File created: 28-Jan-93 17:42:57 **{DSK}<python>lde>lispcore>sources>LLNEW.;2**

changes to: (FNS \COPY)

previous date: 5-Jan-93 00:46:10 {DSK}<python>lde>lispcore>sources>LLNEW.;1

Read Table: INTERLISP

Package: INTERLISP

Format: XCCS

;;


```
(RPAQQ LLNEWCOMS
       ((PROPS (LLNEW FILETYPE))
        [COMS                                                    ; low level memory access
              (FNS \ADDBASE \GETBASE \PUTBASE \PUTBASE.UFN \PUTBASEPTR.UFN \PUTBITS.UFN \GETBASEBYTE
                   \PUTBASEBYTE \GETBASEPTR \PUTBASEPTR \HILOC \LOLOC \VAG2 \RPLPTR \RPLPTR.UFN)
              (FNS EQ EQL)
              (PROP BYTEMACRO EQL)
              (FNS LOC VAG)
              (FNS CREATEPAGES \NEW4PAGE)
              (DECLARE%: DONTCOPY (EXPORT (RECORDS POINTER WORD)
                                          (MACROS PTRGTP .COERCE.TO.SMALLPOSP. .COERCE.TO.BYTE.))
                     (ADDVARS (INEWCOMS (FNS \GETBASEBYTE \PUTBASEBYTE CREATEPAGES \NEW4PAGE))
                            (RDCOMS (FNS \CAR.UFN \CDR.UFN)
                                   (FNS \COPY \UNCOPY)
                                   (FNS \GETBASEBYTE \PUTBASEBYTE))
                            (INITPTRS (\LISTPDTD))
                            (MKI.SUBFNS (\ADDBASE . I.ADDBASE)
                                   (\GETBASE . I.GETBASE)
                                   (\PUTBASE . I.PUTBASE)
                                   (\GETBASEPTR . I.GETBASEPTR)
                                   (\PUTBASEPTR . I.PUTBASEPTR)
                                   (\HILOC . I.HILOC)
                                   (\LOLOC . I.LOLOC)
                                   (\VAG2 . I.VAG2)
                                   (.COERCE.TO.SMALLPOSP. . PROG1)
                                   (.COERCE.TO.BYTE. . PROG1)
                                   (LOCKEDPAGEP . MKI.LOCKEDPAGEP)
                                   (\RPLPTR . I.PUTBASEPTR)
                                   (CONS . I.\CONS.UFN))
                            (RD.SUBFNS (\ADDBASE . VADDBASE)
                                   (\GETBASE . VGETBASE)
                                   (\PUTBASE . VPUTBASE)
                                   (\GETBASEPTR . VGETBASEPTR)
                                   (\PUTBASEPTR . VPUTBASEPTR)
                                   (\HILOC . VHILOC)
                                   (\LOLOC . VLOLOC)
                                   (\VAG2 . VVAG2)
                                   (.COERCE.TO.SMALLPOSP. . PROG1)
                                   (.COERCE.TO.BYTE. . PROG1)
                                   (PTRGTP . IGREATERP)
                                   (\RPLPTR . VPUTBASEPTR)
                                   (CAR . V\CAR.UFN)
                                   (CDR . V\CDR.UFN)
                                   (CAR/CDRERR . T)))
                     EVAL@COMPILE
                     (ADDVARS (DONTCOMPILEFNS CREATEPAGES]
        [COMS                                                    ; cons cells
              (FNS CONS \CONS.UFN \MAIKO.CONS.UFN CAR \CAR.UFN CDR \CDR.UFN RPLACA \RPLACA.UFN RPLACD
                   \RPLACD.UFN DOCOLLECT \RPLCONS ENDCOLLECT \INITCONSPAGE \NEXTCONSPAGE)
              (ADDVARS (\MAIKO.MOVDS (\MAIKO.CONS.UFN \CONS.UFN)))
              (FNS \RESTLIST.UFN \FINDKEY.UFN)
              (INITVARS (CAR/CDRERR 'CDR))
              (DECLARE%: DONTCOPY (GLOBALVARS CAR/CDRERR)
                     (EXPORT (RECORDS LISTP CONSPAGE)
                            (CONSTANTS * CONSCONSTANTS))
                     (MACROS .MAKECONSCELL. .FINDCLOSEPRIOR. .FINDCDRABLEPAIR. .FINDPAIR.)
                                                             ; for MAKEINIT
                     (ADDVARS (INEWCOMS (FNS \CONS.UFN \MAIKO.CONS.UFN \INITCONSPAGE \NEXTCONSPAGE))
                            (EXPANDMACROFNS .MAKECONSCELL. .FINDCLOSEPRIOR. .FINDCDRABLEPAIR. .FINDPAIR.)))
        (COMS                                                ; testing out CONSes
              (FNS CHECKCONSPAGES \CHECKCONSPAGE)
              (DECLARE%: DONTCOPY (MACROS !CHECK]
        [COMS                                                ; other random stuff for makeinit
              (FNS MAKEINITFIRST MAKEINITLAST \COPY \UNCOPY)
              (DECLARE%: DONTCOPY (EXPORT (MACROS LOCAL ALLOCAL))
                     (ADDVARS (MKI.SUBFNS (CHECK . *)
                                   (RAID . HELP)
                                   (UNINTERRUPTABLY . PROGN)
                                   (\StatsAdd1 . *)
                                   (EVQ . I.\COPY)
                                   (COPY . I.\COPY))
```

```
                            (RD.SUBFNS (CHECK . *)
                                   (RAID . HELP)
                                   (UNINTERRUPTABLY . PROGN)
                                   (\StatsAdd1 . *)
                                   (EVQ . V\COPY)
                                   (COPY . V\COPY)
                                   (1ST . V\UNCOPY)))
                        (ADDVARS (INEWCOMS (FNS MAKEINITFIRST \COPY MAKEINITLAST)))
                        EVAL@COMPILE
                        (ADDVARS (DONTCOMPILEFNS MAKEINITFIRST \COPY MAKEINITLAST \UNCOPY]
        (LOCALVARS . T)))

(PUTPROPS LLNEW FILETYPE :BCOMPL)
```

;; low level memory access

```
(DEFINEQ
```

(\ADDBASE

```
  [LAMBDA (X D)                                                    (* lmm " 2-NOV-81 18:33")
```

;; usually done in microcode; this version uses only arithmetic and \VAG2

```
    (PROG (NH NL (XH (\HILOC X))
                 (XL (\LOLOC X)))
          (.UNBOX. D NH NL)
          (COND
             [(IGREATERP XL (IDIFFERENCE MAX.SMALL.INTEGER NL))  ; carry
              (add XH 1)
              (SETQ XL (SUB1 (IDIFFERENCE XL (IDIFFERENCE MAX.SMALL.INTEGER NL]
             (T (add XL NL)))
          (COND
             [(IGREATERP NH MAX.POS.HINUM)
              (SETQ XH (SUB1 (IDIFFERENCE XH (IDIFFERENCE MAX.SMALL.INTEGER NH]
             (T (add XH NH)))
          (RETURN (\VAG2 XH XL])
```

(\GETBASE

```
  [LAMBDA (X D)                                                    (* lmm " 2-NOV-81 18:33")
```

;; usually done in microcode; case where D=0 MUST be done in microcode

```
    (\GETBASE (\ADDBASE X D)
          0])
```

(\PUTBASE

```
  [LAMBDA (X D V)                                                  (* lmm "11-FEB-83 07:35")
```

;; usually done in microcode; case where D=0 MUST be handled there

```
    (\PUTBASE (\ADDBASE X D)
          0
          (.COERCE.TO.SMALLPOSP. V])
```

(\PUTBASE.UFN

```
  [LAMBDA (X V D)                                                  (* lmm "11-FEB-83 07:35")
```

;; usually done in microcode; case where D=0 MUST be handled there

```
    (\PUTBASE (\ADDBASE X D)
          0
          (.COERCE.TO.SMALLPOSP. V])
```

