

File created: 16-May-90 12:27:02 {DSK}<usr>local>lde>lispcore>sources>CLISP.;2

changes to: (VARS CLISPCOMS)

previous date: 26-Nov-86 12:32:58 {DSK}<usr>local>lde>lispcore>sources>CLISP.;1

Read Table: INTERLISP

Package: INTERLISP

Format: XCCS

```
::
:: Copyright (c) 1982, 1983, 1984, 1985, 1986, 1990 by Venue & Xerox Corporation. All rights reserved.
:: The following program was created in 1982 but has not been published
:: within the meaning of the copyright law, is furnished under license,
:: and may not be used, copied and/or disclosed except in accordance
:: with the terms of said license.
```

(RPAQQ CLISPCOMS

[(COMS

; DWIM stuff

```
[INITVARS (NOFIXFNSLST0)
           (NOFIXVARSLST0)
           (NOSPELLFLG)
           (LPARKEY 9)
           (RPARKEY 0)
           (WTFIXCHCONLST ' (NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL
                             NIL NIL))
           (WTFIXCHCONLST1 ' (NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL
                              NIL NIL))

(USERMACROS FIX8 FIX9)
(ADDVARS (DWIMUSERFORMS)
         (LAMBDA SPLST LAMBDA NLAMBDA)
         (OKREEVALST AND OR PROGN SAVESETQ CAR CDR ADD1 SUB1 CONS LIST EQ EQUAL PRINT PRIN1 APPEND
          NEQ NOT NULL)
         (NOFIXFNSLST)
         (NOFIXVARSLST)
         (GLOBALVARS)
         (LOCALVARS)
         (SPECVARS)
         (NLAMA)
         (NLAML)
         (LAMA)
         (LAMS))

(P (MOVD? 'NILL 'FREEVARS))
(PROP FILEDEF BREAKDOWN CALLS CLISPRECORD SETUPHASHARRAY MAKEMATCH)
(VARS (DWIMIFYFLG 'EVAL)
      (COMPILEUSERFN 'COMPILEUSERFN)
      (CLISPTRANFLG 'CLISP% )
      (DWIMESSGAG))

(INITVARS (DWIMCHECK#ARGSFLG T)
          (DWIMCHECKPROGLABELSFLG T)
          (%#CLISPARRAY 250)
          (RECORDHASHFLG T)
          (CLISPRETRANFLG))

(ADDVARS (DWIMEQUIVLST))
(USERMACROS DW !DW CLISP%: NOCLISP PPT))
(COMS (* CLISP props)
      (PROP CLISPSTYPE %')
      [E (SETQQ CLISPCHARS
              (^ * / + - = _ %: %' ~ +- ~= < > @ ! ← ↑))
        (CLISPDEC ' (STANDARD MIXED))]
      [VARS (CLISPFLG T)
            (CLISPCHARS ' (^ * / + - = _ %: %' ~ +- ~= < > @ ! ← ↑)]
      [INITVARS (CLISPHelpFLG T)
                (TREATASCLISPFLG)
                (CLISPINFIXSPLST)
                (CLISPCHARRAY (MAKEBITTABLE CLISPCHARS))
                [LEFT.ARROWS.BITTABLE (MAKEBITTABLE '(_ ←)
                (LEFT.ARROW '_)
                (CLISPISWORDSPLST)
                (CLISPLASTSUB (CONS))
                (CHECKCARATOMFLG)
                (CLISPARIHOPLST '(+ - * / +- LT GT lt gt GEQ LEQ GE LE geq leq ge le))
                (CLISPARIHCLASSLST '(INTEGER FIXED MIXED FLOATING))
                (DWIMINMACROSFNG NIL))
            (IFPROP (CLISPSTYPE LISPFN UNARYOP CLISPCLASS CLISPCLASSDEF CLISPNEG CLISPBRACKET)
                    ↑ ^ * / + - = _ ← %: %' ~ +- ~= < > @ !)]
            (VARS DECLWORDS)
            (IFPROP (CLISPSTYPE LISPFN UNARYOP CLISPINFIX CLISPCLASS CLISPCLASSDEF CLISPNEG BROADSCOPE)
                    *
                    (PROGN DECLWORDS))
            (IFPROP (CLISPSTYPE LISPFN UNARYOP CLISPINFIX CLISPCLASS CLISPCLASSDEF CLISPNEG BROADSCOPE)
                    LT lt GT gt LE le GE ge LEQ leq GEQ geq EQ NEQ EQP EQUAL EQUALS NOT AND OR and or NOR nor
                    MEMBER SETQ IPLUS IMINUS IDIFFERENCE ITIMES IQUOTIENT ILESSP IGREATERP FPLUS FMINUS
                    FDIFFERENCE FTIMES FQUOTIENT FGTP PLUS MINUS DIFFERENCE TIMES QUOTIENT LESSP GREATERP EXPT
```

```

-> =>)
  (PROP SETFN ELT SETA)
  (OPTIMIZERS CLISP% ))
(PROP CLISPCOM AND OR and or ! ! CLISP clisp MATCH match)
(COMS (* IF)
  (VARS CLISPIFWORDSPLST)
  (INITVARS (CLISPIFTRANFLG T))
  (PROP CLISPCOM IF THEN ELSE ELSEIF if then else elseif))
(COMS (* I.S.OPR)
  (VARS (CLISPI.S.GAG))
  (PROP CLISPCOM * INITISOPRS)
  (IFPROP I.S.OPR * (PROGN INITISOPRS))
  [ADDVARS * (LIST (CONS 'I.S.OPRLST INITISOPRS)
    (CONS 'CLISPCOMFORWORDSPLST (SUBSET INITISOPRS 'U-CASEP))
  [VARS (CLISPDUMMYFORVARS '($STEM0 $STEM1 $STEM2 $STEM3 $STEM4 $STEM5 $STEM6)
  (ADDVARS * (LIST (CONS 'SYSLOCALVARS CLISPDUMMYFORVARS)
    (CONS 'INVISIBLEVARS CLISPDUMMYFORVARS)))
  (ADDVARS (SYSLOCALVARS $$VAL $STEM $$LST1 $$LST2 $$LST3 $$LST4 $$LST5 $$LST6 $$END $$EXTREME)
    (INVISIBLEVARS $$VAL $$END $STEM $$LST1 $$LST2 $$LST3 $$LST4 $$LST5 $$LST6 $$EXTREME))
  (FILEPKGCOMS I.S.OPRS)
  (FNS DUMPI.S.OPRS GETDEF.I.S.OPR))
(COMS (* forDuration)
  (ADDVARS (DURATIONCLISPCOM (TIMERUNITS timerUnits timerunits)
    (USINGBOX usingBox usingbox)
    (USINGTIMER usingTimer usingtimer)
    (FORDURATION forDuration forduration DURING during)
    (RESOURCEName resourceName resourcename)
    (UNTILDATE untilDate untildate)))
  (IFPROP (CLISPCOM \DURATIONTRAN)
    *
    (APPLY 'APPEND DURATIONCLISPCOM))
  (RESOURCES \ForDurationOfBox))
(COMS ;; Currently there are four possible entries for the INFO property: EVAL, BINDS, LABELS, PROG, or a list containing any or all of
  ;; these.
  ;; EVAL is used to indicate that an lambda evaluates its arguments. EVAL affects DWIMIFY and CLISPIFY: neither will touch an
  ;; lambda that does not have this property.
  ;; BINDS tells clispify and dwimify that CADR of the form is a list of variables being bound, a lambda prog.
  ;; PROG says that only the last top level expression is being used for value. This affects the way OR's and AND's are clispified, for
  ;; example.
  ;; Finally, LABELS indicates that top level atoms in this expression are not being evaluated. This tells clispify not to create atoms out
  ;; of lists at the top level. LABELS also implies that none of the top level expressions are being used for value.
  ;; For example, FOR has info property just BINDS, (EVAL is unnecessary since FOR is not a function and its dwimifying and clispifying
  ;; affected by its clispword property), whereas PROG has (BINDS EVAL LABELS), and LAMBDA has (EVAL BINDS PROG)
  (PROP INFO PROG PROG* RESETVARS RESETBUFS RESETLST ADV-PROG ADV-SETQ AND ARG COND ERSETQ NLSETQ OR
    PROG1 PROG2 PROG3 RESETFORM RESETSAVE RESETVAR RPAQ RPTQ FRPTQ SAVESETQ SETN SETQ UNDO NLSETQ
    XNLSETQ SETARG LET LET* RETURN))
(PROP FILETYPE CLISP)
(DECLARE%: DONTEVAL@LOAD DOEVAL@COMPILE DONTCOPY COMPILERVARS (ADDVARS (NLAMA DUMPI.S.OPRS)
  (NLAML)
  (LAMA]))

