```
(RPAQQ DICTTOOLCOMS
       ((COMS * DICTTOOLDEPENDENCIES)
        (FILES ANALYZER)
```
;; RMK 2022: DICTCLIENT has disappeared
```
        (* (FILES (FROM {NFS}<PROJECT>DICTSERVER>LISP>)
                DICTCLIENT))
```
;; 1/6/89 jtm: fixed TEdit.PrintDefinition so that SimpleDicts would print their entries in the definition window.

;; 1/6/89 jtm: changed TEdit.SetDictionary and DictForStream so that TEdit.DefaultDictionary reflects the default dictionary to use if no other is
;; specified.

;; 2/28/89 jtm: changed FILES so that DICTCLIENT is loaded from PIGLET.

;; 5/31/89 jtm: changed Dict.AddCommands to put Dictionary menu item on Lafite display windows.

;; 3/27/91 jtm: added TEdit interface to the SearchMenu module

;; 3/1/94 jtm: changed the loading of DICTCLIENT and SEARCHMENU
```
        (FNS TEDIT.INCLUDESTREAM TEdit.PrintDefinition DictTool.PrintDefinition Dict.PrintDefinition
            DictTool.GetEntry TEdit.SetDictionary DictForStream DictTool.Dictionaries PARSEBYCOLONS
            PrintPronunciationGuide ConvertPronunciation)
        (FNS TEdit.SearchMenu TEdit.PrintSearch DictTool.PrintSearch DictTool.MergeSearch NerdForStream
            TEdit.SetNerd DictTool.PromptForCutoff DictTool.PromptForKeywordCutoff PARSESELECTION)
        (FNS TEdit.PrintPhraseSearch DictTool.PrintPhraseSearch)
        (FNS TEdit.PrintSynonyms REMOVEALL CONVERTFUNCTIONSTOFORMS TEdit.PrintNounSynonyms
            DictTool.PrintNounSynonyms DictTool.PrintVerbSynonyms DictTool.PrintAdjSynonyms
            TEdit.PrintVerbSynonyms TEdit.PrintAdjSynonyms DictTool.PrintSynonyms)
        (FNS DictTool.TEditWrapper Dict.OutputStream DictTool.PromptStream)
        (FNS DictTool.Init DictTool.Open DictTool.OpenDictionary DictTool.OpenAnalyzer DictTool.OpenNerd
            Dict.AddCommands DictTool.Close)
        (FNS DictTool.Analyze DictTool.Analyzers DictTool.Pronunciation DictTool.Corrections DictTool.CountWords
            )
        (COMS (* * FINDWORD & SUBSTITUTEWORD)
            (FNS DictTool.FindWord DictTool.SubstituteWord DictTool.CreateConjugationMap DictTool.FindWordInit
                )
            (FNS LingFns.FindWord LingFns.Capitalize LingFns.Capitalization)
            (P (DictTool.FindWordInit)))
        (INITVARS DictTool.TimeOperation Dict.DefWindow Dict.CommandsAdded InvertedDict.List DictTool.LastSearch
            DictTool.LastWord TEdit.DefaultDictionary (DictTool.MinKeywords 2)
            (DictTool.MaxWords 100))
        (GLOBALVARS DictTool.TimeOperation Dict.DefWindow Dict.CommandsAdded InvertedDict.List
            DictTool.MinKeywords DictTool.MaxWords DictTool.LastSearch DictTool.LastWord
            TEdit.DefaultDictionary)
        (P (DictTool.Init))
        (VARS PronunciationGuide PronunciationMap)))

(RPAQQ DICTTOOLDEPENDENCIES
       [(* * code to make sure that the right versions of everything are loaded. The P must be executed before
           any FILES commands.)
        [E (PUTPROP 'DICTTOOL 'DEPENDENCIES (for FILE in (FILECOMSLST 'DICTTOOL 'FILES)
                                               collect
                                                  (CONS FILE (CAAR (GETPROP FILE 'FILEDATES]
        (PROP DEPENDENCIES DICTTOOL)
        (P (for FILE FILEDATE in (GETPROP 'DICTTOOL 'DEPENDENCIES)
               do
               [SETQ FILEDATE (CAAR (GETPROP (CAR FILE)
                                          'FILEDATES]
               (COND ([AND FILEDATE (CDR FILE)
                           (ILESSP (IDATE FILEDATE)
                                   (IDATE (CDR FILE]
                      (* clear FILEDATES to force FILESLOAD to reload the file.)
                      (printout T "Flushing old version of " (CAR FILE)
                             T)
                      (PUTPROP (CAR FILE)
                             'FILEDATES NIL])


           (* * code to make sure that the right versions of everything are loaded.
           The P must be executed before any FILES commands.)

(PUTPROPS DICTTOOL DEPENDENCIES ((ANALYZER . " 9-Mar-89 15:24:58")))
```

```
[for FILE FILEDATE in (GETPROP 'DICTTOOL 'DEPENDENCIES) do [SETQ FILEDATE (CAAR (GETPROP (CAR FILE)
                                                                                    'FILEDATES]
                                                        (COND
                                                          ([AND FILEDATE (CDR FILE)
                                                                (ILESSP (IDATE FILEDATE)
                                                                        (IDATE (CDR FILE]
                                                            (* clear FILEDATES to force FILESLOAD to reload the file.)
                                                            (printout T "Flushing old version of "
                                                                   (CAR FILE)
                                                                   T)
                                                            (PUTPROP (CAR FILE)
                                                                   'FILEDATES NIL]


(FILESLOAD ANALYZER)
```

;; RMK 2022: DICTCLIENT has disappeared

```
                (* * (FILES (FROM {NFS}<PROJECT>DICTSERVER>LISP>) DICTCLIENT))
```
;; 1/6/89 jtm: fixed TEdit.PrintDefinition so that SimpleDicts would print their entries in the definition window.

;; 1/6/89 jtm: changed TEdit.SetDictionary and DictForStream so that TEdit.DefaultDictionary reflects the default dictionary to use if no other is specified.

;; 2/28/89 jtm: changed FILES so that DICTCLIENT is loaded from PIGLET.

;; 5/31/89 jtm: changed Dict.AddCommands to put Dictionary menu item on Lafite display windows.

;; 3/27/91 jtm: added TEdit interface to the SearchMenu module

;; 3/1/94 jtm: changed the loading of DICTCLIENT and SEARCHMENU

```
(DEFINEQ
```

(TEDIT.INCLUDESTREAM
```
  [LAMBDA (TEXTSTREAM INCLUDEDSTREAM)                          (* jtm%: "28-Oct-87 14:41")
    (LET (STARTPOS)
         (SETQ STARTPOS (ADD1 (GETEOFPTR TEXTSTREAM)))
         (TEDIT.COPY (TEDIT.SETSEL INCLUDEDSTREAM 1 (GETEOFPTR INCLUDEDSTREAM)
                            'LEFT)
               (TEDIT.SETSEL TEXTSTREAM STARTPOS 0 'LEFT))
         (TEDIT.SETSEL TEXTSTREAM STARTPOS 0 'LEFT)
         (TEDIT.NORMALIZECARET TEXTSTREAM)
         (TEDIT.STREAMCHANGEDP TEXTSTREAM T])
```

(TEdit.PrintDefinition
```
  [LAMBDA (stream dict words)                                 ; Edited  6-Jan-89 11:46 by jtm:

          (* * prints out the definition of the currently selected text.)

    (OR stream (SETQ stream (Dict.OutputStream)))
    (DictTool.TEditWrapper (OR dict (DictForStream stream))
          [FUNCTION (LAMBDA (dict selection stream)
                    (LET (printFn entry)
                         (for word exists in (PARSEBYCOLONS selection)
                            do [COND
                                  ((AND (SETQ printFn (fetch (Dict printEntryFn) of dict))
                                        (NEQ printFn 'NILL))
                                    (SETQ exists (OR (APPLY* printFn dict word stream)
                                                     exists)))
                                  ((SETQ printFn (fetch (Dict getEntryFn) of dict))
                                   (SETQ entry (APPLY* printFn dict word NIL))
                                   [for def (left _ (LENGTH entry)) inside entry
                                      first (TEDIT.INSERT stream (CONCAT word ": "))
                                      do [COND
                                            ((STREAMP def)
                                             (SETQ def (STREAM.FETCHSTRING def 0 (GETEOFPTR def]
                                         (TEDIT.INSERT stream def)
                                         (add left -1)
                                         (COND
                                            ((IGEQ left 1)
                                             (TEDIT.INSERT stream ", "))
                                            (T (TEDIT.INSERT stream "

                                                           "]
                                   (SETQ exists (OR entry exists]
                            finally (RETURN exists]
          stream words "word to look up:" "Getting definition for"])
```

(DictTool.PrintDefinition
```
  [LAMBDA (dict words stream)                                 (* jtm%: "17-Nov-87 11:02")
    (PROG (def looks found pos (offset 0))
          [for word inside (PARSEBYCOLONS words)
             do (SETQ def (Dict.GetEntry dict word))
                (SETQ looks (Dict.Prop dict 'Looks))
                (COND
                   ((AND (NULL looks)
                         (Dict.Prop dict 'RemoteDict))
                     [SETQ looks (DICTCLIENT.GETLOOKS (Dict.Prop dict 'RemoteDict]
```

```
                                  (Dict.Prop dict 'Looks looks)))
                        (COND
                           [(STRINGP def)
                            (SETQ found T)
                            (TEDIT.INSERT stream def)
                            (for I from 1 to 2 when (NEQ 13 (NTHCHARCODE def (IMINUS I)))
                               do (TEDIT.INSERT stream (CHARACTER 13]
                           ([AND (STRINGP (CAR def))
                                 (NOT (STREQUAL "" (CAR def]
                            (SETQ found T)
                            (TEDIT.INSERT stream (CAR def)
                                   NIL
                                   (CDAR looks))
                            (for I from 1 to 2 when (NEQ 13 (NTHCHARCODE def (IMINUS I)))
                               do (TEDIT.INSERT stream (CHARACTER 13)))   (* assumes that the first look given is the default for the
                                                                             dictionary.)
                            (SETQ pos (TEDIT.GETPOINT stream))            (* setting looks moves the selection)
                            (for i in (CDR def) do (TEDIT.LOOKS stream (CDR (FASSOC (CADDR i)
                                                                                 looks))
                                                       (IPLUS (CAR i)
                                                              offset)
                                                       (CADR i)))
                            (SETQ offset (SUB1 pos))
                            (TEDIT.SETSEL stream pos 0 'LEFT))
                           (NIL (TEDIT.INSERT stream (CONCAT word ": not found."))
                                (TEDIT.LOOKS stream '(FAMILY HELVETICA SIZE 10 FACE BOLD)
                                       1
                                       (ADD1 (NCHARS word)))
                                (TEDIT.LOOKS stream '(FAMILY TIMESROMAN SIZE 10 FACE STANDARD)
                                       (IPLUS 2 (NCHARS word))
                                       11]
                    (RETURN found])
```

