File created: 6-Nov-91 14:39:39 **{DSK}<usr>local>lde>lispcore>sources>ATBL.;6**

changes to: (FNS \ORIGREADTABLE GETSYNTAX SETSYNTAX SYNTAXP \COPYSYNTAX \GETCHARCODE \SETFATSYNCODE CONTROL COPYTERMTABLE DELETECONTROL GETDELETECONTROL ECHOCHAR ECHOCONTROL ECHOMODE GETECHOMODE GETCONTROL GETTERMTABLE RAISE GETRAISE RESETTERMTABLE SETTERMTABLE TERMTABLEP \GETTERMSYNTAX \GTTERMTABLE \ORIGTERMTABLE \SETTERMSYNTAX \TERMCLASSTOCODE \TERMCODETOCLASS \LITCHECK COPYREADTABLE FIND-READTABLE IN-READTABLE ESCAPE GETBRK GETREADTABLE GETSEPR READMACROS READTABLEP READTABLEPROP RESETREADTABLE SETBRK SETREADTABLE SETSEPR \GETREADSYNTAX \GTREADTABLE \GTREADTABLE1 \READCLASSTOCODE \SETMACROSYNTAX \SETREADSYNTAX \READTABLEP.DEFPRINT \ATBLSET MAKE-READER-ENVIRONMENT EQUAL-READER-ENVIRONMENT SET-READER-ENVIRONMENT)
(RECORDS READTABLEP)

previous date: 4-Apr-91 22:36:35 {DSK}<usr>local>lde>lispcore>sources>ATBL.;2

Read Table: INTERLISP

Package: INTERLISP

Format: XCCS

;;


```
(RPAQQ ATBLCOMS
       [(E (RESETSAVE (RADIX 8)))
        (COMS                                                   ; Common features of read and terminal tables
              (DECLARE%: DONTCOPY (EXPORT (MACROS \SYNCODE \SETSYNCODE)
                                          (RECORDS CHARTABLE))
                     (CONSTANTS \NSCHARHASHKEYS \NSCHARHASHOVERFLOW)
                     (MACROS \CREATENSCHARHASH \MAPCHARTABLE))
              (FNS GETSYNTAX SETSYNTAX SYNTAXP \COPYSYNTAX \GETCHARCODE \SETFATSYNCODE))
        (COMS                                                   ; terminal tables
              (FNS CONTROL COPYTERMTABLE DELETECONTROL GETDELETECONTROL ECHOCHAR ECHOCONTROL ECHOMODE
                   GETECHOMODE GETCONTROL GETTERMTABLE RAISE GETRAISE RESETTERMTABLE SETTERMTABLE TERMTABLEP
                   \GETTERMSYNTAX \GTTERMTABLE \ORIGTERMTABLE \SETTERMSYNTAX \TERMCLASSTOCODE \TERMCODETOCLASS
                   \LITCHECK)
              (DECLARE%: DONTCOPY (EXPORT (CONSTANTS * CCECHOMODES)
                                          (CONSTANTS * TERMCLASSES)
                                          (RECORDS TERMCODE TERMTABLEP)))
              (INITRECORDS TERMTABLEP))
        (COMS                                                   ; read tables
              (FNS COPYREADTABLE FIND-READTABLE IN-READTABLE ESCAPE GETBRK GETREADTABLE GETSEPR READMACROS
                   READTABLEP READTABLEPROP RESETREADTABLE SETBRK SETREADTABLE SETSEPR \GETREADSYNTAX
                   \GTREADTABLE \GTREADTABLE1 \ORIGREADTABLE \READCLASSTOCODE \SETMACROSYNTAX \SETREADSYNTAX
                   \READTABLEP.DEFPRINT)
              (PROP ARGNAMES READTABLEPROP)
              (DECLARE%: EVAL@COMPILE DONTCOPY                  ; READCLASSTOKENS Generates READCLASSES and some
                                                                ; interesting SELECTQ's
                                                                ; OTHER must be zero because of initialization.
                     [VARS READCLASSTOKENS (READCLASSES (MAPCAR READCLASSTOKENS (FUNCTION
                                                                                  (LAMBDA
                                                                                   (PAIR)
                                                                                   (LIST (PACK* (CAR PAIR)
                                                                                                ".RC")
                                                                                         (CADR PAIR]
                     (MACROS \COMPUTED.FORM)
                                                                ; This macro ought to be official somehow
                     (RECORDS CONTEXTS ESCAPES WAKEUPS)
                     (EXPORT (MACROS \GETREADMACRODEF \GTREADTABLE \GTREADTABLE1)
                            (CONSTANTS MACROBIT BREAKBIT STOPATOMBIT ESCAPEBIT INNERESCAPEBIT)
                            (CONSTANTS * READCODEMASKS)
                            (CONSTANTS * READMACROCONTEXTS)
                            (CONSTANTS * READCLASSES)
                            (CONSTANTS * READMACROWAKEUPS)
                            (CONSTANTS * READMACROESCAPES)
                            (RECORDS READCODE READMACRODEF READTABLEP))
                     (GLOBALVARS \ORIGREADTABLE \READTABLEHASH \ORIGTERMTABLE))
              (INITRECORDS READTABLEP))
        [COMS (FNS \ATBLSET)
              (INITRECORDS READER-ENVIRONMENT)
                                                                ; Definition is on CMLREAD, need it here to initialize
                                                                ; *OLD-INTERLISP-READ-ENVIRONMENT*
              (FNS MAKE-READER-ENVIRONMENT EQUAL-READER-ENVIRONMENT SET-READER-ENVIRONMENT)
              (INITVARS (*LISP-PACKAGE*)
                     (*INTERLISP-PACKAGE*)
                     (*KEYWORD-PACKAGE*))
              (DECLARE%: DONTEVAL@LOAD DOCOPY (P (\ATBLSET]
        (LOCALVARS . T)
        (DECLARE%: DONTEVAL@LOAD DOEVAL@COMPILE DONTCOPY COMPILERVARS (ADDVARS (NLAMA)
                                                                               (NLAML)
                                                                               (LAMA READTABLEPROP])
```

;; Common features of read and terminal tables

```
(DECLARE%: DONTCOPY
```

```
;; FOLLOWING DEFINITIONS EXPORTED

(DECLARE%: EVAL@COMPILE

(PUTPROPS \SYNCODE DMACRO [OPENLAMBDA (TABLE CHAR)
                                (CHECK (type? CHARTABLE TABLE))       ; 0 is either NONE.TC, REAL.CCE, or OTHER.RC
                                (COND
                                   ((IGREATERP CHAR \MAXTHINCHAR)
                                    (OR (AND (fetch (CHARTABLE NSCHARHASH) of TABLE)
                                             (GETHASH CHAR (fetch (CHARTABLE NSCHARHASH) of TABLE)))
                                        0))
                                   (T (\GETBASEBYTE TABLE CHAR])

(PUTPROPS \SETSYNCODE DMACRO [LAMBDA (TABLE CHAR CODE)
                                (CHECK (type? CHARTABLE TABLE)) ; 0 is REAL.CCE, NONE.TC, OTHER.RC
                                (COND
                                   ((ILEQ CHAR \MAXTHINCHAR)
                                    (\PUTBASEBYTE TABLE CHAR CODE))
                                   (T (\SETFATSYNCODE TABLE CHAR CODE])
)

(DECLARE%: EVAL@COMPILE

(DATATYPE CHARTABLE ((CHARSET0 256 BYTE)
                     (NSCHARHASH FULLPOINTER)))
)

(/DECLAREDATATYPE 'CHARTABLE
       '(BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE
              BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE
              BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE
              BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE
              BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE
              BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE
              BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE
              BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE
              BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE
              BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE
              BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE
              BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE
              BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE
              BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE FULLPOINTER)
       ;; ---field descriptor list elided by lister---
       '130)

;; END EXPORTED DEFINITIONS

(DECLARE%: EVAL@COMPILE

(RPAQQ \NSCHARHASHKEYS 10)

(RPAQQ \NSCHARHASHOVERFLOW 1.3)

(CONSTANTS \NSCHARHASHKEYS \NSCHARHASHOVERFLOW)
)

(DECLARE%: EVAL@COMPILE

(PUTPROPS \CREATENSCHARHASH MACRO (ARGS                       ; added size argument for creation of \ORIGTERMTABLE during
                                                              ; initialization.
                                    (LIST 'HASHARRAY (OR (CAR ARGS)
                                                         '\NSCHARHASHKEYS)
                                          '\NSCHARHASHOVERFLOW)))

(PUTPROPS \MAPCHARTABLE MACRO [LAMBDA (FN CHARTBL)
                                 (for I from 0 to \MAXTHINCHAR do (APPLY* FN (\GETBASEBYTE CHARTBL I)
                                                                          I))
                                 (COND
                                    ((fetch (CHARTABLE NSCHARHASH) of CHARTBL)
                                     (MAPHASH (fetch (CHARTABLE NSCHARHASH) of CHARTBL)
                                            FN])
)
)

(DEFINEQ

(GETSYNTAX
  [LAMBDA (CH TABLE)                                          (* bvm%: " 8-Mar-86 17:22")
    (COND
       [(FIXP (SETQ CH (\GETCHARCODE CH)))
        (COND
           ((type? TERMTABLEP TABLE)
            (\GETTERMSYNTAX CH TABLE))
           (T (\GETREADSYNTAX CH (\GTREADTABLE TABLE T]
```

```
       (T (PROG (TEM CHARTBL RESULT)
                (COND
                   ((SETQ TEM (\READCLASSTOCODE CH))
                    (SETQ CHARTBL (fetch READSA of (\GTREADTABLE TABLE T)))
                    (\MAPCHARTABLE [FUNCTION (LAMBDA (VAL KEY)
                                                (DECLARE (USEDFREE TEM RESULT))
                                                (COND
                                                   ((EQ TEM VAL)
                                                    (push RESULT KEY]
                           CHARTBL))
                   ((EQ CH 'BREAK)
                    (SETQ CHARTBL (fetch READSA of (\GTREADTABLE TABLE T)))
                    (\MAPCHARTABLE [FUNCTION (LAMBDA (VAL KEY)
                                                (DECLARE (USEDFREE TEM RESULT))
                                                (COND
                                                   ((fetch BREAK of VAL)
                                                    (push RESULT KEY]
                           CHARTBL))
                   ((SETQ TEM (\TERMCLASSTOCODE CH))
                    (SETQ CHARTBL (fetch TERMSA of (\GTTERMTABLE TABLE T)))
                    (\MAPCHARTABLE [FUNCTION (LAMBDA (VAL KEY)
                                                (DECLARE (USEDFREE TEM RESULT))
                                                (COND
                                                   ((EQ TEM (fetch TERMCLASS of VAL))
                                                    (push RESULT (PROG1 KEY
                                                                        (* SELECTC TEM ((LIST NONE.TC WORDSEPR.TC)
                                                                        (* ; "Only these classes have multiple members") KEY)
                                                                        (RETURN (CONS KEY)))
                                                                            )]
                           CHARTBL))
                   [(FMEMB CH '(MACRO SPLICE INFIX))
                    (PROG [LST (A (fetch READMACRODEFS of (\GTREADTABLE TABLE T]
                          (COND
                             (A [MAPHASH A (FUNCTION (LAMBDA (DEF C)
                                                       (AND (EQ CH (fetch MACROTYPE of DEF))
                                                            (push LST C]
                                (RETURN LST]
                   ((SETQ TEM (fetch (CONTEXTS VAL) of CH))
                    (SETQ CHARTBL (fetch READSA of (\GTREADTABLE TABLE T)))
                    (\MAPCHARTABLE [FUNCTION (LAMBDA (VAL KEY)
                                                (DECLARE (USEDFREE TEM RESULT))
                                                (COND
                                                   ((EQ TEM (fetch MACROCONTEXT of VAL))
                                                    (push RESULT KEY]
                           CHARTBL))
                   ((SETQ TEM (fetch (WAKEUPS VAL) of CH))
                    (SETQ CHARTBL (fetch READSA of (\GTREADTABLE TABLE T)))
                    (\MAPCHARTABLE [FUNCTION (LAMBDA (VAL KEY)
                                                (DECLARE (USEDFREE TEM RESULT))
                                                (COND
                                                   ((EQ TEM (fetch WAKEUP of VAL))
                                                    (push RESULT KEY]
                           CHARTBL))
                   ((SETQ TEM (fetch (ESCAPES VAL) of CH))
                    (SETQ CHARTBL (fetch READSA of (\GTREADTABLE TABLE T)))
                    (\MAPCHARTABLE [FUNCTION (LAMBDA (VAL KEY)
                                                (DECLARE (USEDFREE TEM RESULT))
                                                (COND
                                                   ((EQ TEM (fetch ESCAPE of VAL))
                                                    (push RESULT KEY]
                           CHARTBL))
                   (T (\ILLEGAL.ARG CH)))
                (RETURN RESULT])
```

