

File created: 17-Mar-87 17:03:54 {DSK}<XAVIER>TRANSOR.;16

changes to: (VARS TRANSORCOMS)
(FNS PRECH1 TRANSOUT)

previous date: 17-Mar-87 17:00:04 {DSK}<XAVIER>TRANSOR.;15

Read Table: INTERLISP

Package: INTERLISP

Format: XCCS

(RPAQQ **TRANSORCOMS**

```
((FNS TRANSOR TRANSOR-PROCEED TRANSORFORM TRANSORFNS TRANSFORM TRANSIT TRANXT TRANSEXIT KEEPLIST TRANSERR
TRANSOUT PPASS1 TRANSLIST TRANSLIST1 PRETEXT WACHADOON PRECH PRECH1 PRECH2 RETAIL LNC PRESCAN)
TRANSORMACROS TRANSOREMARKS TRANSORGLOBALS (VARS (MAXLOOP 1530)
(TESTRAN)
(USERMACROS (APPEND TRANSORMACROS USERMACROS))
(GLOBALVARS (APPEND TRANSORGLOBALS GLOBALVARS))
(EDITCOMSA (UNION ' (NLAM NLAMIT DOTHESE DOTTHIS XFORMER
CONTINUE)
EDITCOMSA))
(EDITCOMSL (UNION ' (REMARK)
EDITCOMSL))
(TRANSITCONSES ' (ORR NIL XFORMER))
(PRESCARRAY (ARRAY 127 127)))

(INITVARS (NLISTPCOMS)
(LAMBDA (COMS)
(TRANSOUTREADTABLE FILERDTBL))
(DECLARE%: DOEVAL@LOAD DOEVAL@COMPILE DONTCOPY (PROP BLKLIBRARYDEF TAILP))
(PROP FILEGROUP TRANSOR)
(BLOCKS (PRECHBLOCK PRECH PRECH1 PRECH2 RETAIL LNC (ENTRIES PRECH)
(BLKLIBRARY TAILP MEMB LAST LENGTH ASSOC GETP))
(TRANSITBLOCK TRANSIT WACHADOON (ENTRIES TRANSIT WACHADOON)
(GLOBALVARS WACHADID WHENTODOIT TRANSITCONSES LAMBDA (COMS) (NLISTPCOMS))
(BLKLIBRARY TAILP MEMB LAST LENGTH ASSOC GETP))
(TRANXTBLOCK TRANXT (ENTRIES TRANXT)
(BLKLIBRARY TAILP MEMB LAST LENGTH ASSOC GETP))
(NIL PRESCAN (GLOBALVARS PRESCARRAY)))
(DECLARE%: DONTEVAL@LOAD DOEVAL@COMPILE DONTCOPY COMPILEVARS (ADDVARS (NLAMA)
(NLAML TRANSERR KEEPLIST
TRANSOR-PROCEED)
(LAMA)))

(EDITHIST TRANSOR)))
```

(DEFINEQ

(**TRANSOR**

```
[LAMBDA (SOURCEFILE) ; Edited 6-Mar-87 14:36 by DJVB
(RESETFORM (SETREADTABLE FILERDTBL)
(PROG (INPUTFILE OUTPUTFILE LISTFILE LISTING NAMEFIELD EXPRESSION TMP)
(DECLARE%: (SPECVARS LISTFILE LISTING))
(COND
((NLISTP TRANSFORMATIONS)
(ERROR "No transformations loaded." '"" T))
((NULL (SETQ INPUTFILE (INFILEP SOURCEFILE)))
(ERROR "Cannot find file:" SOURCEFILE T))
(SETQ NAMEFIELD (FILENAMEFIELD INPUTFILE 'NAME))
(COND
([NULL (SETQ OUTPUTFILE (OPENSTREAM (SETQ TMP (PACKFILENAME 'NAME NAMEFIELD 'EXTENSION
"TRAN"))
'OUTPUT])
(ERROR "Cannot open file:" TMP T))
([NULL (SETQ LISTFILE (OPENSTREAM (SETQ TMP (PACKFILENAME 'NAME NAMEFIELD 'EXTENSION
"LSTRAN"))
'OUTPUT])
(ERROR "Cannot open file." TMP T))
(* All the preliminary error checks ok.
Open files, print headers.)

(SETQ INPUTFILE (OPENSTREAM INPUTFILE 'INPUT))
(PRINT ' "(PRINT (QUOTE %"
Transoring of" OUTPUTFILE)
(PRINT (FULLNAME INPUTFILE)
OUTPUTFILE)
(PRINT ' " done on " OUTPUTFILE)
(PRINT (DATE)
OUTPUTFILE)
(PRINT ' " %" T) " OUTPUTFILE)
(TERPRI OUTPUTFILE)
(TERPRI OUTPUTFILE)
(PRINT ' " Listing from TRANSORing of file " LISTFILE)
(PRINT (FULLNAME INPUTFILE)
LISTFILE)
(PRINT ' " done on " LISTFILE)
(PRINT (DATE)
LISTFILE)
(TERPRI LISTFILE)
(TERPRI LISTFILE)
```

```

LP (COND
  ([NULL (NLSETQ (PROG NIL
                 (* Rebind HELPCLOCK so that when over-read EOF no error
                 message or BREAK will occur.)
                 (SETQ EXPRESSION (READ INPUTFILE]
                 (GO DONE)))
   [SELECTQ EXPRESSION
    (STOP

