```
    File created:   13-Jun-2024 11:46:26  {WMEDLEY}<lispusers>COMPARESOURCES.;138

         edit by:   rmk

      changes to:   (FNS \CS.COMPARE.MASTERS)

   previous date:    7-Feb-2024 16:08:54  {WMEDLEY}<lispusers>COMPARESOURCES.;137

      Read Table:   INTERLISP

         Package:   INTERLISP

          Format:    XCCS


(RPAQQ COMPARESOURCESCOMS
       ((FNS COMPARESOURCES \CS.COMPARE.MASTERS \CS.COMPARE.TYPES \CS.EXAMINE \CS.FIXFNS \CS.SORT.DECLARES
             \CS.SORT.DECLARE1 \CS.FILTER.GARBAGE)
        (FNS \CS.ISFNFORM \CS.COMPARE.FNS \CS.FNSID \CS.ISVARFORM \CS.COMPARE.VARS \CS.ISMACROFORM \CS.ISRECFORM
             \CS.REC.NAME \CS.ISCOURIERFORM \CS.ISTEMPLATEFORM \CS.COMPARE.TEMPLATES \CS.ISPROPFORM
             \CS.PROP.NAME \CS.COMPARE.PROPS \CS.ISADDVARFORM \CS.COMPARE.ADDVARS \CS.ISFPKGCOMFORM
             \CS.COMPARE.FPKGCOMS \CS.COMPARE.DEFINE-FILE-INFO \CS.IGNOREFORMS)
        [COMS (FNS CSOBJ.CREATE CSOBJ.DISPLAYFN CSOBJ.IMAGEBOXFN CSOBJ.BUTTONEVENTINFN CSOBJ.COPYBUTTONEVENTINFN
                   )
              (INITVARS (COMPARESOURCES-IMAGEFNS (IMAGEFNSCREATE 'CSOBJ.DISPLAYFN 'CSOBJ.IMAGEBOXFN NIL NIL NIL
                                                        'CSOBJ.BUTTONEVENTINFN
                                                        'CSOBJ.COPYBUTTONEVENTINFN]
        (VARS COMPARESOURCETYPES DEFAULT.DECLARE.TAGS)
        (COMS (FNS CSBROWSER)
              (FILES (SYSLOAD)
                     OBJECTWINDOW EXAMINEDEFS REGIONMANAGER))
        (DECLARE%: EVAL@COMPILE DONTCOPY (RECORDS CSTYPE)
               (GLOBALVARS COMPARESOURCETYPES CLISPRECORDTYPES MACROPROPS DEFAULT.DECLARE.TAGS))))

(DEFINEQ

(COMPARESOURCES
  [LAMBDA (FILEX FILEY EXAMINE DW? LISTSTREAM IGNOREFORMS LABEL1 LABEL2)
                                                                   ; Edited  7-Feb-2024 16:08 by rmk
                                                                   ; Edited 22-May-2022 18:45 by rmk

;;; Compare two lisp source files, reporting differences.  LISTSTREAM if given is an open stream, or an object window

    (DECLARE (SPECVARS FILEX FILEY EXAMINE DIFFERENCES))
    (PROG (DIFFERENCES BODYX BODYY ENVX ENVY DECLAREX DECLAREY DATECOL (INSERTOBJECTS (AND EXAMINE (OBJWINDOWP
                                                                                                    LISTSTREAM))
                                                                                      )
                 (COMPARESTREAM LISTSTREAM)
                 (CONTEXTSTREAM LISTSTREAM)
                 OBJECTS)
          (DECLARE (SPECVARS INSERTOBJECTS OBJECTABLE))
          (CL:WHEN INSERTOBJECTS
              (SETQ COMPARESTREAM (CL:MAKE-STRING-OUTPUT-STREAM))
              (SETQ CONTEXTSTREAM (CL:MAKE-STRING-OUTPUT-STREAM))
              (LINELENGTH 65535 COMPARESTREAM)                     ; Let the receiver do the wrapping
              (LINELENGTH 65535 CONTEXTSTREAM))
          (CL:UNLESS (OR (STREAMP FILEX)
                         (INFILEP FILEX)
                         (SETQ FILEX (FINDFILE FILEX T)))
              (RETURN (printout CONTEXTSTREAM FILEX " not found" T)))
          (CL:UNLESS (OR (STREAMP FILEY)
                         (INFILEP FILEY)
                         (SETQ FILEY (FINDFILE FILEY T)))
              (RETURN (printout CONTEXTSTREAM FILEY " not found" T)))

     ;; Read the two files, throwing out extraneous forms & such:

          (CL:MULTIPLE-VALUE-SETQ (BODYX ENVX)
                 (READFILE FILEX))
          (SETQ BODYX (\CS.FILTER.GARBAGE BODYX))
          (CL:MULTIPLE-VALUE-SETQ (BODYY ENVY)
                 (READFILE FILEY))
          (SETQ BODYY (\CS.FILTER.GARBAGE BODYY))
          (CL:WHEN (EQMEMB T IGNOREFORMS)                          ; Extra tests to recognize killable comments
              (SETQ IGNOREFORMS (REMOVE T IGNOREFORMS))
              (LET ((*REMOVE-INTERLISP-COMMENTS* T))
                   (DECLARE (SPECVARS *REMOVE-INTERLISP-COMMENTS*))
                   (SETQ BODYX (REMOVE-COMMENTS BODYX))
                   (SETQ BODYY (REMOVE-COMMENTS BODYY))))
          (CL:WHEN IGNOREFORMS
              (SETQ BODYX (\CS.IGNOREFORMS BODYX IGNOREFORMS))
              (SETQ BODYY (\CS.IGNOREFORMS BODYY IGNOREFORMS)))
          (CL:UNLESS LABEL1 (SETQ LABEL1 FILEX))
          (CL:UNLESS LABEL2 (SETQ LABEL2 FILEY))
          [SETQ DATECOL (PLUS 2 (CONSTANT (NCHARS "Comparing"))
                              (IMAX (NCHARS LABEL1)
                                    (NCHARS LABEL2]
          (printout CONTEXTSTREAM "Comparing " LABEL1 .TAB0 DATECOL "dated " (GETFILEINFO FILEX 'CREATIONDATE)
                 .TAB
                 [SUB1 (CONSTANT (IDIFFERENCE (NCHARS "Comparing ")
                                              (NCHARS "and "]
                  " and " LABEL2 .TAB0 DATECOL "dated " (GETFILEINFO FILEY 'CREATIONDATE)
                  T T)
            [SETQ DECLAREX (for EXPR in BODYX collect EXPR when (EQ (CAR EXPR)
                                                                    'DECLARE%:]
            (SETQ BODYX (CL:SET-DIFFERENCE BODYX DECLAREX))
            [SETQ DECLAREY (for EXPR in BODYY collect EXPR when (EQ (CAR EXPR)
                                                                    'DECLARE%:]
            (SETQ BODYY (CL:SET-DIFFERENCE BODYY DECLAREY))
            (WITH-READER-ENVIRONMENT (OR ENVX ENVY (MAKE-READER-ENVIRONMENT))
                (\CS.COMPARE.MASTERS BODYX BODYY DW?)

                ;; Done with the non-DECLARE: expressions.  Nw sort what's left according to when it is eval'ed so that we can hopefully further
                ;; reduce the amount of stuff to compare

                (SETQ BODYX (\CS.SORT.DECLARES DECLAREX))
                (SETQ BODYY (\CS.SORT.DECLARES DECLAREY))
                [SETQ BODYX (APPEND BODYX (for Y in BODYY collect (LIST (CAR Y)) unless (SASSOC (CAR Y)
                                                                                               BODYX]
                                                                   ; Add placeholders for any declaration types in Y not in X to
                                                                   ; simplify what follows
                (for X in BODYX bind Y TYPE do (SETQ Y (SASSOC (CAR X)
                                                               BODYY))
                                               (SETQ TYPE (CAR X))
                                               (SETQ X (CL:SET-DIFFERENCE (CDR X)
                                                               (PROG1 (CDR Y)
                                                                   (SETQ Y (CL:SET-DIFFERENCE (CDR Y)
                                                                                  X :TEST (FUNCTION EQUALALL))))
                                                               :TEST
                                                               (FUNCTION EQUALALL)))
                                               (CL:WHEN (OR X Y)
                                                   (printout CONTEXTSTREAM T "------" [CONS 'DECLARE%:
                                                                                            (APPEND (
                                                                                                   CL:SET-DIFFERENCE
                                                                                                     TYPE
                                                                                                DEFAULT.DECLARE.TAGS
                                                                                                     )
                                                                                                   '(--]
                                                          " forms------" T)
                                                                   ; REVERSE because \CS.SORT.DECLARES delivered
                                                                   ; expressions in reverse order
                                                   (\CS.COMPARE.MASTERS (REVERSE X)
                                                          (REVERSE Y)
                                                          DW?)))
                (TERPRI CONTEXTSTREAM))
            (CL:WHEN INSERTOBJECTS
                (CL:UNLESS (EQ 0 (GETFILEPTR CONTEXTSTREAM))
                    (PUSH OBJECTS (CSOBJ.CREATE (CL:GET-OUTPUT-STREAM-STRING CONTEXTSTREAM))))
                (SETQ OBJECTS (DREVERSE OBJECTS))
                (OBJ.ADDMANYTOW LISTSTREAM OBJECTS))
            (RETURN (OR (REVERSE DIFFERENCES)
                        'SAME])
```