(**Dict.PrintDefinition**
```
    [LAMBDA (dict word stream)                                          (* jtm%: "13-Oct-87 10:27")
       (PROG (scratch start)
             [COND
                ((NULL stream)
                 (SETQ stream (Dict.OutputStream]
             [SETQ scratch (OPENTEXTSTREAM NIL NIL NIL NIL '(LEAVETTY]
             (COND
                ((Dict.PrintEntry dict word scratch)
                 (TEDIT.INSERT scratch (CONCAT (CHARACTER 13)
                                              (CHARACTER 13))
                       (ADD1 (GETEOFPTR scratch)))
                 (SETQ start (ADD1 (GETEOFPTR stream)))
                 (TEDIT.COPY (TEDIT.SETSEL scratch 1 (GETEOFPTR scratch)
                                    'LEFT)
                        (TEDIT.SETSEL stream start 0 'LEFT))
                 (CLOSEF scratch)
                 (TEDIT.SETSEL stream start 0 'RIGHT)
                 (TEDIT.NORMALIZECARET stream)
                 (TEDIT.STREAMCHANGEDP stream T)
                 (RETURN T])
```

(**DictTool.GetEntry**
```
    [LAMBDA (dict uniqueID prop)                                        (* jtm%: " 7-Apr-87 08:39")
       (COND
          [(NUMBERP uniqueID)
           (DICTCLIENT.ENUMERATE uniqueID (Dict.Prop dict 'RemoteDict]
          (T (DICTCLIENT.GETDEFINITION uniqueID (Dict.Prop dict 'RemoteDict])
```

(**TEdit.SetDictionary**
```
    [LAMBDA (stream dict)                                               ; Edited  6-Jan-89 12:24 by jtm:

             (* * sets the dictionary property for the window)

       (PROG (menuItems)
             (OR stream (SETQ stream (Dict.OutputStream)))
             [COND
                ((NULL dict)
                 [SETQ menuItems (for i in Dict.DictionaryList collect (LIST (Dict.Name i)
                                                                            (LIST 'QUOTE i)
                                                                            (if (Dict.Prop i 'RemoteDict)
                                                                                then "Calls the remote dictionary
                                                                                        server"]
                 [COND
                    ((NULL menuItems)
                     (TEDIT.PROMPTPRINT stream "Sorry, no dictionaries loaded." T))
                    ((EQ 1 (LENGTH menuItems))
                     (SETQ dict (CAR Dict.DictionaryList)))
                    (T (SETQ dict (MENU (create MENU
                                              ITEMS _ menuItems
                                              TITLE _ "dictionaries"
```

```
                                               CENTERFLG _ T]
                (COND
                   ((NULL dict)
                    (SETQ dict (STREAMPROP stream 'dict))
                    (TEDIT.PROMPTPRINT stream (CONCAT "Dictionary is " (AND dict (Dict.Name dict))
                                                     ".")
                         T)
                    (RETURN]
               (TEDIT.PROMPTPRINT stream (CONCAT "Setting dictionary to " (AND dict (Dict.Name dict))
                                                "..."")
                    T)
               (Dict.Open dict)
               (STREAMPROP stream 'dict dict)
          ;; 1/6/89 jtm: set TEdit.DefaultDictionary if this is the dictionary window or if it hasn't already been set.

               (if [OR (NULL TEdit.DefaultDictionary)
                       (AND (WINDOWP Dict.DefWindow)
                            (EQ stream (WINDOWPROP Dict.DefWindow 'TEXTSTREAM]
                   then (SETQ TEdit.DefaultDictionary dict))
               (TEDIT.PROMPTPRINT stream "done."))
          dict])
```


(**DictForStream**
```
  [LAMBDA (stream)                                                    ; Edited  6-Jan-89 12:26 by jtm:

     ;; 1/6/89 jtm: Try TEdit.DefaultDictionary if the stream doesn't have it's own dictionary.

     (COND
        ((STREAMPROP stream 'dict))
        (TEdit.DefaultDictionary)
        (T (TEdit.SetDictionary (Dict.OutputStream])
```


(**DictTool.Dictionaries**
```
  [LAMBDA (dict errorStream)                                          (* jtm%: "13-Nov-86 10:57")
     (DICTCLIENT.DICTIONARIES])
```


(**PARSEBYCOLONS**
```
  [LAMBDA (STRING COLONSORSPACES)                                     ; Edited 11-Jan-89 13:55 by jtm:

           (* * Actually, parse by SEMI-colons.)

     (LET (WORDS SEPARATOR (OLDPOS 1)
                 (POS 0))
          (COND
             ((STRINGP STRING)
              (SETQ SEPARATOR (COND
                                  ([AND COLONSORSPACES (NULL (STRPOS ";" STRING (ADD1 POS]
                                   " ")
                                  (T ";")))
              [while (SETQ POS (STRPOS SEPARATOR STRING (ADD1 POS)))
                 do (push WORDS (SUBSTRING STRING OLDPOS (SUB1 POS)))
                    (SETQ OLDPOS (for I from (ADD1 POS) thereis (NEQ 32 (NTHCHARCODE STRING I]
              [COND
                 ((AND (NEQ OLDPOS 0)
                       (ILEQ OLDPOS (NCHARS STRING)))
                  (push WORDS (SUBSTRING STRING OLDPOS (NCHARS STRING]
              (OR (DREVERSE WORDS)
                  STRING))
             (T STRING])
```


(**PrintPronunciationGuide**
```
  [LAMBDA (stream)                                                    (* jtm%: " 9-Feb-87 08:40")
     (LET (startPos)
          (SETQ startPos (GETFILEPTR stream))
          [for i pronCode on PronunciationGuide do (SETQ pronCode (CAR i))
                                                   (TEDIT.INSERT stream (CONCAT (ConvertPronunciation (CAR pronCode)
                                                                                   )
                                                                                ": "
                                                                                (CADR pronCode)
                                                                                " "
                                                                                (ConvertPronunciation (CADDR pronCode
                                                                                   ))
                                                                                (COND
                                                                                   ((CDR i)
                                                                                    "; ")
                                                                                   (T ""]
          (TEDIT.LOOKS stream '(FAMILY CLASSIC SIZE 10 FACE STANDARD)
                 (ADD1 startPos)
                 (IDIFFERENCE (GETFILEPTR stream)
                        startPos])
```


(**ConvertPronunciation**
```
  [LAMBDA (string)                                                    (* jtm%: " 6-Feb-87 17:38")
```

```
      (CONCATLIST (for i char nschars from 1 to (NCHARS string) join (SETQ char (NTHCHAR string i))
                                                              (SETQ nschars (CDR (FASSOC char PronunciationMap
                                                                                    )))
                                                              (COND
                                                                 ((NULL nschars)
                                                                  (LIST char))
                                                                 ((LISTP nschars)
                                                                  (COPY nschars))
                                                                 (T (LIST nschars])

)

(DEFINEQ
```

(**TEdit.SearchMenu**
```
  [LAMBDA (stream dict words)                                          ; Edited  1-Mar-94 10:28 by jtm:
    (LOAD? 'SEARCHMENU.MCOM)
    (if (NOT (OPENWP SearchMenu))
        then (SearchMenu.Create])
```

(**TEdit.PrintSearch**
```
  [LAMBDA (stream dict words)                                          (* jtm%: "13-Oct-87 10:11")

          (* * prints out the definition of the currently selected text.)

    (OR stream (SETQ stream (Dict.OutputStream)))
    (DictTool.TEditWrapper (OR dict (NerdForStream stream))
          'DictTool.PrintSearch stream words "Type keywords to search on:" "Searching for words using"
          'SEARCHKEYS])
```

(**DictTool.PrintSearch**
```
  [LAMBDA (dict selection stream)                                      (* jtm%: " 7-Apr-87 09:52")
    (LET (looks venn)
        (SETQ venn (DictTool.MergeSearch dict selection))
        [for i pos in venn do                                         (* printout header)
                            (SETQ pos (TEDIT.GETPOINT stream))
                            [for header on (CAR i) do (TEDIT.INSERT stream (CONCAT (CAR header)
                                                                               (COND
                                                                                  ((CDR header)
                                                                                   " ")
                                                                                  (T ": "]
                            (push looks (CONS pos (IDIFFERENCE (TEDIT.GETPOINT stream)
                                                         pos)))
                            [for word on (CADR i) do (TEDIT.INSERT stream (CONCAT (CAR word)
                                                                              (COND
                                                                                 ((CDR word)
                                                                                  "; ")
                                                                                 (T ""]
                            (TEDIT.INSERT stream (CHARACTER (CHARCODE CR)))
                            (TEDIT.INSERT stream (CHARACTER (CHARCODE CR]
                                           (* do the looks last to avoid messing up the text placement.)
        (TEDIT.LOOKS stream '(FAMILY TIMESROMAN SIZE 10 FACE STANDARD)
               1
               (TEDIT.GETPOINT stream))
        (for look in looks do (TEDIT.LOOKS stream '(FAMILY HELVETICA SIZE 10 FACE BOLD)
                                     (CAR look)
                                     (CDR look)))
        venn])
```