```

(* Only check for STOP, no check for NIL. Foreign files never have STOPS on them and rarely have extra parens or NIL's. Extra NIL's on a file usually indicate that the reading machinery is screwed up, probably because user forgot to perform (ESCAPE)%, or, even worse, there is a different ESCAPE character. I therefore must ERRORSET protect the READ above anyway, and try to keep reading until can read no further.)

```

      (GO DONE))
    (COND
      ((NLISTP EXPRESSION)
       (TRANSERR NIL "NLISTP expression on file - expression discarded:" (EXPRESSION))
       (GO LP]
      (SETQ EXPRESSION (TRANSFORM EXPRESSION))
      (TRANSOUT EXPRESSION OUTPUTFILE)
      (GO LP)
    DONE
    (AND (OPENP INPUTFILE)
          (CLOSEF INPUTFILE))
    (ENDFILE OUTPUTFILE)
    (TRANSLIST LISTING LISTFILE)
    (CLOSEF LISTFILE)
    (RETURN (LIST (FULLNAME OUTPUTFILE)
                  (FULLNAME LISTFILE]))

```

(TRANSOR-PROCEED

; Edited 6-Mar-87 14:37 by DJVB

```

[NLAMBDA (FLG)
  (PROG ((L L)
        STOPPEDUP WHERE TOGONEXT CONTINUEL CONTINUETAIL TRANSITL TRANSITAIL OLDLENGTH)
    (DECLARE%: (SPECVARS L STOPPEDUP WHERE TOGONEXT CONTINUEL CONTINUETAIL TRANSITL TRANSITAIL OLDLENGTH))
    LP (COND
      ([ERSETQ (SETQ L
                  (EDITL L (SELECTQ FLG
                            (DOTHIS ' [(IF (TAILP (CAR L)
                                             (CADR L))
                                           ((REMARK TAILP/DOTHIS)
                                            1)
                                           NIL)
                                         MARK
                                         (ORR (NX UP (E (PROG (LISPXHIST)
                                                             (SETQ STOPPEDUP (%##)))
                                                             T))
                                                (!NX UP (E (PROG (LISPXHIST)
                                                             (SETQ STOPPEDUP (%##)))
                                                             T))
                                                NIL)
                                         (LPQ (COMS (TRANSIT)
                                                    (TRANXT])
                                         (DOTHESE ' [MARK (ORR [(IF (NOT (TAILP (CAR L)
                                                                    (CADR L])
                                                                    NX UP (E (PROG (LISPXHIST)
                                                                    (SETQ STOPPEDUP (%##)))
                                                                    T))
                                                                    (!NX UP (E (PROG (LISPXHIST)
                                                                    (SETQ STOPPEDUP (%##)))
                                                                    T))
                                                                    NIL)
                                         _ 1 (LPQ (COMS (TRANSIT)
                                                       (TRANXT])
                                         (TRANSFORM ' [(LPQ (COMS (TRANSIT)
                                                                (TRANXT])
                                         (OKCOMS ' [(LPQ (COMS (TRANXT)
                                                                (TRANSIT])
                                         (HELP]
      (SETQ FLG 'OKCOMS)
      (GO LP))
    (T (TRANSERR TRANSERROR "Fail return to TRANSOR from EDITOR. Show Jim Goodwin." (CURRENTFORM
                                                                                       CURRENTCOMS
                                                                                       ]))

```

(TRANSORFORM

; Edited 6-Mar-87 14:36 by DJVB

```

[LAMBDA (FORM)
  (PROG (LISTFILE LISTING)
    (DECLARE%: (SPECVARS LISTFILE LISTING))
    (SETQ FORM (TRANSFORM FORM))
    (AND LISTING (ERSETQ (TRANSLIST LISTING)))
    in TXTEST.)
    (* ERRORSET so user can abort with ^E, especially when used
    (RETURN FORM])

```

(TRANSORFNS

; Edited 6-Mar-87 14:36 by DJVB

```
[LAMBDA (FNLIST)
  (PROG (LISTING LISTFILE DEF)
    (DECLARE%: (SPECVARS LISTING LISTFILE))
    [MAPC FNLIST (FUNCTION (LAMBDA (FN)
      (COND
        ([AND (LITATOM FN)
          (EXPRP (SETQ DEF (VIRGINFN FN)
            (TRANSFORM DEF FN))
          (T (PRINT (CONS FN ' (NOT FOUND))
            T T])
        (ERSETQ (TRANSLIST LISTING))
        (RETURN FNLIST))])
```

(TRANSFORM

; Edited 6-Mar-87 14:37 by DJVB

(* TRANSFORM is the entry to the translator. It returns the translated SOURCEEXPR, and resets LISTING and uses LISTFILE freely (see KEEPLIST)%.

The source expression is embedded one level so that top-level embeds will work (i.e. the case where the source expression is (FOO --) and the transformation for FOO is MBD)%.

FNAME is provided only by TRANSORFNS. Thus if not provided, SOURCEEXPR is a FORM from TRANSORFORM or TRANSOR's file, and we begin translation at SOURCEEXPR, but if FNAME is given, SOURCEEXPR is a LAMBDA expression and we do a 3 command first, to get to a FORM.

