

File created: 11-Sep-2021 12:54:19 {DSK}<home>larry>medley>sources>ASTACK.;2

changes to: (FNS STKARGNAME)

previous date: 23-May-91 14:25:00 {DSK}<home>larry>medley>sources>ASTACK.;1

Read Table: INTERLISP

Package: INTERLISP

Format: XCCS

;;
;; Copyright (c) 1982-1987, 1990-1991 by Venue & Xerox Corporation.

```
(RPAQQ ASTACKCOMS
  [ (COMS
    (FNS ARG SETARG \ARG \ARGPTR \SETARG)) ; ARG and SETARG, unusual cases
    (COMS (FNS \RETURN \STACKARGPTR))
    (COMS
      (FNS STKNTH STKNTHNAME STKNAME SETSTKNAME) ; User level stack management
      (FNS STKPOS STKSCAN RETFROM RETTO RESUME \RESUME)
      (FNS STKARG \STKARG SETSTKARG STKARGNAME \SPREADFRAMEP SETSTKARGNAME STKNARGS FRAMESCAN
        \INTERPFRAMENT \FRAMESCAN \VAROFFSET))
    (COMS
      (FNS \RECLAIMSTACKP)) ; finalization for stackps
    (LOCALVARS . T)
    (DECLARE%: DONTEVAL@LOAD DOEVAL@COMPILE DONTCOPY COMPILEVARS (ADDVARS (NLAMA)
      (NLAML SETARG ARG)
      (LAMA]))
```

;; ARG and SETARG, unusual cases

(DEFINEQ

(**ARG**

```
[NLAMBDA (VAR M) ; (* Imm "24-JUL-81 07:43")
  (GETBASEPTR \STACKSPACE (\ARGPTR VAR (\EVAL M]))
```

(**SETARG**

```
[NLAMBDA (VAR M X) ; (* Imm "24-JUL-81 07:43")
  (PUTBASEPTR \STACKSPACE (\ARGPTR VAR (\EVAL M))
  (\EVAL X])
```

(**\ARG**

```
[LAMBDA (VAR M) ; (* Imm "24-JUL-81 07:43")
  ;; Version of ARG which doesn't EVAL 2nd argument
  (GETBASEPTR \STACKSPACE (\ARGPTR VAR M]))
```

(**\ARGPTR**

```
[LAMBDA (VAR N) ; Edited 18-Feb-91 16:51 by jds
```

;;; Returns a pointer to the basic frame corresponding to the lambda* variable VAR, and tests that N is a legal arg#

```
(PROG ((FRAME (\MYALINK))
  (A (NEW-SYMBOL-CODE VAR (\ATOMVALINDEX VAR)))
  (INTERPDEF (fetch (LITATOM DEFPOINTER) of '\INTERPRETER))
  BFLINK P DEF NARGS)
LP (COND
  ((fetch (FX INVALIDP) of FRAME) ; No frame found
  (LISPERROR "ILLEGAL ARG" VAR))
(COND
  ((EQ (SETQ DEF (fetch (FX FNHEADER) of FRAME))
  INTERPDEF) ; See if this is \INTERPRETER running a LAMBDA*
  (OR [AND (SETQ P (\VAROFFSET FRAME A))
    (EQ P (+ (fetch (BF IVAR) of (SETQ BFLINK (fetch (FX BLINK) of FRAME)))
    (UNFOLD (SETQ NARGS (SUB1 (fetch (BF NARGS) of BFLINK)))
    WORDSPERCELL]
  (GO NXT)))
  [(AND (EQ (fetch (FNHEADER NA) of DEF)
    -1)
  (VAROFFSET FRAME A)) ; FRAME is a Lambda nospread, and binds A. Used to insist
  ; that A also be in slot PVAR0, but that's an awkward restriction
  ; now. Maybe should check that A's value is same as NARGS
  (SETQ NARGS (fetch (BF NARGS) of (SETQ BFLINK (fetch (FX BLINK) of FRAME))
  (T (GO NXT))) ; Found the variable as the first PROG variable of a LSTARP
  ; frame
[RETURN (COND
  ((AND (> N 0)
    (<= N NARGS))
  (+ (fetch (BF IVAR) of BFLINK)
  (UNFOLD (SUB1 N)
```

```

                                WORDSPERCELL)))
      (T (LISPERROR "ILLEGAL ARG" N]
NXT (SETQ FRAME (fetch (FX ALINK) of FRAME))
      (GO LP])

```

(\SETARG

[LAMBDA (VAR M X)

(* Imm "24-JUL-81 07:43")

;; Version of SETARG which doesn't eval 2nd and 3rd arguments.

```

(PUTBASEPTR \STACKSPACE (\ARGPTR VAR M)
  X])