```

;; DWIM stuff

```

(RPAQ? NOFIXFNSLST0 )
(RPAQ? NOFIXVARSLST0 )
(RPAQ? NOSPELLFLG )
(RPAQ? LPARKEY 9)
(RPAQ? RPARKEY 0)
(RPAQ? WTFIXCHCONLST ' (NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL))
(RPAQ? WTFIXCHCONLST1 ' (NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL NIL))
(ADDTOVAR EDITMACROS
  (FIX9 (X N)
    (BIND (E (SETQ %1 (EDITFPAT 'X))
      T)
      (IF (NOT (ATOM (%##)))
        (1))
      (COMS (SPLIT89 RPARKEY N))
      (I F RPARKEY T)
      (E [SETQ %2 (ADD1 (LENGTH (CAR L)
        T)
        !0 MARK (LPQ [IF (OR (NULL %1)
          (NOT (EDIT4E %1 (%## 1]
            UP
            (E (SETQ %3 (LENGTH (CAR L)))
              T)
              (I RI 1 (MINUS %2)))

```

```

(E (SETQ %2 %3)
 T)
1 !0)
)
(DELETE NX))
(FIX9 NIL (FIX9))
(FIX8 (X N)
 (BIND (E (SETQ %1 (EDITFPAT 'X))
 T)
 (IF (LISTP (%##))
 (1)
 (COMS (SPLIT89 LPARKEY N))
 (I F LPARKEY T)
 (1)
 (LI 1)
 (IF (TAILP (CAR L)
 (CADR L))
 (!0)
 NIL)
 (LPQ [IF (OR (NULL %1)
 (NOT (EDIT4E %1 (%## 1]
 UP
 (RO 1)
 !0)))
 (FIX8 NIL (FIX8)))

```

(ADDTOVAR **DWIMUSERFORMS**)

(ADDTOVAR **LAMBDA SPLST** LAMBDA NLAMBDA)

(ADDTOVAR **OKREEVALST** AND OR PROG N SAVESETQ CAR CDR ADD1 SUB1 CONS LIST EQ EQUAL PRINT PRIN1 APPEND NEQ NOT NULL)

(ADDTOVAR **NOFIXFNSLST**)

(ADDTOVAR **NOFIXVARSLST**)

(ADDTOVAR **GLOBALVARS**)

(ADDTOVAR **LOCALVARS**)

(ADDTOVAR **SPECVARS**)

(ADDTOVAR **NLAMA**)

(ADDTOVAR **NLAML**)

(ADDTOVAR **LAMA**)

(ADDTOVAR **LAMS**)

(MOVD? 'NILL 'FREEVARS)

(PUTPROPS **BREAKDOWN FILEDEF** BRKDOWN)

(PUTPROPS **CALLS FILEDEF** MSANALYZE)

(PUTPROPS **CLISPRECORD FILEDEF** RECORD)

(PUTPROPS **SETUPHASHARRAY FILEDEF** (RECORD SETUPHASHARRAY))

(PUTPROPS **MAKEMATCH FILEDEF** MATCH)

(RPAQQ **DWIMIFYFLG** EVAL)

(RPAQQ **COMPILEUSERFN** COMPILEUSERFN)

(RPAQQ **CLISPTRANFLG** CLISP%)

(RPAQQ **DWIMESSGAG** NIL)

(RPAQ? **DWIMCHECK#ARGSFLG** T)

(RPAQ? **DWIMCHECKPROGLABELSFLG** T)

(RPAQ? **%#CLISPARRAY** 250)

(RPAQ? **RECORDHASHFLG** T)

(RPAQ? **CLISPRETRANFLG**)

(ADDTOVAR **DWIMEQUIVLST**)

(ADDTOVAR **EDITMACROS**

```

(DW NIL (BIND (E (PROGN (SETQ %1 (%##))
 (AND (CDR L)
 (%## !0 (E (SETQ %2 L)

```

```

T)))
(AND [SETQ %#3 (DWIMIFY %#1 T (OR %#2 ' (NIL)
EDITCHANGES
(RPLACA (CDR EDITCHANGES)
T)))
T)
(IF (NLISTP %#1)
((I %: %#3)
(IF (LISTP %#3)
(1)
NIL))
NIL)))
(PPT NIL (RESETVAR PRETTYTRANFLG T PP))
(!DW NIL (RESETVAR CLISPRETRANFLG T DW))
(NOCLISP NIL (NOCLISP TTY%:))
(NOCLISP COMS (RESETVAR CLISPTRANFLG NIL . COMS))
(CLISP%: NIL (BIND (E (COND ((SETQ %#1 (AND CLISPARRAY (GETHASH (%##)
CLISPARRAY)))
(SETQQ COM CLISP%:)
(EDITE %#1))
(T (PRIN1 "not translated.
" T)))
T))))

```

(ADDTOVAR EDITCOMSA PPT DW !DW CLISP%:)

(* * CLISP props)

(PUTPROPS %' CLISPTYPE 15)

(RPAQQ CLISPFLG T)

(RPAQQ CLISPCHARS (^ * / + - = _ %: %' ~ +- ~< > @ ! ← ↑))

(RPAQ? CLISPHELPLFLG T)

(RPAQ? TREATASCLISPFLG)

(RPAQ? CLISPINFIXSPLST)

(RPAQ? CLISPCHARRAY (MAKEBITTABLE CLISPCHARS))

(RPAQ? LEFT.ARROWS.BITTABLE (MAKEBITTABLE '(_ ←)))

(RPAQ? LEFT.ARROW ' _)

(RPAQ? CLISPISWORDSPLST)

(RPAQ? CLISPLASTSUB (CONS))

(RPAQ? CHECKCARATOMFLG)

(RPAQ? CLISPARITHOPLST '(+ - * / +- LT GT lt gt GEQ LEQ GE LE geq leq ge le))

(RPAQ? CLISPARITHCLASSLST '(INTEGER FIXED MIXED FLOATING))

(RPAQ? DWIMINMACROSFLG NIL)

(PUTPROPS ↑ CLISPTYPE 6)

(PUTPROPS ^ CLISPTYPE 6)

(PUTPROPS * CLISPTYPE 4)

(PUTPROPS / CLISPTYPE 4)

(PUTPROPS + CLISPTYPE 2)

(PUTPROPS - CLISPTYPE 7)

(PUTPROPS = CLISPTYPE -20)

(PUTPROPS _ CLISPTYPE (8 . -12))

(PUTPROPS ← CLISPTYPE (8 . -12))

(PUTPROPS %: CLISPTYPE (14 . 13))

(PUTPROPS %' CLISPTYPE 15)

(PUTPROPS ~ CLISPTYPE 7)

(PUTPROPS +- CLISPTYPE 2)

