File created: 14-Sep-2022 10:25:40 **{DSK}<users>kaplan>local>medley3.5>working-medley>sources>DWIM.;3**

changes to: (VARS DWIMCOMS)

previous date: 13-Sep-2022 09:16:44 {DSK}<users>kaplan>local>medley3.5>working-medley>sources>DWIM.;2

Read Table: INTERLISP

Package: INTERLISP

Format: XCCS

```
;;
;; Copyright (c) 1982-1986, 1988, 1990-1991, 2021 by Venue & Xerox Corporation.

(RPAQQ DWIMCOMS
       [(FNS DWIM NEWFAULT1 /DWIMCHECKTRAN)
        (INITVARS (DWIMWAIT 10)
               (LCASEFLG T))
        (VARS DWIMODELST)
        (FNS RETDWIM2 RETDWIM3 FIXATOM2 SPLIT89 WTFIXLOADEF CLISP% )
        (COMS (FNS VARSBOUNDINEDITCHAIN VARSBOUNDINFORM)
              (BLOCKS (VARSBOUNDINEDITCHAIN VARSBOUNDINEDITCHAIN VARSBOUNDINFORM)))
        (FNS DWIMLOADFNS?)
        (APPENDVARS (DWIMUSERFORMS (DWIMLOADFNS?)))
        (VARS (DWIMLOADFNSFLG T))
        (FNS CLISPLOOKUP0 CLISPLOOKUP1 CLISPLOOKUP2 CLISPERROR CLISPDEC CLISPDEC0 CLISPDEC1 GETLOCALDEC)
        (FNS COMPILEUSERFN COMPILEUSERFN1 USEDFREE CLISPTRAN compilation)
        (FNS CLISPFORERR CLISPFORERR1 I.S.OPR)
        (DECLARE%: EVAL@COMPILE DONTCOPY (ADDVARS (NLAML BREAK1)))
        (BLOCKS (NEWFAULT1BLOCK NEWFAULT1 /DWIMCHECKTRAN (ENTRIES NEWFAULT1)
                        (GLOBALVARS %#CLISPARRAY)
                        (NOLINKFNS WTFIX))
               (CLISPLOOKUP0 CLISPLOOKUP0 CLISPLOOKUP1 CLISPLOOKUP2 (GLOBALVARS DECLWORDS CLISPRECORDTYPES
                                                                                CLISPTRANFLG)
                      (LOCALFREEVARS WORD CLASS CLASSDEF VAR1 VAR2))
               (CLISPDECBLOCK CLISPDEC CLISPDEC0 CLISPDEC1 GETLOCALDEC (GLOBALVARS CLISPRECORDTYPES DECLWORDS
                                                                                  CLISPARITHOPLST CLISPARITHCLASSLST
                                                                                  COMMENTFLG SKORLST1)
                      (ENTRIES CLISPDEC CLISPDEC0 GETLOCALDEC)
                      (LOCALFREEVARS FAULTFN)))
        (GLOBALVARS DWIMODELST DWIMKEYLST DWIMWAIT LCASEFLG CLISPFORWORDSPLST I.S.OPRLST SKORLST3 DWIMLOADFNSFLG
               CLISPTRANFLG CLISPARRAY %#CLISPARRAY)
        (DECLARE%: DONTEVAL@LOAD DOEVAL@COMPILE DONTCOPY COMPILERVARS (ADDVARS (NLAMA USEDFREE CLISP% )
                                                                                (NLAML)
                                                                                (LAMA FIXATOM2])

(DEFINEQ

(DWIM
  [LAMBDA (X)                                                    (* wt%: "22-OCT-78 21:02")
    (COND
       ((NULL X)
        (/PUTD 'FAULT1 (GETD 'OLDFAULT1))
        (/SETATOMVAL 'DWIMFLG NIL)
        (/SETATOMVAL 'ADDSPELLFLG NIL))
       ((SETQ X (ASSOC X DWIMODELST))
        (/PUTD 'FAULT1 (GETD 'NEWFAULT1))
        (/SETATOMVAL 'DWIMFLG T)
        (/SETATOMVAL 'ADDSPELLFLG T)
        [MAPC (CDDR X)
              (FUNCTION (LAMBDA (X)
                          (AND (LISTP X)
                               (SET (CAR X)
                                    (CDR X]
        (CADR X))
       (T (ERROR '"not on DWIMODELST." '"" T])

(NEWFAULT1
  [LAMBDA (FAULTX FAULTARGS FAULTAPPLYFLG)                       ; Edited 22-Mar-2021 13:01 by larry
                                                                 (* Replaces FAULT1)
    (PROG [(FAULTZ (if FAULTAPPLYFLG
                       then FAULTX
                     elseif (LISTP FAULTX)
                       then (CAR FAULTX]
          (if [AND FAULTZ (LITATOM FAULTZ)
                   (GETD FAULTZ)
                   (SETQ FAULTZ (/DWIMCHECKTRAN (GETD FAULTZ]
              then (if FAULTAPPLYFLG
                       then (GO RETAPPLY)
                     else (SETQ FAULTZ (CONS FAULTZ (CDR FAULTX)))
                          (GO RETEVAL)))

          (* Covers the case where an atom has a definition that has a clisp translation, e.g.
          FOO is defined as (QLAMBDA --) There are two cases, FOO (args) and
          (FOO args))
```

```
          (if (LISTP FAULTX)
              then (if (SETQ FAULTZ (/DWIMCHECKTRAN FAULTX))
                       then

          (* Covers the case where the form has a clis translation itself, (most common)%, and the case where faultx is a function
          object being applied and has a clisptranslation.)

                            (if FAULTAPPLYFLG
                                then (GO RETAPPLY)
                              else (GO RETEVAL)))
                       (if (AND (NULL FAULTAPPLYFLG)
                                (LISTP FAULTX)
                                (LISTP (SETQ FAULTZ (CAR FAULTX)))
                                (SETQ FAULTZ (/DWIMCHECKTRAN FAULTZ)))
                           then

          (* Covers the case where car of form is a function objection with a clisp translation, e.g.
          ((QLAMBDA --) --))

                           (SETQ FAULTZ (CONS FAULTZ (CDR FAULTX)))
                           (GO RETEVAL)))
          (SETQ FAULTZ (WTFIX FAULTX FAULTARGS FAULTAPPLYFLG))   (* info for diagnostic printed by original FAULT1.)
          (RETURN (OLDFAULT1 FAULTX FAULTARGS FAULTAPPLYFLG FAULTZ))
      RETAPPLY
          (RETAPPLY (FUNCTION FAULTAPPLY)
                FAULTZ FAULTARGS T 'INTERNAL)
      RETEVAL
          (RETEVAL 'FAULTEVAL FAULTZ])
```

