

File created: 9-Oct-2024 02:38:17 {DSK}<tmp>loadups-3212>RDSYS.;1

Read Table: INTERLISP

Package: INTERLISP

Format: XCCS

(RPAQQ **RDSYSCOMS**

```
((FNS VREADPAGEMAP VREADPAGEMAPBLOCK VCHECKIFPAGE V\LOCKEDPAGEP V\LOOKUPPAGEMAP VCHECKPAGEMAP
VCHECKFPTOVP VCHECKFPTOVP1 V\SHOWPAGETABLE V\PRINTFPTOVP)
(FNS VRAIDCOMMAND VRAIDSHOWFRAME VRAIDSTACKCMD VRAIDROOTFRAME VPRINTADDRS VPRINTVA VREADVA VREADOCT
VREADATOM VSHOWSTACKBLOCKS VSHOWSTACKBLOCK1 VPRINCOPY VNOSUCHATOM)
(FNS V\BACKTRACE V\STKNAME V\PRINTBF V\PRINTFRAME V\SCANFORNTENTRY V\PRINTSTK)
(FNS V\CHECKARRAYBLOCK V\PARSEARRAYSPACE V\PARSEARRAYSPACE1)
(FNS VPRINTCODE VPRINTCODENT VBROKENDEF)
(FNS V\CAR.UFN V\CDR.UFN)
(FNS V\COPY V\UNCOPY)
(FNS V\GETBASEBYTE V\PUTBASEBYTE)
(FNS VNTYPX VTYPENAME V\TYPENAMEFROMNUMBER)
(FNS VUNCOPYATOM VMAKE.LOCAL.ATOM VSYMBOL.VALUE VSYMBOL.PNAME VSYMBOL.PACKAGE VOLD.FIND.SYMBOL
VLOOKUP-SYMBOL VFIND.PACKAGE VFIND.SYMBOL VPACKAGE.NAME V\MKATOM VGETTOPVAL VGETPROPLIST VSETTOPVAL
VGETDEFN V\ATOMCELL)
(FNS VLISTP)
(VARS (COPYATOMSTR))
(FNS V\GET-COMPILED-CODE-BASE)
;; YOU MUST REMAKE THIS FILE using (DORENAME 'R) (after CONning to library) whenever the SYSOUT layout changes in LLPARAMS
;; (e.g., if MDSTypeTable moves)
(FILE VMEM)
(VARS RDVALS RDPTRS)
(DECLARE%: EVAL@COMPILE DONTCOPY (FILES (LOADCOMP)
VMEM)))
```

(DEFINEQ

(**VREADPAGEMAP**

```
[LAMBDA NIL (*
*)
(PROG (D)
(MAPVMPAGE (IPLUS (LLSH (VHILOC (VVAG2 20 0))
8)
(LRSH (VLOLOC (VVAG2 20 0))
8))
1) (*
*)
(MAPVMPAGE (IPLUS (LLSH (VHILOC (VVAG2 5 0))
8)
(LRSH (VLOLOC (VVAG2 5 0))
8))
(SUB1 (VGETBASE (VVAG2 20 0)
22))) (*
)
(SETVMPTR (VVAG2 5 0))
(for I from 0 to (SUB1 (LRSH (IPLUS 256 31)
5))
as VP from (IPLUS (LLSH (VHILOC (VVAG2 5 0))
8)
(LRSH (VLOLOC (VVAG2 5 0))
8))
by 32 do (*
)
(VREADPAGEMAPBLOCK VP))
(for J from 0 to (SUB1 8) as FP from (SUB1 (VGETBASE (VVAG2 20 0)
23))
do (*
)
(MAPVMPAGE (IPLUS (IPLUS (LLSH (VHILOC (VVAG2 20 512))
8)
(LRSH (VLOLOC (VVAG2 20 512))
8))
J)
FP))
(for I from 0 to (SUB1 (LLSH 8 8)) do (COND
((IEQ (SETQ D (VGETBASE (VVAG2 20 512)
I))
65535))
(T (SETVMPTR (VADDBASE (VVAG2 5 0)
D))
(VREADPAGEMAPBLOCK (LLSH I 5]))
```

(**VREADPAGEMAPBLOCK**

```
[LAMBDA (VP) (*)
(PROG ((B VP)
P)
(FRPTQ 32 [COND
((NEQ (SETQ P (VBIN2))
0)
(MAPVMPAGE B (SUB1 P]
(SETQ B (ADD1 B]))
```

(VCHECKIFPAGE

```

[LAMBDA NIL
(COND
  ((NOT (EQUAL 5603 (VGETBASE (VVAG2 20 0)
                             15)))
   (printout T "Warning: " "Interface page key" "=" " (PROGN 5603)
              ", but \InterfacePage says "
              (VGETBASE (VVAG2 20 0)
                        15)
              T])
  )
)

```

(*)

(VLOCKEDPAGEP

```

[LAMBDA (VP TEMP)
(OR (NEQ 0 (LOGAND (LLSH 1 (IMOD VP 16))
                  (VGETBASE (VADDBASE (VVAG2 20 28672)
                                   (LRSH VP 4))
                            0)))
  NIL)
)

```

(*)
(*)

(VLOOKUPPAGEMAP

```

[LAMBDA (VP)
(LET [(PRIMENTRY (VGETBASE (VVAG2 20 512)
                          (LRSH VP 5))
      (COND
        ((EQ PRIMENTRY 65535)
         0)
        (T (VGETBASE (VVAG2 5 0)
                     (IPLUS PRIMENTRY (LOGAND VP 31))
                    )
         )
      )
)

```

(*)
(*)

(VCHECKPAGEMAP

```

[LAMBDA NIL
(LET ((*PRINT-BASE* 8)
      (NUMOCCUPIED 0)
      (NUMLOCKED 0)
      (CHAINOCCUPIED 0)
      (CHAINLOCKED 0)
      RPTR FPBASE FP VP RP)
  (VCHECKFPTOVP)
  [for RPTINDEX from 1 to (SUB1 VRPTSIZE)
   when (ILESSP (VGETBASE (PROGN (SETQ RPTR (VADDBASE (VADDBASE VREALPAGETABLE (LLSH RPTINDEX 1))
                                                           RPTINDEX)))
                 1)
         65534)
   do (SETQ NUMOCCUPIED (PLUS NUMOCCUPIED 1))
        (SETQ VP (VGETBASE RPTR 1))
        (SETQ FP (VGETBASE RPTR 2))
        (COND
          ((VCHECKFPTOVP1 FP VP RPTINDEX)
           ((NEQ VP (\GETBASEFIXP (SETQ FPBASE (VADDBASE (VVAG2 2 0)
                                                         FP))
                                0))
            (printout T "RPT for RP " (RPFROMRPT RPTINDEX)
                      " says VP ")
            (\PRINTVP VP T)
            (printout T " lives in FP " FP "; but FP Map says that FP contains ")
            (\PRINTVP (\GETBASEFIXP FPBASE 0)
                      T)
            (printout T T))
          (VLOCKEDPAGEP VP)
          (SETQ NUMLOCKED (PLUS NUMLOCKED 1))
          (COND
            ((NOT (NEQ 0 (LRSH (VGETBASE RPTR 0)
                               15)))
             (printout T "VP " VP ", living in RP " (RPFROMRPT RPTINDEX)
                       " should be locked but isn't." T))
            ((IGREATERP FP (DLRPFROMFP (VGETBASE (VVAG2 20 0)
                                                  57)))
             (printout T "VP " VP " is locked, but living in FP " FP ", which is not in the locked
                           page area" T)
            )
          )
        (PROGN (SETQ RPTR VREALPAGETABLE)
               [while (NEQ (SETQ RP (LOGAND (VGETBASE RPTR 0)
                                             32767))
                           0)
                when (ILESSP (VGETBASE (PROGN (SETQ RPTR (VADDBASE (VADDBASE VREALPAGETABLE (LLSH RP 1))
                                                                     RP)))
                              1)
                        65534)
                do (SETQ CHAINOCCUPIED (PLUS CHAINOCCUPIED 1))
                     (COND
                       ((NEQ 0 (LRSH (VGETBASE RPTR 0)
                                     0)
                        )
                      )
                     )
                )
        )
)

```

(*)

(*)

```

15))
  (SETQ CHAINLOCKED (PLUS CHAINLOCKED 1])
(COND
  ((ILESSP CHAINOCCUPIED NUMOCCUPIED)
   (printout T NUMOCCUPIED " occupied pages, but only " CHAINOCCUPIED " are on page chain. "
    NUMLOCKED " pages are permanently locked; " CHAINLOCKED " pages on chain are locked
    somehow." T])

```

(VCHECKFPTOVP

```

[LAMBDA NIL (*]
  (for FP from 1 to (\GETBASEFIXP (VVAG2 20 0)
    82)
   as (FPBASE _ (VADDBASE (VVAG2 2 0)
    1))
   by (VADDBASE FPBASE 1) when (NEQ (VGETBASE FPBASE 0)
    65535)
   do (VCHECKFPTOVP1 FP (\GETBASEFIXP FPBASE 0])

```

(VCHECKFPTOVP1

```

[LAMBDA (FP VP RPTINDEX) (*]
  (PROG ((FP2 (VLOOKUPPAGEMAP VP)))
    (RETURN (COND
      ((NEQ FP2 FP)
       (COND
         (NIL (printout T "RPT for RP " (RPFROMRPT RPTINDEX)))
         (T (printout T "FP map")))
        (printout T " says FP " FP " contains VP ")
        (\PRINTVP VP T)
        (printout T "; but PageMap says that page is in FP " FP2 T)
        T])