```

)

(DEFINEQ

(\RETURN

[LAMBDA (X)

(* bvm%: "11-Nov-86 11:44")

(DECLARE (LOCALVARS . T))

;; for use by LLBREAK--call RAID, then simulate the RETURN opcode.

```

(RAID X)
(PROG1 X
  (\SMASHLINK NIL (SETQ X (fetch (FX CLINK) of (\MYALINK)))
    X)))

```

(\STACKARGPTR

[LAMBDA (POS)

(* bvm%: " 7-Oct-86 22:07")

;; return the index of the frame extension corresponding to POS or cause appropriate error

```

(COND
  [(OR (STACKP POS)
        (TYPENAMEP POS 'PROCESS))
    (LET ((FRAME (fetch EDFXP of POS)))
      (COND
        ((OR (EQ FRAME 0)
              (AND (fetch (FX INVALIDP) of (fetch (FX CLINK) of FRAME))
                   (NEQ (fetch (FX FRAMENAME) of FRAME)
                        T)))
          (LISPERROR "STACK PTR HAS BEEN RELEASED" POS))
        (T FRAME)
        (T (PROG ((FX (\MYALINK))
                  (P POS))
          [COND
            (NULL POS)
            (LISPERROR "ILLEGAL STACK ARG" POS))
          [(EQ POS T)
            ;; scan up for top frame. This could possibly be a constant, although there might be some circumstances where it could move
            (PROG NIL
              TOPLP
              (COND
                ([NOT (fetch (FX INVALIDP) of (SETQ P (fetch (FX CLINK) of FX))
                  (SETQ FX P)
                  (GO TOPLP]
                [(NUMBERP POS)
                  (COND
                    ((EQ (SETQ P (FIX POS))
                        0)
                     (SETQ P 1)))
                    (COND
                      [(IGREATERP P 0)
                        (PROG NIL
                          ALP (COND
                            ((fetch (FX INVALIDP) of (SETQ FX (fetch (FX ALINK) of FX)))
                             (LISPERROR "ILLEGAL STACK ARG" POS))
                            ((NEQ (SETQ P (SUB1 P))
                                0)
                             (GO ALP]
                      (T
                        (PROG NIL
                          (SETQ P (IMINUS P))
                          CLP (COND
                            ((fetch (FX INVALIDP) of (SETQ FX (fetch (FX CLINK) of FX)))
                             (LISPERROR "ILLEGAL STACK ARG" POS))
                            ((NEQ (SETQ P (SUB1 P))
                                0)
                             (GO CLP]
                        (T
                          (PROG NIL
                            SCNLP
                            (COND
                              ((fetch (FX INVALIDP) of (SETQ FX (fetch (FX CLINK) of FX)))
                               (LISPERROR "ILLEGAL STACK ARG" POS))

```

; if POS is STACKP, it is merely the contents

; Either stack pointer has been released explicitly, or somebody
; has already returned to/around the frame in question

; those functions which allow NIL should explicitly check for it.

; Search ALinks

; Search CLinks

; implicit STKPOS searching for a given name

```

        ((NOT (EQMEMB (\STKNAME FX)
                        POS))
         (GO SCNLP]
(COND
  ((IGEQL (fetch (FX USECNT) of FX)
           \MAXSAFEUSECOUNT)
   (LISPERROR "ILLEGAL STACK ARG" POS)))
(RETURN FX])

```

)

;; User level stack management

(DEFINEQ

(STKNTH

(* bvm%: " 5-Feb-85 15:50")

```

[LAMBDA (N IPOS OPOS)
  (PROG ((I (OR N -1))
         CFLAG FRAME)
    [COND
      ((ILESSP I 0)
       (SETQ CFLAG T)
       (SETQ I (IMINUS I)
       [SETQ FRAME (COND
                     (IPOS (\STACKARGPTR IPOS))
                     ((EQ I 0)
                      (LISPERROR "ILLEGAL STACK ARG" N))
                     (T (add I -1)
                        (\MYALINK]
      LP [COND
          ((fetch (FX INVALIDP) of FRAME)
           (RELSTK OPOS)
           (RETURN))
          ((EQ I 0)
           (RETURN (\MAKESTACKP OPOS FRAME)))
          (CFLAG (SETQ FRAME (fetch (FX CLINK) of FRAME)))
          (T (SETQ FRAME (fetch (FX ALINK) of FRAME]
          (SETQ I (SUB1 I))
          (GO LP])