(/DWIMCHECKTRAN

```
  [LAMBDA (X)                                                    (* lmm "10-MAR-83 22:37")
    (DECLARE (GLOBALVARS %#CLISPARRAY CLISPARRAY CLISPTRANFLG))
    (OR (AND CLISPARRAY (GETHASH X CLISPARRAY))
        (AND CLISPTRANFLG (EQ (CAR X)
                              CLISPTRANFLG)
             (PROG1 (CADR X)
                 [COND
                    ((OR CLISPARRAY %#CLISPARRAY)
                     (CLISPTRAN X (CADR X))
                     (/RPLNODE X (CADDR X)
                            (CDDDR X])])

)

(RPAQ? DWIMWAIT 10)

(RPAQ? LCASEFLG T)

(RPAQQ DWIMODELST ((C CAUTIOUS (APPROVEFLG . T))
                   (T TRUSTING (APPROVEFLG))))

(DEFINEQ
```

(RETDWIM2

```
  [LAMBDA (X $TAIL N M)                                          (* wt%: 25-FEB-76 2 3)

          (* N is a printlevel affecting TAILS, M one affecting elementens.
          Value is a copy of X as though printed with these levels.)

    (AND (NULL N)
         (SETQ N 3))
    (AND (NULL M)
         (SETQ M 1))
    (RETDWIM3 X $TAIL N M])
```

(RETDWIM3

```
  [LAMBDA (X $TAIL N1 M1)                                        (* wt%: 25-FEB-76 2 3)
    (COND
       ((NLISTP X)
        X)
       ((ILESSP M1 0)
        '&)
       (T (CONS (RETDWIM3 (CAR X)
                          NIL N1 (SUB1 M1))
                (COND
                   [$TAIL (COND
                              ((EQ X $TAIL)                      (* Only begin counting down when you reach TAIL.)
                               (RETDWIM3 (CDR X)
                                      NIL
                                      (SUB1 N1)
                                      M1))
                              (T (RETDWIM3 (CDR X)
                                           $TAIL N1 M1]
                   ((IGREATERP N1 0)
```

```
                    (RETDWIM3 (CDR X)
                           $TAIL
                           (SUB1 N1)
                           M1))
                  ((CDR X)
                   '(--])
```

(FIXATOM2

```
  [LAMBDA X                                                    (* Value is the last argument on the stack.)
    (ARG X X])
```

(SPLIT89

```
  [LAMBDA (N POS)
```

(* Generates command that replaces atoms containing 8 or 9 with the corresponding atom or atoms separated by the 8 or 9 so macro calling it can determine where to insert or remove parentheses.)

```
    (PROG (X Y Z)
          (SETQ X (DUNPACK (CAR L)
                           SKORLST3))
          [SETQ Y (COND
                     (POS (SETQ Y (NLEFT X POS)))
                     (T (FMEMB N X]
          [COND
             ((NULL Y)
```

(* User has already corrected atom containing 8 or 9 Now we must guess what form it is.
Assume if N is 8, was error of form 8CONS, if 9, X9)

```
              (RETURN (LIST (COND
                                ((EQ N 8)
                                 'B)
                                (T 'A))
                            N]
          [COND
             ((CDR Y)
              (SETQ Z (CONS (PACK (CDR Y))
                            Z]
          (SETQ Z (CONS N Z))
          [COND
             ((NEQ Y X)
              (SETQ Z (CONS (PACK (LDIFF X Y))
                            Z]
          (SETQ SPLIT89FLG Z)
          (RETURN (CONS '%: Z])
```

