

File created: 9-Nov-92 03:25:43 {Pele:mv:envos}<LispCore>library>READSYS.;3

changes to: (FNS VATOMNUMBER)

previous date: 12-Jun-90 10:57:50 {Pele:mv:envos}<LispCore>library>READSYS.;2

Read Table: INTERLISP

Package: INTERLISP

Format: XCCS

::  
;; Copyright (c) 1982, 1983, 1984, 1985, 1986, 1987, 1990, 1992 by Venue & Xerox Corporation. All rights reserved.

### (RPAQQ READSYSCOMS

```
((FNS READSYS TELERAID VLISTGET VLOADFNS VLOADFILEPKGTYPECHANGE VLOADFUNCTIONS VLOADVAR VLOADVARS VRAID
VSAVEWORK SHOWREMOTESCREEN VGETVAL VINSPECT VUNSAVEDEF VCADR VPUTDEFN VYANKDEF)
[INITVARS (RDSYSINIT)
(ATOMPAGELST NIL)
(ATOMCACHE NIL)
(NEWATOMARRAY (HASHARRAY 30))
(TELERAIDPRINTLEVEL ' (2 . 20)
(FNS VATOM VATOMNUMBER)
(DECLARE%: EVAL@COMPILE DONTCOPY (PROP (DMACRO MACRO)
IEQ)
DONTEVAL@LOAD
(FILE (LOADCOMP)
VMEM))
(FILE VMEM))])
```

(DEFINEQ

### (READSYS

```
[LAMBDA (FILE WRITEABLE) ; Edited 6-Mar-87 17:09 by raf
(COND
[FILE (INITVMEM FILE WRITEABLE)
;; clear atom cache
(for X in ATOMPAGELST do (for I from 0 to 255 do (FASTSETA (CDR X)
I 0)))
;; Cache the remote package globals, also used by READSYS.HAS.PACKAGES to determine whether packages are on in the remote
;; sysout.
(SETQ READSYS.PACKAGE.FROM.NAME (VSYMBOL.VALUE '*PACKAGE-FROM-NAME*))
(SETQ READSYS.PACKAGE.FROM.INDEX (VSYMBOL.VALUE '*PACKAGE-FROM-INDEX*))
;; initialize those variables which are renamed 'pointers', e.g., the array free list
[for X in RDPTRS do (SET (PACK* 'V (SUBATOM (CAR X)
2 -1))
(VGETTOPVAL (CAR X)
;; Initialize those variables which are renamed 'values', e.g., \AtomFrLst = # of allocated atoms
(for X in RDVALS do (SET (PACK* 'V (SUBATOM (CAR X)
2 -1))
(VGETVAL (CAR X)
((LISTP VMEMFILE)
(CLOSEREMOTEVMEMFILE))
(T (CLOSEVMEMFILE]))])
```

### (TELERAID

```
[LAMBDA (HOST RAIDIX) (* bvm%: "13-Jul-84 17:24")
(RESETLST
[COND
(HOST (RESETSAVE NIL ' (CLOSEVMEMFILE))
(READSYS (LIST HOST]
(COND
((LISTP VMEMFILE)
(VRAID RAIDIX))))])
```

### (VLISTGET

```
[LAMBDA (LST TOKEN) (* edited%: "11-Jun-85 04:24")
(AND LST (if (EQ TOKEN (V\UNCOPY (V\CAR.UFN LST)))
then (V\UNCOPY (V\CAR.UFN (V\CDR.UFN LST)))
else (VLISTGET (V\CDR.UFN (V\CDR.UFN LST))
TOKEN]))
```

### (VLOADFNS

```
[LAMBDA (FNS) (* mpl " 8-Aug-85 23:05")
(for FN inside FNS do (PRINTOUT T "Reading function " FN)
[SAVEPUT FN 'EXPR (LET [(DEFN (V\UNCOPY (VGETDEFN FN)
(COND
[(NLISTP DEFN)
```

(\* \* Hmm, must have been a compiled function. Let's try to get its proplist and save the defn from the EXPR prop)

```

(LET [(PLIST (V\UNCOPY (VGETPROPLIST FN)
(COND
  ([AND (LISTP PLIST)
    (LISTP (LISTGET PLIST 'EXPR)
    (LISTGET PLIST 'EXPR)
(T DEFN]
(TERPRI T])