(**SETSYNTAX**

```
  [LAMBDA (CHAR CLASS TBL)                                    (* rmk%: "20-Nov-84 15:47")
    (OR (FIXP (SETQ CHAR (\GETCHARCODE CHAR)))
        (\ILLEGAL.ARG CHAR))
    [OR (type? READTABLEP TBL)
        (type? TERMTABLEP TBL)
        (SETQ TBL (COND
                     ((OR (type? TERMTABLEP CLASS)
                          (\TERMCLASSTOCODE CLASS))
                      (\GTTERMTABLE TBL))
                     (T (\GTREADTABLE TBL]
    [COND
       ((OR (type? READTABLEP CLASS)
            (type? TERMTABLEP CLASS)
            (SELECTQ CLASS
                 ((NIL T ORIG)
                      T)
                 NIL))
        (SETQ CLASS (GETSYNTAX CHAR CLASS)))
       ((FIXP (SETQ CLASS (\GETCHARCODE CLASS)))
        (SETQ CLASS (GETSYNTAX CLASS TBL]
    (COND
```

```
       ((type? READTABLEP TBL)
        (PROG1 (\GETREADSYNTAX CHAR TBL)
               (\SETREADSYNTAX CHAR CLASS TBL)))
       (T (PROG1 (\GETTERMSYNTAX CHAR TBL)
                 (\SETTERMSYNTAX CHAR CLASS TBL])
```

(SYNTAXP

```
  [LAMBDA (CODE CLASS TABLE)                                    (* rmk%: " 5-JUN-80 22:40")
    (PROG (D)
          (RETURN (COND
                      ((EQ CLASS 'BREAK)
                       (fetch BREAK of (\SYNCODE (fetch READSA of (\GTREADTABLE TABLE))
                                                 CODE)))
                      ((SETQ D (\READCLASSTOCODE CLASS))
                       (EQ D (\SYNCODE (fetch READSA of (\GTREADTABLE TABLE))
                                       CODE)))
                      [(SETQ D (\TERMCLASSTOCODE CLASS))
                       (EQ D (fetch TERMCLASS of (\SYNCODE (fetch TERMSA of (\GTTERMTABLE TABLE))
                                                           CODE]
                      [(FMEMB CLASS '(MACRO SPLICE INFIX))
                       (AND (SETQ D (fetch READMACRODEFS of (\GTREADTABLE TABLE)))
                            (EQ CLASS (fetch MACROTYPE of (GETHASH CODE D]
                      [(SETQ D (fetch (CONTEXTS VAL) of CLASS))
                       (EQ D (fetch MACROCONTEXT of (\SYNCODE (fetch READSA of (\GTREADTABLE TABLE))
                                                              CODE]
                      [(SETQ D (fetch (WAKEUPS VAL) of CLASS))
                       (EQ D (fetch WAKEUP of (\SYNCODE (fetch READSA of (\GTREADTABLE TABLE))
                                                        CODE]
                      [(SETQ D (fetch (ESCAPES VAL) of CLASS))
                       (EQ D (fetch ESCAPE of (\SYNCODE (fetch READSA of (\GTREADTABLE TABLE))
                                                        CODE]
                      (T (\ILLEGAL.ARG CLASS])
```

(\COPYSYNTAX

```
  [LAMBDA (A B)                                                 (* gbn "15-Sep-85 22:36")
```

;; Copies chartable A into chartable B

```
    (CHECK (AND (type? CHARTABLE A)
                (type? CHARTABLE B)))
    (\MOVEBYTES A 0 B 0 (ADD1 \MAXTHINCHAR))
    (COND
       ((fetch (CHARTABLE NSCHARHASH) of A)
        (replace (CHARTABLE NSCHARHASH) of B with (REHASH (fetch (CHARTABLE NSCHARHASH) of A)
                                                          (\CREATENSCHARHASH])
```

(\GETCHARCODE

```
  [LAMBDA (C)                                                   (* rmk%: "20-Nov-84 15:46")
    (COND
       ((AND (NUMBERP C)
             (\CHARCODEP (FIX C)))
        (FIX C))
       ((AND (LITATOM C)
             (EQ 1 (NCHARS C)))
        (CHCON1 C))
       (T C])
```

(\SETFATSYNCODE

```
  [LAMBDA (TABLE CHAR CODE)                                     (* bvm%: " 8-Mar-86 17:03")
```

;;; Called by \SETSYNCODE macro for fat characters

```
    (SETQ TABLE (\DTEST TABLE 'CHARTABLE))                      ; CODE = 0 is REAL.CCE, NONE.TC, OTHER.RC
    (COND
       ((ILEQ CHAR \MAXTHINCHAR)
        (\PUTBASEBYTE TABLE CHAR CODE))
       ((EQ 0 CODE)
        (COND
           ((fetch (CHARTABLE NSCHARHASH) of TABLE)             ; there was already a table here so record the change
            (PUTHASH CHAR CODE (fetch (CHARTABLE NSCHARHASH) of TABLE)))
           (T                                                   ; No hashtable yet, and only the default is being stored, so don't
                                                                ; build the hashtable
              0)))
       (T (PUTHASH CHAR CODE (OR (fetch (CHARTABLE NSCHARHASH) of TABLE)
                                 (replace (CHARTABLE NSCHARHASH) of TABLE with (\CREATENSCHARHASH])

)

;; terminal tables

(DEFINEQ
```

(CONTROL

```
  [LAMBDA (MODE TTBL)                                           (* rmk%: " 8-FEB-80 11:59")
```

```
    (PROG1 (fetch CONTROLFLG of (SETQ TTBL (\GTTERMTABLE TTBL)))
        (replace CONTROLFLG of TTBL with (AND MODE T)))])
```

(COPYTERMTABLE

```
  [LAMBDA (TTBL)                                                    (* lmm "14-APR-81 14:27")
    (create TERMTABLEP using (SETQ TTBL (\GTTERMTABLE TTBL T))
                             TERMSA _ (create CHARTABLE using (fetch TERMSA of TTBL])
```