(WTFIXLOADEF

```
  [LAMBDA (FAULTEM1)                                           ; Edited  5-Apr-88 16:04 by amd
    ;; FAULTEM1 is the value of the FILEDEF property.
    (PROG (FAULTEM2 FAULTEM3)
          (SETQ FAULTFN NIL)                                   ; So file package wont try to update it
          (RETURN (COND
                     ((AND DWIMIFYFLG DWIMIFYING))
                     ([NULL (SETQ FAULTEM2 (OR (FINDFILE (PACKFILENAME 'BODY [SETQ FAULTEM2
                                                                                   (COND
                                                                                      ((ATOM FAULTEM1)
                                                                                       ; FAULTEM1 is the name of the file.
                                                                                        FAULTEM1)
                                                                                       (T
                                                                                       ; (CAR FAULTEM1) is the name of the file.  CDR is the list of
                                                                                       ; functions.
                                                                                          (PROG1 (CAR FAULTEM1)
                                                                                              (SETQ FAULTEM1
                                                                                               (CDR FAULTEM1)))]
                                                                                 'EXTENSION FASL.EXT)
                                                                          T)
                                                                (FINDFILE (PACKFILENAME 'BODY FAULTEM2 'EXTENSION COMPILE.EXT)
                                                                          T]         ; If file isnt there don't bother to ask.
                      NIL)
                     ((COND
                         ((OR (ATOM FAULTEM1)
                              (NLISTP (CAR FAULTEM1)))
                          (EQ (ASKUSER DWIMWAIT 'Y (LIST '"Shall I load " FAULTEM1)
                                       DWIMKEYLST)
                              'Y))
                         ([STRINGP (SETQ FAULTEM3 (EVAL (PROG1 (CAR FAULTEM1)
                                                                (SETQ FAULTEM1 (CDR FAULTEM1)))]
                          ;; (CAR FAULTEM1) computes either a string to be typed, or T or NIL, meaning do it or dont do it.  not sure if this is
                          ;; being used aaymore
                          (FIXSPELL1 '"" FAULTEM3 '"" NIL 'MUSTAPPROVE))
                         (T FAULTEM3))
                      [COND
```

```
                        ((ATOM FAULTEM1)
                         (LOAD FAULTEM2 'SYSLOAD))
                        (T (LOADFNS FAULTEM1 FAULTEM2 'SYSLOAD]
                    T])
```

(CLISP%

```
  [NLAMBDA CLISPX
    (PROG (CLISPTEM)
          [COND
             ((AND (OR CLISPARRAY %#CLISPARRAY)
                   (EQ [CAR (SETQ CLISPTEM (PROG1 (BLIPVAL 'EVAL (SETQ CLISPTEM (STKNTH -1 CLISPTRANFLG)))
                                                  (RELSTK CLISPTEM]
                       CLISPTRANFLG)
                   (EQ (CDR CLISPTEM)
                       CLISPX))
              (CLISPTRAN CLISPTEM (CADR CLISPTEM))
              (/RPLNODE CLISPTEM (CADDR CLISPTEM)
                        (CDDDR CLISPTEM]
          (RETURN (EVAL (CAR CLISPX)
                        'INTERNAL])

)

(DEFINEQ
```

(VARSBOUNDINEDITCHAIN

```
  [LAMBDA (EDITCHAIN)                                    (* lmm "27-FEB-83 10:55")

          (* Climbs EDITCHAIN and makes list of all bound variabes. Sets EXPR to the top level expression, i.e.
          (CAR (LAST EDITCHAIN)))

    (MAPCONC EDITCHAIN (FUNCTION VARSBOUNDINFORM])
```

(VARSBOUNDINFORM

```
  [LAMBDA (FORM)                                         ; Edited 22-Mar-2021 13:08 by larry
    (DECLARE (GLOBALVARS LAMBDASPLST COMPILERMACROPROPS))
    (PROG ((FN (CAR FORM))
           TEM MACRO)
          (RETURN (AND (LITATOM FN)
                       (COND
                          ((FMEMB FN LAMBDASPLST)
                           (APPEND (ARGLIST FORM)))
                          [(EQMEMB 'BINDS (GETPROP FN 'INFO))
                           (MAPCAR (CADR FORM)
                                   (FUNCTION (LAMBDA (X)
                                                     (COND
                                                        ((NLISTP X)
                                                         X)
                                                        (T (CAR X]
                          ((EQ [CAR (LISTP (SETQ TEM (GETPROP FN 'CLISPWORD]
                               'FORWORD)
                           (PROG ((TAIL FORM)
                                  VAL INVAR ELT)
                             FORWORDLP
                                 (SETQ INVAR (SELECTQ (CDR TEM)
                                                      ((for bind as)
                                                          T)
                                                      NIL))
                             LP  (OR (SETQ TAIL (CDR TAIL))
                                     (RETURN VAL))
                                 (SETQ ELT (CAR TAIL))
                                 [COND
                                    ((NOT (LITATOM ELT))
                                     [COND
                                        ((AND INVAR (EQ (CADR (LISTP ELT))
                                                        '_))
                                         (SETQ VAL (CONS (CAR ELT)
                                                         VAL]
                                     (GO LP))
                                    ((EQ [CAR (LISTP (SETQ TEM (GETPROP ELT 'CLISPWORD]
                                         'FORWORD)
                                     (GO FORWORDLP))
                                    ((EQ ELT '_)
                                     (SETQ TAIL (CDR TAIL)))
                                    (INVAR (SETQ VAL (CONS ELT VAL]
                                 (GO LP)))
                          ((SETQ TEM (CHECKTRAN FORM))
                           (VARSBOUNDINFORM TEM))
                          ((AND (SETQ TEM (GETLIS FN COMPILERMACROPROPS))
                                (NOT (EQUAL (SETQ TEM (MACROEXPANSION FORM (CADR TEM)))
                                            FORM)))
                           (VARSBOUNDINFORM TEM])

)
```

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

(BLOCK%: VARSBOUNDINEDITCHAIN VARSBOUNDINEDITCHAIN VARSBOUNDINFORM)
)

(DEFINEQ
```

(DWIMLOADFNS?