RETAIL also checks this top-level expression. If the top level is (NIL &) it is of no interest to user. But if FNAME was given, top level is (FNAME &) and should be printed, otherwise user will see only a LAMBDA expression and not know where it came from.)

```
(PROG (L PASS1 HELPCLOCK)
  (DECLARE%: (SPECVARS L PASS1))
  [COND
    [FNAME (SETQ L (LIST (CADDR SOURCEEXPR)
      SOURCEEXPR
      (LIST FNAME SOURCEEXPR)
      (T (SETQ L (LIST SOURCEEXPR (LIST NIL SOURCEEXPR)
        (WACHADOON T)
        (TRANSOR-PROCEED TRANSFORM)
        (MAPC (DREVERSE PASS1)
          (FUNCTION PPASS1))
        (RETURN (COND
          (FNAME (CADR L))
          (T (CAR L))
```

(TRANSIT

(* DJVB " 3-Feb-87 13:21")

```
[LAMBDA NIL
  (PROG ((HERE (CAR L))
    TMP)
    (WACHADOON)
    [COND
      ((TAILP HERE (CADR L))
        (TRANSERR TRANSERROR "The function TRANSIT reached a TAILP position; show Jim Goodwin."
          (CURRENTFORM CURRENTCOMS))
        (SETQ L (CONS (SETQ HERE (CAR HERE))
          (CDR L)
          (SETQ CURRENTFORM HERE)
          (SETQ CONTINUEL)
          [SETQ WHERETOGONEXT ' (ORR 2 NX !NX ((E (TRANSEXIT]

        (* The call to TRANSEXIT above causes the exits from PROCEED which occur because of dropoff.)

        (SETQ OLDLENGTH (LENGTH (CADR L)))
        (SETQ TRANSITL L)
        [COND
          ((AND LASTAIL (EQ (CAR LASTAIL)
            HERE))
            (SETQ TRANSITAIL LASTAIL))
          ((OR [NULL (SETQ TRANSITAIL (MEMB HERE (CADR L)
            (MEMB HERE (CDR TRANSITAIL)))
            (HELP ' "The editor lost LASTAIL, and with it its sense of direction." ' "Show Jim Goodwin."])
          [SETQ CURRENTCOMS (COND
            ([AND STOPPEDUP (EQ HERE (CAR STOPPEDUP))
              (OR (LISTP HERE)
                (EQ STOPPEDUP (%## UP] (* Exit on match with STOPPEDUP.)
              (RETFROM ' TRANSOR-PROCEED))
            ((NLISTP HERE)
              NLISTPCOMS)
            ((LITATOM (CAR HERE))
```

(* If user commands cause an error it will be trapped by the ORR and XFORMER will be executed. XFORMER is a TRANSORMACRO which makes a remark on the error. Don't make the list if no commands.)

```
(GETP (CAR HERE)
```

```

      'XFORM))
      ((LISTP (CAR HERE))
       LAMBDA COMS)
      (T ' ((COMSQ (REMARK ILLCAR)
                  DOTHESE]
(RETURN (COND
         (CURRENTCOMS (FRPLACA (CDR TRANSITCONSES)
                               CURRENTCOMS)

```

(* If CURRENTCOMS is NIL, return NIL; otherwise effectively embed CURRENTCOMS in orr such that if CURRENTCOMS fail, xformer will be executed. Xformer is a transormacro which calls TRANSERR appropriately for a faulty transformation.)

```
TRANSITCONSES])
```

(TRANXT

```

[LAMBDA NIL
 (PROG (NEWFORM NEWLENGTH TMP NEWTAIL)
 (COND
  (CONTINUEL (SETQ L CONTINUEL)
             (SETQ LASTAIL CONTINUETAIL)
             (RETURN)))
 (SETQ NEWFORM (CAR TRANSITAIL))
 (SETQ NEWLENGTH (LENGTH (CADR TRANSITL)))
 (COND
  ((NEQ NEWLENGTH OLDLENGTH)
   (GO DELETED))
  ((OR (EQ NEWFORM CURRENTFORM)
        (EQ WHERETOGONEXT 'NLAMIT))

```

(* If containing list still points at same EQ structure, or if he's declared he's done with whatever is there, no problem.)

```

)
((AND (LISTP NEWFORM)
      (EDITFINDP NEWFORM (SETQ TMP (CONS '== CURRENTFORM)
                                         T)))

```

(* Net effect was an MBD. If we went on from here in normal fashion, we would embed it again and again. Find original expression and go on from there.)

```

 (SETQ WHERETOGONEXT (CONS TMP WHERETOGONEXT)))
 ((AND (LISTP CURRENTFORM)
      (EDITFINDP CURRENTFORM (CONS '== NEWFORM)
                              T))

```

(* Net effect was XTR. If we went on normally, we'd miss the form extracted. Set continuation commands to NIL so we stay where we are)

```

 (SETQ WHERETOGONEXT))
 (T

```

(* User did a %: or DELETE where effect was (%: NIL)%. Assume the stuff he put in place of old does not need translation. perform NLAM for him.)

```
(SETQQ WHERETOGONEXT NLAM))
```

(* Ready to return. Fix up L, smashing CAR to point to right thing. In normal case, (EQ CURRENTFORM NEWFORM)%, it already does and this FRPLACA is a NOP.)

```

ZIPPO
 (SETQ L (FRPLACA TRANSITL NEWFORM))
 (SETQ LASTAIL TRANSITAIL)

 (RETURN WHERETOGONEXT)
DELETED
 (COND
  ((NEQ (SUB1 OLDLENGTH)
        NEWLENGTH)

```

(* Finally, return the commands which will locate the next form to translate.)