```

(VSHOWPAGETABLE

```

[LAMBDA (MODE FILE) (*]
  (PROG ((*PRINT-BASE* 8)
    (OUTSTREAM (GETSTREAM FILE 'OUTPUT))
    (RPTR VREALPAGETABLE)
    (RP 0)
    (FLAGS VP STATE FIRSTONE LASTONE)
    (printout OUTSTREAM " RP VP FilePage Status" T)
    (until (SELECTQ MODE
      (CHAIN (EQ (SETQ RP (LOGAND (VGETBASE RPTR 0)
        32767))
        0))
      (NIL (SETQ RP (PLUS RP 1))
        (IGEQ RP VRPTSIZE))
      (\ILLEGAL.ARG MODE))
     do (SETQ RPTR (VADDBASE (VADDBASE VREALPAGETABLE (LLSH RP 1))
      RP))
      (SETQ VP (VGETBASE RPTR 1))
      (COND
        ((AND (NULL MODE)
          (EQ VP STATE))
         (SETQ LASTONE RP))
        (T (COND
          (LASTONE (printout OUTSTREAM "ditto thru " LASTONE T)
            (SETQ LASTONE NIL)))
          (SETQ FIRSTONE RP)
          (SETQ STATE VP)
          (printout OUTSTREAM .I7.8 (RPFROMRPT RP))
          (COND
            ((EQ (VGETBASE RPTR 1)
              65534)
             (PRIN1 " Empty" OUTSTREAM))
            ((NOT (ILESSP (VGETBASE RPTR 1)
              65534))
             (PRIN1 " Unavailable" OUTSTREAM))
            (T (printout OUTSTREAM .I8.8 VP %,)
              (\PRINTVP VP OUTSTREAM)
              (printout OUTSTREAM 28 .I6.8 (VGETBASE RPTR 2)
                %,,)
              (COND
                ((NEQ 0 (LRSH (VGETBASE RPTR 0)
                  15))
                 (COND
                   ((NOT (VLOCKEDPAGEP VP)) (*]
                    (PRIN1 "Temp" OUTSTREAM)))
                    (PRIN1 "Locked " OUTSTREAM)))
                  NIL))
                (TERPRI OUTSTREAM])

```

(VPRINTFPTOVP

```

[LAMBDA (FIRSTPAGE NWORDS TYPEFLG STREAM VPRAWFLG) (*]
  (SETQ STREAM (GETSTREAM STREAM 'OUTPUT))

```

```

(OR FIRSTPAGE (SETQ FIRSTPAGE 1))
(OR NWORDS (SETQ NWORDS (\GETBASEFIXP (VVAG2 20 0)
82)))
(LET ((BASE (VADDBASE (VVAG2 2 0)
(SUB1 FIRSTPAGE)))
(*PRINT-BASE* 8)
(LASTVP -2)
(NEXTFP (SUB1 FIRSTPAGE))
FIRSTFP FIRSTVP NEXTVP LOCKEDP TYPE NEXTLOCKED NEXTTYPE)
(while (IGEQ NWORDS 0) do (SETQ NEXTFP (PLUS NEXTFP 1))
[COND
((EQ NWORDS 0)
(SETQ NEXTVP -1))
((NEQ (SETQ NEXTVP (VGETBASE (SETQ BASE (VADDBASE BASE 1))
0))
65535)
(SETQ NEXTLOCKED (VLOCKEDPAGEP NEXTVP))
(if TYPEFLG
then (SETQ NEXTTYPE (VTYPENAME ([LAMBDA ($$1)
(VVAG2 (LRSH (SETQ $$1 NEXTVP)
8)
(LLSH (LOGAND $$1 255)
8]
NIL)))
(if (NULL NEXTTYPE)
then (SETQ NEXTTYPE (SELECTC (LRSH NEXTVP 8)
((LIST 8 (CL:1+ 8))
"Pnames")
((LIST 10 (CL:1+ 10))
"Definitions")
((LIST 12 (CL:1+ 12))
"Value cells")
((LIST 2 (CL:1+ 2))
"Property lists")
((VHILOC (VVAG2 2 0))
"\FPTOVP")
(1 "Stack")
((VHILOC (VVAG2 22 0))
"GC Main table")
((VHILOC (VVAG2 23 0))
"GC Overflow table")
NIL]
[COND
[COND
((EQ NEXTVP 65535)
(NEQ LASTVP 65535))
(T (OR (NEQ NEXTVP (ADD1 LASTVP))
(NEQ NEXTLOCKED LOCKEDP)
(NEQ TYPE NEXTTYPE)
[COND
((IGEQ LASTVP 0)
(COND
(FIRSTFP (printout STREAM FIRSTFP "-"))
(printout STREAM (SUB1 NEXTFP)
12)
(COND
((EQ LASTVP 65535)
(printout STREAM "empty"))
(T (COND
(FIRSTFP (if VPRAWFLG
then (PRIN1 FIRSTVP STREAM)
else (\PRINTVP FIRSTVP STREAM))
(PRIN1 "-" STREAM)))
(if VPRAWFLG
then (PRIN1 LASTVP STREAM)
else (\PRINTVP LASTVP STREAM))
(COND
(LOCKEDP (PRIN1 '* STREAM)))
(if TYPE
then (printout STREAM 32 TYPE]
(SETQ FIRSTFP)
(TERPRI STREAM)
(SETQ FIRSTVP NEXTVP))
(T
(*)
(OR FIRSTFP (SETQ FIRSTFP (SUB1 NEXTFP)
(SETQ LASTVP NEXTVP)
(SETQ LOCKEDP NEXTLOCKED)
(SETQ TYPE NEXTTYPE)
(SETQ NWORDS (PLUS NWORDS -1]))
)
)
(DEFINEQ
(VRAIDCOMMAND
[LAMBDA NIL
(DECLARE (USEDFREE ROOTFRAME ALINKS? RAIDIX FRAME# VPRINTLEVEL))
(*)

```

```
(FRESHLINE T)
(PROG (CMD)
  (SELECTQ (SETQ CMD (ASKUSER NIL NIL "@ " ' ((Q "uit [confirm]" CONFIRMFLG T)
    (%i "^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:")
(%Å "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)
(%Û " Enter Lisp " EXPLAINSTRING "^Y -- Enter Lisp" RETURN
'^Y))

T))
(^N (RETURN 'RETURN))
(Q (TERPRI T)
(RETURN 'QUIT))
(NIL)
(A (VPRINCOPY (VGETTOPVAL (VREADATOM))))
(P (VPRINCOPY (VGETPROPLIST (VREADATOM))))
(C (VPRINTCODE (VREADATOM)
T RAIDIX))
(V (VPRINCOPY (VREADVA)))
(B (VPRINTADDRS (VREADVA)
(VREADOCT " for (number of words): ")))
(S (VPRINTADDRS (VVAG2 1 (VREADOCT))
(VREADOCT " for (number of words): ")))
(D (VPRINTADDRS (VATOMCELL (PROGN (VREADATOM))
10)
2))
(^O (PRINTNUM .I2 (VATOMNUMBER (VREADATOM)
T))
(^V (PROG ((ATM (VREADATOM))
(printout T " to be ")
(VSETTOPVAL ATM (READ T T))))
((L ^L)
(VRAIDSTACKCMD CMD))
(F [VRAIDSHOWFRAME (SETQ FRAME# (PROG1 (READ T T)
(READC T]))
(LF (OR FRAME# (SETQ FRAME# 0))
(printout T "(" .I1 (SETQ FRAME# (PLUS FRAME# 1))
)" " T)
(VRAIDSHOWFRAME FRAME#))
(^ (COND
((OR (NULL FRAME#)
(ILEQ FRAME# 1))
(printout T "No previous frame" T))
(T (printout T "(" .I1 (SETQ FRAME# (PLUS FRAME# -1))
)" " T)
(VRAIDSHOWFRAME FRAME#))))
(^F (VPRINTBF (VREADOCT)
NIL
(FUNCTION VPRINCOPY)))
(Z [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 [VSHOWSTACKBLOCKS (COND
(EQ (PEEK T)
'%)
(READC T)
(VGETBASE (VVAG2 20 0)
30))

```

```

(T (VREADOCT])
(^X (VPRINTFRAME (VREADOCT)
      'PRINCOPI))
(^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 (VREADVA)))
      (printout T " Currently ")
      (PRINTNUM .I7 (VGETBASE VA 0)
      T)
      (printout T " to be ")
      (VPUTBASE VA 0 (VREADOCT))))
(I (COND
      [(NULL (GETD 'INSPECT)
      (RECLOSE '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])

```

(VRAIDSHOWFRAME

```

[LAMBDA (N)
      (PROG [(FRAME (OR ROOTFRAME (VRAIDROOTFRAME)
      [FRPTQ (SUB1 N)
      (COND
      ((EQ [PROGN (SETQ FRAME (COND
      (ALINKS? (IDIFFERENCE (LOGAND (VGETBASE (VVAG2 1 FRAME)
      1)
      (CONSTANT (LOGXOR (SUB1 2)
      -1)))
      10))
      (T (IDIFFERENCE (COND
      ((NOT (NEQ 0 (LOGAND (VGETBASE (VVAG2 1 FRAME)
      1)
      (VGETBASE (VVAG2 1 FRAME)
      1)
      (T (VGETBASE (VVAG2 1 FRAME)
      9)))
      10]
      0)
      (RETURN (printout T N " is beyond the bottom of the stack" T]
      (VBACKTRACE FRAME FRAME T NIL T T NIL (FUNCTION VPRINCOPI)
      NIL RAIDIX])

