File created: 12-Feb-2021 17:00:02 **{DSK}<home>larry>ilisp>medley>sources>ASKUSER.;8**

changes to: (VARS ASKUSERCOMS)

previous date: 10-Aug-2020 21:18:50 {DSK}<home>larry>ilisp>medley>sources>ASKUSER.;7

Read Table: INTERLISP

Package: INTERLISP

Format: XCCS

```
;;
;; Copyright (c) 1986, 1987, 1990, 2020, 2021 by Venue & Xerox Corporation.

(RPAQQ ASKUSERCOMS
       [(FNS ASKUSER ASKUSERLOOKUP ASKUSERCHAR ASKUSER$ ASKUSER1 ASKUSERSETUP ASKUSEREXPLAIN ASKUSERPRIN1
             MAKEKEYLST)
        ;; RMK: Avoid literal CR's on files.
        (INITVARS [DEFAULTKEYLST (LIST [LIST 'Y (CONCAT "es" (CHARACTER (CHARCODE EOL]
                                       (LIST 'N (CONCAT "o" (CHARACTER (CHARCODE EOL]
               (ASKUSERTTBL (COPYTERMTABLE)))
        (DECLARE%: DONTEVAL@LOAD DOCOPY (P (CONTROL T ASKUSERTTBL)
                                           (ECHOMODE NIL ASKUSERTTBL)))
        (DECLARE%: DOEVAL@COMPILE DONTCOPY (RECORDS ASKUSER OPTIONS)
               (GLOBALVARS DEFAULTKEYLST ASKUSERTTBL))
        (DECLARE%: DONTEVAL@LOAD DOEVAL@COMPILE DONTCOPY COMPILERVARS (ADDVARS (NLAMA)
                                                                              (NLAML)
                                                                              (LAMA])

(DEFINEQ

(ASKUSER
  [LAMBDA (WAIT DEFAULT MESS KEYLST TYPEAHEAD LISPXPRNTFLG OPTIONSLST FILE)
    (DECLARE (SPECVARS LISPXPRNTFLG OPTIONSLST FILE))          ; Edited 10-Aug-2020 20:58 by rmk:
                                                               ; Edited 10-Aug-87 15:45 by jop
                                                               ; reads characters one at a time echoing and/or prompting as
                                                               ; indicated by KEYLST

    ;; RMK: Changed literal ^M's and spaces to use the (CHARACTER (CHARCODE construct), for readability and to allow for EOL conversion from
    ;; other file systems.  We want this always to be the internal EOL (=CR).

    (RESETLST
        (COND
           ((NULL KEYLST)                                      ; Yes, no recognized without conforimation
            (SETQ KEYLST DEFAULTKEYLST)))
        (PROG [OLDTTBL CHAR TEM KEYLST1 ANSWER BUFS (ORIGKEYLST KEYLST)
                       (ORIGMESS MESS)
                       (ORIGDEFAULT DEFAULT)
                       (NC 1)
                       KEY PROMPTSTRING OPTIONS NOECHOFLG CONFIRMFLG NOCASEFLG PRINTLST ECHOEDFLG
                       (EOL (CHARACTER (CHARCODE EOL)))
                       (SPACE (CHARACTER (CHARCODE SPACE]
              (COND
                 ((NULL FILE)
                  (SETQ FILE T))
                 ((NEQ FILE T)
                  (GO MESS)))
              (SETQ OLDTTBL (GETTERMTABLE))
              (RESETSAVE (SETTERMTABLE ASKUSERTTBL))

        ;; ASKUSERTTBL has (CONTROL T) and (RAISE T) performed.  The latter means that if the user types lower case characters, they are
        ;; converted to uppercase.  Note however that this will recognize lower case y and n.  This is so the caller can provide y or n as a default,
        ;; and distinguish the default cse from the case where the user types lowercase y or n (which will be converted to uppercase automatically
        ;; by the terminal table) ASKUSERTTBL also has (ECHOMODE NIL) performed so can handle mistypings and confirations properly.
                                                               ; File can be a file name or a string
              (COND
                 (TYPEAHEAD                                    ; TYPEAHEAD permitted
                        (SETQ TYPEAHEAD (READP T))             ; used in case there is a mistake.  in this case all typeahead is
                                                               ; restored.
                        (GO MESS)))
              (LINBUF)
              (SYSBUF)
              (SETQ BUFS (CLBUFS NIL T READBUF))

        ;; Clear and save typeahead.  This call to CLBUFS will ring the bells if there is any typeahead to warn the user to stop typing.

              (COND
                 [(LISTP MESS)
                  (ASKUSERPRIN1 (CAR MESS))
                  (COND
                     ((SETQ MESS (CDR MESS))
                      (ASKUSERPRIN1 " "))
                     (T (ASKUSERPRIN1 " ? "]
                 (MESS (ASKUSERPRIN1 MESS)
                       (SETQ MESS NIL)))

        ;; The problem with user interactions such as this where typeahead is not allowed is that we have no way of knowing WHEN the user types
        ;; something, i.e.  if he typed it after seeing part of the message or no, without doing a DOBE before doing any printing, and this is not
```

```
;; desirable as it produces a noticeable snag in teletype output.  --- Therefore what we do is the following: all typeahead before the call to
;; ASKUSER is cleared and saved for later restoration, and n the event ther is any typeahead, bells are rung to warn the user to stop typing.
;; (this is done by the call to CLBUFS above.) --- After that we print something, either the first part of the message or the message itself, to
;; give the user time to respond to the warning to stop typing.  IN this interval, anything that is typed is thrown away.  After printing the
;; message, we do a DOBE, and then check to see if user has typed anything.  If he has, this material is discarded, and bells printed again to
;; warn him.
     (DOBE)
     (COND
        ((READP T)
         (PRINTBELLS)
         (DOBE)
         (CLEARBUF T)))
 MESS                                                    ; MESS is either an atom or string or a list, in which case it is
                                                         ; MAPRINTed
     (COND
        ((NULL MESS)                                     ; Either user didnt supply a message or else was printed above.

         )
        ((NLISTP MESS)
         (ASKUSERPRIN1 MESS))
        (T (MAPRINT MESS T NIL " ? " NIL NIL LISPXPRNTFLG)))
     (COND
        ((OR (NOT (NUMBERP WAIT))
             (NULL DEFAULT))                             ; is : either a number, meaning wait that many seconds or NIL,
                                                         ; meaning wait forever
         (GO READLP)))
     [COND
        ((AND DEFAULT (NLISTP DEFAULT))
         (SETQ DEFAULT (LIST DEFAULT]
     (COND
        ((NULL (WAITFORINPUT (ITIMES WAIT 1000)))        ; Assume DEFAULT if nothing typed in WAIT/4 seconds.
         (PRIN1 "..." T)
         (SETQ CHAR (CAR DEFAULT))
         (GO INTERP)))
 READLP
     [COND
        ((AND (STRINGP FILE)
              (NOT (READP FILE T)))
         (SETQ FILE T)
         (SETQ OLDTTBL (GETTERMTABLE))
         (RESETSAVE (SETTERMTABLE ASKUSERTTBL]           ; the string ran out
     (SETQ CHAR (PEEKC FILE))                            ; PEEKC used so that in case of $ as a key, askuser can do a
                                                         ; READ.
     (SETQ ECHOEDFLG NIL)                                ; this character has not yet been echoed.  or read
     (SETQ DEFAULT NIL)
 INTERP

;; KEYLST is a list of elements of the form (KEY PROMPTSTRING . OPTIONS), where KEY is an atom or string (including the empty string)
;; that characters are to be matched against, PROMPTSTRING a string or atom (NIL is equivalent to ''), and OPTIONS a list in property list
;; format which can contain the properties (KEYLST CONFIRMFLG RETURN EXPLAINSTRING NOECHOFLG KEYSTRING PROMPTON
;; COMPLETEON AUTOCOMPLETEFLG) Default options for the entire keylst can be supplied as an argument to ASKUSER --- --- A key is