```

(STKNTHNAME

(* bvm%: " 5-Feb-85 15:51")

```

[LAMBDA (N POS)
  (PROG ((I (OR N -1))
         CFLAG FRAME)
    [COND
      ((ILESSP I 0)
       (SETQ CFLAG T)
       (SETQ I (IMINUS I)
       [SETQ FRAME (COND
                     (POS (\STACKARGPTR POS))
                     ((EQ I 0)
                      (LISPERROR "ILLEGAL STACK ARG" N))
                     (T (add I -1)
                        (\MYALINK]
      LP [COND
          ((fetch (FX INVALIDP) of FRAME)
           (RETURN))
          ((EQ I 0)
           (RETURN (fetch (FX FRAMENAME) of FRAME)))
          (CFLAG (SETQ FRAME (fetch (FX CLINK) of FRAME)))
          (T (SETQ FRAME (fetch (FX ALINK) of FRAME]
          (SETQ I (SUB1 I))
          (GO LP])

```

(STKNAME

(* Imm " 2-Jul-86 12:37")

```

[LAMBDA (POS)
  (\STKNAME (\STACKARGPTR POS])

```

(SETSTKNAME

(* bvm%: "15-Aug-84 11:13")

```

[LAMBDA (POS NAME)
  (PROG ((FRAME (\STACKARGPTR POS))
         FNH)
    [COND
      ((fetch (FX VALIDNAMETABLE) of FRAME)
       (SETQ FNH (fetch (FX NAMETABLE#) of FRAME))
       UNINTERRUPTABLY
       (replace (FX VALIDNAMETABLE) of FRAME with NIL)
      (COND
        ((EQ (\HILOC FNH)
              \STACKHI)
         (replace (FNHEADER %#FRAMENAME) of FNH with NAME))
        (T (replace (FNHEADER FRAMENAME) of FNH with NAME))
        (replace (FX VALIDNAMETABLE) of FRAME with T)))

```

; There is already a copied nametable here, just smash it

; Do this so that the stack remains consistent, even while
; uninterruptable. This for SPY etc.

; Don't refcnt on the stack

```

(T (SETQ FNH (\COPYFNHEADER (fetch (FX FNHEADER) of FRAME)))
  (replace (FNHEADER FRAMENAME) of FNH with NAME)
  (UNINTERRUPTABLY
    (replace (FX NAMETABLE) of FRAME with FNH)))
(RETURN NAME])
)

```

(DEFINEQ

(STKPOS

(* Imm "2-Jul-86 13:02")

```

[LAMBDA (FRAMENAME N IPOS OPOS)
  (PROG (FLAG [FX (COND
    ((NULL IPOS)
     (\MYALINK))
    (T (\STACKARGPTR IPOS]
      (I (OR N -1)))
    [COND
      ((IGREATERP 0 I)
       (SETQ FLAG (SETQ I (IDIFFERENCE 0 I]
    LP [COND
      ((EQ (\STKNAME FX)
        FRAMENAME)
       (COND
        ((ILEQ (SETQ I (SUB1 I))
         0)
        (RETURN (\MAKESTACKP OPOS FX]
      (COND
        ([fetch (FX INVALIDP) of (SETQ FX (COND
          (FLAG (fetch (FX CLINK) of FX))
          (T (fetch (FX ALINK) of FX]
          (RELSTK OPOS)
          (RETURN)))
        (GO LP]))

```

(STKSCAN

; Edited 19-Feb-91 22:58 by jds

```

[LAMBDA (VAR IPOS OPOS)
  (AND (LITATOM VAR)
    (PROG [[FX (COND
      ((NULL IPOS)
       (\MYALINK))
      (T (\STACKARGPTR IPOS]
        (A (NEW-SYMBOL-CODE VAR (\ATOMVALINDEX VAR]
    LP (COND
      ((FRAMESCAN FX A)
       (RETURN (\MAKESTACKP OPOS FX)))
      ((fetch (FX INVALIDP) of (SETQ FX (fetch (FX ALINK) of FX)))
       (RELSTK OPOS)
       (RETURN))
      (T (GO LP]))

```

(RETFROM

(* bvm "22-Nov-86 15:34")

```

[LAMBDA (POS VAL FLG)
  (LET ((P (\STACKARGPTR POS)))
    (COND
      ((fetch (FX INVALIDP) of (SETQ P (fetch (FX CLINK) of P)))
       (LISPERROR "ILLEGAL RETURN" VAL)))
    (\SMASHRETURN NIL P (AND FLG POS))
    VAL]))

```

(RETTO

(* bvm "22-Nov-86 15:34")

```

[LAMBDA (POS VAL FLG)
  (if (EQ POS T)
    then (RESET)
    else (LET ((P (\STACKARGPTR POS)))
      (\SMASHRETURN NIL P (AND FLG POS))
      VAL]))

