File created: 20-Oct-94 10:19:00 **{DSK}<lispcore>library>GCHAX.;6**

changes to: (FNS \SEE-GC-ENTRY \SFLHASHLOOKUP \MAPGC)

previous date: 19-Oct-94 13:22:23 {DSK}<lispcore>library>GCHAX.;4

Read Table: INTERLISP

Package: INTERLISP

Format: XCCS


```
(RPAQQ GCHAXCOMS
       [(COMS                                                    ; Looking at gc table
              (FNS \MAPGC \SHOWGC \GCENTRIES.BY.TYPE \#COLLISIONS \#OVERFLOWS \GCSTATS.AUX \SEE-GC-ENTRY))
        (COMS                                                    ; Hacking free lists
              (FNS \PRINTFREELIST \SHOWFREELISTS \SCANFREELIST \ISONFREELIST)
              (FNS PFL SFL)
              (FNS \COLLECTINUSE \SORTFREELIST \SFLHASHLOOKUP))
        (COMS                                                    ; finding circularities
              (FNS \SHOWCIRCULARITY \SHOWCIRCULARITY1 \SHOWCIRCULARLIST \SHOWCIRCULARPATH))
        (COMS                                                    ; special window storage leak finder
              (FNS \SHOW.CLOSED.WINDOWS \WINDOW.ACCOUNTED.FOR?))
        (COMS                                                    ; Brute force search for raw pointers
              (FNS \FINDPOINTER \FINDPOINTERS.OF.TYPE \FINDPOINTER.FOUND \FINDPOINTER.NEWITEM \FINDPOINTER.LISTP
                   \FINDPOINTER.LISTP.FREE \FINDPOINTER.TYPE \FINDPOINTER.INTERPRET.RECORD \FINDPOINTER.STACK
                   \FINDPOINTER.PARSE.STACK \FINDPOINTER.FOUND.ON.STACK))
        (DECLARE%: EVAL@COMPILE DONTCOPY (MACROS \COERCETOTYPENUMBER .ALLOCATED.PER.PAGE. MDSTYPE#)
               (FILES (LOADCOMP)
                      LLGC LLBASIC)
               (LOCALVARS . T)
               (GLOBALVARS SYSTEMRECLST))
        (DECLARE%: DONTEVAL@LOAD DOEVAL@COMPILE DONTCOPY COMPILERVARS (ADDVARS (NLAMA SFL PFL)
                                                                              (NLAML)
                                                                              (LAMA])

;; Looking at gc table

(DEFINEQ

(\MAPGC
  [LAMBDA (MAPFN INCLUDEZEROCNT)                                 ; Edited 20-Oct-94 10:04 by sybalsky
    (PROG ((I 0)
           ENTRY LINK OVENTRY CNT)
      LP   (SETQ ENTRY (\ADDBASE \HTMAIN (LLSH I 1)))
           [COND
              ((fetch (GC EMPTY) of ENTRY))
              [(NOT (fetch (GC LINKP) of ENTRY))
               (COND
                  ((OR INCLUDEZEROCNT (NEQ (fetch (GC CNT) of ENTRY)
                                           0))
                   (APPLY* MAPFN (\VAG2 (fetch (GC HIBITS) of ENTRY)
                                        (LLSH I 1))
                          [COND
                             ((ILESSP (SETQ CNT (fetch (GC CNT) of ENTRY))
                                      \MAXHTCNT)
                              CNT)
                             (T (\GC.LOOKUP.BIGREFCNT (\VAG2 (fetch (GC HIBITS) of ENTRY)
                                                             (LLSH I 1]
                          NIL]
              (T (SETQ LINK (fetch (GC LINKPTR) of ENTRY))
                 (do (SETQ OVENTRY (\ADDBASE \HTCOLL (LLSH LINK 1)))
                     (SETQ LINK (fetch (GC NXTPTR) of OVENTRY))
                     (COND
                        ((OR (NEQ (SETQ CNT (fetch (GC CNT) of OVENTRY))
                                  0)
                             INCLUDEZEROCNT)
                         (APPLY* MAPFN (\VAG2 (fetch (GC HIBITS) of OVENTRY)
                                              (LLSH I 1))
                                [COND
                                   ((ILESSP CNT \MAXHTCNT)
                                    CNT)
                                   (T (\GC.LOOKUP.BIGREFCNT (\VAG2 (fetch (GC HIBITS) of OVENTRY)
                                                                   (LLSH I 1]
                                T)))
                     repeatuntil (EQ LINK 0]
           (COND
              ((ILESSP (add I 1)
                       32768)
               (GO LP])
```

(\SHOWGC

```
[LAMBDA (ONLYTYPES COLLECT FILE CARLVL CDRLVL MINCNT)          ; Edited 23-Jan-87 16:44 by jop
  (OR CARLVL (SETQ CARLVL 2))
  (OR CDRLVL (SETQ CDRLVL 6))
  (OR MINCNT (SETQ MINCNT 2))
  [COND
     (ONLYTYPES (SETQ ONLYTYPES (for TYPE inside ONLYTYPES collect (\COERCETOTYPENUMBER TYPE]
  (RESETLST
      [RESETSAVE (OUTPUT (COND
                             ((NULL FILE)
                              T)
                             ((OPENP FILE 'OUTPUT))
                             (T [RESETSAVE NIL (LIST 'CLOSEF (SETQ FILE (OPENFILE FILE 'OUTPUT]
                                FILE]
      (printout NIL " cnt   datum" T)
      [PROG [(TOTALCNT 0)
             (COLLCNT 0)
             (MAXCNT 0)
             (SELECTEDITEMS (AND COLLECT (CONS]
            (DECLARE (SPECVARS TOTALCNT COLLCNT MAXCNTSELECTEDITEMS))
            (SETQ RESULT SELECTEDITEMS)
            (\MAPGC [FUNCTION (LAMBDA (PTR CNT COLL?)
                                 (COND
                                    ((AND (OR (NOT ONLYTYPES)
                                              (FMEMB (NTYPX PTR)
                                                     ONLYTYPES))
                                          (IGEQ CNT MINCNT))
                                     (printout NIL (COND
                                                     (COLL? '*)
                                                     (T '% ))
                                           CNT %,,)
                                     (LVLPRINT PTR NIL CARLVL CDRLVL)
                                     (COND
                                        (COLLECT [SETQ SELECTEDITEMS (CDR (FRPLACD SELECTEDITEMS
                                                                                   (CONS PTR NIL]
                                                                    ; Use RPLCONS to minimize refcnt operations

                                                  ))
                                     (add TOTALCNT 1)
                                     (COND
                                        (COLL? (add COLLCNT 1)))
                                     (COND
                                        ((IGEQ CNT \MAXHTCNT)  ; Means its a big ref count case
                                         (add MAXCNT 1]
                    (ILESSP MINCNT 1))
            (printout NIL TOTALCNT " items with reference cnt greater than or equal to " MINCNT T)
            (COND
               ((AND (ILESSP MINCNT \MAXHTCNT)
                     (NEQ MAXCNT 0))
                (printout NIL MAXCNT " items with overflowed reference cnt" T)))
            (printout NIL COLLCNT " collision entries" T)
            (RETURN (COND
                       (COLLECT (CDR RESULT))
                       (T FILE])])
```

