File created: 9-Jun-2021 19:40:59 **{DSK}<Users>kaplan>Local>medley3.5>git-medley>sources>ABASIC.;2**

changes to: (FNS NILL ZERO)

previous date: 16-May-90 11:43:16 {DSK}<Users>kaplan>Local>medley3.5>git-medley>sources>ABASIC.;1

Read Table: INTERLISP

Package: INTERLISP

Format: XCCS

```
(RPAQQ ABASICCOMS
       [(FNS EVALQT \SystemERROR)
        (FNS NILL EVQ TRUE ZERO CL:IDENTITY DUMMYDEF NOTIMP)
        (P (DUMMYDEF (WINDOWWORLDP NILL)))
        (FNS EQUAL NEQ NULL NOT)
        (COMS                                              ; Belong on ACODE except they would clobber 10-versions in
                                                           ; ABC
              (FNS LAPRD DEFC CGETD))
        (FNS NCONC \NCONC2 SORT MERGE SORT1 FASSOC FLAST FLENGTH FMEMB FNTH LIST LIST* COUNT)
        (FNS CHANGENAME1 CHANGENAME1A)
        (FNS CDDR CDAR CADR CAAR CDDDR CDDAR CDADR CDAAR CADDR CADAR CAADR CAAAR CDDDDR CAAAAR CDDDAR CDDADR
             CDDAAR CDAAAR CADADR CDADDR CDADAR CAADDR CDAADR CAADAR CADDDR CADAAR CADDAR CAAADR)
        (FNS SYSTEMTYPE)
        (COMS                                              ; Because can't have bignums in code at makeinit time
              (VARS (\IMAX.FLOAT (FIX MAX.FLOAT))
                    (\IMIN.FLOAT (FIX MIN.FLOAT)))
              (GLOBALVARS \IMAX.FLOAT \IMIN.FLOAT))
        (DECLARE%: DONTEVAL@LOAD DOEVAL@COMPILE DONTCOPY COMPILERVARS
               (ADDVARS (NLAMA DUMMYDEF)
                      (NLAML)
                      (LAMA LIST* LIST NCONC NOTIMP ZERO NILL])

(DEFINEQ

(EVALQT
  [LAMBDA NIL                                              (* lmm "11-MAR-80 15:57")
    (PROG (EVALQTXX)
      EVALQTLP
          (PRIN1 '_ T)
          (PRINT [COND
                    ((NULL (SETQ EVALQTXX (READ T)))
                     (GO EVALQTLP))
                    ((OR (LISTP EVALQTXX)
                         (NULL (READP T)))
                     (EVAL EVALQTXX))
                    (T (APPLY EVALQTXX (READ T]
                 T)
          (GO EVALQTLP])

(\SystemERROR
  [LAMBDA NIL                                              (* lmm " 6-MAY-80 20:41")
    (SELECTQ (GETTOPVAL '\SystemErrorVAL)
        (0 (RAID))
        (1 (LISPERROR "FILE SYSTEM RESOURCES EXCEEDED"))
        (RAID])
)

(DEFINEQ

(NILL
  [LAMBDA NIL                                              ; Edited  9-Jun-2021 19:40 by rmk:
    NIL])

(EVQ
  [LAMBDA (X)                                              (* lmm%: 26 JUN 75 726)
    X])

(TRUE
  [LAMBDA NOBIND                                           (* lmm " 4-OCT-83 03:05")
    T])

(ZERO
  [LAMBDA NIL                                              ; Edited  9-Jun-2021 19:40 by rmk:
    0])
```

```
(CL:IDENTITY
  [LAMBDA (X)                                                      (* bvm%: " 3-Nov-86 22:37")
    X])

(DUMMYDEF
  [NLAMBDA LST                                                     (* lmm " 4-OCT-83 03:08")
    (for X in LST when (NOT (GETD (CAR X))) do (PUTD (CAR X)
                                                     (GETD (CADR X))
                                                     T])

(NOTIMP
  [LAMBDA N                                                        (* lmm " 5-MAR-80 20:17")
    (HELP "UNIMPLEMENTED FUNCTION"])
)

(DUMMYDEF (WINDOWWORLDP NILL))

(DEFINEQ

(EQUAL
  [LAMBDA (X Y)                                                    (* bvm%: " 7-Jul-86 22:41")
                                                                   ; tortured logic to optimize performance
    (OR (EQ X Y)
        (COND
           [(LISTP X)
            (AND (LISTP Y)
                 (PROG NIL
                   RETRYLISTP
                       (RETURN (AND (EQUAL (CAR X)
                                           (CAR Y))
                                    (COND
                                       [(LISTP (SETQ X (CDR X)))
                                        (RETURN (AND (LISTP (SETQ Y (CDR Y)))