(\PUTBASEPTR.UFN

```
  [LAMBDA (X V D)                                                  (* lmm "10-NOV-81 15:12")
```

;; usually done in microcode; this def uses only PUTBASE, ADDBASE, etc

```
    (\PUTBASE X D (\HILOC V))
    (\PUTBASE (\ADDBASE X D)
          1
          (\LOLOC V))
    V])
```

(\PUTBITS.UFN

```
  [LAMBDA (X V N.FD)                                               (* lmm "11-FEB-83 07:35")
    (PROG ((NV (.COERCE.TO.SMALLPOSP. V))
           (WIDTH (ADD1 (LOGAND N.FD 15)))
           (FIRST (LRSH (LOGAND N.FD 255)
                        4))
           MASK SHIFT)
          (SETQ SHIFT (IDIFFERENCE 16 (IPLUS FIRST WIDTH)))
          (SETQ MASK (SUB1 (LLSH 1 WIDTH)))
          (\PUTBASE (SETQ X (\ADDBASE X (LRSH N.FD 8)))
                0
                (LOGOR (LOGAND (\GETBASE X 0)
                               (LOGXOR 65535 (LLSH MASK SHIFT)))
```

```
                        (LLSH (LOGAND NV MASK)
                              SHIFT)))
          (RETURN NV])
```

(\GETBASEBYTE

```
[LAMBDA (PTR N)                                                    (* bvm%: " 5-Feb-85 12:05")
  ;; usually done in microcode;  this def.  uses only \GETBASE and arithmetic --- used by MAKEINIT too
  (COND
     [(EVENP N)
      (fetch (WORD HIBYTE) of (\GETBASE PTR (FOLDLO N BYTESPERWORD]
     (T (fetch (WORD LOBYTE) of (\GETBASE PTR (FOLDLO N BYTESPERWORD])
```

(\PUTBASEBYTE

```
[LAMBDA (PTR DISP BYTE)                                            (* JonL "31-Dec-83 23:48")
  ;; usually done in microcode --- this def used by MAKEINIT too
  (SETQ BYTE (.COERCE.TO.BYTE. BYTE))
  [\PUTBASE PTR (FOLDLO (SETQ DISP (\DTEST DISP 'SMALLP))
                        BYTESPERWORD)
         (COND
            ((EVENP DISP BYTESPERWORD)
             (create WORD using (\GETBASE PTR (FOLDLO DISP BYTESPERWORD))
                                HIBYTE _ BYTE))
            (T (create WORD using (\GETBASE PTR (FOLDLO DISP BYTESPERWORD))
                                  LOBYTE _ BYTE]
  BYTE])
```

(\GETBASEPTR

```
[LAMBDA (X D)                                                      (* lmm " 2-NOV-81 18:34")
  ;; usually done in microcode;  this def.  uses GETBASE, VAG2, etc.  and handles overflows too
  (\VAG2 (fetch LOBYTE of (\GETBASE X D))
         (\GETBASE (\ADDBASE X 1)
                D])
```

(\PUTBASEPTR

```
[LAMBDA (X D V)                                                    (* lmm " 2-NOV-81 18:35")
  ;; usually done in microcode;  this def uses only PUTBASE, ADDBASE, etc
  (\PUTBASE X D (\HILOC V))
  (\PUTBASE (\ADDBASE X D)
         1
         (\LOLOC V))
  V])
```

(\HILOC

```
[LAMBDA (X)                                                        (* lmm "10-MAR-81 15:02")
                                                                   ; MUST be handled in microcode
  (\HILOC X])
```

(\LOLOC

```
[LAMBDA (X)                                                        (* lmm "10-MAR-81 15:03")
                                                                   ; MUST be handled in microcode
  (\LOLOC X])
```

(\VAG2

```
[LAMBDA (H L)                                                      (* JonL "31-Dec-83 23:39")
  ;; case where H is byte and L is smallposp MUST be handled in microcode.  Other cases may run error here.
  (\VAG2 (.COERCE.TO.BYTE. H)
         (.COERCE.TO.SMALLPOSP. L])
```

(\RPLPTR

```
[LAMBDA (OBJ OFFSET VAL)                                           (* lmm " 3-NOV-81 12:10")
  (UNINTERRUPTABLY
      (\ADDREF VAL)
      (\DELREF (\GETBASEPTR (SETQ OBJ (\ADDBASE OBJ OFFSET))
                       0))
      (\PUTBASEBYTE OBJ 1 (\HILOC VAL))                            ; \PUTBASEPTR smashes the high byte
      (\PUTBASE OBJ 1 (\LOLOC VAL))
      VAL)])
```

(\RPLPTR.UFN

```
[LAMBDA (OBJ VAL OFFSET)                                           ; Edited 14-Jan-87 16:34 by Pavel
```

;;; The UFN is different from the function since the offset (inline) gets pushed last.

```
    (LET ((SLOT (\ADDBASE OBJ OFFSET)))
         (UNINTERRUPTABLY
             ;; Fix up the reference counts.
             (\ADDREF VAL)
             (\DELREF (\GETBASEPTR SLOT 0))
             ;; \PUTBASEPTR smashes the high byte, so we use two calls instead.
             (\PUTBASEBYTE SLOT 1 (\HILOC VAL))
             (\PUTBASE SLOT 1 (\LOLOC VAL))
             ;; Be sure to return the OBJ; code generated by the new compiler counts on it.
             OBJ)])

)

(DEFINEQ

(EQ
  [LAMBDA (X Y)                                          (* lmm "10-MAR-81 15:04")
                                                         ; MUST be handled in microcode
    (EQ X Y])

(EQL
  [LAMBDA (X Y)                                          ; Edited  6-Jul-87 09:40 by jop

;;; Like EQ except for numbers

    (COND
       ((OR (NOT (CL:NUMBERP X))
            (TYPEP X 'CL:FIXNUM))
        (EQ X Y))
       [(CL:FLOATP X)
        ;; 32 bit compare --- differs from feqp in that the predicate is not true for -0.0 and 0.0
        (AND (CL:FLOATP Y)
             (EQ (fetch LOWORD of X)
                 (fetch LOWORD of Y))
             (EQ (fetch HIWORD of X)
                 (fetch HIWORD of Y]
       ((CL:INTEGERP X)
        (AND (CL:INTEGERP Y)
             (IEQP X Y)))
       [(TYPEP X 'RATIO)
        (AND (TYPEP Y 'RATIO)
             (EQL (CL::RATIO-NUMERATOR X)
                  (CL::RATIO-NUMERATOR Y))
             (EQL (CL::RATIO-DENOMINATOR X)
                  (CL::RATIO-DENOMINATOR Y]
       ((TYPEP X 'COMPLEX)
        (AND (TYPEP Y 'COMPLEX)
             (EQL (CL::COMPLEX-REALPART X)
                  (CL::COMPLEX-REALPART Y))
             (EQL (CL::COMPLEX-IMAGPART X)
                  (CL::COMPLEX-IMAGPART Y])

)

(PUTPROPS EQL BYTEMACRO COMP.EQ)

(DEFINEQ

(LOC
  [LAMBDA (X)                                            (* lmm " 2-NOV-81 18:29")
                                                         ; Return HILOC-LOLOC pair, for easier traffic with RAID.  VAG
                                                         ; interprets such pairs correctly.
    (CONS (\HILOC X)
          (\LOLOC X])

(VAG
  [LAMBDA (LOC)                                          (* lmm " 2-NOV-81 18:28")
                                                         ; LOC can be a HILOC-LOLOC pair
    (COND
       [(LISTP LOC)
        (\VAG2 (CAR LOC)
               (OR (FIXP (CDR LOC))
                   (FIX (CADR LOC]
       (T (\VAG2 (\HINUM LOC)
                 (\LONUM LOC])

)

(DEFINEQ

(CREATEPAGES
```

```
  [LAMBDA (VA N BLANKFLG LOCKFLG)                                    (* bvm%: "29-MAR-83 16:35")
    ;; called only under MAKEINIT --- BLANKFLG means that MAKEINIT won't write on this page, so fake it --- to prevent storage overflow when
    ;; running on Maxc and init'ing GC table
    (for I from 0 to (SUB1 N) do (\NEWPAGE (\ADDBASE VA (UNFOLD I WORDSPERPAGE))
                                           NIL LOCKFLG BLANKFLG))
    VA])
```

