File created: 18-Oct-93 16:25:22 **{Pele:mv:envos}<LispCore>Sources>CLTL2>LLINTERP.;1**

previous date: 31-Dec-92 01:36:17 {Pele:mv:envos}<LispCore>Sources>LLINTERP.;6

Read Table: INTERLISP

Package: INTERLISP

Format: XCCS

```
(RPAQQ LLINTERPCOMS
       [[E                                                             ; Don't fontify these common functions
           (SETQ FNSLST
                 (LDIFFERENCE FNSLST
                        '(EVALV PROG SET SETQ RETURN GO QUOTE AND OR PROGN COND PROG1 FUNCTION EVAL APPLY]
        (COMS                                                          ; For calling interpreted functions
              (FNS \INTERPRETER \INTERPRETER1 \SETUP-COMPILED-CLOSURE-CALL \STKNAME))
        (COMS                                                          ; EVCALL ufn.
              (FNS \ENVCALL.UFN \SETUP-ENVIRONMENT-CALL))
        (COMS                                                          ; recursive interpreter
              (FNS EVAL \EVAL \EVALFORM \EVALFORMASLAMBDA \EVALOTHER APPLY APPLY* \CHECKAPPLY* \CKAPPLYARGS
                   DEFEVAL)
              (DECLARE%: DONTCOPY (MACROS .APPLY.))
              (COMS                                                    ; Free variable manipulation
                    (FNS EVALV \EVALV1 \EVALVAR BOUNDP SET \SETVAR SETQ \STKSCAN \SETFVARSLOT))
              (COMS                                                    ; PROG and friends
                    (FNS PROG \PROG0 \EVPROG1 RETURN GO EVALA \EVALA ERRORSET SI::ERRORSET-PRINT-FUNCTION))
              (COMS                                                    ; LET and friends -- need these in the init
                    (FNS LET LET* \LET0 \LET*))
              (FNS QUOTE AND OR PROGN COND \EVPROGN PROG1)
              (COMS (VARS (\DEFEVALFNS NIL)
                          (\EVALHOOK))
                    (SPECVARS *EVALHOOK*)
                    (ADDVARS (LAMBDASPLST LAMBDA NLAMBDA CL:LAMBDA OPENLAMBDA))
                    (GLOBALVARS \DEFEVALFNS \EVALHOOK LAMBDASPLST CLISPARRAY)
                    (DECLARE%: DONTEVAL@LOAD DOCOPY (VARS (CLISPARRAY))
                           (P (MOVD? 'SETQ 'SETN NIL T)))
                    (GLOBALVARS CLISPARRAY))
              [COMS                                                    ; Evaluating in different stack environment
                    (FNS ENVEVAL ENVAPPLY FUNCTION \FUNCT1 \MAKEFUNARGFRAME STKEVAL STKAPPLY RETEVAL RETAPPLY)
                    (DECLARE%: DONTEVAL@LOAD DOCOPY                    ; For bootstrapping, IL:FUNCTION is as good as CL:FUNCTION
                           (P (MOVD? 'FUNCTION 'CL:FUNCTION NIL T]
              (COMS                                                    ; Blip and other stack funniness
                    (FNS BLIPVAL SETBLIPVAL BLIPSCAN)
                    (FNS \REALFRAMEP)
                    [INITVARS (OPENFNS '(APPLY* SETQ AND OR COND SELECTQ PROG PROGN PROG1 ARG SETARG ERSETQ
                                               NLSETQ RESETFORM RESETLST RESETVARS RPTQ SAVESETQ SETN UNDONLSETQ
                                               XNLSETQ]
                    (VARS \BLIPNAMES)
                    (GLOBALVARS BRKINFOLST)
                    (GLOBALVARS \BLIPNAMES OPENFNS)))
        (COMS (FNS RAIDCOMMAND RAIDSHOWFRAME RAIDSTACKCMD RAIDROOTFRAME PRINTADDRS PRINTVA READVA READATOM
                   READOCT SHOWSTACKBLOCKS SHOWSTACKBLOCK1 PRINCOPY NOSUCHATOM)
              (FNS BACKTRACE \BACKTRACE \SCANFORNTENTRY \PRINTSTK \PRINTFRAME \PRINTBF)
              (DECLARE%: EVAL@COMPILE DONTCOPY (COMS * RAIDCOMS)))
        (COMS (FNS CCODEP EXPRP SUBRP FNTYP ARGTYPE NARGS ARGLIST \CCODEARGLIST \CCODEIVARSCAN)
              (COMS                                                    ; Translation machinery for new LAMBDA words
                    (PROP VARTYPE LAMBDATRANFNS)
                    (ALISTS (LAMBDATRANFNS)))
              (DECLARE%: DONTCOPY (MACROS \CCODENARGS \CCODEFNTYP \CCODEARGTYPE)))
        (COMS                                                          ; CONSTANTS mechanism
              (FNS CONSTANTS CONSTANTEXPRESSIONP)
              (INITVARS (COMPVARMACROHASH (HASHARRAY 100)))
                                                                       ; We need this initialized for the INIT, so don't put it off. (It used
                                                                       ; to start out NIL and get set later)
              (ADDVARS (CONSTANTFOLDFNS PLUS IPLUS TIMES ITIMES DIFFERENCE IDIFFERENCE QUOTIENT IQUOTIENT IMIN
                              IMAX IABS LLSH LRSH LOGOR LOGXOR LOGAND OR AND))
              (GLOBALVARS COMPVARMACROHASH CONSTANTFOLDFNS))
        (DECLARE%: EVAL@COMPILE DONTCOPY DONTEVAL@LOAD (LOCALVARS . T))
        (SPECVARS *TAIL* *FN* *FORM* *ARGVAL*)
        (DECLARE%: EVAL@COMPILE DONTCOPY (ADDVARS (LAMS FAULTEVAL FAULTAPPLY)))
        (DECLARE%: DONTEVAL@LOAD DOEVAL@COMPILE DONTCOPY COMPILERVARS
               (ADDVARS (NLAMA CONSTANTS PROG1 COND PROGN OR AND QUOTE LET* LET GO PROG SETQ)
                      (NLAML FUNCTION RETURN)
                      (LAMA BOUNDP APPLY* \INTERPRETER])
```

;; For calling interpreted functions

```
(DEFINEQ
(\INTERPRETER
  [LAMBDA N                                                          ; Edited 21-Jan-91 18:02 by jds

    ;; the microcode calls this function instead if it is given an expr or an undefined function to call --- the name of the function/sexpression which is
    ;; supposed to be called is given as an extra argument

    (PROG ((FN (ARG N N))
           (NACTUAL (SUB1 N))
           (NA 0)
           DEF ARGLIST NEXTRA NTSIZE TYPE NNILS ENV)
          (COND
             ((LITATOM FN)
              (CHECK (NOT (fetch (LITATOM CCODEP) of FN)))
              (SETQ DEF (fetch (LITATOM DEFPOINTER) of FN)))
             (T (SETQ DEF FN)))
          [COND
             ((TYPEP DEF 'COMPILED-CLOSURE)
              (RETURN (\MISCAPPLY* (FUNCTION \SETUP-COMPILED-CLOSURE-CALL)
                             DEF)))
             ((TYPEP DEF 'CLOSURE)
              (SETQ ENV (CLOSURE-ENVIRONMENT DEF))
              (SETQ DEF (CLOSURE-FUNCTION DEF]
          (COND
             ((NLISTP DEF)
              (GO ERR)))
          (RETURN (.CALLAFTERPUSHINGNILS.
                   (SELECTQ (CAR DEF)
                       (CL:LAMBDA (RETURN (\INTERPRETER-LAMBDA N DEF ENV FN)))
                       ([LAMBDA NLAMBDA OPENLAMBDA]
                            [SETQ ARGLIST (CAR (OR (LISTP (CDR DEF))
                                                   (GO ERR]
                            (SETQ NNILS (IPLUS (SETQ NEXTRA
                                                (COND
                                                   ((LISTP ARGLIST)
                                                                  ; spread function
                                                    (for X in ARGLIST
                                                       do (COND
                                                             ((OR (NULL (\DTEST X 'LITATOM))
                                                                  (EQ X T))
                                                              (LISPERROR "ATTEMPT TO BIND NIL OR T" X)))
                                                                  ; Process one argument
                                                          (SETQ NA (ADD1 NA)))
                                                    (COND
                                                       ((IGREATERP NA NACTUAL)
                                                        (IDIFFERENCE NA NACTUAL))
                                                       (T 0)))
                                                   ((NULL ARGLIST)
                                                                  ; spread function
                                                    0)
                                                   ((EQ ARGLIST T)
                                                    (LISPERROR "ATTEMPT TO BIND NIL OR T" ARGLIST))
                                                   (T

                                      ;; Nospread--needs to bind exactly one variable, the arg name.  LAMBDA* also needs to set that arg
                                      ;; to the number of actual args, but that can be done by diddling the slot currently occupied by the fn
                                      ;; name.  Never any 'extra' args to worry about

                                                      (\DTEST ARGLIST 'LITATOM)
                                                      (SETQ NA 1)
                                                      0)))
                                               (PROG1 (SETQ NTSIZE (CEIL [ADD1 (UNFOLD NA (CONSTANT (
                                                                                          WORDSPERNAMEENTRY
                                                                                                       ]
                                                                         WORDSPERQUAD))

                                      ;; round number of nametable entries up to next quadword, leaving room for a zero.  add in overhead.
                                      ;; NA is now in units of 'cells' since there two words in a cell.

                                                      )
                                               (FOLDHI (fetch (FNHEADER OVERHEADWORDS) of T)
                                                      WORDSPERCELL)
                                               (SUB1 CELLSPERQUAD))))
                       (FUNARG (GO FUN))
                       (GO ERR))
                   (\INTERPRETER1 ARGLIST NNILS NTSIZE NACTUAL NEXTRA FN DEF)))
      FUN

     ;; FUNARG -- Interlisp "closure" of form (FUNARG fn stackptr).  Subsumed by common lisp lexical closures.

          [RETURN (PROGN (\SMASHLINK NIL (\STACKARGPTR (CADDR DEF)))
                         (SPREADAPPLY (CADR DEF)
                                (for I from 1 to (SUB1 N) collect (ARG N I]
      ERR (RETURN (FAULTAPPLY FN (for I from 1 to NACTUAL collect (ARG N I])

(\INTERPRETER1
  [LAMBDA (ARGLIST NNILS NTSIZE NACTUAL NPVARARGS FN DEF)            ; Edited 27-Jan-91 12:48 by jds
    (PROG ((*TAIL* (CDDR DEF))
           (INTERPFRAME (\MYALINK))
           HEADER NT NILSTART)
          (SETQ HEADER (fetch (FX FNHEADER) of INTERPFRAME))        ; The function header of code for \INTERPRETER

     ;; Build a nametable for INTERPFRAME that identifies the vars in ARGLIST as the NACTUAL IVAR's that were passed to it as arguments plus
     ;; the NPVARARGS extra NIL's that we implement as PVAR's.  We build the nametable out of space that was allocated on the stack by
     ;; \INTERPRETER pushing many NIL's

          (SETQ NT (ADDSTACKBASE (CEIL (+ (SETQ NILSTART (- (fetch (FX NEXTBLOCK) of INTERPFRAME)
                                                            (UNFOLD NNILS WORDSPERCELL)))
                                          (UNFOLD NPVARARGS WORDSPERCELL))
                                       WORDSPERQUAD)))

     ;; Address of our synthesized nametable: NNILS cells back from the end of INTERPFRAME, leaving space for additional 'PVARs' we are using as
     ;; extra NIL args, rounded up to quadword

          (UNINTERRUPTABLY
              [COND
                 ((NOT ARGLIST)                                     ; No args, no nametable

                  )
                 ((LISTP ARGLIST)
                  (for ARG in ARGLIST as ARG# from 0 as NT1 from (fetch (FNHEADER OVERHEADWORDS) of T)
                     by (CONSTANT (WORDSPERNAMEENTRY)) as NT2 from (+ (fetch (FNHEADER OVERHEADWORDS)
                                                                          of T)
                                                                       NTSIZE)
                     by (CONSTANT (WORDSPERNTOFFSETENTRY))
                     do (SETSTKNAME-RAW NT NT1 (\ATOMVALINDEX ARG))
                        [SETSTKNTOFFSET-RAW NT NT2 (COND
                                                      ((< ARG# NACTUAL)
                                                       IVARCODE)
                                                      (T            ; Say it's the nth PVAR, where n is out of the range of the real
                                                                    ; PVARs
                                                         PVARCODE))
                               (COND
                                  ((< ARG# NACTUAL)
                                   ARG#)
                                  (T                                ; Say it's the nth PVAR, where n is out of the range of the real
                                                                    ; PVARs
                                     (IPLUS (FOLDLO (- NILSTART (fetch (FX FIRSTPVAR) of INTERPFRAME))
                                                   WORDSPERCELL)
                                            (- ARG# NACTUAL]        ; (SETSTKNTOFFSET-RAW NT NT2 (COND ((< ARG#
                                                                    ; NACTUAL) (+ IVARCODE ARG#)) (T ; Say it's the nth PVAR,
                                                                    ; where n is out of the range of the real PVARs (+ PVARCODE
                                                                    ; (FOLDLO (- NILSTART (fetch (FX FIRSTPVAR) of
                                                                    ; INTERPFRAME)) WORDSPERCELL) (- ARG# NACTUAL)))))

                        )                                           ; Note: area is initialize to NIL's (zero), so end of nametable
                                                                    ; already has its zeroes

                  )
                 (T                                                 ; Nospread.  Store lone arg in nametable
                    (SETSTKNAME-RAW NT (fetch (FNHEADER OVERHEADWORDS) of T)
                           (\ATOMVALINDEX ARGLIST))
                    (SETSTKNTOFFSET-RAW NT (+ (fetch (FNHEADER OVERHEADWORDS) of T)
                                              NTSIZE)
                           IVARCODE
                           (COND
                              ((EQ (CAR DEF)
                                   'NLAMBDA)                        ; It's the first (and only) arg
                               0)
                              (T                                    ; Use the n+1'st arg, which currently is our framename (FN)
                                 (\PUTBASEPTR \STACKSPACE (+ (fetch (BF IVAR) of (fetch (FX BLINK) of INTERPFRAME)
                                                                 )
                                                             (UNFOLD NACTUAL WORDSPERCELL))
                                        NACTUAL)                    ; set arg's value to be number of real args
                                 NACTUAL]
              ;; now fix up header of NT

              (replace (FNHEADER %#FRAMENAME) of NT with FN)        ; use #FRAMENAME to denote no reference counting
              (replace (FNHEADER NTSIZE) of NT with NTSIZE)
              (replace (FNHEADER NLOCALS) of NT with (fetch (FNHEADER NLOCALS) of HEADER))
                                                                    ; Probably doesn't matter, since there are no FVARS in that
                                                                    ; frame
                                                                    ; Do I need to worry about STK, NA, PV, START, ARGTYPE ?
                                                                    ; --- probably not
              (replace (FX NAMETABLE) of INTERPFRAME with NT))
      EVLP

     ;; Now that we have 'bound' the arguments, just evaluate the forms in the LAMBDA/NLAMBDA as progn

          (COND
             ((CDR *TAIL*)
              (\EVAL (CAR *TAIL*))
              (SETQ *TAIL* (CDR *TAIL*))
              (GO EVLP))
             (T (RETURN (\EVAL (CAR *TAIL*])
```