(* This could happen if user cheated by doing a !0 and deleting several things or inserting things.)

```

 (TRANSERR OUTOFBOUNDS "Your transformations cheated and changed something out of bounds."
 (CURRENTFORM CURRENTCOMS)))
 [(NOT (TAILP TRANSITAIL (CADR L)))
 (RETURN ' (ORR !NX ((E (TRANSEXIT]
 (T

```

(* User deleted the form, but it was the last form on its containing list.)

(* User deleted the form, but was not last thing, thus NEWFORM is already bound to the NX thing after the one we just %'translated' by deleting it. Set WHERETOGONEXT to NIL so we do not move at all.)

```

 (SETQ WHERETOGONEXT)
 (GO ZIPPO))

```

(TRANSEXIT [LAMBDA NIL

(* DJVB " 3-Feb-87 13:22")

```
[COND
  (STOPPEDUP (TRANSERR TRANSERROR "The function TRANSEXIT missed the exit. Show Jim Goodwin." (CURRENTFORM
    CURRENTCOMS
  ]
  (RETFROM 'TRANSOR-PROCEED])
```

(KEEPLIST

```
[NLAMBDA (REMNAME)
  (PROG (TMP)
    [COND
      ((NLISTP LISTING) (* Initialize if first remark in this LISTING.)
        (SETQ LISTING (LIST 1)))
      (T (FRPLACA LISTING (ADD1 (CAR LISTING))
        (SETQ PASS1 (CONS (CONS (CAR LISTING)
          (CONS REMNAME L))
          PASS1)) (* Save pass2 stuff for TRANSLIST.)
        [COND
          (TESTRAN (* Skip pass2 if testing)
            NIL)
          ([NULL (SETQ TMP (FASSOC REMNAME (CDR LISTING))
            (NCONC1 LISTING (LIST REMNAME (CAR LISTING))
            (T (NCONC1 TMP (CAR LISTING))
            (RETURN])) (* First use of this remark.)
```

(TRANSERR

```
[NLAMBDA (REM MESS VARS)
  (AND REM (APPLY (FUNCTION KEEPLIST)
    (LIST REM)))
  [AND MESS (NLSETQ (PROGN (TERPRI T)
    (TERPRI T)
    (PRIN1 ' "
      TRANSOR made a translation error: " T)
    (PRIN1 MESS T)
    (TERPRI T]
  (AND VARS (NLSETQ (PROGN (PRINTLEVEL 3)
    [MAPC VARS (FUNCTION (LAMBDA (X)
      (PRIN2 X T T)
      (PRIN1 '": " T)
      (PRINT (EVALV X)
      (PRINTLEVEL 1000)
      (TERPRI T])
```

(TRANSOUT

```
[LAMBDA (XPR FILE) ; Edited 17-Mar-87 16:37 by DJVB
  (RESETFORM (OUTPUT FILE)
    (RESETFORM (SETREADTABLE TRANSOUTREADTABLE) (* XPR is a transored form which is to be put on the output file.)
    (COND
      ((EQ FILE 'NIL%:))
      ((EQ 'DEFINEQ (CAR XPR)) (* Special formatting for function lists.)
        (PRIN1 ' "(DEFINEQ")
        (TERPRI)
        [MAPC (CDR XPR)
          (FUNCTION (LAMBDA (X)
            (TERPRI)
            (PRIN1 ' %()
            (PRINT (CAR X))
            (PRINTDEF (CADR X)
            NIL T)
            (PRIN1 ' %))
            (TERPRI]
        (PRIN1 ' %))
        (TERPRI))
      [(AND (EQ 'PROGN (CAR XPR))
        (EQ 'DEFUN (CAADR XPR))) (* Special formatting for commonlisp function lists.)
        (MAPC (CDR XPR)
          (FUNCTION (LAMBDA (X)
            (TERPRI)
            (COND
              ((EQ (CAR X)
                'DEFUN)
                (PRIN1 " (DEFUN "
                (AND LAMBDAFONT FONTCHANGEFLG (CHANGEFONT LAMBDAFONT))
                (PRIN2 (CADR X))
                (AND LAMBDAFONT FONTCHANGEFLG (CHANGEFONT DEFAULTFONT))
                (PRIN2 (CADDR X))
                (PRINTDEF (CDDDR X)
                6 T T)
                (PRIN1 " )"))
                (T (PRINTDEF XPR)))
            (TERPRI]
        (T (TERPRI)
          (PRINTDEF XPR)
          (TERPRI])
```

(PPASS1

```
[LAMBDA (P1)
  (PRIN1 (CAR P1)
    LISTFILE)
  (PRIN1 ' " " LISTFILE)
  (PRIN1 (CADR P1)
    LISTFILE)
  (PRIN1 ' " at " LISTFILE)
  (PRECH (CDDR P1)
    NIL LISTFILE T)
  (TERPRI LISTFILE)]
```

(TRANSLIST

```
[LAMBDA (LISTING LISTFILE)
  (PROG (OLDO)
    (COND
      (TESTRAN
        (RETURN))
      ((EQ LISTFILE 'NIL%:)
        (RETURN)))
    (SETQ OLDO (OUTPUT LISTFILE))
    [COND
      ((NULL LISTING)
```

(* TRANSLIST must dump the second half of the listing prettily.)