```

**(VLOADFILEPKGTYPECHANGE**

(\* gbn "19-Jun-86 00:46")

```

[LAMBDA (CHANGE FILEPKGTYPE)
(PRINTOUT T "Reading" %, FILEPKGTYPE %, CHANGE "...")
[LET [(PLIST (V\UNCOPY (VGETPROPLIST CHANGE)
(COND
  ([AND (LITATOM CHANGE)
    (LISTP PLIST)
    (LISTP (LISTGET PLIST FILEPKGTYPE))
    (LISTP (NLSETQ (PUTDEF CHANGE FILEPKGTYPE (LISTGET PLIST FILEPKGTYPE)
(PRINTOUT T "done"))
(T (PRINTOUT T "failed")
(PRINTOUT T T])

```

**(VLOADFUNCTIONS**

(\* gbn "18-Jun-86 22:48")

```

[LAMBDA (FUNCTIONS)
(for FUNCTION inside FUNCTIONS do (PRINTOUT T "Reading function " FUNCTION)
[SAVEPUT FUNCTION 'FUNCTIONS (LET [(PLIST (V\UNCOPY (VGETPROPLIST FUNCTION)
(COND
  ([AND (LISTP PLIST)
    (LISTP (LISTGET PLIST
'FUNCTIONS]
(LISTGET PLIST 'FUNCTIONS]
(TERPRI T])

```

**(VLOADVAR**

(\* edited%: "11-Jun-85 03:09")

```

[LAMBDA (VAR)
(SAVESET VAR (VGETVAL VAR)
T])

```

**(VLOADVARS**

(\* Imm " 7-Aug-85 18:44")

```

[LAMBDA (VARS)
(for VAR inside VARS do (PRINTOUT T "Reading variable: " VAR)
(SAVEPUT VAR 'VALUE (VGETVAL VAR)
(TERPRI T])

```

**(VRAID**

(\* bvm%: "23-Jan-86 18:44")

```

[LAMBDA (RAIDIX)
(DECLARE (SPECVARS RAIDIX ROOTFRAME ALINKS? FRAME# REMOTESCREEN VPRINTLEVEL))
(printout T "virtual RAID" T)
(OR RAIDIX (SETQ RAIDIX 8))
(PROG ((ROOTFRAME)
(ALINKS? T)
(FRAME#)
(REMOTESCREEN)
(VPRINTLEVEL TELERAIDPRINTLEVEL))
(RESETLST
(RESETSAVE (OUTPUT T))
(RESETSAVE (INTCHAR (CHARCODE ^G)))
(SETQ .I2 (NUMFORMATCODE (LIST 'FIX 2 RAIDIX)))
(SETQ .I5 (NUMFORMATCODE (LIST 'FIX 5 RAIDIX)))
(SETQ .I6 (NUMFORMATCODE (LIST 'FIX 6 RAIDIX)))
(SETQ .I7 (NUMFORMATCODE (LIST 'FIX 7 RAIDIX)))
[bind RESULT until [SETQ RESULT (ERSETQ (when (SETQ $$VAL (VRAIDCOMMAND)) do (RETURN $$VAL)
finally (COND
  ((AND (LISTP VMEMFILE)
    (EQ (CAR RESULT)
'RETURN))
(CLEARPAGECACHE)
(REMOTERETURN)]])

```

**(VSAVEWORK**

(\* gbn "19-Jun-86 00:46")

```

[LAMBDA NIL
(LET (FNS VARS FILES CHANGES (ALLCHANGES (LIST NIL)))
(PRINTOUT T "Functions on CHANGEDFNSLSLST: " (SETQ FNS (VGETVAL 'CHANGEDFNSLSLST))
T)
(PRINTOUT T "Variables on CHANGEDVARSLSLST: " (SETQ VARS (VGETVAL 'CHANGEDVARSLSLST))
T)
(PRINTOUT T "Files on FILELST: " (SETQ FILES (VGETVAL 'FILELST))
T)
(for FILE in FILES