```
  [LAMBDA NIL                                                    (* wt%: "27-SEP-79 18:15")
    (PROG [TEM (FN (COND
                       (FAULTAPPLYFLG FAULTX)
                       (T (CAR FAULTX]
          (RETURN (COND
                     ((AND DWIMLOADFNSFLG (NULL (AND DWIMIFYFLG DWIMIFYING))
                           (LITATOM FN)
                           (NULL (FGETD FN))
                           (SETQ TEM (EDITLOADFNS? FN))
                           (OR (EQ (CAR (SETQ TEM (LOADFNS (LISPXPRINT FN T T)
                                                           TEM)))
                                   FN)
                               (PROGN (LISPXPRINT (CAR TEM)
                                                  T)
                                      NIL)))
                      [AND (NULL TYPE-IN?)
                           (SETQ SIDES (CDR (LISTGET1 LISPXHIST 'SIDE]
                      FAULTX])

)

(APPENDTOVAR DWIMUSERFORMS (DWIMLOADFNS?))

(RPAQQ DWIMLOADFNSFLG T)

(DEFINEQ
```

(CLISPLOOKUP0

```
  [LAMBDA (WORD VAR1 VAR2 DECLST LISPFN CLASS CLASSDEF)          (* lmm " 5-SEP-83 23:53")

          (* LISPFN is returned if no local declaration is found affecting FN.
          CLASS is the CLASS for FN, e.g. RPLACA, +, MEMB, etc. CLASS is supplied when looking up local record declaration
          (in his case it is RECORD) or when looking up a local value for a variable, such as VARDEFAULT ina pattern match, in
          which case it is VALUE.)

          (* To define a new class of functions a la RPLACA, FRPLACA, and /RPLACA, one must add all three names to
          DECLWORDS, put the name of the standard one on the property lits of each under property CLISPCLASS, and put under
          the standard one on property CLISPCLASSDEF the property (ACCESS standard undoable fast) version, where undoable or
          fast can be NIL. Then CLISPDEC STANDARD, UNDOABLE, or FAST will have the right effect, and calling CLISPLOOKUP
          on the names of either of the functions will eturn the current "Setting" %.)

    (PROG (TEM)
          [COND
             ((OR (NULL DECLST)
                  (NULL CLASS))                                  (* CLISPLOOKUP0 is always supposed to be called with a
                                                                 non-NIL CLASS and DECLST.)
              (SHOULDNT 'CLISPLOOKUP0]
          [OR CLASSDEF (SETQ CLASSDEF (GETPROP CLASS 'CLISPCLASSDEF]
          (SETQ VAR1 (CLISPLOOKUP2 VAR1))
          (SETQ VAR2 (CLISPLOOKUP2 VAR2))
          (RETURN (COND
                     ((SETQ TEM (CLISPLOOKUP1 DECLST))
                      TEM)
                     (T

          (* The last GETP in the OR below %, i.e. for CLASS, is so we dont have to implement global declaraions by puttig a LISPFN
          property on each member of the class.)

                        (SELECTQ CLASS
                             (VALUE (GETATOMVAL WORD))
                             ((RECORD RECORDFIELD)
                                 NIL)
                             (OR LISPFN (GETPROP WORD 'LISPFN)
                                 (GETPROP CLASS 'LISPFN)
                                 WORD])
```

(CLISPLOOKUP1

```
  [LAMBDA (LST)                                                  (* lmm "23-Aug-84 17:56")
                                                                 (* Searches LST for a delcaration releveant to CLASS, which is
    equal to (GETP WORD (QUOTE CLISPCLASS.)))
    (PROG (TEM VAL)
      LP  (COND
             ((NULL LST)
              (RETURN VAL))
             [(LISTP (SETQ TEM (CAR LST)))
              (AND CLISPTRANFLG (EQ (CAR TEM)
                                    CLISPTRANFLG)
                   (SETQ TEM (CDDR TEM)))
```

```
              (COND
                 [(EQ (CADR TEM)
                      '=)
                  (AND (EQ CLASS 'VALUE)
                       (EQ (CAR TEM)
                           WORD)
                       (SETQ VAL (CADDR TEM]
                 [(OR (EQ CLASS 'RECORD)
                      (EQ CLASS 'RECORDFIELD))
                  (AND (FMEMB (CAR TEM)
                              CLISPRECORDTYPES)
                       (COND
                          ((EQ CLASS 'RECORDFIELD)
                           (FMEMB WORD (RECORDFIELDNAMES TEM)))
                          (T (EQ WORD TEM)))
                       (SETQ VAL (CAR LST]
                 ((EQ (CAR TEM)
                      CLASS)
```

(* So user can look up his own %'classes', e.g. say (CLISP%: (FOOTYPE)) and then look up FOOTYPE.
Terry uses this.)

```
                  (SETQ VAL (CAR LST)))
                 ([AND (OR (EQ (SETQ TEM (CAAR LST))
                               VAR1)
                           (EQ TEM VAR2))
                       (SETQ TEM (CLISPLOOKUP1 (CDAR LST]
                  (RETURN TEM]
            [[ATOM (SETQ TEM (GETPROP (CAR LST)
                                      'CLISPCLASS]
```

(* E.g. WORD is FRPLACA CLASS is RPLACA, and (CAR LST) is /RPLACA.
TEM is also RPLACA.)