(* See TXTEST.)

(* See KEEPLIST for discussion of format of LISTING.)

(* User would like to know if this happens rather than just wondering where his output went.)

```
(PRIN1 ' "
      No REMARKS -- empty listing.
  ")
(T (PRIN1 ' "
```

Index of Remarks

```

      ")
      (MAPC (SORT (CDR LISTING)
                 T)
            (FUNCTION TRANSLIST1])
      (TERPRI)
      (OUTPUT OLDO)
      (RETURN])

```

(TRANSLIST1

```

[LAMBDA (L1)
  (PRIN1 (CAR L1))
  (PRIN1 ' " at ")
  (MAPRINT (CDR L1)
           NIL NIL ' ".
           " ' " " ")
  (PREMTEXT (CAR L1))
  (TERPRI])

```

(* Name of remark.)

(PREMTEXT

```

[LAMBDA (RNAME)
  (PROG (TXT)
    [COND
      ((OR (SETQ TXT (ASSOC RNAME USERNOTES))
           (SETQ TXT (ASSOC RNAME TRANSOREMARKS)))
        (SETQ TXT (CADR TXT)))
      (T (SETQQ TXT

      (SPACES 5)
      [COND
        ((EQ (CADR TXT)
             ' %%)

```

(* The text of this remark was not defined in the TRANSFORMATIONS file.)]

(* Lower-case the comment before using it, if he is testing and it hasn't been dumped before.)

```

      (RPLACD TXT (COMMENT3 (CDDR TXT)
                           NIL T])
      (MAPRINT (CDR TXT))
      (TERPRI])

```

(WACHADOON

```

[LAMBDA (FLG)
  (OR TESTRAN (PROG ((NOW (CLOCK)))
                    (COND
                      (FLG (SETQ WACHADID)
                           (SETQ WHENTODOIT NOW)
                           (RETURN))
                      ((ILESSP NOW WHENTODOIT)
                       (RETURN)))
                    (PRECH L WACHADID T)
                    (SETQ WACHADID L)
                    (SETQ WHENTODOIT (IPLUS 180000 NOW])

```

(PRECH

```

[LAMBDA (ECH OLDECH FILE PRTYFLG)
  (PROG ((OLDO (OUTPUT FILE))
        X)
    [SETQ X (PRECH1 (RETAIL (COND
                            (OLDECH (LNC ECH OLDECH))
                            (T ECH]
        (COND
          (PRTYFLG (PRINTDEF X))
          (T (PRINT X)))
        (TERPRI)
        (OUTPUT OLDO)
        (RETURN])

```

(* Function to Print a Reversed Edit CHain in my special format.)

(PRECH1

```

[LAMBDA (RECH)
  (PROG (LASTALE (N -2)
        LST)
    [COND
      ((NULL (CDR RECH))
       (RETURN (MKSTRING (PRECH2 (CAR RECH)
                                  4]
                         (SETQ LASTALE (SOME (CAR RECH)
                                             (FUNCTION (LAMBDA (E)

```

; Edited 17-Mar-87 14:24 by DJVB

```

                                (ADD1VAR N)
                                (EQ E (CADR RECH])
(AND (MINUSP N)
      (GO OUT))
(SETQ LST (CONS (COND
                 ((NLISTP (CAAR RECH))
                  (CAAR RECH))
                 (T (PRECH2 (CAAR RECH)
                             3)))
                LST))
[SELECTQ N
 (0)
 (1 (SETQ LST (CONS (COND
                    ((NLISTP (CADAR RECH))
                     (CADAR RECH))
                    (T '&))
                    LST)))
(COND
 [ (AND (EQ 'DEFUN (CAAR RECH))
        (LITATOM (CADAR RECH)))
  (SETQ LST (CONS [COND
                  [(EQ N 2)
                   (COND
                    ((NLISTP (CADDAR RECH))
                     (CADDAR RECH))
                    (T '&)]
                  (T (PACK* ' |...| (SUB1 N)
                       ' |...|]
                   (CONS (CADAR RECH)
                          LST]
                  (T (SETQ LST (CONS (PACK* ' |...| N ' |...|)
                                     LST]
                  LST))
  OUT (SETQ LST (CONS (PRECH1 (CDR RECH))
                      LST))
[COND
 ((CDR LASTALE)
  (SETQ LST (CONS '-- LST]
  (RETURN (DREVERSE LST])

```

(* DJVB put in for TO-COMMONLISP to identify DEFUNs)

PRECH2

```

[LAMBDA (X LEVEL)
 (COND
 ((NLISTP X)
  X)
 ((EQ (CAR X)
      COMMENTFLG)
  '***COMMENT***)
 ((ILESSP LEVEL 1)
  '&))
 (T (MAPCAR X [FUNCTION (LAMBDA (XELT)
                        (SUB1VAR LEVEL)
                        (COND
                         ((MINUSP LEVEL)
                          '--)
                         (T (PRECH2 XELT LEVEL])
                        (FUNCTION (LAMBDA (TAIL)