(DELETECONTROL

```
  [LAMBDA (TYPE MESSAGE TTBL)                                       (* lmm " 1-Jan-85 21:34")
    (PROG [VAL (TBL (\GTTERMTABLE TTBL (NULL MESSAGE]
          (SETQ VAL (SELECTQ TYPE
                         ((ECHO NOECHO)
                              (PROG1 (fetch DELCHARECHO of TBL)
                                     (replace DELCHARECHO of TBL with TYPE)))
                         (DELCHARECHO (PROG1 (fetch DELCHARECHO of TBL)
                                             (SELECTQ MESSAGE
                                                  (NIL                ; Called only to get current value
                                                       )
                                                  ((ECHO NOECHO)
                                                       (replace DELCHARECHO of TBL with MESSAGE))
                                                  (LISPERROR "ILLEGAL ARG" MESSAGE))))
                         ((LINEDELETE DELETELINE)
                              (PROG1 (fetch LINEDELETE of TBL)
                                   (AND MESSAGE (replace LINEDELETE of TBL with (\LITCHECK MESSAGE)))))
                         (1STCHDEL (PROG1 (fetch 1STCHDEL of TBL)
                                          (AND MESSAGE (replace 1STCHDEL of TBL with (\LITCHECK MESSAGE)))))
                         (NTHCHDEL (PROG1 (fetch NTHCHDEL of TBL)
                                          (AND MESSAGE (replace NTHCHDEL of TBL with (\LITCHECK MESSAGE)))))
                         (POSTCHDEL (PROG1 (fetch POSTCHDEL of TBL)
                                           (AND MESSAGE (replace POSTCHDEL of TBL with (\LITCHECK MESSAGE)))))
                         (EMPTYCHDEL (PROG1 (fetch EMPTYCHDEL of TBL)
                                            (AND MESSAGE (replace EMPTYCHDEL of TBL with (\LITCHECK MESSAGE)))))
                         (LISPERROR "ILLEGAL ARG" TYPE)))
          (RETURN (COND
                      ((STRINGP VAL)
                       (CONCAT VAL))
                      (T VAL])
```

(GETDELETECONTROL

```
  [LAMBDA (TYPE TTBL)                                               (* lmm " 1-Jan-85 21:20")
    (PROG (TBL VAL)
          (SETQ TBL (\GTTERMTABLE TTBL T))
          (SETQ VAL (SELECTQ TYPE
                         ((ECHO NOECHO)
                              (fetch DELCHARECHO of TBL))
                         (DELCHARECHO (fetch DELCHARECHO of TBL))
                         ((LINEDELETE DELETELINE)
                              (fetch LINEDELETE of TBL))
                         (1STCHDEL (fetch 1STCHDEL of TBL))
                         (NTHCHDEL (fetch NTHCHDEL of TBL))
                         (POSTCHDEL (fetch POSTCHDEL of TBL))
                         (EMPTYCHDEL (fetch EMPTYCHDEL of TBL))
                         (LISPERROR "ILLEGAL ARG" TYPE)))
          (RETURN (COND
                      ((STRINGP VAL)
                       (CONCAT VAL))
                      (T VAL])
```

(ECHOCHAR

```
  [LAMBDA (CHARCODE MODE TTBL)                                      (* lmm " 1-Jan-85 21:29")
    (COND
       ((LISTP CHARCODE)
        (for X in CHARCODE do (ECHOCHAR X MODE TTBL)))
       (T (PROG [B (SA (fetch TERMSA of (\GTTERMTABLE TTBL (NULL MODE]
                (RETURN (PROG1 (SELECTC (fetch CCECHO of (SETQ B (\SYNCODE SA CHARCODE)))
                                        (REAL.CCE 'REAL)
                                        (IGNORE.CCE 'IGNORE)
                                        (SIMULATE.CCE 'SIMULATE)
                                        'INDICATE)
                               [AND MODE (\SETSYNCODE SA CHARCODE (create TERMCODE using B CCECHO _
                                                                                  (SELECTQ MODE
                                                                                       (REAL REAL.CCE)
                                                                                       (IGNORE IGNORE.CCE)
                                                                                       (SIMULATE SIMULATE.CCE)
                                                                                       ((INDICATE UPARROW)
                                                                                           INDICATE.CCE)
                                                                                       (\ILLEGAL.ARG MODE])])
```

(ECHOCONTROL

```
  [LAMBDA (CHAR MODE TTBL)                                          (* rmk%: "20-Nov-84 15:14")
    (PROG ((C (\GETCHARCODE CHAR)))
```

```
          (OR [AND (\THINCHARCODEP C)
                   (OR (ILESSP C 32)
                       (AND (IGEQ C (CHARCODE A))
                            (ILEQ C (CHARCODE Z))
                            (SETQ C (IDIFFERENCE C 64]
              (\ILLEGAL.ARG C))
          (RETURN (ECHOCHAR C MODE TTBL])
```

(ECHOMODE

```
  [LAMBDA (FLG TTBL)                                          (* rmk%: " 8-FEB-80 11:57")
    (PROG1 (fetch ECHOFLG of (SETQ TTBL (\GTTERMTABLE TTBL)))
        (replace ECHOFLG of TTBL with (AND FLG T)))])
```

(GETECHOMODE

```
  [LAMBDA (TTBL)                                              (* lmm " 1-Jan-85 21:21")
    (fetch ECHOFLG of (\GTTERMTABLE TTBL T])
```

(GETCONTROL

```
  [LAMBDA (TTBL)                                              (* lmm " 1-Jan-85 21:21")
    (fetch CONTROLFLG of (\GTTERMTABLE TTBL T])
```

(GETTERMTABLE

```
  [LAMBDA (TTBL)
    (\GTTERMTABLE TTBL NIL])
```

(RAISE

```
  [LAMBDA (FLG TTBL)                                          (* bvm%: "14-Feb-85 00:17")
    (PROG1 (fetch RAISEFLG of (SETQ TTBL (\GTTERMTABLE TTBL)))
        (replace RAISEFLG of TTBL with (COND
                                          ((EQ FLG 0)
                                           0)
                                          (FLG T)))])
```

(GETRAISE

```
  [LAMBDA (TTBL)                                              (* lmm " 1-Jan-85 21:21")
    (fetch RAISEFLG of (\GTTERMTABLE TTBL T])
```

(RESETTERMTABLE

```
  [LAMBDA (TTBL FROM)                                         (* lmm "14-APR-81 14:34")
    (PROG ((FR (\GTTERMTABLE FROM T))
           (TT (\GTTERMTABLE TTBL)))
          (\COPYSYNTAX (fetch TERMSA of FR)
                 (fetch TERMSA of TT))
          (replace RAISEFLG of TT with (fetch RAISEFLG of FR))
          (replace DELCHARECHO of TT with (fetch DELCHARECHO of FR))
          (replace LINEDELETE of TT with (fetch LINEDELETE of FR))
          (replace 1STCHDEL of TT with (fetch 1STCHDEL of FR))
          (replace NTHCHDEL of TT with (fetch NTHCHDEL of FR))
          (replace POSTCHDEL of TT with (fetch POSTCHDEL of FR))
          (replace EMPTYCHDEL of TT with (fetch EMPTYCHDEL of FR))
          (replace CONTROLFLG of TT with (fetch CONTROLFLG of FR))
          (replace ECHOFLG of TT with (fetch ECHOFLG of FR))
          (RETURN TT])
```

(SETTERMTABLE

```
  [LAMBDA (TBL)                                               (* rmk%: " 8-FEB-80 12:16")
    (PROG1 \PRIMTERMTABLE
        [SETQ \PRIMTERMSA (fetch TERMSA of (SETQ \PRIMTERMTABLE (\GTTERMTABLE TBL]) ])
```

(TERMTABLEP

```
  [LAMBDA (TTBL)                                              (* rmk%: "20-FEB-80 12:29")
    (AND (type? TERMTABLEP TTBL)
         TTBL])
```

(\GETTERMSYNTAX

```
  [LAMBDA (C TBL)                                             (* rmk%: "24-APR-80 09:44")
    (\TERMCODETOCLASS (fetch TERMCLASS of (\SYNCODE (fetch TERMSA of TBL)
                                                   C])
```

(\GTTERMTABLE

```
  [LAMBDA (TTBL FLG)                                          (* lmm " 6-MAY-80 20:35")
    (COND
       ((type? TERMTABLEP TTBL)
        TTBL)
       ((NULL TTBL)
        \PRIMTERMTABLE)
```

```
       ((AND (EQ TTBL 'ORIG)
             FLG)
        \ORIGTERMTABLE)
       (T (LISPERROR "ILLEGAL TERMINAL TABLE" TTBL])
```

(\ORIGTERMTABLE

```
[LAMBDA NIL                                                    (* rrb " 5-Oct-85 10:33")
```

;; Creates the original terminal table

;; must be created with a hash table big enough to hold all of the indicates in character set 1 because this gets evaluated in the loadup before
;; HASHOVERFLOW is defined. rrb 5-oct-85

```
(PROG ((TBL (create TERMTABLEP
                    TERMSA _ (create CHARTABLE
                                     NSCHARHASH _ (\CREATENSCHARHASH 300))
                    DELCHARECHO _ 'ECHO
                    ECHOFLG _ T
                    LINEDELETE _ "##
                    "
                    1STCHDEL _ "\"
                    NTHCHDEL _ ""
                    POSTCHDEL _ "\"
                    EMPTYCHDEL _ "##
                    ")))
      (PROGN (\SETTERMSYNTAX (SELECTQ (SYSTEMTYPE)
                                  ((TENEX D)
                                      (CHARCODE ^A))
                                  ((JERICHO VAX TOPS-20)
                                      (CHARCODE DEL))
                                  (SHOULDNT))
             'CHARDELETE TBL)
             (\SETTERMSYNTAX (CHARCODE ^H)
             'CHARDELETE TBL)                                  ; Added ^H as a CHARDELETE character 9/30/85
             (\SETTERMSYNTAX (CHARCODE ^W)
             'WORDDELETE TBL)
             (\SETTERMSYNTAX (SELECTQ (SYSTEMTYPE)
                                  ((TENEX D)
                                      (CHARCODE ^Q))
                                  ((JERICHO VAX)
                                      (CHARCODE ^U))
                                  (SHOULDNT))
             'LINEDELETE TBL)
             (\SETTERMSYNTAX (CHARCODE ^R)
             'RETYPE TBL)
             (\SETTERMSYNTAX (CHARCODE ^V)
             'CTRLV TBL)
             (\SETTERMSYNTAX (CHARCODE EOL)
             'WAKEUPCHAR TBL)
             (for C
                in (CHARCODE (SPACE TAB ! @ %# $ ~ & * - = + %| { } ^ _ %: ; < > %, %. ? /))
                do (\SETTERMSYNTAX C 'WORDSEPR TBL)))
      (PROGN (ECHOCHAR (CHARCODE (NULL ^A ^B ^C ^D ^E ^F ^H ^K ^L ^N ^O ^P ^Q ^R ^S ^T ^U ^V ^W ^X ^Y ^Z
                                       ^\ ^%] ^^))
             'INDICATE TBL)
             (ECHOCHAR (CHARCODE (BELL TAB LF CR))
             'REAL TBL)
             (SELECTQ (SYSTEMTYPE)
                 (D (ECHOCHAR (CHARCODE (NULL ^A ^W ^Q ^R))
                         'IGNORE TBL)
                    (ECHOCHAR (CHARCODE (BELL TAB ESCAPE LF TENEXEOL))
                         'SIMULATE TBL))
                 (JERICHO (ECHOCHAR [CONSTANT (CONS ERASECHARCODE (CHARCODE (BELL TAB ESCAPE EOL]
                                'SIMULATE TBL))
                 (VAX (ECHOCHAR (CHARCODE (TAB ESCAPE EOL DEL))
                           'SIMULATE TBL))
                 NIL))
      (for C from 128 to \MAXTHINCHAR do (ECHOCHAR C 'REAL TBL))
      (for C from (CHARCODE 1,0) to (CHARCODE 1,377) do (ECHOCHAR C 'INDICATE TBL))
      (RETURN TBL])
```