(PUTPROPS < CLISPTYPE BRACKET)

```

{MEDLEY}<sources>CLISP.;1

(PUTPROPS > CLISPTYPE BRACKET)
(PUTPROPS ↑ LISPFN EXPT)
(PUTPROPS ^ LISPFN EXPT)
(PUTPROPS * LISPFN TIMES)
(PUTPROPS / LISPFN QUOTIENT)
(PUTPROPS + LISPFN PLUS)
(PUTPROPS - LISPFN MINUS)
(PUTPROPS = LISPFN EQ)
(PUTPROPS _ LISPFN SETQ)
(PUTPROPS ← LISPFN SETQ)
(PUTPROPS %' LISPFN QUOTE)
(PUTPROPS ~ LISPFN NOT)
(PUTPROPS +- LISPFN DIFFERENCE)
(PUTPROPS - UNARYOP T)
(PUTPROPS %' UNARYOP T)
(PUTPROPS ~ UNARYOP T)
(PUTPROPS < UNARYOP T)
(PUTPROPS > UNARYOP T)
(PUTPROPS * CLISPCLASS *)
(PUTPROPS / CLISPCLASS /)
(PUTPROPS + CLISPCLASS +)
(PUTPROPS - CLISPCLASS -)
(PUTPROPS +- CLISPCLASS +-)
(PUTPROPS * CLISPCLASSDEF (ARITH ITIMES FTIMES TIMES))
(PUTPROPS / CLISPCLASSDEF (ARITH IQUOTIENT FQUOTIENT QUOTIENT))
(PUTPROPS + CLISPCLASSDEF (ARITH IPLUS FPLUS PLUS))
(PUTPROPS - CLISPCLASSDEF (ARITH IMINUS FMINUS MINUS))
(PUTPROPS +- CLISPCLASSDEF (ARITH IDIFFERENCE FDIFFERENCE DIFFERENCE))
(PUTPROPS = CLISPNEG ~=)
(PUTPROPS < CLISPBRACKET (< > SEPARATOR ! DWIMIFY CLISPANGLEBRACKETS CLISPIFY SHRIEKIFY))
(PUTPROPS > CLISPBRACKET (< > SEPARATOR ! DWIMIFY CLISPANGLEBRACKETS CLISPIFY SHRIEKIFY))
(RPAQQ DECLWORDS
(FLOATING FAST FFETCHFIELD FETCHFIELD REPLACEFIELD FREPLACEFIELD /REPLACEFIELD /LISTPUT /LISTPUT1 /MAPCON
 /MAPCONC /NCONC /NCONC1 /PUT /PUTASSOC /PUTHASH /PUTPROP /RPLACA /RPLACD /RPLNODE /RPLNODE2 /SETA
 ASSOC CLISPIFY FASSOC FIXED FLAST FMEMB FNTH FRPLACA FRPLACD FRPLNODE FRPLNODE2 INTEGER LAST
 LISTPUT LISTPUT1 MAPCON MAPCONC MEMB MIXED NCONC NCONC1 NTH PUT PUTASSOC PUTHASH PUTPROP RPLACA
 RPLACD RPLNODE RPLNODE2 SETA STANDARD UNDOABLE))
(PUTPROPS FMEMB CLISPTYPE -20)
(PUTPROPS MEMB CLISPTYPE -20)
(PUTPROPS FETCHFIELD LISPFN FETCHFIELD)
(PUTPROPS REPLACEFIELD LISPFN REPLACEFIELD)
(PUTPROPS FREPLACEFIELD LISPFN FREPLACEFIELD)
(PUTPROPS ASSOC LISPFN ASSOC)
(PUTPROPS LAST LISPFN LAST)
(PUTPROPS LISTPUT LISPFN LISTPUT)
(PUTPROPS LISTPUT1 LISPFN LISTPUT1)

```

(PUTPROPS **MAPCON** LISPFN MAPCON)
(PUTPROPS **MAPCONC** LISPFN MAPCONC)
(PUTPROPS **MEMB** LISPFN MEMB)
(PUTPROPS **NCONC** LISPFN NCONC)
(PUTPROPS **NCONC1** LISPFN NCONC1)
(PUTPROPS **NTH** LISPFN NTH)
(PUTPROPS **PUT** LISPFN PUT)
(PUTPROPS **PUTASSOC** LISPFN PUTASSOC)
(PUTPROPS **PUTHASH** LISPFN PUTHASH)
(PUTPROPS **PUTPROP** LISPFN PUTPROP)
(PUTPROPS **RPLACA** LISPFN RPLACA)
(PUTPROPS **RPLACD** LISPFN RPLACD)
(PUTPROPS **RPLNODE** LISPFN RPLNODE)
(PUTPROPS **RPLNODE2** LISPFN RPLNODE2)
(PUTPROPS **SETA** LISPFN SETA)
(PUTPROPS **FLOATING CLISPCLASS** (ARITH . 2))
(PUTPROPS **FAST CLISPCLASS** (ACCESS . 3))
(PUTPROPS **FFETCHFIELD CLISPCLASS** FETCFIELD)
(PUTPROPS **FETCHFIELD CLISPCLASS** FETCFIELD)
(PUTPROPS **REPLACEFIELD CLISPCLASS** REPLACEFIELD)
(PUTPROPS **FREPLACEFIELD CLISPCLASS** REPLACEFIELD)
(PUTPROPS **/REPLACEFIELD CLISPCLASS** REPLACEFIELD)
(PUTPROPS **/LISTPUT CLISPCLASS** LISTPUT)
(PUTPROPS **/MAPCON CLISPCLASS** MAPCON)
(PUTPROPS **/MAPCONC CLISPCLASS** MAPCONC)
(PUTPROPS **/NCONC CLISPCLASS** NCONC)
(PUTPROPS **/NCONC1 CLISPCLASS** NCONC1)
(PUTPROPS **/PUT CLISPCLASS** PUT)
(PUTPROPS **/PUTASSOC CLISPCLASS** PUTASSOC)
(PUTPROPS **/PUTHASH CLISPCLASS** PUTHASH)
(PUTPROPS **/PUTPROP CLISPCLASS** PUTPROP)
(PUTPROPS **/RPLACA CLISPCLASS** RPLACA)
(PUTPROPS **/RPLACD CLISPCLASS** RPLACD)
(PUTPROPS **/RPLNODE CLISPCLASS** RPLNODE)
(PUTPROPS **/RPLNODE2 CLISPCLASS** RPLNODE2)
(PUTPROPS **/SETA CLISPCLASS** SETA)
(PUTPROPS **ASSOC CLISPCLASS** ASSOC)
(PUTPROPS **FASSOC CLISPCLASS** ASSOC)
(PUTPROPS **FIXED CLISPCLASS** (ARITH . 1))
(PUTPROPS **FLAST CLISPCLASS** LAST)
(PUTPROPS **FMEMB CLISPCLASS** MEMB)
(PUTPROPS **FNTH CLISPCLASS** NTH)
(PUTPROPS **FRPLACA CLISPCLASS** RPLACA)
(PUTPROPS **FRPLACD CLISPCLASS** RPLACD)