(\GCENTRIES.BY.TYPE

```
[LAMBDA (MINREFCNT MINFRACTION)
  (DECLARE (SPECVARS MINREFCNT \#COLLISIONS \#OFENTRIES TYPETABLE))
                                                          ; Edited 26-Jan-87 16:46 by jop
  (OR MINREFCNT (SETQ MINREFCNT 0))
  (OR MINFRACTION (SETQ MINFRACTION 0.002))
  (PROG ((TYPECOUNTS (ARRAY (ADD1 \MaxTypeNumber)
                            'WORD 0 0))
         (TYPECOLLISIONS (ARRAY (ADD1 \MaxTypeNumber)
                                'WORD 0 0))
         (\#OFENTRIES 0)
         (\#COLLISIONS 0)
         (PRINTEDENTRIES 0)
         (PRINTEDCOLLISIONS 0)
         (MAXWIDTH 0)
         CNT FRAC)
        (DECLARE (SPECVARS \#OFENTRIES TYPECOUNTS \#COLLISIONS TYPECOLLISIONS))
        [\MAPGC (FUNCTION (LAMBDA (PTR REFCNT COLL?)
                             (PROG (TYPE)
                                   (COND
                                      ((IGEQ REFCNT MINREFCNT)
                                       (add \#OFENTRIES 1)
                                       (add (ELT TYPECOUNTS (NTYPX PTR))
                                            1)
                                       (COND
                                          (COLL? (add \#COLLISIONS 1)
                                                 (add (ELT TYPECOLLISIONS (NTYPX PTR))
                                                      1]
        [for I from 0 to \MaxTypeNumber bind N do (COND
```

```
                                                    ((IGREATERP (SETQ N (NCHARS (\TYPENAMEFROMNUMBER I)))
                                                                MAXWIDTH)
                                                     (SETQ MAXWIDTH N]
          (COND
             ((IGREATERP MINREFCNT 0)
              (printout T " with reference count at least " .P2 MINREFCNT)))
          (printout T T .FR MAXWIDTH "Type" "   all entries      collisions" T T)
          (for TYPE# from 0 to \MaxTypeNumber when (AND (NEQ (SETQ CNT (ELT TYPECOUNTS TYPE#))
                                                             0)
                                                        (FGREATERP (SETQ FRAC (FQUOTIENT CNT \#OFENTRIES))
                                                                   MINFRACTION))
             do (printout T .FR MAXWIDTH (OR (\TYPENAMEFROMNUMBER TYPE#)
                                             (CONCAT "Type " TYPE#))
                      .I7 CNT .F6.1 (FTIMES 100.0 FRAC)
                      "%%")
                (add PRINTEDENTRIES CNT)
                (COND
                   ([NOT (EQ 0 (SETQ CNT (ELT TYPECOLLISIONS TYPE#]
                    (add PRINTEDCOLLISIONS CNT)
                    (printout T .I10 CNT .F6.1 (FTIMES 100.0 (FQUOTIENT CNT \#COLLISIONS))
                           "%%")))
                (TERPRI T))
          (printout T .FR MAXWIDTH "All other types" .I7 (SETQ CNT (IDIFFERENCE \#OFENTRIES PRINTEDENTRIES))
                 .F6.1
                 (FTIMES 100.0 (FQUOTIENT CNT \#OFENTRIES))
                 "%%")
          (printout T .I10 (SETQ CNT (IDIFFERENCE \#COLLISIONS PRINTEDCOLLISIONS))
                 .F6.1
                 (FTIMES 100.0 (FQUOTIENT CNT \#COLLISIONS))
                 "%%" T)
          (printout T T .FR MAXWIDTH "Total" .I7 \#OFENTRIES .I17 \#COLLISIONS T T])
```

(\#COLLISIONS

```
[LAMBDA NIL                                                    (* JonL "28-Jan-84 04:20")
  (\GCSTATS.AUX '\#COLLISIONS])
```

(\#OVERFLOWS

```
[LAMBDA NIL                                                    (* JonL "28-Jan-84 04:20")
  (\GCSTATS.AUX '\#OVERFLOWS])
```

(\GCSTATS.AUX

```
[LAMBDA (\GCTYPE.AUX)                                          ; Edited 26-Jan-87 16:47 by jop
  (LET ((\#GCENTRIES 0)
        (\#GCLOSERS 0))
       (DECLARE (SPECVARS \#GCENTRIES \#GCLOSERS \GCTYPE.AUX))
       [\MAPGC (FUNCTION (LAMBDA (PTR REFCNT COLLISIONP)
                         (add \#GCENTRIES 1)
                         (SELECTQ \GCTYPE.AUX
                              (\#OVERFLOWS (COND
                                             ((IGEQ REFCNT \MAXHTCNT)
                                              (add \#GCLOSERS 1))))
                              (\#COLLISIONS (COND
                                              (COLLISIONP (add \#GCLOSERS 1))))
                              (SHOULDNT]
       (LIST \#GCENTRIES \#GCLOSERS (QUOTIENT (FLOAT \#GCLOSERS)
                                              \#GCENTRIES)
             (QUOTIENT (FLOAT \#GCENTRIES)
                    \HTMAINSIZE])
```