```
                (AND (EQ TEM CLASS)
                     (SETQ VAL (CAR LST]
               ([AND (EQ (CAR TEM)
                         (CAR CLASSDEF))
                     (SETQ TEM (CAR (NTH (CDR CLASSDEF)
                                         (CDR TEM]
```

(* E.G. WORD is FRPLACA and (CAR LST) is FAST. or WORD is + and
(CAR LST) is FLOATING. The eason for checking that the nth element is not nil is that FAST does not apply to NCONC,
even though both are ACCESS type declarations, similaly, undoable does not apply to LAST.)

```
                (SETQ VAL TEM)))
             LP1 (SETQ LST (CDR LST))
                 (GO LP])
```

(CLISPLOOKUP2

```
[LAMBDA (X)
  (COND
     ((NLISTP X)
      X)
     ((OR (EQ (CAR X)
              'SETQ)
          (EQ (CAR X)
              'SETQQ))
      (CADR X))
     ((EQ (CADR X)
          '_)
      (CAR X])
```

(CLISPERROR

```
[LAMBDA (TYPE FLG)                                          (* wt%: " 1-OCT-78 00:22")
  (COND
     (FLG (EVQ FAULTFN)
          (EVQ PARENT)
          (EVQ TAIL)
          (EVQ TYPE-IN?)))
  (AND (NULL DWIMESSGAG)
       (NEQ TYPE 'ALREADYPRINTED)
       (PROG (TEM AT IN)
             (COND
                ((NULL TYPE-IN?)
                 (FIXPRINTIN FAULTFN)
                 (LISPXSPACES 1 T)))
             (LISPXPRIN1 (SELECTQ [SETQ TEM (COND
                                            ((ATOM TYPE)
                                             TYPE)
                                            (T (CAR TYPE]
                                  (1 '"missing operand")
                                  (2 '"missing operator")
                                  ((%: :%: -> =>)
                                       (LISPXPRIN1 '"improper use of " T)
```

```
                                TEM)
                            (4 '"bad if statement")
                            (_ '"incorrect use of _")
                            (FIELDNAME '"undefined field name")
                            (PHRASE '"can't parse this phrase")
                            (CARATOM '"car or cdr of non-list taken")
                            (COND
                               ((EQ (CAR (LISTP TEM))
                                    'BRACKET)
                                (LISPXPRIN1 '"missing " T)
                                (CADR TEM))
                               (T TEM)))
                      T)
                (COND
                   ((LISTP TYPE)
                    (GO A))
                   ((NEQ PARENT TAIL)
                    (LISPXPRIN1 '" at " T)
                    (LISPXPRIN2 (RETDWIM2 (CAR TAIL))
                           T T)))
                (LISPXPRIN1 '" in    " T)
                (LISPXPRIN2 (RETDWIM2 (OR PARENT FAULTX)
                                  TAIL)
                       T T)
                (LISPXTERPRI T)
                (RETURN)
         A      (SETQ AT (CADR TYPE))
                (SETQ IN (CADDR TYPE))
                (COND
                   ((OR (EQ AT IN)
                        (NULL IN))
                    (LISPXPRIN1 '" in    " T)
                    (LISPXPRINT (RETDWIM2 AT)
                           T T)
                    (RETURN)))
                (LISPXTERPRI T)
                (LISPXPRIN1 '"at    " T)
                (MAPRINT (RETDWIM2 AT (CDDR AT))
                       T
                       '"... "
                       '%) NIL NIL T)
                (LISPXTERPRI T)
                (LISPXPRIN1 '"in    " T)
                (LISPXPRINT (RETDWIM2 IN)
                       T T)
                (RETURN])
```

(CLISPDEC

```
  [LAMBDA (DECLST)                                          (* wt%: "10-AUG-78 00:31")
                                                            (* Does global declaratin)
    (AND DECLST (ATOM DECLST)
         (SETQ DECLST (LIST DECLST)))
    (PROG ((LST DECLST)
           TEM CLASSDEF)
      TOP (COND
             ((NULL LST)
              (RETURN DECLST)))
          (COND
             [(LISTP (CAR LST))
              (COND
                 ((FMEMB (CAAR LST)
                         CLISPRECORDTYPES)
                  (EVAL (CAR LST)))
                 (T (GO ERROR]
             [(FMEMB (CAR LST)
                     CLISPARITHCLASSLST)
              (MAPC CLISPARITHOPLST (FUNCTION (LAMBDA (X)        (* E.g. X IS *, /, +, ETC.)
                                                (COND
                                                   ((SETQ TEM (GETPROP X 'LISPFN))
                                                                  (* May have been disabled)
                                                    (/REMPROP TEM 'CLISPINFIX)
                                                    (COND
                                                       ([SETQ TEM
                                                         (CAR (NTH [CDR (OR (GETPROP X 'CLISPCLASSDEF)
                                                                            (GETPROP (GETPROP X 'CLISPCLASS)
                                                                                     'CLISPCLASSDEF]
                                                                   (CDR (GETPROP (CAR LST)
                                                                                 'CLISPCLASS]
                                                    (/PUT X 'LISPFN TEM)

          (* E.G. CLISPCLASS for FLOATING is (ARITH . 2)%, for * is (ARITH ITIMES FTIMES TIMES) meaning the FLOATING
          version for * is FTIMES.)