(\NEW4PAGE

```
  [LAMBDA (PTR)                                                      ; Edited 24-Oct-92 12:45 by sybalsky:mv:envos
    ;; Instantiates a block of 4 new virtual pages, starting with the one at PTR.
    (\NEWPAGE (\ADDBASE (\NEWPAGE (\ADDBASE (\NEWPAGE (\ADDBASE (\NEWPAGE PTR)
                                                              WORDSPERPAGE))
                                            WORDSPERPAGE))
                        WORDSPERPAGE])

)

(DECLARE%: DONTCOPY

;; FOLLOWING DEFINITIONS EXPORTED

(DECLARE%: EVAL@COMPILE

[ACCESSFNS POINTER [(PAGE# (IPLUS (LLSH (\HILOC DATUM)
                                        8)
                                  (LRSH (\LOLOC DATUM)
                                        8)))
                   (WORDINPAGE (LOGAND (\LOLOC DATUM)
                                       255))
                   (CELLINPAGE (LRSH (fetch WORDINPAGE of DATUM)
                                     1))
                   (BYTEINPAGE (LLSH (fetch WORDINPAGE of DATUM)
                                     1))
                   (SEGMENT# (\HILOC DATUM))
                   (WORDINSEGMENT (\LOLOC DATUM))
                   (CELLINSEGMENT (LRSH (fetch WORDINSEGMENT of DATUM)
                                        1))
                   (WORD# (fetch WORDINPAGE of DATUM))
                   (DBLWORD# (fetch CELLINPAGE of DATUM))
                   (PAGEBASE (\VAG2 (\HILOC DATUM)
                                    (LOGAND (\LOLOC DATUM)
                                            65280]
       (CREATE (\VAG2 (LRSH PAGE# 8)
                      (LLSH (LOGAND PAGE# 255)
                            8]

(ACCESSFNS WORD ((HIBYTE (LRSH DATUM 8))
                 (LOBYTE (LOGAND DATUM 255)))
       (CREATE (IPLUS (LLSH HIBYTE 8)
                      LOBYTE)))
)

(DECLARE%: EVAL@COMPILE

(PUTPROPS PTRGTP MACRO [OPENLAMBDA (X Y)
                         (OR (IGREATERP (\HILOC X)
                                        (\HILOC Y))
                             (AND (EQ (\HILOC X)
                                      (\HILOC Y))
                                  (IGREATERP (\LOLOC X)
                                             (\LOLOC Y]))

(PUTPROPS .COERCE.TO.SMALLPOSP. DMACRO [OPENLAMBDA (X)
                                        (COND
                                           ((SMALLPOSP X)
                                            X)
                                           (T (\ILLEGAL.ARG X]))

(PUTPROPS .COERCE.TO.BYTE. DMACRO [OPENLAMBDA (X)
                                   (COND
                                      ([AND (SMALLPOSP X)
                                            (ILESSP X (CONSTANT (LLSH 1 BITSPERBYTE]
                                       X)
                                      (T (\ILLEGAL.ARG X])
)

;; END EXPORTED DEFINITIONS

(ADDTOVAR INEWCOMS (FNS \GETBASEBYTE \PUTBASEBYTE CREATEPAGES \NEW4PAGE))

(ADDTOVAR RDCOMS (FNS \CAR.UFN \CDR.UFN)
                 (FNS \COPY \UNCOPY)
                 (FNS \GETBASEBYTE \PUTBASEBYTE))
```

```
(ADDTOVAR INITPTRS (\LISTPDTD))

(ADDTOVAR MKI.SUBFNS (\ADDBASE . I.ADDBASE)
                     (\GETBASE . I.GETBASE)
                     (\PUTBASE . I.PUTBASE)
                     (\GETBASEPTR . I.GETBASEPTR)
                     (\PUTBASEPTR . I.PUTBASEPTR)
                     (\HILOC . I.HILOC)
                     (\LOLOC . I.LOLOC)
                     (\VAG2 . I.VAG2)
                     (.COERCE.TO.SMALLPOSP. . PROG1)
                     (.COERCE.TO.BYTE. . PROG1)
                     (LOCKEDPAGEP . MKI.LOCKEDPAGEP)
                     (\RPLPTR . I.PUTBASEPTR)
                     (CONS . I.\CONS.UFN))

(ADDTOVAR RD.SUBFNS (\ADDBASE . VADDBASE)
                    (\GETBASE . VGETBASE)
                    (\PUTBASE . VPUTBASE)
                    (\GETBASEPTR . VGETBASEPTR)
                    (\PUTBASEPTR . VPUTBASEPTR)
                    (\HILOC . VHILOC)
                    (\LOLOC . VLOLOC)
                    (\VAG2 . VVAG2)
                    (.COERCE.TO.SMALLPOSP. . PROG1)
                    (.COERCE.TO.BYTE. . PROG1)
                    (PTRGTP . IGREATERP)
                    (\RPLPTR . VPUTBASEPTR)
                    (CAR . V\CAR.UFN)
                    (CDR . V\CDR.UFN)
                    (CAR/CDRERR . T))


(ADDTOVAR DONTCOMPILEFNS CREATEPAGES)
)

;; cons cells

(DEFINEQ

(CONS
  [LAMBDA (X Y)                                                    (* lmm "11-FEB-82 13:55")
                                                                   ; use microcode UFN to get to \CONS.UFN

    ((OPCODES CONS)
     X Y])


(\CONS.UFN
  [LAMBDA (X Y)                                                    ; Edited  8-Dec-92 16:46 by jds
    [COND
       ((ZEROP CDRCODING)
        (RAID)
        (PROG ((CELL (CREATECELL \LISTP)))
              (replace (LISTP CAR) of CELL with X)
              (replace (LISTP CDR) of CELL with Y)
              (RETURN CELL]
    (UNINTERRUPTABLY
        (\ADDREF X)
        (\ADDREF Y)
        (\StatsAdd1 (fetch DTDCNTLOC of \LISTPDTD))
        (.INCREMENT.ALLOCATION.COUNT. 1)
        (PROG (CNS.PAGE CELL)
              [SETQ CNS.PAGE (COND
                                 ((NOT Y)
                                  [COND
                                     ((AND (SETQ CNS.PAGE (CREATE POINTER
                                                                  PAGE# _ (FETCH DTDNEXTPAGE OF \LISTPDTD)))
                                           (IGREATERP (FETCH (CONSPAGE CNT) OF CNS.PAGE)
                                                  0)))
                                     (T (SETQ CNS.PAGE (\NEXTCONSPAGE]
                                  (.MAKECONSCELL. CNS.PAGE X \CDR.NIL))
                                 ((AND (EQ (NTYPX Y)
                                           \LISTP)
                                       (IGREATERP (fetch (CONSPAGE CNT) of (SETQ CNS.PAGE (fetch (POINTER PAGEBASE)
                                                                                                 of Y)))
                                             0)
                                       (SETQ CELL (.FINDCLOSEPRIOR. CNS.PAGE X Y)))
                                  ;; Test for any cells left on page --- NTYPX rather than LISTP test for benefit of MAKEINIT
                                                                   (* .MAKECONSCELL. CNS.PAGE X
                                                                   (IPLUS \CDR.ONPAGE (fetch (POINTER DBLWORD#) of Y)))
                                  CELL)
                                 (T (.FINDPAIR. X Y]
              (\DELREF CNS.PAGE)
              (RETURN CNS.PAGE)))])
```

(\MAIKO.CONS.UFN

```
[LAMBDA (X Y)                                                    ; Edited  3-Jun-90 21:03 by nm
```