(\SETUP-COMPILED-CLOSURE-CALL

```
  [LAMBDA (CLOSURE)                                                 (* bvm%: "21-Jul-86 11:12")
```

```
;;; Called in the misc context by \INTERPRETER when the function being called is a closure.  Replace the intepreter frame by the frame that would result
;;; if we had correctly called the closure in microcode.  This is a normal function call of the code body in the closure with the one additional wrinkle that
;;; the CLOSURE object is stored in PVAR1.

    (LET ((INTERPFRAME (fetch (IFPAGE MiscFXP) of \InterfacePage))
          (CODE (fetch (COMPILED-CLOSURE FNHEADER) of CLOSURE))
          NA NACTUALS INTERPBF INTERPIVAR INTERPALINK INTERPCLINK SP NEWFX NPVARS STKEND SLOWP OLDBF ENV)
         (SETQ OLDBF (SETQ INTERPBF (fetch (FX BLINK) of INTERPFRAME)))
         (SETQ INTERPIVAR (fetch (BF IVAR) of INTERPBF))
         (SETQ INTERPALINK (fetch (FX %#ALINK) of INTERPFRAME))   ; Note that this is the 'raw' ALINK -- we never look at it, just
                                                                  ; update it in new FX
         [COND
            ((SETQ SLOWP (fetch (FX SLOWP) of INTERPFRAME))

             ;; Usually false, because \INTERPRETER hasn't had any reason to make itself slow.  But it's not uninterruptable, so arbitrary things
             ;; could happen to it

             (SETQ INTERPCLINK (fetch (FX %#CLINK) of INTERPFRAME]
         (SETQ STKEND (fetch (FX NEXTBLOCK) of INTERPFRAME))
         [COND
            ((fetch (BF PADDING) of INTERPBF)                     ; Forget padding.  I don't think anyone pads anymore, except
                                                                  ; maybe Lisp stack mungers
             (SETQ INTERPBF (IDIFFERENCE INTERPBF WORDSPERCELL]
         (SETQ NACTUALS (FOLDLO (IDIFFERENCE INTERPBF INTERPIVAR)
                                WORDSPERCELL))
         (SETQ NA (fetch (FNHEADER NA) of CODE))
         [COND
            ((OR SLOWP (ILESSP (IDIFFERENCE STKEND INTERPBF)
                               (fetch (FNHEADER STKMIN) of CODE)))

             ;; No space for frame, or interpreter frame is slow, do slow case.  This computation is quite conservative, since we aren't counting the
             ;; args

             (LET ((NEWSTACK (\FREESTACKBLOCK (IPLUS (fetch (FNHEADER STKMIN) of CODE)
                                                     (UNFOLD NACTUALS WORDSPERCELL))
                                    INTERPFRAME)))
                  (SETQ STKEND (IPLUS NEWSTACK (fetch (FSB SIZE) of NEWSTACK)))
                  [while (type? FSB STKEND) do (SETQ STKEND (add STKEND (fetch (FSB SIZE) of STKEND]
                  (\BLT (ADDSTACKBASE NEWSTACK)
                        (ADDSTACKBASE INTERPIVAR)
                        (UNFOLD NACTUALS WORDSPERCELL))
                  (SETQ INTERPBF (IPLUS NEWSTACK (UNFOLD NACTUALS WORDSPERCELL)))
                  [COND
                     ((NEQ (fetch (FX USECNT) of INTERPFRAME)
                           0)
                      (add (fetch (FX USECNT) of INTERPFRAME)
                           -1))
                     (T [COND
                           ((NEQ (fetch (BF USECNT) of OLDBF)
                                 0)
                            (add (fetch (BF USECNT) of OLDBF)
                                 -1))
                           (T                                     ; Normal slow case, can flush BF
                              (\MAKEFREEBLOCK INTERPIVAR (IPLUS (IDIFFERENCE OLDBF INTERPIVAR)
                                                                WORDSPERCELL]
                                                                  ; Finally, flush FX.  Has to be separate free block because FX
                                                                  ; and BF not necessarily contiguous
                        (LET [(START (COND
                                        ((EQ OLDBF (fetch (FX DUMMYBF) of INTERPFRAME))
                                                                  ; Normal contiguous case
                                         INTERPFRAME)
                                        (T                        ; Have to blow away the dummy BF in front of the FX
                                           (fetch (FX DUMMYBF) of INTERPFRAME]
                             (\MAKEFREEBLOCK START (IDIFFERENCE (fetch (FX NEXTBLOCK) of INTERPFRAME)
                                                          START]
                  (SETQ INTERPIVAR NEWSTACK)
                  (SETQ SLOWP T]
         [PROGN                                                   ; Do argument adjustment.  In general we should pop excess
                                                                  ; args, but there's really no need for that
                (COND
                   ((GREATERP NA NACTUALS)                        ; Push extra NILs for missing args
                    (FRPTQ (DIFFERENCE NA NACTUALS)
                           (\PUTBASEPTR (STACKADDBASE INTERPBF)
                                  0 NIL)
                           (add INTERPBF WORDSPERCELL]
         (PROGN                                                   ; Fix up BF trailer cell
                (\PUTBASE \STACKSPACE INTERPBF 0)                 ; Clear BF flags
                (replace (BF IVAR) of INTERPBF with INTERPIVAR)
                (replace (BF FLAGS) of INTERPBF with \STK.BF)
                (CHECK (fetch (BF CHECKED) of INTERPBF)))
         (PROGN

                ;; Fix up FX header.  Some of this work is redundant in the case where NEWFX is the same as the old, but in general we did some
                ;; arg adjusting or did the slow case

                (SETQ NEWFX (IPLUS INTERPBF WORDSPERCELL))
                (\PUTBASE \STACKSPACE NEWFX 0)                    ; Clear FX flags
                (replace (FX FLAGS) of NEWFX with \STK.FX)
                (replace (FX NOPUSH) of NEWFX with T)             ; When we return to the user context, don't want any value to
                                                                  ; appear on stack
                (COND
                   (SLOWP (replace (FX %#BLINK) of NEWFX with INTERPBF)
                          (replace (FX %#CLINK) of NEWFX with (OR INTERPCLINK INTERPALINK))
                          ;; If INTERPCLINK is NIL, the original frame was not SLOW, so ALINK = CLINK and INTERPALINK has its low bit off
                          (replace (FX %#ALINK) of NEWFX with (LOGOR INTERPALINK 1)))
                   (T (replace (FX %#ALINK) of NEWFX with INTERPALINK)))
                (replace (FX FNHEADER) of NEWFX with CODE)
                (replace (FX PC) of NEWFX with (fetch (FNHEADER STARTPC) of CODE)))
         [PROGN                                                  ; Initialize PVAR region
                (SETQ SP (fetch (FX FIRSTPVAR) of NEWFX))
                (SETQ NPVARS (UNFOLD (ADD1 (fetch (FNHEADER PV) of CODE))
                                    CELLSPERQUAD))
                (COND
                   ((SETQ ENV (fetch (COMPILED-CLOSURE ENVIRONMENT) of CLOSURE))
                                                                 ; Set first pvar to closure environment
                    (\PUTBASEPTR \STACKSPACE SP ENV)
                    (add SP WORDSPERCELL)
                    (add NPVARS -1)))
                (RPTQ NPVARS (PROGN                              ; Fill in rest of Pvar region with 'unbound'
                                    (\PUTBASE \STACKSPACE SP 65535)
                                    (add SP WORDSPERCELL]
         (PROGN                                                  ; Make free block after this frame
                (replace (FX NEXTBLOCK) of NEWFX with (add SP (fetch (FX PADDING) of NEWFX)))
                                                                 ; Need extra junk quad after the pvar region
                (\MAKEFREEBLOCK SP (IDIFFERENCE STKEND SP))
                (CHECK (fetch (FX CHECKED) of NEWFX)))
         (replace (IFPAGE MiscFXP) of \InterfacePage with NEWFX])
```

(\STKNAME

```
  [LAMBDA (POS)                                                  ; Edited 27-Jan-91 14:20 by jds

    ;; Get the frame name from the stack frame at POS.  If that's an interpreter frame, get it's caller's frame name.

    (LET ((NAME (fetch (FX FRAMENAME) of POS)))
         (if (EQ NAME '\INTERPRETER)
             then [\GETBASEPTR \STACKSPACE (LET ((BFLINK (fetch (FX BLINK) of POS)))
                                                (+ (fetch (BF IVAR) of BFLINK)
                                                   (TIMES (CL:1- (fetch (BF NARGS) of BFLINK))
                                                          WORDSPERCELL]
           else NAME])
)

;; EVCALL ufn.

(DEFINEQ
```

(\ENVCALL.UFN

```
  [LAMBDA (\INTERRUPTABLE)                                       ; Edited 18-Apr-88 18:25 by bvm
    (\MISCAPPLY* (FUNCTION \SETUP-ENVIRONMENT-CALL])
```

(\SETUP-ENVIRONMENT-CALL

```
  [LAMBDA NIL                                                    ; Edited  4-Aug-88 11:39 by bvm

;;; Called in the misc context by ufn for ENVCALL, a variant on APPLYFN that takes on the stack: ..args.. #args codeblock environment.  Replace the ufn
;;; frame by the frame that would result if we had correctly done the call in microcode.  This is a normal function call of the codeblock with the one
;;; additional wrinkle that the environment is stored in PVAR0.

    (LET* ((UFNFX (fetch (IFPAGE MiscFXP) of \InterfacePage))
           (UFNNEXT (fetch (FX NEXTBLOCK) of UFNFX))
           (UFNBF (fetch (FX BLINK) of UFNFX))
           (UFNIVAR (fetch (BF IVAR) of UFNBF))
           (CALLER (fetch (FX CLINK) of UFNFX))
           (CALLNEXT (fetch (FX NEXTBLOCK) of CALLER))
          CODE ENV NA NACTUALS STKEND SLOWP LOCNARGS FREESIZE NEEDEDSIZE ARGSTART NEWSTACK IVAR)
         [SETQ LOCNARGS (STACKADDBASE (- CALLNEXT (UNFOLD 3 WORDSPERCELL]
                                                                 ; Location in caller's stack of number of args, followed by
                                                                 ; CODEBLOCK and ENV
         (SETQ NACTUALS (\GETBASEPTR LOCNARGS 0))                ; Fetch nargs, codeblock and environment from caller's stack
         (SETQ ENV (\GETBASEPTR LOCNARGS 4))
         (SETQ CODE (\GETBASEPTR LOCNARGS 2))
         (SETQ NA (fetch (FNHEADER NA) of CODE))                 ; Number args expected
         (SETQ NEEDEDSIZE (fetch (FNHEADER STKMIN) of CODE))
         (SETQ ARGSTART (- CALLNEXT (UNFOLD (+ NACTUALS 3)
                                            WORDSPERCELL)))      ; Address of first argument to function
         (if (AND (> NACTUALS NA)
                  (>= NA 0))
             then                                                ; More args than expected.  No need to retain/copy them (and it's
                                                                 ; best not to, since STKMIN calculation assumes expected args,
                                                                 ; not lots more)
                  (SETQ NACTUALS NA))
         (if (AND (if (EQ UFNBF (fetch (FX DUMMYBF) of UFNFX))
                      then                                       ; BF and FX contiguous
                           T
                    else                                         ; Yecch.  Non-contiguous BF and FX.  Toss the BF and do slow
```

```
                                                                        ; case
                (\MAKEFREEBLOCK UFNIVAR (+ WORDSPERCELL (- UFNBF UFNIVAR)))
                (SETQ UFNIVAR (fetch (FX DUMMYBF) of UFNFX))
                                                                        ; For next \makefreeblock -- have to blow away this dummybf
                NIL)
           (EQ UFNIVAR CALLNEXT)
           (PROGN                                                       ; Caller contiguous with ufn frame
                (SETQ FREESIZE (- UFNNEXT UFNIVAR))
                (bind SZ while (type? FSB UFNNEXT) do                   ; Add up the lengths of all the contiguous free blocks
                                                                     (add FREESIZE (SETQ SZ (fetch (FSB SIZE)
                                                                                              of UFNNEXT)))
                                                                     (add UFNNEXT SZ))
                (> FREESIZE NEEDEDSIZE)))
       then                                                             ; Normal case: there's enough space to build the frame on top of
                                                                        ; ufn frame.  Args are already there (we will lop them from
                                                                        ; caller's frame), so no copying to do.
           (SETQ STKEND UFNNEXT)
           (SETQ NEWSTACK ARGSTART)
     else                                                               ; Have to make a discontinuous frame elsewhere
         (\MAKEFREEBLOCK UFNIVAR (- UFNNEXT UFNIVAR))                   ; Discard ufn frame (and dummy bf before it in the
                                                                        ; non-contiguous case)
         (SETQ NEWSTACK (\FREESTACKBLOCK (+ NEEDEDSIZE (UNFOLD NACTUALS WORDSPERCELL))
                               UFNIVAR))                                ; Get a free block big enough for fn plus the args we'll have to
                                                                        ; copy
         (SETQ STKEND (+ NEWSTACK (fetch (FSB SIZE) of NEWSTACK)))
         (\BLT (STACKADDBASE NEWSTACK)
               (STACKADDBASE ARGSTART)
               (UNFOLD NACTUALS WORDSPERCELL))                          ; Copy args to new frame
         (\MAKEFREEBLOCK ARGSTART (- CALLNEXT ARGSTART))                ; Free up space taken in caller by args
         (SETQ SLOWP T))
   (SETQ IVAR NEWSTACK)
   (add NEWSTACK (UNFOLD NACTUALS WORDSPERCELL))
   (replace (FX NEXTBLOCK) of CALLER with ARGSTART)                     ; Shorten caller's frame to account for args removed
   (if (>= NA 0)
       then                                                             ; Fill in NIL for defaulted args
           (RPTQ (- NA NACTUALS)
                 (\PUTBASEPTR (STACKADDBASE NEWSTACK)
                        0 NIL)
                 (add NEWSTACK WORDSPERCELL)))
   (PROGN                                                               ; Fix up BF trailer cell
          (\PUTBASE (STACKADDBASE NEWSTACK)
                 0 \STK.BF.WORD)                                        ; Clear BF flags
          (replace (BF IVAR) of NEWSTACK with IVAR)
          (CHECK (fetch (BF CHECKED) of NEWSTACK)))
   (PROGN

         ;; Now build the FX header

          (add NEWSTACK WORDSPERCELL)
          (\PUTBASE (STACKADDBASE NEWSTACK)
                 0
                 (LLSH \STK.FX \STK.FLAGS.SHIFT))                       ; Clear FX flags
          (replace (FX NOPUSH) of NEWSTACK with T)                      ; When we return to the user context, don't want any value to
                                                                        ; appear on stack
          [LET ((ALINK (+ CALLER \#ALINK.OFFSET)))                      ; ALINK field for fast case
               (COND
                  (SLOWP                                                ; Have to make non-contiguous frame slow.  Set each piece
                                                                        ; manually to avoid redundant computation from the computed
                                                                        ; record fields
                         (replace (FX %#BLINK) of NEWSTACK with (fetch (FX DUMMYBF) of NEWSTACK))
                         (replace (FX %#CLINK) of NEWSTACK with ALINK)
                         (replace (FX %#ALINK) of NEWSTACK with (LOGOR ALINK 1))
                                                                        ; 1 is the slow bit

                         )
                  (T (replace (FX %#ALINK) of NEWSTACK with ALINK]
          (replace (FX FNHEADER) of NEWSTACK with CODE)
          (replace (FX PC) of NEWSTACK with (fetch (FNHEADER STARTPC) of CODE)))
   [LET ((SP (fetch (FX FIRSTPVAR) of NEWSTACK))
         (NPVARS (UNFOLD (ADD1 (fetch (FNHEADER PV) of CODE))
                        CELLSPERQUAD)))

        ;; Fill in rest of FX

        [PROGN                                                          ; Initialize PVAR region
               (if ENV
                   then                                                 ; Set first pvar to closure environment
                        (\PUTBASEPTR (STACKADDBASE SP)
                               0 ENV)
                        (add SP WORDSPERCELL)
                        (SETQ NPVARS (SUB1 NPVARS)))
               (RPTQ NPVARS (PROGN                                      ; Fill in rest of Pvar region with 'unbound'
                                   (\PUTBASE (STACKADDBASE SP)
                                          0 65535)
                                   (add SP WORDSPERCELL]
        (PROGN                                                          ; Make free block after this frame
               (replace (FX NEXTBLOCK) of NEWSTACK with (add SP (fetch (FX PADDING) of NEWSTACK)))
                                                                        ; Need extra junk quad after the pvar region
               (CHECK (> STKEND SP))
```

```
                        (\MAKEFREEBLOCK SP (- STKEND SP))
                        (CHECK (fetch (FX CHECKED) of NEWSTACK]
           (replace (IFPAGE MiscFXP) of \InterfacePage with NEWSTACK])

)

;; recursive interpreter

(DEFINEQ

(EVAL
  [LAMBDA (U \INTERNAL)
    (DECLARE (SPECVARS \INTERNAL))                              (* lmm "19-AUG-81 23:04")
    (\EVAL U])

(\EVAL
  [LAMBDA (FORM)                                                (* lmm " 3-NOV-81 15:42")

    ;; ufn for Interlisp EVAL opcode.

    (COND
       ((LISTP FORM)
        (\EVALFORM FORM))
       ((LITATOM FORM)
        (\EVALVAR FORM))
       ((NUMBERP FORM)
        FORM)
       (T (\EVALOTHER FORM])

(\EVALFORM
  [LAMBDA (*FORM* TEMP)
    (DECLARE (SPECVARS *FORM*)
           (ADDTOVAR LAMS FAULTEVAL))                           ; Edited 29-Jun-87 16:42 by amd

;;; eval of LISTP

    (PROG NIL
      RETRY
           [COND
              ((LITATOM (SETQ TEMP (CAR *FORM*)))
               (COND
                  ((fetch (LITATOM CCODEP) of TEMP)
                   (SELECTQ (fetch (LITATOM ARGTYPE) of TEMP)
                        (1 (GO NLSPREAD))
                        (3 (GO NLNOSPREAD))
                        (GO EVLAM)))
                  (T                                            ; EXPR OR UDF
                     (SETQ TEMP (fetch (LITATOM DEFPOINTER) of TEMP]
                                                                ; TEMP is now definition of EXPR
           (CL:TYPECASE TEMP
               ((OR COMPILED-CLOSURE CLOSURE)                   ; falls out
                                                  )
               (CONS (SELECTQ (CAR TEMP)
                         (NLAMBDA (COND
                                     ((OR (LISTP (SETQ TEMP (CADR TEMP)))
                                          (NULL TEMP))
                                      (GO NLSPREAD))
                                     (T (GO NLNOSPREAD))))
                         ((CL:LAMBDA LAMBDA OPENLAMBDA)
                         (GO FAULT)))
               (T (GO FAULT)))
      EVLAM

     ;; THIS FUNCTION'S DEFINITION VERY DEPENDENT ON THE SPECIAL MACRO IN ALAP FOR COMPILING IT.  --- SEE CEVALFORM

           [RETURN (PROG ((*ARGVAL* 0)
                          (*TAIL* *FORM*)
                          (*FN* (CAR *FORM*)))
                         (DECLARE (SPECVARS *ARGVAL* *FN* *TAIL*))
                         (RETURN (.EVALFORM.]
      NLSPREAD
           (RETURN (SPREADAPPLY (CAR *FORM*)
                          (CDR *FORM*)))
      NLNOSPREAD
           (RETURN (SPREADAPPLY* (CAR *FORM*)
                          (CDR *FORM*)))
      FAULT
           (COND
              ([AND CLISPARRAY (LISTP (SETQ TEMP (GETHASH *FORM* CLISPARRAY]
               (SETQ *FORM* TEMP)
               (GO RETRY)))
           (RETURN (FAULTEVAL *FORM*])

(\EVALFORMASLAMBDA
  [LAMBDA (FAULTX)                                              (* lmm "29-Apr-86 13:06")
```

```
    (PROG ((*ARGVAL* 0)
           (*TAIL* FAULTX)
           (*FN* (CAR FAULTX)))
          (DECLARE (SPECVARS *ARGVAL* *FN* *TAIL*))
          (RETURN (.EVALFORM.])
```