```

(RESUME

(* bvm%: "11-Nov-86 20:56")

;; FROMPTR is a stkptr which is smashed to contain a pointer to the caller of RESUME. Control is transferred to the frame specified by TOPTR, releasing that stack pointer. A call to this RESUME returns VAL as the value of the RESUME specified by TOPTR.

```

(PROG [[FROMFX (fetch EDFXP of (\DTEST FROMPTR 'STACKP]
  (TOFX (fetch EDFXP of (\DTEST TOPTR 'STACKP]
  (COND
    ((OR (fetch (FX INVALIDP) of TOFX)
      (fetch (FX INVALIDP) of (fetch (FX CLINK) of TOFX)))
    ; released stack pointer, or stack pointer that has been thrown
    ; thru
    (LISPERROR "STACK PTR HAS BEEN RELEASED" TOPTR)))
  (UNINTERRUPTABLY
    (COND

```

```

        ((NOT (fetch (FX INVALIDP) of FROMFX)) ; Release FROMPTR if it hasn't been yet
         (\DECUSECOUNT FROMFX)))
      (replace EDFXP of FROMPTR with (\MYALINK))
      (replace EDFXP of TOPTR with 0)
      (\RESUME TOFX))
  VAL])

```

```

(\RESUME
 [LAMBDA (FRAME) ; (* bvm%: " 5-Jun-85 17:08")
  (replace (FX ACLINK) of (\MYALINK) with FRAME)
  FRAME])
)

```

```

(DEFINEQ

```

```

(STKARG ; (* Imm " 7-Nov-86 01:37")
 [LAMBDA (N POS DEFAULT)
  (LET ((VAL "NO SUCH ARG"))
   (CL:WHEN (EQ VAL (SETQ VAL (\STKARG N (\STACKARGPTR POS)
                                     DEFAULT VAL)))
    (LISPERROR "ILLEGAL STACK ARG" N))
  VAL])
)

```

```

(\STKARG ; Edited 23-May-91 12:49 by jds
 [LAMBDA (N FRAME DEFAULT NOSUCH)
  ;; Find the value for variable N looking from fFRAME upward (??)
  (PROG ((INDEX N)
         BLINK NARGS NT NTSIZE)
    (SETQ NT (\INTERPPFRAME FRAME))
    [COND
     ((LITATOM N)
      (SETQ INDEX (OR (\FRAMESCAN FRAME (NEW-SYMBOL-CODE N (\ATOMVALINDEX N))
                                NT)
                      (RETURN NOSUCH))
      (COND
       ((ILESSP INDEX 1)
        (RETURN NOSUCH))
       [NT ; Interpreter frame
        (COND
         [(\SPREADFRAMEP FRAME)
          (OR [AND (IGREATERP INDEX 0)
                  (ILEQ INDEX (SETQ NARGS (fetch (BF NARGS) of (SETQ BLINK (fetch (FX BLINK)
                                                                                      of FRAME]
                                                                                      (RETURN NOSUCH))
                  (SETQ INDEX (IPLUS (fetch (BF IVAR) of BLINK)
                                      (UNFOLD (SUB1 INDEX)
                                              WORDSPERCELL]
          ((OR [IGEQ INDEX (FOLDLO (SETQ NTSIZE (fetch (FNHEADER NTSIZE) of NT))
                  (CONSTANT (WORDSPERNAMEENTRY]
          (NULL-NTENTRY (GETSTKNAMEENTRY (\ADDBASE NT (fetch (FNHEADER OVERHEADWORDS) of T))
                        (UNFOLD (IPLUS INDEX -1)
                                (CONSTANT (WORDSPERNAMEENTRY]
                                ; Out of range
          (RETURN NOSUCH))
          (T (SETQ INDEX (IPLUS (SELECTC (NTSLOT-VARTYPE (GETSTKNTOFFSETENTRY
                                                         [SETQ NT (\ADDBASE
                                                         NT
                                                         (IPLUS NTSIZE (UNFOLD
                                                         (SUB1 INDEX)
                                                         (CONSTANT (
                                                         WORDSPERNAMEENTRY
                                                         )))
                                                         (fetch (FNHEADER
                                                         OVERHEADWORDS)
                                                         of T]
                                                         0))
          (IVARCODE (fetch (BF IVAR) of (fetch (FX BLINK) of FRAME)))
          (PVARCODE (fetch (FX FIRSTPVAR) of FRAME))
          (SHOULDN))
          (UNFOLD (NTSLOT-OFFSET (GETSTKNTOFFSETENTRY NT 0))
                  WORDSPERCELL]
    [[ILEQ INDEX (SETQ NARGS (fetch (BF NARGS) of (SETQ BLINK (fetch (FX BLINK) of FRAME]
    (SETQ INDEX (IPLUS (fetch (BF IVAR) of BLINK)
                        (UNFOLD (SUB1 INDEX)
                                WORDSPERCELL]
    [ILEQ (SETQ INDEX (IDIFFERENCE INDEX NARGS))
     (fetch (FX FNHEADER NLOCALS) of FRAME))
    (SETQ INDEX (IPLUS (fetch (FX FIRSTPVAR) of FRAME)
                        (UNFOLD (SUB1 INDEX)
                                WORDSPERCELL]
    (T (RETURN NOSUCH)))
  (RETURN (COND
   ((NOT (fetch (PVARSLT BOUND) of (STACKADDBASE INDEX)))