;; Maiko specific \CONS.UFN. Does not decrement \RECLAIM.COUNTDOWN.

```
[COND
   ((ZEROP CDRCODING)
    (RAID)
    (PROG ((CELL (CREATECELL \LISTP)))
          (replace (LISTP CAR) of CELL with X)
          (replace (LISTP CDR) of CELL with Y)
          (RETURN CELL]
(UNINTERRUPTABLY
    (\ADDREF X)
    (\ADDREF Y)
    (\StatsAdd1 (fetch DTDCNTLOC of \LISTPDTD))
    (.CHECK.ALLOCATION.COUNT. 1)
    (PROG (CNS.PAGE)
          [SETQ CNS.PAGE (COND
                            [(AND (EQ (NTYPX Y)
                                      \LISTP)
                                  (IGREATERP (fetch (CONSPAGE CNT) of (SETQ CNS.PAGE (fetch (POINTER PAGEBASE)
                                                                                        of Y)))
                                             0))                 ; Test for any cells left on page --- NTYPX rather than LISTP test
                                                                 ; for benefit of MAKEINIT
                             (.MAKECONSCELL. CNS.PAGE X (IPLUS \CDR.ONPAGE (fetch (POINTER DBLWORD#)
                                                                              of Y]
                            (T (.MAKECONSCELL. (SETQ CNS.PAGE (\NEXTCONSPAGE))
                                               X
                                               (COND
                                                  ((NULL Y)
                                                   \CDR.NIL)
                                                  (T (IPLUS \CDR.INDIRECT (fetch (POINTER DBLWORD#)
                                                                             of (.MAKECONSCELL. CNS.PAGE Y 0]
          (\DELREF CNS.PAGE)
          (RETURN CNS.PAGE)))])
```

(CAR

```
[LAMBDA (X)                                                      (* lmm "11-FEB-82 13:56")
   ((OPCODES CAR)
    X])
```

(\CAR.UFN

```
[LAMBDA (X)                                                      (* lmm "18-Jul-84 00:07")
```

;; most cases handled in microcode --- this code also used by MAKEINIT/READSYS

```
(\CALLME 'CAR)
(COND
   [(LISTP X)
    (COND
       ((ZEROP CDRCODING)
        (fetch (LISTP CAR) of X))
       (T (COND
             ((EQ (fetch CDRCODE of X)
                  \CDR.INDIRECT)
              (fetch CARFIELD of (fetch CARFIELD of X)))
             (T (fetch CARFIELD of X]
   ((NULL X)
    NIL)
   (T (SELECTQ CAR/CDRERR
           (T (LISPERROR "ARG NOT LIST" X))
           ((NIL CDR)
                (COND
                   ((EQ X T)
                    T)
                   ((LITATOM X)
                    NIL)
                   (T '"{car of non-list}")))
           (COND
              ((EQ X T)
               T)
              ((STRINGP X)
               (LISPERROR "ARG NOT LIST" X))
              (T '"{car of non-list}"])
```

(CDR

```
[LAMBDA (X)                                                      (* lmm "11-FEB-82 13:56")
   ((OPCODES CDR)
    X])
```

(\CDR.UFN

```
[LAMBDA (X)                                                      (* lmm "17-Jul-84 22:26")
```

;; most cases handled in microcode --- this code also used by MAKEINIT/READSYS

```
    (\CALLME 'CDR)
    (COND
       [(LISTP X)
        (COND
           ((ZEROP CDRCODING)
            (fetch (LISTP CDR) of X))
           (T (PROG ((Q (fetch CDRCODE of X)))
                    (RETURN (COND
                                ((EQ Q \CDR.NIL)
                                 NIL)
                                ((IGREATERP Q \CDR.ONPAGE)
                                 (\ADDBASE (fetch (POINTER PAGEBASE) of X)
                                        (LLSH (IDIFFERENCE Q \CDR.ONPAGE)
                                              1)))
                                ((EQ Q \CDR.INDIRECT)
                                 (\CDR.UFN (fetch CARFIELD of X)))
                                (T (fetch CARFIELD of (\ADDBASE (fetch PAGEBASE of X)
                                                                (LLSH Q 1]
       ((NULL X)
        NIL)
       (T (SELECTQ CAR/CDRERR
                ((T CDR)
                     (LISPERROR "ARG NOT LIST" X))
                (NIL (COND
                        ((LITATOM X)
                         (GETPROPLIST X))
                        (T "{cdr of non-list}")))
                (COND
                   ((STRINGP X)
                    (LISPERROR "ARG NOT LIST" X))
                   (T "{cdr of non-list}"])
```

(RPLACA

```
  [LAMBDA (X Y)                                          (* lmm "11-FEB-82 13:55")
                                                         ; invoke \RPLACA.UFN
    ((OPCODES RPLACA)
     X Y])
```

(\RPLACA.UFN

```
  [LAMBDA (X Y)                                          (* lmm " 1-DEC-81 21:17")
    (COND
       [(NLISTP X)
        (COND
           [(NULL X)                                     ; if X is NIL and Y is NIL ok
            (COND
               (Y (LISPERROR "ATTEMPT TO RPLAC NIL" Y]
           (T (LISPERROR "ARG NOT LIST" X]
       (T (COND
             ((ZEROP CDRCODING)
              (replace (LISTP CAR) of X with Y)
              X)
             (T (UNINTERRUPTABLY
                    (\DELREF (CAR X))
                    (\ADDREF Y)
                    (replace CARFIELD of (COND
                                            ((EQ (fetch CDRCODE of X)
                                                 \CDR.INDIRECT)
                                             (fetch CARFIELD of X))
                                            (T X))
                       with Y)
                    X)]))
```

(RPLACD

```
  [LAMBDA (X Y)                                          (* lmm "11-FEB-82 13:55")
    ((OPCODES RPLACD)
     X Y])
```

(\RPLACD.UFN

```
  [LAMBDA (X Y)                                          (* lmm "11-JAN-82 10:15")
    (COND
       [(NLISTP X)
        (COND
           [(NULL X)                                     ; if X is NIL and Y is NIL ok
            (COND
               (Y (LISPERROR "ATTEMPT TO RPLAC NIL" Y]
           (T (LISPERROR "ARG NOT LIST" X]
       ((ZEROP CDRCODING)
        (replace (LISTP CDR) of X with Y)
        X)
       (T (UNINTERRUPTABLY
              (\DELREF (CDR X))
              (\ADDREF Y)
              (PROG (RP.PAGE (RP.Q (fetch CDRCODE of X)))
```

```
                    (COND
                       ((EQ RP.Q \CDR.INDIRECT)
                        (SETQ RP.PAGE (fetch CARFIELD of X))
                        (CHECK (ILEQ (fetch CDRCODE of RP.PAGE)
                                     \CDR.MAXINDIRECT)
                               (NEQ (fetch CDRCODE of RP.PAGE)
                                    \CDR.INDIRECT))
                        (SETQ RP.PAGE (\ADDBASE (fetch PAGEBASE of RP.PAGE)
                                                (LLSH (IDIFFERENCE (fetch CDRCODE of RP.PAGE)
                                                                   \CDR.INDIRECT)
                                                      1)))
                        (CHECK (LISTP RP.PAGE)
                               (EQ 0 (fetch CDRCODE of RP.PAGE)))
                        (replace FULLCARFIELD of RP.PAGE with Y))
                       ((ILEQ RP.Q \CDR.MAXINDIRECT)
                        (SETQ RP.PAGE (\ADDBASE (fetch PAGEBASE of X)
                                                (LLSH (IDIFFERENCE RP.Q \CDR.INDIRECT)
                                                      1)))
                        (CHECK (LISTP RP.PAGE)
                               (EQ 0 (fetch CDRCODE of RP.PAGE)))
                        (replace FULLCARFIELD of RP.PAGE with Y))
                       ((NULL Y)
                        (replace CDRCODE of X with \CDR.NIL))
                       [(EQ (SETQ RP.PAGE (fetch PAGEBASE of X))
                            (fetch PAGEBASE of Y))                    ; New CDR on same page
                        (replace CDRCODE of X with (IPLUS \CDR.ONPAGE (fetch (POINTER DBLWORD#) of Y]
                       [(IGREATERP (fetch (CONSPAGE CNT) of RP.PAGE)
                                   0)                                 ; Room on page for cdr cell
                        (replace CDRCODE of X with (IPLUS \CDR.INDIRECT (fetch (POINTER DBLWORD#)
                                                                                of (.MAKECONSCELL. RP.PAGE Y 0]
                       (T [replace FULLCARFIELD of X with (.MAKECONSCELL. (SETQ RP.PAGE (\NEXTCONSPAGE))
                                                                          (fetch CARFIELD of X)
                                                                          (IPLUS \CDR.INDIRECT (fetch (POINTER DBLWORD#)
                                                                                                      of (.MAKECONSCELL. RP.PAGE
                                                                                                                         Y 0]
                          (replace CDRCODE of X with \CDR.INDIRECT)))
                    (RETURN X)))])
```