(\SEE-GC-ENTRY

```
[LAMBDA (OFFSET)                                               ; Edited 20-Oct-94 10:15 by sybalsky
  (LET ((ENTRY (\ADDBASE \HTMAIN (LLSH OFFSET 1)))
        POINTER LINK OVENTRY CNT)
       (COND
          ((fetch (GC EMPTY) of ENTRY)
           "EMPTY")
          [(NOT (fetch (GC LINKP) of ENTRY))
           (COND
              ((NEQ (fetch (GC CNT) of ENTRY)
                    0)
               (SETQ POINTER (\VAG2 (fetch (GC HIBITS) of ENTRY)
                                    (LLSH OFFSET 1)))
               (SETQ CNT (fetch (GC CNT) of ENTRY))
               [COND
                  ((>= CNT \MAXHTCNT)
                   (SETQ CNT (\GC.LOOKUP.BIGREFCNT POINTER]
               (CL:FORMAT T "ENTRY: ~O POINTER ~S CNT: ~A ~%" ENTRY POINTER CNT]
          (T (SETQ LINK (fetch (GC LINKPTR) of ENTRY))
             (do (SETQ OVENTRY (\ADDBASE \HTCOLL (LLSH LINK 1)))
                 (SETQ LINK (fetch (GC NXTPTR) of OVENTRY))
                 (COND
                    ((NEQ (SETQ CNT (fetch (GC CNT) of OVENTRY))
                          0)
```

```
                    (SETQ POINTER (\VAG2 (fetch (GC HIBITS) of OVENTRY)
                                         (LLSH OFFSET 1)))
                    (SETQ CNT (fetch (GC CNT) of OVENTRY))
                    [COND
                       ((>= CNT \MAXHTCNT)
                        (SETQ CNT (\GC.LOOKUP.BIGREFCNT POINTER]
                    (CL:FORMAT T "OVENTRY: ~O LINKPOINTER: ~S CNT: ~A LINK: ~A~%" OVENTRY POINTER CNT LINK))
                  )
                repeatuntil (EQ LINK 0])

)

;; Hacking free lists

(DEFINEQ

(\PRINTFREELIST
  [LAMBDA (TYPE DETAILS FILE)                                   ; Edited 19-Oct-94 12:27 by sybalsky
    (SETQ TYPE (\COERCETOTYPENUMBER TYPE))
    (SETQ FILE (\GETSTREAM FILE 'OUTPUT))
    (PROG ((SIZE (fetch DTDSIZE of (\GETDTD TYPE)))
           INFO TOTALPAGES MAXFREE TOTALLYFREE FREE)
          (CL:FORMAT FILE "Type ~S: " (\TYPENAMEFROMNUMBER TYPE))
          (if (EQ SIZE 0)
              then (printout FILE "not an allocated type" T T)
                   (RETURN))
          (if [AND (SETQ INFO (\SCANFREELIST TYPE DETAILS FILE))
                   (NOT (FIXP (CDAR INFO]
              then                                              ; Silly LISTP case
                   (printout FILE T (pop INFO)
                             T T)
                   (RETURN))
          (if (EQ (SETQ FREE (for X in INFO sum (CDR X)))
                  0)
              then (printout FILE "Free list is empty" T T)
                   (RETURN))
          (CL:FORMAT FILE "~D cells free~%Free list covers ~D Pages with ~D extra hops~%" FREE
                 [SETQ TOTALPAGES (LENGTH (for X in INFO do (pushnew $$VAL (CAR X]
                 (- (LENGTH INFO)
                    TOTALPAGES))
          (SETQ MAXFREE (IQUOTIENT \MDSIncrement SIZE))
          (SETQ TOTALLYFREE 0)
          (for PAIR in (SORT (APPEND INFO)
                             T)
             bind (PREVPAGE _ 0)
                  PREVFREE
             do (if [NOT (OR (EQ (CAR PAIR)
                                 PREVPAGE)
                             (EQ (CAR PAIR)
                                 (+ PREVPAGE 1]
                    then                                        ; not part of same page pair previously counted
                         (SETQ PREVPAGE (CAR PAIR))
                         (SETQ PREVFREE 0))
                (if (EQ (add PREVFREE (CDR PAIR))
                        MAXFREE)
                    then                                        ; completely free page pair
                         (add TOTALLYFREE 2)
                         (SETQ PREVPAGE 0)))
          (if (> TOTALLYFREE 0)
              then (CL:FORMAT FILE "~D pages are reclaimable~%" TOTALLYFREE))
          (COND
             (DETAILS (printout FILE "Details (page/#free):" T)
                      [for TAIL on (REVERSE INFO) bind (I _ 0)
                                                       (N _ (SUB1 (IQUOTIENT (LINELENGTH NIL FILE)
                                                                             12)))
                         do (printout FILE .TAB (ITIMES I 12)
                                          .I6.8
                                          (CAR (CAR TAIL))
                                          "/" .I3 (CDR (CAR TAIL)))
                            [COND
                               ((ASSOC (CAAR TAIL)
                                       (CDR TAIL))
                                (printout FILE '+]
                            (COND
                               ((> (add I 1)
                                   N)
                                (SETQ I 0]
                      (TERPRI FILE)))
          (TERPRI FILE)
          (RETURN TOTALLYFREE])

(\SHOWFREELISTS
  [LAMBDA (DETAILS FILE)                                        ; Edited  5-Feb-87 15:01 by bvm:
    (SETQ FILE (\GETSTREAM FILE 'OUTPUT))
    (for I from 2 to \MaxTypeNumber sum (OR (\PRINTFREELIST I DETAILS FILE)
                                            0)
```

```
       unless [OR (EQ I \LISTP)
                  (NULL (fetch DTDFREE of (\GETDTD I]
       finally (RETURN (CONCAT $$VAL " total free pages"])
```

(\SCANFREELIST

```
  [LAMBDA (TYPE DETAILS FILE)                                   ; Edited 19-Oct-94 12:27 by sybalsky
    ;; Scans free list of type TYPE and returns a list of pairs (page# . freecount) indicating how many free items are on each page
    (PROG (RESULT FREE (LASTPAGE -1)
           LASTPAGECOUNT THISPAGE)
          (SETQ TYPE (\COERCETOTYPENUMBER TYPE))
          [COND
             ((EQ TYPE \LISTP)
              (RETURN (CONS (LIST "LISTP scan not implemented"]
          (SETQ FREE (fetch DTDFREE of (\GETDTD TYPE)))
          (while FREE do (AND DETAILS (PRINT FREE FILE))
                         (SETQ THISPAGE (fetch (POINTER PAGE#) of FREE))
                         (COND
                            ((IEQP THISPAGE LASTPAGE)
                             (add LASTPAGECOUNT 1))
                            (T [COND
                                  (LASTPAGE (push RESULT (CONS LASTPAGE LASTPAGECOUNT]
                               (COND
                                  ((NEQ (NTYPX FREE)
                                        TYPE)
                                   (push RESULT (LIST "Bad free list at" (\HILOC FREE)
                                                      (\LOLOC FREE)))
                                   (RETURN)))
                               (SETQ LASTPAGE THISPAGE)
                               (SETQ LASTPAGECOUNT 1)))
                         (SETQ FREE (fetch FREELINK of FREE)))
          [COND
             (LASTPAGE (push RESULT (CONS LASTPAGE LASTPAGECOUNT]
          (RETURN RESULT])
```

(\ISONFREELIST

```
  [LAMBDA (OBJECT)                                              (* bvm%: " 6-Dec-83 17:44")
    (PROG ((TYPE (NTYPX OBJECT))
           FREE)
          (COND
             ((EQ TYPE \LISTP)
              (RETURN "LISTP scan not implemented")))
          (SETQ FREE (fetch DTDFREE of (\GETDTD TYPE)))
          (RETURN (while FREE do (COND
                                    ((EQ OBJECT FREE)
                                     (RETURN T)))
                                 (SETQ FREE (fetch FREELINK of FREE])

)

(DEFINEQ
```

(PFL

```
  [NLAMBDA X                                                    (* bvm%: " 5-Dec-83 18:46")
    (\PRINTFREELIST X T])
```

(SFL

```
  [NLAMBDA X                                                    (* bvm%: " 5-Dec-83 18:47")
    (\SORTFREELIST X)
    (\PRINTFREELIST X T])

)

(DEFINEQ
```

