP**

```
   [LAMBDA (X)                                                           (* lmm "12-Apr-85 09:42")
      (if (OR (NLISTP X)
              (EQ (CAR X)
                  'QUOTE)
              (EQ (CAR X)
                  'CONSTANT)
              (FMEMB (CAR X)
                     CONSTANTFOLDFNS))
          then                                                     (* Unfortunately, CONSTANT has a macro property which may
                                                                   conflict with the action of LISPFORM.SIMPLIFY)

              (CONSTANTEXPRESSIONP X)
        else (if (LISTP X)
                 then (SETQ X (LISPFORM.SIMPLIFY X T)))
             (if (NLISTP X)
                 then (CONSTANTEXPRESSIONP X)
               elseif (NNLITATOM (CAR X))
                 then [if (\WALKABLE.SPECIALFORMP (CAR X))
                          then (if (\WALKOVER.SPECIALFORMS (FUNCTION EVALUABLE.CONSTANTP)
                                                           X)
                                   then
```

(* This branch currently has a bug in it -- we'd like a version of EVAL which didn't just do an EVALV on litatoms, but first
check CONSTANTEXPRESSIONP on them. The problem occurs in cross-compilation.)

```
                                       (LIST (EVAL X)))
                        elseif (AND [NOT (FMEMB (CAR X)
                                                '(CONS LIST \ALLOCKBLOCK ARRAY MKSTRING MKATOM ALLOCSTRING SYSTEMTYPE
                                                       MACHINETYPE GETD]
                                    (NO.SIDEEFFECTS.FNP (CAR X)))
                            then
```

(* If a random function without side-effects, then it is constant when applied to constant args, except for consers of various
kinds.)

```
                                 (PROG [(VALS (for Z in (CDR X) collect (CAR (OR (EVALUABLE.CONSTANTP Z)
                                                                                 (RETURN]
                                       (RETURN (if VALS
                                                   then (LIST (APPLY (CAR X)
                                                                     VALS]
                        elseif (AND (LISTP (CAR X))
                                    (EQ (CAAR X)
                                        'LAMBDA))
                            then (if (NLISTP (CADAR X))
                                     then                               (* Arglist is NIL or some non-list.)
                                         [EVALUABLE.CONSTANTP (CONS 'PROGN (APPEND (CDR X)
                                                                                   (CDDAR X]
                                   else (for Z VALS in (CDR X) do       (* Be sure that any "arguments" are all constant.
                                                                        Then do "beta" reduction.)
                                            [push VALS (KWOTE (CAR (OR (EVALUABLE.CONSTANTP Z)
                                                                       (RETURN]
                                         finally (RETURN (EVALUABLE.CONSTANTP (CODE.SUBPAIR (CADAR X)
                                                                                           VALS
                                                                                           (CONS 'PROGN (CDDAR X]
```

(**EVALUABLE.CONSTANT.FIXP**
```
   [LAMBDA (X)                                                           (* JonL "25-FEB-83 20:36")
      (FIXP (CAR (EVALUABLE.CONSTANTP X])
```

)

```
(DECLARE%: EVAL@COMPILE

(PUTPROPS EVALUABLE.CONSTANT.FIXP MACRO [(X)
                                         (FIXP (CAR (EVALUABLE.CONSTANTP X])

(PUTPROPS CARCDR.FNP MACRO ((X)
                            (GETPROP X 'CROPS)))
)

(DEFINEQ
```

(**DECL.COMNT.PROCESS**
```
   [LAMBDA (FORMS)                                                       (* JonL "17-OCT-83 22:01")
```

(* Returns a list whose first element is the list of all declarations preceeding significand, whose second element is the list of
all comments preceeding significand, and whose remaining elements are the "body" of FORMS)

```
      (for L DECLS COMNTS Y on FORMS while [AND (LISTP (SETQ Y (CAR L)))
                                               (OR (EQ COMMENTFLG (SETQ Y (CAR Y)))
                                                   (EQ Y 'DECLARE]
         do (if (EQ COMMENTFLG Y)
                then (push COMNTS (CAR L))
              elseif (EQ Y 'DECLARE)
                then (push DECLS (CAR L)))
         finally (RETURN (CONS DECLS (CONS COMNTS L)))
```

)