(\SETTERMSYNTAX

```
[LAMBDA (C CLASS TBL)                                          (* rmk%: "26-Mar-85 23:45")
```

;; Changes the terminal syntax class for charcode C. Unlike Interlisp-10, does not turn off previous characters for CHARDELETE, etc. classes

```
(\SETSYNCODE (fetch TERMSA of TBL)
       C
       (create TERMCODE using (\SYNCODE (fetch TERMSA of TBL)
                                        C)
                              TERMCLASS _ (OR (\TERMCLASSTOCODE CLASS)
                                              (LISPERROR "ILLEGAL ARG" CLASS])
```

(\TERMCLASSTOCODE

```
[LAMBDA (CLASS)                                                (* rmk%: "11-FEB-82 21:24")
  (SELECTQ CLASS
      ((EOL WAKEUPCHAR)
```

```
          EOL.TC)
        (NONE NONE.TC)
        (CHARDELETE CHARDELETE.TC)
        (WORDDELETE WORDDELETE.TC)
        (WORDSEPR WORDSEPR.TC)
        (LINEDELETE LINEDELETE.TC)
        (RETYPE RETYPE.TC)
        ((CTRLV CNTRLV)
            CTRLV.TC)
        NIL])
```

(\TERMCODETOCLASS

```
  [LAMBDA (CODE)                                          (* rmk%: "11-FEB-82 21:24")
    (SELECTC CODE
         (EOL.TC 'EOL)
         (NONE.TC 'NONE)
         (CHARDELETE.TC
              'CHARDELETE)
         (WORDDELETE.TC
              'WORDDELETE)
         (WORDSEPR.TC 'WORDSEPR)
         (LINEDELETE.TC
              'LINEDELETE)
         (RETYPE.TC 'RETYPE)
         (CTRLV.TC 'CNTRLV)
         NIL])
```

(\LITCHECK

```
  [LAMBDA (X)                                             (* rmk%: "11-FEB-82 21:26")
    (COND
       ((EQ X 'BACKUP)                                    ; Means take terminal/implementation dependent backup action
        X)
       ((LITATOM X)
        (MKSTRING X))
       ((STRINGP X)
        (CONCAT X))
       (T (\ILLEGAL.ARG X])

)

(DECLARE%: DONTCOPY

;; FOLLOWING DEFINITIONS EXPORTED

(RPAQQ CCECHOMODES (REAL.CCE IGNORE.CCE SIMULATE.CCE INDICATE.CCE))

(DECLARE%: EVAL@COMPILE

(RPAQQ REAL.CCE 0)

(RPAQQ IGNORE.CCE 8)

(RPAQQ SIMULATE.CCE 16)

(RPAQQ INDICATE.CCE 24)

(CONSTANTS REAL.CCE IGNORE.CCE SIMULATE.CCE INDICATE.CCE)
)

(RPAQQ TERMCLASSES (NONE.TC EOL.TC CHARDELETE.TC WORDDELETE.TC WORDSEPR.TC LINEDELETE.TC RETYPE.TC CTRLV.TC))

(DECLARE%: EVAL@COMPILE

(RPAQQ NONE.TC 0)

(RPAQQ EOL.TC 1)

(RPAQQ CHARDELETE.TC 2)

(RPAQQ WORDDELETE.TC 6)

(RPAQQ WORDSEPR.TC 7)

(RPAQQ LINEDELETE.TC 3)

(RPAQQ RETYPE.TC 4)

(RPAQQ CTRLV.TC 5)

(CONSTANTS NONE.TC EOL.TC CHARDELETE.TC WORDDELETE.TC WORDSEPR.TC LINEDELETE.TC RETYPE.TC CTRLV.TC)
)

(DECLARE%: EVAL@COMPILE

(ACCESSFNS TERMCODE ((CCECHO (LOGAND DATUM 24))
```

```
                    (TERMCLASS (LOGAND DATUM 7)))          ; We assume that values are appropriately shifted
       (CREATE (LOGOR CCECHO TERMCLASS)))

(DATATYPE TERMTABLEP (TERMSA RAISEFLG DELCHARECHO LINEDELETE 1STCHDEL NTHCHDEL POSTCHDEL EMPTYCHDEL (CONTROLFLG
                                                                                                      FLAG)
                           (ECHOFLG FLAG))
       TERMSA _ (create CHARTABLE))
)

(/DECLAREDATATYPE 'TERMTABLEP '(POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER FLAG FLAG)
       ;; ---field descriptor list elided by lister---
       '16)
)

;; END EXPORTED DEFINITIONS

(/DECLAREDATATYPE 'TERMTABLEP '(POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER FLAG FLAG)
       ;; ---field descriptor list elided by lister---
       '16)

;; read tables

(DEFINEQ

(COPYREADTABLE
  [LAMBDA (RDTBL)                                          (* rmk%: " 2-FEB-80 12:26")
    (RESETREADTABLE (create READTABLEP)
           (\GTREADTABLE RDTBL T])

(FIND-READTABLE
  [LAMBDA (NAME)                                           (* bvm%: "27-Jul-86 15:53")
    (GETHASH NAME \READTABLEHASH])

(IN-READTABLE
  [LAMBDA (RDTBL)                                          (* bvm%: "27-Jul-86 15:55")
    (SETQ *READTABLE* (\GTREADTABLE RDTBL T])

(ESCAPE
  [LAMBDA (FLG RDTBL)                                      (* rmk%: " 1-FEB-80 13:12")
    (PROG1 (fetch ESCAPEFLG of (SETQ RDTBL (\GTREADTABLE RDTBL)))
        (replace ESCAPEFLG of RDTBL with (NEQ FLG NIL)))])

(GETBRK
  [LAMBDA (RDTBL)                                          (* rmk%: " 2-MAY-80 17:04")
    (GETSYNTAX 'BREAK RDTBL])

(GETREADTABLE
  [LAMBDA (RDTBL)                                          (* lmm%: 4-FEB-76 3 50)
    (\GTREADTABLE RDTBL])

(GETSEPR
  [LAMBDA (RDTBL)                                          (* rmk%: " 2-MAY-80 17:05")
    (GETSYNTAX 'SEPR RDTBL])

(READMACROS
  [LAMBDA (FLG RDTBL)                                      (* rmk%: " 1-FEB-80 13:11")
    (PROG1 (fetch READMACROFLG of (SETQ RDTBL (\GTREADTABLE RDTBL)))
        (replace READMACROFLG of RDTBL with (NEQ FLG NIL)))])

(READTABLEP
  [LAMBDA (RDTBL)                                          (* rmk%: "20-FEB-80 12:32")
    (AND (type? READTABLEP RDTBL)
         RDTBL])

(READTABLEPROP
  [LAMBDA ARGS                                             ; Edited  3-Apr-91 20:57 by jrb:
    (COND
       ((LESSP ARGS 2)
        (\ILLEGAL.ARG NIL))
       ((GREATERP ARGS 3)
        (\ILLEGAL.ARG (ARG ARGS 4)))
       (T (LET [(RDTBL (\GTREADTABLE (ARG ARGS 1)))
                (NEWVALUEP (EQ ARGS 3))
                (NEWVALUE (AND (EQ ARGS 3)
                               (ARG ARGS 3]
```

```
            (SELECTQ (ARG ARGS 2)
                 (NUMBERBASE (PROG1 (fetch (READTABLEP NUMBERBASE) of RDTBL)
                                    (COND
                                       (NEWVALUEP (replace (READTABLEP NUMBERBASE) of RDTBL with NEWVALUE)))))
                 (NAME [LET ((OLDNAME (fetch (READTABLEP READTBLNAME) of RDTBL)))
                           (PROG1 OLDNAME
                               (COND
                                  (NEWVALUEP (COND
                                                (OLDNAME (REMHASH OLDNAME \READTABLEHASH)))
                                             (replace (READTABLEP READTBLNAME) of RDTBL with NEWVALUE)
                                             (PUTHASH NEWVALUE RDTBL \READTABLEHASH)))))])
                 (COMMONLISP (PROG1 (fetch (READTABLEP COMMONLISP) of RDTBL)
                                    [COND
                                       (NEWVALUEP (replace (READTABLEP COMMONLISP) of RDTBL with NEWVALUE)
                                              (if NEWVALUE
                                                  then                ; COMMONLISP implies COMMONNUMSYNTAX and not
                                                                      ; USESILPACKAGE
                                                       (replace (READTABLEP COMMONNUMSYNTAX) of RDTBL
                                                          with T)
                                                       (replace (READTABLEP USESILPACKAGE) of RDTBL with NIL]))
                 (COMMONNUMSYNTAX
                      (PROG1 (fetch (READTABLEP COMMONNUMSYNTAX) of RDTBL)
                          (COND
                             (NEWVALUEP (replace (READTABLEP COMMONNUMSYNTAX) of RDTBL with NEWVALUE)))))
                 (USESILPACKAGE
                      (PROG1 (fetch (READTABLEP USESILPACKAGE) of RDTBL)
                          (COND
                             (NEWVALUEP (replace (READTABLEP USESILPACKAGE) of RDTBL with NEWVALUE)))))
                 (CASEINSENSITIVE
                      (PROG1 (fetch (READTABLEP CASEINSENSITIVE) of RDTBL)
                          (COND
                             (NEWVALUEP (replace (READTABLEP CASEINSENSITIVE) of RDTBL with NEWVALUE)))))
                 (LOWER/FLIPCASE
                      (PROG1 (fetch (READTABLEP LOWER/FLIPCASE) of RDTBL)
                          (COND
                             (NEWVALUEP (replace (READTABLEP LOWER/FLIPCASE) of RDTBL with NEWVALUE)))))
                 (ESCAPECHAR (PROG1 (fetch (READTABLEP ESCAPECHAR) of RDTBL)
                                    (COND
                                       (NEWVALUEP (\SETREADSYNTAX NEWVALUE 'ESCAPE RDTBL)
                                              (replace (READTABLEP ESCAPECHAR) of RDTBL with NEWVALUE)))))
                 (MULTIPLE-ESCAPECHAR
                      (PROG1 (fetch (READTABLEP MULTESCAPECHAR) of RDTBL)
                          (COND
                             (NEWVALUEP (\SETREADSYNTAX NEWVALUE 'MULTIPLE-ESCAPE RDTBL)
                                    (replace (READTABLEP MULTESCAPECHAR) of RDTBL with NEWVALUE)))))
                 (PACKAGECHAR (PROG1 (fetch (READTABLEP PACKAGECHAR) of RDTBL)
                                  (COND
                                     (NEWVALUEP (\SETREADSYNTAX NEWVALUE 'PACKAGEDELIM RDTBL)
                                            (replace (READTABLEP PACKAGECHAR) of RDTBL with NEWVALUE)))))
                 (HASHMACROCHAR
                      (PROG1 (fetch (READTABLEP HASHMACROCHAR) of RDTBL)
                          (COND
                             (NEWVALUEP (\SETREADSYNTAX NEWVALUE '(INFIX ALWAYS NONIMMEDIATE ESCQUOTE
                                                                         READVBAR)
                                                        RDTBL)
                                    (replace (READTABLEP HASHMACROCHAR) of RDTBL with NEWVALUE)))))
                 (\ILLEGAL.ARG (ARG ARGS 2])
```