(\CS.COMPARE.MASTERS

```
  [LAMBDA (BODY1 BODY2 DW?)                                        ; Edited 13-Jun-2024 11:46 by rmk
                                                                   ; Edited 17-Jun-2023 15:19 by rmk
                                                                   ; Edited 25-Feb-2022 18:02 by rmk
                                                                   ; Edited 18-Jan-2022 22:00 by rmk
                                                                   ; Edited 19-Dec-2021 21:05 by rmk
                                                                   ; Edited  5-Sep-2020 19:01 by rmk:
                                                                   ; Edited 15-Apr-88 14:41 by bvm
    (DECLARE (USEDFREE DIFFERENCES COMPARESTREAM))
    (LET (THING2 THING1 PRED DIFS TMP)
         (SETQ BODY1 (CL:REMOVE-IF (FUNCTION EDITDATE?)
                            BODY1))                                ; We don't care about editdate comments
         (SETQ BODY2 (CL:REMOVE-IF (FUNCTION EDITDATE?)
                            BODY2))
         (SETQ BODY1 (\CS.FIXFNS BODY1 DW?))
         (SETQ BODY2 (\CS.FIXFNS BODY2 DW?))
         (CL:WHEN (AND (SETQ THING1 (ASSOC 'DEFINE-FILE-INFO BODY1))
                       (SETQ THING2 (ASSOC 'DEFINE-FILE-INFO BODY2))
                       (\CS.COMPARE.DEFINE-FILE-INFO THING1 THING2))
             (SETQ BODY1 (REMOVE THING1 BODY1))
             (SETQ BODY2 (REMOVE THING2 BODY2)))

         ;; These are for commonlispy definers

         [for TYPE DEFFERS in FILEPKGTYPES when (AND (CL:SYMBOLP TYPE)
                                                     (SETQ DEFFERS (GET TYPE :DEFINED-BY)))
            do ;; handle definer based things

               (for DEFFER in DEFFERS WHEN [AND (SETQ THING1 (for X in BODY1 collect X
                                                                  when (EQ (CAR X)
                                                                           DEFFER)))
                                                (SETQ THING2 (for X in BODY2 collect X
                                                                  when (EQ (CAR X)
                                                                           DEFFER]
                  do ;; Take out all of the THINGS we are about to do.
```

```
                (SETQ BODY1 (CL:SET-DIFFERENCE BODY1 THING1 :TEST (FUNCTION EQUALALL)))
                (SETQ BODY2 (CL:SET-DIFFERENCE BODY2 THING2 :TEST (FUNCTION EQUALALL)))
                (CL:WHEN (SETQ DIFS (\CS.COMPARE.TYPES THING1 THING2 (CONCAT (OR (CL:DOCUMENTATION
                                                                                  TYPE
                                                                                  'DEFINE-TYPES)
                                                                                 TYPE)
                                                                             " defined by " DEFFER)
                                                   NIL
                                                   (GET DEFFER :DEFINITION-NAME)
                                                   TYPE))
                    [COND
                       ((SETQ TMP (ASSOC TYPE DIFFERENCES))
                        (NCONC TMP DIFS))
                       (T (push DIFFERENCES (CONS TYPE DIFS])]
          ;; These are for other filepkage types, as registered in COMPARESOURCETYPES
          [for TYPE in COMPARESOURCETYPES EACHTIME (SETQ PRED (fetch (CSTYPE PREDFN) of TYPE))
             WHEN [AND (SETQ THING1 (for X in BODY1 collect X when (CL:FUNCALL PRED X)))
                       (SETQ THING2 (for X in BODY2 collect X when (CL:FUNCALL PRED X]
             do (SETQ BODY1 (CL:SET-DIFFERENCE BODY1 THING1 :TEST (FUNCTION EQUALALL)))
                (SETQ BODY2 (CL:SET-DIFFERENCE BODY2 THING2 :TEST (FUNCTION EQUALALL)))
                (CL:WHEN [SETQ DIFS (\CS.COMPARE.TYPES THING1 THING2 (OR (fetch (CSTYPE TITLE) of TYPE)
                                                                          (MKSTRING (fetch (CSTYPE FPKGTYPE)
                                                                                       of TYPE)))
                                           (fetch (CSTYPE COMPAREFN) of TYPE)
                                           (OR (fetch (CSTYPE IDFN) of TYPE)
                                               (FUNCTION CADR))
                                           (SETQ TYPE (fetch (CSTYPE FPKGTYPE) of TYPE]
                    [COND
                       ((SETQ TMP (ASSOC TYPE DIFFERENCES))
                        (NCONC TMP DIFS))
                       (T (push DIFFERENCES (CONS TYPE DIFS])]
          (SETQ BODY2 (CL:SET-DIFFERENCE BODY2 (PROG1 BODY1
                                                   (SETQ BODY1 (CL:SET-DIFFERENCE BODY1 BODY2 :TEST
                                                                       (FUNCTION EQUALALL))))
                             :TEST
                             (FUNCTION EQUALALL)))
          (COND
             ((OR BODY1 BODY2)
              (printout CONTEXTSTREAM T "---Expressions:" T)
              (LET ((COMMENTX 0)
                    (COMMENTY 0))                                     ; Remove comments
                   [SETQ BODY1 (for X in BODY1 collect X unless (COND
                                                                     ((EQ (CAR X)
                                                                          COMMENTFLG)
                                                                      (add COMMENTX 1)
                                                                      T]
                   [SETQ BODY2 (for Y in BODY2 collect Y unless (COND
                                                                     ((EQ (CAR Y)
                                                                          COMMENTFLG)
                                                                      (add COMMENTY 1)
                                                                      T]
                   (COND
                      ((OR (NEQ COMMENTX 0)
                           (NEQ COMMENTY 0))
                       (printout CONTEXTSTREAM .I1 COMMENTX " comments -> " .I1 COMMENTY " comments." T T)))
                   [COND
                      [BODY1 (COND
                                (BODY2 (COMPARELISTS BODY1 BODY2 COMPARESTREAM)
                                       (\CS.EXAMINE BODY1 BODY2 NIL 'Expression))
                                (T (printout COMPARESTREAM "These are not on File 2:" T)
                                   (FOR X IN BODY1 DO (LVLPRINT X COMPARESTREAM 2 3)
                                                      (\CS.EXAMINE X NIL T NIL 'Expression]
                      (BODY2 (printout COMPARESTREAM "These are not on File 1:" T)
                             (FOR Y IN BODY2 DO (LVLPRINT Y COMPARESTREAM 2 3)
                                                (\CS.EXAMINE NIL Y T NIL 'Expression]
                   (OR (ASSOC 'Other DIFFERENCES)
                       (push DIFFERENCES (LIST 'Other '--])
```