```

(* At last!!!! I get to use the second functional argument to a mapping function. To implement a triangular PRINTLEVEL, step the LEVEL down in the first function and select hyphens when it hits bottom; cut off the rest of the MAP by checking for bottom here.)

```

(AND (NULL (MINUSP LEVEL))
      (CDR TAIL])

```

RETAIL

```

[LAMBDA (L)
 (DECLARE%: (SPECVARS L))
 (PROG (RES)
  [SETQ RES (LIST (COND
                  ((TAILP (CAR L)
                           (CADR L))
                   (CAAR L))
                  (T (CAR L]
  [MAP (CDR L)
   (FUNCTION (LAMBDA (TAIL)
              (COND
               [(NULL (CDR TAIL))

```

; Edited 6-Mar-87 14:39 by DJVB

(* At end. If top-most expression is (NIL &) don't include it. Otherwise is from TRANSORFNS, so include it. See TRANSFORM.)

```

(AND (CAAR TAIL)
      (SETQ RES (CONS (CAR TAIL)
                       RES]
      (MEMB (CAR TAIL)

```


(CADR TAIL))

(* If not a TAIL, must be MEMB, otherwise edit chain screwed up. We want every one that's MEMB.)

(SETQ RES (CONS (CAR TAIL) RES])

(RETURN RES])

(LNC

[LAMBDA (L1 L2)

(* LNC is for Last New Cons. Returns last tail of L1 such that it is not common with L2. L1 is the edit chain representing TRANSOR's current location; L2 is the chain from the last call to WACHADOON. Value is (LAST L1) if nothing in common, i.e. we are transoring an entirely different source expression.)

(PROG (X) (COND

((NLISTP L1) (HELP)) ((NEQ (SETQ X (LAST L1)) (LAST L2)) source expression.) (RETURN X)))

(* Quick check for commonest case, we are in a totally different

LP (COND ((TAILP (CDR L1) L2) (RETURN L1))) (SETQ L1 (CDR L1)) (GO LP])

(PRESCAN

[LAMBDA (FILE CHARLST PRESCANFN)

(* DJVB "22-NOV-83 21:41")

(* FIX UP TO ALLOW NULL'S (ZERO'S) IN CHARLST. AT PRESENT JUST FILTERS %'EM ALL OUT, IF YOU PUT 0 IN CHARLST GIVES ILLEGAL SETA.)

(* PRESCAN is for pre-digesting files from alien environments where special characters, etc., are all different.

- FILE is input file; output goes to next higher version. - CHARLST is list of dot-pairs of character codes (old . new)%, so that you can for example replace all tabs in a file with spaces by including (9 . 32) on CHARLST. - PRESCANFN is function for user. If the new character code for any character is NIL, then PRESCANFN is called giving the character code as its first argument. PRESCANFN can then do what it needs to process the upcoming file information. The second argument to PRESCANFN is the input file, and the third is the output file. -

Original impetus for this was MIT Lisp's special recognition of semicolon%: any line beginning with semicolon was comment, a la macro files. With (59) on CHARLST, where 59 is character code for semicolon, PRESCANFN can process those lines, making them into regular comments. Note that no output is done for these special characters unless PRESCANFN does it.)

(PROG ((INF (INPUT (INFILE FILE))) [OUTF (OUTPUT (OUTFILE (NAMEFIELD FILE T) (I 127))

TOP (COND ((NOT (ZEROP I)) (SETA PRESCARRAY I I) (SUB1VAR I) (GO TOP)))

[MAPC CHARLST (FUNCTION (LAMBDA (PR) (SETA PRESCARRAY (CAR PR) (OR (CDR PR) 0]

(SELECTQ (SYSTEMTYPE) ((TENEX TOPS-20) (D-ASSEM::ASSEMBLE NIL (CQ INF) (FASTCALL IFSET) (HRRZI 1 %, FCHAR (FX)) (PUSHN) (HRRZ 1 %, FILEN (FX)) (PUSHN) (VARS (HRRZ 2 %, OUTF)) (FASTCALL IFSET) (HRRZ 1 %, FILEN (FX)) (PUSHN) (CQ PRESCARRAY) (SKIP 4 (TIMES %, 1)) (XWD 2 1) (add 4 %, 1)

(* Store ptr to single-character buffer for input file on -2<np>.)

(* Store input jfn on -1<np>.)

(* Store output jfn on 0<np>.)

(* Ac4 now has PRESCARRAY<2> i.e. indirect ref thru 4 will get Nth element of PRESCARRAY, where N is in ac2. Note ac4 must be saved on CP since LH is bits, RH is ptr.)