(RESETREADTABLE

```
[LAMBDA (RDTBL FROM)                                    ; Edited 3-Apr-91 21:00 by jrb:
   [replace READMACROFLG of (SETQ RDTBL (\GTREADTABLE RDTBL)) with (fetch READMACROFLG of (SETQ FROM
                                                                                            (\GTREADTABLE FROM T]
   (replace ESCAPEFLG of RDTBL with (fetch ESCAPEFLG of FROM))
   (replace (READTABLEP COMMONLISP) of RDTBL with (fetch (READTABLEP COMMONLISP) of FROM))
   (replace (READTABLEP NUMBERBASE) of RDTBL with (fetch (READTABLEP NUMBERBASE) of FROM))
   (replace (READTABLEP CASEINSENSITIVE) of RDTBL with (fetch (READTABLEP CASEINSENSITIVE) of FROM))
   (replace (READTABLEP LOWER/FLIPCASE) of RDTBL with (fetch (READTABLEP LOWER/FLIPCASE) of FROM))
   (replace (READTABLEP COMMONNUMSYNTAX) of RDTBL with (fetch (READTABLEP COMMONNUMSYNTAX) of FROM))
   (replace (READTABLEP USESILPACKAGE) of RDTBL with (fetch (READTABLEP USESILPACKAGE) of FROM))
   (replace (READTABLEP HASHMACROCHAR) of RDTBL with (fetch (READTABLEP HASHMACROCHAR) of FROM))
   (replace (READTABLEP ESCAPECHAR) of RDTBL with (fetch (READTABLEP ESCAPECHAR) of FROM))
   (replace (READTABLEP MULTESCAPECHAR) of RDTBL with (fetch (READTABLEP MULTESCAPECHAR) of FROM))
   (replace (READTABLEP PACKAGECHAR) of RDTBL with (fetch (READTABLEP PACKAGECHAR) of FROM))
   (replace (READTABLEP DISPATCHMACRODEFS) of RDTBL with (COPY (fetch (READTABLEP DISPATCHMACRODEFS)
                                                                 of FROM)))

   ;; Placeholder.  If DISPATCHMACRODEFS ends up containing a CHARTABLE or a hash table, will have to do a REHASH or \COPYSYNTAX as
   ;; well

   [LET ((RDEFS (fetch (READTABLEP READMACRODEFS) of RDTBL))
         (FDEFS (fetch (READTABLEP READMACRODEFS) of FROM))
         N)
        (COND
           (RDEFS (CLRHASH RDEFS)))
        (AND FDEFS (REHASH FDEFS (OR RDEFS (replace (READTABLEP READMACRODEFS) of RDTBL
                                              with (HASHARRAY (HARRAYSIZE FDEFS)
                                                              7]
```

```
    (\COPYSYNTAX (fetch READSA of FROM)
          (fetch READSA of RDTBL))
    RDTBL])
```

(SETBRK

```
  [LAMBDA (LST FLG RDTBL)                                   (* rmk%: "13-AUG-81 00:01")
                                                            ; This is a very ugly def which needs to be cleaned up cause a
                                                            ; lot of people call SETBRK

    (COND
       [(EQ LST T)
        [MAPC (GETSYNTAX 'BREAK RDTBL)
              (FUNCTION (LAMBDA (X)
                        (SETSYNTAX X 'OTHER RDTBL]
        (MAPC (GETSYNTAX 'BREAK (COND
                                   ((EQ RDTBL T)
                                    'ORIG)
                                   (T T)))
              (FUNCTION (LAMBDA (X)
                        (SETSYNTAX X 'BREAK RDTBL]
       (T (SELECTQ FLG
              (NIL                                          ; reset
                   [MAPC (GETSYNTAX 'BREAK RDTBL)
                         (FUNCTION (LAMBDA (X)
                                     (OR (MEMB X LST)
                                         (SETSYNTAX X 'OTHER RDTBL]
                   [MAPC LST (FUNCTION (LAMBDA (X)
                                         (SETSYNTAX X 'BREAK RDTBL])
              (0                                            ; clear out lst
                 [MAPC LST (FUNCTION (LAMBDA (X)
                                       (SETSYNTAX X 'OTHER RDTBL])
              (1                                            ; add chars
                 [MAPC LST (FUNCTION (LAMBDA (X)
                                       (SETSYNTAX X 'BREAK RDTBL])
              NIL])
```

(SETREADTABLE

```
  [LAMBDA (RDTBL FLG)                                       (* bvm%: " 4-May-86 16:32")
    (PROG1 *READTABLE*
        (SETQ *READTABLE* (\GTREADTABLE RDTBL)))])
```

(SETSEPR

```
  [LAMBDA (LST FLG RDTBL)                                   (* rmk%: " 8-JUN-80 07:16")
                                                            ; This one also needs to be cleaned up

    (COND
       [(EQ LST T)
        [MAPC (GETSYNTAX 'SEPR RDTBL)
              (FUNCTION (LAMBDA (X)
                        (SETSYNTAX X 'OTHER RDTBL]
        (MAPC (GETSYNTAX 'SEPR (COND
                                  ((EQ RDTBL T)
                                   'ORIG)
                                  (T T)))
              (FUNCTION (LAMBDA (X)
                        (SETSYNTAX X 'SEPR RDTBL]
       (T (SELECTQ FLG
              (NIL                                          ; reset
                   [MAPC (GETSYNTAX 'SEPR RDTBL)
                         (FUNCTION (LAMBDA (X)
                                     (SETSYNTAX X 'OTHER RDTBL]
                   [MAPC LST (FUNCTION (LAMBDA (X)
                                         (SETSYNTAX X 'SEPR RDTBL])
              (0                                            ; clear out lst
                 [MAPC LST (FUNCTION (LAMBDA (X)
                                       (SETSYNTAX X 'OTHER RDTBL])
              (1                                            ; add chars
                 [MAPC LST (FUNCTION (LAMBDA (X)
                                       (SETSYNTAX X 'SEPR RDTBL])
              NIL])
```

(\GETREADSYNTAX

```
  [LAMBDA (C TBL)                                           (* bvm%: "30-Jun-86 17:49")
    (LET ((B (\SYNCODE (fetch READSA of TBL)
                    C)))
```

;; This will turn into a SELECTQ that keys off syntax code numbers and produces class tokens. The default clause at the end: if it's not a
;; built-in class, must be a macro

;; Sample code:

```
(* (SELECTQ B (0 (QUOTE OTHER))
(96 (QUOTE SEPRCHAR)) (112
(QUOTE BREAKCHAR)) (113 (QUOTE STRINGDELIM))
(114 (QUOTE LEFTPAREN)) (115
(QUOTE RIGHTPAREN)) (116 (QUOTE LEFTBRACKET))
(117 (QUOTE RIGHTBRACKET))
(70 (QUOTE ESCAPE)) (71 (QUOTE MULTIPLE-ESCAPE))
```

```
                                                                    (69 (QUOTE PACKAGEDELIM)) <default>))
        (\COMPUTED.FORM `(SELECTQ B
                              (\,@ [for PAIR in READCLASSTOKENS collect (LIST (EVAL (CADR PAIR))
                                                                              (KWOTE (CAR PAIR])
                              (LET ((E (\GETREADMACRODEF C TBL))
                                    KEY)
                                   `(,(fetch MACROTYPE of E)
                                     ,(fetch (CONTEXTS KEY) of (fetch MACROCONTEXT of B))
                                     ,@(AND (NEQ (SETQ KEY (fetch (WAKEUPS KEY) of (fetch WAKEUP of B)))
                                                 'NONIMMEDIATE)
                                            (LIST KEY))
                                     ,@(AND (NEQ (SETQ KEY (fetch (ESCAPES KEY) of (fetch ESCAPE of B)))
                                                 'ESCQUOTE)
                                            (LIST KEY))
                                     ,(fetch MACROFN of E])
```