(PUTPROPS **FRPLNODE CLISPCLASS** RPLNODE)
(PUTPROPS **FRPLNODE2 CLISPCLASS** RPLNODE2)
(PUTPROPS **INTEGER CLISPCLASS** (ARITH . 1))
(PUTPROPS **LAST CLISPCLASS** LAST)
(PUTPROPS **LISTPUT CLISPCLASS** LISTPUT)
(PUTPROPS **LISTPUT1 CLISPCLASS** LISTPUT1)
(PUTPROPS **MAPCON CLISPCLASS** MAPCON)
(PUTPROPS **MAPCONC CLISPCLASS** MAPCONC)
(PUTPROPS **MEMB CLISPCLASS** MEMB)
(PUTPROPS **MIXED CLISPCLASS** (ARITH . 3))
(PUTPROPS **NCNC CLISPCLASS** NCNC)
(PUTPROPS **NCNC1 CLISPCLASS** NCNC1)
(PUTPROPS **NTH CLISPCLASS** NTH)
(PUTPROPS **PUT CLISPCLASS** PUT)
(PUTPROPS **PUTASSOC CLISPCLASS** PUTASSOC)
(PUTPROPS **PUTHASH CLISPCLASS** PUTHASH)
(PUTPROPS **PUTPROP CLISPCLASS** PUTPROP)
(PUTPROPS **RPLACA CLISPCLASS** RPLACA)
(PUTPROPS **RPLACD CLISPCLASS** RPLACD)
(PUTPROPS **RPLNODE CLISPCLASS** RPLNODE)
(PUTPROPS **RPLNODE2 CLISPCLASS** RPLNODE2)
(PUTPROPS **SETA CLISPCLASS** SETA)
(PUTPROPS **STANDARD CLISPCLASS** (ACCESS . 1))
(PUTPROPS **UNDOABLE CLISPCLASS** (ACCESS . 2))
(PUTPROPS **FETCHFIELD CLISPCLASSDEF** (ACCESS FETCHFIELD NIL FFETCHFIELD))
(PUTPROPS **REPLACEFIELD CLISPCLASSDEF** (ACCESS REPLACEFIELD /REPLACEFIELD FREPLACEFIELD))
(PUTPROPS **ASSOC CLISPCLASSDEF** (ACCESS ASSOC NIL FASSOC))
(PUTPROPS **LAST CLISPCLASSDEF** (ACCESS LAST NIL FLAST))
(PUTPROPS **LISTPUT CLISPCLASSDEF** (ACCESS LISTPUT /LISTPUT))
(PUTPROPS **LISTPUT1 CLISPCLASSDEF** (ACCESS LISTPUT1 /LISTPUT1))
(PUTPROPS **MAPCON CLISPCLASSDEF** (ACCESS MAPCON /MAPCON))
(PUTPROPS **MAPCONC CLISPCLASSDEF** (ACCESS MAPCONC /MAPCONC))
(PUTPROPS **MEMB CLISPCLASSDEF** (ACCESS MEMB NIL FMEMB))
(PUTPROPS **NCNC CLISPCLASSDEF** (ACCESS NCNC /NCNC))
(PUTPROPS **NCNC1 CLISPCLASSDEF** (ACCESS NCNC1 /NCNC1))
(PUTPROPS **NTH CLISPCLASSDEF** (ACCESS NTH NIL FNTH))
(PUTPROPS **PUT CLISPCLASSDEF** (ACCESS PUT /PUT))
(PUTPROPS **PUTASSOC CLISPCLASSDEF** (ACCESS PUTASSOC /PUTASSOC))
(PUTPROPS **PUTHASH CLISPCLASSDEF** (ACCESS PUTHASH /PUTHASH))
(PUTPROPS **PUTPROP CLISPCLASSDEF** (ACCESS PUTPROP /PUTPROP))
(PUTPROPS **RPLACA CLISPCLASSDEF** (ACCESS RPLACA /RPLACA FRPLACA))
(PUTPROPS **RPLACD CLISPCLASSDEF** (ACCESS RPLACD /RPLACD FRPLACD))
(PUTPROPS **RPLNODE CLISPCLASSDEF** (ACCESS RPLNODE /RPLNODE FRPLNODE))

(PUTPROPS **RPLNODE2 CLISPCLASSDEF** (ACCESS RPLNODE2 /RPLNODE2 FRPLNODE2))

(PUTPROPS **SETA CLISPCLASSDEF** (ACCESS SETA /SETA))

(PUTPROPS **FMEMB CLISPNEG** ~FMEMB)

(PUTPROPS **MEMB CLISPNEG** ~MEMB)

(PUTPROPS **FMEMB BROADSCOPE** T)

(PUTPROPS **MEMB BROADSCOPE** T)

(PUTPROPS **LT CLISPTYPE** -20)

(PUTPROPS **lt CLISPTYPE** -20)

(PUTPROPS **GT CLISPTYPE** -20)

(PUTPROPS **gt CLISPTYPE** -20)

(PUTPROPS **LE CLISPTYPE** -20)

(PUTPROPS **le CLISPTYPE** -20)

(PUTPROPS **GE CLISPTYPE** -20)

(PUTPROPS **ge CLISPTYPE** -20)

(PUTPROPS **LEQ CLISPTYPE** -20)

(PUTPROPS **leq CLISPTYPE** -20)

(PUTPROPS **GEQ CLISPTYPE** -20)

(PUTPROPS **geq CLISPTYPE** -20)

(PUTPROPS **EQ CLISPTYPE** -20)

(PUTPROPS **NEQ CLISPTYPE** -20)

(PUTPROPS **EQP CLISPTYPE** -20)

(PUTPROPS **EQUAL CLISPTYPE** -20)

(PUTPROPS **EQUALS CLISPTYPE** -20)

(PUTPROPS **AND CLISPTYPE** -25)

(PUTPROPS **OR CLISPTYPE** -26)

(PUTPROPS **and CLISPTYPE** -25)

(PUTPROPS **or CLISPTYPE** -26)

(PUTPROPS **NOR CLISPTYPE** -25)

(PUTPROPS **nor CLISPTYPE** -25)

(PUTPROPS **MEMBER CLISPTYPE** -20)

(PUTPROPS **ILESSP CLISPTYPE** -20)

(PUTPROPS **IGREATERP CLISPTYPE** -20)

(PUTPROPS **FGTP CLISPTYPE** -20)

(PUTPROPS **MINUS CLISPTYPE** 8)

(PUTPROPS **LESSP CLISPTYPE** -20)

(PUTPROPS **GREATERP CLISPTYPE** -20)

(PUTPROPS **-> CLISPTYPE** 7)

(PUTPROPS **=> CLISPTYPE** 7)

(PUTPROPS **LT LISPFN** LESSP)

(PUTPROPS **lt LISPFN** LESSP)

(PUTPROPS **GT LISPFN** GREATERP)

(PUTPROPS **gt LISPFN** GREATERP)

(PUTPROPS **LE LISPFN** LEQ)

(PUTPROPS **le LISPFN** LEQ)

(PUTPROPS **GE** LISPFN GEQ)
(PUTPROPS **ge** LISPFN GEQ)
(PUTPROPS **LEQ** LISPFN LEQ)
(PUTPROPS **leq** LISPFN LEQ)
(PUTPROPS **GEQ** LISPFN GEQ)
(PUTPROPS **geq** LISPFN GEQ)
(PUTPROPS **EQUALS** LISPFN EQUAL)
(PUTPROPS **AND** LISPFN AND)
(PUTPROPS **OR** LISPFN OR)
(PUTPROPS **and** LISPFN AND)
(PUTPROPS **or** LISPFN OR)
(PUTPROPS **NOR** LISPFN AND)
(PUTPROPS **nor** LISPFN AND)
(PUTPROPS **NOT UNARYOP** T)
(PUTPROPS **MINUS UNARYOP** T)
(PUTPROPS **LEQ CLISPINFIX** le)
(PUTPROPS **GEQ CLISPINFIX** ge)
(PUTPROPS **EQ CLISPINFIX** =)
(PUTPROPS **NOT CLISPINFIX** ~)
(PUTPROPS **AND CLISPINFIX** and)
(PUTPROPS **OR CLISPINFIX** or)
(PUTPROPS **SETQ CLISPINFIX** _)
(PUTPROPS **IPLUS CLISPINFIX** +)
(PUTPROPS **IMINUS CLISPINFIX** -)
(PUTPROPS **IDIFFERENCE CLISPINFIX** +-)
(PUTPROPS **ITIMES CLISPINFIX** *)
(PUTPROPS **IQUOTIENT CLISPINFIX** /)
(PUTPROPS **ILESSP CLISPINFIX** lt)
(PUTPROPS **IGREATERP CLISPINFIX** gt)
(PUTPROPS **PLUS CLISPINFIX** +)
(PUTPROPS **MINUS CLISPINFIX** -)
(PUTPROPS **DIFFERENCE CLISPINFIX** +-)
(PUTPROPS **TIMES CLISPINFIX** *)
(PUTPROPS **QUOTIENT CLISPINFIX** /)
(PUTPROPS **LESSP CLISPINFIX** lt)
(PUTPROPS **GREATERP CLISPINFIX** gt)
(PUTPROPS **EXPT CLISPINFIX** ^)
(PUTPROPS **LT CLISPCLASS** LT)
(PUTPROPS **lt CLISPCLASS** LT)
(PUTPROPS **GT CLISPCLASS** GT)
(PUTPROPS **gt CLISPCLASS** GT)
(PUTPROPS **LE CLISPCLASS** LEQ)
(PUTPROPS **le CLISPCLASS** LEQ)

```

{MEDLEY}<sources>CLISP.;1

(PUTPROPS GE CLISPCLASS GEQ)
(PUTPROPS ge CLISPCLASS GEQ)
(PUTPROPS LEQ CLISPCLASS LEQ)
(PUTPROPS leq CLISPCLASS LEQ)
(PUTPROPS GEQ CLISPCLASS GEQ)
(PUTPROPS geq CLISPCLASS GEQ)
(PUTPROPS IPLUS CLISPCLASS +)
(PUTPROPS IMINUS CLISPCLASS -)
(PUTPROPS IDIFFERENCE CLISPCLASS +-)
(PUTPROPS ITIMES CLISPCLASS *)
(PUTPROPS IQUOTIENT CLISPCLASS /)
(PUTPROPS ILESSP CLISPCLASS LT)
(PUTPROPS IGREATERP CLISPCLASS GT)
(PUTPROPS FPLUS CLISPCLASS +)
(PUTPROPS FMINUS CLISPCLASS -)
(PUTPROPS FDIFFERENCE CLISPCLASS +-)
(PUTPROPS FTIMES CLISPCLASS *)
(PUTPROPS FQUOTIENT CLISPCLASS /)
(PUTPROPS FGTP CLISPCLASS GT)
(PUTPROPS PLUS CLISPCLASS +)
(PUTPROPS MINUS CLISPCLASS -)
(PUTPROPS DIFFERENCE CLISPCLASS +-)
(PUTPROPS TIMES CLISPCLASS *)
(PUTPROPS QUOTIENT CLISPCLASS /)
(PUTPROPS LESSP CLISPCLASS LT)
(PUTPROPS GREATERP CLISPCLASS GT)
(PUTPROPS LT CLISPCLASSDEF (ARITH ILESSP LESSP LESSP))
(PUTPROPS GT CLISPCLASSDEF (ARITH IGREATERP FGTP GREATERP))
(PUTPROPS LE CLISPCLASSDEF (ARITH ILEQ LEQ LEQ))
(PUTPROPS GE CLISPCLASSDEF (ARITH IGEQ GEQ GEQ))
(PUTPROPS LEQ CLISPCLASSDEF (ARITH ILEQ LEQ LEQ))
(PUTPROPS GEQ CLISPCLASSDEF (ARITH IGEQ GEQ GEQ))
(PUTPROPS LT CLISPNEG GEQ)
(PUTPROPS GT CLISPNEG LEQ)
(PUTPROPS EQUALS CLISPNEG ~EQUAL)
(PUTPROPS MEMBER CLISPNEG ~MEMBER)
(PUTPROPS LT BROADSCOPE T)
(PUTPROPS lt BROADSCOPE T)
(PUTPROPS GT BROADSCOPE T)
(PUTPROPS gt BROADSCOPE T)
(PUTPROPS LE BROADSCOPE T)
(PUTPROPS le BROADSCOPE T)
(PUTPROPS GE BROADSCOPE T)
(PUTPROPS ge BROADSCOPE T)