LOOP (NREF (MOVE 1 %, -1)) (JSYS 40)

(* BIN)

```

(JUMPE 2 %, DONE)
MIDDLE
(SKIPG 0 %, @ 4)
(JRST SPECIAL)
(MOVE 2 %, @ 4)
LOUT
(NREF (MOVE 1 %, 0))
(JSYS 41) (* BOUT)
(JRST LOOP)
DONE
(JSYS 20) (* GTSTS)
(TLNE 2 %, 512)
(JRST DONE!) (* Filter NULL's.)
(JRST LOOP)
SPECIAL
(MOVE 1 %, 2)
(CLISP% (SETQ CP (CONS %, (CONS 4 CP)))
  push CP %, 4)
(CQ (SETQ I (LOC (AC)))
  (APPLY* PRESCANFN I INF OUTF))
(CLISP% (PROG1 (CAR CP)
  (SETQ CP (CDR CP)))
  pop CP %, 4)
(NREF (HRRZ 2 %, @ -2)) (* If single-char buff. empty,)
(JUMPE 2 %, LOOP) (* Then next char. must be read from file,)
(NREF (HLRM 2 %, @ -2))

```

(* Else clear buff. to prevent next call to PRESCANFN from seeing it with READC or whatever,)

```

(JRST MIDDLE) (* And be sure the char from buff gets matched and output.)
DONE!
(POPNN 3))
(VAX (SETQ INF (\GETOFD INF 'INPUT))
  (SETQ OUTF (\GETOFD OUTF 'OUTPUT))
  [CLISP% (PROG ($$VAL AC AT)
    $$LP
    (COND
      ((\EOFP INF)
        (RETURN $$VAL))
      ((ZEROP (SETQ AC (BIN INF)))
        (GO $$ITERATE)))
    (COND
      ((SETQ AT (ELT PRESCARRAY AC))
        (BOUT OUTF AT))
      (T (APPLY* PRESCANFN AC INF OUTF)))
    $$ITERATE
    (GO $$LP))
  until
  (\EOFP INF)
  bind AC AT unless (ZEROP (SETQ AC (BIN INF)))
  do
  (COND
    ((SETQ AT (ELT PRESCARRAY AC))
      (BOUT OUTF AT))
    (T (APPLY* PRESCANFN AC INF OUTF]))
  (D (SETQ INF (GETSTREAM INF 'INPUT))
    (SETQ OUTF (GETSTREAM OUTF 'OUTPUT))
    [CLISP% (PROG ($$VAL AC AT)
      $$LP
      (COND
        ((EOFP INF)
          (RETURN $$VAL))
        ((ZEROP (SETQ AC (BIN INF)))
          (GO $$ITERATE)))
        (COND
          ((SETQ AT (ELT PRESCARRAY AC))
            (BOUT OUTF AT))
          (T (APPLY* PRESCANFN AC INF OUTF)))
        $$ITERATE
        (GO $$LP))
      until
      (EOFP INF)
      bind AC AT unless (ZEROP (SETQ AC (BIN INF)))
      do
      (COND
        ((SETQ AT (ELT PRESCARRAY AC))
          (BOUT OUTF AT))
        (T (APPLY* PRESCANFN AC INF OUTF]))
      (HELP))
    (CLOSEF INF)
    (RETURN (CLOSEF OUTF))

```

)

(RPAQQ TRANSORMACROS

```

((REMARK (TXT)
  (E (KEEPLIST TXT)

```

```

      T))
(NLAM NIL (E (SETQ WHERETOGONEXT NLAMIT)
             T))
[NLAMIT NIL (ORR NX !NX ((E (TRANSEXIT)
                             DO THESE NIL (E (TRANSOR-PROCEED DO THESE)
                                             T)
                                             NLAM)
                             DOTHIS NIL (E (TRANSOR-PROCEED DOTHIS)
                                             T)
                                             NLAM)
                             XFORMER NIL (E (TRANSERR TRANSFORMATIONERROR "FAULTY TRANSFORMATION" (CURRENTFORM CURRENTCOMS)
                                             T))))

```

(RPAQQ TRANSOREMARKS

```

((TRANSFORMATIONERROR (* The TRANSFORMATIONS specified for this form failed to work properly. The TTY
                        message %'FAULTY TRANSFORMATION' was printed, any commands remaining in the
                        transformation after the erroneous one were skipped, and translation continued
                        as if the transformation had been normally completed. The user should treat the
                        translated form with caution and amend his transformation to avoid future
                        problems.))
(TRANSERR (* TRANSOR got confused at this point. The TTY message %'SHOW JIM GOODWIN' was printed and
            translation continued with the next form, but the user should treat the compromised area
            of code with caution.))
(BLAMBDA1 (* Non-atomic CAR of form, but not an open lambda. Either a parenthesis error or computed CAR
            of form. Computed CAR of form is no longer legal in BBN-LISP; APPLY* is used instead. If
            computed CAR of form was intended, the translation to APPLY* will run ok. See manual for
            discussion of APPLY*.)
(BLAMBDA2 (* Open LAMBDA with wrong number of args. What can it mean?))
(BLAMBDA3 (* Lambda-expression without forms. What can it mean?))
(ILLCAR (* Illegal data-type encountered as CAR of form Expression treated as list of forms.))
(TAILP/DOTHIS (* When the transormacro DOTHIS is executed at a TAILP position, TRANSOR does a 1 command
               first, assuming that the current position is a list of forms and CAR of it is the form
               intended. The user should make sure that this is what was intended by the
               TRANSFORMATIONS which called DOTHIS, i.e. the TRANSFORMATIONS for the form containing
               this one.))))