```

```

do [SETQ CHANGES (CDR (VLISTGET (VGETPROPLIST FILE)
                                'FILE)]
  (if CHANGES
    then (PRINTOUT T FILE " has changes " CHANGES T)
    [for TYPEPAIR in CHANGES
      do (LET ((FILEPKGTYPE (CAR TYPEPAIR))
              (FILEPKGTYPECHANGES (CDR TYPEPAIR)))
          (SELECTQ FILEPKGTYPE
            (FNS (SETQ FNS (UNION FNS FILEPKGTYPECHANGES)))
            (VARS (SETQ VARS (UNION VARS FILEPKGTYPECHANGES)))
            (PROGN (* "try to grab random filepkgtypes off the prop list. It's gets lots
                    of cases, so is better than just giving up.")
                  (PUTASSOC FILEPKGTYPE (UNION (CDR (ASSOC FILEPKGTYPE ALLCHANGES))
                                              FILEPKGTYPECHANGES)
                                ALLCHANGES]
          else (PRINTOUT T FILE " has no changes recorded." T)))
  (for FN in (INTERSECTION FNS FNS) when (EQ 'Y (ASKUSER DWIMWAIT 'Y (LIST "save function" FN)
                                          NIL T))
    do (VLOADFNS FN))
  (for VAR in (INTERSECTION VARS VARS) when (EQ 'Y (ASKUSER DWIMWAIT 'Y (LIST "save variable" VAR)
                                          NIL T))
    do (VLOADVARS VAR))
  (for TYPEPAIR in ALLCHANGES
    do (LET ((FILEPKGTYPE (CAR TYPEPAIR))
            (for FILEPKGTYPECHANGE in (CDR TYPEPAIR)
              do (if (EQ 'Y (ASKUSER DWIMWAIT 'Y (LIST "save" FILEPKGTYPE FILEPKGTYPECHANGE)
                                          NIL T))
                    then (VLOADFILEPKGTYPECHANGE FILEPKGTYPECHANGE FILEPKGTYPE))

```

(SHOWREMOTESCREEN

; Edited 23-Nov-86 11:58 by MASINTER

```

[LAMBDA NIL
  (DECLARE (USEDFREE REMOTESCREEN))
  (RESETLST
    (PROG ((WINDOW (AND (BOUNDP 'REMOTESCREEN)
                       REMOTESCREEN))
          HEIGHT WIDTH BITMAPBASE LASTPAGE NWORDS POS NEWPOS MINBOTTOM MINLEFT DELTAX DELTAY REG X Y)
      (COND
        ((NOT WINDOW)
          (SETQ WINDOW (CREATEW [CREATEREGION 0 0 (SETQ WIDTH (VGETVAL 'SCREENWIDTH))
                                (SETQ HEIGHT (VGETVAL 'SCREENHEIGHT)
                                NIL 0 T))
                    ; WINDOW has the dimensions of the remote screen
          [SETQ BITMAPBASE (fetch BITMAPBASE of (WINDOWPROP WINDOW 'IMAGECOVERED)]
          (SETQ NWORDS (TIMES HEIGHT (QUOTIENT WIDTH 16)))

```

;;; Now fetch remote display to local window. Display memory is contiguous bitmap, and its virtual address is known constant

```

[COND
  [(LISTP VMEMFILE)
    ; Remote machine. Get it a page at a time with REMOTEPMAP
    ; then finish any leftover specially
    (for I from \VP.DISPLAY to [SUB1 (SETQ LASTPAGE (IPLUS \VP.DISPLAY (IQUOTIENT NWORDS 256)]
      do (REMOTEPMAP VMEMFILE I BITMAPBASE)
        (SETQ BITMAPBASE (\ADDBASE BITMAPBASE 256)))
    (COND
      ((NEQ (SETQ NWORDS (IMOD NWORDS 256))
        0)
        ; Screen bitmap not an integral number of pages, so have to get
        ; the rest of it more carefully
        (LET [(BUF (NCREATE 'VMEMPAGEP]
              (REMOTEPMAP VMEMFILE LASTPAGE BUF)
              (\BLT BITMAPBASE BUF NWORDS]
          (T (SETVMPTR (CL:* \VP.DISPLAY 256))
            (\BINS (GETSTREAM VMEMFILE)
                  BITMAPBASE 0 (CL:* \NP.DISPLAY 512)
                  (CL:* NWORDS 2]
            (SETQ REMOTESCREEN WINDOW))
          (T (SETQ WIDTH (WINDOWPROP WINDOW 'WIDTH))
            (SETQ HEIGHT (WINDOWPROP WINDOW 'HEIGHT))
            (MOVEW WINDOW 0 0)))
          (RESETSAVE NIL (LIST 'CLOSEW WINDOW))
          (OPENW WINDOW)
          [COND
            ((OR (GREATERP HEIGHT SCREENHEIGHT)
                 (GREATERP WIDTH SCREENWIDTH))
              ; Remote screen is bigger than local, so allow user to move
              ; window around
              (SETQ MINLEFT (IMIN 0 (IDIFFERENCE SCREENWIDTH WIDTH)))
              (SETQ MINBOTTOM (IMIN 0 (IDIFFERENCE SCREENHEIGHT HEIGHT)))
              (SETQ POS (CURSORPOSITION))
            (until (OR (READP T)
                      (NOT (OPENWP WINDOW)))
              do
                ; Keep window on top until user types something or explicitly
                ; closes the window
                (COND
                  ((AND POS (NOT (EQUAL (SETQ NEWPOS (CURSORPOSITION NIL NIL NEWPOS))
                                      POS)))
                    ; Track mouse while button down