(\EVALOTHER

```
  [LAMBDA (X)                                                    (* lmm "10-MAY-80 17:03")

    ;; evaluate some other data type (not atom or list)

    (PROG NIL
          (RETURN (SPREADAPPLY* (CDR (OR (FASSOC (TYPENAME X)
                                                 \DEFEVALFNS)
                                         (RETURN X)))
                                X])
```

(APPLY

```
  [LAMBDA (U V \INTERNAL)
    (DECLARE (SPECVARS \INTERNAL))                               (* lmm "15-Aug-84 17:53")
    (.APPLY. U V])
```

(APPLY*

```
  [LAMBDA U                                                      (* lmm " 5-Jun-86 03:28")
    (PROG [(DEF (AND (IGREATERP U 0)
                     (ARG U 1]
       LP  (COND
              [(LITATOM DEF)
               (COND
                  [(fetch (LITATOM CCODEP) of DEF)
                   (COND
                      ((EQ (fetch (LITATOM ARGTYPE) of DEF)
                           3)
                       (GO NOSPR))
                      (T (GO SPR]
                  (T                                             ; EXPR
                     (SETQ DEF (OR (LISTP (fetch (LITATOM DEFPOINTER) of DEF))
                                   (GO FAULT]
              ((CCODEP DEF)
               (GO SPR))
              ((NLISTP DEF)
               (GO FAULT)))
          (SELECTQ (CAR DEF)
              ([LAMBDA CL:LAMBDA]
                   NIL)
              (FUNARG                                            ; Ignore environment of funarg?  This is obsolete anyway.
                      (SETQ DEF (CADR DEF))
                      (GO LP))
              (NLAMBDA (COND
                           ((AND (CAR (LISTP (CDR DEF)))
                                 (NLISTP (CADR DEF)))
                            (GO NOSPR))))
              (OPENLAMBDA)
              (GO FAULT))
       SPR [RETURN (SELECTQ U
                       (1                                        ; no args
                          (SPREADAPPLY* (ARG U 1)))
                       (2                                        ; 1 arg
                          (SPREADAPPLY* (ARG U 1)
                                 (ARG U 2)))
                       (3                                        ; 2 args
                          (SPREADAPPLY* (ARG U 1)
                                 (ARG U 2)
                                 (ARG U 3)))
                       (4                                        ; 3 args
                          (SPREADAPPLY* (ARG U 1)
                                 (ARG U 2)
                                 (ARG U 3)
                                 (ARG U 4)))
                       (SPREADAPPLY (ARG U 1)
                              (for I from 2 to U collect (ARG U I]
       FAULT
          [RETURN (FAULTAPPLY DEF (for I from 2 to U collect (ARG U I]
       NOSPR
                                                                 ; NLAMBDA*
          (RETURN (SPREADAPPLY* (ARG U 1)
                         (for I from 2 to U collect (ARG U I])
```

(\CHECKAPPLY*

```
  [LAMBDA (FN)                                                   (* bvm%: " 7-Jul-86 17:13")

;;; APPLY* compiles open as: [PUSH each arg, PUSH #args, PUSH FN, CHECKAPPLY*, APPLYFN] CHECKAPPLY* should merely return FN in the
;;; case where FN is a LAMBDA or a NLAMBDA spread.  The only case it needs to handle special is NLAMBDA nospread.
```

```
    (PROG ((DEF FN))
          [COND
             [(LITATOM DEF)
              (COND
                 ((NOT (fetch (LITATOM CCODEP) of DEF))              ; EXPR
                  (SETQ DEF (fetch (LITATOM DEFPOINTER) of DEF)))
                 ((EQ (fetch (LITATOM ARGTYPE) of DEF)
                      3)
                  (GO NOSPR))
                 (T (RETURN FN]
             ((AND NIL (TYPEP DEF 'COMPILED-CLOSURE))

              ;; Give a symbol this definition so APPLYFN can call it.  This is an utter kludge.  It can never work with preemptive scheduling, and
              ;; even without it is vulnerable to an interrupt between CHECKAPPLY* and APPLYFN

              (\PUTD '*\CHECKAPPLY*\HACK DEF)
              (RETURN '*\CHECKAPPLY*\HACK]
          (COND
             ((AND (LISTP DEF)
                   (EQ (CAR DEF)
                       'NLAMBDA)
                   (LISTP (SETQ DEF (CDR DEF)))
                   (CAR DEF)
                   (NLISTP (CAR DEF)))
              (GO NOSPR))
             (T (RETURN FN)))
      NOSPR
          (RETURN (LIST 'LAMBDA NIL (LIST 'QUOTE (SPREADAPPLY* FN (\CKAPPLYARGS])
```

(\CKAPPLYARGS

```
  [LAMBDA NIL                                                          (* lmm "10-NOV-81 22:26")
    (PROG ((FRAME (fetch (FX ALINK) of (\MYALINK)))
           ACNT PTR VAL)
          [SETQ ACNT (STACKGETBASEPTR (SETQ PTR (IDIFFERENCE (fetch (FX NEXTBLOCK) of FRAME)
                                                         WORDSPERCELL]
          (CHECK (SMALLPOSP ACNT))
          [FRPTQ ACNT (push VAL (STACKGETBASEPTR (SETQ PTR (IDIFFERENCE PTR WORDSPERCELL]
          (RETURN VAL])
```

(DEFEVAL

```
  [LAMBDA (TYPE FN)                                                    (* edited%: "13-DEC-78 23:18")
    (PROG ((F (FASSOC TYPE \DEFEVALFNS)))
          [COND
             (F (SETQ \DEFEVALFNS (DREMOVE F \DEFEVALFNS]
          [COND
             (FN (SETQ \DEFEVALFNS (CONS (CONS TYPE FN)
                                         \DEFEVALFNS]
          (RETURN (CDR F])
)

(DECLARE%: DONTCOPY

(DECLARE%: EVAL@COMPILE

(PUTPROPS .APPLY. MACRO [(U V)                                         (* body for APPLY, used by RETAPPLY too)
                         (PROG ((DEF U))
                           LP  [COND
                                  ((LITATOM DEF)
                                   (COND
                                      ((NOT (fetch (LITATOM CCODEP) of DEF))
                                                                       (* EXPR)
                                       (SETQ DEF (fetch (LITATOM DEFPOINTER) of DEF)))
                                      ((EQ (fetch (LITATOM ARGTYPE) of DEF)
                                           3)
                                       (GO NLSTAR))
                                      (T (GO NORMAL]
                               [COND
                                  ((LISTP DEF)
                                   (SELECTQ (CAR DEF)
                                       (NLAMBDA (AND (NLISTP (CADR DEF))
                                                     (CADR DEF)
                                                     (GO NLSTAR)))
                                       (FUNARG (SETQ DEF (CADR DEF))
                                               (GO LP))
                                       NIL))
                                  ((NULL DEF)
                                   (RETURN (FAULTAPPLY U V]
                           NORMAL
                               (RETURN (SPREADAPPLY U V))
                           NLSTAR
                                                                       (* NLAMBDA*)
                               (RETURN (SPREADAPPLY* U V])
)
)
```