                                                     (OR (EQ X Y)
                                                         (GO RETRYLISTP]
                                       (T (EQUAL X (CDR Y]
           ((OR (LITATOM X)
                (LITATOM Y))                                       ; can’t be EQUAL and not EQ
            NIL)
           [(NUMBERP X)
            (AND (NUMBERP Y)
                 (SELECTC (NTYPX X)
                      (\FLOATP [SELECTC (NTYPX Y)
                                   (\FLOATP (FEQP X Y))
                                   (\SMALLP (FEQP X Y))
                                   (\FIXP (FEQP X Y))
                                   (PROGN                          ; If comparing against bignum, avoid floating overflow when
                                                                   ; coercing a too-large bignum to floatp
                                          (AND (NOT (GREATERP Y \IMAX.FLOAT))
                                               (NOT (LESSP Y \IMIN.FLOAT))
                                               (FEQP X Y])
                      (\SMALLP (SELECTC (NTYPX Y)
                                   (\FLOATP (FEQP X Y))
                                   (\SMALLP                        ; both small, not EQ
                                             NIL)
                                   (\FIXP                          ; should allow small in fixp boxes?
                                          [AND (EQ (ffetch (FIXP LONUM) of Y)
                                                   (\LOLOC X))
                                               (EQ (ffetch (FIXP HINUM) of Y)
                                                   (COND
                                                      ((ILESSP X 0)
                                                       MAX.SMALLP)
                                                      (T 0])
                                   NIL))
                      (\FIXP (SELECTC (NTYPX Y)
                                 (\FLOATP (FEQP X Y))
                                 (\SMALLP                          ; should allow small in fixp boxes?
                                          [AND (EQ (ffetch (FIXP LONUM) of X)
                                                   (\LOLOC Y))
                                               (EQ (ffetch (FIXP HINUM) of X)
                                                   (COND
                                                      ((ILESSP Y 0)
                                                       MAX.SMALLP)
                                                      (T 0])
                                 (\FIXP (AND (EQ (ffetch (FIXP HINUM) of X)
                                                 (ffetch (FIXP HINUM) of Y))
                                             (EQ (ffetch (FIXP LONUM) of X)
                                                 (ffetch (FIXP LONUM) of Y))))
                                 (PROGN                            ; fixp bignum
                                        NIL)))
                      (SELECTC (NTYPX Y)
                          (\FLOATP                                 ; BIGNUM \FLOATP
                                   (AND (NOT (GREATERP X \IMAX.FLOAT))
                                        (NOT (LESSP X \IMIN.FLOAT))
                                   (FEQP X Y)))
                        (\SMALLP                                    ; BIGNUM \SMALLP
                             NIL)
                        (\FIXP                                      ; BIGNUM \FIXP
                           NIL)
                        (PROGN                                      ; BIGNUM BIGNUM
                               (EQ (\BIGNUM.COMPARE X Y)
                                   0]
          ((NUMBERP Y)
           NIL)
          ((STRINGP X)
           (STREQUAL X Y))
          (T (\EXTENDED.EQP X Y])
```