;; considered to be complete when (1) all of its characters have been matched and it is the only key left, i.e.  there are no other keys for
;; which this key is a substring, (2) all of its characters have been matched, and CONFIRMFLG is NIL, and the next character matches one
;; of the keys on its KEYLST, (3) all of its characters have been matched, and a confirming character is typed, i.e.  a c.r., space, or member
;; of CONFIRMFLG (This option is used for implementing TENEX protocosl, where CONFIRMFLG is ($)) or (4) there is only one key left and
;; a confirming character is typed.  --- --- When a key is complete, PROMPTSTRING is printed.  Then if CONFIRMFLG is non-NIL and the
;; key was not completed via a confirming character (case 3 and 4 above) askuser waits for a confirming character.  --- --- After confirmation,
;; if KEYLST is non NIL, askuser descends into KEYLST.  Otherwise askuser returns a value which is the value of (eval of) the RETURN
;; field, if non-NIL, otherwise the result of packing all the keys or keystrings, if present --- see below on the path.  --- At any point, the user
;; can type an alt-mode which is equivalent to typing the next n shared characters.  (if there are none, a bell is rung.) Typing a confirming
;; character has the same effect as typing an alt-mode, i.e.  the next n shared characters will be supplied.  If the key is the only key left,
;; confirmation is not required.  (this is case 4 above).  If the key is not the only key left, a bell is rung.  --- --- special options: ---
;; EXPLAINSTRING if non-nil, used in place of key/keystring + promptstring when user types a ?  --- NOECHOFLG if non-nil, characters that
;; are matched are not echoed --- KEYSTRING if non-nil, characters that are matched are echoed from keystring.  The main reason for this
;; feature echoing, since ASKUSER converts everything to a canonical upper case form, keys will always be represented in uppercase.
;; KEYSTRING can be used to provide for lower case echoing, and for returning a lower case value.  i.e.  if the RETURN option is not
;; specified, and KEYSTRING is specified, then KEYSTRING will be used in constructing the value to be returned, rather than KEY.  ---
;; PROMPTON if non-NIL, PROMPTSTRING is printed only when the key is confirmed with a member of PROMPTON.  This feature is used
;; for implementing TENEX protocols, in which case PROMPTON would be ($) Note that this doesnt make much sense unless
;; CONFIRMFLG is also non-NIL and includes the elements on PROMPTON --- --- COMPLETEON when a confirming character is typed, the
;; n characters that are supplied are not echoed unless the confirming charactter is a member of COMPLETEON.  This is used for
;; implementing tenex protocols in which case COMPLETEON is ($), i.e.  user could complete a command with space or c.r.  but completion
;; and prompting would take place only for $ --- --- AUTOCOMPLETEFLG if T, says supply characters as soon as they are unambiguous, i.e.
;; act as though alt-mode were typed after each character (but dont ring a bell) --- MACROCHARS, a list of characters and forms.  if one of
;; the characters is typed, and doesnt match as a key, then the form is evaluated for effect and everything else stays the same, e.g.  ?  could
;; have been implemented this way.  this feature is probably most useful when MACROCHARS is supplied on OPTIONSLST since one
;; probably wants a global set of MACROCHARS for a call single call to askuser.  --- --- & as a key matches any character.  --- --- '' can be
;; used as a key It starts out with all of its characters matched, so that it is complete if it is the only key left, (1) above, or the next character
;; mtches one of the keys on its KEYLST, etc.  --- --- $ can be used as a key to match the result of doing a READ.  For example, the filepkg
;; has as one of its entries on its keylst ('' 'file/list: ' KEYLST ($)) which means that if a character is typpd that does not match any of the other
;; charactters on its keylst, the prompt message file/list: is printed, and a read is then performed and eturned as the value of the call to
;; askuser.  --- --- --- For the more common useage, KEY is the same as (KEY NIL CONFIRMFLG T), and (KEY . PROMPT) the same as
;; (KEY PROMPT)
     [SETQ KEYLST1 (for ENTRY in KEYLST eachtime (ASKUSERSETUP ENTRY) collect ENTRY
                      when (COND
```

```
                ((ASKUSERCHAR CHAR (SETQ TEM (NTHCHAR KEY NC)))
                                                ; char matches the corresponding character in key.
                 T)
                ((OR TEM $$VAL (EQ CHAR '?))

              ;; There was another character in the key, and char didnt match it.  The $$VAL check is to insure that
              ;; once there has been a match with a character in a key atthis level, we do not treat space or c.r.  as
              ;; terminators, so that space and c.r.  can be used as keys themselves, nor do we descend into
              ;; subkeylists, and so thatthe user can specify a default match via '' as a place marker, and have it
              ;; operate ONLY when other elements are not matched by placing it last on the keylst.  e.g.  if keylst is
              ;; of the form ((c.r.  --) -- ('' -- subkeylst)) and a c.r.  is typed, matching wont go into subkeylst
              ;; ADDTOFILES uses this feature

                 NIL)
                ((AND (NULL (ASKUSERLOOKUP 'CONFIRMFLG))
                      (ASKUSERLOOKUP 'KEYLST)
                      (ASKUSER1 ENTRY CHAR))

              ;; We have already matched all the characters in key, and entry contains a lower keylst.  and char
              ;; matches one of its elements, therefore do any prompting necessary for this key, and descend

                 (SETQ ANSWER (NCONC1 ANSWER (OR (ASKUSERLOOKUP 'KEYSTRING)
                                                 KEY)))
                 [AND (NULL NOECHOFLG)
                      (SETQ PRINTLST (NCONC1 PRINTLST (OR (ASKUSERLOOKUP 'KEYSTRING)
                                                          KEY]
                 [AND PROMPTSTRING (SETQ PRINTLST (NCONC1 PRINTLST (PRIN1 PROMPTSTRING T]

                 ;; PRINTLST is maintained to implement the ?  feature and to be able to replay the output to put
                 ;; on the history.

                 (SETQ KEYLST (ASKUSERLOOKUP 'KEYLST))
                 (SETQ NC 1)                    ; CHAR will then be matched aainst the lower keylst.
                 (GO INTERP))
                ([COND
                    ((LISTP CONFIRMFLG)
                     (MEMB CHAR CONFIRMFLG))
                    (T (OR (EQ CHAR EOL)
                           (EQ CHAR SPACE]

              ;; all of its characters were matched, and this character was a c.r.  or space.  e.g.  CHARLST= (CLISP
              ;; CLISPFLG CLISPTRANFLG) and CLISP c.r.  has been typed The check is made after the other
              ;; checks so that space and carriage return themselves can be used in keys.  Note that it doesnt
              ;; matter whether confirmflg is T or not, the user can still use c.r.  or space to terminate a key.

                 (AND (NULL NOECHOFLG)
                      (SETQ PRINTLST (NCONC1 PRINTLST CHAR)))
                 T]
(ASKUSERSETUP (CAR KEYLST))
[COND
   (KEYLST1 (SETQ KEYLST KEYLST1)
            (GO RIGHT))
   ((AND (NULL ANSWER)
         (EQ NC 1)
         (NULL DEFAULT)
         (OR (EQ CHAR SPACE)
             (EQ CHAR EOL)))                    ; user typed eol  or space simply to keep dwim from defaulting on
                                                ; him.
    (AND (NULL NOECHOFLG)
         (PRIN1 CHAR T))
    (AND (READC FILE))
    (GO READLP))