(\CS.COMPARE.TYPES

```
  [LAMBDA (XTHING YTHING TITLE COMPAREFN IDFN TYPE)           ; Edited 25-Feb-2022 17:49 by rmk
                                                              ; Edited  9-Dec-2021 23:19 by rmk
                                                              ; Edited  1-Dec-2021 23:25 by rmk:
                                                              ; Edited 30-Nov-2021 23:07 by rmk:
                                                              ; Edited 27-Nov-2021 12:32 by rmk:
                                                              ; Edited 25-Nov-2021 13:29 by rmk:
                                                              ; Edited 29-Dec-86 11:49 by jds

;;; Compare things using COMPAREFN.  Deltas -> COMPARESTREAM.  Anything that passes the WHEN predicate has a difference somewhere, will
;;; produce some output.

    (DECLARE (USEDFREE CONTEXTSTREAM COMPARESTREAM))
    (LET (X Y RESULT NAME)
         (CL:WHEN (AND (OR XTHING YTHING)
                       (PROGN (SETQ XTHING (CL:SET-DIFFERENCE XTHING (PROG1 YTHING
                                                                    (SETQ YTHING
                                                                     (CL:SET-DIFFERENCE
                                                                      YTHING XTHING :TEST
                                                                      (FUNCTION EQUALALL))))
                                                :TEST
                                                (FUNCTION EQUALALL)))
                               (OR XTHING YTHING)))
```