```

(PUTPROPS **LEQ BROADSCOPE** T)
(PUTPROPS **leq BROADSCOPE** T)
(PUTPROPS **GEQ BROADSCOPE** T)
(PUTPROPS **geq BROADSCOPE** T)
(PUTPROPS **EQ BROADSCOPE** T)
(PUTPROPS **NEQ BROADSCOPE** T)
(PUTPROPS **EQP BROADSCOPE** T)
(PUTPROPS **EQUAL BROADSCOPE** T)
(PUTPROPS **EQUALS BROADSCOPE** T)
(PUTPROPS **NOT BROADSCOPE** T)
(PUTPROPS **AND BROADSCOPE** T)
(PUTPROPS **OR BROADSCOPE** T)
(PUTPROPS **and BROADSCOPE** T)
(PUTPROPS **or BROADSCOPE** T)
(PUTPROPS **NOR BROADSCOPE** T)
(PUTPROPS **nor BROADSCOPE** T)
(PUTPROPS **MEMBER BROADSCOPE** T)
(PUTPROPS **ILESSP BROADSCOPE** T)
(PUTPROPS **IGREATERP BROADSCOPE** T)
(PUTPROPS **FGTP BROADSCOPE** T)
(PUTPROPS **LESSP BROADSCOPE** T)
(PUTPROPS **GREATERP BROADSCOPE** T)
(PUTPROPS **ELT SETFN** SETA)
(PUTPROPS **SETA SETFN** (ELT))

(DEFOPTIMIZER **CLISP%** (X &REST Y)
X)

(PUTPROPS **AND CLISPWORD** T)
(PUTPROPS **OR CLISPWORD** T)
(PUTPROPS **and CLISPWORD** T)
(PUTPROPS **or CLISPWORD** T)
(PUTPROPS **! CLISPWORD** T)
(PUTPROPS **!! CLISPWORD** T)
(PUTPROPS **CLISP CLISPWORD** (PREFIXFN . clisp))
(PUTPROPS **clisp CLISPWORD** (PREFIXFN . clisp))
(PUTPROPS **MATCH CLISPWORD** (MATCHWORD . match))
(PUTPROPS **match CLISPWORD** (MATCHWORD . match))

(* * IF)

(RPAQQ **CLISPIFWORDSPLST** (THEN ELSE ELSEIF IF))
(RPAQ? **CLISPIFTRANFLG** T)
(PUTPROPS **IF CLISPWORD** (IFWORD . if))
(PUTPROPS **THEN CLISPWORD** (IFWORD . then))
(PUTPROPS **ELSE CLISPWORD** (IFWORD . else))
(PUTPROPS **ELSEIF CLISPWORD** (IFWORD . elseif))

```

(PUTPROPS if CLISPCWORD (IFWORD . if))
(PUTPROPS then CLISPCWORD (IFWORD . then))
(PUTPROPS else CLISPCWORD (IFWORD . else))
(PUTPROPS elseif CLISPCWORD (IFWORD . elseif))

      (* I.S.OPR)

(RPAQQ CLISPI.S.GAG NIL)
(RPAQQ INITISOPRS
  (ALWAYS AS BIND BY COLLECT COUNT DECLARE DECLARE%: DO EACHTIME FCOLLECT FINALLY FIND FIRST FOR FROM IN
   INSIDE ISTHERE JOIN LARGEST NEVER OLD ON ORIGINAL REPEATUNTIL REPEATWHILE SMALLEST SUCHTHAT SUM
   THEREIS THRU TO UNLESS UNTIL WHEN WHERE WHILE always as bind by collect count declare declare%: do
    eachtime fcollect finally find first for from in inside isthere join largest never old on original
    repeatuntil repeatwhile smallest suchthat sum thereis thru to unless until when where while))

(PUTPROPS ALWAYS CLISPCWORD (FORWORD . always))
(PUTPROPS AS CLISPCWORD (FORWORD . as))
(PUTPROPS BIND CLISPCWORD (FORWORD . bind))
(PUTPROPS BY CLISPCWORD (FORWORD . by))
(PUTPROPS COLLECT CLISPCWORD (FORWORD . collect))
(PUTPROPS COUNT CLISPCWORD (FORWORD . count))
(PUTPROPS DECLARE CLISPCWORD (FORWORD . declare))
(PUTPROPS DECLARE%: CLISPCWORD (FORWORD declare%: DECLARE))
(PUTPROPS DO CLISPCWORD (FORWORD . do))
(PUTPROPS EACHTIME CLISPCWORD (FORWORD . eachtime))
(PUTPROPS FCOLLECT CLISPCWORD (FORWORD . fcollect))
(PUTPROPS FINALLY CLISPCWORD (FORWORD . finally))
(PUTPROPS FIND CLISPCWORD (FORWORD find FOR))
(PUTPROPS FIRST CLISPCWORD (FORWORD . first))
(PUTPROPS FOR CLISPCWORD (FORWORD . for))
(PUTPROPS FROM CLISPCWORD (FORWORD . from))
(PUTPROPS IN CLISPCWORD (FORWORD . in))
(PUTPROPS INSIDE CLISPCWORD (FORWORD . inside))
(PUTPROPS ISTHERE CLISPCWORD (FORWORD isthere THEREIS))
(PUTPROPS JOIN CLISPCWORD (FORWORD . join))
(PUTPROPS LARGEST CLISPCWORD (FORWORD . largest))
(PUTPROPS NEVER CLISPCWORD (FORWORD . never))
(PUTPROPS OLD CLISPCWORD (FORWORD . old))
(PUTPROPS ON CLISPCWORD (FORWORD . on))
(PUTPROPS ORIGINAL CLISPCWORD (FORWORD . original))
(PUTPROPS REPEATUNTIL CLISPCWORD (FORWORD . repeatuntil))
(PUTPROPS REPEATWHILE CLISPCWORD (FORWORD . repeatwhile))
(PUTPROPS SMALLEST CLISPCWORD (FORWORD . smallest))
(PUTPROPS SUCHTHAT CLISPCWORD (FORWORD suchthat THEREIS))
(PUTPROPS SUM CLISPCWORD (FORWORD . sum))
(PUTPROPS THEREIS CLISPCWORD (FORWORD . thereis))
(PUTPROPS THRU CLISPCWORD (FORWORD thru TO))
(PUTPROPS TO CLISPCWORD (FORWORD . to))