;; Free variable manipulation

```
(DEFINEQ

(EVALV
  [LAMBDA (VAR POS RELFLG)                                   (* lmm " 6-Apr-84 16:37")

    ;; EVAL of a LITATOM without uba error

    [COND
       (POS (\SMASHLINK NIL (\STACKARGPTR POS]
    (PROG1 (\EVALV1 VAR)
        (COND
           (RELFLG (RELSTK POS))))])

(\EVALV1
  [LAMBDA (VAR)                                              (* lmm "24-DEC-81 00:08")
    (COND
       ((OR (NULL (\DTEST VAR 'LITATOM))
            (EQ VAR T))
        VAR)
       (T (\GETBASEPTR (\STKSCAN VAR)
                 0])

(\EVALVAR
  [LAMBDA (VAR)                                              ; Edited 30-Jan-87 13:28 by Pavel

    ;; EVAL of a LITATOM

    (COND
       ((OR (NULL VAR)
            (EQ VAR T))
        VAR)
       (T (LET ((VP (\STKSCAN VAR))
                VAL)
               (COND
                  ((EQ (SETQ VAL (\GETBASEPTR VP 0))
                       'NOBIND)                              ; Value is NOBIND, even if it was not found as the top-level
                                                             ; value.  This is consistent with BOUNDP.
                   (FAULTEVAL VAR))
                  (T VAL])

(BOUNDP
  [CL:LAMBDA (CL::VAR)                                       ; Edited 30-Jan-87 16:59 by Pavel

         ;; True if VAR is bound or has top level value

         (AND (CL:SYMBOLP CL::VAR)
              (NOT (EQ (\GETBASEPTR (\STKSCAN CL::VAR)
                                   0)
                       'NOBIND])

(SET
  [LAMBDA (VAR VALUE)                                        (* lmm "24-FEB-82 16:11")
    (COND
       ((NULL VAR)
        (AND VALUE (LISPERROR "ATTEMPT TO SET NIL OR T" VALUE)))
       (T (PROG [(VP (\STKSCAN (\DTEST VAR 'LITATOM]
                (COND
                   ((EQ (\HILOC VP)
                        \STACKHI)
                    (\PUTBASEPTR VP 0 VALUE))
                   ((EQ VAR T)
                    (OR (EQ VALUE T)
                        (LISPERROR "ATTEMPT TO SET NIL OR T" VALUE)))
                   (T (\RPLPTR VP 0 VALUE)))
                (RETURN VALUE])

(\SETVAR
  [LAMBDA (VAR VALUE)                                        (* lmm "24-FEB-82 16:11")
    (COND
       ((NULL VAR)
        (AND VALUE (LISPERROR "ATTEMPT TO SET NIL OR T" VALUE)))
       (T (PROG [(VP (\STKSCAN (\DTEST VAR 'LITATOM]
                (COND
                   ((EQ (\HILOC VP)
                        \STACKHI)
                    (\PUTBASEPTR VP 0 VALUE))
                   ((EQ VAR T)
                    (OR (EQ VALUE T)
                        (LISPERROR "ATTEMPT TO SET NIL OR T" VALUE)))
                   (T (\RPLPTR VP 0 VALUE)))
                (RETURN VALUE])
```

```
(SETQ
  [NLAMBDA U                                                    (* lmm "24-DEC-81 00:19")
    (\SETVAR (CAR U)
           (PROG ((*TAIL* (CDR U)))
                 (DECLARE (SPECVARS *TAIL*))
                 (RETURN (PROG1 (\EVAL (CAR *TAIL*))
                                [PROG NIL                       ; (SETQ X Y + 3) must try to eval + in order to get dwim involved.
                                  LP  (COND
                                             ((LISTP (SETQ *TAIL* (CDR *TAIL*)))
                                              (\EVAL (CAR *TAIL*))
                                              (GO LP])])


(\STKSCAN
  [LAMBDA (VAR)                                                 ; Edited 27-Jan-91 14:48 by jds

    ;; Returns pointer to place where VAR is bound

    (PROG ((FX (fetch (FX ALINK) of (\MYALINK)))
           (ATOM# (\ATOMVALINDEX VAR))
           NTSIZE A VARINFO PVAROFFSET NT FVAR)
      FRAMELP
          [COND
             ((fetch (FX INVALIDP) of FX)                       ; Reached top of stack without finding a binding
              (RETURN (fetch (VALINDEX VCELL) of ATOM#]
          (SETQ NT (fetch (FX NAMETABLE) of FX))
          (SETQ NTSIZE (fetch (FNHEADER NTSIZE) of NT))
          (SETQ NT (\ADDBASE NT (fetch (FNHEADER OVERHEADWORDS) of T)))
      TABLELP
          [COND
             ((EQP (SETQ A (GETNAMEENTRY NT 0))
                   0)                                           ; End of name table
              (GO ENDTABLE))
             ((EQP A ATOM#)                                     ; Found ATOM#.  See if it is really bound here
              (SELECTC (NTSLOT-VARTYPE (SETQ VARINFO (GETNTOFFSETENTRY NT NTSIZE)))
                  (IVARCODE                                     ; Is bound in BF
                                                                ; IVAR
                            [RETURN (STACKADDBASE (IPLUS (UNFOLD (\LOLOC (NTSLOT-OFFSET VARINFO))
                                                                 WORDSPERCELL)
                                                         (fetch (BF IVAR) of (fetch (FX BLINK) of FX])
                  (PVARCODE                                     ; Local may or may not be bound yet
                            (SETQ PVAROFFSET (IPLUS (UNFOLD (\LOLOC (NTSLOT-OFFSET VARINFO))
                                                            WORDSPERCELL)
                                                    (fetch (FX FIRSTPVAR) of FX)))
                            [COND
                               ((fetch (PVARSLOT BOUND) of (STACKADDBASE PVAROFFSET))
                                                                ; PVAR
                                (RETURN (STACKADDBASE PVAROFFSET])
                  (FVARCODE                                     ; If FVAR is looked up, we can use it.
                            [SETQ FVAR (ADDSTACKBASE (IPLUS (UNFOLD (\LOLOC (NTSLOT-OFFSET VARINFO))
                                                                    WORDSPERCELL)
                                                            (fetch (FX FIRSTPVAR) of FX]
                            (COND
                               ((fetch (FVARSLOT LOOKEDUP) of FVAR)
                                (SETQ FVAR (fetch (FVARSLOT BINDINGPTR) of FVAR))
                                (RETURN FVAR))
                               (T (GO ENDTABLE))))
                  (SHOULDNT]
          [SETQ NT (\ADDBASE NT (CONSTANT (WORDSPERNAMEENTRY]
          (GO TABLELP)
      ENDTABLE
          (SETQ FX (fetch (FX ALINK) of FX))
          (GO FRAMELP])


(\SETFVARSLOT
  [LAMBDA (VAR NEWBINDING)                                      ; Edited 20-Feb-91 01:07 by jds

    ;; Sets the freevar binding slot of VAR in caller's frame to point at NEWBINDING

    (PROG ((FX (\MYALINK))
           (ATOM# (NEW-SYMBOL-CODE VAR (\ATOMVALINDEX VAR)))
           NTSIZE A VARINFO NT)
          (SETQ NT (fetch (FX NAMETABLE) of FX))
          (SETQ NTSIZE (fetch (FNHEADER NTSIZE) of NT))
          (SETQ NT (\ADDBASE NT (fetch (FNHEADER OVERHEADWORDS) of T)))
      TABLELP
          (COND
             ((NULL-NTENTRY (SETQ A (GETSTKNAMEENTRY NT 0)))    ; End of name table
              (ERROR "Binding slot not found in caller's frame" VAR))
             ((AND (EQP A ATOM#)
                   (EQP (NTSLOT-VARTYPE (SETQ VARINFO (GETSTKNTOFFSETENTRY NT NTSIZE)))
                        FVARCODE))
              (replace (FVARSLOT BINDINGPTR) of (ADDSTACKBASE (IPLUS (UNFOLD (\LOLOC (NTSLOT-OFFSET VARINFO))
                                                                             WORDSPERCELL)
                                                                     (fetch (FX FIRSTPVAR) of FX)))
                 with NEWBINDING)
              (RETURN NEWBINDING)))
          [SETQ NT (\ADDBASE NT (CONSTANT (WORDSPERNAMEENTRY]
          (GO TABLELP])

)

;; PROG and friends

(DEFINEQ

(PROG
  [NLAMBDA U                                                    ; Edited 21-Jan-91 18:33 by jds

    ;; PROG unpacks the argument list and changes any EVAL type forms by evaluating the form and then smashing the name and value

    ;; NOTE --- this mechanism might confuse DWIM someday because the arguments inside the PROG are evaluated at a time when the PROG
    ;; frame is in a very funny state: the 'values' are the variables, and the variables are NIL

    (PROG ((NVARS 0)
           (VARLST (CAR U))
           NTSIZE NNILS)
          (for VAR in VARLST do                                 ; Count number of vars to bind, check validity
                                 (COND
                                    ((OR (NULL (\DTEST (COND
                                                          ((LISTP VAR)
                                                           (SETQ VAR (CAR VAR)))
                                                          (T VAR))
                                                      'LITATOM))
                                         (EQ VAR T))
                                     (LISPERROR "ATTEMPT TO BIND NIL OR T" VAR)))
                                 (add NVARS 1))
          (RETURN (.CALLAFTERPUSHINGNILS. (SETQ NNILS (IPLUS NVARS (SETQ NTSIZE
                                                                     (CEIL [ADD1 (UNFOLD NVARS (CONSTANT (
                                                                                                WORDSPERNAMEENTRY
                                                                                                       ]
                                                                           WORDSPERQUAD))
                                                              (FOLDHI (fetch (FNHEADER OVERHEADWORDS)
                                                                             of T)
                                                                     WORDSPERCELL)
                                                              (SUB1 CELLSPERQUAD)))
                           (\PROG0 U U NNILS NVARS NTSIZE VARLST])


(\PROG0
  [LAMBDA (*FIRSTTAIL* *TAIL* NNILS NVARS NTSIZE VARLST)          ; Edited 27-Jan-91 14:32 by jds
    (DECLARE (SPECVARS *TAIL* *FIRSTTAIL*))
    (PROG NIL
          [COND
             (VARLST

                      ;; Create a nametable inside progframe where PROG pushed all those nils

                      (PROG ((PROGFRAME (\MYALINK))
                             HEADER NT NILSTART)
                            (SETQ HEADER (fetch (FX FNHEADER) of PROGFRAME))
                            (SETQ NT (ADDSTACKBASE (CEIL (IPLUS (SETQ NILSTART (IDIFFERENCE (fetch (FX NEXTBLOCK
                                                                                                         )
                                                                                                  of PROGFRAME)
                                                                                           (UNFOLD NNILS
                                                                                                  WORDSPERCELL)))
                                                                (UNFOLD NVARS WORDSPERCELL))
                                                         WORDSPERQUAD)))

                       ;; NT is address of our synthesized nametable: beginning of NIL's, not counting additional PVARs we are about to bind,
                       ;; rounded up to quadword

                            [for VAR in VARLST as VALUEOFF from NILSTART by WORDSPERCELL
                               do                                   ; evaluate initial values first
                                  (COND
                                     ((LISTP VAR)
                                      (PUTBASEPTR \STACKSPACE VALUEOFF (\EVPROG1 (CDR VAR]
                                                                    ; then build NT
                            (UNINTERRUPTABLY
                                (for VAR in VARLST as VAR# from (FOLDLO (IDIFFERENCE NILSTART (fetch (FX FIRSTPVAR)
                                                                                                  of PROGFRAME))
                                                                      WORDSPERCELL)
                                   as NT1 from (fetch (FNHEADER OVERHEADWORDS) of T) by (WORDSPERNAMEENTRY)
                                   as NT2 from (IPLUS (fetch (FNHEADER OVERHEADWORDS) of T)
                                                      NTSIZE)
                                   by (CONSTANT (WORDSPERNTOFFSETENTRY))
                                   do [SETSTKNAME-RAW NT NT1 (\ATOMVALINDEX (COND
                                                                                ((LISTP VAR)
                                                                                 (CAR VAR))
                                                                                (T VAR]
                                      (SETSTKNTOFFSET-RAW NT NT2 PVARCODE VAR#))
                                (replace (FNHEADER %#FRAMENAME) of NT with 'PROG)
                                (replace (FNHEADER NTSIZE) of NT with NTSIZE)
                                                                    ; Do I need to worry about STK, NA, PV, START, ARGTYPE
                                                                    ; NLOCALS ?  --- no
                                (replace (FX NAMETABLE) of PROGFRAME with NT))]
      EVLP
          (COND
               ((NULL (SETQ *TAIL* (CDR *TAIL*)))
                (RETURN NIL))
               (T (\EVAL (OR (LISTP (CAR *TAIL*))
                             (GO EVLP)))
                  (GO EVLP])


(\EVPROG1
  [LAMBDA (*TAIL*)                                            (* lmm "14-MAY-80 13:00")
    (DECLARE (SPECVARS *TAIL*))
    (PROG1 (\EVAL (CAR *TAIL*))
        [PROG NIL
           LP  (COND
                  ((LISTP (SETQ *TAIL* (CDR *TAIL*)))
                   (\EVAL (CAR *TAIL*))
                   (GO LP])])


(RETURN
  [NLAMBDA (FORM)
    (DECLARE (LOCALVARS . T))                                 (* bvm%: "10-Nov-86 18:22")
    (PROG ((MV (CL:MULTIPLE-VALUE-LIST (\EVAL FORM)))
           (FRAME (\MYALINK)))
      LP  (COND
             ((EQ (fetch (FNHEADER FRAMENAME) of (fetch (FX NAMETABLE) of FRAME))
                  (FUNCTION \PROG0))
              (SETQ FRAME (fetch (FX CLINK) of FRAME))                ; Its caller, i.e. PROG
              (\SMASHRETURN NIL FRAME)                                ; Make us return to PROG with this value
              (RETURN (CL:VALUES-LIST MV)))
             ([NOT (fetch (FX INVALIDP) of (SETQ FRAME (fetch (FX CLINK) of FRAME]
              (GO LP))
             (T (LISPERROR "ILLEGAL RETURN"])


(GO
  [NLAMBDA U                                                  (* bvm%: "10-Nov-86 18:17")
    (PROG ((FRAME (\MYALINK))
           (LABEL (CAR U))
           GOTAIL FIRSTARG)
      LP  [COND
             ((EQ (fetch (FNHEADER FRAMENAME) of (fetch (FX NAMETABLE) of FRAME))
                  (FUNCTION \PROG0))
              (COND
                 ([SETQ GOTAIL (FMEMB LABEL (CDR (STACKGETBASEPTR (SETQ FIRSTARG (fetch (BF IVAR)
                                                                                     of (fetch (FX BLINK)
                                                                                            of FRAME]

                  ;; first argument of \PROG0 is the actual tail of the prog, which can contain the labels.  Second argument is the 'current' *TAIL*

                  (STACKPUTBASEPTR (IPLUS FIRSTARG WORDSPERCELL)
                         GOTAIL)                                      ; Reset *TAIL* in the \PROG0 frame
                  (\SMASHRETURN NIL FRAME)                            ; Fix it so we return to \PROG0 to continue evaluating after label
                  (RETURN NIL]
          (COND
             ([NOT (fetch (FX INVALIDP) of (SETQ FRAME (fetch (FX CLINK) of FRAME]
              (GO LP))
             (T (LISPERROR "UNDEFINED OR ILLEGAL GO" LABEL])


(EVALA
  [LAMBDA (X A)                                               ; Edited 21-Jan-91 18:13 by jds

;;; Evaluate X after spreading alist A on stack

    (PROG ((NVARS 0)
           NTSIZE NNILS TMP)
          (for VAR in A do                                    ; Count number of vars to bind, check validity
                           (COND
                              ((OR [NULL (SETQ TMP (\DTEST (CAR (\DTEST VAR 'LISTP))
                                                           'LITATOM]
                                   (EQ TMP T))
                               (LISPERROR "ATTEMPT TO BIND NIL OR T" TMP)))
                           (add NVARS 1))
          (RETURN (.CALLAFTERPUSHINGNILS. (SETQ NNILS (IPLUS NVARS (SETQ NTSIZE
                                                                    (CEIL [ADD1 (UNFOLD NVARS (CONSTANT (
                                                                                               WORDSPERNAMEENTRY
                                                                                                          ]
                                                                          WORDSPERQUAD))
                                                            (FOLDHI (fetch (FNHEADER OVERHEADWORDS)
                                                                       of T)
                                                                   WORDSPERCELL)
                                                            (SUB1 CELLSPERQUAD)))
                           (\EVALA NNILS NVARS NTSIZE X A])


(\EVALA
  [LAMBDA (NNILS NVARS NTSIZE FORM ALIST)                     ; Edited 27-Jan-91 16:39 by jds
    (PROG ((CALLER (\MYALINK))
           NILSTART NT HEADER)
     ;; Create a nametable inside CALLER where EVALA pushed all those nils
          (SETQ HEADER (fetch (FX FNHEADER) of CALLER))            ; The function header of code for EVALA
          (SETQ NT (ADDSTACKBASE (CEIL (IPLUS (SETQ NILSTART (IDIFFERENCE (fetch (FX NEXTBLOCK) of CALLER)
                                                                               (UNFOLD NNILS WORDSPERCELL)))
                                                    (UNFOLD NVARS WORDSPERCELL))
                                          WORDSPERQUAD)))
     ;; Address of our synthesized nametable: beginning of NIL's, not counting additional PVARs we are about to bind, rounded up to quadword
          (UNINTERRUPTABLY
              (for PAIR in ALIST as VAR# from (FOLDLO (IDIFFERENCE NILSTART (fetch (FX FIRSTPVAR) of CALLER))
                                                       WORDSPERCELL)
                 as NT1 from (fetch (FNHEADER OVERHEADWORDS) of T) by (WORDSPERNAMEENTRY) as NT2
                 from (IPLUS (fetch (FNHEADER OVERHEADWORDS) of T)
                             NTSIZE)
                 by (CONSTANT (WORDSPERNTOFFSETENTRY)) as VALUEOFF from NILSTART by WORDSPERCELL
                 do (PUTBASEPTR \STACKSPACE VALUEOFF (CDR PAIR))
                    (SETSTKNAME-RAW NT NT1 (\ATOMVALINDEX (CAR PAIR)))
                    (SETSTKNTOFFSET-RAW NT NT2 PVARCODE VAR#))
              ;; now fix up header of NT
              (replace (FNHEADER %#FRAMENAME) of NT with 'EVALA)
              (replace (FNHEADER NTSIZE) of NT with NTSIZE)         ; Do I need to worry about STK, NA, PV, START, ARGTYPE ?
                                                                      ; --- probably not
              (replace (FX NAMETABLE) of CALLER with NT))
          (RETURN (\EVAL FORM])

(ERRORSET
  [LAMBDA (FORM FLAG)                                                   ; Edited  6-Mar-87 14:39 by amd
      ;; (proceed-case (handler-bind ...)), but open-coded to aviod a proceed-case literal
    (LET (SI::NLSETQ-VALUE)
         (CL:IF (EQ (LET [(*PROCEED-CASES* (CONS SI::NLSETQ-PROCEED-CASE *PROCEED-CASES*))
                          (SI::*NLSETQFLAG* (NEQ FLAG T))
                          (*CONDITION-HANDLER-BINDINGS* (CL:IF FLAG
                                                            *CONDITION-HANDLER-BINDINGS*
                                                            (CONS '(CL:ERROR . SI::NLSETQHANDLER)
                                                                  *CONDITION-HANDLER-BINDINGS*))]
                         (DECLARE (SPECVARS SI::*NLSETQFLAG*))
                         (CL:CATCH *PROCEED-CASES*
                             (CL:SETQ SI::NLSETQ-VALUE (LIST (\EVAL FORM)))
                             :NORMAL))
                 :NORMAL)
          SI::NLSETQ-VALUE
          NIL)])

(SI::ERRORSET-PRINT-FUNCTION
  [LAMBDA (DATUM STREAM)
    (DECLARE (IGNORE DATUM))                                            (* bvm%: "11-Nov-86 21:40")
    (PRIN1 "Unwind to ERRORSET" STREAM])
)

;; LET and friends -- need these in the init

(DEFINEQ

(LET
  [NLAMBDA U
    (DECLARE (LOCALVARS . T))                                           ; Edited 21-Jan-91 18:14 by jds
      ;; LET unpacks the argument list and changes any EVAL type forms by evaluating the form and then smashing the name and value
    (LET ((NVARS 0)
          (VARLST (CAR U))
          NTSIZE NNILS)
         (for VAR in VARLST do                                          ; Count number of vars to bind, check validity
                              (COND
                                 ((OR (NULL (\DTEST (COND
                                                       ((LISTP VAR)
                                                        (SETQ VAR (CAR VAR)))
                                                       (T VAR))
                                                   'LITATOM))
                                      (EQ VAR T))
                                  (LISPERROR "ATTEMPT TO BIND NIL OR T" VAR)))
                              (add NVARS 1))
         (.CALLAFTERPUSHINGNILS. (SETQ NNILS (IPLUS NVARS (SETQ NTSIZE (CEIL [ADD1 (UNFOLD NVARS (CONSTANT
                                                                                                    (
                                                                                               WORDSPERNAMEENTRY
                                                                                                      ]
                                                                              WORDSPERQUAD))
                                                            (FOLDHI (fetch (FNHEADER OVERHEADWORDS) of T)
                                                                    WORDSPERCELL)
                                                            (SUB1 CELLSPERQUAD)))
                 (\LET0 (CDR U)
                         NNILS NVARS NTSIZE VARLST])

(LET*
  [NLAMBDA U
    (DECLARE (LOCALVARS . T))                                    ; Edited 21-Jan-91 23:02 by jds

    ;; LET* unpacks the argument list and changes any EVAL type forms by evaluating the form and then smashing the name and value

    (LET* ((NVARS 0)
           (VARLST (CAR U))
           NTSIZE NNILS)
          (for VAR in VARLST do                                  ; Count number of vars to bind, check validity
                                (COND
                                   ((OR (NULL (\DTEST (COND
                                                         ((LISTP VAR)
                                                          (SETQ VAR (CAR VAR)))
                                                         (T VAR))
                                                     'LITATOM))
                                        (EQ VAR T))
                                    (LISPERROR "ATTEMPT TO BIND NIL OR T" VAR)))
                                (add NVARS 1))
          (.CALLAFTERPUSHINGNILS. (SETQ NNILS (IPLUS NVARS (SETQ NTSIZE (CEIL [ADD1 (UNFOLD NVARS (CONSTANT
                                                                                                      (
                                                                                                 WORDSPERNAMEENTRY
                                                                                                      ]
                                                                                   WORDSPERQUAD))
                                                     (FOLDHI (fetch (FNHEADER OVERHEADWORDS) of T)
                                                             WORDSPERCELL)
                                                     (SUB1 CELLSPERQUAD)))
                 (\LET* (CDR U)
                        NNILS NVARS NTSIZE VARLST])

(\LET0
  [LAMBDA (*TAIL* NNILS NVARS NTSIZE VARLST)
    (DECLARE (LOCALVARS . T)
           (SPECVARS *TAIL*))                                    ; Edited 27-Jan-91 14:35 by jds
    (PROG NIL
          [COND
             (VARLST (PROG ((PROGFRAME (\MYALINK))
                            HEADER NT NILSTART)
                           (SETQ HEADER (fetch (FX FNHEADER) of PROGFRAME))
                           (SETQ NT (ADDSTACKBASE (CEIL (IPLUS (SETQ NILSTART (IDIFFERENCE
                                                                                (fetch (FX NEXTBLOCK)
                                                                                   of PROGFRAME)
                                                                                (UNFOLD NNILS WORDSPERCELL)))
                                                                (UNFOLD NVARS WORDSPERCELL))
                                                         WORDSPERQUAD)))

                     ;; NT is address of our synthesized nametable: beginning of NIL's, not counting additional PVARs we are about to bind,
                     ;; rounded up to quadword

                           [for VAR in VARLST as VALUEOFF from NILSTART by WORDSPERCELL
                              do                                 ; evaluate initial values first
                                 (COND
                                    ((LISTP VAR)
                                     (PUTBASEPTR \STACKSPACE VALUEOFF (\EVPROG1 (CDR VAR]
                                                                 ; then build NT
                           (UNINTERRUPTABLY
                               (for VAR in VARLST as VAR# from (FOLDLO (IDIFFERENCE NILSTART (fetch (FX FIRSTPVAR
                                                                                                        )
                                                                                                 of PROGFRAME))
                                                                      WORDSPERCELL)
                                  as NT1 from (fetch (FNHEADER OVERHEADWORDS) of T) by (CONSTANT (
                                                                                                 WORDSPERNAMEENTRY
                                                                                                   ))
                                  as NT2 from (IPLUS (fetch (FNHEADER OVERHEADWORDS) of T)
                                                     NTSIZE)
                                  by (CONSTANT (WORDSPERNTOFFSETENTRY))
                                  do [SETSTKNAME-RAW NT NT1 (\ATOMVALINDEX (COND
                                                                             ((LISTP VAR)
                                                                              (CAR VAR))
                                                                             (T VAR]
                                        (SETSTKNTOFFSET-RAW NT NT2 PVARCODE VAR#))
                               (replace (FNHEADER %#FRAMENAME) of NT with 'LET)
                               (replace (FNHEADER NTSIZE) of NT with NTSIZE)
                                                                 ; Do I need to worry about STK, NA, PV, START, ARGTYPE
                                                                 ; NLOCALS ?  --- no
                               (replace (FX NAMETABLE) of PROGFRAME with NT))))]
      EVLP
          (COND
             [(NULL (CDR *TAIL*))
              (RETURN (\EVAL (CAR *TAIL*]
             (T (\EVAL (CAR *TAIL*))
                (SETQ *TAIL* (CDR *TAIL*))
                (GO EVLP])
```

```
(\LET*
  [LAMBDA (*TAIL* NNILS NVARS NTSIZE VARLST)
    (DECLARE (LOCALVARS . T)
           (SPECVARS *TAIL*))                                           ; Edited 27-Jan-91 14:37 by jds
    (PROG NIL
          [COND
             (VARLST (PROG ((PROGFRAME (\MYALINK))
                            HEADER NT NILSTART)
                           (SETQ HEADER (fetch (FX FNHEADER) of PROGFRAME))
                           (SETQ NT (ADDSTACKBASE (CEIL (IPLUS (SETQ NILSTART (IDIFFERENCE
                                                                               (fetch (FX NEXTBLOCK)
                                                                                  of PROGFRAME)
                                                                               (UNFOLD NNILS WORDSPERCELL)))
                                                                (UNFOLD NVARS WORDSPERCELL))
                                                          WORDSPERQUAD)))

                      ;; NT is address of our synthesized nametable: beginning of NIL's, not counting additional PVARs we are about to bind,
                      ;; rounded up to quadword

                      ;; First build the nametable.  This differs from LET in that the nametable is built in reverse order and the PVARs are
                      ;; marked unbound before we start evaluating the forms.  This way, we get the right semantics of nested LET's.

                           (UNINTERRUPTABLY
                               (FOR VAR IN VARLST AS VAR# FROM (FOLDLO (IDIFFERENCE NILSTART
                                                                              (FETCH (FX FIRSTPVAR)
                                                                                 OF PROGFRAME))
                                                                       WORDSPERCELL)
                                  AS VALUEOFF FROM NILSTART BY WORDSPERCELL AS NT1
                                  FROM (IPLUS (UNFOLD (SUB1 NVARS)
                                                     (CONSTANT (WORDSPERNAMEENTRY)))
                                              (FETCH (FNHEADER OVERHEADWORDS) OF T))