;; We know we are going to have some output. Strings can go directly onto theCONTEXTSTREAM, and objects may then be inserted.

```
          (AND TITLE (printout CONTEXTSTREAM T "---" TITLE ":" T T))
          (for TAIL on XTHING do [SETQ NAME (CL:FUNCALL IDFN (SETQ X (CAR TAIL]
                                 [COND
                                    ([NOT (SETQ Y (find Y in YTHING suchthat (EQUAL (CL:FUNCALL IDFN Y)
                                                                                    NAME]
                                     (printout COMPARESTREAM .FONT BOLDFONT .P2 NAME .FONT DEFAULTFONT " is
                                            not on File 2" T T)
                                     (\CS.EXAMINE X NIL T NAME TYPE))
                                    (T (printout COMPARESTREAM .FONT BOLDFONT .P2 NAME ":" .FONT DEFAULTFONT T
                                              )
                                       (COND
                                          (COMPAREFN (CL:FUNCALL COMPAREFN X Y COMPARESTREAM))
                                          (T (COMPARELISTS X Y COMPARESTREAM)))
                                       (\CS.EXAMINE X Y NIL NAME TYPE)
                                       (RPLACA (FMEMB Y YTHING]
                                 (RPLACA TAIL)
                                 (push RESULT NAME))
          (for Y in (CL:SET-DIFFERENCE YTHING XTHING :TEST (FUNCTION EQUALALL))
             do (SETQ NAME (CL:FUNCALL IDFN Y))
                (printout COMPARESTREAM .FONT BOLDFONT .P2 NAME .FONT DEFAULTFONT " is not on File 1" T T)
                (\CS.EXAMINE Y NIL T NAME TYPE)
                (push RESULT NAME))
          RESULT)])
```

(\CS.EXAMINE

```
  [LAMBDA (X Y ONLYONE NAME TYPE)                                ; Edited 22-May-2022 16:28 by rmk
                                                                 ; Edited 27-Nov-2021 11:21 by rmk:
    (DECLARE (USEDFREE EXAMINE INSERTOBJECTS COMPARESTREAM CONTEXTSTREAM OBJECTS))
```

;; ONLYONE as a flag, because we don't want to test X or Y for NIL, that could be the contrasting value.

;; I don't understand MISC: changed but otherwise unclassified. Does that mean just an unknown type?

;; The only call seemed to be from \CS.COMPARE.MASTERS, where EXTRAS is set to either BODYX or BODYY if the other one is NIL. It may be
;; that that call only happens in the MISC case.

```
    (CL:UNLESS NAME (SETQ NAME "from File"))
```

;; Context gets printed to the CONTEXTSTREAM, diffs go to the COMPARESTREAM. If we aren't doing objects, those are the same streams, and
;; the output gets printed in the right order. Nothing to do here.

```
    (IF INSERTOBJECTS
        THEN [LET (STRING)
```

;; Take out last EOL, let SEPDIST space things out.

```
                  (CL:UNLESS (EQ 0 (GETFILEPTR CONTEXTSTREAM))
                      (SETQ STRING (CL:GET-OUTPUT-STREAM-STRING CONTEXTSTREAM))
                      (CL:WHEN (EQ (CHARCODE EOL)
                                   (NTHCHARCODE STRING -1))
                          (SETQ STRING (OR (SUBSTRING STRING 1 -2)
                                           "")))
                      (PUSH OBJECTS (CSOBJ.CREATE STRING)))
                  (CL:UNLESS (EQ 0 (GETFILEPTR COMPARESTREAM))
                      (SETQ STRING (CL:GET-OUTPUT-STREAM-STRING COMPARESTREAM))
```