```

```

(SETQ X (fetch (REGION LEFT) of REG))
(SETQ Y (fetch (REGION BOTTOM) of REG))
(SETQ DELTAX (IDIFFERENCE [IMAX MINLEFT (IMIN 0 (IPLUS X
(IDIFFERENCE (fetch XCOORD
of NEWPOS)
(fetch XCOORD
of POS]
X))
(SETQ DELTAY (IDIFFERENCE [IMAX MINBOTTOM (IMIN 0
(IPLUS Y
(IDIFFERENCE (fetch YCOORD
of NEWPOS)
(fetch YCOORD
of POS]
Y))
(COND
((OR (NEQ DELTAX 0)
(NEQ DELTAY 0))
;; Bound the movement so that window always covers our screen. Don't call MOVEW if no actual movement,
;; so as to avoid excess flashing
(RELMOVEW WINDOW (create POSITION
XCOORD _ DELTAX
YCOORD _ DELTAY]
(swaps POS NEWPOS)))
(TOTOPW WINDOW)
(BLOCK))))))

```

(VGETVAL

```

[LAMBDA (X) (* Imm "20-AUG-81 12:51")
(V\UNCOPY (VGETTOPVAL X])

```

(VINSPECT

```

[LAMBDA (HI LO ASTYPE) (* kbr%: " 8-Aug-85 19:05")
(* Virtual inspector. *)
(* TBW%: This is not completely generalized. *)
(PROG (PTR OBJECT D FIELDSPEC WINDOW)
(SETQ PTR (VVAG2 HI LO))
(SETQ OBJECT (NCREATE ASTYPE))
[FOR DESCRIPTOR IN (GETDESCRIPTORS ASTYPE) DO (SETQ D (CADR DESCRIPTOR))
(SETQ FIELDSPEC (CADDR DESCRIPTOR))
(COND
[(EQ FIELDSPEC 'POINTER)
(\PUTBASEPTR OBJECT D (V\UNCOPY (VGETBASEPTR PTR D)
(EQUAL FIELDSPEC '(BITS . 15))
(\PUTBASE OBJECT D (VGETBASE PTR D)
(SETQ WINDOW (INSPECT OBJECT ASTYPE))
(WINDOWPROP WINDOW 'TITLE (CONCAT (V\UNCOPY PTR)
" Inspector"))
(RETURN WINDOW])

```

(VUNSAVEDEF

```

[LAMBDA (SYMBOL) (* gbn " 8-Aug-85 15:37")
(for (X _ (VGETPROPLIST SYMBOL)) by (V\CDR.UFN (V\CDR.UFN X)) while X
do (SELECTQ (V\UNCOPY (V\CAR.UFN X))
(CODE (PRINTOUT T "Found a CODE property, doing UNSAVEDEF" T)
(VPUTDEFN SYMBOL (LOGOR (VGETBASEPTR0 (VCADR X))
(LLSH 1 31)))
(RETURN))
(BROKEN (PRINTOUT T "Found a BROKEN property, unbreaking" T)
(RETURN (VYANKDEF SYMBOL (VCADR X))))
(ADVISED (PRINTOUT T "Found a ADVISED property, unbreaking" T)
(RETURN (VYANKDEF SYMBOL (VCADR X))))
NIL)
finally (PRINTOUT T "No CODE property found" T])