                                  BY (IMINUS (CONSTANT (WORDSPERNAMEENTRY))) AS NT2
                                  FROM (IPLUS (UNFOLD (SUB1 NVARS)
                                                     (CONSTANT (WORDSPERNAMEENTRY)))
                                              (FETCH (FNHEADER OVERHEADWORDS) OF T)
                                              NTSIZE)
                                  BY (IMINUS (CONSTANT (WORDSPERNTOFFSETENTRY)))
                                  DO [SETSTKNAME-RAW NT NT1 (\ATOMVALINDEX (COND
                                                                              ((LISTP VAR)
                                                                               (CAR VAR))
                                                                              (T VAR]
                                     (SETSTKNTOFFSET-RAW NT NT2 PVARCODE VAR#)
                                     (PUTBASE \STACKSPACE VALUEOFF 65535))
                               (replace (FNHEADER %#FRAMENAME) of NT with 'LET*)
                               (REPLACE (FNHEADER NTSIZE) OF NT WITH NTSIZE)
                                                                ; Do I need to worry about STK, NA, PV, START, ARGTYPE
                                                                ; NLOCALS ?  --- no
                               (REPLACE (FX NAMETABLE) OF PROGFRAME WITH NT))
                           (FOR VAR IN VARLST AS VALUEOFF FROM NILSTART BY WORDSPERCELL
                              DO                                ; evaluate initial values first
                                 (PUTBASEPTR \STACKSPACE VALUEOFF (IF (LISTP VAR)
                                                                      THEN (\EVPROG1 (CDR VAR))
                                                                    ELSE NIL]
      EVLP
          (COND
             [(NULL (CDR *TAIL*))
              (RETURN (\EVAL (CAR *TAIL*]
             (T (\EVAL (CAR *TAIL*))
                (SETQ *TAIL* (CDR *TAIL*))
                (GO EVLP])
)

(DEFINEQ

(QUOTE
  [NLAMBDA U
    (CAR U]))


(AND
  [NLAMBDA U
    (DECLARE (SPECVARS *TAIL*))
    (OR (NLISTP U)
        (PROG ((*TAIL* U))
          LP  (RETURN (COND
                         ((NLISTP (CDR *TAIL*))
                          (\EVAL (CAR *TAIL*)))
                         ((\EVAL (CAR *TAIL*))
                          (SETQ *TAIL* (CDR *TAIL*))
                          (GO LP])


(OR
  [NLAMBDA U
    (DECLARE (SPECVARS *TAIL*))                                   (* lmm " 9-May-86 13:45")
    (AND (LISTP U)
         (PROG ((*TAIL* U))
           LP  (RETURN (COND
                                ((NLISTP (CDR *TAIL*))
                                 (\EVAL (CAR *TAIL*)))
                                (T (OR (\EVAL (CAR *TAIL*))
                                       (PROGN (SETQ *TAIL* (CDR *TAIL*))
                                              (GO LP])

(PROGN
  [NLAMBDA U                                          ; MUST be a NLAMBDA* with internal call to EVAL for dwimsake
    (DECLARE (SPECVARS *TAIL*))
    (AND (LISTP U)
         (PROG ((*TAIL* U))
           LP  (COND
                  [(NLISTP (CDR *TAIL*))
                   (RETURN (\EVAL (CAR *TAIL*]
                  (T (\EVAL (CAR *TAIL*))
                     (SETQ *TAIL* (CDR *TAIL*))
                     (GO LP])

(COND
  [NLAMBDA U
    (DECLARE (SPECVARS *TAIL*))                       (* lmm "25-APR-80 18:03")
    (PROG ((*TAIL* U)
           VAL)
      LP  (RETURN (COND
                     ((NLISTP *TAIL*)
                      (COND
                         (*TAIL* (LISPERROR "UNUSUAL CDR ARG LIST" *TAIL*))
                         (T NIL)))
                     ((SETQ VAL (\EVAL (CAAR *TAIL*)))
                      (COND
                         ((CDAR *TAIL*)
                          (\EVPROGN (CDAR *TAIL*)))
                         (T VAL)))
                     (T (SETQ *TAIL* (CDR *TAIL*))
                        (GO LP])

(\EVPROGN
  [LAMBDA (*TAIL*)                                    (* lmm "18-Feb-86 01:44")
    (DECLARE (SPECVARS *TAIL*))
    (PROG NIL
      LP  (COND
             ((CDR *TAIL*)
              (\EVAL (CAR *TAIL*))
              (SETQ *TAIL* (CDR *TAIL*))
              (GO LP))
             (T (RETURN (\EVAL (CAR *TAIL*])

(PROG1
  [NLAMBDA U
    (DECLARE (SPECVARS *TAIL*))                       (* lmm "14-MAY-80 12:59")
    (AND (LISTP U)
         (PROG ((*TAIL* U))
               (RETURN (PROG1 (\EVAL (CAR *TAIL*))
                              [PROG NIL
                                 LP  (COND
                                        ((LISTP (SETQ *TAIL* (CDR *TAIL*)))
                                         (\EVAL (CAR *TAIL*))
                                         (GO LP])])
)

(RPAQQ \DEFEVALFNS NIL)

(RPAQQ \EVALHOOK NIL)

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(SPECVARS *EVALHOOK*)
)

(ADDTOVAR LAMBDASPLST LAMBDA NLAMBDA CL:LAMBDA OPENLAMBDA)

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS \DEFEVALFNS \EVALHOOK LAMBDASPLST CLISPARRAY)
)

(DECLARE%: DONTEVAL@LOAD DOCOPY

(RPAQQ CLISPARRAY NIL)

(MOVD? 'SETQ 'SETN NIL T)
)
```

```
(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS CLISPARRAY)
)

;; Evaluating in different stack environment

(DEFINEQ

(ENVEVAL
  [LAMBDA (FORM APOS CPOS AFLG CFLG)                                  (* bvm%: "18-AUG-81 23:29")
    (\CALLME '*ENV*)
    (\SMASHLINK NIL (AND APOS (\STACKARGPTR APOS))
           (AND CPOS (\STACKARGPTR CPOS)))
    (COND
       (AFLG (RELSTK APOS)))
    (COND
       (CFLG (RELSTK CPOS)))
    (\EVAL FORM])

(ENVAPPLY
  [LAMBDA (FN ARGS APOS CPOS AFLG CFLG)                               (* lmm "15-Aug-84 17:53")
    (\CALLME '*ENV*)
    (\SMASHLINK NIL (AND APOS (\STACKARGPTR APOS))
           (AND CPOS (\STACKARGPTR CPOS)))
    (COND
       (AFLG (RELSTK APOS)))
    (COND
       (CFLG (RELSTK CPOS)))
    (.APPLY. FN ARGS])

(FUNCTION
  [NLAMBDA (FN ENV)                                                   (* bvm%: "11-Nov-86 21:23")
    (COND
       (ENV (if NIL
                then (LIST 'FUNARG FN (STKNTH -1 'FUNCTION))
              else (ERROR "FUNARGs no longer supported--use Common Lisp lexical closures" FN)))
       (T FN])

(\FUNCT1
  [LAMBDA (NNILS NVARS NTSIZE VARLST)                                 ; Edited 27-Jan-91 14:39 by jds
    (PROG ((FUNCTFRAME (\MYALINK))
           HEADER NT NILSTART)
          (SETQ HEADER (fetch (FX FNHEADER) of FUNCTFRAME))
          (SETQ NT (ADDSTACKBASE (CEIL (IPLUS (SETQ NILSTART (IDIFFERENCE (fetch (FX NEXTBLOCK) of FUNCTFRAME)
                                                                       (UNFOLD NNILS WORDSPERCELL)))
                                              (UNFOLD NVARS WORDSPERCELL))
                                       WORDSPERQUAD)))

     ;; NT is address of our synthesized nametable: beginning of NIL's, not counting additional PVARs we are about to bind, rounded up to quadword

          (for VAR in VARLST as VALUEOFF from NILSTART by WORDSPERCELL do (\PUTBASEPTR (ADDSTACKBASE VALUEOFF)
                                                                                 0
                                                                                 (\EVAL VAR)))
                                                                      ; then build NT
          (UNINTERRUPTABLY
              (for VAR in VARLST as VAR# from (FOLDLO (IDIFFERENCE NILSTART (fetch (FX FIRSTPVAR) of FUNCTFRAME))
                                                     WORDSPERCELL)
                 as NT1 from (fetch (FNHEADER OVERHEADWORDS) of T) by (CONSTANT (WORDSPERNAMEENTRY)) as NT2
                 from (IPLUS (fetch (FNHEADER OVERHEADWORDS) of T)
                             NTSIZE)
                 by (CONSTANT (WORDSPERNTOFFSETENTRY)) do (SETSTKNAME-RAW NT NT1 (\ATOMVALINDEX VAR))
                                                          (SETSTKNTOFFSET-RAW NT NT2 PVARCODE VAR#))
              (replace (FNHEADER %#FRAMENAME) of NT with '*FUNARG*)
              (replace (FNHEADER NTSIZE) of NT with NTSIZE)
              (replace (FX NAMETABLE) of FUNCTFRAME with NT))
          (RETURN (\MAKESTACKP NIL FUNCTFRAME])

(\MAKEFUNARGFRAME
  [LAMBDA (ENV)                                                       ; Edited 21-Jan-91 18:19 by jds
    (\CALLME 'FUNARG)
    (PROG ((NVARS 0)
           NTSIZE NNILS)
          (for VAR in ENV do                                          ; Count number of vars to bind, check validity
                             (COND
                                ((OR (NULL (\DTEST VAR 'LITATOM))
                                     (EQ VAR T))
                                 (LISPERROR "ATTEMPT TO BIND NIL OR T" VAR)))
                             (add NVARS 1))
          (SETQ ENV (.CALLAFTERPUSHINGNILS. (SETQ NNILS (IPLUS NVARS (SETQ NTSIZE
                                                                        (CEIL [ADD1 (UNFOLD NVARS (CONSTANT
                                                                                                   (
                                                                                            WORDSPERNAMEENTRY
```

```
                                                                                                              ]
                                                                                              WORDSPERQUAD))
                                                                                   (FOLDHI (fetch (FNHEADER OVERHEADWORDS)
                                                                                               of T)
                                                                                          WORDSPERCELL)
                                                                                   (SUB1 CELLSPERQUAD)))
                                        (\FUNCT1 NNILS NVARS NTSIZE ENV)))   ; ENV POINTS TO COPY OF FUNCTION FRAME
              (\SMASHLINK (fetch (STACKP EDFXP) of ENV)
                     0 0)
              (RETURN ENV])

(STKEVAL
  [LAMBDA (POS FORM FLG INTERNALFLG)                                 (* lmm "25-APR-80 00:08")
    (\SMASHLINK NIL (\STACKARGPTR POS))
    (AND FLG (RELSTK POS))
    (\EVAL FORM])

(STKAPPLY
  [LAMBDA (POS FN ARGS FLG)                                          (* lmm "15-Aug-84 17:55")
    (\CALLME '*ENV*)
    (\SMASHLINK NIL (\STACKARGPTR POS))
    (AND FLG (RELSTK POS))
    (.APPLY. FN ARGS])

(RETEVAL
  [LAMBDA (POS FORM FLG INTERNALFLG)                                 (* bvm%: "11-Nov-86 20:53")

    ;; Return from POS with the value of evaluating FORM in the dynamic context of POS

    ;; Anyone know what INTERNALFLG is for?

    (\CALLME '*ENV*)
    (LET ((FX (\STACKARGPTR POS))
          RETURNEE)
         (if (fetch (FX INVALIDP) of (SETQ RETURNEE (fetch (FX CLINK) of FX)))
             then (LISPERROR "ILLEGAL STACK ARG" POS))
         (\SMASHRETURN NIL FX)                                       ; unwind stack back to POS--we need to keep that dynamic
                                                                     ; environment
         (AND FLG (RELSTK POS))
         (CL:MULTIPLE-VALUE-PROG1 (\EVAL FORM)                       ; finally, return from POS
                (SI::UNWIND RETURNEE])

(RETAPPLY
  [LAMBDA (POS FN ARGS FLG)                                          (* bvm%: "11-Nov-86 12:04")

    ;; Return from POS with the value of applying FN to ARGS in the dynamic context of POS

    (\CALLME '*ENV*)
    (LET ((FX (\STACKARGPTR POS))
          RETURNEE)
         (if (fetch (FX INVALIDP) of (SETQ RETURNEE (fetch (FX CLINK) of FX)))
             then (LISPERROR "ILLEGAL STACK ARG" POS))
         (\SMASHRETURN NIL FX)                                       ; unwind stack back to POS--we need to keep that dynamic
                                                                     ; environment
         (AND FLG (RELSTK POS))
         (CL:MULTIPLE-VALUE-PROG1 (.APPLY. FN ARGS)                  ; finally, return from POS
                (SI::UNWIND RETURNEE])
)

(DECLARE%: DONTEVAL@LOAD DOCOPY

(MOVD? 'FUNCTION 'CL:FUNCTION NIL T)
)

;; Blip and other stack funniness

(DEFINEQ

(BLIPVAL
  [LAMBDA (BLIPTYP IPOS FLG)                                         ; Edited 18-Feb-91 16:48 by jds
    (PROG ([FRAME (COND
                     ((NULL IPOS)
                      (\MYALINK))
                     (T (\STACKARGPTR IPOS]
           (A (NEW-SYMBOL-CODE BLIPTYP (\ATOMVALINDEX BLIPTYP)))
           I)
          (SELECTQ BLIPTYP
              ((*TAIL* *FORM* *FN* *ARGVAL*))
              (RETURN (AND (EQ FLG T)
                           0)))
          (RETURN
           (COND
              ((EQ FLG T)                                            ; Count number of blips of type BLIPTYP at FRAME
               (COND
                    ((NOT (SETQ I (\VAROFFSET FRAME A)))
                     0)
                    ((EQ BLIPTYP '*ARGVAL*)                            ; the value of *ARGVAL* is the number of *ARGVAL* blips in this
                                                                       ; frame
                     (OR (\GETBASEPTR \STACKSPACE I)
                         0))
                    (T 1)))
                (T (PROG NIL
                         (OR FLG (SETQ FLG 1))
                     FRAMELP
                         [COND
                            ((SETQ I (\VAROFFSET FRAME A))
                             (SELECTQ BLIPTYP
                                  (*ARGVAL* [COND
                                                ((IGREATERP FLG (SETQ I (OR (\GETBASEPTR \STACKSPACE I)
                                                                            0)))
                                                                       ; Fewer blips here than FLG
                                                 (SETQ FLG (IDIFFERENCE FLG I)))
                                                (T                     ; Scan the temporary region for the value of the FLG'th
                                                                       ; *ARGVAL* blip
                                                   (RETURN (PROG ((NXT (fetch (FX NEXTBLOCK) of FRAME))
                                                                  (P (fetch (FX FIRSTTEMP) of FRAME)))
                                                             LP  (CHECK (ILESSP P NXT))
                                                                 [COND
                                                                    ((EQ (\GETBASEPTR \STACKSPACE P)
                                                                         '*ARGVAL*)

                                                        ;; \EVALFORM pushes the atom *ARGVAL*, then each argument.  We want
                                                        ;; the FLG'th arg, counting from the end backwards

                                                                     (add P (UNFOLD (ADD1 (IDIFFERENCE I FLG))
                                                                                    WORDSPERCELL))
                                                                     (CHECK (ILESSP P NXT))
                                                                     (RETURN (\GETBASEPTR \STACKSPACE P]
                                                                 (add P WORDSPERCELL)
                                                                 (GO LP])
                                       (COND
                                          ((ILESSP (SETQ FLG (SUB1 FLG))
                                                   1)
                                           (RETURN (\GETBASEPTR \STACKSPACE I]
                     NEXT
                         (COND
                            ([NOT (fetch (FX INVALIDP) of (SETQ FRAME (fetch (FX CLINK) of FRAME]
                             (GO FRAMELP])
```

```
(SETBLIPVAL
  [LAMBDA (BLIPTYP IPOS N VAL)                                         ; Edited 18-Feb-91 16:49 by jds
    (PROG ([FRAME (COND
                      ((NULL IPOS)
                       (\MYALINK))
                      (T (\STACKARGPTR IPOS]
           (A (NEW-SYMBOL-CODE BLIPTYP (\ATOMVALINDEX BLIPTYP)))
           I)
          (SELECTQ BLIPTYP
              ((*TAIL* *FORM* *FN* *ARGVAL*))
              (RETURN))
          (COND
             ((NOT N)
              (SETQ N 1))
             ((ILESSP N 1)
              (\ILLEGAL.ARG N)))
      FRAMELP
          [COND
             ((SETQ I (\VAROFFSET FRAME A))
              (SELECTQ BLIPTYP
                   (*ARGVAL* [COND
                                 ((NOT (SETQ I (\GETBASEPTR \STACKSPACE I)))
                                                                       ; No argvals

                                  )
                                 ((IGREATERP N I)
                                  (SETQ N (IDIFFERENCE N I)))
                                 (T                                    ; Scan the temporary region for the value of the Nth *ARGVAL*
                                                                       ; blip
                                    (RETURN (PROG ((NXT (fetch (FX NEXTBLOCK) of FRAME))
                                                   (P (fetch (FX FIRSTTEMP) of FRAME)))
                                              LP  (CHECK (ILESSP P NXT))
                                                  [COND
                                                     ((EQ (\GETBASEPTR \STACKSPACE P)
                                                          '*ARGVAL*)
                                                                       ; \EVALFORM pushes the atom *ARGVAL*, then each
                                                                       ; argument.  We want the N'th arg from the end
                                                      (add P (UNFOLD (ADD1 (IDIFFERENCE I N))
                                                                     WORDSPERCELL))
                                                      (CHECK (ILESSP P NXT))
                                                      (RETURN (\PUTBASEPTR \STACKSPACE P VAL]
                                                  (add P WORDSPERCELL)
```

```
                                                         (GO LP])
                         (COND
                            ((ILESSP (SETQ N (SUB1 N))
                                     1)                                    ; All other blip types are just the value of the blip binding
                             (RETURN (\PUTBASEPTR \STACKSPACE I VAL]
                 (COND
                    ([NOT (fetch (FX INVALIDP) of (SETQ FRAME (fetch (FX CLINK) of FRAME]
                     (GO FRAMELP])

(BLIPSCAN
  [LAMBDA (BLIPTYP IPOS)                                                   ; Edited 18-Feb-91 16:50 by jds
    (PROG ([FRAME (COND
                     ((NULL IPOS)
                      (\MYALINK))
                     (T (\STACKARGPTR IPOS]
           OFF A)
          (SETQ A (SELECTQ BLIPTYP
                      ((*FORM* *TAIL* *FN* *ARGVAL*)
                           (NEW-SYMBOL-CODE BLIPTYP (\ATOMVALINDEX BLIPTYP)))
                      (RETURN)))
      LP  (COND
             ([AND (SETQ OFF (\VAROFFSET FRAME A))
                   (NOT (AND (EQ BLIPTYP '*ARGVAL*)
                             (NULL (GETBASEPTR \STACKSPACE OFF]
              (RETURN (\MAKESTACKP NIL FRAME)))
             ([NOT (fetch (FX INVALIDP) of (SETQ FRAME (fetch (FX CLINK) of FRAME]
              (GO LP))
             (T (RETURN])
)

(DEFINEQ

(\REALFRAMEP
  [LAMBDA (FRAME INTERPFLG)                                                (* lmm " 7-Nov-86 01:53")
    (LET ((NAME (fetch (FNHEADER FRAMENAME) of (fetch (FX FNHEADER) of FRAME)))
          BFLINK)

         ;; note that the selection is on the fnheader's name rather than the nametable name. \REALFRAMEP is thus not affected by SETSTKNAME

         (AND (CL:SYMBOLP NAME)
              (SELECTQ NAME
                  (*ENV*                                                   ; *ENV* is used by ENVEVAL etc.
                         NIL)
                  (\INTERPRETER T)
                  (ERRORSET (NEQ (\STKARG 2 FRAME)
                                 'INTERNAL))
                  ((EVAL APPLY)
                       (\SMASHLINK NIL FRAME)
                       (SELECTQ \INTERNAL
                           ((INTERNAL SELECTQ)
                                NIL)
                           T))
                  (OR (NOT (LITATOM NAME))
                      (COND
                         ((FMEMB NAME OPENFNS)
                          INTERPFLG)
                         (T (OR (NEQ (CHCON1 NAME)
                                     (CHARCODE \))
                                (EXPRP NAME)
                                (FASSOC NAME BRKINFOLST])
)

(RPAQ? OPENFNS '(APPLY* SETQ AND OR COND SELECTQ PROG PROGN PROG1 ARG SETARG ERSETQ NLSETQ RESETFORM RESETLST
                       RESETVARS RPTQ SAVESETQ SETN UNDONLSETQ XNLSETQ))

(RPAQQ \BLIPNAMES (*TAIL* *FORM* *FN* *ARGVALS*))

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS BRKINFOLST)
)

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS \BLIPNAMES OPENFNS)
)

(DEFINEQ

(RAIDCOMMAND
  [LAMBDA NIL                                                              (* lmm "18-Mar-86 09:52")
    (DECLARE (USEDFREE ROOTFRAME ALINKS? RAIDIX FRAME# VPRINTLEVEL))
    (FRESHLINE T)
    (PROG (CMD)
          (SELECTQ (ALLOCAL (SETQ CMD (ASKUSER NIL NIL "@" '((Q "uit [confirm]" CONFIRMFLG T)
(%Ï "^N - remote return [confirm]" NOECHOFLG T
   CONFIRMFLG T RETURN '^N)
(L "isp stack ")
(%
 "Lisp stack " NOECHOFLG T EXPLAINSTRING "^L -- Lisp stack from arbitrary frame or context" RETURN '^L)
                                                                            (F "rame ")
                                                                            (%
 "Next frame " EXPLAINSTRING "LF - next frame" RETURN 'LF)
                                                                            (^ " Previous frame ")
                                                                            (A "tom top-level value of atom: ")
                                                                            (D "efinition for atom: ")
                                                                            (P "roperty list for atom: ")
                                                                            (V " -- show object at Virtual address: ")
                                                                            (B "lock of storage starting at address: ")
                                                                            (S "how raw stack from address: ")
                                                                            (C "ode for function:")
                                                                            (%Ã "Basic frame at: " EXPLAINSTRING "^F - print
                                                                               basic frame at octal address" RETURN
                                                                               '^F)
                                                                            (%Û "frame extension at: " EXPLAINSTRING "^X -
                                                                               print frame extension at octal address" RETURN
                                                                               '^X)
                                                                            (W "alk stack blocks starting at: ")
                                                                            (K "" EXPLAINSTRING "K -- Set linKtype for stack
                                                                               ops")
                                                                            (_ " Set word at address: ")
                                                                            (%- " Set value of atom " EXPLAINSTRING "^V -- Set
                                                                                value of atom" RETURN '^V)
                                                                            (%Ì "atom number for atom: " EXPLAINSTRING "^O -
                                                                               look up atom" RETURN '^O)
                                                                            (Z "Zap Print level to: ")
                                                                            (I "nspect InterfacePage [confirm]" CONFIRMFLG T)
                                                                            (U " -- Show remote screen [confirm]" CONFIRMFLG T
                                                                               )
                                                                            ("
                                                                             " "" RETURN NIL)
                                                                            (%Ü " Enter Lisp " EXPLAINSTRING "^Y -- Enter
                                                                               Lisp" RETURN '^Y))
                                                         T)))
                  (^N (RETURN 'RETURN))
                  (Q (TERPRI T)
                     (RETURN 'QUIT))
                  (NIL)
                  (A (PRINCOPY (GETTOPVAL (READATOM))))
                  (P (PRINCOPY (GETPROPLIST (READATOM))))
                  (C (PRINTCODE (READATOM)
                            T RAIDIX))
                  (V (PRINCOPY (READVA)))
                  (B (PRINTADDRS (READVA)
                            (READOCT " for (number of words): ")))
                  (S (PRINTADDRS (ADDSTACKBASE (READOCT))
                            (READOCT " for (number of words): ")))
                  (D (PRINTADDRS (fetch (LITATOM DEFINITIONCELL) of (READATOM))
                            2))
                  (^O (PRINTNUM .I2 (\ATOMVALINDEX (READATOM))
                             T))
                  (^V (PROG ((ATM (READATOM)))
                             (printout T " to be ")
                             (SETTOPVAL ATM (READ T T))))
                  ((L ^L)
                       (RAIDSTACKCMD CMD))
                  (F [RAIDSHOWFRAME (SETQ FRAME# (PROG1 (READ T T)
                                                        (READC T])
                  (LF (OR FRAME# (SETQ FRAME# 0))
                      (printout T "(" .I1 (add FRAME# 1)
                             ")" T)
                      (RAIDSHOWFRAME FRAME#))
                  (^ (COND
                        ((OR (NULL FRAME#)
                              (ILEQ FRAME# 1))
                         (printout T "No previous frame" T))
                        (T (printout T "(" .I1 (add FRAME# -1)
                                  ")" T)
                           (RAIDSHOWFRAME FRAME#))))
                  (^F (\PRINTBF (READOCT)
                             NIL
                             (FUNCTION PRINCOPY)))
                  (Z [ALLOCAL (LET [(A (PROG1 (READ T T)
                                              (READC T)))
                                    (D (PROG1 (READ T T)
                                              (READC T]
                                   (COND
                                      ((AND (FIXP A)
                                             (FIXP D))
                                       (SETQ VPRINTLEVEL (CONS A D)))
                                      (T (PRINTOUT T "Must be two integers, car level then cdr level" T)
                                         (ERROR!])
                  (W [SHOWSTACKBLOCKS (COND
                                         ((EQ (PEEKC T)
                                              '%
)
```

```
                                                (READC T)
                                                (fetch (IFPAGE StackBase) of \InterfacePage))
                                               (T (READOCT])
                 (^X (\PRINTFRAME (READOCT)
                              'PRINCOPY))
                 (^Y (TERPRI T)
                     (USEREXEC ':%:))
                 (K (SETQ ALINKS? (EQ (ASKUSER NIL NIL " Set link type for stack operations to "
                                                 '((A "links
                                                       ")
                                                   (C "links
                                                       "))
                                                 T)
                                           'A)))
                 (_ (PROG ((VA (READVA)))
                          (printout T " Currently ")
                          (PRINTNUM .I7 (GETBASE VA 0)
                                 T)
                          (printout T " to be ")
                          (PUTBASE VA 0 (READOCT))))
                 (I [ALLOCAL (COND
                                  [(NULL (GETD 'INSPECT]
                                  ((RECLOOK 'IFPAGE)
                                   (INSPECT [COND
                                                ((LISTP VMEMFILE)
                                                 (VMAPPAGE (fetch (POINTER PAGE#) of \InterfacePage)))
                                                (T (PROG [(PAGE (NCREATE 'VMEMPAGEP]
                                                         (SETVMPTR (VGETTOPVAL '\InterfacePage))
                                                         (\BINS (GETSTREAM VMEMFILE)
                                                                PAGE 0 BYTESPERPAGE)
                                                         (RETURN PAGE]
                                            'IFPAGE))
                                  (T (PRIN1 " Can't -- no record for IFPAGE"]
                    (TERPRI T))
                 (U (SHOWREMOTESCREEN))
                 (HELP))
             (RETURN NIL])
```