;; Don't know why, but SEPTDIST doesn't work if there if there isn't at least one EOL. Magically, this gets the right
;; appearance and behavior.

```
                      (CL:WHEN (AND (EQ (CHARCODE EOL)
                                        (NTHCHARCODE STRING -1))
                                    (EQ (CHARCODE EOL)
                                        (NTHCHARCODE STRING -2)))
                          (SETQ STRING (OR (SUBSTRING STRING 1 -2)
                                           "")))
                      (PUSH OBJECTS (CSOBJ.CREATE STRING (LIST NAME TYPE X Y LABEL1 LABEL2)
                                           ONLYONE)))]
      ELSEIF (OR (LISTP X)
                 (LISTP Y))
        THEN                                                     ; No point in bringing up an editor on a non-list
             (IF ONLYONE
                 THEN (IF (OR (EQMEMB T EXAMINE)
                              (EQMEMB 'NEW EXAMINE))
                          THEN (EDITE (OR X Y)))
               ELSEIF (OR (EQMEMB T EXAMINE)
                          (EQMEMB 'OLD EXAMINE)
                          (EQMEMB 'MISCC))
                 THEN (IF (EQMEMB '2WINDOWS EXAMINE)
                          THEN (EXAMINEDEFS X Y NAME TYPE)
                        ELSE (EDITE (LIST X Y])
(\CS.FIXFNS
  [LAMBDA (BODY DW?)                                                ; Edited 29-Nov-2021 20:42 by rmk:
                                                                    ; Edited 26-Nov-2021 13:34 by rmk:

    ;; RMK: Functions are special in that they are grouped under DEFINEQ and they may need dwimifying.  We don't want to deal with these
    ;; idiosyncracies below, so our strategy is to split each multi-fn defineq into a sequence of single-fn defineqs , one for each function, then let it fall
    ;; through.  After dwimifying, things should be standard.

    (LET (DEFINEQS FNS (NOSPELLFLG T))
         (DECLARE (SPECVARS NOSPELLFLG))
         [SETQ DEFINEQS (for EXPR in BODY collect EXPR when (EQ (CAR EXPR)
                                                                'DEFINEQ]
         (SETQ BODY (CL:SET-DIFFERENCE BODY DEFINEQS))              ; Remove all the multiple function defineqs, so we can pack on
                                                                    ; the exploded forms
         [SETQ FNS (for DFQ in DEFINEQS join (FOR FN IN (CDR DFQ) COLLECT
                                                                         ;; FN is a single (NAME DEF) pair

                                                                         `(DEFINEQ (,@FN]
         (CL:WHEN DW?
             (FOR FN IN FNS DO (DWIMIFY (CADADR FN)
                                        T)))
         (SETQ BODY (APPEND FNS BODY])

(\CS.SORT.DECLARES
  [LAMBDA (DECLS)                                                   (* bvm%: "15-Nov-85 18:58")

;;; Sorts DECLS, a list of (DECLARE: --) expressions, into a set of declarations by tag, returning a list of entries of the form (tags  . expressions)

    (LET (RESULT)
         (DECLARE (SPECVARS RESULT))
         (for DEC in DECLS do (\CS.SORT.DECLARE1 DEC DEFAULT.DECLARE.TAGS))
        RESULT])

(\CS.SORT.DECLARE1
  [LAMBDA (DEC TAGLST)                                              (* bvm%: "15-Nov-85 19:09")
    (DECLARE (USEDFREE RESULT))

;;; Process one DECLARE: expression, partitioning it into subdeclarations put on RESULT assuming that the default tags in effect by the time you get
;;; here are in TAGLST

    (for TAIL on (CDR DEC) bind CURRENT TAG COMPLEMENT
       do (COND
             [(NLISTP (SETQ TAG (CAR TAIL)))                        ; Canonicalize tag
              (SELECTQ TAG
                  (DOEVAL@LOAD (SETQQ TAG EVAL@LOAD))
                  (DOEVAL@COMPILE
                       (SETQQ TAG EVAL@COMPILE))
                  (DOCOPY (SETQQ TAG COPY))
                  NIL)
              (COND
                 ((NOT (MEMB TAG TAGLST))
                  [SETQ TAGLST (COND
                                  [(STRPOS 'WHEN TAG)               ; These take an extra expression
                                   (APPEND TAGLST (LIST TAG (CAR (SETQ TAIL (CDR TAIL]
                                  ((FMEMB (SETQ COMPLEMENT (SELECTQ TAG
                                                               (COPY 'DONTCOPY)
                                                               (DONTCOPY 'COPY)
                                                               (EVAL@COMPILE 'DONTEVAL@COMPILE)
                                                               (DONTEVAL@COMPILE
                                                                    'EVAL@COMPILE)
                                                               (EVAL@LOAD 'DONTEVAL@LOAD)
                                                               (DONTEVAL@LOAD
                                                                    'EVAL@LOAD)
                                                               (FIRST 'NOTFIRST)
                                                               (NOTFIRST 'FIRST)
                                                              NIL))
                                          TAGLST)
                                   (SUBST TAG COMPLEMENT TAGLST))
                                  (T (APPEND TAGLST (LIST TAG]
                  (SETQ CURRENT NIL]
             ((EQ (CAR TAG)
                  'DECLARE%:)                                       ; Process embedded declaration
              (\CS.SORT.DECLARE1 TAG TAGLST))
             (T                                                     ; Stick this expression on the entry for the tags that tell when to
                                                                    ; eval it
                [COND
                   ([AND (NOT CURRENT)
                         (NOT (SETQ CURRENT (SASSOC TAGLST RESULT]
                    (SETQ RESULT (NCONC1 RESULT (SETQ CURRENT (LIST TAGLST]
                (push (CDR CURRENT)
                      TAG])

(\CS.FILTER.GARBAGE
  [LAMBDA (FILECONTENTS)                                            ; Edited 29-Dec-86 10:44 by jds
```

;;; Remove "Uninteresting" items from files to be compared. Removes FILECREATED form, filemap, copyright notice, and DECLARE: DONTCOPY
;;; items.

```
    (for X in FILECONTENTS collect X unless (OR (EQ (CAR X)
                                                    'FILECREATED)
                                                (AND (EQ (CAR X)
                                                         'DECLARE%:)
                                                     (EQ (CADR X)
                                                         'DONTCOPY)
                                                     (LISTP (CADDR X))
                                                     (OR (FMEMB 'COPYRIGHT (CADDR X))
                                                         (FMEMB 'FILEMAP (CADDR X])

)

(DEFINEQ
```

(\CS.ISFNFORM
```
  [LAMBDA (X)                                                  ; Edited 29-Nov-2021 20:34 by rmk:
                                                               ; Edited 26-Nov-2021 13:19 by rmk:
    (EQ 'DEFINEQ (CAR (LISTP X])
```

(\CS.COMPARE.FNS
```
  [LAMBDA (DQX DQY STREAM)                                     ; Edited 29-Nov-2021 20:51 by rmk:
    ;; CADADR is the body
    (COMPARELISTS (CADADR DQX)
           (CADADR DQY)
           STREAM])
```

(\CS.FNSID
```
  [LAMBDA (DQX)                                                ; Edited 29-Nov-2021 20:50 by rmk:
    (CAR (CADR DQX])
```

(\CS.ISVARFORM
```
  [LAMBDA (X)                                                  (* bvm%: "25-Sep-85 12:05")
    (SELECTQ (CAR X)
        ((RPAQ RPAQQ RPAQ?)
             T)
        NIL])
```

(\CS.COMPARE.VARS
```
  [LAMBDA (X Y STREAM)                                         ; Edited 29-Dec-86 12:15 by jds

;;; Compares two variable setting forms

    (COND
       ((EQ (CAR X)
            (CAR Y))                                           ; Same type of setting fn
        (COMPARELISTS (CADDR X)
               (CADDR Y)
               STREAM))
       (T (LET [[XVAL (COND
                         ((EQ (CAR X)
                              'RPAQQ)
                          (KWOTE (CADDR X)))
                         (T (CADDR X]
                (YVAL (COND
                         ((EQ (CAR Y)
                              'RPAQQ)
                          (KWOTE (CADDR Y)))
                         (T (CADDR Y]
               (COND
                  ((EQUAL XVAL YVAL)                           ; Same value, different setter
                   (printout STREAM (COND
                                       ((EQ (CAR X)
                                            'RPAQ?)
                                        'INITVARS)
                                       (T 'VARS))
                          " -> "
                          (COND
                             ((EQ (CAR Y)
                                  'RPAQ?)
                              'INITVARS)
                             (T 'VARS))
                          T))
                  (T (COMPARELISTS XVAL YVAL STREAM])
```