(**NEQ**
```
  [LAMBDA (X Y)
    (NOT (EQ X Y])
```

(**NULL**
```
  [LAMBDA (X)
    (EQ X NIL])
```

(**NOT**
```
  [LAMBDA (X)
    (EQ X NIL])
)
```

;; Belong on ACODE except they would clobber 10-versions in ABC

```
(DEFINEQ
```

(**LAPRD**
```
  [LAMBDA (FN)                                                      (* wsh%: "20-JUL-79 12:27")
    (PROG (X Y)
          (RETURN (COND
                     ([OR (NEQ (PEEKC)
                               '% )
                          [NOT (LITATOM (SETQ X (SETQ Y (READ]
                          (NOT (LISTP (SETQ X (GETP X 'CODEREADER]
                      (ERROR '"Bad compiled function" FN))
                     (T (APPLY* (CAR X)
                               FN])
```

(**DEFC**
```
  [LAMBDA (NM DF)                                                   ; Edited 10-Nov-87 13:01 by jds

    ;; Put in a new definition (DF) for the name NM.

    (PROG ((PROP 'CODE))
          (COND
             ((OR (NULL DFNFLG)
                  (EQ DFNFLG T))

              ;; OK to redefine it.

              [COND
                 ((GETD NM)
                  (VIRGINFN NM T)
                  (COND
                     ((NULL DFNFLG)
                      (PRINT (CONCAT NM " redefined")
                             T T)                                   ; NOTE: this call to PRINT is changed to LISPXPRINT later in
                                                                    ; the loadup.
                      (SAVEDEF NM]
              (PUTD NM DF T)                                        ; NOTE: this call to PUTD is changed to /PUTD later in the
                                                                    ; loadup.

              )
             (T ;; DFNFLG is PROP, so save the definition.

                (PUTPROP NM PROP DF)                                ; NOTE: this call to PUTPROP is changed to /PUTPROP later in
                                                                    ; the loadup.

                ))
          (RETURN DF])
```

(**CGETD**
```
  [LAMBDA (X)
    (COND
       ((LITATOM X)
        (GETD X))
       (T X])
```

```
)

(DEFINEQ

(NCONC
  [LAMBDA N
    (AND (NEQ N 0)
         (PROG ((L (ARG N N))
                (J N)
                X)
           LP  (COND
                  ((EQ (SETQ J (SUB1 J))
                       0)
                   (RETURN L))
                  ((LISTP (SETQ X (ARG N J)))
                   (FRPLACD (LAST X)
                          L)
                   (SETQ L X)))
               (GO LP])

(\NCONC2
  [LAMBDA (X Y)                                                   (* lmm "15-APR-82 22:10")
    (COND
       ((LISTP X)
        (RPLACD (LAST X)
               Y)
        X)
       (T Y])

(SORT
  [LAMBDA (DATA COMPAREFN)
    (DECLARE (LOCALVARS . T))                                     (* lmm%: "11-NOV-76 23:48:23")
    (COND
       [(NLISTP DATA)
        (COND
           (DATA (ERROR '"DATA NOT LIST:" DATA]
       (T (OR COMPAREFN (SETQ COMPAREFN (FUNCTION ALPHORDER)))
          (FRPLACD (LAST DATA)
                 NIL)
          (SORT1 DATA NIL COMPAREFN])

(MERGE
  [LAMBDA (A B COMPAREFN)                                         (* lmm " 6-MAY-80 21:36")
    (PROG (ATAIL BTAIL)
          [COND
             ((NULL B)                                            ; MERGE will work if either arg is NIL.
              (RETURN A))
             ((NULL A)
              (RETURN B))
             ((NLISTP B)                                          ; No possible meaning here;  user must be in error.
              (ERRORX (LIST 4 B)))
             ((NLISTP A)
              (ERRORX (LIST 4 A)))
             ([NOT (SELECTQ COMPAREFN
                       (T (ALPHORDER (CAAR A)
                                 (CAAR B)))
                       (NIL (ALPHORDER (CAR A)
                                   (CAR B)))
                       (APPLY* COMPAREFN (CAR A)
                              (CAR B]

              ;; (CAR A) must be before (CAR B) at LOOP (see comment below).  If not, swap A and B.  --- The SELECTQ compares the next
              ;; things on A and B.

              (SETQ A (PROG1 B (SETQ B A]
          (SETQ ATAIL A)

      ;; It is desireable to make the value of the merged list available to the user not only as the return from MERGE, but also on both the CONSES
      ;; given as arguments.  To this end, the MERGE is actually performed on the lists A and (CONS (CAR B) (CDR B)), so that when we return, the
      ;; original B may be smashed with (CAR A) and (CDR A).

          (SETQ BTAIL (CONS (CAR B)
                            (CDR B)))

      ;; Whenever we pass LOOP, we know that ATAIL is LISTP, BTAIL is LISTP, and (CAR ATAIL) belongs before (CAR BTAIL).  We therefore look to
      ;; see if there is anything more on ATAIL;  if not, tie on BTAIL and return.  Otherwise, compare (CADR ATAIL) to (CAR BTAIL).  If ATAIL wins, just
      ;; take one CDR and go around.  But if BTAIL wins, then we swap variable/structures: ATAIL is rplacd'd to the structure that was on BTAIL, and
      ;; BTAIL is bound to the old CDR of ATAIL.  We then take the CDR and go around.  Observe that this swapping preserves the assumptions made
      ;; at LOOP.

      LOOP
          [COND
             [(NLISTP (CDR ATAIL))
              (FRPLACD ATAIL BTAIL)
              (RETURN (FRPLACA (FRPLACD B (CDR A))
                             (CAR A]
             [(SELECTQ COMPAREFN
                  (NIL (ALPHORDER (CADR ATAIL)
                                  (CAR BTAIL)))
                  (T (ALPHORDER (CAADR ATAIL)
                                (CAAR BTAIL)))
                  (APPLY* COMPAREFN (CADR ATAIL)
                         (CAR BTAIL]
             (T (FRPLACD ATAIL (PROG1 BTAIL
                                      (SETQ BTAIL (CDR ATAIL)))]
          (SETQ ATAIL (CDR ATAIL))
          (GO LOOP])
```