(**RAIDSHOWFRAME**

```
  [LAMBDA (N)                                                     (* bvm%: "27-Jan-85 15:27")
    (PROG [(FRAME (OR ROOTFRAME (RAIDROOTFRAME]
          [FRPTQ (SUB1 N)
                 (COND
                    ([fetch (FX INVALIDP) of (SETQ FRAME (COND
                                                             (ALINKS? (fetch (FX ALINK) of FRAME))
                                                             (T (fetch (FX CLINK) of FRAME]
                     (RETURN (printout T N " is beyond the bottom of the stack" T]
          (\BACKTRACE FRAME FRAME T NIL T T NIL (FUNCTION PRINCOPY)
                 NIL RAIDIX])
```

(**RAIDSTACKCMD**

```
  [LAMBDA (CMD)                                                   (* bvm%: "28-Jan-85 12:16")
    (DECLARE (USEDFREE FRAME# ROOTFRAME))
    (PROG (FRAME)
          (SETQ FRAME# 0)
          [COND
             ((EQ CMD 'L)
              (RAIDROOTFRAME))
             (T (SETQ ROOTFRAME (SELECTQ (SETQ FRAME (ASKUSER NIL NIL "in context (? for help): "
                                                                '((P "age fault")
                                                                  (G "arbage collection")
                                                                  (K "eyboard handler")
                                                                  (H "ard Return")
                                                                  (S "tack manipulator")
                                                                  (R "eset")
                                                                  (M "iscellaneous")
                                                                  (F "rame at location: "))
                                                                T))
                                         (P (fetch (IFPAGE FAULTFXP) of \InterfacePage))
                                         (G (fetch (IFPAGE GCFXP) of \InterfacePage))
                                         (K (fetch (IFPAGE KbdFXP) of \InterfacePage))
                                         (H (fetch (IFPAGE HardReturnFXP) of \InterfacePage))
                                         (S (fetch (IFPAGE SubovFXP) of \InterfacePage))
                                         (R (fetch (IFPAGE ResetFXP) of \InterfacePage))
                                         (M (fetch (IFPAGE MiscFXP) of \InterfacePage))
                                         (COND
                                            ((AND (ILESSP (SETQ FRAME (READOCT))
                                                          WORDSPERPAGE)
                                                  (ILESSP (\GETBASE \InterfacePage FRAME)
                                                          (fetch (IFPAGE EndOfStack) of \InterfacePage))
                                                  (type? FX (\GETBASE \InterfacePage FRAME)))
                                             (\GETBASE \InterfacePage FRAME))
                                            ((type? FX FRAME)
                                             FRAME)
                                            (T (PRINTNUM .I7 FRAME)
                                              (printout T " not a valid frame." T)
                                              (RETURN]
          (FRESHLINE T)
          (\BACKTRACE ROOTFRAME NIL T NIL NIL NIL ALINKS? (FUNCTION PRINCOPY)
                1 RAIDIX])
```

(RAIDROOTFRAME

```
  [LAMBDA NIL                                                       (* bvm%: "27-Jan-85 15:26")
    (SETQ ROOTFRAME (PROG1 (COND
                             ((ALLOCAL (LISTP VMEMFILE))
                              (PRIN1 "in TeleRaid Context" T)
                              (fetch (IFPAGE TELERAIDFXP) of \InterfacePage))
                             (T (fetch (IFPAGE CurrentFXP) of \InterfacePage)))
                         (TERPRI T])
```

(PRINTADDRS

```
  [LAMBDA (BASE CNT)                                                (* bvm%: "13-Feb-85 22:42")
    (PRIN1 "words from ")
    (PRINTVA BASE)
    (PRIN1 " to ")
    (PRINTVA (\ADDBASE BASE (SUB1 CNT)))
    (TERPRI)
    (SPACES 7)
    (for I from 0 to 7 do (PRINTNUM .I7 I))
    (PROG ((NB (\VAG2 (\HILOC BASE)
                      (FLOOR (\LOLOC BASE)
                             8)))
           (LB (\ADDBASE BASE CNT)))
          (do (COND
                 ((EVENP (\LOLOC NB)
                         8)
                  (TAB 0 0)
                  (PRINTNUM .I5 (\LOLOC NB))
                  (PRIN1 ": ")))
              [COND
                 ((PTRGTP BASE NB)
                  (SPACES 7))
                 (T (PRINTNUM .I7 (\GETBASE NB 0]
              (SETQ NB (\ADDBASE NB 1)) repeatwhile (PTRGTP LB NB))
          (TAB 0 0])
```

(PRINTVA

```
  [LAMBDA (X)                                                       (* bvm%: "12-Feb-85 10:41")
    (PRIN1 "{")
    (PRINTNUM .I2 (HILOC X))
    (PRIN1 ",")
    (PRINTNUM .I2 (LOLOC X))
    (PRIN1 "}"])
```

(READVA

```
  [LAMBDA NIL                                                       (* lmm "21-AUG-81 12:55")
    (VAG2 (READOCT)
          (READOCT])
```