```

(VRAIDSTACKCMD

```

[LAMBDA (CMD)
      (DECLARE (USEDFFREE FRAME# ROOTFRAME))
      (PROG (FRAME)
      (SETQ FRAME# 0)
      [COND
      ((EQ CMD 'L)
      (VRAIDROOTFRAME))
      (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 (VGETBASE (VVAG2 20 0)
      6))
      (G (VGETBASE (VVAG2 20 0)
      5))
      (K (VGETBASE (VVAG2 20 0)
      3))

```

```

(H (VGETBASE (VVAG2 20 0)
4))
(S (VGETBASE (VVAG2 20 0)
2))
(R (VGETBASE (VVAG2 20 0)
1))
(M (VGETBASE (VVAG2 20 0)
14))
(COND
((AND (ILESSP (SETQ FRAME (VREADOCT))
256)
(ILESSP (VGETBASE (VVAG2 20 0)
FRAME)
(VGETBASE (VVAG2 20 0)
7))
(IEQ (LRSH (VGETBASE [VVAG2 1 (PROGN (PROGN (VGETBASE (VVAG2 20 0)
FRAME]
0)
13)
6))
(VGETBASE (VVAG2 20 0)
FRAME))
((IEQ (LRSH (VGETBASE (VVAG2 1 FRAME)
0)
13)
6)
FRAME)
(T (PRINTNUM .I7 FRAME)
(printout T " not a valid frame." T)
(RETURN)
(FRESHLINE T)
(VBACKTRACE ROOTFRAME NIL T NIL NIL NIL ALINKS? (FUNCTION VPRINCOPY)
1 RAIDIX])

```

(VRAIDROOTFRAME

```

[LAMBDA NIL (*)
(SETQ ROOTFRAME (PROG1 (COND
((LISTP VMEMFILE)
(PRIN1 "in TeleRaid Context" T)
(VGETBASE (VVAG2 20 0)
24))
(T (VGETBASE (VVAG2 20 0)
0)))
(TERPRI T])

```

(VPRINTADDRS

```

[LAMBDA (BASE CNT) (*)
(PRIN1 "words from ")
(VPRINTVA BASE)
(PRIN1 " to ")
(VPRINTVA (VADDBASE BASE (SUB1 CNT)))
(TERPRI)
(SPACES 7)
(for I from 0 to 7 do (PRINTNUM .I7 I))
(PROG ([NB (VVAG2 (VHILOC BASE)
(LOGAND (VLOLOC BASE)
(CONSTANT (LOGXOR (SUB1 8)
-1]
(LB (VADDBASE BASE CNT)))
(do (COND
((EVENP (VLOLOC NB)
8)
(TAB 0 0)
(PRINTNUM .I5 (VLOLOC NB))
(PRIN1 ": ")))
[COND
((IGREATERP BASE NB)
(SPACES 7))
(T (PRINTNUM .I7 (VGETBASE NB 0)
(SETQ NB (VADDBASE NB 1)) repeatwhile (IGREATERP LB NB))
(TAB 0 0])

```

(VPRINTVA

```

[LAMBDA (X) (*)
(PRIN1 "{")
(PRINTNUM .I2 (VHILOC X))
(PRIN1 ",")
(PRINTNUM .I2 (VLOLOC X))
(PRIN1 "}"]

```

(VREADVA

```

[LAMBDA NIL (*)
(VVAG2 (VREADOCT))

```

(VREADOCT])

(VREADOCT

```

[LAMBDA (PROMPT)
(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]))

```

(VREADATOM

```

[LAMBDA NIL
(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]))

```

(VSHOWSTACKBLOCKS

```

[LAMBDA (SCANPTR WAITFLG)
(PROG ((EASP (VGETBASE (VVAG2 20 0)
7)))
SCAN
[SELECTC (LRSH (VGETBASE (VVAG2 1 SCANPTR)
0)
13)
(5 (VSHOWSTACKBLOCK1 SCANPTR "free block" (IEQ (VGETBASE (VVAG2 1 SCANPTR)
0)
40960))
(SETQ SCANPTR (PLUS SCANPTR (VGETBASE (VVAG2 1 SCANPTR)
1))))
(7 (VSHOWSTACKBLOCK1 SCANPTR "guard block" T)
(SETQ SCANPTR (PLUS SCANPTR (VGETBASE (VVAG2 1 SCANPTR)
1))))
(6 [VSHOWSTACKBLOCK1
SCANPTR "Frame extn = "
(AND (IEQ (LRSH (VGETBASE (VVAG2 1 SCANPTR)
0)
13)
6)
(OR [IEQ (IDIFFERENCE SCANPTR 2)
(COND
((NOT (NEQ 0 (LOGAND (VGETBASE (VVAG2 1 SCANPTR)
1)
1))))
(IDIFFERENCE SCANPTR 2))
(T (VGETBASE (VVAG2 1 SCANPTR)
8]
(AND (NEQ 0 (LOGAND (LRSH (VGETBASE (VVAG2 1 (PROGN (IDIFFERENCE SCANPTR 2)))
0)

```

```

9)
1))
(IEQ (VGETBASE (VVAG2 1 (PROGN (IDIFFERENCE SCANPTR 2)))
1)
(VGETBASE [VVAG2 1 (PROGN (COND
1)
(NOT (NEQ 0 (LOGAND (VGETBASE (VVAG2 1
SCANPTR
)
1)
1)))
(IDIFFERENCE SCANPTR 2))
(T (VGETBASE (VVAG2 1 SCANPTR)
8]
1]
(PRIN2 (VUNCOPY (VGETBASEPTR [PROGN (COND
1)
((NEQ 0 (LOGAND (LRSH (VGETBASE (VVAG2 1 SCANPTR)
0)
9)
1))
(VGETBASEPTR (VVAG2 1 SCANPTR)
6)
(T (VGETBASEPTR (VVAG2 1 SCANPTR)
2]
4)))
(SETQ SCANPTR (VGETBASE (VVAG2 1 SCANPTR)
4)))
(PROG ((ORIG SCANPTR)
IVAR)
(*)
(while (EQ (LRSH (VGETBASE (VVAG2 1 SCANPTR)
0)
13)
0)
do (SETQ SCANPTR (PLUS SCANPTR 2)))
(COND
((NOT (IEQ (LRSH (VGETBASE (VVAG2 1 SCANPTR)
0)
13)
4))
(VSHOWSTACKBLOCK1 ORIG "Garbage" T))
(T (SETQ IVAR (VGETBASE (VVAG2 1 SCANPTR)
1))
[COND
((NEQ 0 (LOGAND (LRSH (VGETBASE (VVAG2 1 SCANPTR)
0)
9)
1))
(VSHOWSTACKBLOCK1 SCANPTR "Residual BF" (EQ SCANPTR ORIG))
(PRIN1 " with IVar = ")
(PRINTNUM .I7 IVAR))
(T (VSHOWSTACKBLOCK1
SCANPTR "Basic frame"
(AND (EQ ORIG IVAR)
(AND (IEQ (LRSH (VGETBASE (VVAG2 1 SCANPTR)
0)
13)
4)
(for I from (VGETBASE (VVAG2 1 SCANPTR)
1)
to (IDIFFERENCE SCANPTR 2) by 2
always (IEQ 0 (LRSH (VGETBASE (VVAG2 1 I)
0)
13]
(SETQ SCANPTR (PLUS SCANPTR 2]
(TERPRI)
(COND
((IGREATERP SCANPTR EASP)
(RETURN)))
(AND WAITFLG (READC T))
(GO SCAN])

```

```

(VSHOWSTACKBLOCK1
[LAMBDA (PTR STR GOODFLG)
(PRINTNUM .I7 PTR)
(SPACES 1)
(OR GOODFLG (PRIN1 "[bad] "))
(PRIN1 STR)]
(*)

```

```

(VPRINCOPY
[LAMBDA (X)
(PRINT (VUNCOPY X (CAR VPRINTLEVEL)
(CDR VPRINTLEVEL))
T T)]
(*)