(\CS.ISMACROFORM
```
  [LAMBDA (X)                                                  (* bvm%: "25-Sep-85 12:19")
    (SELECTQ (CAR X)
        (DEFMACRO T)
```

```
         (PUTPROPS (FMEMB (CADDR X)
                          MACROPROPS))
        NIL])
```

(\CS.ISRECFORM

```
  [LAMBDA (X)                                                    ; Edited 25-Feb-2022 15:17 by rmk
                                                                 (* bvm%: "25-Sep-85 12:20")
    (OR (FMEMB (CAR X)
               CLISPRECORDTYPES)
        (EQ (CAR X)
            '/DECLAREDATATYPE])
```

(\CS.REC.NAME

```
  [LAMBDA (FORM)                                                 ; Edited 25-Feb-2022 15:24 by rmk
    (IF (AND (EQ (CAR FORM)
                 '/DECLAREDATATYPE)
             (EQ (CAR (CADR FORM))
                 'QUOTE))
        THEN (CADR (CADR FORM))
      ELSE (CADR FORM])
```

(\CS.ISCOURIERFORM

```
  [LAMBDA (X)                                                    (* bvm%: "13-Mar-86 16:21")
    (EQ (CAR X)
        'COURIERPROGRAM])
```

(\CS.ISTEMPLATEFORM

```
  [LAMBDA (X)                                                    (* bvm%: "13-Mar-86 16:20")
    (EQ (CAR X)
        'SETTEMPLATE])
```

(\CS.COMPARE.TEMPLATES

```
  [LAMBDA (X Y STREAM)                                           ; Edited 29-Dec-86 12:15 by jds
```

;;; Templates usually look like (SETTEMPLATE (QUOTE FN) (QUOTE TEMPLATE))

```
    (COND
       ((AND (EQUAL (CADR X)
                    (CADR Y))
             (EQ (CAR (CADDR X))
                 'QUOTE)
             (EQ (CAR (CADDR Y))
                 'QUOTE))
        (COMPARELISTS (CADR (CADDR X))
               (CADR (CADDR Y))
               STREAM))
       (T (COMPARELISTS X Y STREAM])
```

(\CS.ISPROPFORM

```
  [LAMBDA (X)                                                    (* bvm%: "13-Mar-86 16:34")
```

;;; (PUTPROPS SYMBOL PROP VALUE)

```
    (AND (EQ (CAR X)
             'PUTPROPS)
         (NULL (CDDDDR X])
```

(\CS.PROP.NAME

```
  [LAMBDA (X)                                                    (* bvm%: "13-Mar-86 16:29")
```

;;; The 'Name' of a property is its atom/value pair

```
    (LIST (CADR X)
          (CADDR X])
```

(\CS.COMPARE.PROPS

```
  [LAMBDA (X Y STREAM)                                           ; Edited 29-Dec-86 12:15 by jds
```

;;; Compare the values

```
    (COMPARELISTS (CADDDR X)
           (CADDDR Y)
           STREAM])
```

(\CS.ISADDVARFORM

```
  [LAMBDA (X)                                                    (* bvm%: "13-Mar-86 16:40")
    (EQ (CAR X)
        'ADDTOVAR])
```

```
(\CS.COMPARE.ADDVARS
  [LAMBDA (X Y STREAM)                                    ; Edited 29-Dec-86 12:15 by jds

;;; (ADDTOVAR ListName . values)

    (COMPARELISTS (CDDR X)
           (CDDR Y)
           STREAM])

(\CS.ISFPKGCOMFORM
  [LAMBDA (X)                                             (* bvm%: "13-Mar-86 16:50")

            (* * (PUTDEF (QUOTE name) (QUOTE FILEPKGCOMS) (QUOTE stuff)))

    (AND (EQ (CAR X)
             'PUTDEF)
         (EQUAL (CADDR X)
                ''FILEPKGCOMS])

(\CS.COMPARE.FPKGCOMS
  [LAMBDA (X Y STREAM)                                    ; Edited 29-Dec-86 12:16 by jds

            (* * (PUTDEF (QUOTE name) (QUOTE FILEPKGCOMS) (QUOTE stuff)))

    (COMPARELISTS (CADR (CADDDR X))
           (CADR (CADDDR Y))
           STREAM])

(\CS.COMPARE.DEFINE-FILE-INFO
  [LAMBDA (DFI1 DFI2)                                     ; Edited 19-Dec-2021 21:02 by rmk
    (AND (EQUAL (LISTGET :READTABLE DFI1)
                (LISTGET :READTABLE DFI2))
         (EQUAL (LISTGET :PACKAGE DFI1)
                (LISTGET :PACKAGE DFI2))
         (EQ (OR (LISTGET :BASE DFI1)
                 10)
             (OR (LISTGET :BASE DFI2)
                 10))
         (EQ (OR (LISTGET :FORMAT DFI1)
                 *DEFAULT-EXTERNALFORMAT*)
             (OR (LISTGET :FORMAT DFI2)
                 *DEFAULT-EXTERNALFORMAT*])

(\CS.IGNOREFORMS
  [LAMBDA (EXPR IGNOREFORMS CDRFLG)                       ; Edited  7-Feb-2024 15:39 by rmk
    ;; Replaces each sublist of X that begins with something in IGNOREFORMS by '**IGNORED**.  If IGNOREFORMS is or contains T, then
    ;; comment-looking forms will be deleted.

    ;; Tricky because we only want to check the CAR's at each level, but also don't want to lose any NLISTP tails.

    (if (NLISTP EXPR)
        then EXPR
      elseif (AND (NOT CDRFLG)
                  (EQMEMB (CAR EXPR)
                         IGNOREFORMS))
        then '**IGNORE**
      else (CONS (\CS.IGNOREFORMS (CAR EXPR)
                         IGNOREFORMS)
                 (\CS.IGNOREFORMS (CDR EXPR)
                         IGNOREFORMS T])
)

(DEFINEQ

(CSOBJ.CREATE
  [LAMBDA (STRING COMPAREDATA ONLYONE)                    ; Edited  4-Dec-2021 09:57 by rmk
                                                          ; Edited  1-Dec-2021 13:26 by rmk:
    (LET ((OBJ (IMAGEOBJCREATE STRING COMPARESOURCES-IMAGEFNS)))
         (IMAGEOBJPROP OBJ 'COMPAREDATA COMPAREDATA)
         (IMAGEOBJPROP OBJ 'ONLYONE ONLYONE)
         OBJ])

(CSOBJ.DISPLAYFN
  [LAMBDA (OBJ WINDOW)                                    ; Edited  4-Dec-2021 08:24 by rmk
                                                          ; Edited  1-Dec-2021 14:18 by rmk:
    (DSPFONT DEFAULTFONT WINDOW)
    (FOR I C (FONTARRAY _ (FONTMAPARRAY))
          (STRING _ (IMAGEOBJPROP OBJ 'OBJECTDATUM)) FROM 1
       DO (SELCHARQ (SETQ C (NTHCHARCODE STRING I))
               (EOL (TERPRI WINDOW))
```

```
                (NIL (RETURN))
                (IF (EQ C (CONSTANT (CHARCODE.DECODE FONTESCAPECHAR)))
                    THEN (DSPFONT (ELT FONTARRAY (NTHCHARCODE STRING (ADD I 1)))
                                 WINDOW)
                  ELSE (PRINTCCODE C WINDOW])
```