(\GTREADTABLE

```
  [LAMBDA (X FLG)                                                   (* bvm%: " 5-May-86 11:05")
    (SELECTQ X
         ((NIL T)
             (\DTEST *READTABLE* 'READTABLEP))
         (\GTREADTABLE1 X FLG])
```

(\GTREADTABLE1

```
  [LAMBDA (X FLG)                                                   (* bvm%: "27-Jul-86 15:37")
    (COND
       ((type? READTABLEP X)
        X)
       ((AND FLG (GETHASH X \READTABLEHASH)))
       (T (LISPERROR "ILLEGAL READTABLE" X])
```

(\ORIGREADTABLE

```
  [LAMBDA NIL                                                       ; Edited  6-Nov-91 14:37 by jrb:
    ;; Creates a copy of the 'original' read-table.
    (LET ((FOO (\TYPEGLOBALVARIABLE 'READTABLEP T)))
         (DECLARE (SPECVARS FOO))
         (PROG [(TBL (create READTABLEP
                             READMACROFLG _ T
                             ESCAPEFLG _ T
                             NUMBERBASE _ 10
                             USESILPACKAGE _ T
                             ESCAPECHAR _ (CHARCODE %%)
                             PACKAGECHAR _ (PROGN
                                     ;; Need to have a character for package delimiter in all read tables, but for old read tables want one
                                     ;; that is unlikely to have appeared in a symbol in an old source file.  Also would like it to be a 7-bit
                                     ;; char, so we don't needlessly force MAKEFILE to produce binary files.
                                                  (CHARCODE "^^"))
                             HASHMACROCHAR _ (CHARCODE "|"]
          ;; Actually, '|' is not defined in ORIG table, but rather later.  But the radix printer and others want it, and this is better than nothing
               (SETSEPR (CHARCODE (SPACE TENEXEOL CR ^L LF TAB))
                        1 TBL)
               (\SETREADSYNTAX (CHARCODE %])
                        'RIGHTBRACKET TBL)
               (\SETREADSYNTAX (CHARCODE %[)
                        'LEFTBRACKET TBL)
               (\SETREADSYNTAX (CHARCODE %))
                        'RIGHTPAREN TBL)
               (\SETREADSYNTAX (CHARCODE %()
                        'LEFTPAREN TBL)
               (\SETREADSYNTAX (CHARCODE %%)
                        'ESCAPE TBL)
               (\SETREADSYNTAX (CHARCODE %")
                        'STRINGDELIM TBL)
               (\SETREADSYNTAX 167 'PACKAGEDELIM TBL)           ; Old choice for package delim char: the NS section symbol.
                                                                ; Keep for compatibility with Lyric Beta files
               (\SETREADSYNTAX (CHARCODE "^^")
                        'PACKAGEDELIM TBL)
               (RETURN TBL])
```

(\READCLASSTOCODE

```
  [LAMBDA (CLASS)                                                   (* bvm%: " 9-Jul-85 00:43")
```

;;; This turns into a SELECTQ that goes from CLASS token to numeric code

```
    (\COMPUTED.FORM `(SELECTQ CLASS
                         (\,@ READCLASSTOKENS)
                         (SEPR                                  ; Synonym for SEPRCHAR
                               SEPRCHAR.RC)
                         NIL])
```

(\SETMACROSYNTAX

```
[LAMBDA (C CLASS TBL)                                   (* rmk%: " 3-Jan-84 13:20")
  (OR (AND (FMEMB (CAR CLASS)
                  '(MACRO SPLICE INFIX))
           (CDR CLASS))
      (\ILLEGAL.ARG CLASS))
  (PROG (CONTEXT WAKEUP ESCAPE (LST CLASS)
                (A (fetch READMACRODEFS of TBL)))
    LP  (COND
           ([CDR (SETQ LST (LISTP (CDR LST]
            (OR [AND (NULL CONTEXT)
                     (SETQ CONTEXT (fetch (CONTEXTS VAL) of (CAR LST]
                [AND (NULL WAKEUP)
                     (SETQ WAKEUP (fetch (WAKEUPS VAL) of (CAR LST]
                [AND (NULL ESCAPE)
                     (SETQ ESCAPE (fetch (ESCAPES VAL) of (CAR LST]
                (\ILLEGAL.ARG CLASS))
            (GO LP)))
           (OR (LISTP LST)
               (\ILLEGAL.ARG CLASS))
           [COND
              (A ;; This hack guarantees that the hasharray will not overflow and cause an error in the uninterruptable PUTHASH below.  If it didn't
                 ;; already have a value for C, then the macro bits are not set in C's syntax code, so the T value is harmless.
                 (OR (GETHASH C A)
                     (PUTHASH C T A)))
              (T (replace READMACRODEFS of TBL with (SETQ A (HASHARRAY 7 7]
           (UNINTERRUPTABLY
               (PUTHASH C (create READMACRODEF
                                  MACROTYPE _ (CAR CLASS)
                                  MACROFN _ (CAR LST))
                          A)
               (\SETSYNCODE (fetch READSA of TBL)
                      C
                      (LOGOR (OR CONTEXT ALWAYS.RMC)
                             (OR ESCAPE ESC.RME)
                             (OR WAKEUP NONIMMEDIATE.RMW)))))])
```

(\SETREADSYNTAX

```
[LAMBDA (C CLASS TBL)                                   (* bvm%: " 8-Mar-86 16:37")
  (PROG ((OLDSYNTAX (\SYNCODE (fetch (READTABLEP READSA) of TBL)
                              C))
         TEM)
        [COND
           ((EQ CLASS 'BREAK)
            (COND
               ((fetch BREAK of OLDSYNTAX)
                (RETURN))
               (T (SETQ CLASS 'BREAKCHAR]                ; If already a BREAK character but also something else, like
                                                         ; LPAR, leave it alone
        (COND
           ((LISTP CLASS)
            (\SETMACROSYNTAX C CLASS TBL))
           ((SETQ TEM (\READCLASSTOCODE CLASS))
            (UNINTERRUPTABLY
                [COND
                   ((fetch MACROP of OLDSYNTAX)          ; No longer a macro
                    (REMHASH C (fetch READMACRODEFS of TBL]
                (\SETSYNCODE (fetch READSA of TBL)
                       C TEM)))
           (T (\ILLEGAL.ARG CLASS])
```