(DOCOLLECT

```
[LAMBDA (ITEM LST)                                                  (* lmm%: "30-SEP-76 13:03:33")
  (COND
     ((NLISTP LST)
      (FRPLACD (SETQ LST (LIST ITEM))
             LST))
     (T (CDR (FRPLACD LST (CONS ITEM (CDR LST])
```

(\RPLCONS

```
[LAMBDA (LST ITEM)                                                  (* bvm%: " 5-Feb-85 22:49")
                                                                    (* (CDR (RPLACD LST (CONS ITEM NIL))))
  (COND
     [(AND (NEQ CDRCODING 0)
           (LISTP LST)
           (UNINTERRUPTABLY
               ;; Have to go uninterruptable here so that someone doesn't change the CNT field to zero out from under us
               [PROG ((CPAGE (fetch (POINTER PAGEBASE) of LST))
                      CELL)
                     (RETURN (COND
                                ((AND (NEQ (fetch (CONSPAGE CNT) of CPAGE)
                                           0)
                                      (IGREATERP (fetch CDRCODE of LST)
                                                 \CDR.MAXINDIRECT))
                                 (\ADDREF ITEM)
                                 (\DELREF (CDR LST))
                                 (SETQ CELL (.MAKECONSCELL. CPAGE ITEM \CDR.NIL))
                                 (\StatsAdd1 (fetch DTDCNTLOC of \LISTPDTD))
                                 (.INCREMENT.ALLOCATION.COUNT. 1)
                                 (replace CDRCODE of LST with (IPLUS \CDR.ONPAGE (fetch (POINTER DBLWORD#)
                                                                                        of CELL)))
                                 CELL])]
      (T (SETQ ITEM (CONS ITEM NIL))                                ; Have to be careful how this part is written, or compiler will turn it
                                                                    ; into RPLCONS !
         (RPLACD LST ITEM)
         ITEM])
```

(ENDCOLLECT

```
[LAMBDA (X Y)                                                       (* lmm "21-MAR-81 13:37")
  (COND
     ((NULL X)
      Y)
     (T (PROG1 (CDR X)
               (RPLACD X Y])
```

(\INITCONSPAGE

```
  [LAMBDA (BASE LINK)                                        ; Edited  4-Dec-92 04:13 by jds
    (COND
       ((ZEROP CDRCODING)
        (RAID))
       (T (PROG ((J (replace (CONSPAGE NEXTCELL) of BASE with 254))
                 CELL)
            LP   (COND
                    ((IGREATERP J 4)
                     (SETQ CELL (\ADDBASE BASE J))
                     (replace (LISTP FULLCARFIELD) of CELL with NIL)
                     (replace (LISTP NEXTFREE) of CELL with (SETQ J (IDIFFERENCE J 2)))
                     (GO LP)))
                 (replace (CONSPAGE CNT) of BASE with 126)          ; if LINK=NIL, stores a 0. This assumes that the pagebase of NIL
                                                                    ; is NIL
                 (replace NEXTPAGE of BASE with (fetch (POINTER PAGE#) of LINK))
                 (RETURN BASE])
```

(\NEXTCONSPAGE

```
  [LAMBDA NIL                                                ; Edited  8-Dec-92 01:57 by jds
    (CHECK (NULL \INTERRUPTABLE))
    (PROG ((N (fetch DTDNEXTPAGE of \LISTPDTD))
           PG)
          (SETQ PG (\ALLOCMDSPAGE (fetch DTDTYPEENTRY of \LISTPDTD)))
          (\INITCONSPAGE PG (\INITCONSPAGE (\ADDBASE PG WORDSPERPAGE)
                                           (CREATE POINTER
                                                   PAGE# _ N)))
          (replace DTDNEXTPAGE of \LISTPDTD with (PAGELOC PG))
          (RETURN PG])

)

(ADDTOVAR \MAIKO.MOVDS (\MAIKO.CONS.UFN \CONS.UFN))

(DEFINEQ
```

(\RESTLIST.UFN

```
  [LAMBDA (TAIL LASTN FIRSTN)                                (* bvm%: "31-Aug-86 16:30")
```

;;; Handles &REST args by building a list of the args from FIRSTN thru LASTN, all consed onto the front of TAIL, which could be non-NIL in the case
;;; where the microcode has started the job

```
    (COND
       (TAIL                                                 ; Some already done, better take care of gc
             (\GC.HANDLEOVERFLOW)))
    (LET* [(CALLER (\MYALINK))
           (BLINK (fetch (FX BLINK) of CALLER))
           (IVAR (fetch (BF IVAR) of BLINK))
           (BASE (STACKADDBASE (IDIFFERENCE IVAR WORDSPERCELL]
          (for I from LASTN to FIRSTN by -1 do (SETQ TAIL (CONS (\GETBASEPTR BASE (UNFOLD I WORDSPERCELL))
                                                                TAIL))
                                                ;; Might want to experiment with stopping after one iteration to let the microcode do the rest
                                                ;; of the consing

            finally (RETURN TAIL])
```