(**SORT1**

```
  [LAMBDA (DATA END COMPAREFN)
    (DECLARE (LOCALVARS . T))                                         (* lmm%: "11-NOV-76 23:49:27")
    (COND
       ((OR (EQ DATA END)
            (EQ (CDR DATA)
                END))
        DATA)
       (T (PROG ((L DATA)
                 (A DATA)
                 TM)                                                  ; Split DATA by setting A to one cell before its midpoint.  DATA
                                                                      ; remains EQ to the original list.
            LP  (COND
                   ((AND (NEQ (SETQ L (CDR L))
                              END)
                         (NEQ (SETQ L (CDR L))
                              END))
                    (SETQ A (CDR A))
                    (GO LP)))
                (SETQ TM (SORT1 DATA (CDR A)
                                COMPAREFN))
                (SORT1 (SETQ L (CDR A))
                       END COMPAREFN)

          ;; Merge DATA thru A with L (= (CDR A)) up to END.  This is a little tricky because DATA must remain EQ to its original value.

            ALP (COND
                   ((EQ TM L)                                         ; Exhausted first list.
                    (RETURN DATA)))
            BLP (COND
                   ((SELECTQ COMPAREFN
                        (T [ALPHORDER (COND
                                         ((LISTP (CAR TM))
                                          (CAAR TM))
                                         (T (CAR TM)))
                                  (COND
                                     ((LISTP (CAR L))
                                      (CAAR L))
                                     (T (CAR L])
                        (APPLY* COMPAREFN (CAR TM)
                               (CAR L)))
                    (SETQ TM (CDR TM))
                    (GO ALP)))

          ;; Move first element of second list (L = (CDR A)) to before first element of first list (TM).  This must be done by exchanging the CARs and
          ;; then patching up the CDRs, to retain the EQ property.  This is a 'critical section' in that data will be lost if a hard interrupt occurs, but it
          ;; cannot be interrupted by ^H because it does no function calls.

          ;; NOT TRUE IN INTERLISP-D--THIS SECTION IS AN UNPROTECTED CRITICAL REGION.

                [COND
                   [(EQ TM A)                                         ; Special case.
                    [FRPLACA TM (PROG1 (CAR L)
                                       (FRPLACA L (CAR TM)))]
                    (SETQ L (CDR (SETQ TM (SETQ A L]
                   (T [FRPLACD A (PROG1 (CDR L)
                                        (FRPLACA TM (PROG1 (CAR L)
                                                           (FRPLNODE2 L TM)
                                                           (FRPLACD TM L))))]
                      (SETQ TM L)
                      (SETQ L (CDR A]
                (COND
                   ((NEQ L END)
                    (GO BLP)))                                        ; Exhausted second list.
                (RETURN DATA])
```

(**FASSOC**

```
  [LAMBDA (KEY ALST)                                                  (* lmm " 5-MAR-80 20:55")
    (COND
       ((NULL ALST)
        NIL)
       ((EQ KEY (CAAR ALST))
        (CAR ALST))
       (T (FASSOC KEY (CDR ALST])
```

(**FLAST**

```
  [LAMBDA (X)                                                       (* lmm " 5-MAR-80 20:57")
    (PROG ((Y X))
          (GO LP0)
      LP  (SETQ X Y)
          (SETQ Y (CDR X))
      LP0 (COND
             (Y (GO LP)))
          (RETURN X])
```