(\READTABLEP.DEFPRINT

```
[LAMBDA (RDTBL STREAM)                                  (* bvm%: "13-Oct-86 17:32")
  ;; Print read table as, for example, #<ReadTable name/76,5432>
  (LET ((NAME (fetch (READTABLEP READTBLNAME) of RDTBL)))
       [.SPACECHECK. STREAM (IPLUS (CONSTANT (NCHARS "<ReadTable />"))
                                   (PROGN                ; Longest address is '177,177777'
                                          10)
                                   (COND
                                      (NAME (NCHARS NAME))
                                      (T 0]
       (\OUTCHAR STREAM (fetch (READTABLEP HASHMACROCHAR) of *READTABLE*))
       (\SOUT "<ReadTable" STREAM)
       (COND
          (NAME (\OUTCHAR STREAM (CHARCODE SPACE))
                (\SOUT (MKSTRING NAME)
                       STREAM)))
       (\OUTCHAR STREAM (CHARCODE /))
       (\PRINTADDR RDTBL STREAM)
       (\OUTCHAR STREAM (CHARCODE >))
       T])
```

```
)

(PUTPROPS READTABLEPROP ARGNAMES (RDTBL PROP NEWVALUE))

(DECLARE%: EVAL@COMPILE DONTCOPY

(RPAQQ READCLASSTOKENS
       ((OTHER 0)
        (SEPRCHAR (LOGOR ESCAPEBIT STOPATOMBIT 0))
        (BREAKCHAR (LOGOR ESCAPEBIT STOPATOMBIT BREAKBIT 0))
        (STRINGDELIM (LOGOR ESCAPEBIT STOPATOMBIT BREAKBIT 1))
        (LEFTPAREN (LOGOR ESCAPEBIT STOPATOMBIT BREAKBIT 2))
        (RIGHTPAREN (LOGOR ESCAPEBIT STOPATOMBIT BREAKBIT 3))
        (LEFTBRACKET (LOGOR ESCAPEBIT STOPATOMBIT BREAKBIT 4))
        (RIGHTBRACKET (LOGOR ESCAPEBIT STOPATOMBIT BREAKBIT 5))
        (ESCAPE (LOGOR ESCAPEBIT INNERESCAPEBIT 6))
        (MULTIPLE-ESCAPE (LOGOR ESCAPEBIT INNERESCAPEBIT 7))
        (PACKAGEDELIM (LOGOR ESCAPEBIT INNERESCAPEBIT 1))))

(RPAQ READCLASSES [MAPCAR READCLASSTOKENS (FUNCTION (LAMBDA (PAIR)
                                                      (LIST (PACK* (CAR PAIR)
                                                                   ".RC")
                                                            (CADR PAIR])

(DECLARE%: EVAL@COMPILE

(PUTPROPS \COMPUTED.FORM MACRO [X (CONS 'PROGN (MAPCAR X (FUNCTION EVAL])
)

(DECLARE%: EVAL@COMPILE

[ACCESSFNS CONTEXTS ((KEY (SELECTC DATUM
                             (ALWAYS.RMC 'ALWAYS)
                             (FIRST.RMC 'FIRST)
                             (ALONE.RMC 'ALONE)
                             NIL))
                    (VAL (SELECTQ DATUM
                             (ALWAYS ALWAYS.RMC)
                             (FIRST FIRST.RMC)
                             (ALONE ALONE.RMC)
                             NIL]

[ACCESSFNS ESCAPES ((KEY (SELECTC DATUM
                           (ESC.RME 'ESCQUOTE)
                           (NOESC.RME 'NOESCQUOTE)
                           NIL))
                   (VAL (SELECTQ DATUM
                           ((ESCQUOTE ESC)
                               ESC.RME)
                           ((NOESCQUOTE NOESC)
                               NOESC.RME)
                           NIL]

[ACCESSFNS WAKEUPS ((KEY (SELECTC DATUM
                           (IMMEDIATE.RMW
                               'IMMEDIATE)
                           (NONIMMEDIATE.RMW
                               'NONIMMEDIATE)
                           NIL))
                   (VAL (SELECTQ DATUM
                           ((IMMEDIATE IMMED WAKEUP)
                               IMMEDIATE.RMW)
                           ((NONIMMEDIATE NONIMMED NOWAKEUP)
                               NONIMMEDIATE.RMW)
                           NIL]
)

;; FOLLOWING DEFINITIONS EXPORTED

(DECLARE%: EVAL@COMPILE

(PUTPROPS \GETREADMACRODEF MACRO ((C TBL)
                                  (GETHASH C (fetch READMACRODEFS of TBL))))

(PUTPROPS \GTREADTABLE MACRO [ARGS (COND
                                     [(LITATOM (CAR ARGS))
                                      (SUBPAIR '(X . FLG)
                                             ARGS
                                             '(SELECTQ X
                                                   ((NIL T)
                                                       (\DTEST *READTABLE* 'READTABLEP))
                                                   (\GTREADTABLE1 X . FLG]
                                     (T 'IGNOREMACRO])

(PUTPROPS \GTREADTABLE1 DMACRO [ARGS (COND
                                       [(NULL (CDR ARGS))
                                        (LIST '\DTEST (CAR ARGS)
```

```
                                                ''READTABLEP]
                                            (T 'IGNOREMACRO])
)

(DECLARE%: EVAL@COMPILE

(RPAQQ MACROBIT 8)

(RPAQQ BREAKBIT 16)

(RPAQQ STOPATOMBIT 32)

(RPAQQ ESCAPEBIT 64)

(RPAQQ INNERESCAPEBIT 4)

(CONSTANTS MACROBIT BREAKBIT STOPATOMBIT ESCAPEBIT INNERESCAPEBIT)
)

(RPAQQ READCODEMASKS ((CONTEXTMASK (LOGOR MACROBIT STOPATOMBIT BREAKBIT 1))
                      (WAKEUPMASK (LOGOR MACROBIT 2))))

(DECLARE%: EVAL@COMPILE

(RPAQ CONTEXTMASK (LOGOR MACROBIT STOPATOMBIT BREAKBIT 1))

(RPAQ WAKEUPMASK (LOGOR MACROBIT 2))

(CONSTANTS (CONTEXTMASK (LOGOR MACROBIT STOPATOMBIT BREAKBIT 1))
       (WAKEUPMASK (LOGOR MACROBIT 2)))
)

(RPAQQ READMACROCONTEXTS ((ALWAYS.RMC (LOGOR MACROBIT STOPATOMBIT BREAKBIT 0))
                          (FIRST.RMC (LOGOR MACROBIT 0))
                          (ALONE.RMC (LOGOR MACROBIT 1))))

(DECLARE%: EVAL@COMPILE

(RPAQ ALWAYS.RMC (LOGOR MACROBIT STOPATOMBIT BREAKBIT 0))

(RPAQ FIRST.RMC (LOGOR MACROBIT 0))

(RPAQ ALONE.RMC (LOGOR MACROBIT 1))

(CONSTANTS (ALWAYS.RMC (LOGOR MACROBIT STOPATOMBIT BREAKBIT 0))
       (FIRST.RMC (LOGOR MACROBIT 0))
       (ALONE.RMC (LOGOR MACROBIT 1)))
)

(RPAQQ READCLASSES
       ((OTHER.RC 0)
        (SEPRCHAR.RC (LOGOR ESCAPEBIT STOPATOMBIT 0))
        (BREAKCHAR.RC (LOGOR ESCAPEBIT STOPATOMBIT BREAKBIT 0))
        (STRINGDELIM.RC (LOGOR ESCAPEBIT STOPATOMBIT BREAKBIT 1))
        (LEFTPAREN.RC (LOGOR ESCAPEBIT STOPATOMBIT BREAKBIT 2))
        (RIGHTPAREN.RC (LOGOR ESCAPEBIT STOPATOMBIT BREAKBIT 3))
        (LEFTBRACKET.RC (LOGOR ESCAPEBIT STOPATOMBIT BREAKBIT 4))
        (RIGHTBRACKET.RC (LOGOR ESCAPEBIT STOPATOMBIT BREAKBIT 5))
        (ESCAPE.RC (LOGOR ESCAPEBIT INNERESCAPEBIT 6))
        (MULTIPLE-ESCAPE.RC (LOGOR ESCAPEBIT INNERESCAPEBIT 7))
        (PACKAGEDELIM.RC (LOGOR ESCAPEBIT INNERESCAPEBIT 1))))

(DECLARE%: EVAL@COMPILE

(RPAQQ OTHER.RC 0)

(RPAQ SEPRCHAR.RC (LOGOR ESCAPEBIT STOPATOMBIT 0))

(RPAQ BREAKCHAR.RC (LOGOR ESCAPEBIT STOPATOMBIT BREAKBIT 0))

(RPAQ STRINGDELIM.RC (LOGOR ESCAPEBIT STOPATOMBIT BREAKBIT 1))

(RPAQ LEFTPAREN.RC (LOGOR ESCAPEBIT STOPATOMBIT BREAKBIT 2))

(RPAQ RIGHTPAREN.RC (LOGOR ESCAPEBIT STOPATOMBIT BREAKBIT 3))

(RPAQ LEFTBRACKET.RC (LOGOR ESCAPEBIT STOPATOMBIT BREAKBIT 4))

(RPAQ RIGHTBRACKET.RC (LOGOR ESCAPEBIT STOPATOMBIT BREAKBIT 5))

(RPAQ ESCAPE.RC (LOGOR ESCAPEBIT INNERESCAPEBIT 6))

(RPAQ MULTIPLE-ESCAPE.RC (LOGOR ESCAPEBIT INNERESCAPEBIT 7))

(RPAQ PACKAGEDELIM.RC (LOGOR ESCAPEBIT INNERESCAPEBIT 1))

(CONSTANTS (OTHER.RC 0)
```

```
       (SEPRCHAR.RC (LOGOR ESCAPEBIT STOPATOMBIT 0))
       (BREAKCHAR.RC (LOGOR ESCAPEBIT STOPATOMBIT BREAKBIT 0))
       (STRINGDELIM.RC (LOGOR ESCAPEBIT STOPATOMBIT BREAKBIT 1))
       (LEFTPAREN.RC (LOGOR ESCAPEBIT STOPATOMBIT BREAKBIT 2))
       (RIGHTPAREN.RC (LOGOR ESCAPEBIT STOPATOMBIT BREAKBIT 3))
       (LEFTBRACKET.RC (LOGOR ESCAPEBIT STOPATOMBIT BREAKBIT 4))
       (RIGHTBRACKET.RC (LOGOR ESCAPEBIT STOPATOMBIT BREAKBIT 5))
       (ESCAPE.RC (LOGOR ESCAPEBIT INNERESCAPEBIT 6))
       (MULTIPLE-ESCAPE.RC (LOGOR ESCAPEBIT INNERESCAPEBIT 7))
       (PACKAGEDELIM.RC (LOGOR ESCAPEBIT INNERESCAPEBIT 1)))
)

(RPAQQ READMACROWAKEUPS ((IMMEDIATE.RMW (LOGOR MACROBIT 2))
                         (NONIMMEDIATE.RMW (LOGOR MACROBIT 0))))

(DECLARE%: EVAL@COMPILE

(RPAQ IMMEDIATE.RMW (LOGOR MACROBIT 2))

(RPAQ NONIMMEDIATE.RMW (LOGOR MACROBIT 0))

(CONSTANTS (IMMEDIATE.RMW (LOGOR MACROBIT 2))
       (NONIMMEDIATE.RMW (LOGOR MACROBIT 0)))
)

(RPAQQ READMACROESCAPES ((ESC.RME ESCAPEBIT)
                         (NOESC.RME 0)))

(DECLARE%: EVAL@COMPILE

(RPAQ ESC.RME ESCAPEBIT)

(RPAQQ NOESC.RME 0)

(CONSTANTS (ESC.RME ESCAPEBIT)
       (NOESC.RME 0))
)

(DECLARE%: EVAL@COMPILE

[ACCESSFNS READCODE ((ESCAPE (LOGAND DATUM ESCAPEBIT))
                     (ESCQUOTE (BITTEST DATUM ESCAPEBIT))
                     (STOPATOM (BITTEST DATUM STOPATOMBIT))
                     (INNERESCQUOTE (BITTEST DATUM (LOGOR STOPATOMBIT INNERESCAPEBIT)))
                     (MACROCONTEXT (LOGAND DATUM CONTEXTMASK))
                     (MACROP (BITTEST DATUM MACROBIT))
                     (WAKEUP (LOGAND DATUM WAKEUPMASK))
                     (BREAK (BITTEST DATUM BREAKBIT]

(RECORD READMACRODEF (MACROTYPE . MACROFN))

(DATATYPE READTABLEP ((READSA POINTER)                          ; A CHARTABLE defining syntax of each char
                      (READMACRODEFS POINTER)                   ; A hash table associating macro chars with macro definitions
                      (READMACROFLG FLAG)                       ; True if read macros are enabled (turned off by Interlisp's crufty
                                                                ; READMACROS function)
                      (ESCAPEFLG FLAG)                          ; True if the char(s) with escape syntax are enabled (turned off
                                                                ; by Interlisp's crufty ESCAPE function)