(\FINDKEY.UFN

```
  [LAMBDA (KEY ARGN)                                         (* bvm%: "15-Jul-86 16:51")
```

;;; Searches argument list of current function for an argument EQ to KEY. Search starts at the argument index given as the alpha byte ARGN and
;;; examines every other argument. The first arg is numbered 1; i.e., arg(i) is located at ivar0 + 2*(i-1). If KEY is found as arg i, returns i+1 (which is later
;;; to be fed to ARG0); otherwise returns NIL.

```
    (LET* [(CALLER (\MYALINK))
           (BLINK (fetch (FX BLINK) of CALLER))
           (IVAR (fetch (BF IVAR) of BLINK))
           (NARGS (SUB1 (FOLDLO (IDIFFERENCE BLINK IVAR)
                                WORDSPERCELL]
          (for I from ARGN to NARGS by 2 as [BASE _ (STACKADDBASE (PLUS IVAR (UNFOLD (SUB1 ARGN)
                                                                                       WORDSPERCELL]
            by (\ADDBASE BASE (TIMES 2 WORDSPERCELL)) when (EQ (\GETBASEPTR BASE 0)
                                                               KEY)
            do (RETURN (ADD1 I])

)

(RPAQ? CAR/CDRERR 'CDR)

(DECLARE%: DONTCOPY

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS CAR/CDRERR)
)
```

```
;; FOLLOWING DEFINITIONS EXPORTED

(DECLARE%: EVAL@COMPILE

(BLOCKRECORD LISTP (;; Describes a CONS cell.
                    (CAR POINTER)
                    (CDR POINTER))
       (CREATE (CREATECELL \LISTP))
       ;; FOLLOWING ARE CDR-CODE FIELDS
       (BLOCKRECORD LISTP ((CDRCODE BITS 4)
                           (CARFIELD XPOINTER)))
       ;; For chaining together free cells on a page:
       (BLOCKRECORD LISTP ((NEXTFREE BYTE)
                           (NIL BITS 24)))
       [ACCESSFNS LISTP ((FULLCARFIELD NIL (\PUTBASEPTR DATUM 0 NEWVALUE]
       ;; because replace of XPOINTER is slow, the CAR field is stored with PUTBASEPTR, even though that smashes the hi byte

       )

(BLOCKRECORD CONSPAGE ((CNT BYTE)
                       (NEXTCELL BYTE)
                       (NIL WORD)
                       (NEXTPAGE FIXP)))
)

(RPAQQ CONSCONSTANTS (\CDR.ONPAGE \CDR.NIL \CDR.INDIRECT \CDR.MAXINDIRECT \CONSPAGE.LAST))

(DECLARE%: EVAL@COMPILE

(RPAQQ \CDR.ONPAGE 8)

(RPAQQ \CDR.NIL 8)

(RPAQQ \CDR.INDIRECT 0)

(RPAQQ \CDR.MAXINDIRECT 7)

(RPAQQ \CONSPAGE.LAST 65535)

(CONSTANTS \CDR.ONPAGE \CDR.NIL \CDR.INDIRECT \CDR.MAXINDIRECT \CONSPAGE.LAST)
)

;; END EXPORTED DEFINITIONS

(DECLARE%: EVAL@COMPILE

(PUTPROPS .MAKECONSCELL. MACRO (OPENLAMBDA (PAGE A D)
                                  (PROG [(.MK.NEWCELL (\ADDBASE PAGE (fetch (CONSPAGE NEXTCELL) of PAGE]
                                        (CHECK (NEQ (fetch (CONSPAGE CNT) of PAGE)
                                                    0)
                                               (EVENP (fetch (CONSPAGE NEXTCELL) of PAGE)))
                                        (replace (CONSPAGE NEXTCELL) of PAGE with (fetch (LISTP NEXTFREE)
                                                                                         of .MK.NEWCELL))
                                        (CHECK (EVENP (fetch (CONSPAGE NEXTCELL) of PAGE)))
                                        (add (fetch (CONSPAGE CNT) of PAGE)
                                             -1)
                                        (replace (LISTP FULLCARFIELD) of .MK.NEWCELL with A)
                                        (replace (LISTP CDRCODE) of .MK.NEWCELL with D)
                                        (RETURN .MK.NEWCELL))))

(PUTPROPS .FINDCLOSEPRIOR. MACRO [OPENLAMBDA (PG A D)
                                    (LET ((CDROFFSET (LOGAND (\LOLOC D)
                                                             255))
                                          (OFFSET (fetch (CONSPAGE NEXTCELL) of PG))
                                          CELL PRIOR)
                                         (WHILE (NEQ OFFSET 0)
                                            DO (COND
                                                   ((AND (ILEQ OFFSET CDROFFSET)
                                                         (IGEQ OFFSET (IDIFFERENCE CDROFFSET 14)))
                                                    ;; There's a cell close enough.  Take it off the chain and return it.
                                                    [COND
                                                       [PRIOR ;; There was a prior entry in the chain; detach this one.
                                                              (REPLACE (LISTP NEXTFREE) OF (\ADDBASE PG PRIOR)
                                                                 WITH (FETCH (LISTP NEXTFREE)
                                                                         OF (SETQ CELL (\ADDBASE PG OFFSET]
                                                       (T ;; No prior entry; set the conspage's NEXTCELL entry.
                                                          (REPLACE (CONSPAGE NEXTCELL) OF PG
                                                             WITH (FETCH (LISTP NEXTFREE)
                                                                     OF (SETQ CELL (\ADDBASE PG OFFSET]
```

```
                                                    (add (fetch (CONSPAGE CNT) of PG)
                                                         -1)
                                                    (replace (LISTP FULLCARFIELD) of CELL with A)
                                                    (replace (LISTP CDRCODE) of CELL
                                                       with (LOGOR \CDR.ONPAGE (LRSH (IDIFFERENCE CDROFFSET
                                                                                              OFFSET)
                                                                                     1)))
                                                    (RETURN CELL)))
                                                (SETQ PRIOR OFFSET)
                                                (SETQ OFFSET (FETCH (LISTP NEXTFREE) OF (\ADDBASE PG OFFSET])

(PUTPROPS .FINDCDRABLEPAIR. MACRO [OPENLAMBDA (PG A D)
                                     (LET ((OFFSET (fetch (CONSPAGE NEXTCELL) of PG))
                                           CELL PRIOR PRIORPRIOR)
                                          (AND (IGEQ (FETCH (CONSPAGE CNT) OF PG)
                                                     2)
                                               (WHILE (NEQ OFFSET 0)
                                                  DO (COND
                                                        ((AND PRIOR (ILEQ OFFSET PRIOR)
                                                              (IGEQ OFFSET (IDIFFERENCE PRIOR 14)))
                                                         ;; There's a cell close enough.  Take it off the chain and return it.
                                                         [COND
                                                            [PRIORPRIOR
                                                                    ;; There was a prior entry in the chain; detach this one.
                                                                    (REPLACE (LISTP NEXTFREE)
                                                                       OF (\ADDBASE PG PRIORPRIOR)
                                                                       WITH (FETCH (LISTP NEXTFREE)
                                                                                   OF (SETQ CELL (\ADDBASE PG
                                                                                                           OFFSET]
                                                            (T ;; No prior entry; set the conspage's NEXTCELL entry.
                                                               (REPLACE (CONSPAGE NEXTCELL) OF PG
                                                                  WITH (FETCH (LISTP NEXTFREE)
                                                                              OF (SETQ CELL (\ADDBASE PG OFFSET]
                                                         (add (fetch (CONSPAGE CNT) of PG)
                                                              -2)
                                                         (\PUTBASEPTR (\ADDBASE PG PRIOR)
                                                                0 D)
                                                         (REPLACE (LISTP FULLCARFIELD) OF CELL WITH A)
                                                         (REPLACE (LISTP CDRCODE) OF CELL
                                                            WITH (LRSH (IDIFFERENCE PRIOR OFFSET)
                                                                       1))
                                                         (RETURN CELL)))
                                                    (SETQ PRIORPRIOR PRIOR)
                                                    (SETQ PRIOR OFFSET)
                                                    (SETQ OFFSET (FETCH (LISTP NEXTFREE)
                                                                    OF (\ADDBASE PG OFFSET])

(PUTPROPS .FINDPAIR. MACRO [OPENLAMBDA (A D)
                              (LET ((PG (fetch DTDNEXTPAGE of \LISTPDTD))
                                    CELL CPG)
                                   [WHILE (IGREATERP PG 0) DO (COND
                                                                 ((SETQ CELL (.FINDCDRABLEPAIR. (SETQ CPG
                                                                                                   (CREATE
                                                                                                         POINTER
                                                                                                         PAGE# _
                                                                                                         PG))
                                                                                                A D))
                                                                  (RETURN CELL))
                                                                 (T (SETQ PG (FETCH (CONSPAGE NEXTPAGE)
                                                                                OF CPG]
                                   (OR CELL (.FINDCDRABLEPAIR. (\NEXTCONSPAGE)
                                                               A D])
)

(ADDTOVAR INEWCOMS (FNS \CONS.UFN \MAIKO.CONS.UFN \INITCONSPAGE \NEXTCONSPAGE))

(ADDTOVAR EXPANDMACROFNS .MAKECONSCELL. .FINDCLOSEPRIOR. .FINDCDRABLEPAIR. .FINDPAIR.)
)

;; testing out CONSes

(DEFINEQ

(CHECKCONSPAGES
  [LAMBDA NIL                                                    (* bvm%: "29-Jan-85 22:51")
    (COND
       ((ZEROP CDRCODING)
        NIL)
       (T [for (CPAGE _ (create POINTER
                                PAGE# _ (fetch DTDNEXTPAGE of \LISTPDTD)))
             do (COND
                   ((NULL CPAGE)                                 ; End of free list
                    (RETURN))
                   ((NEQ (NTYPX CPAGE)
```

```
                        \LISTP)
                    ;; Free list not pointing at a cons page.  Test is not for LISTP because LISTP is formally defined to be false for list page bases
                    (HELP CPAGE))
                   (T (SETQ CPAGE (create POINTER
                                        PAGE# _ (fetch (CONSPAGE NEXTPAGE) of CPAGE]
          (\MAPMDS 'LISTP (FUNCTION \CHECKCONSPAGE])
```