(**DictTool.MergeSearch**
```
  [LAMBDA (dict synonymclasses minKeywords minWord maxWord)            (* jtm%: " 2-Aug-88 13:15")
    (LET (minWord maxWord VennSearchFn)
        [for i on synonymclasses do (COND
                                       ((NLISTP (CAR i))
                                        (RPLACA i (LIST (CAR i]
        [COND
           ((AND (NULL minWord)
                 (NULL maxWord))
            (COND
               ((AND (EQUAL synonymclasses DictTool.LastSearch)
                     (NEQ 0 DictTool.MaxWords))
                (COND
                   ((NULL DictTool.LastWord)
                    (SETQ DictTool.LastWord 0)))
                (SETQ minWord (ADD1 DictTool.LastWord))
                [SETQ maxWord (COND
                                 ((EQ 0 DictTool.MaxWords)
                                  0)
                                 (T (IPLUS DictTool.MaxWords DictTool.LastWord]
                (SETQ DictTool.LastWord maxWord))
               (T (SETQ minWord 0)
                  (SETQ maxWord DictTool.MaxWords)
                  (SETQ DictTool.LastSearch synonymclasses)
                  (SETQ DictTool.LastWord maxWord]
```

```
      (COND
        [(InvertedDict.Prop dict 'RemoteDict)
         (DICTCLIENT.SEARCHFORWORD synonymclasses DictTool.MinKeywords (OR minWord 0)
                 (OR maxWord DictTool.MaxWords)
                 (InvertedDict.Prop dict 'RemoteDict]
        ((SETQ VennSearchFn (InvertedDict.Prop dict 'VENNSEARCHFN))
         (APPLY* VennSearchFn dict synonymclasses DictTool.MinKeywords (OR minWord 0)
                 (OR maxWord DictTool.MaxWords)))
        (T (InvertedDict.MergeSearch dict synonymclasses DictTool.MinKeywords (OR minWord 0)
                 (OR maxWord DictTool.MaxWords])
```

(**NerdForStream**
```
  [LAMBDA (stream)                                              (* jtm%: "17-Nov-87 11:14")

          (* * comment)

    (COND
       ((STREAMPROP stream 'nerd))
       ((STREAMPROP (Dict.OutputStream)
               'nerd))
       (T (TEdit.SetNerd (Dict.OutputStream])
```

(**TEdit.SetNerd**
```
  [LAMBDA (stream nerd)                                         (* jtm%: "14-Oct-87 12:50")

          (* * sets the dictionary property for the window)

    (PROG (menuItems)
          (OR stream (SETQ stream (Dict.OutputStream)))
          [COND
             ((NULL nerd)
              [SETQ menuItems (for i in InvertedDict.List collect (LIST (InvertedDict.Name i)
                                                                        (LIST 'QUOTE i)
                                                                        (if (InvertedDict.Prop i 'RemoteDict)
                                                                            then "Calls the remote dictionary
                                                                                  server"]
              [COND
                 ((NULL menuItems))
                 ((EQ 1 (LENGTH menuItems))
                  (SETQ nerd (CAR InvertedDict.List)))
                 (T (SETQ nerd (MENU (create MENU
                                             ITEMS _ menuItems
                                             TITLE _ "databases"
                                             CENTERFLG _ T]
              (COND
                 ((NULL nerd)
                  (SETQ nerd (STREAMPROP stream 'nerd))
                  (TEDIT.PROMPTPRINT stream (CONCAT "Database is " (AND nerd (InvertedDict.Name nerd))
                                                     "."))
                       T)
                  (RETURN]
          (TEDIT.PROMPTPRINT stream (CONCAT "Setting database to " (AND nerd (InvertedDict.Name nerd))
                                             "..."))
                 T)
          (InvertedDict.Open nerd)
          (STREAMPROP stream 'nerd nerd)
          (TEDIT.PROMPTPRINT stream "done.")
          (SETQ DictTool.LastSearch NIL)                        (* so that you can do the same search on a different data base.)
       )
     nerd])
```

(**DictTool.PromptForCutoff**
```
  [LAMBDA (STREAM)                                              (* jtm%: " 2-Feb-87 11:33")
    (OR STREAM (SETQ STREAM (Dict.OutputStream)))
    (TEDIT.PROMPTPRINT STREAM (CONCAT "Current maximum = " DictTool.MaxWords "."))
         T)
    (SETQ DictTool.MaxWords (RNUMBER "Enter the maximum number of words that each combination of keywords may
                              return.  (0 = no limit)"))
    (TEDIT.PROMPTPRINT STREAM (CONCAT "New maximum = " DictTool.MaxWords "."))
         T])
```

(**DictTool.PromptForKeywordCutoff**
```
  [LAMBDA (STREAM)                                              (* jtm%: " 2-Feb-87 11:33")
    (OR STREAM (SETQ STREAM (Dict.OutputStream)))
    (TEDIT.PROMPTPRINT STREAM (CONCAT "Current minimum = " DictTool.MinKeywords "."))
         T)
    (SETQ DictTool.MinKeywords (RNUMBER "Enter the minimum number of keywords that a word must have to be
                                  accepted. e.g. 2 = at least two keywords, 0 = all of the keywords given,
                                  -2 = all but two of the keywords given, etc."))
    (TEDIT.PROMPTPRINT STREAM (CONCAT "New minimum = " DictTool.MinKeywords "."))
         T])
```

(PARSESELECTION
```
  [LAMBDA (selection)                                          (* jtm%: "20-Mar-87 14:39")
    (LET (words temp)
        [for i charcode startPos alpha priorAlpha word from 1 to (ADD1 (NCHARS selection))
           do (SETQ charcode (NTHCHARCODE selection i))
              (SETQ priorAlpha alpha)
              [SETQ alpha (AND charcode (OR (ALPHACHARP charcode)
                                            (EQ charcode (CHARCODE -]
              [COND
                 ((AND alpha (NULL priorAlpha))
                  (SETQ startPos i))
                 ((AND priorAlpha (NULL alpha))
                  (SETQ word (SUBSTRING selection startPos (SUB1 i)))
                  (COND
                     ((NULL temp)
                      (push temp word))
                     (T (NCONC1 temp word]
              (COND
                 ((EQ charcode (CHARCODE %())
                  (SETQ words (APPEND words temp))
                  (SETQ temp NIL))
                 ((EQ charcode (CHARCODE %)))
                  (SETQ words (APPEND words (LIST temp)))
                  (SETQ temp NIL]
        (SETQ words (APPEND words temp))
        words])
)

(DEFINEQ
```

(TEdit.PrintPhraseSearch
```
  [LAMBDA (stream dict words)                                  (* jtm%: "26-May-87 09:26")

          (* * prints out the definitions that have a particular phrase in them.)

    (OR stream (SETQ stream (Dict.OutputStream)))
    (DictTool.TEditWrapper (NerdForStream stream)
         'DictTool.PrintPhraseSearch stream words "Type phrase to search for:" "Searching for phrase using"
         'SEARCHPHRASE])
```

(DictTool.PrintPhraseSearch
```
  [LAMBDA (dict selection stream)                              (* jtm%: "26-May-87 09:29")
    (LET (looks words fn pos)
        [SETQ words (COND
                       [(InvertedDict.Prop dict 'RemoteDict)
                        (DICTCLIENT.SEARCHFORPHRASE selection (InvertedDict.Prop dict 'RemoteDict]
                       ((SETQ fn (InvertedDict.Prop dict 'SEARCHFORPHRASEFN))
                        (APPLY* fn dict selection]
        (SETQ pos (TEDIT.GETPOINT stream))
        (TEDIT.INSERT stream (CONCAT "%"" selection "%": "))
        (SETQ looks (CONS pos (IDIFFERENCE (TEDIT.GETPOINT stream)
                                           pos)))
        [for word on words do (TEDIT.INSERT stream (CONCAT (CAR word)
                                                           (COND
                                                              ((CDR word)
                                                               "; ")
                                                              (T ""]
        (TEDIT.INSERT stream (CHARACTER (CHARCODE CR)))
        (TEDIT.INSERT stream (CHARACTER (CHARCODE CR)))              (* do the looks last to avoid messing up the text placement.)
        (TEDIT.LOOKS stream '(FAMILY TIMESROMAN SIZE 10 FACE STANDARD)
               1
               (TEDIT.GETPOINT stream))
        (TEDIT.LOOKS stream '(FAMILY HELVETICA SIZE 10 FACE BOLD)
               (CAR looks)
               (CDR looks))
        words])
)

(DEFINEQ
```

(TEdit.PrintSynonyms
```
  [LAMBDA (stream dict words)                                  (* jtm%: "14-Oct-87 12:44")

          (* * prints out the synonyms of the selected word)

    (DictTool.TEditWrapper T 'DictTool.PrintSynonyms stream words "synonym to look up:" "Getting synonyms for"
         'USERSYNONYM])
```

(REMOVEALL
```
  [LAMBDA (X L)                                                (* jtm%: "14-Oct-87 12:39")
    (for TAIL on X unless (EQUAL L (CAR TAIL)) collect (COND
                                                          ((LISTP (CAR TAIL))
                                                           (REMOVEALL (CAR TAIL)
```

```
                                                              L))
                                               (T (COPY (CAR TAIL])
```

(**CONVERTFUNCTIONSTOFORMS**
```
  [LAMBDA (LIST)                                        (* jtm%: "14-Oct-87 12:57")
    (for ELT in LIST collect (COND
                                 [(EQ (CAR ELT)
                                      'FUNCTION)
                                  (LIST 'QUOTE (LIST (CADR ELT]
                                 ((LISTP ELT)
                                  (CONVERTFUNCTIONSTOFORMS ELT))
                                 (T (COPY ELT])
```

(**TEdit.PrintNounSynonyms**
```
  [LAMBDA (stream dict words)                           (* jtm%: "14-Oct-87 12:43")
```

            (* * prints out the synonyms of the selected word)

```
    (DictTool.TEditWrapper T (FUNCTION DictTool.PrintNounSynonyms)
           stream words "synonym to look up:" "Getting noun synonyms for" 'USERSYNONYM])
```

(**DictTool.PrintNounSynonyms**
```
  [LAMBDA (dict words stream)                           (* jtm%: "14-Oct-87 12:32")
    (DictTool.PrintSynonyms dict words stream "n"])
```

(**DictTool.PrintVerbSynonyms**
```
  [LAMBDA (dict words stream)                           (* jtm%: "14-Oct-87 12:34")
    (DictTool.PrintSynonyms dict words stream "v"])
```

(**DictTool.PrintAdjSynonyms**
```
  [LAMBDA (dict words stream)                           (* jtm%: "14-Oct-87 12:35")
    (DictTool.PrintSynonyms dict words stream "adj"])
```