                      (COMMONLISP FLAG)                         ; True if table is a Common Lisp read table and hence must obey
                                                                ; Common Lisp syntax rules
                      (NUMBERBASE BITS 5)                       ; Not used
                      (CASEINSENSITIVE FLAG)                    ; If true, unescaped lowercase chars are converted to uppercase
                                                                ; in symbols
                      (COMMONNUMSYNTAX FLAG)                    ; True if number notation includes Common Lisp numbers:
                                                                ; rationals as a/b, and the dfls exponent markers
                      (USESILPACKAGE FLAG)                      ; If true, IL:READ ignores *PACKAGE* and reads in the IL
                                                                ; package
                      (LOWER/FLIPCASE FLAG)                     ; This flag plus CASEINSENSITIVE = CL:READTABLE-CASE as
                                                                ; required by CLtL2, as follows:
                                                                ; :PRESERVE - (not ci) and (not lf)
                                                                ; :INVERT - (not ci) and lf
                                                                ; :UPCASE - ci and (not lf)
                                                                ; :DOWNCASE - ci and lf
                                                                ; You are urged to use CL:READTABLE-CASE and its setf
                                                                ; instead of accessing these flags directly
                      (NIL 4 FLAG)
                      (DISPATCHMACRODEFS POINTER)               ; An a-list of dispatching macro char and its dispatch definitions
                      (HASHMACROCHAR BYTE)                      ; The character code used in this read table for the # dispatch
                                                                ; macro
                      (ESCAPECHAR BYTE)                         ; The character code used in this read table for single escape
                      (MULTESCAPECHAR BYTE)                     ; The character code used in this read table for multiple escape
                      (PACKAGECHAR BYTE)                        ; The character code used in this read table for package delimiter
                      (READTBLNAME POINTER)                     ; The canonical 'name' of this read table

                      )
       READSA _ (create CHARTABLE))
)
```

```
(/DECLAREDATATYPE 'READTABLEP
       '(POINTER POINTER FLAG FLAG FLAG (BITS 5)
              FLAG FLAG FLAG FLAG FLAG FLAG FLAG FLAG POINTER BYTE BYTE BYTE BYTE POINTER)
       ;; ---field descriptor list elided by lister---
       '10)

;; END EXPORTED DEFINITIONS

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS \ORIGREADTABLE \READTABLEHASH \ORIGTERMTABLE)
)
)

(/DECLAREDATATYPE 'READTABLEP
       '(POINTER POINTER FLAG FLAG FLAG (BITS 5)
              FLAG FLAG FLAG FLAG FLAG FLAG FLAG FLAG POINTER BYTE BYTE BYTE BYTE POINTER)
       ;; ---field descriptor list elided by lister---
       '10)

(DEFINEQ

(\ATBLSET
  [LAMBDA NIL                                                ; Edited  3-Dec-86 18:07 by Pavel
    (DECLARE (GLOBALVARS \ORIGREADTABLE \ORIGTERMTABLE))
    (COND
       ((NULL (BOUNDP '\PRIMREADTABLE))
        (initrecord CHARTABLE)
        ;; Read tables
        (SETQ \READTABLEHASH (HASHARRAY 20 NIL (FUNCTION STRING-EQUAL-HASHBITS)
                                  (FUNCTION STRING-EQUAL)))
        (LET (TRDTBL NEW-IL-RDTBL)
             (PROGN                                          ; The ORIG read table
                    (SETQ \ORIGREADTABLE (\ORIGREADTABLE))
                    (READTABLEPROP \ORIGREADTABLE 'NAME 'ORIG))
             (PROGN                                          ; The old Interlisp T read table.  May not have a use for this any
                                                             ; more
                    (SETQ TRDTBL (COPYREADTABLE \ORIGREADTABLE))
                    (SETSYNTAX (CHARCODE "|")
                           '(MACRO READVBAR)
                           TRDTBL)
                    (SETSYNTAX (CHARCODE "`")
                           '(MACRO FIRST READBQUOTE)
                           TRDTBL)
                    (SETSYNTAX (CHARCODE ",")
                           '(MACRO FIRST READBQUOTECOMMA)
                           TRDTBL)
                    (SETSYNTAX (CHARCODE "'")
                           '(MACRO FIRST READQUOTE)
                           TRDTBL)
                    (READTABLEPROP TRDTBL 'NAME "OLD-INTERLISP-T")
                    (PROGN                                   ; Temporary
                           (SETTOPVAL '%#CURRENTRDTBL# TRDTBL)))
             (PROGN                                          ; The old FILERDTBL
                    (SETQ FILERDTBL (COPYREADTABLE \ORIGREADTABLE))
                    (SETSYNTAX (CHARCODE "|")
                           TRDTBL FILERDTBL)
                    (READTABLEPROP FILERDTBL 'NAME "OLD-INTERLISP-FILE")
                    (SETQ *OLD-INTERLISP-READ-ENVIRONMENT* (create READER-ENVIRONMENT
                                                              REREADTABLE _ FILERDTBL
                                                              REBASE _ 10))
                                                             ; need this to read files in the loadup

                    )
             (PROGN (SETQ NEW-IL-RDTBL (COPYREADTABLE TRDTBL)) ; The new Interlisp read table is more common lispy
                    (READTABLEPROP NEW-IL-RDTBL 'MULTIPLE-ESCAPECHAR (CHARCODE "|"))
                    (READTABLEPROP NEW-IL-RDTBL 'HASHMACROCHAR (CHARCODE "#"))
                    (SET-DEFAULT-HASHMACRO-SETTINGS NEW-IL-RDTBL)
                    (READTABLEPROP NEW-IL-RDTBL 'COMMONNUMSYNTAX T)
                    (READTABLEPROP NEW-IL-RDTBL 'USESILPACKAGE NIL)
                    (READTABLEPROP NEW-IL-RDTBL 'NAME "INTERLISP")
                    (for I from 1 to 26 do (SETSYNTAX I 'SEPRCHAR FILERDTBL)
                                                             ; Make font switch chars seprs
                                           (SETSYNTAX I 'SEPRCHAR NEW-IL-RDTBL))
                    (SETQ *READTABLE* NEW-IL-RDTBL))
        ;; Make ^Y like #. in the old T readtable and the new INTERLISP one.
        (SETSYNTAX (CHARCODE ^Y)
               '[MACRO ALWAYS (LAMBDA (FILE RDTBL)
                                 (EVAL (READ FILE RDTBL]
               TRDTBL)
        (SETSYNTAX (CHARCODE ^Y)
               TRDTBL NEW-IL-RDTBL)
        (DEFPRINT 'READTABLEP '\READTABLEP.DEFPRINT))
```

```
        ;; Terminal tables
        (SETQ \ORIGTERMTABLE (\ORIGTERMTABLE))
        (SETQ \PRIMTERMTABLE (COPYTERMTABLE \ORIGTERMTABLE))
        (SETQ \PRIMTERMSA (fetch TERMSA of \PRIMTERMTABLE))
        (PUTD '\ATBLSET)
        (PUTD '\ORIGTERMTABLE)
        NIL])

)

(/DECLAREDATATYPE 'READER-ENVIRONMENT '(POINTER POINTER POINTER POINTER)
       ;; ---field descriptor list elided by lister---
       '8)

;; Definition is on CMLREAD, need it here to initialize *OLD-INTERLISP-READ-ENVIRONMENT*

(DEFINEQ

(MAKE-READER-ENVIRONMENT
  [LAMBDA (PACKAGE READTABLE BASE)                          ; Edited 18-Dec-86 18:28 by bvm:
    (create READER-ENVIRONMENT
            REPACKAGE _ (COND
                          (PACKAGE (\DTEST PACKAGE 'PACKAGE))
                          (T *PACKAGE*))
            REREADTABLE _ (COND
                            (READTABLE (\DTEST READTABLE 'READTABLEP))
                            (T *READTABLE*))
            REBASE _ (COND
                       (BASE (\CHECKRADIX BASE))
                       (T *PRINT-BASE*])

(EQUAL-READER-ENVIRONMENT
  [LAMBDA (ENV1 ENV2)                                       (* bvm%: "31-Jul-86 12:54")
    (AND (EQ (fetch (READER-ENVIRONMENT REREADTABLE) of ENV1)
             (fetch (READER-ENVIRONMENT REREADTABLE) of ENV2))
         (EQ (fetch (READER-ENVIRONMENT REPACKAGE) of ENV1)
             (fetch (READER-ENVIRONMENT REPACKAGE) of ENV2))
         (EQ (fetch (READER-ENVIRONMENT REBASE) of ENV1)
             (fetch (READER-ENVIRONMENT REBASE) of ENV2])

(SET-READER-ENVIRONMENT
  [LAMBDA (ENV)                                             (* bvm%: "28-Aug-86 17:44")

;;; Sets the reader environment variables from ENV.  Should usually only be called inside a WITH-READER-ENVIRONMENT.

    [SETQ *PACKAGE* (ffetch REPACKAGE of (\DTEST ENV 'READER-ENVIRONMENT]
    (SETQ *READTABLE* (ffetch REREADTABLE of ENV))
    (SETQ *READ-BASE* (SETQ *PRINT-BASE* (ffetch REBASE of ENV)))
    ENV])

)

(RPAQ? *LISP-PACKAGE* )

(RPAQ? *INTERLISP-PACKAGE* )

(RPAQ? *KEYWORD-PACKAGE* )

(DECLARE%: DONTEVAL@LOAD DOCOPY

(\ATBLSET)
)

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(LOCALVARS . T)
)

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

(ADDTOVAR NLAMA )

(ADDTOVAR NLAML )

(ADDTOVAR LAMA READTABLEPROP)
)

(PUTPROPS ATBL COPYRIGHT ("Venue & Xerox Corporation" 1982 1983 1984 1985 1986 1987 1990 1991))
```

FUNCTION INDEX

CONSTANT INDEX

VARIABLE INDEX

RECORD INDEX

MACRO INDEX

PROPERTY INDEX