```

```

    DEFAULT)
    (T (STACKGETBASEPTR INDEX))

```

(SETSTKARG

; Edited 19-Feb-91 22:49 by jds

```

[LAMBDA (N POS VAL)
  (PROG ((FRAME (\STACKARGPTR POS))
    (INDEX N)
    BLINK NARGS NT NTSIZE)
    (SETQ NT (\INTERPFRAMENT FRAME))
    [COND
      ((LITATOM N)
        (SETQ INDEX (OR (\FRAMESCAN FRAME (NEW-SYMBOL-CODE N (\ATOMVALINDEX N))
          NT)
          (LISPERROR "ILLEGAL STACK ARG" N)
        )
      )
      [SETQ INDEX (COND
        ((ILESSP INDEX 1)
          (LISPERROR "ILLEGAL STACK ARG" INDEX))
        [NT
          (COND
            ([OR [IGEQ INDEX (FOLDLO (SETQ NTSIZE (fetch (FNHEADER NTSIZE) of NT))
              (CONSTANT (WORDSPERNAMEENTRY)
              (NULL-NTENTRY (GETSTKNAMEENTRY NT (IPLUS (fetch (FNHEADER OVERHEADWORDS)
                of T)
                (UNFOLD (SUB1 INDEX)
                  (CONSTANT (WORDSPERNAMEENTRY
                    ]
                    ; Out of range
                    (LISPERROR "ILLEGAL STACK ARG" INDEX))
                    (T (IPLUS (SELECTC (NTSLOT-VARTYPE (GETSTKNTOFFSETENTRY
                      [SETQ NT (\ADDBASE
                      NT
                      (IPLUS NTSIZE (fetch (FNHEADER OVERHEADWORDS)
                        )
                        of T)
                        (UNFOLD INDEX (CONSTANT
                          (WORDSPERNAMEENTRY
                            ]
                            0))
                            (IVARCODE (fetch (BF IVAR) of (fetch (FX BLINK) of FRAME)))
                            (PVARCODE (fetch (FX FIRSTPVAR) of FRAME))
                            (SHOULDNT))
                            (UNFOLD (NTSLOT-OFFSET (GETSTKNTOFFSETENTRY NT 0))
                              WORDSPERCELL]
                            ((ILEQ INDEX (SETQ NARGS (fetch (BF NARGS) of (SETQ BLINK (fetch (FX BLINK) of FRAME)
                              (IPLUS (fetch (BF IVAR) of BLINK)
                                (UNFOLD (SUB1 INDEX)
                                  WORDSPERCELL)))
                              ((ILEQ (SETQ INDEX (IDIFFERENCE INDEX NARGS))
                                (fetch (FX FNHEADER NLOCALS) of FRAME))
                                (IPLUS (fetch (FX FIRSTPVAR) of FRAME)
                                  (UNFOLD (SUB1 INDEX)
                                    WORDSPERCELL)))
                                (T (LISPERROR "ILLEGAL STACK ARG" N]
                                (RETURN (COND
                                  ((fetch (PVARSLT BOUND) of (STACKADDBASE INDEX))
                                    (STACKPUTBASEPTR INDEX VAL))
                                  (T (LISPERROR "ILLEGAL STACK ARG" N])
                                )

```

(STKARGNAME

; Edited 11-Sep-2021 12:51 by larry

;; Given an interpreted frame and an argument number, return the name of that argument (actually, just the n-th NameTable entry)

;; OR, Given the name of an argument and a frame to start looking from, return the nametable offset entry.