(**FLENGTH**

```
  [LAMBDA (X)                                                       (* lmm%: "11-NOV-76 23:53:54")
    (PROG ((N 0)
           Y)
          (SETQ Y X)
          (GO LP0)
      LP  (SETQ X Y)
          (SETQ N (ADD1 N))
          (SETQ Y (CDR X))
      LP0 (COND
             (Y (GO LP)))
          (RETURN N])
```

(**FMEMB**

```
  [LAMBDA (X Y)                                                     (* lmm "27-MAR-80 12:53")
    (PROG NIL
      LP  (RETURN (COND
                     ((NULL Y)
                      NIL)
                     ((EQ (CAR Y)
                          X)
                      Y)
                     (T (SETQ Y (CDR Y))
                        (GO LP])
```

(**FNTH**

```
  [LAMBDA (X N)                                                     (* lmm " 6-MAY-80 22:29")
    (COND
       ((IGREATERP 1 N)
        (CONS NIL X))
       (T (PROG ((X0 X))
                (DECLARE (LOCALVARS X0))
            LP  (COND
                   ((NULL X)
                    (RETURN NIL))
                   ((NOT (IGREATERP N 1))
                    (RETURN X)))
                (SETQ N (SUB1 N))
                (SETQ X (CDR X))
                (GO LP])
```

(**LIST**

```
  [LAMBDA N                                                         (* JonL "24-Apr-84 14:49")
    (PROG ((J N)
           L)
      LP  (COND
             ((EQ 0 J)
              (RETURN L)))
          (SETQ L (CONS (ARG N J)
                        L))
          (SETQ J (SUB1 J))
          (GO LP])
```

(**LIST***

```
  [LAMBDA NARGS                                                     (* JonL "27-Sep-84 21:19")
    (COND
       ((EQ 0 NARGS)
        NIL)
       ((EQ 1 NARGS)
        (ARG NARGS 1))
       (T (bind (VAL _ (ARG NARGS NARGS)) for I from (SUB1 NARGS) by -1 until (ILEQ I 0)
             do (push VAL (ARG NARGS I)) finally (RETURN VAL])
```