                                                    (/PUT TEM 'CLISPINFIX X]
             [(SETQ CLASSDEF (GETPROP (CAR LST)
                                      'CLISPCLASS))
```

```
            (COND
               [(LISTP CLASSDEF)                                    (* e.g. clipdec (fast))
                (MAPC DECLWORDS (FUNCTION (LAMBDA (X)
                                          (COND
                                             ([AND [EQ (CAR CLASSDEF)
                                                       (CAR (SETQ TEM (GETPROP X 'CLISPCLASSDEF]
                                                   (SETQ TEM (CAR (NTH (CDR TEM)
                                                                       (CDR CLASSDEF]
                                              (/PUT X 'LISPFN TEM]
               (T                                                   (* e.g. clispdec (fassoc))
                  (/PUT CLASSDEF 'LISPFN (CAR LST]
            [(FMEMB (CAR LST)
                    DECLWORDS)
             (COND
                ([ATOM (SETQ TEM (GETPROP (CAR LST)
                                          'CLISPCLASS]
                 (/PUT TEM 'LISPFN (CAR LST)))
                (T (GO ERROR]
            ((SETQ TEM (OR (PROG (TYPE-IN? FAULTFN)
                                 (RETURN (FIXSPELL (CAR LST)
                                                   NIL DECLWORDS)))
                           (GO ERROR)))
             (/RPLNODE LST TEM (CDR LST))
             (GO TOP)))
          (SETQ LST (CDR LST))
          (GO TOP)
      ERROR
          (ERROR '"illegal declaration" (CAR LST])
```

(**CLISPDEC0**

```
  [LAMBDA (X FN)                                                    (* wt%: 29-JUL-76 20 56)
    (/RPLNODE X COMMENTFLG (CONS 'DECLARATIONS%: (CLISPDEC1 (CDR X)
                                                            FN)))
    (CDDR X])
```

(**CLISPDEC1**

```
  [LAMBDA (X FAULTFN)                                               (* wt%: "13-JUN-78 17:31")
    (MAPCON X (FUNCTION (LAMBDA (X)
                        (PROG (TEM TYPE-IN?)
                          TOP (RETURN (COND
                                      [(LISTP (CAR X))
                                       (LIST (COND
                                                ((OR (EQ (CADAR X)
                                                         '=)
                                                     (FMEMB (CAAR X)
                                                            CLISPRECORDTYPES)
                                                     (EQ (CAAR X)
                                                         'RECORDS))
                                                 (CAR X))
                                                (T (CONS (CAAR X)
                                                         (CLISPDEC1 (CDAR X]
                                      ((FMEMB (CAR X)
                                              DECLWORDS)
                                       (LIST (CAR X)))
                                      ((FIXSPELL (CAR X)
                                                 NIL DECLWORDS NIL X NIL NIL NIL (DUNPACK (CAR X)
                                                                                          SKORLST1))
                                       (GO TOP))
                                      (T (ERROR '"illegal declaration" (CAR X])
```

(**GETLOCALDEC**

```
  [LAMBDA (EXPR FN)                                                 (* lmm "26-Sep-84 16:38")
    (AND (LISTP EXPR)
         (COND
            ((FMEMB (CAR EXPR)
                    LAMBDASPLST)
             (for (TL _ (CDDR EXPR)) by (CDR TL) while TL bind X when (LISTP (SETQ X (CAR TL)))
                do (SELECTQ (CAR X)
                            (BREAK1 (SETQ TL (CADR X)))
                            (ADV-PROG [SETQ TL (CADR (CAR (LAST (CADDR (CADDR X])
                            (COND
                               ((AND (EQ (CAR X)
                                         COMMENTFLG)
                                     (EQ (CADR X)
                                         'DECLARATIONS%:))
                                (RETURN (CDDR X)))
                               [(EQ (CAR X)
                                    'CLISP%:)
                                (RETURN (CLISPDEC0 X (OR FN FAULTFN]
                               ((FMEMB (CAR X)
                                       '(DECLARE DECLARE%:))
                                (RETURN (for Y in (CDR X) do [COND
                                                               ((EQ (CAR Y)
                                                                    'CLISP%:)
```

```
                                                        (RETURN (CDR Y]
                                                    (COND
                                                       ((AND (EQ (CAR Y)
                                                                 COMMENTFLG)
                                                             (EQ (CADR Y)
                                                                 'DECLARATIONS%:))
                                                        (RETURN (CDDR Y])

)

(DEFINEQ
```

(**COMPILEUSERFN**

```
  [LAMBDA (X Y)                                                    ; Edited 22-Mar-2021 13:09 by larry
```

(* * this is an awful patch to fix the fact that COMPILEUSERFN1 is UNIONing something with OTHERVARS, which is an unbound specvar)

```
    (OR (BOUNDP 'OTHERVARS)
        (SETQ OTHERVARS NIL))
    (PROG (TEM)
          (RETURN (COND
                      ((CHECKTRAN Y))
                      [(LISTP (CAR Y))
                       (COND
                          ((SETQ TEM (CHECKTRAN (CAR Y)))
                           (CONS TEM (CDR Y)))
                          (DWIMFLG (COMPILEUSERFN1 Y)
                                 (COND
                                    ((CHECKTRAN Y))
                                    ((SETQ TEM (CHECKTRAN (CAR Y)))
                                     (CONS TEM (CDR Y]
                      ([AND (NLISTP (GETPROP (CAR Y)
                                             'CLISPWORD))
                            (NOT (AND (FMEMB (CAR Y)
                                             LAMBDASPLST)
                                      (NOT (FMEMB (CAR Y)
                                                  '(LAMBDA NLAMBDA]
                       NIL)
                      (DWIMFLG (COMPILEUSERFN1 Y)
                             (COND
                                ((AND CLISPARRAY (GETHASH Y CLISPARRAY)))
                                ((AND CLISPTRANFLG (EQ (CAR Y)
                                                       CLISPTRANFLG))
                                 (CADR Y))
                                ((NULL (GETPROP (CAR Y)
                                                'CLISPWORD))        (* IF's are transled directly into COND's, and dont use hashing.)
                                 Y)
                                ((NULL DWIMESSGAG)
```

(* user can set DWIMESSGAG to T and go away and the compilation will go through.)