(\COLLECTINUSE

```
  [LAMBDA (TYPE PRED)                                           ; Edited  5-Feb-87 15:46 by bvm:
    (SETQ TYPE (\COERCETOTYPENUMBER TYPE))
    (RPTQ 20 (RECLAIM))
    (RESETFORM (RECLAIMMIN MAX.SMALLP)
           (UNINTERRUPTABLY
               (PROG ((HASHTABLE (\SORTFREELIST TYPE T))
                      (SIZE (fetch DTDSIZE of (\GETDTD TYPE)))
                      RESULT FIRSTFREE LASTFREE HASHENT LASTPAGE LIMIT)
                     (OR HASHTABLE (RETURN))
                     (OR (EVENP SIZE)
                         (SHOULDNT "Odd size?"))
                     (COND
                        ((.ALLOCATED.PER.PAGE. SIZE)
                         (SETQ LASTPAGE (SUB1 \PagesPerMDSUnit))
                         (SETQ LIMIT WORDSPERPAGE))
                        (T (SETQ LASTPAGE 0)
                           (SETQ LIMIT \MDSIncrement)))
                     [for MDSPAGE# from 0 by \PagesPerMDSUnit while (<= MDSPAGE# \MAXVMPAGE)
```

```
                    when (EQ (MDSTYPE# MDSPAGE#)
                             TYPE)
                    do [COND
                           ([SETQ FIRSTFREE (COND
                                                ((SETQ HASHENT (OR (\SFLHASHLOOKUP MDSPAGE# HASHTABLE)
                                                                   (\SFLHASHLOOKUP (LOGOR MDSPAGE# 1)
                                                                                   HASHTABLE)))
                                                 (\VAG2 (FOLDLO MDSPAGE# PAGESPERSEGMENT)
                                                        (fetch HASHFIRSTOFFSET of HASHENT]
                            (SETQ LASTFREE (fetch HASHLASTFREE
                                              of (OR (AND (EVENP (fetch HASHPAGE# of HASHENT))
                                                          (\SFLHASHLOOKUP (LOGOR MDSPAGE# 1)
                                                                          HASHTABLE))
                                                     HASHENT]
                        ;; Now collect all pointers not on free list.  This code parallels \INITMDSPAGE
                        (for N from 0 to LASTPAGE
                           do (for (DISP _ 0) while (<= (add DISP SIZE)
                                                        LIMIT)
                                 as (DATUMBASE _ (create POINTER
                                                         PAGE# _ (IPLUS N MDSPAGE#)))
                                 by (\ADDBASE DATUMBASE SIZE)
                                 when (AND (OR (NOT FIRSTFREE)
                                               (for (X _ FIRSTFREE) by (fetch FREELINK of X)
                                                  never (EQ X DATUMBASE) repeatuntil (EQ X LASTFREE)))
                                           (OR (NOT PRED)
                                               (CL:FUNCALL PRED DATUMBASE)))
                                 do (push RESULT DATUMBASE]
                    (RETURN RESULT)))])
```

(**SORTFREELIST**

```
  [LAMBDA (TYPE FLG READONLY)                                   ; Edited  5-Feb-87 15:47 by bvm:
    (SETQ TYPE (\COERCETOTYPENUMBER TYPE))
    (PROG ((DTD (\GETDTD TYPE))
           NPAGES HASHTABLE HASHENT HSIZE NEXTFREE NEXTPAGE LASTPAGE FIRSTFREE LASTFREE OTHERLASTFREE
           PREVPAGELASTFREE PROBE MASK)
          (COND
             ((EQ TYPE \LISTP)
              (RETURN)))
          (SETQ NPAGES (ITIMES (for I from 0 to \MAXVMPAGE by 2 count (EQ (MDSTYPE# I)
                                                                          TYPE))
                               2))
          (SETQ HSIZE (FIX (TIMES NPAGES 1.4)))                 ; Good size of hashtable for hashing pages of this type into
          (SETQ HSIZE (find I from 8 by I suchthat (IGREATERP I HSIZE)))
                                                                ; Get a power of 2
          (SETQ HASHTABLE (\ALLOCBLOCK (ITIMES HSIZE 2)))
          (replace HASHMASK of HASHTABLE with (SUB1 (ITIMES HSIZE 4)))
          (SETQ NEXTFREE (fetch DTDFREE of DTD))
          [do (COND
                 ((NEQ (SETQ NEXTPAGE (fetch (POINTER PAGE#) of NEXTFREE))
                       LASTPAGE)                                ; Cell on a new page
                  [COND
                     ((AND NEXTFREE (NEQ (NTYPX NEXTFREE)
                                         TYPE))
                      (RETURN (RAID "Bad free list" NEXTFREE]
                  (COND
                     (LASTPAGE                                  ; Hash LASTPAGE and see if we have already seen cells free on
                                                                ; this page
                           (SETQ HASHENT (\SFLHASHLOOKUP LASTPAGE HASHTABLE T))
                           (COND
                              [(SETQ OTHERLASTFREE (fetch HASHLASTFREE of HASHENT))
                                                                ; Yes, we have seen others.  Link this section of the free list into
                                                                ; it
                               (COND
                                  ((EQ (fetch FREELINK of OTHERLASTFREE)
                                       FIRSTFREE)
                                   ;; Aha, already in order.  This happens when we have a sequence LASTPAGE -> x ...  -> LASTPAGE
                                   ;; where everything in between the two LASTPAGE's got moved to earlier in the freelist
                                   (SETQ PREVPAGELASTFREE LASTFREE))
                                  ((NOT READONLY)
                                   (UNINTERRUPTABLY
                                       [replace FREELINK of OTHERLASTFREE
                                          with (PROG1 FIRSTFREE
                                                      (replace FREELINK of (OR PREVPAGELASTFREE (RETURN (RAID "No
                                                                                                              PREVPA
GELASTFREE")))
                                                         with NEXTFREE)
                                                      (replace FREELINK of LASTFREE with (fetch FREELINK of
                                                                                                OTHERLASTFREE
                                                                                            ))))])
                                  (T (replace HASHFIRSTOFFSET of HASHENT with (\LOLOC FIRSTFREE))
                                     (SETQ PREVPAGELASTFREE LASTFREE)))
                               (replace HASHLASTFREE of HASHENT with LASTFREE)))
                     (OR (SETQ FIRSTFREE NEXTFREE)
                         (RETURN))
```

```
                (SETQ LASTPAGE NEXTPAGE)))
            (SETQ NEXTFREE (fetch FREELINK of (SETQ LASTFREE NEXTFREE]
        (SETQ LASTPAGE (SETQ PREVPAGELASTFREE))
        (SETQ NEXTFREE (fetch DTDFREE of DTD))
```

;; Now take a quick second pass to link all odd pages immediately after the corresponding even pages. Might possibly have done this in the ;; previous loop, but the logic gets pretty messy

```
      [do (COND
             ((NEQ (SETQ NEXTPAGE (fetch (POINTER PAGE#) of NEXTFREE))
                   LASTPAGE)                                  ; Cell on a new page
              [COND
                 (LASTPAGE (COND
                              [(AND (ODDP LASTPAGE)
                                    (SETQ HASHENT (\SFLHASHLOOKUP (LOGXOR LASTPAGE 1)
                                                                  HASHTABLE))
                                    (NEQ (fetch FREELINK of (SETQ OTHERLASTFREE (fetch HASHLASTFREE
                                                                                   of HASHENT)))
                                         FIRSTFREE))          ; There is an entry for our partner even page, and it is not
                                                              ; immediately followed by its odd partner
                               (OR READONLY
                                   (UNINTERRUPTABLY
                                      [replace FREELINK of OTHERLASTFREE
                                         with (PROG1 FIRSTFREE
                                                     (COND
                                                        (PREVPAGELASTFREE (replace FREELINK of PREVPAGELASTFREE
                                                                             with NEXTFREE))
                                                        (T (OR (EQ FIRSTFREE (fetch DTDFREE of DTD))
                                                               (RAID "No PREVPAGELASTFREE"))
                                                           (replace DTDFREE of DTD with NEXTFREE)))
                                                     (replace FREELINK of LASTFREE with (fetch FREELINK
                                                                                           of OTHERLASTFREE)))]))]
                              (T (SETQ PREVPAGELASTFREE LASTFREE]
              (OR (SETQ FIRSTFREE NEXTFREE)
                  (RETURN))
              (SETQ LASTPAGE NEXTPAGE)))
          (SETQ NEXTFREE (fetch FREELINK of (SETQ LASTFREE NEXTFREE]
        (RETURN (AND FLG HASHTABLE])
```