(**TEdit.PrintVerbSynonyms**
```
  [LAMBDA (stream dict words)                           (* jtm%: "14-Oct-87 12:44")
```

            (* * prints out the synonyms of the selected word)

```
    (DictTool.TEditWrapper T 'DictTool.PrintVerbSynonyms stream words "synonym to look up:" "Getting verb synonyms
           for" 'USERSYNONYM])
```

(**TEdit.PrintAdjSynonyms**
```
  [LAMBDA (stream dict words)                           (* jtm%: "14-Oct-87 12:44")
```

            (* * prints out the synonyms of the selected word)

```
    (DictTool.TEditWrapper T 'DictTool.PrintAdjSynonyms stream words "synonym to look up:" "Getting adjective
           synonyms for" 'USERSYNONYM])
```

(**DictTool.PrintSynonyms**
```
  [LAMBDA (dict words stream form)                      (* jtm%: "14-Oct-87 12:31")
    (PROG (synonyms found startPos headerPos endPos)
          (for word inside words do (SETQ synonyms (DICTCLIENT.SYNONYMS word))
                                    (AND synonyms (SETQ found T))
                                    (SETQ startPos (TEDIT.GETPOINT stream))
                                    (TEDIT.INSERT stream (CONCAT word ": "))
                                    (SETQ headerPos (TEDIT.GETPOINT stream))
                                    [for class in synonyms when (OR (NULL form)
                                                                    (EQUAL form (CAR class)))
                                       do (TEDIT.INSERT stream (CONCAT (CAR class)
                                                                       ": "))
                                          [for word on (CDR class) do (TEDIT.INSERT stream (CONCAT (CAR word)
                                                                                             (COND
                                                                                                ((CDR word)
                                                                                                 ", ")
                                                                                                (T ""]
                                          (TEDIT.INSERT stream (CHARACTER (CHARCODE CR]
                                    (TEDIT.INSERT stream (CHARACTER (CHARCODE CR)))
                                    (SETQ endPos (TEDIT.GETPOINT stream))
                                    (TEDIT.LOOKS stream '(FAMILY HELVETICA SIZE 10 FACE BOLD)
                                           startPos
                                           (IDIFFERENCE (SUB1 headerPos)
                                                  startPos))
                                    (TEDIT.LOOKS stream '(FAMILY TIMESROMAN SIZE 10 FACE STANDARD)
                                           headerPos
                                           (IDIFFERENCE endPos headerPos))
                                    (TEDIT.SETSEL stream endPos 0 'LEFT))
          (RETURN found])
```

)

(DEFINEQ

(**DictTool.TEditWrapper**
  [LAMBDA (dict proc stream selection promptString waitString cachePropName)
                                                                    (* jtm%: "29-Jun-88 09:56")

          (* * handles the TEdit user interface)

    (PROG (scratchStream textStream startPos startTime textObj)

          (* * set things up)

          [COND
             ((NULL stream)
              (SETQ stream (**Dict.OutputStream**]
          [COND
             ((NULL dict)
              (TEDIT.PROMPTPRINT stream "Please select a dictionary." T)
              (RETURN))
             ((NULL selection)
              (SETQ selection (TEDIT.SEL.AS.STRING stream))
              (COND
                 ((ILEQ (NCHARS selection)
                        1)
                  (SETQ selection NIL)))

          (* * "rht 4/27/88: No longer passes value of PROMPTWINDOW textprop to MOUSECONFIRM since it could be DON'T. Now
          looks for promptwindow on the WINDOWPROP of the stream's main window.")

              (COND
                 [(AND selection (MOUSECONFIRM (CONCAT "CONFIRM INPUT: " selection)
                                               ""
                                               (CAR (WINDOWPROP (CAR (**fetch** (TEXTOBJ \WINDOW) **of** (TEXTOBJ stream)))
                                                                'PROMPTWINDOW]
                 ([NULL (SETQ selection (TEDIT.GETINPUT stream (OR promptString "input: ")
                                               (AND cachePropName (STREAMPROP stream cachePropName]
                  (TEDIT.PROMPTPRINT stream " Aborted." T)
                  (RETURN]

          (* * print the results.)

          (SETQ startTime (CLOCK 0))
          (AND cachePropName (STREAMPROP stream cachePropName selection))
          (TEDIT.PROMPTPRINT stream (CONCAT (OR waitString "processing")
                                           " '" selection "' . . . ")
                  T)
          [RESETSAVE (OUTPUT (CAR (WINDOWPROP Dict.DefWindow 'PROMPTWINDOW]
                                                                    (* redirects errors to the promptwindow)
          [SETQ scratchStream (OPENTEXTSTREAM NIL NIL NIL NIL '(LEAVETTY]
          (COND
             [(APPLY* proc dict selection scratchStream)
              (**TEDIT.INCLUDESTREAM** (**Dict.OutputStream**)
                      scratchStream)
              (COND
                 (DictTool.TimeOperation (TEDIT.PROMPTPRINT stream (CONCAT "Elapsed Time: "
                                                                   (QUOTIENT (DIFFERENCE (CLOCK 0)
                                                                                         startTime)
                                                                             1000.0)
                                                                   " seconds.")))
                 (T (TEDIT.PROMPTPRINT stream "Done."]
             (T (TEDIT.PROMPTPRINT stream "not found.")
                (TEDIT.PROMPTFLASH stream)))
          (CLOSEF scratchStream])

(**Dict.OutputStream**
  [LAMBDA (REGION)   ; Edited 12-Oct-88 09:20 by rmk:
                                                                    ; Edited 7-Oct-88 12:01 by jtm:
    (LET (TEXTSTREAM HIDDENFN UNHIDEFN)
         [COND
            ((AND Dict.DefWindow (NOT (OPENWP Dict.DefWindow))
                  (WINDOWPROP Dict.DefWindow 'TEXTSTREAM))   (* window is shrunk.)
             (OPENW Dict.DefWindow))
            ((AND Dict.DefWindow (CL:FIND-PACKAGE "ROOMS")
                  (SETQ HIDDENFN (CL:FIND-SYMBOL "WINDOW-HIDDEN?" "ROOMS"))
                  (GETD HIDDENFN)
                  (CL:FUNCALL HIDDENFN Dict.DefWindow))

          (* the FIND-SYMBOL calls are used to avoid a break that happens when you access the ROOMS package when it hasn't
          been loaded.)

             (SETQ UNHIDEFN (CL:FIND-SYMBOL "UN-HIDE-WINDOW" "ROOMS"))
             (CL:FUNCALL UNHIDEFN Dict.DefWindow))
            ((OR (NULL Dict.DefWindow)
                 (NOT (OPENWP Dict.DefWindow)))

```
            (SETQ Dict.DefWindow (CREATEW [OR REGION (AND Dict.DefWindow (WINDOWPROP Dict.DefWindow
                                                                                    'REGION]
                                             "Definitions"))
            (SETQ TEXTSTREAM (OPENTEXTSTREAM NIL Dict.DefWindow))
            (replace TXTFILE of (TEXTOBJ TEXTSTREAM) with "Definitions")
```

(* do the replace before you spawn a TEDIT process in order to avoid a race condition where sometimes the label on the icon was "T")

```
            (PROCESSPROP (TEDIT TEXTSTREAM Dict.DefWindow NIL '(LEAVETTY))
                   'NAME
                   'DICTIONARY]
       (TEXTSTREAM Dict.DefWindow])
```

(**DictTool.PromptStream**
```
  [LAMBDA (stream)                                              (* jtm%: "29-Sep-86 11:11")
    (COND
       [(STREAMPROP stream)
        (for window inside (STREAMPROP stream 'WINDOW) do (COND
                                                              ((WINDOWPROP window 'PROMPTWINDOW)
                                                               (RETURN (WINDOWPROP window 'PROMPTWINDOW]
       (T PROMPTWINDOW])
)
```

```
(DEFINEQ
```

(**DictTool.Init**
```
  [LAMBDA (serverName)                                          (* jtm%: "13-Oct-87 11:37")
    (PROG (analyzer dict wordNerd)
```

(* * start up the interface)

(**Dict.AddCommands**)

(* * create the analyzer)

```
       [Analyzer.Establish (SETQ analyzer (create Morphalyzer
                                               analyzerName _ 'DictServer
                                               openFn _ (FUNCTION DictTool.OpenAnalyzer)
                                               closeFn _ (FUNCTION DictTool.Close)
                                               analyzeFn _ (FUNCTION DictTool.Analyze)
                                               correctionsFn _ (FUNCTION DictTool.Corrections]
       (Analyzer.Prop analyzer 'CountWords (FUNCTION DictTool.CountWords))
```

(* * create the dictionary)

```
       [Dict.Establish (SETQ dict (create Dict
                                       dictName _ 'DictServer
                                       openFn _ (FUNCTION DictTool.OpenDictionary)
                                       closeFn _ (FUNCTION DictTool.Close)
                                       getEntryFn _ (FUNCTION DictTool.GetEntry)
                                       printEntryFn _ (FUNCTION DictTool.PrintDefinition]
```

(* * create the remote inverted dict.)

```
       [InvertedDict.Establish (SETQ wordNerd (create INVERTEDDICT
                                                   INVERTEDDICTNAME _ 'DictServer]
       (InvertedDict.Prop wordNerd 'OPENFN (FUNCTION DictTool.OpenNerd))
       (InvertedDict.Prop wordNerd 'DICTIONARY dict])
```

(**DictTool.Open**
```
  [LAMBDA (analyzer errors)                                     (* jtm%: "13-Oct-87 10:43")
```