(DEFINEQ

(\WALKOVER.SPECIALFORMS
```
  [LAMBDA (PRED FORM REST1 REST2 REST3)                                    (* JonL "29-JAN-83 21:30")


          (* Loser! What I really need is a &REST argument L, and use (APPLY PRED <specific-item> L) instead of the APPLY*)

     (SELECTQ (CAR (LISTP FORM))
         (COND [EVERY (CDR FORM)
                       (FUNCTION (LAMBDA (CLZ)
                                    (OR (NLISTP CLZ)
                                        (\WALKOVER.SF.LIST PRED CLZ REST1 REST2 REST3])
          ((SELECTQ SELECTC)
              (AND (APPLY* PRED (CADR FORM)
                           REST1 REST2 REST3)
                   (APPLY* PRED (CAR (LAST FORM))
                           REST1 REST2 REST3)
                   (for LL on (CDDR FORM) until (NULL (CDR LL)) do (OR (\WALKOVER.SF.LIST PRED (CDAR LL)
                                                                                          REST1 REST2 REST3)
                                                                       (RETURN))
                     finally (RETURN T))))
          ((AND OR FRPTQ SETQ)
               (\WALKOVER.SF.LIST PRED (CDR FORM)
                       REST1 REST2 REST3))
          ((APPLY APPLY*)
               (AND (\WALKOVER.FUNCTION PRED (CADR FORM)
                       REST1 REST2 REST3)
                    (\WALKOVER.SF.LIST PRED (CDDR FORM)
                       REST1 REST2 REST3)))
          ((MAP MAPLIST MAPC MAPCAR MAPCON MAPCONC MAPHASH EVERY SOME NOTEVERY NOTANY)
               (AND (APPLY* PRED (CADR FORM)
                       REST1 REST2 REST3)
                    (CAR (SETQ FORM (CDDR FORM)))
                    (\WALKOVER.FUNCTION PRED (CAR FORM)
                       REST1 REST2 REST3)
                    (OR (NLISTP (CDR FORM))
                        (\WALKOVER.FUNCTION PRED (CADR FORM)
                            REST1 REST2 REST3))))
          ((MAPATOMS)
               (\WALKOVER.FUNCTION PRED (CADR FORM)
                   REST1 REST2 REST3))
          ((PROG)                                                   (* FooBar! Note that we can't currently walk over a PROG --
                                                                       30 JAN 1983)
               [AND [EVERY (CADR FORM)
                           (FUNCTION (LAMBDA (L)
                                         (OR (NLISTP L)
                                             (NLISTP (CDR L))
                                             (APPLY* PRED (CADR L)
                                                     REST1 REST2 REST3]
                    (EVERY (CDDR FORM)
                           (FUNCTION (LAMBDA (L)
                                         (OR (NLISTP L)
                                             (APPLY* PRED L REST1 REST2 REST3])
          (SHOULDNT])
```

(\WALKOVER.SF.LIST
```
  [LAMBDA (PRED L REST1 REST2 REST3)                                        (* JonL "21-NOV-82 15:04")
    (EVERY L (FUNCTION (LAMBDA (X)
                          (APPLY* PRED X REST1 REST2 REST3])
```

(\WALKOVER.FUNCTION
```
  [LAMBDA (PRED FN REST1 REST2 REST3)                                       (* JonL "21-NOV-82 15:11")
                                                                            (* Analyze case where FN is being applied
                                                                            (e.g. as in MAPCAR))

     (if [OR (NLISTP FN)
             (NOT (MEMB (CAR FN)
                        ''FUNCTION]
         then (AND (APPLY* PRED FN REST1 REST2 REST3)
                   (APPLY* PRED '(\TypicalUnknownFunction)
                           REST1 REST2 REST3))
        else (APPLY* PRED (if (NLISTP (SETQ FN (CADR FN)))
                              then (LIST FN)
                              else FN)
                  REST1 REST2 REST3])
```

)

(DECLARE%: DONTCOPY

(DECLARE%: EVAL@COMPILE

(RPAQQ **\QUOTIFYING.NLS** (QUOTE FUNCTION DECLARE CONSTANT DEFERREDCONSTANT))

(RPAQQ \WALKABLE.SPECIALFORMS (COND SELECTQ SELECTC AND OR SETQ FRPTQ APPLY APPLY* MAP MAPLIST MAPC MAPCAR
                                    MAPCON MAPCONC MAPHASH MAPATOMS EVERY SOME NOTEVERY NOTANY))

(CONSTANTS \QUOTIFYING.NLS \WALKABLE.SPECIALFORMS)
)

(DECLARE%: EVAL@COMPILE

(PUTPROPS \WALKABLE.SPECIALFORMP MACRO ((FORM)
                                        (MEMB FORM \WALKABLE.SPECIALFORMS)))
)
)

(ADDTOVAR CONSTANTFOLDFNS IMIN IMAX IABS LOGOR LOGXOR LOGAND)

(RPAQQ NOSIDEFNS (fetch CONS NLISTP PROGN APPEND LIST NEQ MEMB MEMBER FMEMB ASSOC TAILP COPY create ELT ELTD
                        AND OR ADD1 SUB1 IPLUS IDIFFERENCE EQ EQUAL NOT NULL))

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS CLISPARRAY CONSTANTFOLDFNS)
)

(PUTPROPS MACROAUX FILETYPE COMPILE-FILE)

(PUTPROPS MACROAUX COPYRIGHT ("Venue & Xerox Corporation" 1983 1984 1985 1986 1990))

FUNCTION INDEX

MACRO INDEX

VARIABLE INDEX

CONSTANT INDEX

PROPERTY INDEX

OPTIMIZER INDEX