```

(PUTPROPS **UNLESS CLISPPWORD** (FORWORD . unless))
(PUTPROPS **UNTIL CLISPPWORD** (FORWORD . until))
(PUTPROPS **WHEN CLISPPWORD** (FORWORD . when))
(PUTPROPS **WHERE CLISPPWORD** (FORWORD where WHEN))
(PUTPROPS **WHILE CLISPPWORD** (FORWORD . while))
(PUTPROPS **always CLISPPWORD** (FORWORD . always))
(PUTPROPS **as CLISPPWORD** (FORWORD . as))
(PUTPROPS **bind CLISPPWORD** (FORWORD . bind))
(PUTPROPS **by CLISPPWORD** (FORWORD . by))
(PUTPROPS **collect CLISPPWORD** (FORWORD . collect))
(PUTPROPS **count CLISPPWORD** (FORWORD . count))
(PUTPROPS **declare CLISPPWORD** (FORWORD . declare))
(PUTPROPS **declare%: CLISPPWORD** (FORWORD declare%: DECLARE))
(PUTPROPS **do CLISPPWORD** (FORWORD . do))
(PUTPROPS **eachtime CLISPPWORD** (FORWORD . eachtime))
(PUTPROPS **fcollect CLISPPWORD** (FORWORD . fcollect))
(PUTPROPS **finally CLISPPWORD** (FORWORD . finally))
(PUTPROPS **find CLISPPWORD** (FORWORD find FOR))
(PUTPROPS **first CLISPPWORD** (FORWORD . first))
(PUTPROPS **for CLISPPWORD** (FORWORD . for))
(PUTPROPS **from CLISPPWORD** (FORWORD . from))
(PUTPROPS **in CLISPPWORD** (FORWORD . in))
(PUTPROPS **inside CLISPPWORD** (FORWORD . inside))
(PUTPROPS **isthere CLISPPWORD** (FORWORD isthere thereis))
(PUTPROPS **join CLISPPWORD** (FORWORD . join))
(PUTPROPS **largest CLISPPWORD** (FORWORD . largest))
(PUTPROPS **never CLISPPWORD** (FORWORD . never))
(PUTPROPS **old CLISPPWORD** (FORWORD . old))
(PUTPROPS **on CLISPPWORD** (FORWORD . on))
(PUTPROPS **original CLISPPWORD** (FORWORD . original))
(PUTPROPS **repeatuntil CLISPPWORD** (FORWORD . repeatuntil))
(PUTPROPS **repeatwhile CLISPPWORD** (FORWORD . repeatwhile))
(PUTPROPS **smallest CLISPPWORD** (FORWORD . smallest))
(PUTPROPS **suchthat CLISPPWORD** (FORWORD suchthat THEREIS))
(PUTPROPS **sum CLISPPWORD** (FORWORD . sum))
(PUTPROPS **thereis CLISPPWORD** (FORWORD . thereis))
(PUTPROPS **thru CLISPPWORD** (FORWORD thru TO))
(PUTPROPS **to CLISPPWORD** (FORWORD . to))
(PUTPROPS **unless CLISPPWORD** (FORWORD . unless))
(PUTPROPS **until CLISPPWORD** (FORWORD . until))
(PUTPROPS **when CLISPPWORD** (FORWORD . when))
(PUTPROPS **where CLISPPWORD** (FORWORD where WHEN))
(PUTPROPS **while CLISPPWORD** (FORWORD . while))
(PUTPROPS **always I.S.OPR** ((COND ((NULL BODY)

```

                (SETQ $$VAL NIL)
                (GO $$OUT))
    BIND
    (SETQ $$VAL T))

(PUTPROPS collect I.S.OPR ((SETQ $$VAL (NCONC1 $$VAL BODY)))

(PUTPROPS count I.S.OPR ((AND BODY (SETQ $$VAL (ADD1 $$VAL)))
    BIND
    ($$VAL _ 0)))

(PUTPROPS do I.S.OPR (BODY))

(PUTPROPS fcollect I.S.OPR [(= SUBPAIR '(VAR1 VAR2)
    (LIST (GETDUMMYVAR T)
          (GETDUMMYVAR T))
    '(PROGN (SETQ VAR1 BODY)
            (COND [VAR2 (FRPLACD VAR2 (SETQ VAR2 (LIST VAR1)
                (T (SETQ $$VAL (SETQ VAR2 (LIST VAR1]))

(PUTPROPS inside I.S.OPR [NIL = SUBST (GETDUMMYVAR)
    'VAR
    '(bind (VAR _ BODY)
           eachtime
           (COND ((NULL VAR)
                  (GO $$OUT))
                 ((NLISTP VAR)
                  (SETQ I.V. VAR)
                  (SETQ VAR NIL))
                 (T (SETQ I.V. (CAR VAR))
                    (SETQ VAR (CDR VAR]))

(PUTPROPS join I.S.OPR ((SETQ $$VAL (NCONC $$VAL BODY)))

(PUTPROPS largest I.S.OPR [NIL = SUBST (GETDUMMYVAR)
    '$$TEMP
    '(BIND $$EXTREME $$TEMP DO (SETQ $$TEMP BODY)
      (COND ((OR (NULL $$EXTREME)
                 (GREATERP $$TEMP $$EXTREME))
             (SETQ $$EXTREME $$TEMP)
             (SETQ $$VAL I.V.])

(PUTPROPS never I.S.OPR ((COND (BODY (SETQ $$VAL NIL)
    (GO $$OUT)))
    BIND
    ($$VAL _ T))

(PUTPROPS old I.S.OPR MODIFIER)

(PUTPROPS smallest I.S.OPR [NIL = SUBST (GETDUMMYVAR)
    '$$TEMP
    '(BIND $$EXTREME $$TEMP DO (SETQ $$TEMP BODY)
      (COND ((OR (NULL $$EXTREME)
                 (LESSP $$TEMP $$EXTREME))
             (SETQ $$EXTREME $$TEMP)
             (SETQ $$VAL I.V.])

(PUTPROPS sum I.S.OPR ((SETQ $$VAL (PLUS $$VAL BODY))
    BIND
    ($$VAL _ 0))

(PUTPROPS thereis I.S.OPR [(COND (BODY (SETQ $$VAL (OR I.V. T))
    (GO $$OUT))

(ADDTOVAR I.S.OPRLST ALWAYS AS BIND BY COLLECT COUNT DECLARE DECLARE%: DO EACHTIME FCOLLECT FINALLY FIND FIRST
FOR FROM IN INSIDE ISTHERE JOIN LARGEST NEVER OLD ON ORIGINAL REPEATUNTIL REPEATWHILE
SMALLEST SUCHTHAT SUM THEREIS THRU TO UNLESS UNTIL WHEN WHERE WHILE always as bind by
collect count declare declare%: do eachtime fcollect finally find first for from in
inside isthere join largest never old on original repeatuntil repeatwhile smallest
suchthat sum thereis thru to unless until when where while)

(ADDTOVAR CLISPFORWORDSPLST ALWAYS AS BIND BY COLLECT COUNT DECLARE DECLARE%: DO EACHTIME FCOLLECT FINALLY
FIND FIRST FOR FROM IN INSIDE ISTHERE JOIN LARGEST NEVER OLD ON ORIGINAL
REPEATUNTIL REPEATWHILE SMALLEST SUCHTHAT SUM THEREIS THRU TO UNLESS UNTIL
WHEN WHERE WHILE)

(RPAQQ CLISPDUMMYFORVARS ($$STEM0 $$STEM1 $$STEM2 $$STEM3 $$STEM4 $$STEM5 $$STEM6))

(ADDTOVAR SYSLOCALVARS $$STEM0 $$STEM1 $$STEM2 $$STEM3 $$STEM4 $$STEM5 $$STEM6)

(ADDTOVAR INVISIBLEVARS $$STEM0 $$STEM1 $$STEM2 $$STEM3 $$STEM4 $$STEM5 $$STEM6)

(ADDTOVAR SYSLOCALVARS $$VAL $$STEM $$LST1 $$LST2 $$LST3 $$LST4 $$LST5 $$LST6 $$END $$EXTREME)

(ADDTOVAR INVISIBLEVARS $$VAL $$END $$STEM $$LST1 $$LST2 $$LST3 $$LST4 $$LST5 $$LST6 $$EXTREME)

[PUTDEF 'I.S.OPRS 'FILEPKGCOMS '((COM MACRO [X (DECLARE%: EVAL@COMPILE (P * (DUMPI.S.OPRS . X]