   ([OR [EQ CHAR (CONSTANT (CHARACTER (CHARCODE ESCAPE]
        (COND
           ((LISTP CONFIRMFLG)
            (MEMB CHAR CONFIRMFLG))
           (T (OR (EQ CHAR EOL)
                  (EQ CHAR SPACE]

    ;; altmode c.r.  or space says supply characters from atoms in this level of keylst until there are two or more atms with different
    ;; characters at thatposition.  C.R.  and space is same as alt mode except if there is only one atom, then return without
    ;; confirmation after supplying the characters.  If thee are not atms with common characters beyond this point, then ring a bell
    ;; and take no action.

    [COND
       ((NULL (SETQ TEM (ASKUSER$ KEYLST CHAR NC)))
        (GO WRONG))
       (T (SETQ NC (ADD1 TEM]
    (AND (NULL DEFAULT)
         (READC FILE))
    (COND
       ((NULL (CDR KEYLST))                     ; only one.  Therefore this character completes the key,
        (GO COMPLETED))
       ((OR (EQ CHAR SPACE)
            (EQ CHAR EOL))
        (PRIN1 (CHARACTER (CHARCODE BELL))
               T)                               ; print a bell.

        ))
    (GO NEXT))
   ((OR (SYNTAXP (SETQ TEM (CHCON1 CHAR))
                 'CHARDELETE)
```

```
        (SYNTAXP TEM 'LINEDELETE))                    ; control-a, q,
    (GO RETRY))
   ([AND (NULL DEFAULT)
         (EQ FILE T)
         (SETQ TEM (FASSOC CHAR (ASKUSERLOOKUP 'MACROCHARS]
    (READC T)
    (SETTERMTABLE OLDTTBL)
    (EVAL (CDR TEM))
    (SETTERMTABLE ASKUSERTTBL)
    (GO READLP))
   ((AND (NULL DEFAULT)
         (EQ CHAR '?)
         (EQ FILE T))
    (TERPRI T)
    (READC T)
    [NLSETQ (PROGN (PRIN1 (OR (fetch (OPTIONS EXPLAINSTRING) of OPTIONSLST)
                              (CONCAT "one of:" EOL))
                          T)
                   (ASKUSEREXPLAIN KEYLST PRINTLST OPTIONSLST (OR (ASKUSERLOOKUP
                                                                      'EXPLAINDELIMITER)
                                                                  EOL]
    (TERPRI T)
    [AND ORIGMESS (COND
                      ((NLISTP ORIGMESS)
                       (ASKUSERPRIN1 ORIGMESS))
                      (T (MAPRINT ORIGMESS T NIL " ? " NIL NIL LISPXPRNTFLG]
    [MAPC PRINTLST (FUNCTION (LAMBDA (X)
                               (PRIN1 X T]
    (AND (NEQ NC 1)
         (PRIN1 (SUBSTRING [COND
                               ((NLISTP (CAR KEYLST))
                                (CAR KEYLST))
                               (T (OR (fetch (ASKUSER KEYSTRING) of (CAR KEYLST))
                                      (fetch (ASKUSER KEY) of (CAR KEYLST]
                           1
                           (SUB1 NC))
                T))
    ;; These are the characters that have been matched on this level key, but not yet added to answer or printlst.
    (GO READLP))
   ([SETQ KEYLST1 (find X in KEYLST
                     suchthat (SELECTC X
                                       ([LIST '& (CHARACTER (CHARCODE ESCAPE))
                                              (PACKC (CHARCODE (ESCAPE ESCAPE]
                                        (SETQ KEY X)
                                        T)
                                       (AND (LISTP X)
                                            (SELECTC (CAR X)
                                                 ('&
                                                      (COND
                                                         ((OR [NULL (SETQ TEM (LISTGET1 X 'CLASS]
                                                              (APPLY* TEM CHAR))
                                                          (SETQ KEY (CAR X))
                                                          T)))
                                                 ([LIST (CHARACTER (CHARCODE ESCAPE))
                                                        (PACKC (CHARCODE (ESCAPE ESCAPE]
                                                      (SETQ KEY (CAR X))
                                                      T)
                                                 (AND (LISTP (CAR X))
                                                      (SETQ KEY (CAR X]
    (COND
       ((EQ KEY '&)
        [SETQ KEYLST (LIST (CONS CHAR (AND (LISTP KEYLST1)
                                           (CDR KEYLST1]
        (GO RIGHT))
       (T                                             ; altmode.  or double-altmode
                                                      (* (AND (EQ FILE T) (PRIN1 CHAR T)))
        ;; The character would not have been echoed since the PEEKC was done with echomode off.  Since it has already been
        ;; seen by LISP, it wold not be echoed by the READ below, even though ECHOMODE would then be turned on.
        ;; Therefore must print it.
        (SETTERMTABLE OLDTTBL)
        (OR (PROG1 [NLSETQ (COND
                               ([EQ KEY (CONSTANT (CHARACTER (CHARCODE ESCAPE]
                                (SETQ TEM (READ FILE T)))
                               [[EQ KEY (CONSTANT (PACKC (CHARCODE (ESCAPE ESCAPE]
                                (LET (READBUF)
                                     (DECLARE (SPECVARS READBUF))
                                     ;; since READ is used, rather than lispxread for $ key, we should not have readline be affected by
                                     ;; readbuf, e.g.  if user is redoing an event contaig an askuser, he wants to type in tuff again.
                                     (SETQ TEM (READLINE T]
                                     (T (SETQ TEM (EVAL KEY]
                   (SETTERMTABLE ASKUSERTTBL))
            (GO RETRY))
        (SETQ KEYLST (LIST (create ASKUSER using (LISTP KEYLST1)
                                                 KEY _ TEM)))
```

```
                    (SETQ NC (ADD1 (NCHARS TEM)))
                    (SETQ ECHOEDFLG T)                          ; so that the character terminatng the read wont be echoed twice
                    [COND
                       [(SYNTAXP [SETQ TEM (CHCON1 (SETQ CHAR (LASTC FILE]
                                 'SEPR T)                       ; character was included as part of the read
                        (replace OPTIONS of (CAR KEYLST) with (CONS 'CONFIRMFLG (CONS (LIST CHAR)
                                                                                      (fetch OPTIONS
                                                                                         of (CAR KEYLST]
                       ((SYNTAXP TEM 'BREAK T)                  ; e.g.  read of a lit
                        (GO READLP))
                       (T (SETQ CHAR (READC FILE]
                    ;; (COND ((EQ KEY (CONSTANT (CHARACTER (CHARCODE ESCAPE)))) (* (61  . 965) 130 <NEWLISP>ASSIST.;8
                    ;; NIL) (SETQ CHAR (READC FILE))) ((EQ KEY (CONSTANT (PACKC (CHARCODE (ESCAPE ESCAPE))))) (SETQ
                    ;; CHAR (LASTC FILE)) (replace OPTIONS of (CAR KEYLST) with (CONS (QUOTE CONFIRMFLG) (CONS (QUOTE (]
                    ;; )) (fetch (ASKUSER OPTIONS) of (CAR KEYLST)))))) ((LISTP KEY) (* (73  . 955) 107 <NEWLISP>ASSIST.;30 NIL))
                    ;; (T (SHOULDNT)))
                    (SETQ DEFAULT '(T))
                    ;; so wont attempt to read the character again.  reason we have to read it here, in the case of read, is that it has already
                    ;; been echoed, and in the case of a lower keylst, there would be no way to psass on the information about it having
                    ;; been echoed without setting echoedflg to T.  thus we cant go back to READLP, sice that wold set echoflg to NIL.
                    (GO INTERP]
WRONG