```

(VNOSUCHATOM

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

(DEFINEQ

(VBACKTRACE

```
[LAMBDA (IPOS EPOS NAMES VARS LOCALS JUNK ALINKS PRINTFN CNT RADIX)
  (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 ": ")
        (SETQ CNT (PLUS CNT 1]
      (SETQ NAME (VSTKNAME IPOS))
      (COND
        (JUNK (TERPRI)
          (TERPRI)
          (PRIN1 "Basic frame at ")
          [PRINTNUM .I7 (SETQ BLINK (COND
            ((NOT (NEQ 0 (LOGAND (VGETBASE (VVAG2 1 IPOS)
              1)))
              (IDIFFERENCE IPOS 2))
            (T (VGETBASE (VVAG2 1 IPOS)
              8])
          (TERPRI)
          (VPRINTBF BLINK (COND
            ((NEQ 0 (LOGAND (LRSH (VGETBASE (VVAG2 1 IPOS)
              0)
              9)
              1))
            (VGETBASEPTR (VVAG2 1 IPOS)
              6))
            (T (VGETBASEPTR (VVAG2 1 IPOS)
              2)))
          PRINTFN)
        (PROGN (TERPRI)
          (PRIN1 "Frame xtn at ")
          (PRINTNUM .I7 IPOS)
          (PRIN1 ", frame name= "))
          (APPLY* PRINTFN NAME)
          (VPRINTFRAME IPOS PRINTFN))
      [(OR VARS LOCALS)
        (VPRINTBF (COND
          ((NOT (NEQ 0 (LOGAND (VGETBASE (VVAG2 1 IPOS)
            1)
            1)))
            (IDIFFERENCE IPOS 2))
          (T (VGETBASE (VVAG2 1 IPOS)
            8)))
          (COND
            ((NEQ 0 (LOGAND (LRSH (VGETBASE (VVAG2 1 IPOS)
              0)
              9)
              1))
              (VGETBASEPTR (VVAG2 1 IPOS)
                6))
            (T (VGETBASEPTR (VVAG2 1 IPOS)
              2)))
          PRINTFN
          (COND
            (LOCALS 'LOCALS)
            (T T)))
          (COND
            (NAMES (APPLY* PRINTFN NAME)
              (TERPRI)))
          (VPRINTFRAME IPOS PRINTFN (COND
            (LOCALS 'LOCALS)
            (T T]
            (NAMES (APPLY* PRINTFN NAME))))
      (COND
        ((AND (NEQ EPOS IPOS)
          (NOT (EQ [PROGN (SETQ IPOS (COND
            (ALINKS (IDIFFERENCE (LOGAND (VGETBASE (VVAG2 1 IPOS)
              1)
              (CONSTANT (LOGXOR (SUB1 2)
                -1)))
            10))
            (T (IDIFFERENCE (COND
              ((NOT (NEQ 0 (LOGAND (VGETBASE (VVAG2 1 IPOS)
                )
```

```

1)
(VGETBASE (VVAG2 1 IPOS)
1))
(T (VGETBASE (VVAG2 1 IPOS)
9)))
10]
0)))
(GO POSLP)))
(RETURN T])

```

(VSTKNAME

```

[LAMBDA (POS)
(*)
(*)
(LET ((NAME (VGETBASEPTR [PROGN (COND
((NEQ 0 (LOGAND (LRSH (VGETBASE (VVAG2 1 POS)
0)
9)
1))
(VGETBASEPTR (VVAG2 1 POS)
6))
(T (VGETBASEPTR (VVAG2 1 POS)
2]
4)))
(if (EQ NAME '\INTERPRETER)
then [VGETBASEPTR (VVAG2 1 0)
(LET [(BFLINK (COND
((NOT (NEQ 0 (LOGAND (VGETBASE (VVAG2 1 POS)
1)))
(IDIFFERENCE POS 2))
(T (VGETBASE (VVAG2 1 POS)
8]
(+ (VGETBASE (VVAG2 1 BFLINK)
1)
(TIMES (CL:1- (IDIFFERENCE (LRSH (IDIFFERENCE BFLINK (VGETBASE (VVAG2 1 BFLINK)
1)
1)))
(LOGAND (LRSH (VGETBASE (VVAG2 1 BFLINK)
0)
8)
1)))
2]
else NAME])

```

(VPRINTBF

```

[LAMBDA (BL NMT PRINTFN VARONLY)
(*)
[bind NM for I from (VGETBASE (VVAG2 1 BL)
1)
by 2 as J from 0 to (SUB1 (IDIFFERENCE (LRSH (IDIFFERENCE BL (VGETBASE (VVAG2 1 BL)
1)
1))
(LOGAND (LRSH (VGETBASE (VVAG2 1 BL)
0)
8)
1)))
do (OR VARONLY (VPRINTSTK I))
[COND
([OR (SETQ NM (VSCANFORNTENTRY [OR NMT (RETURN (OR VARONLY (TERPRI)
(MAKE-NTENTRY 0 J)))
(AND (NEQ VARONLY T)
(SETQ NM '*local*')
(AND VARONLY (SPACES 3))
(PRIN2 NM)
(SPACES 1)
(APPLY* PRINTFN (VGETBASEPTR (VVAG2 1 0)
I]
finally (OR VARONLY (while (ILESSP I BL) do (VPRINTSTK I)
(printout NIL "[padding]" T)
(SETQ I (PLUS I 2]
(COND
((NOT VARONLY)
(VPRINTSTK BL)
(COND
((NEQ 0 (LOGAND (LRSH (VGETBASE (VVAG2 1 BL)
0)
9)
1))
(PRIN1 "residual ")))
(COND
((NEQ (LOGAND (VGETBASE (VVAG2 1 BL)
0)
255)
0)
(printout NIL "usecnt= " (LOGAND (VGETBASE (VVAG2 1 BL)

```



```

(COND
  ((EQ (LRSH (VGETBASE (PROGN (VVAG2 1 I))
                    0)
        12)
       0)
   (AND VARONLY (SPACES 3))
   (PRIN2 TMP)
   (SPACES 1)
   (APPLY* PRINTFN (VGETBASEPTR (VVAG2 1 I)
                               0)))
  ((NOT VARONLY)
   (printout NIL TMP " [unbound]" T)
  (NOT VARONLY)
  (COND
    ((SETQ TMP (VSCANFORNTENTRY NMT (MAKE-NTENTRY 49152 J)))
     (printout NIL "[fvar " .P2 TMP " "
                  (COND
                    ((EVENP (VGETBASE (PROGN (VVAG2 1 I))
                                         0))
                     (COND
                       ((EQ [SETQ TMP (VHILOC ([LAMBDA ($$1)
                                               (VVAG2 (VGETBASE (PROGN $$1)
                                                             1)
                                                             (VGETBASE $$1 0]
                                               (VVAG2 1 I]
                                               1)
                                               " on stack]"
                       ((NEQ (LOGAND TMP (CONSTANT (LOGXOR (SUB1 2)
                                                           -1)))
                            (VHILOC (VVAG2 12 0)))
                            (*))
                        " non-stack binding]"
                        (T " top value]"
                        (T " not looked up]"
                        T))
                    (T (printout NIL "[padding]" T)
  (COND
    ((NOT VARONLY)
     (SETQ FT (VGETBASE (VVAG2 1 FRAME)
                       4))
     (for old I by 2 while (ILESSP I FT) do
       (VPRINTSTK I)
       (COND
         ((EQ (LRSH (VGETBASE (PROGN (VVAG2 1 I))
                             0)
                   12)
              0)
          (APPLY* PRINTFN (VGETBASEPTR (VVAG2 1 I)
                                       0)))
         (T (TERPRI])

```

(VSCANFORNTENTRY

```

[LAMBDA (NMT NTENTRY)
  (bind NM for NT1 from (PROGN 8) by (CONSTANT (PROGN 2)) as NT2 from (IPLUS (PROGN 8)
                                                                              (VGETBASE NMT 6))
   by (CONSTANT (WORDSPERNTOFFSETENTRY)) do (COND
     ((NULL-NTENTRY (SETQ NM (GETSTKNAMEENTRY NMT NT1)))
      (RETURN))
     (COND
      ((IEQP NTENTRY (GETSTKNTOFFSETENTRY NMT NT2))
       (RETURN (VATOM NM])

```

(VPRINTSTK

```

[LAMBDA (I)
  (PRINTNUM .I7 I)
  (PRIN1 ": ")
  (PRINTNUM .I7 (VGETBASE (VVAG2 1 0)
                        I))
  (PRINTNUM .I7 (VGETBASE (VVAG2 1 0)
                        (ADD1 I)))
  (SPACES 1])

```

)

(DEFINEQ

(VCHECKARRAYBLOCK

```

[LAMBDA (BASE FREE ONFREELIST)
  (COND
    (T (PROG (ERROR TRAILER)
      (COND
        ((NEQ (LRSH (VGETBASE BASE 0)
                    3)
             5461)

```

```

    (SETQ ERROR "ARRAYBLOCK Password wrong"))
  ((NEQ (NEQ 0 (LOGAND (VGETBASE BASE 0)
    1))
    (NOT FREE))
    (SETQ ERROR "ARRAYBLOCK INUSE bit set wrong"))
  (NIL (SETQ ERROR "Free ARRAYBLOCK with RefCnt not 1"))
  ((NEQ (LRSH (VGETBASE (SETQ TRAILER ([LAMBDA (BASE N)
    (DECLARE (LOCALVARS BASE N)
      (VADDBASE (VADDBASE BASE N)
        N]
        BASE
      (IDIFFERENCE (VGETBASE BASE 1)
        1)))
    0)
    3)
    5461)
    (SETQ ERROR "ARRAYBLOCK Trailer password wrong"))
  ((NEQ (VGETBASE BASE 1)
    (VGETBASE TRAILER 1))
    (SETQ ERROR "ARRAYBLOCK Header and Trailer length don't match"))
  ((NEQ (NEQ 0 (LOGAND (VGETBASE BASE 0)
    1))
    (NOT FREE))
    (SETQ ERROR "ARRAYBLOCK Trailer INUSE bit set wrong"))
  ((OR (NOT ONFREELIST)
    (ILESSP (VGETBASE BASE 1)
      4))
    (*))
  (RETURN))
  ((OR (NOT (EQUAL (VGETBASEPTR (VGETBASEPTR BASE 4)
    2)
    BASE))
    (NOT (EQUAL (VGETBASEPTR (VGETBASEPTR BASE 2)
    4)
    BASE)))
    (SETQ ERROR "ARRAYBLOCK links fouled"))
  [(bind (FBL _ ([LAMBDA (BASE N)
    (DECLARE (LOCALVARS BASE N)
      (VADDBASE (VADDBASE BASE N)
        N]
        VFREEBLOCKBUCKETS
      (IMIN (INTEGERLENGTH (VGETBASE BASE 1)
        30)))
    ROVER first (OR (SETQ ROVER (VGETBASEPTR FBL 0))
      (RETURN (SETQ ERROR "Free block's bucket empty"))))
    do (AND (EQUAL ROVER BASE)
      (RETURN))
    (VCHECKARRAYBLOCK ROVER T)
    repeatuntil (EQ (SETQ ROVER (VGETBASEPTR ROVER 2))
      (VGETBASEPTR FBL 0])
      (*))
    (T (RETURN)))
  (ERROR BASE ERROR)
  (RETURN ERROR)]