(READATOM

```
  [LAMBDA NIL                                                       ; Edited 16-Feb-87 15:44 by raf
    (ALLOCAL (PROG1 (HANDLER-BIND [[XCL:MISSING-EXTERNAL-SYMBOL #'(LAMBDA (CONDITION)

                                                                           ;; MAKE AN INTERNAL SYMBOL INSTEAD

                                                                           (CL:INTERN (
                                                                                 XCL:MISSING-EXTERNAL-SYMBOL-NAME
                                                                                  CONDITION)
                                                                                  (XCL:MISSING-EXTERNAL-SYMBOL-PACKAGE
                                                                                   CONDITION]
                                          (XCL:MISSING-PACKAGE #'(LAMBDA (CONDITION)

                                                                  ;; FAKE A PACKAGE BY THIS NAME AND MAKE THE SYMBOL IN
                                                                  ;; IT

                                                                  (CL:INTERN (XCL:MISSING-PACKAGE-SYMBOL-NAME
                                                                                  CONDITION)
                                                                         (CL:MAKE-PACKAGE (
                                                                                 XCL:MISSING-PACKAGE-PACKAGE-NAME
                                                                                        CONDITION)
                                                                                :USE NIL]
                                  (CL:READ T))
                    (READC T])
```

(READOCT

```
  [LAMBDA (PROMPT)                                                  (* bvm%: "28-Jan-85 11:51")
    (DECLARE (USEDFREE RAIDIX))
    (COND
        ((AND PROMPT (NOT (READP T)))
         (printout T PROMPT)))
    (bind STR while (EQUAL (SETQ STR (RSTRING T T))
                           "")
       do (READC T)
       finally (RETURN (PROG1 (OR (FIXP (SELECTQ RAIDIX
                                            (8 (MKATOM (CONCAT STR "Q")))
                                            (16 (bind (N _ 0)
                                                       CHAR while (SETQ CHAR (GNC STR))
                                                   do [SETQ N (IPLUS (ITIMES N 16)
                                                                     (COND
                                                                        ((FIXP CHAR)
                                                                         CHAR)
                                                                        ((AND (IGEQ (SETQ CHAR (CHCON1 CHAR))
                                                                                    (CHARCODE A))
                                                                              (ILEQ CHAR (CHARCODE F)))
                                                                         (IPLUS (IDIFFERENCE CHAR (CHARCODE
                                                                                                   A))
                                                                                10))
                                                                        (T (ERROR CHAR '? T]
                                                   finally (RETURN N)))
                                            (SHOULDNT)))
                                  (PROGN (PRIN1 "?" T)
                                         (ERROR!)))
                              (READC T])
```

(SHOWSTACKBLOCKS

```
  [LAMBDA (SCANPTR WAITFLG)                                    (* bvm%: "18-AUG-83 12:05")
                                                               ; show stack
    (PROG ((EASP (fetch EndOfStack of \InterfacePage)))
       SCAN
           [SELECTC (fetch (STK FLAGS) of SCANPTR)
                (\STK.FSB (SHOWSTACKBLOCK1 SCANPTR "free block" (fetch (FSB CHECKED) of SCANPTR))
                          (add SCANPTR (fetch (FSB SIZE) of SCANPTR)))
                (\STK.GUARD (SHOWSTACKBLOCK1 SCANPTR "guard block" T)
                            (add SCANPTR (fetch (FSB SIZE) of SCANPTR)))
                (\STK.FX                                       ; frame extension
                         (SHOWSTACKBLOCK1 SCANPTR "Frame extn = " (fetch (FX CHECKED) of SCANPTR))
                         (PRIN2 (\UNCOPY (fetch (FX FRAMENAME) of SCANPTR)))
                         (SETQ SCANPTR (fetch (FX NEXTBLOCK) of SCANPTR)))
                (PROG ((ORIG SCANPTR)
                       IVAR)                                   ; must be a basic frame
                      (while (EQ (fetch (STK FLAGS) of SCANPTR)
                                 \STK.NOTFLAG)
                         do (add SCANPTR WORDSPERCELL))
                      (COND
                         ((NOT (type? BF SCANPTR))
                          (SHOWSTACKBLOCK1 ORIG "Garbage" T))
                         (T (SETQ IVAR (fetch (BF IVAR) of SCANPTR))
                            [COND
                               ((fetch (BF RESIDUAL) of SCANPTR)
                                (SHOWSTACKBLOCK1 SCANPTR "Residual BF" (EQ SCANPTR ORIG))
                                (PRIN1 " with IVar = ")
                                (PRINTNUM .I7 IVAR))
                               (T (SHOWSTACKBLOCK1 SCANPTR "Basic frame" (AND (EQ ORIG IVAR)
                                                                              (fetch (BF CHECKED)
                                                                                 of SCANPTR]
                            (add SCANPTR WORDSPERCELL]
           (TERPRI)
           (COND
              ((IGREATERP SCANPTR EASP)
               (RETURN)))
           (AND WAITFLG (READC T))
           (GO SCAN])
```

(SHOWSTACKBLOCK1

```
  [LAMBDA (PTR STR GOODFLG)                                    (* bvm%: " 6-AUG-83 23:59")
    (PRINTNUM .I7 PTR)
    (SPACES 1)
    (OR GOODFLG (PRIN1 "[bad] "))
    (PRIN1 STR])
```

(PRINCOPY

```
  [LAMBDA (X)                                                  (* bvm%: "24-Jan-86 12:33")
    (PRINT (\UNCOPY X (LOCAL (CAR VPRINTLEVEL))
                   (LOCAL (CDR VPRINTLEVEL)))
           T T])
```

(NOSUCHATOM

```
  [LAMBDA (ATM)                                                ; Edited  9-Nov-92 15:24 by sybalsky:mv:envos
```