```
                                 (PRIN1 '"unable to dwimify " T)
                                 (PRINT Y T)
                                 (CAR (NLSETQ (compilation Y])
```

(**COMPILEUSERFN1**

```
  [LAMBDA (Y)                                                      (* lmm "19-Jun-86 13:59")
    (PROG [(FLG (AND (LISTP COREFLG)
                     (CDR (FASSOC FN COREFLG]
          (LET ((NOSPELLFLG (OR NOSPELLFLG (NULL FLG)))
                (FILEPKGFLG (AND FILEPKGFLG FLG)))
```

(* FILEKGFLG is T when when compiling from in core, so that if function is changed, it will be marked.)

```
               (SETQ NOFIXFNSLST0 NOFIXFNSLST)
               (SETQ NOFIXVARSLST0 NOFIXVARSLST)
               (DWIMIFY0 Y FN (UNION ARGS OTHERVARS)
                      DEF)
               (COND
                  ((TAILP NOFIXFNSLST NOFIXFNSLST0)
```

(* For purposes of compilation, want anything added to NOFIXFNSLST0 to persist throughout copiling the whole file.)

```
                   (SETQ NOFIXFNSLST NOFIXFNSLST0)))
               (COND
                  ((TAILP NOFIXVARSLST NOFIXVARSLST0)
                   (SETQ NOFIXVARSLST NOFIXVARSLST0])
```

(**USEDFREE**

```
  [NLAMBDA A                                                       (* wt%: "20-SEP-77 22:10")
```

(* permits the user to declare freevars which will then be "noticed" by dwimify in thatthey wont be spelling corrected.)

```
    (SETQ FREEVARS (APPEND A FREEVARS])
```

(CLISPTRAN

```
  [LAMBDA (X TRAN)                                          (* bvm%: "21-Jan-86 00:41")
    (COND
       ((OR CLISPARRAY (COND
                          (%#CLISPARRAY (SETQ CLISPARRAY (HASHARRAY %#CLISPARRAY NIL NIL NIL T))
                                 (SETQ %#CLISPARRAY NIL)        (* Latter so user can turn clisphashing on and off by simply
                                                                reseting CLISPARRAY.)
                                 T)))                           (* Otherwise use CLISP% translation.)
        (/PUTHASH X TRAN CLISPARRAY))
       (TRAN                                                    (* Can be called erase a translation.)
             (/RPLNODE X CLISPTRANFLG (CONS TRAN (CONS (CAR X)
                                                       (CDR X])
```

(compilation

```
  [LAMBDA (EXP)
    (BREAK1 EXP T compilation])

)

(DEFINEQ
```

(CLISPFORERR

```
  [LAMBDA (X Y TYPE)                                        (* lmm " 4-SEP-83 22:56")
    (AND (NULL DWIMESSGAG)
         (PROG (TEM)
               (AND (FIXPRINTIN FAULTFN)
                    (SPACES 1 T))
               (LISPXPRIN1 '"error in iterative statement" T)
               (AND TYPE (LISPXPRINT '%, T)
                    (LISPXPRIN1 (SELECTQ TYPE
                                         (BOTH '"can't use both of these operators together")
                                         (TWICE '"operator appears twice")
                                         (MISSING '"missing operand")
                                         (WHAT (LISPXPRIN1 (CADR X)
                                                           T)
                                               '" what ?  (no i.v. specified)")
                                         NIL)
                                T))
               (LISPXPRINT '%: T)
               (COND
                  ((OR (AND X (NLISTP X))
                       (AND Y (NLISTP Y)))
                   (LISPXPRIN2 X T T)
                   (AND Y (LISPXPRIN2 Y T T))
                   (RETURN))
                  ((TAILP X Y)
                   (SETQ TEM X)
                   (SETQ X Y)
                   (SETQ Y TEM)))
               (CLISPFORERR1 X Y)
               (COND
                  (Y (LISPXSPACES 1 T)
                     (CLISPFORERR1 Y)))
               (TERPRI T)
               (RETURN)))
    (DWIMERRORRETURN])
```

(CLISPFORERR1

```
  [LAMBDA (X Y)                                             (* wt%: 25-MAR-77 22 58)
    (PROG (TEM)
          (COND
             ((NEQ X I.S.)
              (LISPXPRIN1 '" ... " T)))
          (SETQ TEM (OR [CADADR (SOME I.S.PTRS (FUNCTION (LAMBDA (Z)
                                                           (TAILP (CADR Z)
                                                                  X]
                                Y))
      LP  (LISPXPRIN2 (RETDWIM2 (CAR X)
                                NIL 3)
                      T T)
          (COND
             ((AND (SETQ X (CDR X))
                   (NEQ X TEM))
              (LISPXSPACES 1 T)
              (GO LP])
```

(I.S.OPR

```
  [LAMBDA (NAME FORM OTHERS EVALFLG)                        (* wt%: "18-SEP-78 23:22")

          (* E.g. NAME=SUM, FORM= (SETQ $$VAL ($$VAL + BODY))%, OTHERS=
          (FIRST $$VAL_0) I f evalflg is T, means form and others are to be EVALUATED at translation time.)
```