(* * we import the interface here instead of in DictTool.Init to avoid hanging the LOAD.)

```
    (PROG (analyzers dictionaries menuItems)
          (COND
             [(type? Morphalyzer analyzer)
              (COND
                 ((NULL (Analyzer.Prop analyzer 'RemoteDict))
                  (SETQ analyzers (DictTool.Analyzers analyzer errors))
                  [SETQ menuItems (for i in analyzers collect (LIST i (LIST 'QUOTE i]
                  (COND
                     ((IGEQ 1 (LENGTH menuItems))
                      (Analyzer.Prop analyzer 'RemoteDict (CAR analyzers)))
                     (T (Analyzer.Prop analyzer 'RemoteDict (OR (MENU (create MENU
                                                                          ITEMS _ menuItems
                                                                          TITLE _ (CONCAT (fetch (Morphalyzer
                                                                                                   analyzerName
                                                                                                   )
                                                                                             of analyzer)
                                                                                       " analyzers")
                                                                          CENTERFLG _ T))
                                                                (CAR analyzers)))
```

```
                                    (for i analyzerName in analyzers do (SETQ analyzerName (MKATOM (CONCAT
                                                                                    (fetch (Morphalyzer
                                                                                              analyzerName
                                                                                              )
                                                                                       of analyzer)
                                                                                    ": " i)))
                                                          (COND
                                                             ([NOT (for j in Analyzer.List
                                                                      thereis (EQ analyzerName (Analyzer.Name
                                                                                                  j]
                                                               (push Analyzer.List (create Morphalyzer
                                                                                      copying analyzer))
                                                               (Analyzer.Prop (CAR Analyzer.List)
                                                                      'RemoteDict i]
                    ((type? Dict analyzer)
                     (COND
                        ((NULL (Dict.Prop analyzer 'RemoteDict))
                         (SETQ dictionaries (DictTool.Dictionaries analyzer errors))
                         [SETQ menuItems (for i in dictionaries collect (LIST i (LIST 'QUOTE i]
                         (COND
                            ((IGEQ 1 (LENGTH menuItems))
                             (Dict.Prop analyzer 'RemoteDict (CAR dictionaries)))
                            (T (Dict.Prop analyzer 'RemoteDict (OR (MENU (create MENU
                                                                            ITEMS _ menuItems
                                                                            TITLE _ (CONCAT (fetch (Dict dictName)
                                                                                               of analyzer)
                                                                                       " dictionaries")
                                                                            CENTERFLG _ T))
                                                                   (CAR dictionaries)))
                            (for i dictName in dictionaries do (SETQ dictName (MKATOM (CONCAT (fetch (Dict dictName)
                                                                                                of analyzer)
                                                                                       ": " i)))
                                                          (COND
                                                             ([NOT (for j in Dict.DictionaryList
                                                                      thereis (EQ dictName (Dict.Name j]
                                                               (push Dict.DictionaryList (create Dict copying
                                                                                                  analyzer
                                                                                                  ))
                                                               (Dict.Prop (CAR Dict.DictionaryList)
                                                                      'RemoteDict i])
```

(**DictTool.OpenDictionary**
```
  [LAMBDA (dict errors)                                             (* jtm%: "13-Oct-87 10:38")
```

          (* * we import the interface here instead of in DictTool.Init to avoid hanging the LOAD.)

```
    (PROG (dictionaries menuItems)
          (COND
             ((type? Dict dict)
              (COND
                 ((NULL (Dict.Prop dict 'RemoteDict))
                  (SETQ dictionaries (DICTCLIENT.DICTIONARIES))
                  [SETQ menuItems (for i in dictionaries collect (LIST i (LIST 'QUOTE i]
                  (COND
                     ((IGEQ 1 (LENGTH menuItems))
                      (Dict.Prop dict 'RemoteDict (CAR dictionaries)))
                     (T (Dict.Prop dict 'RemoteDict (OR (MENU (create MENU
                                                                 ITEMS _ menuItems
                                                                 TITLE _ (CONCAT (fetch (Dict dictName)
                                                                                    of dict)
                                                                            " dictionaries")
                                                                 CENTERFLG _ T))
                                                        (CAR dictionaries)))
                     (for i dictName in dictionaries do (SETQ dictName (MKATOM (CONCAT (fetch (Dict dictName)
                                                                                         of dict)
                                                                                ": " i)))
                                                   (COND
                                                      ([NOT (for j in Dict.DictionaryList
                                                               thereis (EQ dictName (Dict.Name j]
                                                        (push Dict.DictionaryList (create Dict copying dict)
                                                               )
                                                        (Dict.Prop (CAR Dict.DictionaryList)
                                                               'RemoteDict i])
```

(**DictTool.OpenAnalyzer**
```
  [LAMBDA (analyzer errors)                                         (* jtm%: "13-Oct-87 10:43")
```

          (* * we import the interface here instead of in DictTool.Init to avoid hanging the LOAD.)

```
    (PROG (analyzers menuItems)
          (COND
             ((type? Morphalyzer analyzer)
              (COND
                 ((NULL (Analyzer.Prop analyzer 'RemoteDict))
                  (SETQ analyzers (DICTCLIENT.LANGUAGES))
```

```
                        [SETQ menuItems (for i in analyzers collect (LIST i (LIST 'QUOTE i]
                        (COND
                           ((IGEQ 1 (LENGTH menuItems))
                            (Analyzer.Prop analyzer 'RemoteDict (CAR analyzers)))
                           (T (Analyzer.Prop analyzer 'RemoteDict (OR (MENU (create MENU
                                                                                    ITEMS _ menuItems
                                                                                    TITLE _ (CONCAT (fetch (Morphalyzer
                                                                                                              analyzerName
                                                                                                              )
                                                                                                        of analyzer)
                                                                                                    " analyzers")
                                                                                    CENTERFLG _ T))
                                                                       (CAR analyzers)))
                        (for i analyzerName in analyzers do (SETQ analyzerName (MKATOM (CONCAT
                                                                                         (fetch (Morphalyzer
                                                                                                   analyzerName
                                                                                                   )
                                                                                           of analyzer)
                                                                                         ": " i)))
                           (COND
                              ([NOT (for j in Analyzer.List
                                        thereis (EQ analyzerName (Analyzer.Name
                                                                   j]
                               (push Analyzer.List (create Morphalyzer
                                                            copying analyzer))
                               (Analyzer.Prop (CAR Analyzer.List)
                                              'RemoteDict i])
```

(**DictTool.OpenNerd**
```
 [LAMBDA (nerd errors)                                        (* jtm%: "13-Oct-87 14:35")

            (* * we import the interface here instead of in DictTool.Init to avoid hanging the LOAD.)

   (PROG (nerds menuItems dict remote)
         (COND
            ((type? INVERTEDDICT nerd)
             (COND
                ((NULL (InvertedDict.Prop nerd 'RemoteDict))
                 (SETQ nerds (DICTCLIENT.RESOURCES 'INDICES))
                 [SETQ menuItems (for i in nerds collect (LIST i (LIST 'QUOTE i]
                 (COND
                    ((IGEQ 1 (LENGTH menuItems))
                     (InvertedDict.Prop nerd 'RemoteDict (CAR nerds)))
                    (T [InvertedDict.Prop nerd 'RemoteDict (SETQ remote
                                                              (OR (MENU (create MENU
                                                                                ITEMS _ menuItems
                                                                                TITLE _ (CONCAT
                                                                                          (fetch (INVERTEDDICT
                                                                                                    INVERTEDDICTNAME
                                                                                                    )
                                                                                            of nerd)
                                                                                          " databases")
                                                                                CENTERFLG _ T))
                                                                  (CAR nerds]
                 (COND
                    ((SETQ dict (InvertedDict.Prop nerd 'DICTIONARY))
                     (SETQ dict (COPYALL dict))
                     (Dict.Prop dict 'RemoteDict remote)
                     (InvertedDict.Prop nerd 'DICTIONARY dict)))
                 (for i in nerds do (COND
                                       ((NOT (InvertedDictFromName (fetch (INVERTEDDICT INVERTEDDICTNAME)
                                                                     of nerd)
                                                i))
                                        (push InvertedDict.List (create INVERTEDDICT copying nerd))
                                        (InvertedDict.Prop (CAR InvertedDict.List)
                                                           'RemoteDict i)
                                        (COND
                                           ((SETQ dict (InvertedDict.Prop (CAR InvertedDict.List)
                                                          'DICTIONARY))
                                            (SETQ dict (COPYALL dict))
                                            (Dict.Prop dict 'RemoteDict i)
                                            (InvertedDict.Prop (CAR InvertedDict.List)
                                                               'DICTIONARY dict])
```

(**Dict.AddCommands**
```
 [LAMBDA NIL                                                  ; Edited 27-Mar-91 17:19 by jtm:
                                                              ; Edited 31-May-89 15:06 by jtm:
                                                              ; Edited 31-May-89 15:00 by jtm:
                                                              ; Edited 31-May-89 13:36 by jtm:
   (LET (menuItems)
        [SETQ menuItems '(Dictionary (FUNCTION TEdit.PrintDefinition)
                                     "Prints the definition of the selected word.  Prompts the user for a word if
                                     there isn't a selection."
                                     (SUBITEMS (Set% Dictionary (FUNCTION TEdit.SetDictionary)
                                                     "Gives the user a menu of dictionaries to select from.")
```

```
                                          (Get% Definition (FUNCTION TEdit.PrintDefinition)
                                                      "Prints the definition of the selected word.  Prompts the user
                                                      for a word if there isn't a selection.")
                                          (Get% Synonyms (FUNCTION TEdit.PrintSynonyms)
                                                      "Prints the synonyms of the selected word.  Prompts the user for
                                                      a word if there isn't a selection."
                                                      (SUBITEMS (nouns (FUNCTION TEdit.PrintNounSynonyms)
                                                                        "Only prints the noun form synonyms.")
                                                              (verbs (FUNCTION TEdit.PrintVerbSynonyms)
                                                                        "Only prints the verb form synonyms.")
                                                              (adjectives (FUNCTION TEdit.PrintAdjSynonyms)
                                                                        "Only prints the adjective form synonyms.")))
                                          (Relevance% Feedback (FUNCTION TEdit.SearchMenu))
                                          (|Search For Word| (FUNCTION TEdit.PrintSearch)
                                                      "Prints the words in the dictionary containing at least two of
                                                      the keywords in the selection.  Prompts the user for keywords if
                                                      there aren't any keywords selected."
                                                      (SUBITEMS (Set% Database (FUNCTION TEdit.SetNerd)
                                                                        "Gives the user a menu of dictionaries to select
                                                                        from.")
                                                              (Max% Words (FUNCTION DictTool.PromptForCutoff)
                                                                        "Lets the user set the maximum number of words to
                                                                        be returned for a set of keywords.")
                                                              (Min% Keywords (FUNCTION DictTool.PromptForKeywordCutoff)
                                                                        "Lets the user determine the minimum number of
                                                                        keywords needed by a word for it to accepted.")
                                          (|Search For Phrase| (FUNCTION TEdit.PrintPhraseSearch)
                                                      "Searches a dictionary for a particular phrase,
                                                      using the Search For Word database to narrow the
                                                      search.  This can be an expensive operation, so
                                                      please use it sparingly."]
            (TEDIT.ADD.MENUITEM TEDIT.DEFAULT.MENU menuItems)
            ;; add menu item to Lafite's display menu if Lafite has been loaded.

            [COND
                ((BOUNDP '\LAFITE.ACTIVE)
                 (pushnew LAFITE.EXTRA.DISPLAY.COMMANDS menuItems)
                 (if \LAFITE.ACTIVE
                     then (LAFITE.COMPUTE.CACHED.VARS]
            (PUTASSOC 'Dictionary (CONVERTFUNCTIONSTOFORMS (CDR menuItems))
                    BackgroundMenuCommands)
            (SETQ BackgroundMenu NIL])
```