(\SFLHASHLOOKUP

```
  [LAMBDA (PAGE# HASHTABLE INSERT)                            (* JonL "28-Dec-84 19:33")
    (bind (MASK _ (fetch HASHMASK of HASHTABLE))
          PROBE HASHENT first (SETQ PROBE (LOGAND (LLSH PAGE# 2)
                                                  MASK))
       do [COND
             ((IEQP (fetch HASHPAGE# of (SETQ HASHENT (\ADDBASE HASHTABLE PROBE)))
                    PAGE#)
              (RETURN HASHENT))
             ((EQ 0 (fetch HASHPAGE# of HASHENT))
              (RETURN (COND
                         (INSERT (replace HASHPAGE# of HASHENT with PAGE#)
                                 HASHENT]
          (SETQ PROBE (LOGAND (IPLUS PROBE 4)
                              MASK])
```

)

;; finding circularities

```
(DEFINEQ
```

(\SHOWCIRCULARITY

```
  [LAMBDA (OBJECT MAXLEVEL)                                   (* bvm%: "13-Dec-83 12:57")
    (DECLARE (SPECVARS CIRCLEHASH OBJECT MAXLEVEL))
    (PROG [(CIRCLEHASH (LIST (HARRAY 100]
          (OR (AND (FIXP MAXLEVEL)
                   (IGREATERP MAXLEVEL 0))
              (SETQ MAXLEVEL 1000))
          (\SHOWCIRCULARITY1 OBJECT])
```

(\SHOWCIRCULARITY1

```
  [LAMBDA (OBJ PATH)                                          (* bvm%: "13-Dec-83 12:09")
    (DECLARE (USEDFREE OBJECT CIRCLEHASH MAXLEVEL))
    (COND
       ((AND (EQ OBJ OBJECT)
             PATH)
        (\SHOWCIRCULARPATH PATH))
       (T (PROG ((TYPE (NTYPX OBJ))
                 PTRS B)
                (SELECTC TYPE
                     (\LISTP (push PATH OBJ)
                             (\SHOWCIRCULARLIST (CAR OBJ)
                                    PATH MAXLEVEL)
                             (\SHOWCIRCULARLIST (CDR OBJ)
                                    PATH MAXLEVEL))
                     (\STRINGP                                ; No circularity possible, although it does have one pointer field
```

```
                        )
                (0 [COND
                     ((AND (type? ARRAYBLOCK OBJ)
                           [IEQ \ArrayBlockPassword (fetch PASSWORD of (SETQ B (\ADDBASE OBJ (IMINUS
                                                                                \ArrayBlockHeaderWords
                                                                                ]
                           (fetch (ARRAYBLOCK INUSE) of B)
                           (EQ (fetch (ARRAYBLOCK GCTYPE) of B)
                               PTRBLOCK.GCT)
                           (NOT (GETHASH OBJ CIRCLEHASH))) ; B points to arrayblock header, OBJ to first and subsequent data
                                                           ; words
                      (PUTHASH OBJ T CIRCLEHASH)
                      (push PATH OBJ)
                      (for old OBJ (TRAILER _ (fetch (ARRAYBLOCK TRAILER) of B))
                         by (\ADDBASE OBJ WORDSPERCELL) until (EQ OBJ TRAILER)
                         do (\SHOWCIRCULARITY1 (\GETBASEPTR OBJ 0)
                                   PATH])
                (COND
                   ((AND (SETQ PTRS (fetch DTDPTRS of (\GETDTD TYPE)))
                         (NOT (GETHASH OBJ CIRCLEHASH)))
                    (PUTHASH OBJ T CIRCLEHASH)
                    (push PATH OBJ)
                    (for I in PTRS do (\SHOWCIRCULARITY1 (\GETBASEPTR OBJ I)
                                              PATH])
```

(\SHOWCIRCULARLIST

```
  [LAMBDA (LST PATH DEPTH)                                  (* bvm%: " 6-Dec-83 16:53")
    (DECLARE (USEDFREE OBJECT))
    (COND
       ((NLISTP LST)
        (\SHOWCIRCULARITY1 LST PATH))
       ((EQ LST OBJECT)
        (\SHOWCIRCULARITY PATH))
       ((NEQ DEPTH 0)
        (\SHOWCIRCULARLIST (CAR LST)
               PATH
               (SUB1 DEPTH))
        (\SHOWCIRCULARLIST (CDR LST)
               PATH
               (SUB1 DEPTH])
```

(\SHOWCIRCULARPATH

```
  [LAMBDA (PATH)                                            (* bvm%: " 6-Dec-83 16:39")
    (TERPRI T)
    [for X in (REVERSE (CONS OBJECT PATH)) bind PREFIX do (COND
                                                             (PREFIX (PRIN1 " -> " T))
                                                             (T (SETQ PREFIX T)))
                                                          (COND
                                                             ((LISTP X)
                                                              (LVLPRIN2 X T 1 3))
                                                             (T (PRIN2 X]
    (TERPRI T])

)

;; special window storage leak finder

(DEFINEQ
```

(\SHOW.CLOSED.WINDOWS

```
  [LAMBDA NIL                                               (* bvm%: "11-Oct-84 21:42")
    (for (TAIL _ (\COLLECTINUSE 'WINDOW)) while TAIL bind (OPEN _ (OPENWINDOWS))
                                                          W MAIN
       unless (\WINDOW.ACCOUNTED.FOR? (SETQ W (pop TAIL))) sum (OPENW W)
                                                               (CURSORPOSITION '(0 . 0)
                                                                      W)
                                                               (if (MOUSECONFIRM "Click LEFT to close window,
                                                                         RIGHT to save" T)
                                                                   then (CLOSEW W)
                                                                 elseif (MOUSECONFIRM "Find pointers?  Click
                                                                            LEFT to search, RIGHT to leave
                                                                            window open and go on" T)
                                                                   then (CLOSEW W)
                                                                        (RPTQ 10 (RECLAIM))
                                                                        (\FINDPOINTER W))
                                                               1])
```