                                                                ; user typed invalid answer
    (AND (NEQ FILE T)
         (ERROR!))
    (AND (NULL DEFAULT)
         (READC FILE))
    (COND
       (TYPEAHEAD (GO RETRY1)))
    (PRINTBELLS)
    (DOBE)
    (CLEARBUF T)
    (GO READLP)
RIGHT
                                                                ; character matched.
    (AND (NULL DEFAULT)
         (READC FILE))
RIGHT1
    (ASKUSERSETUP (CAR KEYLST))
    (COND
       ((OR (CDR KEYLST)
            (ILESSP NC (NCHARS KEY)))                           ; More than one candidate.  or this candidate not finished yet.
        (AND (NULL NOECHOFLG)
             (EQ FILE T)
             (SETQ TEM (COND
                          ((SETQ TEM (ASKUSERLOOKUP 'KEYSTRING))
                           ;; primarily to allow specifying of echoing in lower case, even though askuser always converts to
                           ;; uppercase when it reads.
                           (NTHCHAR TEM NC))
                          (T CHAR)))
             (PRIN1 TEM T))
        (SETQ NC (ADD1 NC))
        [COND
           ((AND (ASKUSERLOOKUP 'AUTOCOMPLETEFLG)
                 (SETQ TEM (ASKUSER$ KEYLST CHAR NC)))
            (COND
               ((AND (NULL (CDR KEYLST))
                     (EQ (SETQ NC TEM)
                         (NCHARS KEY)))
                (GO COMPLETED))
               (T (SETQ NC (ADD1 TEM]
        (GO NEXT)))                                             ; There is only one entry left, and all of its characters are
                                                                ; matched.
    (AND (NULL NOECHOFLG)
         (EQ FILE T)
         (EQ NC (NCHARS KEY))
         (SETQ TEM (COND
                      ((SETQ TEM (ASKUSERLOOKUP 'KEYSTRING))
                       (NTHCHAR TEM NC))
                      (T CHAR)))
         (PRIN1 TEM T))
;; the character is the last one in the key.  the case where a c.r.  was typed to terminate a key is handled below.
COMPLETED
    (SETQ ANSWER (NCONC1 ANSWER (OR (ASKUSERLOOKUP 'KEYSTRING)
                                    KEY)))
    [AND (NULL NOECHOFLG)
         (SETQ PRINTLST (NCONC1 PRINTLST (OR (ASKUSERLOOKUP 'KEYSTRING)
                                             KEY]
    [AND PROMPTSTRING (OR [NULL (SETQ TEM (ASKUSERLOOKUP 'PROMPTON]
                          (MEMB CHAR TEM))
         (SETQ PRINTLST (NCONC1 PRINTLST (PRIN1 PROMPTSTRING T]
;; If PROMPTON is present, must wait till after confirmation to see if confirming charactter is PROMPTON (usually $).  this enables tenex like
;; protocols.
```

```
        (AND (NULL NOECHOFLG)
             (EQ FILE T)
             (IGREATERP NC (NCHARS KEY))
             (PRIN1 (COND
                        ([AND (EQ CHAR EOL)
                              (NULL (ASKUSERLOOKUP 'KEYLST]
                         ;; space is echoed for all confirming characters except on a terminal leaf,in which char isused itself.
                         CHAR)
                        (T SPACE))
                    T))
        (COND
           ([OR (NULL CONFIRMFLG)
                (COND
                   ((LISTP CONFIRMFLG)
                    (MEMB CHAR CONFIRMFLG))
                   (T (OR (EQ CHAR EOL)
                          (EQ CHAR SPACE]
            ;; CONFIRMFLG can be a list of characters that are acceptable for confirming.  e.g.  ($) can be used to implemente tenex like
            ;; protocols.
            (GO CONFIRMED))
           (T (GO CONFIRM)))
    NEXT
        (SETQ DEFAULT (CDR DEFAULT))
    ;; DEFAULT stays one behind the current character so that we can tell if the character came from a default list.
        (COND
           ((NULL DEFAULT)
            (GO READLP))
           (T (SETQ CHAR (CAR DEFAULT))
              (GO INTERP)))
        (GO INTERP)
    CONFIRM
        (COND
           ((ASKUSERLOOKUP 'PROMPTCONFIRMFLG)
            (PRIN1 " [confirm] " T)))
        [COND
           ((AND (STRINGP FILE)
                 (NOT (READP FILE T)))
            (SETQ FILE T)
            (SETQ OLDTTBL (GETTERMTABLE))
            (RESETSAVE (SETTERMTABLE ASKUSERTTBL]
        [SETQ CHAR (COND
                      ((SETQ DEFAULT (CDR DEFAULT))
                       (CAR DEFAULT))
                      (T (READC FILE]
        (COND
           ((OR (SYNTAXP (SETQ TEM (CHCON1 CHAR))
                         'CHARDELETE)
                (SYNTAXP TEM 'LINEDELETE))                    ; control-a or q
            (GO RETRY))
           [(LISTP CONFIRMFLG)
            (COND
               ((MEMB CHAR CONFIRMFLG)                       ; used for TENEX mode.
                [AND PROMPTSTRING (SETQ TEM (ASKUSERLOOKUP 'PROMPTON))
                     (MEMB CHAR TEM)
                     (SETQ PRINTLST (NCONC1 PRINTLST (PRIN1 PROMPTSTRING T]
                (AND (NULL NOECHOFLG)
                     (PRIN1 SPACE T))
                (GO CONFIRMED]
               ((OR (EQ CHAR SPACE)
                    (EQ CHAR EOL))
                [COND
                   ((NULL NOECHOFLG)
                    (SETQ PRINTLST (NCONC1 PRINTLST (PRIN1 (COND
                                                              ((NULL (ASKUSERLOOKUP 'KEYLST))
                                                               CHAR)
                                                              (T SPACE))
                                                           T]
                (GO CONFIRMED))
               ([SETQ TEM (FASSOC CHAR (ASKUSERLOOKUP 'MACROCHARS]
                (SETTERMTABLE OLDTTBL)
                (EVAL (CDR TEM))
                (SETTERMTABLE ASKUSERTTBL)
                (GO CONFIRM)))
        (COND
           ((NEQ CHAR '?)
            (PRIN1 (PACKC (CHARCODE (BELL ?)))
                   T)
            (DOBE)
            (CLEARBUF T)))
        (PRIN1 " [confirm] " T)
        (GO CONFIRM)
    CONFIRMED
        (COND
```

```
                ((SETQ TEM (ASKUSERLOOKUP 'KEYLST))
                 (SETQ KEYLST TEM)
                 (SETQ NC 1)
                 (GO NEXT)))
             (COND
                (LISPXPRNTFLG [MAPC PRINTLST (FUNCTION (LAMBDA (X)
                                                       (ASKUSERPRIN1 X T]
                                                              ; fakes the printing for the history list.

                      ))
             (COND
                (BUFS (BKBUFS BUFS)))
             (RETURN (COND
                       [(SETQ TEM (OR (FMEMB 'RETURN OPTIONS)
                                      (FMEMB 'RETURN OPTIONSLST)))
                        (SETTERMTABLE OLDTTBL)
                        (COND
                           ([SETQ TEM (NLSETQ (EVAL (CADR TEM]
                            ;; ASKUSERLOOKUP (QUOTE not) used since then couldnt distinguish case where RETURN NIL was
                            ;; specified from case where RETURN was not specified at all.
                            ;; This permits user to return ANSWER as a list itself, or to take some other action, and then restart by simply
                            ;; generateing an error.
                            (CAR TEM))
                           (T (SETTERMTABLE ASKUSERTTBL)
                              (GO RETRY]
                       (ANSWER (PACK ANSWER))
                       (T (NOTCHECKED)
                          KEY)))
      RETRY
             (COND
                (TYPEAHEAD (GO RETRY1)))
             (PRIN1 "___" T)
             (TERPRI T)
             (DOBE)
             (CLEARBUF T)
             (SETQ KEYLST ORIGKEYLST)
             (SETQ PRINTLST NIL)
             (SETQ NC 1)
             (SETQ ANSWER NIL)
             (GO READLP)
      RETRY1
```