```

CONTENTS NIL)
(TYPE DESCRIPTION "i.s. operators" GETDEF GETDEF.I.S.OPR WHENCHANGED (
CLEARCLISPARRAY
])

(DEFINEQ

(DUMPI.S.OPRS

[NLAMBDA X

(* Imm "14-Aug-84 18:34")
(* Dump I.S.OPRS definitions. -
redefined to dump out same case as given)

(for Y in X collect (OR (GETDEF.I.S.OPR Y)
(PROG1 NIL
(LISPXPRT (LIST 'I.S.OPR Y 'not 'defined)
T T)))]

(GETDEF.I.S.OPR

[LAMBDA (Y

(* Imm "14-Aug-84 18:34")

(PROG (TEM BODY EVALFLG)
(RETURN (CONS 'I.S.OPR
(CONS (KWOTE Y)
(OR [AND [SETQ TEM (LISTP (GETPROP Y 'CLISPPWORD)
(EQ (CAR TEM)
'FORWORD)
(COND
[[AND (NLISTP (CDR TEM))
(SETQ BODY (GETPROP (CDR TEM)
'I.S.OPR]
(COND
[(LISTP BODY)
(CONS [KWOTE (COND
((EQ (CAR (LISTP (CAR BODY)))
'=)
(SETQ EVALFLG T)
(CDR BODY))
(T (CAR BODY)
(COND
((EQ (CADR BODY)
'=)
(LIST (KWOTE (CDDR BODY)
T))
[(CDR BODY)
(COND
(EVALFLG (SHOULDNT)))

(* somehow there was an = in front of the i.s.type and not in front of the others.
this shouldnt happen)

(LIST (KWOTE (CDR BODY)
(EVALFLG '(NIL T)
(T (LIST (KWOTE BODY)
((AND (LISTP (CDR TEM))
(CADDR TEM))
(LIST (KWOTE (CADDR TEM)
(RETURN)]

)

(* * forDuration)

(ADDTOVAR DURATIONCLISPPWORDS (TIMERUNITS timerUnits timerunits)
(USINGBOX usingBox usingbox)
(USINGTIMER usingTimer usingtimer)
(FORDURATION forDuration forduration DURING during)
(RESOURCEName resourceName resourcename)
(UNTILDATE untilDate untildate))

(PUTPROPS TIMERUNITS CLISPPWORD (FORWORD . timerUnits))

(PUTPROPS timerUnits CLISPPWORD (FORWORD . timerUnits))

(PUTPROPS timerunits CLISPPWORD (FORWORD . timerUnits))

(PUTPROPS USINGBOX CLISPPWORD (FORWORD . usingBox))

(PUTPROPS usingBox CLISPPWORD (FORWORD . usingBox))

(PUTPROPS usingbox CLISPPWORD (FORWORD . usingBox))

(PUTPROPS USINGTIMER CLISPPWORD (FORWORD . usingTimer))

(PUTPROPS usingTimer CLISPPWORD (FORWORD . usingTimer))

(PUTPROPS usingtimer CLISPPWORD (FORWORD . usingTimer))

```
(PUTPROPS FORDURATION CLISPCWORD (FORWORD . forDuration))
(PUTPROPS forDuration CLISPCWORD (FORWORD . forDuration))
(PUTPROPS forduration CLISPCWORD (FORWORD . forDuration))
(PUTPROPS DURING CLISPCWORD (FORWORD . during))
(PUTPROPS during CLISPCWORD (FORWORD . during))
(PUTPROPS RESOURCEName CLISPCWORD (FORWORD . resourceName))
(PUTPROPS resourceName CLISPCWORD (FORWORD . resourceName))
(PUTPROPS resourceName CLISPCWORD (FORWORD . resourceName))
(PUTPROPS UNTILDATE CLISPCWORD (FORWORD . untildate))
(PUTPROPS untilDate CLISPCWORD (FORWORD . untilDate))
(PUTPROPS untildate CLISPCWORD (FORWORD . untildate))
(PUTPROPS timerUnits \DURATIONTRAN T)
(PUTPROPS usingBox \DURATIONTRAN T)
(PUTPROPS usingTimer \DURATIONTRAN T)
(PUTPROPS forDuration \DURATIONTRAN T)
(PUTPROPS during \DURATIONTRAN T)
(PUTPROPS resourceName \DURATIONTRAN T)
(PUTPROPS untilDate \DURATIONTRAN T)
(PUTPROPS untildate \DURATIONTRAN T)
(DECLARE%: EVAL@COMPILE
[PUTDEF '\ForDurationOfBox 'RESOURCES '(NEW (\TIMER.MAKETIMER)
)
```

;; Currently there are four possible entries for the INFO property: EVAL, BINDS, LABELS, PROG, or a list containing any or all of these.
 ;; EVAL is used to indicate that an nlambda evaluates its arguments. EVAL affects DWIMIFY and CLISPIFY: neither will touch an nlambda that does not
 ;; have this property.
 ;; BINDS tells clispify and dwimify that CADR of the form is a list of variables being bound, a la prog.
 ;; PROG says that only the last top level expression is being used for value. This affects the way OR's and AND's are clispified, for example.
 ;; Finally, LABELS indicates that top level atoms in this expression are not being evaluated. This tells clispify not to create atoms out of lists at the top
 ;; level. LABELS also implies that none of the top level expressions are being used for value.
 ;; For example, FOR has info property just BINDS, (EVAL is unnecessary since FOR is not a function and its dwimifying and clispifying affected by its
 ;; clispword property), whereas PROG has (BINDS EVAL LABELS), and LAMBDA has (EVAL BINDS PROG)

```
(PUTPROPS PROG INFO (EVAL BINDS LABELS))
(PUTPROPS PROG* INFO (EVAL BINDS LABELS))
(PUTPROPS RESETVARS INFO (EVAL BINDS LABELS))
(PUTPROPS RESETBUFS INFO EVAL)
(PUTPROPS RESETLST INFO (EVAL PROG))
(PUTPROPS ADV-PROG INFO (EVAL BINDS LABELS))
(PUTPROPS ADV-SETQ INFO EVAL)
(PUTPROPS AND INFO EVAL)
(PUTPROPS ARG INFO EVAL)
(PUTPROPS COND INFO EVAL)
(PUTPROPS ERSETQ INFO EVAL)
(PUTPROPS NLSETQ INFO EVAL)
(PUTPROPS OR INFO EVAL)
(PUTPROPS PROG1 INFO EVAL)
(PUTPROPS PROG2 INFO EVAL)
(PUTPROPS PROGN INFO (EVAL PROG))
```



```
(PUTPROPS RESETFORM INFO EVAL)
(PUTPROPS RESETSAVE INFO EVAL)
(PUTPROPS RESETVAR INFO EVAL)
(PUTPROPS RPAQ INFO EVAL)
(PUTPROPS RPTQ INFO EVAL)
(PUTPROPS FRPTQ INFO EVAL)
(PUTPROPS SAVESETQ INFO EVAL)
(PUTPROPS SETN INFO EVAL)
(PUTPROPS SETQ INFO EVAL)
(PUTPROPS UNDONLSETQ INFO EVAL)
(PUTPROPS XNLSETQ INFO EVAL)
(PUTPROPS SETARG INFO EVAL)
(PUTPROPS LET INFO (BINDS EVAL))
(PUTPROPS LET* INFO (BINDS EVAL))
(PUTPROPS RETURN INFO EVAL)
(PUTPROPS CLISP FILETYPE CL:COMPILE-FILE)
(DECLARE%: DONTEVAL@LOAD DOEVAL@COMPILE DONTCOPY COMPILERVERS
(ADDTOVAR NLAMA DUMPI.S.OPRS)
(ADDTOVAR NLAML )
(ADDTOVAR LAMA )
)
(PUTPROPS CLISP COPYRIGHT ("Venue & Xerox Corporation" T 1982 1983 1984 1985 1986 1990))
```