(\CHECKCONSPAGE

```
  [LAMBDA (PN)                                                          (* bvm%: "27-Jan-85 14:52")
                                                                        ; check if page PN is ok
    (PROG ((PTR (create POINTER
                        PAGE# _ PN))
           NXT CNT)
          (SETQ CNT (fetch (CONSPAGE CNT) of PTR))
          (!CHECK (EVENP (SETQ NXT (fetch (CONSPAGE NEXTCELL) of PTR))
                         WORDSPERCELL))
      LP  (COND
             ((IGREATERP CNT 0)
              (!CHECK (AND (NEQ NXT 0)
                           (EVENP (SETQ NXT (fetch (LISTP CDRCODE) of (\ADDBASE PTR NXT)))
                                  WORDSPERCELL)))
              (add CNT -1)
              (GO LP)))
          (!CHECK (EQ NXT 0])

)

(DECLARE%: DONTCOPY

(DECLARE%: EVAL@COMPILE

(PUTPROPS !CHECK MACRO [(X)
                        (OR X (RAID 'X])
)
)

;; other random stuff for makeinit

(DEFINEQ

(MAKEINITFIRST
  [LAMBDA NIL                                                           (* bvm%: "13-Jun-86 15:41")
    (CREATEMDSTYPETABLE)
    (\SETUP.HUNK.TYPENUMBERS)
    (INITDATATYPES)
    (PREINITARRAYS)
    (\TURN.ON.HUNKING)
    (INITATOMS)
    (INITDATATYPENAMES)
    (INITUFNTABLE)
    (INITGC)
    (\NEWPAGE \InterfacePage NIL T])

(MAKEINITLAST
  [LAMBDA (VERSIONS)                                                    (* Pavel "17-Oct-86 12:42")
    (SETUPSTACK T)
    (MAKEINITBFS)
    (PROGN                                                              ; fold in property list and values gathered from boot files
           [SELECTQ (SYSTEMTYPE)
                ((D ALTO)
                     [LOCAL (MAPHASH MKI.PLHA (FUNCTION (LAMBDA (P A)
                                                          (SETPROPLIST A (COPY P]
                     [LOCAL (MAPHASH MKI.TVHA (FUNCTION (LAMBDA (V A)
                                                          (SETTOPVAL A (COPY (LOCAL (CDR V])
                (PROG (AL GAG)
                      ;; the reason this is set up this way is because there is a bug in Interlisp-10 suchthat if a garbage collection happens in the middle
                      ;; of a MAPHASH, some of the values in the hash array may be missed because the garbage collector has moved stuff around
                      ;; and rehashed the data in the array.  Thus we are careful to set things up so that no garbage collection happens
                            [ALLOCAL (PROGN [MINFS (IMAX (MINFS)
                                                         (ITIMES 2 (ARRAYSIZE (CAR MKI.PLHA)))
                                                         (ARRAYSIZE (CAR MKI.TVHA]
                                            (RECLAIM)
                                            (SETQ GAG (GCGAG "[***** GARBAGE COLLECTION - ERROR ******]"))
                                            [MAPHASH MKI.PLHA (FUNCTION (LAMBDA (P A)
                                                                          (push AL (CONS A P]
                                            (SETQ GAG (GCGAG GAG]
                            [LOCAL (MAPC AL (FUNCTION (LAMBDA (X)
                                                        (SETPROPLIST (CAR X)
                                                               (COPY (CDR X]
                            (ALLOCAL (PROGN (SETQ AL)
                                            (RECLAIM)
                                            (SETQ GAG (GCGAG GAG))
                                            [MAPHASH MKI.TVHA (FUNCTION (LAMBDA (V A)
```

```
                                                                    (push AL (RPLACA V A]
                                        (GCGAG GAG)))
                        (LOCAL (MAPC AL (FUNCTION (LAMBDA (X)
                                                  (SETTOPVAL (CAR X)
                                                         (COPY (CDR X]
                                                                    ; set most initial variables

         )
    (PROG ((AFL (FILEARRAYBASE)))                                   ; put output on a double page boundary --- output at least one
                                                                    ; page
          [LOCAL (BOUTZEROS (IDIFFERENCE (TIMES 2 BYTESPERPAGE)
                                         (UNFOLD (IMOD (\LOLOC AFL)
                                                       (TIMES 2 WORDSPERPAGE))
                                                 BYTESPERWORD]
          (SETQ MKI.CODELASTPAGE (PAGELOC (FILEARRAYBASE)))
```

;; now we can update the string/array space freelist to point beyond the code area --- We call POSTINITARRAYS with (a) pointer to word after
;; end of compiled code, (b) page number of beginning of compiled code, and (c) page number after compiled code

```
          (POSTINITARRAYS AFL (IPLUS \FirstArrayPage MKI.CODESTARTOFFSET)
                 MKI.CODELASTPAGE))
    [MAPC (ALLOCAL (APPEND INITVALUES INITPTRS))
          (FUNCTION (LAMBDA (X)                                     ; make sure atoms exist for initial atoms
                    (\ATOMVALINDEX (LOCAL (CAR X]
    [for X in INITVALUES as A in MKI.VALUES do (SETQ A (LOCAL (EVALV A)))
                                               (SETTOPVAL (LOCAL (CAR X))
                                                      (COND
                                                         ([ALLOCAL (OR (EQ A T)
                                                                       (EQ A NIL)
                                                                       (AND (FIXP A)
                                                                            (IGEQ A -65536)
                                                                            (ILEQ A 65535]
                                                          (COPY A))
                                                         (T (SHOULDNT]
    [for X in INITPTRS as A in MKI.PTRS do (SETTOPVAL (LOCAL (CAR X))
                                                      (LOCAL (EVALV A]
    (for X in LOCKEDVARS do
                                ;; If the variable exists, then we lock it.  Otherwise, just print a message and proceed anyway, hoping the fellow
                                ;; knows what he's doing.  We don't want to create a new piece of storage at this point because we've already made
                                ;; a note of where our last allocated page is.
                                (IF (GETHASH X MKI.ATOMARRAY)
                                    THEN (\LOCKVAR X)
                                  ELSE (printout T "***Note: Locked var " X " does not exist, proceeding anyway." T))
        )
    (SETUPPAGEMAP)
    (DUMPINITPAGES (IPLUS \FirstArrayPage MKI.CODESTARTOFFSET)
           MKI.CODELASTPAGE VERSIONS])
```

(\COPY

```
  [LAMBDA (X)                                                       ; Edited 28-Jan-93 17:42 by jds
```

;; Prints X into the MAKEINIT / READSYS system

```
    (SELECTQ (LOCAL (TYPENAME X))
         ((LITATOM NEW-ATOM)
              (UNLESSRDSYS (MKI.ATOM X)
                     (VATOMNUMBER X T)))
         (LISTP (PROG [(R (LOCAL (REVERSE X)))
                       (V (\COPY (LOCAL (CDR (LOCAL (LAST X]
                  LP   (COND
                          ((LOCAL (LISTP R))
                           (SETQ V (CONS (\COPY (LOCAL (CAR R)))
                                         V))
                           (SETQ R (LOCAL (CDR R)))
                           (GO LP)))
                        (RETURN V)))
         ((FIXP SMALLP)
              (PROG (V)
                    [COND
                       [(LOCAL (IGREATERP 0 X))                     ; negative
                        (COND
                           ((LOCAL (IGREATERP X -65537))            ; small neg
                            (RETURN (\ADDBASE \SMALLNEGSPACE (LOCAL (LOGAND X 65535]
                       ((LOCAL (ILESSP X 65536))                    ; small pos
                        (RETURN (\ADDBASE \SMALLPOSPSPACE X]        ; need to create a boxed integer
                    (SETQ V (CREATECELL \FIXP))
                    (\PUTBASE V 0 (LOGOR (COND
                                            ((IGREATERP 0 X)
                                             32768)
                                            (T 0))
                                         (LOGAND (LRSH X 16)
                                                 32767)))
                    (\PUTBASE V 1 (LOGAND X 65535))
                    (RETURN V)))
         (ONED-ARRAY (%%COPY-ONED-ARRAY X))
         (STRINGP                                                   ; For bootstrapping only
                  (%%COPY-STRING-TO-ARRAY X))
         (FLOATP (PROG ((VAL (CREATECELL \FLOATP)))
```

```
                    (SELECTQ (SYSTEMTYPE)
                        ((ALTO D)
                             (\PUTBASE VAL 0 (LOCAL (\GETBASE X 0)))
                             (\PUTBASE VAL 1 (LOCAL (\GETBASE X 1))))
                        (MKI.IEEE X VAL))
                    (RETURN VAL)))
        (CHARACTER (\VAG2 \CHARHI (LOCAL (CL:CHAR-CODE X))))
        (ERROR X "can't be copied to remote file"])
```