(CSOBJ.IMAGEBOXFN

```
  [LAMBDA (OBJ IMAGESTREAM CURRENTX RIGHTMARGIN)                      ; Edited  9-Dec-2021 23:02 by rmk
                                                                      ; Edited  7-Dec-2021 10:50 by rmk
                                                                      ; Edited  5-Dec-2021 23:52 by rmk
                                                                      ; Edited  4-Dec-2021 08:24 by rmk
                                                                      ; Edited  1-Dec-2021 13:27 by rmk:

    ;; Calculate the height of each line, and the width of the widest line.

    ;; Probably ought to compute the max height per line, at every font change, add it at each EOL.

    (SETQ IMAGESTREAM (GETSTREAM IMAGESTREAM 'OUTPUT))
    (FOR I C (STRING _ (IMAGEOBJPROP OBJ 'OBJECTDATUM))
         (FONT _ (FONTCREATE DEFAULTFONT NIL NIL NIL IMAGESTREAM))
         (HEIGHT _ 0)
         (LINELENGTH _ 0)
         (MAXLINELENGTH _ 0)
         (FONTARRAY _ (FONTMAPARRAY)) FROM 1
       DO (SELCHARQ (SETQ C (NTHCHARCODE STRING I))
                (EOL (ADD HEIGHT (FONTPROP FONT 'HEIGHT))
                     (CL:WHEN (IGREATERP LINELENGTH MAXLINELENGTH)
                            (SETQ MAXLINELENGTH LINELENGTH))
                     (SETQ LINELENGTH 0))
                (NIL                                                  ; end of string
                     (CL:WHEN (IGREATERP LINELENGTH MAXLINELENGTH)
                            (SETQ MAXLINELENGTH LINELENGTH))
                     (RETURN (CREATE IMAGEBOX
                                     XSIZE _ MAXLINELENGTH
                                     YSIZE _ HEIGHT
                                     YDESC _ (DIFFERENCE HEIGHT (FONTPROP FONT 'HEIGHT))
                                     XKERN _ 0)))
                (IF (EQ C (CONSTANT (CHARCODE.DECODE FONTESCAPECHAR)))
                    THEN (SETQ FONT (FONTCREATE (ELT FONTARRAY (NTHCHARCODE STRING (ADD I 1)))
                                           NIL NIL NIL IMAGESTREAM))
                  ELSE (ADD LINELENGTH (CHARWIDTH C FONT])
```