```

(RPAQQ TRANSORGLOBALS (USERNOTES USERNOTES TESTFORM TESTFORM TRANSFORMATIONS TRANSFORMATIONS XFORMSFNS XFORMSVARS XFORMSVARS DUMPFIL TRANSFORMATIONS TRANSFORMATIONS TRANSFORMATIONS TRANSFORMATIONS))

(RPAQQ MAXLOOP 1530)

(RPAQQ TESTRAN NIL)

(RPAQ USERMACROS (APPEND TRANSORMACROS USERMACROS))

(RPAQ GLOBALVARS (APPEND TRANSORGLOBALS GLOBALVARS))

(RPAQ EDITCOMSA (UNION '(NLAM NLAMIT DO THESE DOTHIS XFORMER CONTINUE) EDITCOMSA))

(RPAQ EDITCOMSL (UNION '(REMARK) EDITCOMSL))

(RPAQQ TRANSITCONSES (ORR NIL XFORMER))

(RPAQ PRESCARRAY (ARRAY 127 127))

(RPAQ? NLISTPCOMS)

(RPAQ? LAMBDACOMS)

(RPAQ? TRANSOUTREADTABLE FILERDTBL)

(DECLARE%: DOEVAL@LOAD DOEVAL@COMPILE DONTCOPY

```

(PUTPROPS TAILP BLKLIBRARYDEF [LAMBDA (.BLKVAR.X .BLKVAR.Y)
                              (* True if .BLKVAR.X is A tail of .BLKVAR.Y .BLKVAR.X and .BLKVAR.Y
                              non-null.)
                              (* Included with editor for block compilation purposes.)
                              (AND .BLKVAR.X (PROG NIL LP (COND ((NLISTP .BLKVAR.Y)
                                                                (RETURN NIL))
                                                                ((EQ .BLKVAR.X .BLKVAR.Y)
                                                                (RETURN .BLKVAR.X))))
                              (SETQ .BLKVAR.Y (CDR .BLKVAR.Y))
                              (GO LP])
)

```

(PUTPROPS TRANSOR FILEGROUP (TRANSOR TSET))

(DECLARE%: DONTVAL@LOAD DOEVAL@COMPILE DONTCOPY

(BLOCK%: PRECHBLOCK PRECH PRECH1 PRECH2 RETAIL LNC (ENTRIES PRECH) (BLKLIBRARY TAILP MEMB LAST LENGTH ASSOC GETP))

(BLOCK%: TRANSITBLOCK TRANSIT WACHADOON (ENTRIES TRANSIT WACHADOON) (GLOBALVARS WACHADID WHENTODOIT TRANSITCONSES LAMBDACOMS NLISTPCOMS))

```
(BLKLIBRARY TAILP MEMB LAST LENGTH ASSOC GETP))

(BLOCK%: TRANXTBLOCK TRANXT (ENTRIES TRANXT)
 (BLKLIBRARY TAILP MEMB LAST LENGTH ASSOC GETP))

(BLOCK%: NIL PRESCAN (GLOBALVARS PRESCARRAY))
)

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

(ADDTOVAR NLAMA )

(ADDTOVAR NLAML TRANSERR KEEPLIST TRANSOR-PROCEED)

(ADDTOVAR LAMA )
)

(DECLARE%: DONTCOPY

(ADDTOVAR EDITHISTALIST
 (TRANSOR (" 5-Feb-87 16:18:06" DJVB {DSK}<XAVIER>TRANSOR.;11 (TRANSOR)
 (FIXED TO WORK WITH NEW FILE RULES IN LYRIC))
 (" 6-Feb-87 15:24:20" DJVB {DSK}<XAVIER>TRANSOR.;12 (TRANSOR))
 (" 6-Mar-87 14:41:26" DJVB {DSK}<XAVIER>TRANSOR.;13 (TRANSOR TRANSOR-PROCEED TRANSORFORM
 TRANSORFNS TRANSFORM RETAIL))
 ("17-Mar-87 17:01:53" DJVB {DSK}<XAVIER>TRANSOR.;15 (PRECH1 TRANSOUT)
 (ADDED SPLIT READ/WRITE READTABLES AND PP FOR DEFUN))))
)

(PUTPROPS TRANSOR COPYRIGHT (NONE))
```

FUNCTION INDEX

KEEPLIST	5	PRECH2	8	TRANSEXIT	4	TRANSOR	1	TRANXT	4
LNC	9	PREMTEXT	7	TRANSFORM	3	TRANSOR-PROCEED	2	WACHADOON	7
PPASS1	6	PRESCAN	9	TRANSIT	3	TRANSORFNS	3		
PRECH	7	RETAIL	8	TRANSLIST	6	TRANSORFORM	2		
PRECH1	7	TRANSERR	5	TRANSLIST1	7	TRANSOUT	5		

VARIABLE INDEX

EDITCOMSA	11	LAMBDAOMS	11	PRESCARRAY	11	TRANSOREMARKS	11	TRANSOUTREADTABLE	11
EDITCOMSL	11	MAXLOOP	11	TESTRAN	11	TRANSORGLOBALS	11	USERMACROS	11
EDITHISTALIST	12	NLISTPCOMS	11	TRANSITCONSES	11	TRANSORMACROS	10		

PROPERTY INDEX

TAILP	11	TRANSOR	11
-------------	----	---------------	----