;; User has typed ahead before the call to askuser1 and his resonse is invalid. therefore assume he didnt know that askuser would be called
;; and his typeahead was intended for what follows. clear and ave the typeahead and continue with interaction.

```
             (LINBUF)
             (SYSBUF)
             (SETQ BUFS (CLBUFS NIL T READBUF))
             [SETQ TEM (APPLY 'CONCAT (NCONC ANSWER [AND (NEQ NC 1)
                                                         (LIST (SUBSTRING (COND
                                                                            ((LISTP (CAR KEYLST))
                                                                             (CAAR KEYLST))
                                                                            (T (CAR KEYLST)))
                                                                          1
                                                                          (SUB1 NC]
                                                (LIST CHAR]
             [COND
                ((NULL BUFS)
                 (SETQ BUFS (CONS NIL TEM)))
                (T (RPLACD BUFS (COND
                                  ((CDR BUFS)
                                   (CONCAT TEM (CDR BUFS)))
                                  (T TEM]
             (SETQ TYPEAHEAD NIL)                              ; so this is only done once
             (SETQ ANSWER NIL)
             (SETQ KEYLST ORIGKEYLST)
             (SETQ MESS ORIGMESS)
             (SETQ DEFAULT ORIGDEFAULT)
             (SETQ PRINTLST NIL)
             (TERPRI T)
             (GO MESS)))])
```