;; Called only under TeleRaid when V\MKATOM fails to find atom ATM. JDS: And in MAKEINIT, ditto. For Makeinit (since Teleraid is essentially
;; dead), changed ERROR! to ERROR.

```
    (printout T "No such atom: " ATM T)
    (ERROR "No such atom: "])

)

(DEFINEQ

(BACKTRACE
  [LAMBDA (IPOS EPOS FLAGS FILE PRINTFN)                                  (* bvm%: "13-Feb-85 22:42")
    (RESETFORM (OUTPUT FILE)
           (\BACKTRACE (\STACKARGPTR (OR IPOS -1))
                  (\STACKARGPTR (OR EPOS T))
                  [EQ 0 (LOGAND 8 (OR FLAGS (SETQ FLAGS 0]
                  (NEQ 0 (LOGAND FLAGS 1))
                  (NEQ 0 (LOGAND FLAGS 4))
                  (NEQ 0 (LOGAND FLAGS 32))
                  (EQ 0 (LOGAND FLAGS 16))
                  (OR PRINTFN (FUNCTION PRINT))
                  NIL])

(\BACKTRACE
  [LAMBDA (IPOS EPOS NAMES VARS LOCALS JUNK ALINKS PRINTFN CNT RADIX)
                                                                          (* lmm " 2-Jul-86 13:00")
    (OR RADIX (SETQ RADIX 8))
    (PROG [NARGS NPVARS NAME ARGNAME BLINK (.I7 (NUMFORMATCODE (LIST 'FIX 7 RADIX]
          (DECLARE (SPECVARS .I7))
      POSLP
          (COND
             (CNT (printout NIL .I3 CNT ": ")
                  (add CNT 1)))
          (SETQ NAME (\STKNAME IPOS))
          (COND
             (JUNK (TERPRI)
                   (TERPRI)
                   (PRIN1 "Basic frame at ")
                   (PRINTNUM .I7 (SETQ BLINK (fetch (FX BLINK) of IPOS)))
                   (TERPRI)
                   (\PRINTBF BLINK (fetch (FX NAMETABLE) of IPOS)
                          PRINTFN)
                   (PROGN (TERPRI)
                          (PRIN1 "Frame xtn at ")
                          (PRINTNUM .I7 IPOS)
                          (PRIN1 ", frame name= "))
                   (APPLY* PRINTFN NAME)
                   (\PRINTFRAME IPOS PRINTFN))
             [(OR VARS LOCALS)
              (\PRINTBF (fetch (FX BLINK) of IPOS)
                     (fetch (FX NAMETABLE) of IPOS)
                     PRINTFN
                     (COND
                        (LOCALS 'LOCALS)
                        (T T)))
              (COND
                 (NAMES (APPLY* PRINTFN NAME)
                        (TERPRI)))
              (\PRINTFRAME IPOS PRINTFN (COND
                                           (LOCALS 'LOCALS)
                                           (T T]
             (NAMES (APPLY* PRINTFN NAME)))
          (COND
             ([AND (NEQ EPOS IPOS)
                   (NOT (fetch (FX INVALIDP) of (SETQ IPOS (COND
                                                             (ALINKS (fetch (FX ALINK) of IPOS))
                                                             (T (fetch (FX CLINK) of IPOS]
              (GO POSLP)))
          (RETURN T])

(\SCANFORNTENTRY
  [LAMBDA (NMT NTENTRY)                                                   ; Edited 18-Feb-91 15:18 by jds

    ;; Scan thru the name table pointedto by NMT, looking for the name of the variable with offset entry NTENTRY, which must be in the form
    ;; appropriate to the architecture (2- or 3-byte).

    (bind NM for NT1 from (fetch (FNHEADER OVERHEADWORDS) of T) by (CONSTANT (WORDSPERNAMEENTRY)) as NT2
       from (IPLUS (fetch (FNHEADER OVERHEADWORDS) of T)
                   (fetch (FNHEADER NTSIZE) of NMT))
       by (CONSTANT (WORDSPERNTOFFSETENTRY)) do (COND
                                                   ((NULL-NTENTRY (SETQ NM (GETSTKNAMEENTRY NMT NT1)))
                                                    (RETURN)))
                                                (COND
                                                   ((IEQP NTENTRY (GETSTKNTOFFSETENTRY NMT NT2))
                                                    (RETURN (\INDEXATOMVAL NM])

(\PRINTSTK
  [LAMBDA (I)                                                             (* lmm "23-MAY-82 22:09")
```

```
    (PRINTNUM .I7 I)
    (PRIN1 ": ")
    (PRINTNUM .I7 (GETBASE \STACKSPACE I))
    (PRINTNUM .I7 (GETBASE \STACKSPACE (ADD1 I)))
    (SPACES 1])


(\PRINTFRAME
  [LAMBDA (FRAME PRINTFN VARSONLY)                          ; Edited 30-Jan-91 01:48 by jds
    (PROG ((NMT (fetch (FX NAMETABLE) of FRAME))
           (I 0)
           (FT (fetch (FX FIRSTTEMP) of FRAME))
           TMP NLOCALS)
          [COND
             ((NOT VARSONLY)
              (\PRINTSTK FRAME)
              (PRIN1 "[")
              (PROGN (PSTKFLD FAST "F, " FAST)
                     (PSTKFLD INCALL "C, " INCALL)
                     (PSTKFLD VALIDNAMETABLE "V, " VALIDNAMETABLE)
                     (PSTKFLD NOPUSH "N, " NOPUSH)
                     (PSTKFLD USECNT "USE=" (NEQ USECNT 0)
                            NIL ", ")
                     (PSTKFLD SLOWP "X, " SLOWP)
                     (PSTKFLD ALINK " alink]" T))
              (TERPRI)
              (PSTK 2 (FNHEADER "[fn header]" T))
              (PSTK 4 (NEXTBLOCK "[next, pc]" T))
              (PSTK 6 (NAMETABLE "[nametable]" T))
              (PSTK 8 (BLINK "[blink, clink]" T]
          (SETQ NLOCALS (fetch (FNHEADER NLOCALS) of NMT))
          [for old I from (fetch (FX FIRSTPVAR) of FRAME) by WORDSPERCELL while (ILESSP I FT) as J from 0
             do (OR VARSONLY (\PRINTSTK I))
                (COND
                   [(ILESSP J NLOCALS)
                    (COND
                       ((OR (SETQ TMP (\SCANFORNTENTRY NMT (MAKE-NTENTRY PVARCODE J)))
                            (AND (NEQ VARSONLY T)
                                 (SETQ TMP "local")))
                        (COND
                           ((fetch (PVARSLOT BOUND) of (ADDSTACKBASE I))
                            (AND VARSONLY (SPACES 3))
                            (PRIN2 TMP)
                            (SPACES 1)
                            (APPLY* PRINTFN (\GETBASEPTR (ADDSTACKBASE I)
                                                   0)))
                           ((NOT VARSONLY)
                            (printout NIL TMP " [unbound]" T]
                   ((NOT VARSONLY)
                    (COND
                       ((SETQ TMP (\SCANFORNTENTRY NMT (MAKE-NTENTRY FVARCODE J)))
                        (printout NIL "[fvar " .P2 TMP " " (COND
                                                               ((fetch (FVARSLOT LOOKEDUP) of (ADDSTACKBASE I))
                                                                (COND
                                                                   ((EQ [SETQ TMP (\HILOC (fetch (FVARSLOT
                                                                                                  BINDINGPTR
                                                                                                   )
                                                                                             of (ADDSTACKBASE
                                                                                                    I]
                                                                        \STACKHI)
                                                                    " on stack]")
                                                                   ((NEQ (FLOOR TMP 2)
                                                                         (\HILOC \VALSPACE))
                                                            ; See comment in BOUNDP
                                                                    " non-stack binding]")
                                                                   (T " top value]")))
                                                               (T " not looked up]"))
                               T))
                       (T (printout NIL "[padding]" T]
          (COND
             ((NOT VARSONLY)
              (SETQ FT (fetch (FX NEXTBLOCK) of FRAME))
              (for old I by 2 while (ILESSP I FT) do        ; 2 = WORDSPERCELL but for doesn't translate correctly with
                                                             ; WORDSPERCELL
                                                 (\PRINTSTK I)
                                                 (COND
                                                    ((fetch (PVARSLOT BOUND) of (ADDSTACKBASE I))
                                                     (APPLY* PRINTFN (\GETBASEPTR (ADDSTACKBASE I)
                                                                            0)))
                                                    (T (TERPRI])


(\PRINTBF
  [LAMBDA (BL NMT PRINTFN VARSONLY)                         ; Edited 30-Jan-91 01:49 by jds
    [bind NM for I from (fetch (BF IVAR) of BL) by 2 as J from 0 to (SUB1 (fetch (BF NARGS) of BL))
       do (OR VARSONLY (\PRINTSTK I))
          [COND
```

```
              ([OR (SETQ NM (\SCANFORNTENTRY [OR NMT (RETURN (OR VARSONLY (TERPRI]
                                       (MAKE-NTENTRY IVARCODE J)))
                   (AND (NEQ VARSONLY T)
                        (SETQ NM '*local*]
               (AND VARSONLY (SPACES 3))
               (PRIN2 NM)
               (SPACES 1)
               (APPLY* PRINTFN (GETBASEPTR \STACKSPACE I]
        finally (OR VARSONLY (while (ILESSP I BL) do (\PRINTSTK I)
                                                     (printout NIL "[padding]" T)
                                                     (add I 2]
     (COND
        ((NOT VARSONLY)
         (\PRINTSTK BL)
         (COND
            ((fetch (BF RESIDUAL) of BL)
             (PRIN1 "residual ")))
         (COND
            ((NEQ (fetch (BF USECNT) of BL)
                  0)
             (printout NIL "usecnt= " (fetch (BF USECNT) of BL)
                   %,)))
         (TERPRI])
)

(DECLARE%: EVAL@COMPILE DONTCOPY

(RPAQQ RAIDCOMS
       ((MACROS PSTKFLD PRINTSTKFIELDS PSTK PRINTVA)
        (ADDVARS (RDCOMS (FNS RAIDCOMMAND RAIDSHOWFRAME RAIDSTACKCMD RAIDROOTFRAME PRINTADDRS PRINTVA READVA
                              READOCT READATOM SHOWSTACKBLOCKS SHOWSTACKBLOCK1 PRINCOPY NOSUCHATOM)
                         (FNS \BACKTRACE \STKNAME \PRINTBF \PRINTFRAME \SCANFORNTENTRY \PRINTSTK))
               (EXPANDMACROFNS PSTKFLD PRINTSTKFIELDS PSTK PRINTVA))
        (ADDVARS (DONTCOMPILEFNS RAIDCOMMAND RAIDSHOWFRAME RAIDSTACKCMD RAIDROOTFRAME PRINTADDRS PRINTVA READVA
                         READATOM READOCT SHOWSTACKBLOCKS SHOWSTACKBLOCK1 PRINCOPY NOSUCHATOM))))

(DECLARE%: EVAL@COMPILE

(PUTPROPS PSTKFLD MACRO [(FLD STR TEST FMT STR2)
                          (PROG ((FLD (fetch (FX FLD) of FRAME)))
                                (DECLARE (LOCALVARS FLD))
                                (COND
                                   (TEST (PRIN1 'STR)
                                         (SELECTQ (CONSTANT (NTHCHAR 'STR -1))
                                              (= (printout NIL %, FLD STR2))
                                              NIL)
                                         T])

(PUTPROPS PRINTSTKFIELDS MACRO [FIELDS (CONS 'PROGN (MAPCAR FIELDS (FUNCTION (LAMBDA (X)
                                                                                (CONS 'PSTKFLD X])

(PUTPROPS PSTK MACRO ((N . FIELDS)
                      (\PRINTSTK (IPLUS FRAME N))
                      (PRINTSTKFIELDS . FIELDS)
                      (TERPRI)))

(PUTPROPS PRINTVA MACRO [LAMBDA (X)
                          (printout NIL "{" (HILOC X)
                                 ","
                                 (LOLOC X)
                                 "}"])
)

(ADDTOVAR RDCOMS (FNS RAIDCOMMAND RAIDSHOWFRAME RAIDSTACKCMD RAIDROOTFRAME PRINTADDRS PRINTVA READVA READOCT
                      READATOM SHOWSTACKBLOCKS SHOWSTACKBLOCK1 PRINCOPY NOSUCHATOM)
                 (FNS \BACKTRACE \STKNAME \PRINTBF \PRINTFRAME \SCANFORNTENTRY \PRINTSTK))

(ADDTOVAR EXPANDMACROFNS PSTKFLD PRINTSTKFIELDS PSTK PRINTVA)

(ADDTOVAR DONTCOMPILEFNS RAIDCOMMAND RAIDSHOWFRAME RAIDSTACKCMD RAIDROOTFRAME PRINTADDRS PRINTVA READVA
                         READATOM READOCT SHOWSTACKBLOCKS SHOWSTACKBLOCK1 PRINCOPY NOSUCHATOM)
)

(DEFINEQ

(CCODEP
  [LAMBDA (FN)                                               ; Edited 30-Jan-87 13:36 by Pavel
     (COND
        [(LITATOM FN)
         (COND
            ((fetch (LITATOM CCODEP) of FN)
             (NOT (fetch (LITATOM PSEUDOCODEP) of FN)))
            (T (TYPEP (fetch (LITATOM DEFPOINTER) of FN)
                      'COMPILED-CLOSURE]
        (T (CL:COMPILED-FUNCTION-P FN])
(EXPRP
  [LAMBDA (FN)                                                    (* lmm "17-FEB-82 23:50")
    (PROG ((DEF FN))
          [COND
             ((LITATOM DEF)
              [COND
                 ((fetch (LITATOM CCODEP) of DEF)
                  (RETURN (fetch (LITATOM PSEUDOCODEP) of DEF]
              (SETQ DEF (fetch (LITATOM DEFPOINTER) of DEF]
          (RETURN (COND
                     ((LISTP DEF)
                      T])


(SUBRP
  [LAMBDA (FN)                                                    (* lmm "17-AUG-81 21:57")
    NIL])


(FNTYP
  [LAMBDA (FN)                                                    (* bvm%: " 7-Jul-86 16:43")
    (PROG ((DEF FN))
          [COND
             ((LITATOM DEF)
              (SETQ DEF (fetch (LITATOM DEFINITIONCELL) of DEF))
              (COND
                 ((fetch (DEFINITIONCELL PSEUDOCODEP) of DEF)
                  (SETQ DEF (\PSEUDOCODE.REALDEF DEF)))
                 ((PROG1 (fetch (DEFINITIONCELL CCODEP) of DEF)
                      (SETQ DEF (fetch (DEFINITIONCELL DEFPOINTER) of DEF)))
                  (RETURN (\CCODEFNTYP DEF]
          (RETURN (COND
                     ((LISTP DEF)
                      (SELECTQ (CAR DEF)
                           (CL:LAMBDA 'EXPR*)
                           ([LAMBDA OPENLAMBDA]
                                (COND
                                   ((AND (NLISTP (SETQ DEF (CADR DEF)))
                                         DEF)
                                    'EXPR*)
                                   (T 'EXPR)))
                           (NLAMBDA (COND
                                       ((AND (NLISTP (SETQ DEF (CADR DEF)))
                                             DEF)
                                        'FEXPR*)
                                       (T 'FEXPR)))
                           (FUNARG 'EXPR)
                          NIL))
                     ((TYPEP DEF 'COMPILED-CLOSURE)
                      (\CCODEFNTYP (fetch (COMPILED-CLOSURE FNHEADER) of DEF])


(ARGTYPE
  [LAMBDA (FN)                                                    (* bvm%: "16-Jul-86 22:47")
    (LET ((DEF FN))
         (CL:TYPECASE DEF
             (CL:SYMBOL (COND
                           ((PROG1 (fetch (LITATOM CCODEP) of DEF)
                                (SETQ DEF (fetch (LITATOM DEFPOINTER) of DEF)))
                            (\CCODEARGTYPE DEF))
                           (DEF (ARGTYPE DEF))))
             (CONS (SELECTQ (CAR DEF)
                        (CL:LAMBDA 2)
                        ([LAMBDA OPENLAMBDA]
                             (COND
                                ((AND (NLISTP (SETQ DEF (CADR DEF)))
                                      DEF)
                                 2)
                                (T 0)))
                        (NLAMBDA (COND
                                    ((AND (NLISTP (SETQ DEF (CADR DEF)))
                                          DEF)
                                     3)
                                    (T 1)))
                        (FUNARG (ARGTYPE (CADR DEF)))
                        (SELECTQ (FNTYP DEF)
                             (EXPR 0)
                             (FEXPR 1)
                             (EXPR* 2)
                             (FEXPR* 3)
                            NIL)))
             (CLOSURE 2)
             (COMPILED-CLOSURE (\CCODEARGTYPE (fetch (COMPILED-CLOSURE FNHEADER) of DEF))))])


(NARGS
  [LAMBDA (FN)                                                          (* bvm%: " 7-Jul-86 17:07")
    (LET ((DEF FN))
         (COND
            ([AND (LITATOM DEF)
                  (PROG1 (fetch (LITATOM CCODEP) of DEF)
                         (SETQ DEF (fetch (LITATOM DEFPOINTER) of DEF)))]
             (\CCODENARGS DEF))
            ((LISTP DEF)
             (SELECTQ (CAR DEF)
                 (CL:LAMBDA 1)
                 ([LAMBDA NLAMBDA OPENLAMBDA]
                      (COND
                         ((NULL (SETQ DEF (CADR DEF)))
                          0)
                         ((NLISTP DEF)
                          1)
                         (T (in DEF sum 1))))
                 (FUNARG (NARGS (CADR DEF)))
                 NIL))
            ((TYPEP DEF 'COMPILED-CLOSURE)
             (\CCODENARGS (fetch (COMPILED-CLOSURE FNHEADER) of DEF])

(ARGLIST
  [LAMBDA (FN SMARTP)                                                   ; Edited 15-Jan-88 15:10 by bvm:
    (PROG ((DEF FN)
           TEMP)
          [COND
             ((LITATOM DEF)
              (COND
                 ((PROG1 (fetch (LITATOM CCODEP) of DEF)
                         (SETQ DEF (fetch (LITATOM DEFPOINTER) of DEF)))
                  (RETURN (\CCODEARGLIST DEF SMARTP)))
                 ((NULL DEF)
                  (SETQ DEF (GETPROP FN 'EXPR]
          [RETURN (COND
                     ((LISTP DEF)
                      (SELECTQ (CAR DEF)
                          (CL:LAMBDA 'U)
                          ([LAMBDA NLAMBDA OPENLAMBDA]
                               (CADR DEF))
                          (FUNARG (ARGLIST (CADR DEF)))
                          (GO UNDEF)))
                     ((TYPEP DEF 'COMPILED-CLOSURE)
                      (\CCODEARGLIST (fetch (COMPILED-CLOSURE FNHEADER) of DEF)
                             SMARTP))
                     (T (GO UNDEF]
      UNDEF
          (COND
             ((AND (SETQ DEF (FNCHECK FN T))
                   (NEQ DEF FN))
              (RETURN (ARGLIST DEF)))
             (T (ERROR '"Args not available:" FN])

(\CCODEARGLIST
  [LAMBDA (FNHD SMARTP)                                                 ; Edited 10-May-88 12:18 by MASINTER

    ;; Computes the arglist for raw code object FNHD.  If SMARTP is true, we're allowed to return a Common Lisp arg list if we find one; otherwise, we
    ;; have to comply with Interlisp arglist semantics.

    (PROG ((N (fetch (FNHEADER NA) of FNHD))
           IVARS SIZE LOCALSIZE ENDT)
          [COND
             ((EQ N 0)                                                  ; No args
              (RETURN NIL))
             ((AND (< N 0)
                   (NOT SMARTP))                                        ; LAMBDA*
              (RETURN 'U]
          (SETQ SIZE (fetch (FNHEADER NTSIZE) of FNHD))
          [COND
             ((EQ [SETQ LOCALSIZE (- (FOLDLO (if (fetch (FNHEADER NATIVE) of FNHD)
                                                 then (- (fetch (FNHEADER STARTPC) of FNHD)
                                                         4)
                                               else (fetch (FNHEADER STARTPC) of FNHD))
                                            BYTESPERWORD)
                                    (SETQ ENDT (+ (fetch (FNHEADER OVERHEADWORDS) of T)
                                                  (COND
                                                     ((EQ SIZE 0)
                                                                        ; No nametable, but there's a quad of zeros there anyway
                                                      WORDSPERQUAD)
                                                     (T (UNFOLD SIZE 2]
                  0)                                                    ; Nothing extra here

              )
             [(> LOCALSIZE WORDSPERCELL)                                ; There is a second nametable between the first and the code.
              (SETQ IVARS (\CCODEIVARSCAN FNHD ENDT (FOLDLO LOCALSIZE 2]
             ((AND (LISTP (SETQ ENDT (\GETBASEPTR FNHD ENDT)))
                    (LISTP (CAR ENDT)))                             ; It's exactly a pointer to debugging info, car of which is a stylized
                                                                    ; arglist
              (SETQ ENDT (if (AND (EQ (CAAR ENDT)
                                      '&OPTIONAL)
                                  (LISTGET (CDR ENDT)
                                         :INTERLISP))
                             then                                   ; The &OPTIONAL, while strictly correct, is misleading, since it's
                                                                    ; technically true for ALL Interlisp functions.

                                  (CDAR ENDT)
                           else (CAR ENDT)))
              (RETURN (COND
                         (SMARTP ENDT)
                         (T                                         ; Note that if we got this far, function can't be a nospread (we
                                                                    ; caught this in the very first COND up above), which means
                                                                    ; there can't be any &key or &rest
                            (for X in ENDT unless (EQ X '&OPTIONAL) collect (COND
                                                                               ((STRINGP X)
                                                                    ; Callers of ARGLIST are expecting to get something that would
                                                                    ; actually function as one
                                                                                (MKATOM X))
                                                                               (T X]
          [COND
             ((< N 0)                                               ; Waited until now to see if there was a stored arglist, but we
                                                                    ; didn't find one--give up
              (RETURN 'U]
          [COND
             ((NEQ SIZE 0)                                          ; Scan specials name table
              (SETQ IVARS (\CCODEIVARSCAN FNHD (fetch (FNHEADER OVERHEADWORDS) of T)
                                 SIZE IVARS]
          [SETQ IVARS (for I from 0 to (SUB1 N) collect (OR (CDR (ASSOC I IVARS))
                                                           (PACK* '*ARG* I]
          (RETURN (SELECTQ (fetch (FNHEADER ARGTYPE) of FNHD)
                      (3 (CAR IVARS))
                      IVARS])

(\CCODEIVARSCAN
  [LAMBDA (FNHD START SIZE IVARS)                                   ; Edited 30-Jan-91 02:00 by jds

    ;; Search nametable starting at offset START in FNHD for all ivars.  Return list of dotted pairs (index . name) consed onto front of IVARS.  NTSIZE
    ;; is size of nt in words

    (bind NM CODE for NTOFFSET FROM (UNFOLD START BYTESPERWORD) by (CONSTANT (BYTESPERNAMEENTRY))
       while (SETQ NM (\INDEXATOMVAL (GETNAMEENTRY FNHD NTOFFSET)))
       do                                                           ; Note that entry = 0 => NM = NIL terminates the loop
          [COND
             ((EQP [NTSLOT-VARTYPE (SETQ CODE (GETNTOFFSETENTRY FNHD (IPLUS NTOFFSET (UNFOLD SIZE BYTESPERWORD]
                   IVARCODE)
              (push IVARS (CONS (NTSLOT-OFFSET CODE)
                                NM]
       finally (RETURN IVARS])
)

;; Translation machinery for new LAMBDA words

(PUTPROPS LAMBDATRANFNS VARTYPE ALIST)

(ADDTOVAR LAMBDATRANFNS )

(DECLARE%: DONTCOPY

(DECLARE%: EVAL@COMPILE

(PUTPROPS \CCODENARGS MACRO ((FNH)
                             ([LAMBDA (N)
                                 (COND
                                    ((ILESSP N 0)
                                     1)
                                    (T N]
                              (fetch (FNHEADER NA) of FNH))))

(PUTPROPS \CCODEFNTYP MACRO ((FNH)
                             (SELECTQ (\CCODEARGTYPE FNH)
                                 (0 'CEXPR)
                                 (1 'CFEXPR)
                                 (2 'CEXPR*)
                                 'CFEXPR*)))

(PUTPROPS \CCODEARGTYPE MACRO ((FNH)
                               (fetch (FNHEADER ARGTYPE) of FNH)))
)
)

;; CONSTANTS mechanism

(DEFINEQ
(CONSTANTS
  [NLAMBDA VARS                                                          (* rmk%: " 3-Jan-84 13:20")
    (OR COMPVARMACROHASH (SETQ COMPVARMACROHASH (HASHARRAY 100)))
    [for X in VARS do (COND
                         ((LISTP X)
                          (PUTHASH (CAR X)
                                 (LIST 'CONSTANT (CADR X))
                                 COMPVARMACROHASH))
                         (T (PUTHASH X (LIST 'CONSTANT X)
                                   COMPVARMACROHASH]
    VARS])

(CONSTANTEXPRESSIONP
  [LAMBDA (FORM)                                                         ; Edited 21-Jan-91 23:45 by jds
    (COND
       [(LITATOM FORM)
        (COND
           ((OR (NULL FORM)
                (EQ FORM T))
            (LIST FORM))
           ((AND COMPVARMACROHASH (SETQ FORM (GETHASH FORM COMPVARMACROHASH)))
            (CONSTANTEXPRESSIONP FORM]
       [(LISTP FORM)
        (SELECTQ (CAR FORM)
             (QUOTE (CDR FORM))
             (FUNCTION (AND (LITATOM (CADR FORM))
                            (NULL (CDDR FORM))
                            (CDR FORM)))
             (CONSTANT [LET (VALUE)
                            [SETQ VALUE (NLSETQ (EVAL (CADR FORM]
                            (COND
                               (VALUE (LIST (CAR VALUE)))
                               (T (SETQ FORM (COMPILER:OPTIMIZE-AND-MACROEXPAND (CADR FORM)
                                                    *BC-MACRO-ENVIRONMENT*
                                                    (COMPILER:MAKE-CONTEXT :VALUES-USED 1 :PREDICATE-P NIL)))
                                  (LIST (EVAL FORM])
             (COND
                [(FMEMB (CAR FORM)
                        CONSTANTFOLDFNS)
                 (for X in (CDR FORM) collect (CAR (OR (CONSTANTEXPRESSIONP X)
                                                       (RETURN)))
                    finally (RETURN (LIST (APPLY (CAR FORM)
                                                 $$VAL]
                ((NOT (GETD (CAR FORM)))
                 (PROG ((MAC (GETMACROPROP (CAR FORM)
                                    COMPILERMACROPROPS)))
                       (RETURN (AND MAC [NOT (EQUAL FORM (SETQ FORM (MACROEXPANSION FORM MAC]
                                    (CONSTANTEXPRESSIONP FORM]
       ((NUMBERP FORM)
        (LIST FORM])
)

(RPAQ? COMPVARMACROHASH (HASHARRAY 100))
```

;; We need this initialized for the INIT, so don't put it off. (It used to start out NIL and get set later)

```
(ADDTOVAR CONSTANTFOLDFNS PLUS IPLUS TIMES ITIMES DIFFERENCE IDIFFERENCE QUOTIENT IQUOTIENT IMIN IMAX IABS
                                LLSH LRSH LOGOR LOGXOR LOGAND OR AND)

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS COMPVARMACROHASH CONSTANTFOLDFNS)
)

(DECLARE%: EVAL@COMPILE DONTCOPY DONTEVAL@LOAD

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(LOCALVARS . T)
)
)

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(SPECVARS *TAIL* *FN* *FORM* *ARGVAL*)
)

(DECLARE%: EVAL@COMPILE DONTCOPY

(ADDTOVAR LAMS FAULTEVAL FAULTAPPLY)
)

(DECLARE%: DONTEVAL@LOAD DOEVAL@COMPILE DONTCOPY COMPILERVARS
(ADDTOVAR NLAMA CONSTANTS PROG1 COND PROGN OR AND QUOTE LET* LET GO PROG SETQ)

(ADDTOVAR NLAML FUNCTION RETURN)

(ADDTOVAR LAMA BOUNDP APPLY* \INTERPRETER)
)

(PUTPROPS LLINTERP COPYRIGHT ("Venue & Xerox Corporation" T 1981 1982 1983 1984 1985 1986 1987 1988 1990 1991
                                     1992 1993))
```

{MEDLEY}<CLTL2>LLINTERP.;1 28-Jun-2024 18:34:02

-- Listed on 30-Jun-2024 13:12:15 --

FUNCTION INDEX

VARIABLE INDEX

MACRO INDEX

PROPERTY INDEX