(\UNCOPY

```
  [LAMBDA (X CARLVL CDRLVL)                                   ; Edited 18-Mar-87 16:51 by raf
    (SELECTC (NTYPX X)
        (\SMALLP (COND
                    ((EQ (\HILOC X)
                         \SmallPosHi)
                     ;; This test used to be SMALLPOSP until its definition changed to test (IGREATERP X 0), which doesn't work renamed
                     (\LOLOC X))
                    (T (IPLUS (\LOLOC X)
                              -65536))))
        (\FIXP                                                ; INTEGER
               (LOCAL (create FIXP
                              HINUM _ (ffetch (FIXP HINUM) of X)
                              LONUM _ (ffetch (FIXP LONUM) of X))))
        (\FLOATP (LOCAL (create FLOATP
                                HIWORD _ (ffetch (FLOATP HIWORD) of X)
                                LOWORD _ (ffetch (FLOATP LOWORD) of X))))
        (\LITATOM (VATOM (\LOLOC X)))
        (\STRINGP (PROG ((PTR (ffetch (STRINGP BASE) of X))
                         (OFFST (ffetch (STRINGP OFFST) of X))
                         (LENGTH (ffetch (STRINGP LENGTH) of X))
                         (I 1)
                         STR)                                 ; Use ffetch to avoid bogus DTEST's in the renamed version
                        (SETQ STR (LOCAL (ALLOCSTRING LENGTH)))
                        (FRPTQ LENGTH [LOCAL (RPLSTRING STR I (LOCAL (FCHARACTER (\GETBASEBYTE PTR OFFST]
                               (add I 1)
                               (add OFFST 1))
                        (RETURN STR)))
        (\CHARACTERP (LOCAL (\VAG2 \CHARHI (\LOLOC X))))
        (%%ONED-ARRAY (LET ((SIZE (ffetch (ONED-ARRAY TOTAL-SIZE) of X))
                            (BASE (ffetch (ONED-ARRAY BASE) of X))
                            (OFFSET (ffetch (ONED-ARRAY OFFSET) of X))
                            (TYPENUMBER (ffetch (ONED-ARRAY TYPE-NUMBER) of X))
                            NCELLS LOCAL-ARRAY LOCAL-BASE)
                           (if (EQ (%%TYPENUMBER-TO-GC-TYPE TYPENUMBER)
                                   PTRBLOCK.GCT)
                               then (LOCAL (VTYPEDPOINTER (TYPENAME X)
                                                          X))
                               else (SETQ NCELLS (FOLDHI (ITIMES (IPLUS SIZE OFFSET)
                                                                 (%%TYPENUMBER-TO-BITS-PER-ELEMENT TYPENUMBER))
                                                         BITSPERCELL))
                                    (SETQ LOCAL-ARRAY (LOCAL (create ONED-ARRAY)))
                                    (SETQ LOCAL-BASE (LOCAL (\ALLOCBLOCK NCELLS)))
                                    (LOCAL (freplace (ONED-ARRAY BASE) of LOCAL-ARRAY with LOCAL-BASE))
                                    (LOCAL (freplace (ONED-ARRAY STRING-P) of LOCAL-ARRAY with (%%CHAR-TYPE-P
                                                                                                TYPENUMBER)))
                                    (LOCAL (freplace (ONED-ARRAY FILL-POINTER-P) of LOCAL-ARRAY
                                              with (ffetch (ONED-ARRAY FILL-POINTER-P) of X)))
                                    (LOCAL (freplace (ONED-ARRAY TYPE-NUMBER) of LOCAL-ARRAY with TYPENUMBER))
                                    (LOCAL (freplace (ONED-ARRAY FILL-POINTER) of LOCAL-ARRAY
                                              with (ffetch (ONED-ARRAY FILL-POINTER) of X)))
                                    (if (NEQ OFFSET 0)
                                        then (LOCAL (freplace (ONED-ARRAY OFFSET) of LOCAL-ARRAY with OFFSET))
                                             (LOCAL (freplace (ONED-ARRAY DISPLACED-P) of LOCAL-ARRAY with T)))
                                    (LOCAL (freplace (ONED-ARRAY TOTAL-SIZE) of LOCAL-ARRAY with SIZE))
                                    [for I from 0 to (SUB1 (LLSH NCELLS 1))
                                       do (LOCAL (\PUTBASE LOCAL-BASE I (\GETBASE BASE I]
                                    LOCAL-ARRAY)))
        (\LISTP [COND
                   [(LISTP X)
                    (COND
                       ((EQ CDRLVL 0)                         ; Abbreviate
                        '(--))
                       (T (LOCAL (CONS [COND
                                          ([OR (EQ CARLVL 0)
                                               (AND (OR (EQ CARLVL 1)
                                                        (EQ CDRLVL 1))
                                                    (LISTP (CAR X]
                                           '&)
                                          (T (\UNCOPY (CAR X)
                                                      (AND CARLVL (SUB1 CARLVL))
                                                      (AND CDRLVL (SUB1 CDRLVL]
                                       (\UNCOPY (CDR X)
                                              CARLVL
                                              (AND CDRLVL (SUB1 CDRLVL]
                   (T                                         ; Redundant LISTP test in case X is list page header
```

```
                        (ALLOCAL (VTYPEDPOINTER 'LISTP X])
         (0 (LOCAL (VTYPEDPOINTER NIL X)))
         (LOCAL (VTYPEDPOINTER (TYPENAME X)
                       X])

)

(DECLARE%: DONTCOPY

;; FOLLOWING DEFINITIONS EXPORTED

(DECLARE%: EVAL@COMPILE

(PUTPROPS LOCAL MACRO ((X)
                       X))

(PUTPROPS ALLOCAL MACRO ((X)
                         X))
)

;; END EXPORTED DEFINITIONS

(ADDTOVAR MKI.SUBFNS (CHECK . *)
                     (RAID . HELP)
                     (UNINTERRUPTABLY . PROGN)
                     (\StatsAdd1 . *)
                     (EVQ . I.\COPY)
                     (COPY . I.\COPY))

(ADDTOVAR RD.SUBFNS (CHECK . *)
                    (RAID . HELP)
                    (UNINTERRUPTABLY . PROGN)
                    (\StatsAdd1 . *)
                    (EVQ . V\COPY)
                    (COPY . V\COPY)
                    (1ST . V\UNCOPY))

(ADDTOVAR INEWCOMS (FNS MAKEINITFIRST \COPY MAKEINITLAST))


(ADDTOVAR DONTCOMPILEFNS MAKEINITFIRST \COPY MAKEINITLAST \UNCOPY)
)

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(LOCALVARS . T)
)

(PUTPROPS LLNEW COPYRIGHT ("Venue & Xerox Corporation" 1981 1982 1983 1984 1985 1986 1987 1990 1992 1993))
```

FUNCTION INDEX

MACRO INDEX

VARIABLE INDEX

CONSTANT INDEX

RECORD INDEX

PROPERTY INDEX