```
    (PROG ((UC (U-CASE NAME))
           LC NEWPROP OLDPROP NEWFLG)
          [COND
             ((NEQ NAME UC)

          (* LC is the name used for clispifying. for mostcases it is the lower case, but thi check lets users define i.s.oprs contaiing
          some lowercase and some uppercase letters)

              (SETQ LC NAME))
             (T (SETQ LC (L-CASE NAME]                          (* so tha user can call it with either loer or uppercase version.)
          (SETQ NEWFLG (NEQ (CAR (GETP LC 'CLISPWORD))
                            'FORWORD))
          (COND
             ((AND FORM (ATOM FORM)
                   (NEQ FORM 'MODIFIER))                        (* Synonym)
              (/PUT UC 'CLISPWORD (SETQ NEWPROP (LIST 'FORWORD LC FORM)))
              (SETQ OLDPROP (GETP LC 'CLISPWORD))
              (/PUT LC 'CLISPWORD NEWPROP)
              (/REMPROP LC 'I.S.OPR))
             ((AND OTHERS (NLISTP OTHERS)
                   (NULL EVALFLG))
              (ERROR "OTHERS must be a list of operators and operands" OTHERS))
             ((AND OTHERS (NEQ (CAR (GETPROP (CAR OTHERS)
                                             'CLISPWORD))
                               'FORWORD)
                   (NULL EVALFLG))
              (ERROR '"OTHERS must begin with an operator" OTHERS))
             (T (/PUT UC 'CLISPWORD (SETQ NEWPROP (CONS 'FORWORD LC)))
                (/PUT LC 'CLISPWORD NEWPROP)
                [SETQ NEWPROP (COND
                                 ((EQ FORM 'MODIFIER)
                                  'MODIFIER)
                                 [EVALFLG (CONS (AND FORM (CONS '= FORM))
                                                (AND OTHERS (CONS '= OTHERS]
                                 (T (CONS FORM OTHERS]
                (SETQ OLDPROP (GETP LC 'I.S.OPR))
                (/PUT LC 'I.S.OPR NEWPROP)))
          [COND
             ((EQUAL NEWPROP OLDPROP)
              (RETURN NAME))
             [(NULL NEWFLG)                                     (* redefined)
              [COND
                 ((EQ UC 'COLLECT)
                  (/REMPROP 'fcollect 'I.S.OPR]
              (AND (NEQ DFNFLG T)
                   (LISPXPRINT [CONS 'i.s.opr (CONS NAME '(redefined]
                               T))
              (AND CLISPARRAY (MAPHASH CLISPARRAY (FUNCTION (LAMBDA (TRAN EXP)
                                                              (AND (OR (MEMB UC EXP)
                                                                       (MEMB LC EXP))
                                                                   (/PUTHASH EXP NIL CLISPARRAY]
             (T                                                 (* defined for the first time)
                (/NCONC1 CLISPFORWORDSPLST UC)
                (/NCONC I.S.OPRLST (LIST UC LC]
          (AND FILEPKGFLG (MARKASCHANGED (COND
                                            ((EQ NAME UC)
                                             UC)
                                            (T

          (* file package doesnt care whether you give upper or lower case named to dumpi.s.oprs, however if user took pains to
          define thi i.ssop giving it a owercase definition, (Or mixed upper and lower case) then inform him about this i.s.opr in that
          fashion.)

                                                LC))
                                         'I.S.OPRS NEWFLG))
          (RETURN NAME])

)

(DECLARE%: EVAL@COMPILE DONTCOPY

(ADDTOVAR NLAML BREAK1)
)

(DECLARE%: DONTEVAL@LOAD DOEVAL@COMPILE DONTCOPY

(BLOCK%: NEWFAULT1BLOCK NEWFAULT1 /DWIMCHECKTRAN (ENTRIES NEWFAULT1)
       (GLOBALVARS %#CLISPARRAY)
       (NOLINKFNS WTFIX))

(BLOCK%: CLISPLOOKUP0 CLISPLOOKUP0 CLISPLOOKUP1 CLISPLOOKUP2 (GLOBALVARS DECLWORDS CLISPRECORDTYPES CLISPTRANFLG
                                                                         )
       (LOCALFREEVARS WORD CLASS CLASSDEF VAR1 VAR2))

(BLOCK%: CLISPDECBLOCK CLISPDEC CLISPDEC0 CLISPDEC1 GETLOCALDEC (GLOBALVARS CLISPRECORDTYPES DECLWORDS
                                                                            CLISPARITHOPLST CLISPARITHCLASSLST
                                                                            COMMENTFLG SKORLST1)
```

```
       (ENTRIES CLISPDEC CLISPDEC0 GETLOCALDEC)
       (LOCALFREEVARS FAULTFN))
)

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS DWIMODELST DWIMKEYLST DWIMWAIT LCASEFLG CLISPFORWORDSPLST I.S.OPRLST SKORLST3 DWIMLOADFNSFLG
       CLISPTRANFLG CLISPARRAY %#CLISPARRAY)
)

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

(ADDTOVAR NLAMA USEDFREE CLISP% )

(ADDTOVAR NLAML )

(ADDTOVAR LAMA FIXATOM2)
)

(PUTPROPS DWIM COPYRIGHT ("Venue & Xerox Corporation" 1982 1983 1984 1985 1986 1988 1990 1991 2021))
```

FUNCTION INDEX

VARIABLE INDEX