(ASKUSERLOOKUP

```
  [LAMBDA (FIELD)                                              (* bvm%: "26-Apr-86 17:14")
```

(* * this wuld be just a fetch, xcept want to lok it up on optionslst if not found on options.)

```
    (CADR (OR (FMEMB FIELD OPTIONS)
              (FMEMB FIELD OPTIONSLST])
```

(ASKUSERCHAR

```
  [LAMBDA (C1 C2)                                              (* bvm%: "26-Apr-86 17:27")
    (COND
       ((EQ C1 C2))
```

```
((AND (NULL NOCASEFLG)
      C2)
 (SETQ C1 (CHCON1 C1))
 (SETQ C2 (CHCON1 C2))
 (COND
    [(AND (IGEQ C1 (CHARCODE a))
          (ILEQ C1 (CHARCODE z)))
     (EQ C2 (IDIFFERENCE C1 (IDIFFERENCE (CHARCODE a)
                                         (CHARCODE A]
    ((AND (IGEQ C2 (CHARCODE a))
          (ILEQ C2 (CHARCODE z)))
     (EQ C1 (IDIFFERENCE C2 (IDIFFERENCE (CHARCODE a)
                                         (CHARCODE A])
```

(**ASKUSER$**

```
[LAMBDA (KEYLST CHAR NC)                                    (* bvm%: "26-Apr-86 17:13")
  (for ENTRY bind NC0 KEY0 TEM in KEYLST eachtime [SETQ KEY (COND
                                                             ((NLISTP ENTRY)
                                                              ENTRY)
                                                             (T (fetch (ASKUSER KEY) of ENTRY]
     when [AND [NEQ KEY (CONSTANT (CHARACTER (CHARCODE ESCAPE]
               (NEQ KEY (CONSTANT (PACKC (CHARCODE (ESCAPE ESCAPE]
     do [COND
          ((NULL KEY0)                                      (* first time through)
           [SETQ KEY0 (COND
                        ((NLISTP (CAR KEYLST))
                         (CAR KEYLST))
                        (T (fetch (ASKUSER KEY) of (CAR KEYLST]
           (SETQ NC0 (NCHARS KEY0)))
          (T
```