;; Brother, what an overloading!!

```

(PROG ((FRAME (\STACKARGPTR POS))
  NT NM (NTENTRY N)
  NARGS)
  (SETQ NT (\INTERPFRAMENT FRAME))
  [COND
    ((LITATOM NTENTRY)
      (SETQ NTENTRY (\FRAMESCAN FRAME (NEW-SYMBOL-CODE NTENTRY (\ATOMVALINDEX NTENTRY))
        NT)
    )
    [COND
      (NT
        (RETURN (COND
          ((\SPREADFRAMEP FRAME)
            (* (LIST (QUOTE ARG) (\INDEXATOMVAL
              (GETBASE NT (fetch (FNHEADER OVERHEADWORDS) of T))
            ) N))
          (NIL)
          (T (OR [AND (IGREATERP NTENTRY 0)
            (ILESSP NTENTRY (fetch (FNHEADER NTSIZE) of NT))

```

```

(\INDEXATOMVAL (GETSTKNAMEENTRY (\ADDBASE NT (fetch (FNHEADER
                                                    OVERHEADWORDS)
                                                    of T)))
  (UNFOLD (IPLUS NTENTRY -1)
    (CONSTANT (WORDSPERNAMEENTRY]
    (LISPERROR "ILLEGAL STACK ARG" N]
  (SETQ NT (fetch (FX NAMETABLE) of FRAME))
  [SETQ NTENTRY (COND
    ((ILEQ NTENTRY 0)
      (LISPERROR "ILLEGAL STACK ARG" N))
    ([ILEQ NTENTRY (SETQ NARGS (fetch (BF NARGS) of (fetch (FX BLINK) of FRAME]
      (MAKE-NTENTRY IVARCODE (SUB1 NTENTRY)))
    ((ILEQ (SETQ NTENTRY (IDIFFERENCE NTENTRY NARGS))
      (fetch (FNHEADER NLOCALS) of NT))
    (COND
      ([NOT (fetch (PVARSLT BOUND) of (STACKADDBASE (IPLUS (fetch (FX FIRSTPVAR)
                                                                    of FRAME)
        (UNFOLD (SUB1 NTENTRY)
          WORDSPERCELL]
        (RETURN)))
      (MAKE-NTENTRY PVARCODE (SUB1 NTENTRY)))
      (NOERROR (RETURN))
      (T (LISPERROR "ILLEGAL STACK ARG" N]
    (RETURN (for NT1 from (fetch (FNHEADER OVERHEADWORDS) of T) by (CONSTANT (WORDSPERNAMEENTRY))
      as NT2 from (IPLUS (fetch (FNHEADER OVERHEADWORDS) of NT)
        (fetch (FNHEADER NTSIZE) of NT))
      by (CONSTANT (WORDSPERNTOFFSETENTRY)) until (NULL-NTENTRY (SETQ NM (GETSTKNAMEENTRY NT NT1)))
      do (COND
        ((EQP NTENTRY (GETSTKNTOFFSETENTRY NT NT2))
          (RETURN (\INDEXATOMVAL NM]))

```

(SPREADFRAMEP

```

[LAMBDA (FRAME)
  (LET (NARGS BFLINK)
    (EQ (\GETBASEPTR \STACKSPACE (IPLUS (fetch (BF IVAR) of (SETQ BFLINK (fetch (FX BLINK) of FRAME)))
      (UNFOLD (SETQ NARGS (SUB1 (fetch (BF NARGS) of BFLINK)))
        WORDSPERCELL)))
    NARGS))

```

(* Imm "1-Jun-86 17:19")

(SETSTKARGNAME

```

[LAMBDA (N POS NAME)
  (PROG ((FRAME (\STACKARGPTR POS))
    NT NM (NTENTRY N)
    NARGS)
    (SETQ NT (\INTERPFAMENT FRAME))
    [COND
      ((LITATOM NTENTRY)
        (SETQ NTENTRY (\FRAMESCAN FRAME (NEW-SYMBOL-CODE NTENTRY (\ATOMVALINDEX NTENTRY))
          NT]
      (COND
        (NT
          (RETURN (OR [AND (IGREATERP NTENTRY 0)
            [ILESSP NTENTRY (FOLDLO (fetch (FNHEADER NTSIZE) of NT)
              (CONSTANT (WORDSPERNAMEENTRY]
              (\INDEXATOMVAL (GETSTKNAMEENTRY (\ADDBASE NT (fetch (FNHEADER OVERHEADWORDS)
                of T)))
              (UNFOLD (IPLUS NTENTRY -1)
                (CONSTANT (WORDSPERNAMEENTRY]
              (LISPERROR "ILLEGAL STACK ARG" N]
            (SETQ NT (\COPYFNHEADER (fetch (FX NAMETABLE) of FRAME)))
          ]SETQ NTENTRY (COND
            ((ILEQ NTENTRY 0)
              (LISPERROR "ILLEGAL STACK ARG" N))
            ([ILEQ NTENTRY (SETQ NARGS (fetch (BF NARGS) of (fetch (FX BLINK) of FRAME]
              (MAKE-NTENTRY IVARCODE (SUB1 NTENTRY)))
            ((ILEQ (SETQ NTENTRY (IDIFFERENCE NTENTRY NARGS))
              (fetch (FNHEADER NLOCALS) of NT))
            (MAKE-NTENTRY PVARCODE (SUB1 NTENTRY)))
            (T (LISPERROR "ILLEGAL STACK ARG" N]
          (for NT1 from (fetch (FNHEADER OVERHEADWORDS) of T) by (CONSTANT (WORDSPERNAMEENTRY)) as NT2
            from [IPLUS (fetch (FNHEADER OVERHEADWORDS) of T)
              (UNFOLD (fetch (FNHEADER NTSIZE) of NT)
                (CONSTANT (WORDSPERNAMEENTRY]
            by (CONSTANT (WORDSPERNTOFFSETENTRY)) until (NULL-NTENTRY (SETQ NM (GETSTKNAMEENTRY NT NT1)))
            do (COND
              ((EQP NTENTRY (GETSTKNTOFFSETENTRY NT NT2))
                (SETSTKNAMEENTRY NT NT1 (\ATOMVALINDEX NAME))
                (UNINTERRUPTABLY
                  (replace (FX NAMETABLE) of FRAME with NT))
                (RETURN NAME])