```

(V**PARSEARRAYSPACE**

```

[LAMBDA (FN)
  (COND
    ((NEQ VArrayFrLst2 (VVAG2 64 0))
      (VPARSEARRAYSPACE1 FN (VVAG2 46 0)
        VArrayFrLst2)
      (VPARSEARRAYSPACE1 FN (VVAG2 64 0)
        VArrayFrLst))
    (T (VPARSEARRAYSPACE1 FN (VVAG2 46 0)
      VArrayFrLst]))
  (*))

```

(V**PARSEARRAYSPACE1**

```

[LAMBDA (FN START END)
  (for (ROVER _ START) repeatuntil [EQUAL END (SETQ ROVER ([LAMBDA (BASE N)
    (DECLARE (LOCALVARS BASE N)
      (VADDBASE (VADDBASE BASE N)
        N]
        ROVER
      (VGETBASE ROVER 1)
    ))]
    do (VCHECKARRAYBLOCK ROVER (NOT (NEQ 0 (LOGAND (VGETBASE ROVER 0)
      1)))
      (AND (NOT (NEQ 0 (LOGAND (VGETBASE ROVER 0)
        1)))
        (VGETBASEPTR ROVER 2)))
      (AND FN (APPLY* FN ROVER (VGETBASE ROVER 1)
        (NEQ 0 (LOGAND (VGETBASE ROVER 0)
          1))
        (LOGAND (LRSH (VGETBASE ROVER 0)
          1)
          3))

```

)

(DEFINEQ

(VPRINTCODE

```
[LAMBDA (FN LVFLG RADIX OUTF FIRSTBYTE PC FN.IS.CODEBASE)
  (*
  (*
  (*
  (*
  (*
  (*
```

```
(DECLARE (SPECVARS OUTF))
(OR RADIX (SETQ RADIX 16))
(LET ([CODEBASE (COND
```

```
(FN.IS.CODEBASE FN)
  (T (OR (VGET-COMPILED-CODE-BASE FN)
    [AND (LITATOM FN)
      (VGET-COMPILED-CODE-BASE (GET FN 'CODE])
      (ERROR FN "not compiled code"]
    (I4 (NUMFORMATCODE (LIST 'FIX (if (IGREATERP RADIX 15)
      then 3
      else 4)
      RADIX)))
    (I6 (NUMFORMATCODE (LIST 'FIX (if (IGREATERP RADIX 15)
      then 5
      else 6)
      RADIX)))
  NTSIZE STARTPC TAG TEMP OP# PVARs FVARs IVARs)
```

```
(DECLARE (SPECVARS CODEBASE IVARs PVARs FVARs I4 I6)) (*)
(LET ((*PRINT-BASE* RADIX))
  (for I from 0 by 2 while (ILESSP I (LLSH (PROGN 8)
    1))
```

```
  do (PRINTNUM I4 I OUTF)
      (PRIN1 ": " OUTF)
      (PRINTNUM I6 (LOGOR (LLSH (COND
        ((NEQ 0 (LOGAND (LRSH (VGETBASE CODEBASE 4)
          14)
          1))
        (VGETBASEBYTE CODEBASE (LOGXOR I 3)))
        (T (VGETBASEBYTE CODEBASE I)))
        8)
      ([LAMBDA (CODEBASE OFFSET)
        (DECLARE (LOCALVARs CODEBASE OFFSET))
        (COND
          ((NEQ 0 (LOGAND (LRSH (VGETBASE CODEBASE 4)
            14)
            1))
          (VGETBASEBYTE CODEBASE (LOGXOR OFFSET 3)))
          (T (VGETBASEBYTE CODEBASE OFFSET)
            CODEBASE
            (ADD1 I)))
        OUTF)
      (SELECTQ I
        (0 (PRIN1 " stkmin" OUTF))
        (2 (PRIN1 " na" OUTF))
        (4 (PRIN1 " pv" OUTF))
        (6 (PRIN1 " startpc" OUTF))
        (8 (AND (NEQ 0 (LRSH (VGETBASE CODEBASE 4)
          15))
          (PRIN1 "[CLOSUREP]" OUTF))
          (printout OUTF " byteswapped: " (NEQ 0 (LOGAND (LRSH (VGETBASE CODEBASE 4)
            14)
            1)))
          (printout OUTF " argtype: " (LOGAND (LRSH (VGETBASE CODEBASE 4)
            12)
            3)))
        (10 (printout OUTF " frame name: " .P2 (VUNCOPY (VGETBASEPTR CODEBASE 4))))
        (12 (PRIN1 " ntsize" OUTF))
        (14 (printout OUTF " nlocals: " (LRSH (VGETBASE CODEBASE 7)
          8)
          (printout OUTF " fvaroffset: " (LOGAND (VGETBASE CODEBASE 7)
            255)))
          NIL)
          (*)
```

```
(TERPRI OUTF))
(SETQ NTSIZE (VGETBASE CODEBASE 6))
(VPRINTCODENT "name table: " (LLSH (PROGN 8)
  1)
  (LLSH NTSIZE 1))
(SETQ STARTPC (VGETBASE CODEBASE 3))
(COND
  ((GREATERP [SETQ NTSIZE (IDIFFERENCE (COND
```

```
  ((PROGN NIL) (*))
  (- STARTPC 4))
  (T STARTPC))
  (SETQ TEMP (IPLUS (LLSH (PROGN 8)
    1)
    (COND
```



```

14)
(VGETBASEBYTE 1) CODEBASE (LOGXOR OFFSET 3))
(T (VGETBASEBYTE CODEBASE OFFSET]
CODEBASE
(IPLUS 2 CODELOC])
NIL)
[COND
(AND LEVEL (SETQ LEVADJ (fetch LEVADJ of TAG)))
[COND
((LISTP LEVADJ)
(SETQ LEVADJ (CAR LEVADJ)
(SELECTQ LEVADJ
(FNX [SETQ LEVEL (PLUS LEVEL (IDIFFERENCE 1
(COND
((NEQ 0 (LOGAND (LRSH (VGETBASE CODEBASE 4)
14)
1))
(VGETBASEBYTE CODEBASE (LOGXOR CODELOC 3
)))
(T (VGETBASEBYTE CODEBASE CODELOC])
(POP.N [SETQ LEVEL (IDIFFERENCE LEVEL (COND
((NEQ 0 (LOGAND (LRSH (VGETBASE CODEBASE
4)
14)
1))
(VGETBASEBYTE CODEBASE
(LOGXOR CODELOC 3)))
(T (VGETBASEBYTE CODEBASE CODELOC])
((JUMP UNWIND)
(SETQ LEVEL))
((CJUMP NCJUMP)
(SETQ LEVEL (PLUS LEVEL -1)))
(COND
((NUMBERP LEVADJ)
(SETQ LEVEL (PLUS LEVEL LEVADJ]
(add CODELOC (fetch OPNARGS of TAG))
(GO LP)))
[SETQ LEN (fetch OPNARGS of (SETQ TAG (\FINDOP (SETQ B
([LAMBDA (CODEBASE OFFSET)
(DECLARE (LOCALVARS CODEBASE OFFSET))
(COND
((NEQ 0 (LOGAND (LRSH (VGETBASE CODEBASE 4)
14)
1))
(VGETBASEBYTE CODEBASE (LOGXOR OFFSET 3))
)
(T (VGETBASEBYTE CODEBASE OFFSET]
CODEBASE
(PROG1 CODELOC
(SETQ CODELOC (PLUS CODELOC 1))))]
(PRINTNUM I4 B OUTF)
(COND
((IGREATERP LEN 0)
(PRINTNUM I4 [SETQ B1 ([LAMBDA (CODEBASE OFFSET)
(DECLARE (LOCALVARS CODEBASE OFFSET))
(COND
((NEQ 0 (LOGAND (LRSH (VGETBASE CODEBASE 4)
14)
1))
(VGETBASEBYTE CODEBASE (LOGXOR OFFSET 3)))
(T (VGETBASEBYTE CODEBASE OFFSET]
CODEBASE
(PROG1 CODELOC
(SETQ CODELOC (PLUS CODELOC 1))))]
OUTF)))
(COND
((IGREATERP LEN 1)
(PRINTNUM I4 [SETQ B2 ([LAMBDA (CODEBASE OFFSET)
(DECLARE (LOCALVARS CODEBASE OFFSET))
(COND
((NEQ 0 (LOGAND (LRSH (VGETBASE CODEBASE 4)
14)
1))
(VGETBASEBYTE CODEBASE (LOGXOR OFFSET 3)))
(T (VGETBASEBYTE CODEBASE OFFSET]
CODEBASE
(PROG1 CODELOC
(SETQ CODELOC (PLUS CODELOC 1))))]
OUTF)))
(COND
((IGREATERP LEN 2)
(PRINTNUM I4 [SETQ B3 ([LAMBDA (CODEBASE OFFSET)
(DECLARE (LOCALVARS CODEBASE OFFSET))
(COND
((NEQ 0 (LOGAND (LRSH (VGETBASE CODEBASE 4)
14)