(* Goes through keylst and looks at each key and determines the largest N for which NTHCHAR of thatcharacter is equal for every atom.)

```
               (SETQ NC0 (for I from 1 to NC0 while (EQ (NTHCHARCODE KEY I)
                                                        (NTHCHARCODE KEY0 I))
                            finally (RETURN (SUB1 I]
     finally (COND
                ((OR (NULL NC0)
                     (ILESSP NC0 NC))                       (* all atoms have different characters at this position.)
                 (RETURN NIL)))
             (ASKUSERSETUP (CAR KEYLST))
             [SETQ TEM (AND (OR [NULL (SETQ TEM (ASKUSERLOOKUP 'COMPLETEON]
                                (MEMB CHAR TEM))
                            (SUBSTRING (OR (ASKUSERLOOKUP 'KEYSTRING)
                                           KEY)
                                   NC
                                   (COND
                                      ((EQ (NCHARS KEY0)
                                           NC0)             (* reason for this is in case KEYSTRING is longer, will get all of
                                       it.)
                                       -1)
                                      (T NC0]
```

(* if COMPLETEON is $ means only complete on alt-mode. this is used for tenex type protocol)

```
             (AND (NULL NOECHOFLG)
                  TEM
                  (PRIN1 TEM T))
```

(* Reason for not just using value of noechoflg is that askusersetup oul have set noechoflg to T when reading from a string in order to suppress echoing of the character, but this does not mean that we do not echo the characters that are supplied for copleting.)

```
             (RETURN NC0])
```

(**ASKUSER1**

```
[LAMBDA (ENTRY CHAR)                                        (* DD%: "26-Oct-81 12:34")
```

(* We know that ENTRY contains a subkeylst. This function sees if char could conceivably match one of the entries on keylst.)

```
  (thereis ENTRY bind TEM in (fetch (ASKUSER KEYLST) of ENTRY) eachtime [SETQ TEM (COND
                                                                                   ((NLISTP ENTRY)
                                                                                    ENTRY)
                                                                                   (T (fetch (ASKUSER KEY)
                                                                                         of ENTRY]
     suchthat (OR (EQ TEM '&)
                  [EQ TEM (CONSTANT (CHARACTER (CHARCODE ESCAPE]
                  [EQ TEM (CONSTANT (PACKC (CHARCODE (ESCAPE ESCAPE]
                  (LISTP TEM)
                  (EQ (SETQ TEM (NTHCHAR TEM 1))
                      CHAR)
                  (AND (NULL TEM)
                       (LISTP ENTRY)
```

```
                    (LISTP (CDR ENTRY))
                    (ASKUSER1 ENTRY CHAR])
```

(ASKUSERSETUP

```
  [LAMBDA (ENTRY)                                          (* bvm%: "26-Apr-86 17:13")
                                                           (* Sets free variables KEY, CONFIRMFLG, QUIETFLG, and
                                                           PROMPTSTRING)
    (PROG (TEM)
          [COND
             [(NLISTP ENTRY)
              (SETQ KEY ENTRY)
              (SETQ PROMPTSTRING NIL)
              (SETQ OPTIONS NIL)                           (* The default is for NOECHOFLG to be NIL and CONFIRMFLG
              to be T.)
              (SETQ CONFIRMFLG (COND
                                   ((SETQ TEM (MEMB 'CONFIRMFLG OPTIONSLST))
                                    (CADR TEM))
                                   (T T]
             [(NLISTP (CDR ENTRY))
              (SETQ KEY (CAR ENTRY))
              (SETQ PROMPTSTRING (CDR ENTRY))
              (SETQ OPTIONS NIL)
              (SETQ CONFIRMFLG (COND
                                   ((SETQ TEM (MEMB 'CONFIRMFLG OPTIONSLST))
                                    (CADR TEM))
                                   (T T]
             (T (SETQ KEY (fetch (ASKUSER KEY) of ENTRY))
                (SETQ PROMPTSTRING (fetch (ASKUSER PROMPTSTRING) of ENTRY))
                (SETQ OPTIONS (fetch (ASKUSER OPTIONS) of ENTRY))
                (SETQ CONFIRMFLG (ASKUSERLOOKUP 'CONFIRMFLG]
          (SETQ NOECHOFLG (ASKUSERLOOKUP 'NOECHOFLG))
          (SETQ NOCASEFLG (ASKUSERLOOKUP 'NOCASEFLG))
          (AND ECHOEDFLG (SETQ NOECHOFLG T))
          (COND
             ((AND (NEQ FILE T)
                   (STRINGP FILE)
                   (READP FILE T))
              (SETQ NOECHOFLG T)
              (SETQ PROMPTSTRING NIL)
```

(* askusersetup is called after the character has been read. Thus, this sets noechoflg to T and promptstring to NIL only if there are more characters to be read. However, the check on whether or not the character JUST read is to bechoed alsoincludes an (EQ FILE T) check)

```
          ])
```

(ASKUSEREXPLAIN

```
  [LAMBDA (KEYLST PREV OPTIONSLST DELIMITER)               (* bvm%: "26-Apr-86 17:13")
    (MAPC KEYLST (FUNCTION (LAMBDA (ENTRY)
                           (PROG (KEY CONFIRMFLG NOECHOFLG PROMPTSTRING TEM OPTIONS (FILE T))
                                 (ASKUSERSETUP ENTRY)
                                 (COND
                                    ((SETQ TEM (ASKUSERLOOKUP 'KEYLST))
                                                           (* entry is of the form (key prompt charlst))
                                     (ASKUSEREXPLAIN TEM
                                            [COND
                                               ((SETQ TEM (fetch (OPTIONS EXPLAINSTRING) of OPTIONS))
```