(CSOBJ.BUTTONEVENTINFN

```
  [LAMBDA (OBJ WINDOW)                                                ; Edited 28-Jan-2022 18:22 by rmk
                                                                      ; Edited 25-Jan-2022 16:04 by rmk
                                                                      ; Edited 23-Jan-2022 18:11 by rmk
    (LET
     [(COMPAREDATA (IMAGEOBJPROP OBJ 'COMPAREDATA]
     (CL:WHEN (AND COMPAREDATA (MOUSESTATE LEFT)
                   (UNTILMOUSESTATE (NOT LEFT)))
         (LET
          ((NAME (POP COMPAREDATA))
           (TYPE (POP COMPAREDATA))
           (DEF1 (POP COMPAREDATA))
           (DEF2 (POP COMPAREDATA))
           (TITLE1 (POP COMPAREDATA))
           (TITLE2 (CAR COMPAREDATA)))

          ;; Move the cursor to just slightly below the current object, so that the edit windows are well aligned. We have to figure out the bottom of the
          ;; current object, in screen coordinates.

          [LET ((OBJREGION (OBJ.FIND.REGION WINDOW OBJ)))
               (\CURSORPOSITION (IPLUS 20 LASTMOUSEX)
                      (IPLUS (IDIFFERENCE (FETCH (REGION BOTTOM) OF OBJREGION)
                                    (FETCH (REGION HEIGHT) OF OBJREGION))
                             (FETCH (REGION TOP) OF (WINDOWREGION WINDOW]
          (LET [EWINDOW (RELPOS (RELCREATEPOSITION `(,WINDOW 0.5)
                                       `(,WINDOW 0 -2]
               (CLOSEWITH.DOIT WINDOW)
               (SETQ EWINDOW
                (IF (IMAGEOBJPROP OBJ 'ONLYONE)
                    THEN [SEDIT:GET-WINDOW
                                (SEDIT:SEDIT (OR DEF1 DEF2)
                                       `(:REGION ,(RELCREATEREGION 600 (CL:IF (ILESSP (COUNT (OR DEF1 DEF2))
                                                                                      100)
                                                                          150
                                                                          400)
                                                         (CL:IF DEF1
                                                             'RIGHT
                                                             'LEFT)
                                                         'TOP RELPOS NIL T]
                  ELSE                                                ; Spread the arguments
                        (EXAMINEDEFS NAME TYPE DEF1 DEF2 TITLE1 TITLE2 RELPOS)))
               (CLOSEWITH EWINDOW WINDOW)
               (MOVEWITH EWINDOW WINDOW)
               EWINDOW))))])
```

```
(CSOBJ.COPYBUTTONEVENTINFN
  [LAMBDA (OBJ WINDOW REGION)                                    ; Edited  3-Jan-2022 08:36 by rmk
    (CL:WHEN (CAR (IMAGEOBJPROP OBJ 'COMPAREDATA))
        [COPYINSERT (CAR (IMAGEOBJPROP OBJ 'COMPAREDATA])])

)

(RPAQ? COMPARESOURCES-IMAGEFNS (IMAGEFNSCREATE 'CSOBJ.DISPLAYFN 'CSOBJ.IMAGEBOXFN NIL NIL NIL
                                        'CSOBJ.BUTTONEVENTINFN 'CSOBJ.COPYBUTTONEVENTINFN))

(RPAQQ COMPARESOURCETYPES
       ((FNS \CS.ISFNFORM \CS.COMPARE.FNS \CS.FNSID "FNS defined by DEFINEQ")
        (VARS \CS.ISVARFORM \CS.COMPARE.VARS)
        (MACROS \CS.ISMACROFORM)
        (RECORDS \CS.ISRECFORM NIL \CS.REC.NAME)
        (PROPS \CS.ISPROPFORM \CS.COMPARE.PROPS \CS.PROP.NAME "Properties")
        (ADDVARS \CS.ISADDVARFORM \CS.COMPARE.ADDVARS CADR "Additions to lists")
        (TEMPLATES \CS.ISTEMPLATEFORM \CS.COMPARE.TEMPLATES CADADR)
        (COURIERPROGRAMS \CS.ISCOURIERFORM)
        (FILEPKGCOMS \CS.ISFPKGCOMFORM \CS.COMPARE.FPKGCOMS CADADR)))

(RPAQQ DEFAULT.DECLARE.TAGS (EVAL@LOAD DONTEVAL@COMPILE COPY NOTFIRST))

(DEFINEQ

(CSBROWSER
  [LAMBDA (FILE1 FILE2 DW? LABEL1 LABEL2 REGION IGNOREFORMS TITLE)
    ;; Edited  7-Feb-2024 15:52 by rmk
    ;; Edited 17-Jun-2023 15:21 by rmk
    ;; Edited 22-May-2022 18:42 by rmk
    ;; Edited 12-May-2022 10:16 by rmk
    ;; Edited 24-Jan-2022 23:11 by rmk: EXAMINE is non-NIL, we run the compare twice.  Once to get the TEDIT up as a kind of table of contents, and
    ;; the second time to run through all of the SEDIT windows.
    ;; If EXAMINE is non-NIL, we run the compare twice.  Once to get the TEDIT up as a kind of table of contents, and the second time to run through
    ;; all of the SEDIT windows.
    ;; Returns browser window
    ;; Don't use the INFILEP value, because that might screw with capitalization that the caller prefers.  If the file can be found that way, then lower
    ;; functions will find it.
    (DECLARE (SPECVARS LABEL1 LABEL2))
    (SETQ FILE1 (OR (STREAMP FILE1)
                    (INFILEP FILE1)
                    (FINDFILE FILE1 NIL DIRECTORIES)
                    (ERROR "FILE NOT FOUND" FILE1)))
    (SETQ FILE2 (OR (STREAMP FILE2)
                    (INFILEP FILE2)
                    (FINDFILE FILE2 NIL DIRECTORIES)
                    (ERROR "FILE NOT FOUND" FILE2)))
    (CL:UNLESS (LISPSOURCEFILEP FILE1)
           (ERROR FILE1 " is not a Medley source file"))
    (CL:UNLESS LABEL1
        (SETQ LABEL1 (PACKFILENAME 'HOST NIL 'DIRECTORY NIL 'BODY FILE1)))
    (CL:UNLESS LABEL2
        (SETQ LABEL2 (PACKFILENAME 'HOST NIL 'DIRECTORY NIL 'BODY FILE2)))
    (CL:UNLESS (LISPSOURCEFILEP FILE2)
           (ERROR FILE1 " is not a Medley source file"))
    (CL:UNLESS TITLE
        (SETQ TITLE (CONCAT "COMPARESOURCES of " LABEL1 " and " LABEL2)))
    (LET [(WINDOW (OBJ.CREATEW 'VERTICAL REGION TITLE NIL T (FONTPROP DEFAULTFONT 'HEIGHT]
         (WINDOWPROP WINDOW 'UNDERSCONTRUCTION T)
         (GETPROMPTWINDOW WINDOW T)
         (WINDOWPROP WINDOW 'UNDERSCONTRUCTION NIL)
         (COMPARESOURCES FILE1 FILE2 '(T 2WINDOWS)
                DW? WINDOW IGNOREFORMS LABEL1 LABEL2)
         (OPENW WINDOW)
         WINDOW])

)

(FILESLOAD (SYSLOAD)
       OBJECTWINDOW EXAMINEDEFS REGIONMANAGER)

(DECLARE%: EVAL@COMPILE DONTCOPY

(DECLARE%: EVAL@COMPILE

(RECORD CSTYPE (FPKGTYPE PREDFN COMPAREFN IDFN TITLE))
)

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS COMPARESOURCETYPES CLISPRECORDTYPES MACROPROPS DEFAULT.DECLARE.TAGS)
)
)
```

FUNCTION INDEX

VARIABLE INDEX

RECORD INDEX