(**DictTool.Close**

```
  [LAMBDA (analyzer)                                        (* jtm%: "13-Nov-86 10:58")
    (CLOSEF DICTSERVERSTREAM])

)

(DEFINEQ
```

(**DictTool.Analyze**

```
  [LAMBDA (analyzer stream fromLoc length analFn)           (* jtm%: "14-Apr-87 14:16")
    (PROG (buffer bufferStream bufferLength char returnValue userWords (substring (ALLOCSTRING 0 32))
                (maxBufferLength 5100)
                (offset fromLoc))
        (SETQ userWords (Analyzer.Prop analyzer 'UserDict))
        [COND
            ((NULL stream)
             NIL)
            [(STRINGP stream)
             (HELP "DictTool.Analyze not implemented for STRING")
             [SETQ returnValue (DICTCLIENT.PROOFREAD stream (Analyzer.Prop analyzer 'RemoteDict]
             (COND
                ((EQUAL 0 (CDR returnValue))
                 (RETURN (SETQ returnValue NIL]
            (T

        (* * break up the stream into strings of ~5000 characters.)

                (SETQ buffer (ALLOCSTRING (IMIN length maxBufferLength)))
                (SETQ bufferStream (OPENSTRINGSTREAM buffer 'OUTPUT))
                (SETFILEPTR stream fromLoc)
                (while (IGREATERP length 0)
                   do (SETFILEPTR bufferStream 0)
                      (SETQ bufferLength 0)
                      [do (SETQ char (BIN stream))
                          [COND
                             ((OR (NOT (NUMBERP char))
                                  (IGREATERP char 255))
                              (SETQ char (CHARCODE % ]
                          (BOUT bufferStream char)
                          (add length -1)
                          (add bufferLength 1)
                          (COND
                             ((EQUAL length 0)
```

```
                                    (RETURN))
                                  ((EQUAL bufferLength maxBufferLength)
                                   (RETURN))
                                  ((IGREATERP bufferLength (IDIFFERENCE maxBufferLength 200))
                                   (COND
                                      ([OR (EQ char (CHARCODE CR))
                                           (AND (EQ char (CHARCODE SP))
                                                (IGREATERP bufferLength (IDIFFERENCE maxBufferLength 50]
                                       (RETURN]
                          [SETQ returnValue (DICTCLIENT.PROOFREAD (COND
                                                                     ((EQUAL bufferLength (NCHARS buffer))
                                                                      buffer)
                                                                     (T (SUBSTRING buffer 1 bufferLength substring)))
                                                                   )
                                             (Analyzer.Prop analyzer 'RemoteDict]
                          (COND
                             ((EQUAL 0 (CDR returnValue))
                              (SETQ returnValue NIL)
                              (add offset bufferLength))
                             ((AND userWords (Dict.GetEntry userWords (SUBSTRING buffer (ADD1 (CAR returnValue))
                                                                                (IPLUS (CAR returnValue)
                                                                                       (CDR returnValue))
                                                                       substring)))
                              [add length (IPLUS bufferLength (IMINUS (IPLUS (CAR returnValue)
                                                                             (CDR returnValue]
                               (add offset (IPLUS (CAR returnValue)
                                                  (CDR returnValue)))
                              (SETFILEPTR stream offset)
                              (SETQ returnValue NIL))
                             (returnValue (add (CAR returnValue)
                                               offset)
                                  (RETURN returnValue]
                  (RETURN returnValue))
```

(**DictTool.Analyzers**
```
  [LAMBDA (analyzer errorStream)                                    (* jtm%: "13-Nov-86 10:57")

        (* * wraps DictTool.RPCCall around a call to RemoteDict.Analyzers)

  (DICTCLIENT.LANGUAGES)]
```

(**DictTool.Pronunciation**
```
  [LAMBDA (word dictName)                                           (* jtm%: "13-Nov-86 10:58")
    [COND
       ((NOT (STRINGP word))
        (SETQ word (MKSTRING word]
    [COND
       ((NULL dictName)
        (SETQ dictName 'AmericanHeritage]
    (DICTCLIENT.PRONUNCIATION word dictName])
```

(**DictTool.Corrections**
```
  [LAMBDA (analyzer stream loc len)                                 (* jtm%: "13-Nov-86 10:58")
    (DICTCLIENT.CORRECTIONS (COND
                              ((STRINGP stream)
                               stream)
                              (T (STREAM.FETCHSTRING stream loc len)))
         (Analyzer.Prop analyzer 'RemoteDict])
```

(**DictTool.CountWords**
```
  [LAMBDA (analyzer stream fromLoc length analFn)                   (* jtm%: "13-Nov-86 14:19")
    (PROG (buffer bufferStream bufferLength char (n 0)
                 (substring (ALLOCSTRING 0 32))
                 (maxBufferLength 5100)
                 (offset fromLoc))
         [COND
            ((NULL stream)
             NIL)
            [(STRINGP stream)
             (HELP "DictTool.Analyze not implemented for STRING")
             [SETQ n (DICTCLIENT.PROOFREAD stream (Analyzer.Prop analyzer 'RemoteDict]
             (COND
                ((EQUAL 0 (CDR n))
                 (RETURN (SETQ n NIL]
            (T

        (* * break up the stream into strings of ~5000 characters.)

             (SETQ buffer (ALLOCSTRING (IMIN length maxBufferLength)))
             (SETQ bufferStream (OPENSTRINGSTREAM buffer 'OUTPUT))
             (SETFILEPTR stream fromLoc)
             (while (IGREATERP length 0)
                do (SETFILEPTR bufferStream 0)
```

```
                                (SETQ bufferLength 0)
                                [do (SETQ char (BIN stream))
                                    [COND
                                        ((OR (NOT (NUMBERP char))
                                             (IGREATERP char 255))
                                         (SETQ char (CHARCODE % ]
                                    (BOUT bufferStream char)
                                    (add length -1)
                                    (add bufferLength 1)
                                    (COND
                                        ((EQUAL length 0)
                                         (RETURN))
                                        ((EQUAL bufferLength maxBufferLength)
                                         (RETURN))
                                        ((IGREATERP bufferLength (IDIFFERENCE maxBufferLength 200))
                                         (COND
                                             ([OR (EQ char (CHARCODE CR))
                                                  (AND (EQ char (CHARCODE SP))
                                                       (IGREATERP bufferLength (IDIFFERENCE maxBufferLength 50]
                                             (RETURN]
                                    [add n (DICTCLIENT.COUNTWORDS (COND
                                                                      ((EQUAL bufferLength (NCHARS buffer))
                                                                       buffer)
                                                                      (T (SUBSTRING buffer 1 bufferLength substring)))
                                           (Analyzer.Prop analyzer 'RemoteDict]
                                    (add offset bufferLength]
                (RETURN n])
    )
```

          (* * FINDWORD & SUBSTITUTEWORD)

```
(DEFINEQ
```

(**DictTool.FindWord**
```
  [LAMBDA (STREAM WORD CH)                                               (* jtm%: "30-Apr-86 10:30")
                                                                         (* the TEDIT interface to FindWord)

    (PROG (SEL (TEXTOBJ (TEXTOBJ STREAM)))

          (* * prompt the user for a string if none is given.)

          [COND
              ((NULL WORD)
               (SETQ WORD (TEDIT.GETINPUT TEXTOBJ "Word to find: " (WINDOWPROP W 'TEDIT.LAST.FIND.STRING)
                                  (CHARCODE (EOL LF ESC]

          (* * search for the word.)

          [COND
              (WORD (SETQ SEL (fetch SEL of TEXTOBJ))
                    (\SHOWSEL SEL NIL NIL)
                    (TEDIT.PROMPTPRINT TEXTOBJ "Searching..." T)
                    (SETQ CH (LingFns.FindWord STREAM WORD CH))

          (* * show the user what we found)

                    (COND
                        (CH (TEDIT.PROMPTPRINT TEXTOBJ "Done.")
                            (replace CH# of SEL with (CAR CH))
                            [replace DCH of SEL with (IPLUS (CAR CH)
                                                           (IMINUS (CADR CH]
                            (replace CHLIM of SEL with (ADD1 (CADR CH)))
                            (replace POINT of SEL with 'RIGHT)
                            (TEDIT.RESET.EXTEND.PENDING.DELETE SEL)
                            (\FIXSEL SEL TEXTOBJ)
                            (TEDIT.NORMALIZECARET TEXTOBJ)
                            (\SHOWSEL SEL NIL T)
                            (WINDOWPROP W 'TEDIT.LAST.FIND.STRING WORD)
                                                                         (* And get it into the window)
                            )
                        (T (TEDIT.PROMPTPRINT TEXTOBJ "(not found).")
                           (\SHOWSEL SEL NIL T]
          (replace \INSERTNEXTCH of TEXTOBJ with -1])
```