```

```

                                1))
                                (\VGETBASEBYTE CODEBASE (LOGXOR OFFSET 3))
                                (T (\VGETBASEBYTE CODEBASE OFFSET]
CODEBASE
(PROG1 CODELOC
 (SETQ CODELOC (PLUS CODELOC 1))))]
                                OUTF)))
(COND
 ((IGREATERP LEN 3)
 (PRINTNUM I4 [SETQ B4 ([LAMBDA (CODEBASE OFFSET)
 (DECLARE (LOCALVARS CODEBASE OFFSET))
 (COND
 ((NEQ 0 (LOGAND (LRSH (VGETBASE CODEBASE 4)
 14)
 1))
 (\VGETBASEBYTE CODEBASE (LOGXOR OFFSET 3))
 (T (\VGETBASEBYTE CODEBASE OFFSET]
CODEBASE
(PROG1 CODELOC
 (SETQ CODELOC (PLUS CODELOC 1))))]
                                OUTF)))
(COND
 ((IGREATERP LEN 4)
 (PRINTNUM I4 [SETQ B5 ([LAMBDA (CODEBASE OFFSET)
 (DECLARE (LOCALVARS CODEBASE OFFSET))
 (COND
 ((NEQ 0 (LOGAND (LRSH (VGETBASE CODEBASE 4)
 14)
 1))
 (\VGETBASEBYTE CODEBASE (LOGXOR OFFSET 3))
 (T (\VGETBASEBYTE CODEBASE OFFSET]
CODEBASE
(PROG1 CODELOC
 (SETQ CODELOC (PLUS CODELOC 1))))]
                                OUTF)))
(PROGN (printout OUTF 30 (fetch OPCODENAME of TAG))
 (SETQ OP# (fetch OP# of TAG))
 (SETQ LEVADJ (fetch LEVADJ of TAG)))
[COND
 ((LISTP OP#)
 (SETQ OP# (CAR OP#])
[SELECTQ (SETQ TAG (OR (fetch OPPRINT of TAG)
 (fetch OPCODENAME of TAG)))
 (-X- (TERPRI OUTF)
 (RETURN))
 (IVAR (TAB 40 NIL OUTF)
 (PROGN
 (PROG NIL
 (PRIN2 [CADR (OR (ASSOC (SELECTQ LEN
 (0 (IDIFFERENCE B OP#))
 (LRSH B1 1))
 IVARS)
 (RETURN (printout OUTF "[" 'ivar
 (SELECTQ LEN
 (0 (IDIFFERENCE B OP#))
 (LRSH B1 1))
 "]"")
                                OUTF))))
 (PVAR (TAB 40 NIL OUTF)
 (PROGN
 (PROG NIL
 (PRIN2 [CADR (OR (ASSOC (SELECTQ LEN
 (0 (IDIFFERENCE B OP#))
 (LRSH B1 1))
 PVAR)
 (RETURN (printout OUTF "[" 'pvar
 (SELECTQ LEN
 (0 (IDIFFERENCE B OP#))
 (LRSH B1 1))
 "]"")
                                OUTF))))
 (FVAR (TAB 40 NIL OUTF)
 (PROGN
 (PROG NIL
 (PRIN2 [CADR (OR (ASSOC (SELECTQ LEN
 (0 (IDIFFERENCE B OP#))
 (LRSH B1 1))
 FVAR)
 (RETURN (printout OUTF "[" 'fvar
 (SELECTQ LEN
 (0 (IDIFFERENCE B OP#))
 (LRSH B1 1))
 "]"")
                                OUTF))))
 (JUMP ([LAMBDA (N)
 (PRIN1 "->" OUTF)
 (PRINTNUM I4 [SETQ N (IPLUS N (IDIFFERENCE CODELOC (ADD1 LEN)

```



```

(IPLUS (LLSH B2 8)
      B3])
(TYPEP (printout OUTF "(" .P2 (OR (VTYPEFROMNUMBER B1)
      '?')
      ")"))
(UNBIND (AND LEVEL (SETQ LEVEL (pop STK)))
(DUNBIND [AND LEVEL (SETQ LEVEL (SUB1 (pop STK))
(RETURN (SETQ LEVEL))
(SUBRCALL (printout OUTF 40 (for X in \INITSUBRS when (EQ B1 (CADR X))
      do (RETURN (CAR X)) finally (RETURN "?"))))
      [AND LEVEL (SETQ LEVEL (ADD1 (IDIFFERENCE LEVEL B2))
(MISCN (printout OUTF 40 (for X in \USER-SUBR-LIST when (EQ B1 (CADR X))
      do (RETURN (CAR X)) finally (RETURN "?"))))
      [AND LEVEL (SETQ LEVEL (ADD1 (IDIFFERENCE LEVEL B2))
(COND
  ((LISTP TAG)
   (printout OUTF 40 (CAR (NTH TAG (ADD1 B1)
(TERPRI OUTF)
[COND
  ((AND LEVEL LEVADJ)
   (SELECTQ LEVADJ
    (FNX (SETQ LEVEL (PLUS LEVEL (IDIFFERENCE 1 B1))))
    (POP.N (SETQ LEVEL (IDIFFERENCE LEVEL B1)))
    ((JUMP UNWIND)
     (SETQ LEVEL))
    ((CJUMP NCJUMP)
     (SETQ LEVEL (PLUS LEVEL -1)))
    (COND
     ((NUMBERP LEVADJ)
      (SETQ LEVEL (PLUS LEVEL LEVADJ]
(GO LP])

```

(VPRINTCODENT

```

[LAMBDA (STR START1 START2)
  (DECLARE (USEDFREE CODEBASE IVARS PVARs FVARs I4 I6 OUTF)) (*)
  (LET (NAME TAG)
    (COND
      ((ILESSP START1 (SETQ START2 (IPLUS START2 START1)))
       (printout OUTF STR T)
       (for NT1 from START1 by (LLSH (CONSTANT (PROGN 2))
          1)
        while (ILESSP NT1 START2) as NT2 from START2 by (LLSH (PROGN 2)
          1)
        do (PRINTNUM I4 NT1 OUTF)
           (PRIN1 ": " OUTF)
           (for I from 0 to (CONSTANT (SUB1 (LLSH (CONSTANT (PROGN 2))
              1)))
            do (PRINTNUM I4 ([LAMBDA (CODEBASE OFFSET)
                (DECLARE (LOCALVARs CODEBASE OFFSET))
                (COND
                  ((NEQ 0 (LOGAND (LRSH (VGETBASE CODEBASE 4)
                      14)
                    1))
                   (VGETBASEBYTE CODEBASE (LOGXOR OFFSET 3)))
                  (T (VGETBASEBYTE CODEBASE OFFSET]
                CODEBASE
                (IPLUS NT1 I))
              OUTF))
            (SPACES 2 OUTF)
            (PRINTNUM I4 NT2 OUTF)
            (PRIN1 ": " OUTF)
            (COND
              ((SETQ NAME (VATOM (CODEBASEGETNAME CODEBASE NT1)))
               (SETQ TAG (GETNTOFFSET CODEBASE NT2))
               (printout OUTF .SP 1 (SELECTC (NTSLOT-VARTYPE (GETNTOFFSETENTRY CODEBASE NT2))
                  (0 (push IVARS (LIST TAG NAME))
                    'IVAR)
                  (32768 (push PVARs (LIST TAG NAME))
                    'PVAR)
                  (PROGN (push FVARs (LIST TAG NAME))
                    'FVAR))
                  " " TAG ": " .P2 NAME)))
            (TERPRI OUTF])

```

(VBROKENDEF

```

[LAMBDA (DEF WHEN) (*)
  (PROG ((CA (VGET-COMPILED-CODE-BASE DEF))
        BEFORE AFTER SIZE FIRSTBYTE NEWCA)
        (SETQ FIRSTBYTE (VGETBASE CA 3))
    NIL (*)
    [PROGN (*)
      (PROGN (*)
        (SETQ NEWCA CA)
        (SETQ AFTER T)) (*)

```

```

(COND
  (AFTER
    (bind OP
      do (SELECTQ [CADR (SETQ OP (\FINDOP (COND
        ((NEQ 0 (LOGAND (LRSH (VGETBASE NEWCA 4)
          14)
            1))
          (VGETBASEBYTE NEWCA (LOGXOR FIRSTBYTE 3)
            ))
          (T (VGETBASEBYTE NEWCA FIRSTBYTE])
        (-X- (RETURN))
        (GCONST NIL)
        (RETURN [[LAMBDA (CODEBASE OFFSET NEWVALUE)
          (DECLARE (LOCALVARS CODEBASE OFFSET NEWVALUE))
          (COND
            ((NEQ 0 (LOGAND (LRSH (VGETBASE CODEBASE 4)
              14)
                1))
              (VPUTBASEBYTE CODEBASE (LOGXOR OFFSET 3)
                NEWVALUE))
            (T (VPUTBASEBYTE CODEBASE OFFSET NEWVALUE]
              NEWCA FIRSTBYTE (VCAR.UFN (\FINDOP '\RETURN])
            NIL)
          (SETQ FIRSTBYTE (PLUS FIRSTBYTE 1 (CADDR OP]
        (RETURN NEWCA])
    )
  )
)

```

(DEFINEQ

(VCAR.UFN [LAMBDA (X)