(\WINDOW.ACCOUNTED.FOR?

```
  [LAMBDA (WINDOW)                                          (* bvm%: "30-Jul-84 14:57")
    (OR (OPENWP WINDOW)
        (OPENWP (WINDOWPROP WINDOW 'ICONWINDOW))
        (OPENWP (WINDOWPROP WINDOW 'ICONFOR))
        (PROG [(MAIN (WINDOWPROP WINDOW 'MAINWINDOW]
              (RETURN (AND MAIN (\WINDOW.ACCOUNTED.FOR? MAIN])
```

```
)

;; Brute force search for raw pointers

(DEFINEQ

(\FINDPOINTER
  [LAMBDA (PTR COLLECT/INSPECT? ALLFLG MARGIN ALLBACKFLG)        ; Edited 13-Mar-87 14:55 by bvm:
    (DECLARE (SPECVARS MARGIN REFSFOUND COLLECT/INSPECT?))
    (OR MARGIN (SETQ MARGIN 0))
    (PROG ((*PRINT-BASE* 10)
           (REFCNT (\REFCNT PTR))
           (SAFESEGMENTS (SELECTQ ALLFLG
                                  ((NIL :STACK)
                                      [LIST (\HILOC \FPTOVP)
                                            (\HILOC \PAGEMAP)
                                            (\HILOC \PageMapTBL)
                                            (\HILOC \AtomHashTable)
                                            (\HILOC \PNPSPACE)
                                            (ADD1 (\HILOC \PNPSPACE))
                                            (\HILOC \SMALLPOSPSPACE)
                                            (\HILOC \SMALLNEGSPACE)
                                            (\HILOC \HTMAIN)
                                            (\HILOC \HTCOLL)
                                            (\HILOC (fetch BITMAPBASE of (SCREENBITMAP])
                                  NIL))
           (STACKSEG (\HILOC \STACKSPACE))
           (REFSFOUND 0)
           RESULT ATOMSEGMENTS SEGBASE POINTERSOURCE)
          (COND
             ((OR (NEQ REFCNT 1)
                  (NOT ALLBACKFLG))
              (printout T .TAB0 MARGIN "Reference count = " REFCNT T)))
          (COND
             ((AND (EQ REFCNT 0)
                   (NOT ALLFLG))                                ; if zero ref count, only reasonable hope is to find it on stack
              (\FINDPOINTER.STACK PTR)
              (GO DONE)))
          (IF (NULL ALLFLG)
              THEN                                              ; Don't scan stack unless allflg = :stack
                   (PUSH SAFESEGMENTS (\HILOC \STACKSPACE)))
          [for DTDPAGESONLY in '(T NIL)
             do                                                 ; More likely to find on typed pages first
             (for SEGMENT from 1 to \MAXVMSEGMENT unless (FMEMB SEGMENT SAFESEGMENTS)
                do
                (SETQ SEGBASE (\VAG2 SEGMENT 0))
                (for PAGEINSEG from 0 to (SUB1 PAGESPERSEGMENT) as PAGE# from (UNFOLD SEGMENT PAGESPERSEGMENT)
                   as (PAGEBASE _ SEGBASE) by (\ADDBASE PAGEBASE WORDSPERPAGE) bind TYPE DTD STR
                   when [COND
                           [DTDPAGESONLY (AND (NEQ (SETQ TYPE (NTYPX PAGEBASE))
                                                   0)
                                              (fetch DTDPTRS of (SETQ DTD (\GETDTD TYPE]
                           (T (AND (EQ (SETQ TYPE (NTYPX PAGEBASE))
                                       0)
                                   (NEQ (\LOOKUPPAGEMAP PAGE#)
                                        0)
                                   (OR (NEQ SEGMENT STACKSEG)
                                       (PROGN                   ; Don't look at released stack pages, even though they exist in
                                                                ; the vmem -- could get stack fault
                                              (ILESSP (\LOLOC PAGEBASE)
                                                      (fetch (IFPAGE EndOfStack) of \InterfacePage]
                   do                                           ; Page exists and might contain pointers
                      (to CELLSPERPAGE as (BASE _ PAGEBASE) by (\ADDBASE BASE WORDSPERCELL)
                         when (EQ (\GETBASEPTR BASE 0)
                                  PTR)
                         do (COND
                               ((SETQ POINTERSOURCE
                                  (SELECTC TYPE
                                      (0 (COND
                                            ([SETQ STR (CADR (ASSOC (FLOOR SEGMENT 2)
                                                                    (OR ATOMSEGMENTS
                                                                        (SETQ ATOMSEGMENTS
                                                                         (LIST (LIST (\HILOC \VALSPACE)
                                                                                     "value")
                                                                               (LIST (\HILOC \DEFSPACE)
                                                                                     "function definition")
                                                                               (LIST (\HILOC \PLISTSPACE)
                                                                                     "property list"]
                                               (\FINDPOINTER.NEWITEM T)
                                               (printout T "as " STR " of atom " .P2
                                                      [SETQ STR (\INDEXATOMPNAME (+ (LRSH (\LOLOC BASE)
                                                                                          1)
                                                                                    (IF (ODDP SEGMENT)
                                                                                        THEN
                                                                                                    ; Second generation of symbols...
                                                                                        (LLSH 1 15)
```

```
                                                                    ELSE 0]
                                                  T)
                                               STR)
                                              ((EQ SEGMENT STACKSEG)
                                                                    ; Interpret stack
                                               (\FINDPOINTER.FOUND.ON.STACK BASE))
                                              (T (\FINDPOINTER.FOUND BASE)
                                                 NIL)))
                                         (\LISTP (\FINDPOINTER.LISTP BASE ALLBACKFLG ALLFLG))
                                         (\FINDPOINTER.TYPE BASE DTD ALLFLG)))
                                                                    ; Accounted for a valid reference
                                    (COND
                                       (COLLECT/INSPECT? (push RESULT POINTERSOURCE)))
                                    (COND
                                       ((AND (NOT ALLFLG)
                                             (EQ REFCNT REFSFOUND))
                                        (GO DONE]
      DONE
           (COND
              ((AND COLLECT/INSPECT? (NEQ COLLECT/INSPECT? 'COLLECT))
               (INSPECT RESULT)
               (SETQ RESULT NIL)))
           (RETURN (OR RESULT REFSFOUND])
```

(\FINDPOINTERS.OF.TYPE

```
  [LAMBDA (TYPE FILTER)                                             (* bvm%: "28-Jun-84 11:51")
    (for (TAIL _ (\COLLECTINUSE TYPE)) while TAIL bind PTR (FILTERFNP _ (FNTYP FILTER)) declare (SPECVARS PTR)
       when [PROGN (SETQ PTR (pop TAIL))
                   (OR (NULL FILTER)
                       (COND
                          (FILTERFNP (APPLY* FILTER PTR))
                          (T (EVAL FILTER]
       do ;; This odd control structure so that we get rid of the extra pointer from the list returned by \COLLECTINUSE
          (RECLAIM)
          (RECLAIM)
          (\FINDPOINTER (PRINT PTR T))
          (TERPRI T])
```

(\FINDPOINTER.FOUND

```
  [LAMBDA (BASE MSG)                                                (* bvm%: "20-Jan-84 16:22")
    (\FINDPOINTER.NEWITEM NIL)
    (COND
       (MSG (printout T MSG)))
    (printout T "at location " .I2.8 (\HILOC BASE)
           '%, .I6.8 (\LOLOC BASE)
           T])
```