(**DictTool.SubstituteWord**
```
  [LAMBDA (TEXTSTREAM PATTERN REPLACEMENT CONFIRM? DICTNAME)            (* jtm%: "24-Mar-87 08:58")
                                                                         (* this procedure is a modification of TEDIT.SUBSTITUTE.)
    (PROG (SEARCHSTRING REPLACESTRING ABORTFLG OUTOFRANGEFLG (TEXTOBJ (TEXTOBJ TEXTSTREAM))
                ENDCHAR# STARTCHAR# RANGE (REPLACEDFLG 0)
                (YESLIST '("y" "Y" "yes" "Yes" "YES" "T"))
                CONFIRMFLG SEL PC# SELCH# SELCHLIM SELPOINT CRSEEN DICT)
          (COND
              ([NULL (SETQ SEARCHSTRING (OR PATTERN (TEDIT.GETINPUT TEXTOBJ "Search word:]
                                                                         (* If the search pattern is empty, bail out.)
               (TEDIT.PROMPTPRINT TEXTOBJ "[Aborted]")
```

```
            (RETURN)))
        (SETQ REPLACEMENT (OR REPLACEMENT (TEDIT.GETINPUT TEXTOBJ "Replace word:")
                              ""))                                          (* jtm%: use REPLACEMENT for the original,
                                                                           REPLACESTRING for the modified word.)
        (SETQ CRSEEN (STRPOS (CHARACTER (CHARCODE CR))
                             REPLACEMENT))                                  (* jtm%: use REPLACEMENT instead of REPLACESTRING)
        (COND
           (PATTERN                                                        (* If a pattern is specd in the call, use the caller's confirm flag.)
                 (SETQ CONFIRMFLG CONFIRM?))
           (T                                                              (* Otherwise, ask for one.)
              (SETQ CONFIRMFLG T)                                          (* SETQ CONFIRMFLG (MEMBER
                                                                           (TEDIT.GETINPUT TEXTOBJ "Ask before each replace?" "Yes"
                                                                           (CHARCODE (EOL SPACE ESCAPE LF TAB))) YESLIST))
                                                                           (* jtm%: change default to "Yes")
              ))
        (TEDIT.PROMPTPRINT TEXTOBJ "Substituting..." T)
        (SETQ DICT (DictTool.CreateConjugationMap DICTNAME SEARCHSTRING REPLACEMENT))
        (SETQ SEL (fetch SEL of TEXTOBJ))                                  (* STARTCHAR# and ENDCHAR# are the bound of the search)
        (\SHOWSEL SEL NIL NIL)
        (TEDIT.RESET.EXTEND.PENDING.DELETE SEL)                            (* Turn off any blue pending delete)
        (SETQ STARTCHAR# (fetch CH# of SEL))
        [SETQ ENDCHAR# (IPLUS STARTCHAR# (SUB1 (fetch DCH of SEL]
        (while (AND (SETQ RANGE (LingFns.FindWord TEXTSTREAM SEARCHSTRING STARTCHAR# DICT)
                                )                                          (* jtm%: use FindWord for TEDIT.FIND)
                    (NOT ABORTFLG))
           do (SETQ REPLACESTRING (CADDR RANGE))                           (* jtm%: add the suffix.)
              [PROG (PENDING.SEL CHOICE)
                    (COND
                       [CONFIRMFLG (SETQ PENDING.SEL (TEDIT.SETSEL TEXTSTREAM (CAR RANGE)
                                                          (IDIFFERENCE (CADR RANGE)
                                                                 (SUB1 (CAR RANGE)))
                                                          'RIGHT))
                              (TEDIT.SHOWSEL TEXTSTREAM T PENDING.SEL)
                              (TEDIT.NORMALIZECARET TEXTOBJ SEL)
                              [SETQ CHOICE (COND
                                              [(LISTP REPLACESTRING)
                                               (SETQ REPLACESTRING
                                                 (MENU (create MENU
                                                              ITEMS _ (CONS "*QUIT*" REPLACESTRING)
                                                              CENTERFLG _ T
                                                              CHANGEOFFSETFLG _ T
                                                              TITLE _ "substitutions"]
                                              (T (TEDIT.GETINPUT TEXTOBJ (CONCAT "Substitute '" REPLACESTRING
                                                                       "'? ['q' quits]")
                                                        "Yes"
                                                        (CHARCODE (EOL SPACE ESCAPE LF TAB]
                              (COND
                                 ((MEMBER CHOICE '("*QUIT*" "Q" "q"))
                                  (SETQ ABORTFLG T)
                                  (GO L1))
                                 ((MEMBER CHOICE '(NIL "n" "N" "no" "NO"))
                                                                           (* turn off selection)
                                  (TEDIT.SHOWSEL TEXTSTREAM NIL PENDING.SEL)
                                  (RPLACA RANGE (IDIFFERENCE (CADR RANGE)
                                                       (NCHARS REPLACESTRING)))
                                  (GO L1))
                                 (T                                        (* OK to replace)
                                    (TEDIT.DELETE TEXTSTREAM PENDING.SEL)
                                                                           (* make the replacement)
                                    (COND
                                       ((NOT (EQUAL REPLACESTRING ""))
                                                                           (* If the replacestring is nothing, why bother to add nothing)
                                        (TEDIT.INSERT TEXTSTREAM REPLACESTRING (CAR RANGE))
                                        [SETQ ENDCHAR# (IPLUS ENDCHAR# (IDIFFERENCE (NCHARS REPLACESTRING)
                                                                            (IDIFFERENCE (CADR RANGE)
                                                                                   (SUB1 (CAR RANGE]
                                        (add REPLACEDFLG 1]
                       (T                                                  (* No confirmation required. Do the substitutions without
                                                                           showing intermediate work)
                              (SETQ PC# (\DELETECH (CAR RANGE)
                                          (CADR RANGE)
                                          (ADD1 (IDIFFERENCE (CADR RANGE)
                                                       (CAR RANGE)))
                                          TEXTOBJ))
                              (\FIXDLINES (fetch LINES of TEXTOBJ)
                                     SEL
                                     (CAR RANGE)
                                     (CADR RANGE)
                                     TEXTOBJ)
                              [COND
                                 ((NOT (EQUAL REPLACESTRING ""))           (* If the replacestring is nothing, why bother to add nothing)
                                  (COND
                                     [CRSEEN (for ACHAR instring REPLACESTRING as NCH# from (CAR RANGE)
                                                 by 1 do (SELCHARQ ACHAR
                                                             (CR (\INSERTCR ACHAR NCH# TEXTOBJ))
                                                             (\INSERTCH ACHAR NCH# TEXTOBJ]
```

```
                                      (T (\INSERTCH REPLACESTRING (CAR RANGE)
                                             TEXTOBJ PC#)))
                            (SETQ ENDCHAR# (IPLUS ENDCHAR# (IDIFFERENCE (NCHARS REPLACESTRING)
                                                             (IDIFFERENCE (CADR RANGE)
                                                                 (SUB1 (CAR RANGE]
                   (add REPLACEDFLG 1)))
             L1  (SETQ STARTCHAR# (IPLUS (CAR RANGE)
                                     (NCHARS REPLACESTRING]
                                                                  (* start looking where you left off)
           )
        (COND
           ((ZEROP REPLACEDFLG)
            (TEDIT.PROMPTPRINT TEXTOBJ "No replacements made." T))
           ((EQUAL REPLACEDFLG 1)
            (TEDIT.PROMPTPRINT TEXTOBJ "1 Replacement made." T))
           (T (TEDIT.PROMPTPRINT TEXTOBJ (CONCAT (MKSTRING REPLACEDFLG)
                                                 " Replacements made.")
                   T)))
        (COND
           ((AND (NOT CONFIRMFLG)
                 (NOT (ZEROP REPLACEDFLG)))        (* There WERE replacements, and they were not confirmed.)
            (replace CHLIM of SEL with ENDCHAR#)
            [replace DCH of SEL with (ADD1 (IDIFFERENCE (fetch CHLIM of SEL)
                                              (fetch CH# of SEL]
            (\TEDIT.MARK.LINES.DIRTY TEXTOBJ (fetch CH# of SEL)
                (fetch CHLIM of SEL))
            (TEDIT.UPDATE.SCREEN TEXTOBJ)
            (\FIXSEL SEL TEXTOBJ)
            (\SHOWSEL SEL NIL T)))
        (RETURN REPLACEDFLG])
```

(**DictTool.CreateConjugationMap**
```
  [LAMBDA (language word1 word2)                              (* jtm%: "24-Mar-87 09:06")

          (* * creates a conjugation dictionary that maps word1 into word2.)

    (PROG [fullconj1 fullconj2 pp1 pp2 prior (dict (SimpleDict.New 'map]
          [COND
             [word2 (SETQ fullconj1 (DICTCLIENT.CONJUGATE word1 NIL NIL language))
                    (SETQ fullconj2 (DICTCLIENT.CONJUGATE word2 NIL NIL language))
                    (SETQ pp1 (FASSOC 'pp fullconj1))
                    (SETQ pp2 (FASSOC 'pp fullconj2))
                    [COND
                       [(AND pp1 (NULL pp2)
                             (FASSOC 'v fullconj2))
                        (push fullconj2 (CONS 'pp (CDR (FASSOC 'pst fullconj2]
                       ((AND pp2 (NULL pp1)
                             (FASSOC 'v fullconj1))
                        (push fullconj1 (CONS 'pp (CDR (FASSOC 'pst fullconj1]
                    (for conj1 conj2 entry in fullconj1
                       do (SETQ conj2 (FASSOC (CAR conj1)
                                         fullconj2))
                          (AND conj2 (for caps oldValue newValue in '(NONE FIRST ALL)
                                        do (SETQ entry (LingFns.Capitalize (CADR conj1)
                                                        caps))
                                           (SETQ oldValue (Dict.GetEntry dict entry))
                                           (SETQ newValue (LingFns.Capitalize (CADR conj2)
                                                            caps))
                                           (SETQ newValue (COND
                                                             ((for i inside oldValue thereis (STREQUAL i newValue))
                                                              oldValue)
                                                             ((LISTP oldValue)
                                                              (CONS newValue oldValue))
                                                             (oldValue (LIST newValue oldValue))
                                                             (T newValue)))
                                           (Dict.PutEntry dict entry newValue]
             (T (for conjugation in (DICTCLIENT.CONJUGATE word1 NIL language)
                   do (for caps in '(NONE FIRST ALL) do (Dict.PutEntry dict (LingFns.Capitalize (CADR conjugation)
                                                                             caps)
                                                          T]
          (RETURN dict])
```