```

(VCADR

```

[LAMBDA (X)
(V\CAR.UFN (V\CDR.UFN X])

```

(VPUTDEFN

```

[LAMBDA (SYMBOL VDEF CODEP) (* gbn " 8-Aug-85 15:40")
(LET ((CELL (V\ATOMCELL SYMBOL 10)))
(VPUTBASE0 CELL (LRSH VDEF 16))
(VPUTBASE0 (ADD1 CELL)
(LOGAND VDEF 65535])

```

(VYANKDEF

```

[LAMBDA (NEWSYMBOL OLDSYMBOL)
(VPUTDEFN NEWSYMBOL (VGETDEFN OLDSYMBOL])

```

)

(RPAQ? **RDSYSINIT** )

(RPAQ? **ATOMPAGE**LST NIL)

(RPAQ? **ATOMCACHE** NIL)

(RPAQ? **NEWATOMARRAY** (HASHARRAY 30))

(RPAQ? **TELERAIDPRINTLEVEL** ' (2 . 20))

(DEFINEQ

**(VATOM**

[LAMBDA (N)

(\* Imm " 6-Aug-84 13:20")

(\* Converts a VM atom number into a Lisp atom.)

(PROG ((PAGE (FASSOC (LRSH N 8)  
                  ATOMPAGE)LST))

  ATM FPTR)

  (COND

    ((AND PAGE (NEQ (SETQ ATM (FASTELT (CDR PAGE)  
  (LOGAND N 255))))

      0))

    (RETURN ATM)))

  (SETQ ATM (VUNCOPYATOM N))

  [COND

    ((NULL PAGE)

      (SETQ PAGE (CONS (LRSH N 8)  
                      (POINTERARRAY 256 0)))

    (COND

      (ATOMCACHE (ATTACH PAGE ATOMCACHE))

      (T (SETQ ATOMPAGELST (NCONC ATOMPAGELST (SETQ ATOMCACHE (LIST PAGE]

          (FASTSETA (CDR PAGE)

          (LOGAND N 255)

          ATOM)

      (RETURN ATM])

**(VATOMNUMBER**

[LAMBDA (ATOM NEWOK)

; Edited 9-Nov-92 03:24 by sybalsky:mv:envos

;;; See the comment on MAKE.LOCAL.ATOM for a warning about symbols being created with the wrong package.

(COND

  ((FIXP ATOM)

    ;; ALREADY HAVE THE ATOM'S NUMBER

  ATOM)

  (T (if (READSYS.HAS.PACKAGES)

**then** [VFIND.SYMBOL (CL:SYMBOL-NAME ATOM)

              (VFIND.PACKAGE (CL:PACKAGE-NAME (CL:SYMBOL-PACKAGE ATOM]

**else** (VOLD.FIND.SYMBOL ATOM 1 (NCHARS ATOM])

)

(DECLARE%: EVAL@COMPILE DONTCOPY

(PUTPROPS **IEQ DMACRO** (= . EQ))

(PUTPROPS **IEQ MACRO** ((X Y)  
                      (IEQP X Y)))

(FILESLOAD (LOADCOMP)

  VMEM)

)

(FILESLOAD VMEM)

(PUTPROPS **READSYS COPYRIGHT** ("Venue & Xerox Corporation" 1982 1983 1984 1985 1986 1987 1990 1992))

---

**FUNCTION INDEX**

READSYS .....	1	VCADR .....	4	VLOADFNS .....	1	VRAID .....	2
SHOWREMOTESCREEN .....	3	VGETVAL .....	4	VLOADFUNCTIONS .....	2	VSAVEWORK .....	2
TELERAID .....	1	VINSPECT .....	4	VLOADVAR .....	2	VUNSAVEDEF .....	4
VATOM .....	5	VLISTGET .....	1	VLOADVARS .....	2	VYANKDEF .....	4
VATOMNUMBER .....	5	VLOADFILEPKGTYPECHANGE ..	2	VPUTDEFN .....	4		

---

**VARIABLE INDEX**

ATOMCACHE .....	5	ATOMPAGELST .....	5	NEWATOMARRAY .....	5	RDSYSINIT .....	5	TELERAIDPRINTLEVEL	5
-----------------	---	-------------------	---	--------------------	---	-----------------	---	--------------------	---

---

**MACRO INDEX**

IEQ .....	5
-----------	---

---