(**COUNT**

```
  [LAMBDA (X)                                                       (* lmm%: 24 JUN 75 32)
    (PROG ((N 0))
      LP  (COND
             ((NLISTP X)
              (RETURN N))
             (T (SETQ N (IPLUS (COUNT (CAR X))
                               1 N))
                (SETQ X (CDR X))
                (GO LP])
)

(DEFINEQ

(CHANGENAME1
  [LAMBDA (DEF X Y)                                    (* rmk%: "15-APR-80 17:29")
                                                       ; This isn't on ACODE because it would smash the 10 version in
                                                       ; ABC

    (COND
       ((EXPRP DEF)
        (NLSETQ (ESUBST Y X DEF)))
       ((CCODEP DEF)
        [COND
           ((LITATOM DEF)
            (SETQ DEF (GETD DEF]
        (CHANGENAME1A DEF X Y (CHANGECCODE X X DEF])

(CHANGENAME1A
  [LAMBDA (DEF OLD NEW MAP)                            (* lmm "20-MAY-80 09:43")
    (COND
       ((AND MAP (find X in (CDR MAP) suchthat (find Y in (CDR X) suchthat Y)))
        (CHANGECCODE NEW MAP DEF)
        (UNDOSAVE (LIST 'CHANGENAME1A DEF NEW OLD MAP))
        T])
)

(DEFINEQ

(CDDR
  [LAMBDA (X)
    (CDR (CDR X])

(CDAR
  [LAMBDA (X)
    (CDR (CAR X])

(CADR
  [LAMBDA (X)
    (CAR (CDR X])

(CAAR
  [LAMBDA (X)
    (CAR (CAR X])

(CDDDR
  [LAMBDA (X)
    (CDR (CDDR X])

(CDDAR
  [LAMBDA (X)
    (CDR (CDAR X])

(CDADR
  [LAMBDA (X)
    (CDR (CADR X])

(CDAAR
  [LAMBDA (X)
    (CDR (CAAR X])

(CADDR
  [LAMBDA (X)
    (CAR (CDDR X])

(CADAR
  [LAMBDA (X)
    (CAR (CDAR X])

(CAADR
  [LAMBDA (X)
    (CAR (CADR X])
(CAAAR
  [LAMBDA (X)
    (CAR (CAAR X])

(CDDDDR
  [LAMBDA (X)
    (CDDR (CDDR X])

(CAAAAR
  [LAMBDA (X)
    (CAAR (CAAR X])

(CDDDAR
  [LAMBDA (X)
    (CDDR (CDAR X])

(CDDADR
  [LAMBDA (X)
    (CDDR (CADR X])

(CDDAAR
  [LAMBDA (X)
    (CDDR (CAAR X])

(CDAAAR
  [LAMBDA (X)
    (CDAR (CAAR X])

(CADADR
  [LAMBDA (X)
    (CADR (CADR X])

(CDADDR
  [LAMBDA (X)
    (CDAR (CDDR X])

(CDADAR
  [LAMBDA (X)
    (CDAR (CDAR X])

(CAADDR
  [LAMBDA (X)
    (CAAR (CDDR X])

(CDAADR
  [LAMBDA (X)
    (CDAR (CADR X])

(CAADAR
  [LAMBDA (X)
    (CAAR (CDAR X])

(CADDDR
  [LAMBDA (X)
    (CADR (CDDR X])

(CADAAR
  [LAMBDA (X)
    (CADR (CAAR X])

(CADDAR
  [LAMBDA (X)
    (CADR (CDAR X])

(CAAADR
  [LAMBDA (X)
    (CAAR (CADR X])
)

(DEFINEQ
(SYSTEMTYPE
  [LAMBDA NIL                                                                    (* lmm "17-AUG-81 15:50")
                                                                                 ; let the macro decide
    (SYSTEMTYPE])
)

;; Because can't have bignums in code at makeinit time

(RPAQ \IMAX.FLOAT (FIX MAX.FLOAT))

(RPAQ \IMIN.FLOAT (FIX MIN.FLOAT))

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS \IMAX.FLOAT \IMIN.FLOAT)
)

(DECLARE%: DONTEVAL@LOAD DOEVAL@COMPILE DONTCOPY COMPILERVARS

(ADDTOVAR NLAMA DUMMYDEF)

(ADDTOVAR NLAML )

(ADDTOVAR LAMA LIST* LIST NCONC NOTIMP ZERO NILL)
)

(RPAQQ ABASICCOMS
       [(FNS EVALQT \SystemERROR)
        (FNS NILL EVQ TRUE ZERO CL:IDENTITY DUMMYDEF NOTIMP)
        (P (DUMMYDEF (WINDOWWORLDP NILL)))
        (FNS EQUAL NEQ NULL NOT)
        (COMS                                                                    ; Belong on ACODE except they would clobber 10-versions in
                                                                                 ; ABC
              (FNS LAPRD DEFC CGETD))
        (FNS NCONC \NCONC2 SORT MERGE SORT1 FASSOC FLAST FLENGTH FMEMB FNTH LIST LIST* COUNT)
        (FNS CHANGENAME1 CHANGENAME1A)
        (FNS CDDR CDAR CADR CAAR CDDDR CDDAR CDADR CDAAR CADDR CADAR CAADR CAAAR CDDDDR CAAAAR CDDDAR CDDADR
             CDDAAR CDAAAR CADADR CDADDR CDADAR CAADDR CDAADR CAADAR CADDDR CADAAR CADDAR CAAADR)
        (FNS SYSTEMTYPE)
        (COMS                                                                    ; Because can't have bignums in code at makeinit time
              (VARS (\IMAX.FLOAT (FIX MAX.FLOAT))
                    (\IMIN.FLOAT (FIX MIN.FLOAT)))
              (GLOBALVARS \IMAX.FLOAT \IMIN.FLOAT))
        (DECLARE%: DONTEVAL@LOAD DOEVAL@COMPILE DONTCOPY COMPILERVARS (ADDVARS (NLAMA DUMMYDEF)
                                                                                (NLAML)
                                                                                (LAMA LIST* LIST NCONC NOTIMP])

(DECLARE%: DONTEVAL@LOAD DOEVAL@COMPILE DONTCOPY COMPILERVARS

(ADDTOVAR NLAMA DUMMYDEF)

(ADDTOVAR NLAML )

(ADDTOVAR LAMA LIST* LIST NCONC NOTIMP)
)

(PUTPROPS ABASIC COPYRIGHT ("Venue & Xerox Corporation" 1982 1983 1984 1985 1986 1987 1990 2021))
```

FUNCTION INDEX

VARIABLE INDEX