(\FINDPOINTER.NEWITEM

```
  [LAMBDA (COUNTIT)
    (DECLARE (USEDFREE MARGIN REFSFOUND))                           (* bvm%: "20-Jan-84 16:13")
    (printout T .TAB0 MARGIN)
    (IF COUNTIT
        THEN (printout T (add REFSFOUND 1)
                       ". "))
    (printout T "Found ")
    COUNTIT])
```

(\FINDPOINTER.LISTP

```
  [LAMBDA (BASE ALLBACKFLG ALLFLG)                                  ; Edited 2-Feb-87 12:46 by bvm:
    (DECLARE (USEDFREE REFSFOUND MARGIN))
    (PROG ((PAGEBASE (fetch (POINTER PAGEBASE) of BASE))
           (WORDOFFSET (fetch (POINTER WORDINPAGE) of BASE))
           (NEWMARGIN MARGIN)
           CDRCODE TYPE DESIREDCDRCODE)
          (COND
             ((\FINDPOINTER.LISTP.FREE PAGEBASE WORDOFFSET)
              (if ALLFLG
                  then (\FINDPOINTER.FOUND BASE "in freed list cell "))
              (RETURN NIL)))
          (SETQ DESIREDCDRCODE (LRSH WORDOFFSET 1))
      LP                                                            ; Track this listp back on page
          (for I from 2 to (IDIFFERENCE WORDSPERPAGE WORDSPERCELL) by WORDSPERCELL
             when (AND (EQ (LOGAND (SETQ CDRCODE (fetch (LISTP CDRCODE) of (\ADDBASE PAGEBASE I)))
                                   127)
                           DESIREDCDRCODE)
                       (NOT (\FINDPOINTER.LISTP.FREE PAGEBASE I)))
             do [OR TYPE (SETQ TYPE (COND
                                       ((IGREATERP CDRCODE \CDR.MAXINDIRECT)
                                                                    ; CDR on page
                                        "an element")
                                       (T                           ; CDR indirect on page
                                          "a tail"]
```

```
               (SETQ BASE (\ADDBASE PAGEBASE I))
               (SETQ DESIREDCDRCODE (LRSH I 1))
               (GO LP))
            (COND
               ((AND (NULL TYPE)
                     (EQ 0 (\GETBASEBYTE BASE 0)))
                ;; What we found was a full indirect cell pointing at a LISTP cell that (probably) someone told us to chase.  So just chase it explicitly
                ;; now.
                (add REFSFOUND 1)
                (GO SEARCHMORE))
               [ALLBACKFLG (COND
                              ((EQ (\REFCNT BASE)
                                   1)
                               (add ALLBACKFLG 1)
                               (add REFSFOUND 1)
                               (printout T '%.)
                               (GO SEARCHMORE))
                              (T (\FINDPOINTER.NEWITEM T)
                                 (printout T "somewhere inside list "]
               (T (\FINDPOINTER.NEWITEM T)
                  (printout T "as " (OR TYPE 'CAR)
                            " of list ")))
            (LVLPRINT BASE T 1 3)
            (TAB NEWMARGIN 0 T)
            (SELECTQ (PROG1 [ASKUSER DWIMWAIT 'N "Shall I search for pointers to this list? "
                                    '((Y "es")
                                      (N "o")
                                      (A "ll the way back"]
                            (TERPRI T))
                (N (RETURN BASE))
                (A (SETQ ALLBACKFLG 1))
                NIL)
            (add NEWMARGIN 3)
         SEARCHMORE
            (RETURN (COND
                       ((NLISTP (SETQ TYPE (\FINDPOINTER BASE 'COLLECT NIL NEWMARGIN ALLBACKFLG)))
                        BASE)
                       ((CDR TYPE)
                        TYPE)
                       (T (CAR TYPE])
```

(**FINDPOINTER.LISTP.FREE**

```
[LAMBDA (PAGEBASE WORDOFFSET)                                   (* bvm%: "20-Jan-84 11:39")
  ;; True if the cell at WORDOFFSET after PAGEBASE is a free list cell
  (for (FREE _ (fetch (CONSPAGE NEXTCELL) of PAGEBASE)) by (fetch (LISTP CDRCODE) of (\ADDBASE PAGEBASE FREE))
     as I to (fetch (CONSPAGE CNT) of PAGEBASE) thereis (EQ FREE WORDOFFSET])
```

(**FINDPOINTER.TYPE**

```
[LAMBDA (BASE DTD ALLFLG)                                       (* bvm%: " 3-Jan-85 21:21")
  (DECLARE (USEDFREE MARGIN COLLECT/INSPECT?))
  (PROG ((SIZE (fetch DTDSIZE of DTD))
         WORDINPAGEGROUP SEGMENTBASE VALIDPOINTERP FREEP ORIGIN OFFSET OBJECT TYPENAME DEC)
        [COND
           ((.ALLOCATED.PER.PAGE. SIZE)
            (SETQ WORDINPAGEGROUP (IMOD (\LOLOC BASE)
                                        WORDSPERPAGE))
            (SETQ SEGMENTBASE (FLOOR (\LOLOC BASE)
                                     WORDSPERPAGE)))
           (T (SETQ WORDINPAGEGROUP (IMOD (\LOLOC BASE)
                                          \MDSIncrement))
              (SETQ SEGMENTBASE (FLOOR (\LOLOC BASE)
                                       \MDSIncrement]
        (SETQ ORIGIN (ITIMES (IQUOTIENT WORDINPAGEGROUP SIZE)
                             SIZE))
        (SETQ OBJECT (\VAG2 (\HILOC BASE)
                            (IPLUS SEGMENTBASE ORIGIN)))
        (SETQ VALIDPOINTERP (MEMB (SETQ OFFSET (IDIFFERENCE WORDINPAGEGROUP ORIGIN))
                                  (fetch DTDPTRS of DTD)))
        (IF (AND (SETQ FREEP (\ISONFREELIST OBJECT))
                 (NOT ALLFLG))
            THEN (RETURN))
        (\FINDPOINTER.NEWITEM (AND VALIDPOINTERP (NOT FREEP)))
        (printout T "at offset " .P2 OFFSET)
        (COND
           ([SETQ DEC (OR [RECLOOK (SETQ TYPENAME (\VAG2 0 (fetch DTDNAME of DTD]
                          (find X in SYSTEMRECLST suchthat (EQ (CADR X)
                                                               TYPENAME]
            (\FINDPOINTER.INTERPRET.RECORD DEC OFFSET)))
        (COND
           ((NOT VALIDPOINTERP)
            (printout T " (not a pointer field)")))
        (printout T " in" (COND
                             (FREEP " freed ")
```

```
                              (T " "))
                    "object " OBJECT T)
          (RETURN (COND
                      ((AND VALIDPOINTERP (NOT FREEP))
                       (OR (SELECTQ (PROG1 [ASKUSER DWIMWAIT 'N "Shall I search for pointers to this object? "
                                                  '((Y "es")
                                                    (N "o")
                                                    (I "nspect it"]
                                           (TERPRI T))
                                  (Y (\FINDPOINTER OBJECT COLLECT/INSPECT? NIL (IPLUS MARGIN 3)))
                                  (N NIL)
                                  (PROGN (INSPECT OBJECT)
                                         NIL))
                           OBJECT])
```