(* reason for not using askuserlookup is that don't want top level explainstring on ptionslst, if any. doesnt make sense to print it each time. it is printed only once.)

```
                                                (APPEND PREV (LIST TEM)))
                                               (T (APPEND PREV (AND (NULL NOECHOFLG)
                                                                    (LIST (OR (ASKUSERLOOKUP 'KEYSTRING)
                                                                              KEY)))
                                                          (AND PROMPTSTRING (LIST PROMPTSTRING]
                                            OPTIONSLST DELIMITER)
                                     (RETURN)))
                                 [MAPC PREV (FUNCTION (LAMBDA (X)
                                                      (COND
                                                         ((LISTP X)
                                                          (MAPRINT X T))
                                                         (T (PRIN1 X T]
                                 [COND
                                    [(SETQ TEM (fetch (OPTIONS EXPLAINSTRING) of OPTIONS))
                                     (COND
                                        ((LISTP TEM)
                                         (MAPRINT TEM T))
                                        (T (PRIN1 TEM T]
                                    ((SETQ TEM (OR (ASKUSERLOOKUP 'KEYSTRING)
                                                   KEY))
                                     (AND (NULL NOECHOFLG)
                                          [NEQ TEM (CONSTANT (CHARACTER (CHARCODE ESCAPE]
                                          (NEQ TEM '&)
                                          (PRIN1 TEM T))
```

```
        (* If the user wants to explain the & or $, he can include the appropriate text in the prompt field.)

                                        (AND PROMPTSTRING (PRIN1 PROMPTSTRING T]
                                    (AND (NEQ (POSITION T)
                                              0)
                                         (PRIN1 DELIMITER T))
                                    (RETURN])
```

(ASKUSERPRIN1

```
  [LAMBDA (X NODOFLG)                                                    (* wt%: % 4-DEC-75 00%:39)

        (* does a lispxprin1 if lispxprntflg is non-NIL. used to be done by having everythin printed with lispxprin1 and doing a
        resetsave on lisxpprintflg, but this costs several conses each call.)

    (COND
       ((NULL LISPXPRNTFLG)
        (OR NODOFLG (PRIN1 X T)))
       (T (LISPXPRIN1 X T NIL NODOFLG)))
    X])
```

(MAKEKEYLST

```
  [LAMBDA (LST DEFAULTKEY LCASFLG AUTOCOMPLETEFLG)                       (* wt%: "14-NOV-78 02:03")
    (PROG (TEM)
          (RETURN (NCONC [SETQ TEM (MAPCAR LST (FUNCTION (LAMBDA (KEY)
                                                   (LIST KEY NIL 'KEYSTRING
                                                         (CONCAT (COND
                                                                    ((AND LCASFLG (EQUAL KEY
                                                                                         (U-CASE KEY)))
                                                         (* when ucasep gets in system, use it instead)
                                                                     (L-CASE KEY))
                                                                    (T KEY))
                                                                 " ")
                                                         'CONFIRMFLG T 'AUTOCOMPLETEFLG AUTOCOMPLETEFLG
                                                         'RETURN
                                                         (KWOTE KEY]
                                [for X in TEM bind KEYSTRING as I from 1 collect (SETQ KEYSTRING (LISTGET X 'KEYSTRING))
                                                                                 (LIST I KEYSTRING 'NOECHOFLG T
                                                                                       'EXPLAINSTRING
                                                                                       (CONCAT I " - " KEYSTRING)
                                                                                       'CONFIRMFLG T 'RETURN
                                                                                       (LIST 'PROGN '(TERPRI T)
                                                                                             (KWOTE (CAR X]
                                (COND
                                   [(NULL DEFAULTKEY)
                                    (LIST '("No - none of the above " "" CONFIRMFLG T AUTOCOMPLETEFLG T RETURN NIL]
                                   ((LISTP DEFAULTKEY)                   (* so user can specify no default key by simply calling with
                                                                         defaultkey=T)
                                    (LIST DEFAULTKEY])

)

;; RMK: Avoid literal CR's on files.

(RPAQ? DEFAULTKEYLST [LIST [LIST 'Y (CONCAT "es" (CHARACTER (CHARCODE EOL]
                           (LIST 'N (CONCAT "o" (CHARACTER (CHARCODE EOL])

(RPAQ? ASKUSERTTBL (COPYTERMTABLE))

(DECLARE%: DONTEVAL@LOAD DOCOPY

(CONTROL T ASKUSERTTBL)

(ECHOMODE NIL ASKUSERTTBL)
)

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(DECLARE%: EVAL@COMPILE

(RECORD ASKUSER (KEY PROMPTSTRING . OPTIONS)
       (SYSTEM))

(PROPRECORD OPTIONS (KEYLST CONFIRMFLG RETURN EXPLAINSTRING NOECHOFLG KEYSTRING PROMPTON COMPLETEON
                            AUTOCOMPLETEFLG MACROCHARS NOCASEFLG PROMPTCONFIRMFLG CLASS)
       (SYSTEM))
)

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS DEFAULTKEYLST ASKUSERTTBL)
)
)

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

```
(ADDTOVAR NLAMA )

(ADDTOVAR NLAML )

(ADDTOVAR LAMA )
)

(PUTPROPS ASKUSER COPYRIGHT ("Venue & Xerox Corporation" 1986 1987 1990 2020 2021))
```

FUNCTION INDEX

RECORD INDEX

VARIABLE INDEX