```

; Edited 20-Feb-91 01:04 by jds

; Interpreted frame

; Need to copy nametable in order to smash the var name

(STKNARGS

[LAMBDA (POS INCLUDEPVAR)

; Edited 19-Feb-91 17:09 by jds

```

(PROG ((FRAME (\STACKARGPTR POS))
      NA INTERPNT)
  (RETURN (COND
    ((EQ (fetch (FX FRAMENAME) of FRAME)
      '\INTERPRETER)
      (SETQ NA (fetch (BF NARGS) of (fetch (FX BLINK) of FRAME)))
      (RETURN (SUB1 NA)))
    ((SETQ INTERPNT (\INTERPFRAMENT FRAME)) ; this is an interpreted frame. INTERPNT points at the name
      ; table of the frame

      [COND
        ((\SPREADFRAMEP FRAME)
          (RETURN (SUB1 (fetch (BF NARGS) of (fetch (FX BLINK) of FRAME]
        [SETQ NA (FOLDLO (fetch (FNHEADER NTSIZE) of INTERPNT)
          (CONSTANT (WORDSPERNAMEENTRY]
          ; Return number of VARS in nt. Padded with up to 4 zeros at
          ; end, so have to check

      [COND
        ((IGREATERP NA 0)
          (do (add NA -1) repeatwhile (NULL-NTENTRY (GETSTKNAMEENTRY (\ADDBASE INTERPNT
            (fetch (FNHEADER
              OVERHEADWORDS
            )
            of T))
          (UNFOLD (IPLUS NA -1)
            (CONSTANT (WORDSPERNAMEENTRY]
          NA)
        (T (SETQ NA (fetch (BF NARGS) of (fetch (FX BLINK) of FRAME)))
          (RETURN (COND
            (INCLUDEPVAR (IPLUS NA (fetch (FX FNHEADER NLOCALS) of FRAME)))
            (T NA]))

```

(FRAMESCAN

```

[LAMBDA (ATOM POS) ; Edited 19-Feb-91 22:56 by jds
  (PROG ((FX (\STACKARGPTR POS)))
    (RETURN (\FRAMESCAN FX (COND
      ((LITATOM ATOM)
        (NEW-SYMBOL-CODE ATOM (\ATOMVALINDEX ATOM)))
      (T (RETURN NIL)))
    (\INTERPFRAMENT FX]))

```

(\INTERPFRAMENT

```

[LAMBDA (FX) (* bvm%: " 2-OCT-81 23:32")
  ;; If FX is an interpreter frame (nametable is on stack), returns its nametable
  (AND (fetch (FX VALIDNAMETABLE) of FX)
    (EQ (fetch (FX NAMETABHI) of FX)
      \STACKHI)
    (fetch (FX NAMETABLE#) of FX]))