(\FINDPOINTER.INTERPRET.RECORD

```
[LAMBDA (DEC OFFSET)                                                    (* bvm%: "21-Feb-86 12:11")
```

;;; Figures out the field name associated with word offset OFFSET in record declaration DEC. Is simpleminded, gives up easily

```
    (for FIELD in (CADDR DEC) bind (N _ 0)
                                   BITCOUNT
       unless (EQ (CAR (LISTP FIELD))
                  '*)
       do (SELECTQ (COND
                       ((LISTP FIELD)
                        (CADR FIELD))
                       (T 'POINTER))
                ((XPOINTER POINTER FULLXPOINTER)
                     (COND
                        (BITCOUNT (COND
                                     ((OR (AND (LISTP FIELD)
                                               (EQ (CADR FIELD)
                                                   'FULLXPOINTER))
                                          (IGREATERP BITCOUNT BITSPERBYTE))
                                      (RETURN)))
                                  (SETQ BITCOUNT NIL)))
                     (COND
                        ((EQ N OFFSET)
                         (printout T " (" (OR (CAR (LISTP FIELD))
                                              FIELD)
                                 ")")
                         (RETURN)))
                     (add N WORDSPERCELL))
                (WORD [COND
                         (BITCOUNT (COND
                                      ((EQ BITCOUNT BITSPERWORD)
                                       (SETQ BITCOUNT NIL)
                                       (add N 1))
                                      (T (RETURN]
                       (add N 1))
                (BYTE (COND
                         ((NOT BITCOUNT)
                          (SETQ BITCOUNT BITSPERBYTE))
                         ((IGREATERP BITCOUNT BITSPERBYTE)
                          (RETURN))
                         (T (add N 1)
                            (SETQ BITCOUNT NIL))))
                ((FLAG BITS)
                     (SETQ BITCOUNT (IPLUS (OR BITCOUNT 0)
                                           (OR (CADDR FIELD)
                                               1))))
                (RETURN])
```

(\FINDPOINTER.STACK

```
[LAMBDA (PTR)
  (FOR I FROM 0 TO (fetch (IFPAGE EndOfStack) of \InterfacePage) WHEN (EQ (\GETBASEPTR (STACKADDBASE I)
                                                                                      0)
                                                                          PTR)
    DO (\FINDPOINTER.FOUND.ON.STACK (STACKADDBASE I])
```

(\FINDPOINTER.PARSE.STACK

```
[LAMBDA (BASE)                                                          (* bvm%: "18-AUG-83 12:05")
  (PROG ((SCANPTR (fetch StackBase of \InterfacePage))
         (EASP (fetch EndOfStack of \InterfacePage))
         (TARGET (\LOLOC BASE)))
        (if (< TARGET SCANPTR)
            then (RETURN "System context"))
    SCAN
        [SELECTC (fetch (STK FLAGS) of SCANPTR)
            ((LIST \STK.FSB \STK.GUARD)                                 ; free block or guard block all the same to us
                 (if (< TARGET (add SCANPTR (fetch (FSB SIZE) of SCANPTR)))
                     then (RETURN "Free block")))
            (\STK.FX                                                    ; frame extension
```

```
                        (if (< TARGET (fetch (FX NEXTBLOCK) of SCANPTR))
                            then (RETURN (fetch (FX FRAMENAME) of SCANPTR)))
                        (if (>= SCANPTR (SETQ SCANPTR (fetch (FX NEXTBLOCK) of SCANPTR)))
                            then                                          ; avoid looping in malformed stack
                                 (RETURN)))
                (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))                        ; stack is garbage, can't parse
                          (RETURN))
                         (T (add SCANPTR WORDSPERCELL)
                            (if (< TARGET SCANPTR)
                                then                                      ; in this basic frame--locate fx to get name
                                     (RETURN (if (EQ (fetch (STK FLAGS) of SCANPTR)
                                                     \STK.FX)
                                                 then (fetch (FX FRAMENAME) of SCANPTR)
                                               else "Argument"]
           (COND
              ((IGREATERP SCANPTR EASP)
               (RETURN)))
           (GO SCAN])
```

(**FINDPOINTER.FOUND.ON.STACK**

```
  [LAMBDA (BASE)
    (LET ((WHERE (\MISCAPPLY* (FUNCTION \FINDPOINTER.PARSE.STACK)
                              BASE)))
         (IF (OR (NULL WHERE)
                 (NOT (STRPOS "FINDPOINTER" WHERE)))
             THEN                                                         ; be sure to filter out our own bindings!
                  (\FINDPOINTER.NEWITEM NIL)
                  (CL:FORMAT *TERMINAL-IO* "in stack cell ~O " (\LOLOC BASE))
                  (IF WHERE
                      THEN (CL:FORMAT *TERMINAL-IO* (IF (STRINGP WHERE)
                                                        THEN "(~A)"
                                                      ELSE "(~S)")
                                                    WHERE))
                  (TERPRI T))
         NIL])

)

(DECLARE%: EVAL@COMPILE DONTCOPY

(DECLARE%: EVAL@COMPILE

(PUTPROPS \COERCETOTYPENUMBER MACRO (OPENLAMBDA (TYPE)
                                      (OR (FIXP TYPE)
                                          (\TYPENUMBERFROMNAME TYPE)
                                          (ERROR "Not a valid type" TYPE))))

(PUTPROPS .ALLOCATED.PER.PAGE. MACRO (OPENLAMBDA (SIZE)                   (* Maybe change this some day to a fetch of a flag from the DTD)
                                       (AND (IGEQ (LISPVERSION)
                                                  37384)
                                            (ILESSP (IREMAINDER WORDSPERPAGE SIZE)
                                                    (LRSH SIZE 1))
                                            (ILESSP SIZE WORDSPERPAGE))))

(PUTPROPS MDSTYPE# MACRO ((PAGE#)
                          (LOGAND (\GETBASE \MDSTypeTable (LRSH PAGE# 1))
                                  \TT.TYPEMASK)))
)

(FILESLOAD (LOADCOMP)
       LLGC LLBASIC)

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(LOCALVARS . T)
)

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS SYSTEMRECLST)
)
)

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

(ADDTOVAR NLAMA SFL PFL)

(ADDTOVAR NLAML )
```

```
(ADDTOVAR LAMA )
)

(PUTPROPS GCHAX COPYRIGHT ("Syntelligence Systems, Inc.  This program or documentation contains confidential
                           information and trade secrets of Syntelligence Systems, Inc.  Reverse engineering,
                           reverse compiling and disassembling of object code are prohibited.  Use of this
                           program or documentation is governed by written agreement with Syntelligence
                           Systems, Inc.  Use of copyright notice is precautionary and does not imply
                           publication or disclosure of trade secrets" 1982 1983 1984 1985 1986 1987 1990 1992
                                 1994))
```

FUNCTION INDEX

MACRO INDEX