(*
(*
(*

```

(COND
  [(VLISTP X)
    (COND
      ((ZEROP 1)
        (VGETBASEPTR X 0))
      (T (COND
        ((EQ (LRSH (VGETBASE X 0)
          12)
            0)
          (VGETBASEPTR (VGETBASEPTR X 0)
            0))
        (T (VGETBASEPTR X 0]
      ((NULL X)
        NIL)
      (T (SELECTQ T
        (T (LISPERROR "ARG NOT LIST" X))
        ((NIL V\CDR.UFN)
          (COND
            ((EQ X T)
              T)
            ((LITATOM X)
              NIL)
            (T '{car of non-list})))
        (COND
          ((EQ X T)
            T)
          ((STRINGP X)
            (LISPERROR "ARG NOT LIST" X))
          (T '{car of non-list}])
      )

```

(VCDR.UFN [LAMBDA (X)

(*
(*
(*

```

(COND
  [(VLISTP X)
    (COND
      ((ZEROP 1)
        (VGETBASEPTR X 2))
      (T (PROG ((Q (LRSH (VGETBASE X 0)
        12)))
        (RETURN (COND
          ((EQ Q 8)
            NIL)
          ((IGREATERP Q 8)
            (VADDBASE (VVAG2 (VHILOC X)
              (LOGAND (VLOLOC X)
                65280))
              (LLSH (IDIFFERENCE Q 8)
                1)))
          ((EQ Q 0)
            (VCDR.UFN (VGETBASEPTR X 0)))
        )

```

```

(T (VGETBASEPTR (VADDBASE (VVAG2 (VHILOC X)
                                (LOGAND (VLOLOC X)
                                          65280))
                                (LLSH Q 1))
  0]

```

```

((NULL X)
NIL)
(T (SELECTQ T
  ((T V\CDR.UFN)
  (LISPERROR "ARG NOT LIST" X))
  (NIL (COND
    ((LITATOM X)
     (VGETPROPLIST X))
    (T "{cdr of non-list}")))
  (COND
    ((STRINGP X)
     (LISPERROR "ARG NOT LIST" X))
    (T "{cdr of non-list}")))
)

```

(DEFINEQ

(VCOPY

```

[LAMBDA (X) (*)
  (SELECTQ (TYPENAME X) (*)
    ((LITATOM NEW-ATOM)
     (VATOMNUMBER X T))
    (VLISTP (PROG [(R (REVERSE X))
                   (V (VCOPY (CDR (LAST X)
                              LP (COND
                                ((LISTP R)
                                 (SETQ V (CONS (VCOPY (CAR R)
                                                    V))
                                 (SETQ R (CDR R))
                                 (GO LP)))
                                (RETURN V))])
              ((FIXP SMALLP)
               (PROG (V)
                 [COND
                  [(IGREATERP 0 X) (*)
                   (COND
                    ((IGREATERP X -65537) (*)
                     (RETURN (VADDBASE (VVAG2 15 0)
                                         (LOGAND X 65535)
                                         ((ILESSP X 65536) (*)
                                          (RETURN (VADDBASE (VVAG2 14 0)
                                                              X] (*)
                                                              (SETQ V (CREATECELL 2))
                                                              (VPUTBASE V 0 (LOGOR (COND
                                                                ((IGREATERP 0 X)
                                                                 32768)
                                                                (T 0))
                                                                (LOGAND (LRSH X 16)
                                                                 32767)))
                                                              (VPUTBASE V 1 (LOGAND X 65535))
                                                              (RETURN V))])
                  (ONED-ARRAY (%COPY-ONED-ARRAY X)) (*)
                  (STRINGP
                   (%COPY-STRING-TO-ARRAY X))
                  (FLOATP (PROG ((VAL (CREATECELL 3))
                                (SELECTQ (SYSTEMTYPE)
                                  ((ALTO D)
                                   (VPUTBASE VAL 0 (\GETBASE X 0))
                                   (VPUTBASE VAL 1 (\GETBASE X 1)))
                                   (MKI.IEEE X VAL))
                                   (RETURN VAL)))
                  (CHARACTER (VVAG2 7 (CL:CHAR-CODE X)))
                  (ERROR X "can't be copied to remote file"]
)

```

(VUNCOPY

```

[LAMBDA (X CARLVL CDRLVL) (*)
  (SELECTC (VNTYPX X)
    (1 (COND
      ((EQ (VHILOC X)
           14) (*)
       (VLOLOC X))
      (T (IPLUS (VLOLOC X)
                -65536)))
    (2 (create FIXP
              HINUM _ (VGETBASE X 0)
              LONUM _ (VGETBASE X 1))) (*)
    (3 (create FLOATP
              HIWORD _ (VGETBASE X 0)

```

```

LOWORD _ (VGETBASE X 1))
(4 (VATOM (VLOLOC X)))
(7 (PROG ([PTR (COND
    ((NEQ 0 (LOGAND (LRSH (VGETBASE X 2)
        14)
        1))
    (%ARRAY-BASE X))
    (T (VGETBASEPTR X 0)
    [OFFST (COND
        ((NEQ 0 (LOGAND (LRSH (VGETBASE X 2)
            14)
            1))
        (%ARRAY-OFFSET X))
        (T (VGETBASE X 3)
        (LENGTH (\GETBASEFIXP X 4))
        (I 1)
        STR)
        (SETQ STR (ALLOCSTRING LENGTH))
        (FRPTQ LENGTH (RPLSTRING STR I (FCHARACTER (VGETBASEBYTE PTR OFFST)))
        (SETQ I (PLUS I 1))
        (SETQ OFFST (PLUS OFFST 1)))
        (RETURN STR)))
(9 (\VAG2 7 (VLOLOC X)))
(%ONED-ARRAY (LET ((SIZE (\GETBASEFIXP X 6))
    (BASE (VGETBASEPTR X 0))
    (OFFSET (VGETBASE X 3))
    (TYPENUMBER (LOGAND (VGETBASE X 2)
        255))
    NCELLS LOCAL-ARRAY LOCAL-BASE)
    (if (EQ (%TYPENUMBER-TO-GC-TYPE TYPENUMBER)
        1)
        then (VTYPEDPOINTER (VTYPENAME X)
            X)
        else (SETQ NCELLS (LRSH (IPLUS (ITIMES (IPLUS SIZE OFFSET)
            (%TYPENUMBER-TO-BITS-PER-ELEMENT TYPENUMBER)
            )
            31)
            5))
        (SETQ LOCAL-ARRAY (create ONED-ARRAY))
        (SETQ LOCAL-BASE (\ALLOCBLOCK NCELLS))
        (freplace (ONED-ARRAY BASE) of LOCAL-ARRAY with LOCAL-BASE)
        (freplace (ONED-ARRAY STRING-P) of LOCAL-ARRAY with (%CHAR-TYPE-P TYPENUMBER))
        (freplace (ONED-ARRAY FILL-POINTER-P) of LOCAL-ARRAY
            with (NEQ 0 (LOGAND (LRSH (VGETBASE X 2)
                9)
                1)))
        (freplace (ONED-ARRAY TYPE-NUMBER) of LOCAL-ARRAY with TYPENUMBER)
        (freplace (ONED-ARRAY FILL-POINTER) of LOCAL-ARRAY with (\GETBASEFIXP X 4))
        (if (NEQ OFFSET 0)
            then (freplace (ONED-ARRAY OFFSET) of LOCAL-ARRAY with OFFSET)
            (freplace (ONED-ARRAY DISPLACED-P) of LOCAL-ARRAY with T))
        (freplace (ONED-ARRAY TOTAL-SIZE) of LOCAL-ARRAY with SIZE)
        (for I from 0 to (SUB1 (LLSH NCELLS 1)) do (\PUTBASE LOCAL-BASE I
            (VGETBASE BASE I)))
        LOCAL-ARRAY)))
(5 (COND
    [(VLISTP X)
    (COND
        ((EQ CDRLVL 0)
        ' (--))
        (T (CONS [COND
            ([OR (EQ CARLVL 0)
                (AND (OR (EQ CARLVL 1)
                    (EQ CDRLVL 1))
                (VLISTP (VCAR.UFN X)
                ' &)
                (T (VUNCOPY (VCAR.UFN X)
                    (AND CARLVL (SUB1 CARLVL))
                    (AND CDRLVL (SUB1 CDRLVL))
                (VUNCOPY (VCDR.UFN X)
                    CARLVL
                    (AND CDRLVL (SUB1 CDRLVL))
                (T
                    (VTYPEDPOINTER 'LISTP X)))
                (VTYPEDPOINTER NIL X))
                (VTYPEDPOINTER (VTYPENAME X)
                    X])
        )
    (DEFINEQ
    (VGETBASEBYTE
    [LAMBDA (PTR N)
    (COND
        ((EVENP N)

```



```

LP
[COND
((NEQ 0 (SETQ HASHENT (VGETBASE (VVAG2 21 0)
                                HASH)))
(COND
([STREQUAL (CL:SYMBOL-NAME BASE)
(VSYMBOL.PNAME (SETQ ATM# (SUB1 HASHENT)
(RETURN ATM#))
(T
[SETQ HASH (IPLUS16 HASH (OR REPROBE (SETQ REPROBE (LOGAND 63 (LOGOR 1 (LOGXOR FIRSTBYTE
                                                                HASH)
                                                                (GO LP)
(RETURN (PROGN (LET ((NEWATOM (VNOSUCHATOM BASE OFFST LEN FATP FATPARSEENP))
NIL NEWATOM])