FUNCTION INDEX

DUMPI.S.OPRS15 GETDEF.I.S.OPR15

PROPERTY INDEX

!11 DECLARE12 FRPLACD6 NCONC6,7 SETN17
!!11 declare13 FRPLNODE7 NCONC16,7 SETQ9,17
%'4,5 DECLARE%:12 FRPLNODE27 NEQ8,11 SETUPHASHARRAY3
*4,5 declare%:13 FRPTQ17 NEVER12 SMALLEST12
+4,5 DIFFERENCE9,10 FTIMES10 never13,14 smallest13,14
+-4,5 DO12 GE8,9,10 NLSETQ16 STANDARD7
-4,5 do13,14 ge8,9,10 NOR8,9,11 SUCHTHAT12
->8 DURING16 GEQ8,9,10,11 nor8,9,11 suchthat13
/4,5 during16 geq8,9,10,11 NOT9,11 SUM12
/CLISTPUT6 EACHTIME12 GREATERP8,9,10,11 NTH6,7 sum13,14
/CLISTPUT6 eachtime13 GT8,9,10 OLD12 THEN11
/CLISTPUT6 ELSE11 gt8,9,10 old13,14 then12
/CLISTPUT6 /NCONC6 ELSEIF11 IDIFFERENCE9,10 ON9,12 THEREIS12
/CLISTPUT6 /NCONC16 ELSEIF11 IF11 on13 thereis13,14
/CLISTPUT6 /PUT6 elseif12 if12 or8,9,11 THRU12
/CLISTPUT6 /PUTASSOC6 ELT11 IGREATERP8,9,10,11 OR8,9,11,16 thru13
/CLISTPUT6 /PUTHASH6 EQ8,9,11 ILESSP8,9,10,11 ORIGINAL12 TIMERUNITS15
/CLISTPUT6 /PUTPROP6 EQP8,11 IMINUS9,10 original13 timerunits15
/CLISTPUT6 /REPLACEFIELD6 EQUAL8,11 IN12 PLUS9,10 timerUnits15,16
/CLISTPUT6 /RPLACA6 EQUALS8,9,10,11 in13 PROG16 TIMES9,10
/CLISTPUT6 /RPLACD6 ERSETQ16 INSIDE12 PROG*16 TO12
/CLISTPUT6 /RPLNODE6 EXPT9 inside13,14 PROG116 to13
/CLISTPUT6 /RPLNODE26 FASSOC6 INTEGER7 PROG216 UNDOABLE7
/CLISTPUT6 /SETA6 FAST6 IPLUS9,10 PROGN16 UNDONLSETQ17
%:4 FCOLLECT12 IQUOTIENT9,10 PUT6,7 PUTASSOC6,7 UNLESS13
<4,5 fcollect13,14 ISTHERE12 PUTASSOC6,7 UNLESS13
=4,5 FDIFFERENCE10 isthere13 PUTHASH6,7 UNTIL13
=>8 FETCHFIELD5,6,7 ITIMES9,10 PUTPROP6,7 UNTIL13
>5 FFETCHFIELD6 JOIN12 QUOTIENT9,10 UNTILDATE16
ADV-PROG16 FGTP8,10,11 join13,14 REPEATUNTIL12 untilDate16
ADV-SETQ16 FINALLY12 LARGEST12 REPEATUNTIL13 untildate16
ALWAYS12 finally13 largest13,14 REPEATWHILE12 USINGBOX15
always13 FIND12 LAST5,7 REPEATWHILE13 usingbox15
and8,9,11 find13 LE8,9,10 REPLACEFIELD5,6,7 usingBox15,16
AND8,9,11,16 FIRST12 le8,9,10 RESETBUFFS16 USINGTIMER15
ARG16 first13 LEQ8,9,10,11 RESETFORM17 usingtimer15
AS12 FIXED6 leq8,9,10,11 RESETLST16 USINGTimer15,16
as13 FLAST6 LET17 RESETSAVE17 WHEN13
ASSOC5,6,7 FLOATING6 LET*17 RESETVAR17 when13
BIND12 FMEMB5,6,8 LET*17 RESETVARS16 WHERE13
bind13 FMINUS10 LISTPUT5,7 RESOURCENAME16 where13
BREAKDOWN3 FNTH6 LISTPUT15,7 resourceName16 WHILE13
BY12 FOR12 lt8,9,10 resourceName16 while13
by13 for13 lt8,9,10 RETURN17 RPAQ17 XNLSETQ17
CALLS3 FORDURATION16 MAKEMATCH3 RPLACA6,7 ^4,5
clisp11 forDuration16 MAPCON6,7 RPLACD6,7 _4,5
CLISP11,17 forDuration16 MAPCONC6,7 RPLACD6,7 ~4,5
CLISP11,17 FPLUS10 MATCH11 RPLNODE6,7 ←4,5
5 COLLECT12 FQUOTIENT10 match11 RPLNODE26,7,8 ↑4,5
collect13,14 FREPLACEFIELD5,6 MEMB5,6,7,8 RPTQ17
COND16 FROM12 MEMBER8,10,11 SAVESETQ17
COUNT12 from13 MINUS8,9,10 SETA6,7,8,11
count13,14 FRPLACA6 MIXED7 SETARG17

VARIABLE INDEX

%#CLISPARRAY3 CLISPINFIXSPLST4 DWIMINMACROSLST4 NOFIXFNSLST02
CHECKCARATOMFLG4 CLISPISWORDSPLST4 DWIMUSERFORMS3 NOFIXVARSLST3
CLISPARIITHCLASSLST4 CLISPPLASTSUB4 EDITCOMSA4 NOFIXVARSLST02
CLISPARIITHOPLST4 CLISPTRANFLG3 I.S.OPRLST14 NOSPELLFLG2
CLISPCHARRAY4 CLISPTRANFLG3 I.S.OPRLST14 OKREEVALST3
CLISPCHARS4 COMPILERUSERFN3 INITISOPRS12 RECORDHASHFLG3
CLISPDUMMYFORVARS14 DECLWORDS5 INVISIBLEVARS14 RPARKEY2
CLISPFLG4 DURATIONCLISPWORDS15 LAMBDAASPLST3 SYSLOCALVARS14
CLISPFORWORDSPLST14 DWIMCHECK#ARGSFLG3 LAMS3 TREATASCLISPFLG4
CLISPHELPFLG4 DWIMCHECKPROGLABELSFLG3 LEFT.ARROW4 WTFIXCHCONLST2
CLISPI.S.GAG12 DWIMEQUIVLIST3 LEFT.ARROWS.BITTABLE4 WTFIXCHCONLST12
CLISPITRANFLG11 DWIMESSGAG3 LPARKEY2
CLISPITRANFLG11 DWIMIFYFLG3 NOFIXFNSLST3

{MEDLEY}<sources>CLISP.;1

OPTIMIZER INDEX

CLISP%11