(**DictTool.FindWordInit**
```
  [LAMBDA NIL                                                 (* jtm%: "26-Feb-87 13:46")

          (* * add items to TEDIT's menu.)

    [for ITEM on (fetch (MENU ITEMS) of TEDIT.DEFAULT.MENU)
       do (COND
             [(EQ (CAR ITEM)
                  'Find)
              (RPLACA ITEM '(Find 'Find NIL (SUBITEMS (FindWord (FUNCTION DictTool.FindWord)
                                                           "Looks for a word independent of its inflection or
                                                           capitalization."]
             ((EQ (CAR ITEM)
```

```
                         'Substitute)
               (RPLACA ITEM '(Substitute 'Substitute NIL (SUBITEMS (SubstituteWord (FUNCTION
                                                                       DictTool.SubstituteWord)
                                                       "Substitutes one word for another,
                                                       keeping the same capitalization and
                                                       inflectional form."]
```

(* * force the menu to be recomputed.)

```
       (COND
           ((EQ (fetch MENUCOLUMNS of TEDIT.DEFAULT.MENU)
                1)                                                        (* If there is only one column, force a re-figuring of the number
                                                                         of rows)
            (replace MENUROWS of TEDIT.DEFAULT.MENU with NIL))
           ((EQ (fetch MENUROWS of TEDIT.DEFAULT.MENU)
                1)                                                        (* There's only one row, so recompute %# of columns.)
            (replace MENUCOLUMNS of TEDIT.DEFAULT.MENU with NIL)))
       (replace ITEMWIDTH of TEDIT.DEFAULT.MENU with 10000)
       (replace ITEMHEIGHT of TEDIT.DEFAULT.MENU with 10000)
       (replace IMAGE of TEDIT.DEFAULT.MENU with NIL)                     (* Force it to create a new menu image.)
       (UPDATE/MENU/IMAGE TEDIT.DEFAULT.MENU])
)

(DEFINEQ
```

(**LingFns.FindWord**
```
  [LAMBDA (STREAM WORD CH DICT)                                          (* jtm%: "24-Mar-87 09:28")

       (* * finds the next instance of WORD in the text stream, independent of how it is conjugated or capitalized.
       returns the first character index, the last character index, the suffix, and the capitalization.)

       (PROG (CHAR NODE END EXPO FIRSTCHAR LASTCHAR U-FIRSTCHAR EOFPTR dictCreated)

            (* * build the dictionary)

            [COND
                (WORD (SETQ WORD (MKSTRING WORD))
                      [COND
                          ((NULL DICT)
                           (SETQ DICT (STREAMPROP STREAM 'FINDWORDMAP))
                           (COND
                               ((EQUAL WORD (CAR DICT))
                                (SETQ DICT (CDR DICT)))
                               (T (SETQ DICT (DictTool.CreateConjugationMap NIL WORD))
                                  (STREAMPROP STREAM 'FINDWORDMAP (CONS WORD DICT]

            (* * initialize.)

                      [COND
                          ((NULL CH)
                           (SETQ CH (TEDIT.GETPOINT STREAM]
                      (SETQ CH (SUB1 CH))
                      (SETQ EOFPTR (GETEOFPTR STREAM))
                      (COND
                          ((GREATERP CH EOFPTR)
                           (RETURN))
                          (T (SETFILEPTR STREAM CH)))
                      [SETQ FIRSTCHAR (CHCON1 (L-CASE (NTHCHAR WORD 1]
                      [SETQ U-FIRSTCHAR (CHCON1 (U-CASE (NTHCHAR WORD 1]

            (* * search for a word that begins with the first letter.)

                      (while (NEQ EOFPTR (GETFILEPTR STREAM))
                         do (SETQ LASTCHAR CHAR)
                            (SETQ CHAR (BIN STREAM))
                            (COND
                                ([AND [OR (NULL LASTCHAR)
                                          (AND (NUMBERP LASTCHAR)
                                               (NOT (ALPHACHARP LASTCHAR]
                                      (NUMBERP CHAR)
                                      (SETQ NODE (FASSOC (CHARACTER CHAR)
                                                     (fetch (SimpleDict.Node subnodes) of (fetch (Dict contents)
                                                                                             of DICT]
                                 (SETQ CH (GETFILEPTR STREAM))
                                 [while NODE do (COND
                                                   ((EQP EOFPTR (GETFILEPTR STREAM))
                                                    (SETQ END EOFPTR)
                                                    (RETURN))
                                                   ([AND (SETQ CHAR (BIN STREAM))
                                                         (NUMBERP CHAR)
                                                         (ALPHACHARP CHAR)
                                                         (SETQ NODE (FASSOC (CHARACTER CHAR)
                                                                        (fetch (SimpleDict.Node subnodes)
                                                                            of NODE]
                                                              (* is this a legal character?)
                                                    )
```

```
                                                  (T (RETURN]
                              (COND
                                 ((SETQ EXPO (fetch (SimpleDict.Node value) of NODE))
                                  (RETURN]

          (* * we are done.)

          (RETURN (COND
                      ((AND EXPO CH)
                       [COND
                          ((NULL END)
                           (SETQ END (SUB1 (GETFILEPTR STREAM]
                       (LIST CH END EXPO]])
```

(**LingFns.Capitalize**
```
  [LAMBDA (word caps)                                          (* jtm%: " 6-Aug-84 12:53")

          (* * capitalizes word according to the parameter "caps")

     (COND
         ((LISTP word)
          (for w in word collect (LingFns.Capitalize w caps)))
         (T (PROG (stringP litAtom)
                  (COND
                     ((STRINGP word)
                      (SETQ word (UNPACK word))
                      (SETQ stringP T))
                     ((LITATOM word)
                      (SETQ word (UNPACK word))
                      (SETQ litAtom T)))
                  [SELECTQ caps
                      (FIRST [COND
                                ((NOT (U-CASEP (CAR word)))
                                 (RPLACA word (U-CASE (CAR word]
                             [for char on (CDR word) do (COND
                                                           ((U-CASEP (CAR char))
                                                            (RPLACA char (L-CASE (CAR char])
                      (ALL [for char on word do (COND
                                                   ((NOT (U-CASEP (CAR char)))
                                                    (RPLACA char (U-CASE (CAR char])
                      (for char on word do (COND
                                              ((U-CASEP (CAR char))
                                               (RPLACA char (L-CASE (CAR char]
                  [COND
                     [stringP (SETQ word (MKSTRING (PACK word]
                     (litAtom (SETQ word (PACK word]
                  (RETURN word])
```

(**LingFns.Capitalization**
```
  [LAMBDA (word)                                               (* jtm%: "18-Jul-84 15:19")

          (* * returns NIL, ALL or FIRST)

     (COND
         ([OR (NULL word)
              (NOT (U-CASEP (CAR word]
          NIL)
         ([OR (NULL (CDR word))
              (NOT (U-CASEP (CADR word]
          'FIRST)
         (T 'ALL])
)
```

(**DictTool.FindWordInit**)

(RPAQ? **DictTool.TimeOperation** NIL)

(RPAQ? **Dict.DefWindow** NIL)

(RPAQ? **Dict.CommandsAdded** NIL)

(RPAQ? **InvertedDict.List** NIL)

(RPAQ? **DictTool.LastSearch** NIL)

(RPAQ? **DictTool.LastWord** NIL)

(RPAQ? **TEdit.DefaultDictionary** NIL)

(RPAQ? **DictTool.MinKeywords** 2)

(RPAQ? **DictTool.MaxWords** 100)

(DECLARE%: DOEVAL@COMPILE DONTCOPY

```
(GLOBALVARS DictTool.TimeOperation Dict.DefWindow Dict.CommandsAdded InvertedDict.List DictTool.MinKeywords
        DictTool.MaxWords DictTool.LastSearch DictTool.LastWord TEdit.DefaultDictionary)
)

(DictTool.Init)

(RPAQQ PronunciationGuide
        (("q" "cat" "(kqt)")
         ("A" "pay" "(pA)")
         ("Q" "care" "(kQr)")
         ("*" "father" "(f*%"T5r)")
         ("b" "bike" "(bIk)")
         ("ch" "church" "(ch/rch)")
         ("d" "deed" "(dEd)")
         ("4" "pet" "(p4t)")
         ("E" "seed" "(sEd)")
         ("I" "fife" "(fIf)")
         ("g" "gag" "(gqg)")
         ("h" "hat" "(hqt)")
         ("hw" "which" "(hw9ch)")
         ("9" "pit" "(p9t)")
         ("I" "lie" "(lI)")
         ("7" "pier" "(p7r)")
         ("j" "judge" "(j8j)")
         ("k" "kick" "(k9k)")
         ("l" "lid" "(l9d)")
         ("l" "needle" "(nEd%"l)")
         ("m" "mum" "(m8m)")
         ("n" "no, sudden" "(nO)")
         ("ng" "thing" "(th9ng)")
         ("0" "pot" "(p0t)")
         ("O" "toe" "(tO)")
         ("—" "paw" "(p—)")
         ("oi" "noise" "(noiz)")
         ("ou" "out" "(out)")
         ("1" "book" "(b1k)")
         ("|" "boot" "(b|t)")
         ("p" "people" "(pE%"p5l)")
         ("r" "roar" "(r—r)")
         ("s" "sauce" "(s—s)")
         ("sh" "ship" "(sh9p)")
         ("t" "tight" "(tIt)")
         ("th" "thin" "(th9n)")
         ("T" "this" "(T9s)")
         ("8" "cut" "(k8t)")
         ("/" "urge" "(/rj)")
         ("v" "valve" "(vqlv)")
         ("w" "with" "(w9T, w9th)")
         ("y" "yes" "(y4s)")
         ("z" "zebra" "(zE%"br5)")
         ("zh" "vision" "(v9zh%"5n)")
         ("5" "about" "(5-bout%)")
         ("KH" "loch" "(l0KH, l0k)")
         ("N" "bon" "(b0n; French b—N).")))

(RPAQQ PronunciationMap
        ((%" %')
         (5 ɛ)
         (/ ˘ u)
         (8 ˘ u)
         (T ˛ t h)
         (%| ˙ ¯ o ¯ o)
         (1 ˘ o ˘ o)
         (%— ô)
         (O ¯ o)
         (0 ˘ o)
         (7 î)
         (I í)
         (9 ì)
         (E ¯ e)
         (4 ˘ e)
         (* ä)
         (Q â)
         (A ¯ a)
         (q ˘ a)))

(PUTPROPS DICTTOOL COPYRIGHT ("Xerox Corporation" 1986 1987 1988 1989 1991 1994))
```

FUNCTION INDEX

VARIABLE INDEX

PROPERTY INDEX