```

(VLOOKUP-SYMBOL

```

[LAMBDA (TABLE STRING SXHASH ENTRY-HASH)
(LET* ((VEC (VGETBASEPTR TABLE 0))
(HASH (VGETBASEPTR TABLE 2))
(LEN (\GETBASEFIXP VEC 6))
[H2 (ADD1 (IREMAINDER SXHASH (IDIFFERENCE LEN 2]
)
(DECLARE (TYPE (CL:SIMPLE-ARRAY (CL:UNSIGNED-BYTE 8))
HASH)
(TYPE (CL:SIMPLE-ARRAY (CL:UNSIGNED-BYTE 16))
VEC))
(PROG ((INDEX-VAR (IREMAINDER SXHASH LEN))
SYMBOL-NUMBER EHASH)
(IF NIL
THEN (CL:FORMAT T "Probe @ ~s~%" INDEX-VAR))
LOOP
(SETQ EHASH (VGETBASEBYTE (VGETBASEPTR HASH 0)
INDEX-VAR))
[COND
[(EQL EHASH ENTRY-HASH)
(IF NIL
THEN (CL:FORMAT T "Entry hash MATCHES~%")
(LET [(SYMBOL-NAME (VSYMBOL.PNAME (SETQ SYMBOL-NUMBER (VGETBASE (VGETBASEPTR VEC 0)
INDEX-VAR]
)
(IF NIL
THEN (CL:FORMAT T "Got symbol index~%")
(COND
((STREQUAL SYMBOL-NAME STRING)
(IF NIL
THEN (CL:FORMAT T " found~%")
(GO DOIT))
(T (IF NIL
THEN (CL:FORMAT T "Didn't match~%")
(EQL 0 EHASH)
(IF NIL
THEN (CL:FORMAT T "Hit deleted entry (no match)~%")
(SETQ INDEX-VAR NIL)
(GO DOIT))
(T (IF NIL
THEN (CL:FORMAT T "Entry hash does not match~%")
(SETQ INDEX-VAR (IREMAINDER (IPLUS INDEX-VAR H2)
LEN))
(IF NIL
THEN (CL:FORMAT T "Reprobe @ ~s~%" INDEX-VAR))
(GO LOOP)
DOIT
(RETURN SYMBOL-NUMBER])

```

(VFIND.PACKAGE

```

[LAMBDA (NAME)
(PROG ((ITEM (MKSTRING NAME))
(HA READSYS.PACKAGE.FROM.NAME)
BITS INDEX SLOT SKEY FIRSTINDEX REPROBE LIMIT ABASE VALUE)
(SETQ BITS (STRINGHASHBITS ITEM))
(SETQ INDEX (LOGAND BITS (VGETBASE HA 1)))
(SETQ ABASE (VGETBASEPTR HA 2))
(SETQ FIRSTINDEX INDEX)
(SETQ REPROBE (LOGOR (LOGAND (LOGXOR BITS (LRSH BITS 8))
(IMIN 63 (VGETBASE HA 1)))
1))
(SETQ LIMIT (VGETBASE HA 1))
LP (SETQ SLOT ([LAMBDA (BASEA0182)
(DECLARE (LOCALVARS BASEA0182))
(VADDBASE (VADDBASE BASEA0182 INDEX)
INDEX]

```



```

(COND
  (INDEX (COND
    ((NEQ PREV HEAD)
      (LET* ((A0347 PREV)
             (A0346 (V\CDR.UFN A0347))
             (A0349 TABLE)
             (A0348 (V\CDR.UFN A0349))
             (A0351 HEAD)
             (A0350 (V\CDR.UFN A0351)))
            (V\CDR.UFN (RPLACD A0347 A0348))
            (V\CDR.UFN (RPLACD A0349 A0350))
            (V\CDR.UFN (RPLACD A0351 TABLE))
            A0346)))
      (SETQ SYM INDEX)
      (SETQ WHERE :INHERITED)
      (SETQ DONE T))
    (T)
  (PROGN (SETQ PREV (PROG1 TABLE
    (PROGN (SETQ TABLE (V\CDR.UFN TABLE)
      NIL)))
    NIL)
  (GO USED-PACKAGE-LOOP]
  (CL:VALUES SYM WHERE])

```

(VPACKAGE.NAME

```

[LAMBDA (RMPKG) (*)
  (AND RMPKG (V\UNCOPY (VGETBASEPTR RMPKG 4]))

```

(VMKATOM

```

[LAMBDA (BASE OFFST LEN FATP NONNUMERICP) (*)
  (PROG ([FATCHARSEENP (AND FATP (NOT (NULL (for I from OFFST to (SUB1 (IPLUS OFFST LEN))
    suchthat (IGREATERP (VGETBASE BASE I)
      255]
    HASH HASHENT ATM# PNBASE FIRSTCHAR FIRSTBYTE REPROBE) (*)
  (COND
    ((EQ LEN 0) (*)
      (SETQ HASH 0)
      (SETQ FIRSTBYTE 255)
      (GO LP)))
    (SETQ FIRSTCHAR (NTHCHARCODE BASE OFFST)) (*)
  NIL (SETQ FIRSTBYTE (LOGAND FIRSTCHAR 255)) (*)
  [PROGN (SETQ HASH (LLSH FIRSTBYTE 8))
    (for CHAR# from (ADD1 OFFST) to (SUB1 (IPLUS OFFST LEN))
      do (SETQ HASH (IPLUS16 (IPLUS16 (SETQ HASH (IPLUS16 HASH (LLSH (LOGAND HASH 4095)
        2)))
        (LLSH (LOGAND HASH 255)
          8))
        (NTHCHARCODE BASE CHAR#]
    LP (COND
      ((NEQ 0 (SETQ HASHENT (VGETBASE (VVAG2 21 0)
        HASH))) (*)
      (COND
        ((EQ (VATOM (SETQ ATM# (SUB1 HASHENT))
          BASE)
          (RETURN (VADDBASE (VVAG2 0 0)
            ATM#))) (*)
        (T
          [SETQ HASH (IPLUS16 HASH (OR REPROBE (SETQ REPROBE (LOGAND 63 (LOGOR 1 (LOGXOR FIRSTBYTE
            HASH]
            (GO LP]
          (RETURN (PROGN (LET ((NEWATOM (VNOSUCHATOM BASE OFFST LEN FATP FATCHARSEENP)))
            NIL NEWATOM])

```

(VGETTOPVAL

```

[LAMBDA (X) (*)
  (VGETBASEPTR (VATOMCELL X 12)
    0])

```

(VGETPROPLIST

```

[LAMBDA (ATM) (*)
  (VGETBASEPTR (VATOMCELL ATM (CONSTANT 2))
    0])

```

(VSETTOPVAL

```

[LAMBDA (ATM VAL) (*)
  (SELECTQ ATM
    (NIL (AND VAL (LISPERROR "ATTEMPT TO SET NIL OR T" VAL)))
    (T (OR (EQ VAL T)

```


)

:: YOU MUST REMAKE THIS FILE using (DORENAME 'R) (after CONNing to library) whenever the SYSOUT layout changes in LLPARAMS (e.g., if
:: MDSTypeTable moves)

(FILESLOAD VMEM)

(RPAQQ **RDVALS** ((\RPTSIZE)
(\MaxTypeNumber)
(\AtomFrLst)
(\ArrayFrLst)
(\ArrayFrLst2)))

(RPAQQ **RDPTRS** ((\REALPAGETABLE)
(\FREEBLOCKBUCKETS)))

(DECLARE%: EVAL@COMPILE DONTCOPY

(FILESLOAD (LOADCOMP)
VMEM)

)

FUNCTION INDEX

VBROKENDEF	22	VPRINTVA	8	V\CDR.UFN	23
VCHECKFPTOVP	3	VRAIDCOMMAND	4	V\CHECKARRAYBLOCK	15
VCHECKFPTOVP1	3	VRAIDROOTFRAME	8	V\COPY	24
VCHECKIFPAGE	2	VRAIDSHOWFRAME	7	V\GET-COMPILED-CODE-BASE	31
VCHECKPAGEMAP	2	VRAIDSTACKCMD	7	V\GETBASEBYTE	25
VFIND.PACKAGE	28	VREADATOM	9	V\LOCKEDPAGEP	2
VFIND.SYMBOL	29	VREADOCT	9	V\LOOKUPPAGEMAP	2
VGETDEFN	31	VREADPAGEMAP	1	V\MKATOM	30
VGETPROPLIST	30	VREADPAGEMAPBLOCK	1	V\PARSEARRAYSPACE	16
VGETTOPVAL	30	VREADVA	8	V\PARSEARRAYSPACE1	16
VLISTP	31	VSETTOPVAL	30	V\PRINTBF	12
VLOOKUP-SYMBOL	28	VSHOWSTACKBLOCK1	10	V\PRINTFPTOVP	3
VMAKE.LOCAL.ATOM	26	VSHOWSTACKBLOCKS	9	V\PRINTFRAME	13
VNOSUCHATOM	10	VSYMBOL.PACKAGE	27	V\PRINTSTK	15
VNTYPX	26	VSYMBOL.PNAME	27	V\PUTBASEBYTE	26
VOLD.FIND.SYMBOL	27	VSYMBOL.VALUE	27	V\SCANFORNTENTRY	15
VPACKAGE.NAME	30	VTYPENAME	26	V\SHOWPAGETABLE	3
VPRINCOPY	10	VUNCOPYATOM	26	V\STKNAME	12
VPRINTADDRS	8	V\ATOMCELL	31	V\TYPENAMEFROMNUMBER	26
VPRINTCODE	17	V\BACKTRACE	11	V\UNCOPY	24
VPRINTCODENT	22	V\CAR.UFN	23		

VARIABLE INDEX

COPYATOMSTR	31	RDPTRS	32	RVALS	32
-------------------	----	--------------	----	-------------	----