```

(\FRAMESCAN

```

[LAMBDA (FRAME ATOM# INTERPNT) ; Edited 18-Feb-91 13:01 by jds

```

;;; Returns index of binding of atom number ATOM# in FRAME. Indices of ivars start at 1, of pvars at nargs+1. If INTERPNT is given, this is an interpreter frame, and we merely return index of atom in its nametable, regardless of type

```

(for OFFSET from (fetch (FNHEADER OVERHEADWORDS) of T) by (CONSTANT (WORDSPERNAMEENTRY))
  bind (NT _ (OR INTERPNT (fetch (FX NAMETABLE) of FRAME)))
  TMP NAME
  until (NULL-NTENTRY (SETQ NAME (GETSTKNAMEENTRY NT OFFSET)))
  do (COND
    ((EQ NAME ATOM#)
      ;; Found ATOM# in nametable. Now look in second half of table to see what kind of binding and where it lies
      (COND
        [INTERPNT (RETURN (ADD1 (FOLDLO (IDIFFERENCE OFFSET (fetch (FNHEADER OVERHEADWORDS) of T))
          (CONSTANT (WORDSPERNAMEENTRY]
        (T (SELECTC [NTSLOT-VARTYPE (SETQ TMP (GETSTKNTOFFSETENTRY NT (IPLUS OFFSET (fetch (FNHEADER
          NTSIZE)
          of NT]
          (IVARCODE (RETURN (ADD1 (NTSLOT-OFFSET TMP))))
          (PVARCODE (AND [fetch (PVARSLT BOUND) of (ADDSTACKBASE (IPLUS (fetch (FX FIRSTPVAR)
            of FRAME)
            (UNFOLD (SETQ TMP
              (NTSLOT-OFFSET
                TMP)
              WORDSPERCELL]
            (RETURN (IPLUS TMP (fetch (BF NARGS) of (fetch (FX BLINK) of FRAME))
              1))))
          (FVARCODE (RETURN))
          (RAID]))

```

(\VAROFFSET

```

[LAMBDA (FRAME ATN) ; Edited 18-Feb-91 15:19 by jds

```


::: Returns stack offset to binding of atom number ATN in FRAME, or NIL if it is not bound here.

```

(for OFFSET from (fetch (FNHEADER OVERHEADWORDS) of T) by (CONSTANT (WORDSPERNAMEENTRY))
  bind (NT _ (fetch (FX NAMETABLE) of FRAME))
    TMP NAME
  until (NULL-NTENTRY (SETQ NAME (GETSTKNAMEENTRY NT OFFSET)))
  do (COND
    ( (EQ NAME ATN)
      ;; Found ATN in nametable. Now look in second half of table to see what kind of binding and where it lies
      (SELECTC [NTSLOT-VARTYPE (SETQ TMP (GETSTKNTOFFSETENTRY NT (IPLUS OFFSET (fetch (FNHEADER NTSIZE)
                                                                                          of NT])
                                                                                          (IVARCODE (RETURN (IPLUS (fetch (BF IVAR) of (fetch (FX BLINK) of FRAME))
                                                                                          (UNFOLD (NTSLOT-OFFSET TMP)
                                                                                          WORDSPERCELL))))))
      (PVARCODE (AND [fetch (PVARSLT BOUND) of (ADDSTACKBASE (SETQ TMP (IPLUS (fetch (FX FIRSTPVAR)
                                                                                          of FRAME)
                                                                                          (UNFOLD (NTSLOT-OFFSET
                                                                                          TMP)
                                                                                          WORDSPERCELL])
      (RETURN TMP)))
      (FVARCODE (RETURN))
      (RAID])
    )
  )

)

;; finalization for stackps

(DEFINEQ
(\RECLAIMSTACKP
  [LAMBDA (PTR)
    ;; Finalization for STACKP's -- release the stack frames tied down by PTR
    (LET ((FX (fetch (STACKP EDFXP) of PTR)))
      (IF (NOT (fetch (FX INVALIDP) of FX))
        THEN (\DECUSECOUNT FX)
        NIL))
    )
  )

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(LOCALVARS . T)
)

(DECLARE%: DONTVAL@LOAD DOEVAL@COMPILE DONTCOPY COMPILEVARS

(ADDTOVAR NLAMA )

(ADDTOVAR NLAML SETARG ARG)

(ADDTOVAR LAMA )
)

(PUTPROPS ASTACK COPYRIGHT ("Venue & Xerox Corporation" 1982 1983 1984 1985 1986 1987 1990 1991))

```

; Edited 4-Mar-87 10:43 by bvm:

; return NIL to say it's ok to reclaim

FUNCTION INDEX

ARG1	SETARG1	STKARGNAME6	STKPOS4	\INTERPFRAGMENT 8	\SPREADFRAMEP .7
FRAMESCAN8	SETSTKARG6	STKNAME3	STKSCAN4	\RECLAIMSTACKP 9	\STACKARGPTR ..2
RESUME4	SETSTKARGNAME .7	STKNARGS7	\ARG1	\RESUME5	\STKARG5
RETFROM4	SETSTKNAME3	STKNTH3	\ARGPTR1	\RETURN2	\VAROFFSET8
RETTO4	STKARG5	STKNTHNAME3	\FRAMESCAN8	\SETARG2	
