File created: 13-Oct-87 12:01:34 **{QV}<DICT>TOOLS>PROOFREADER.;34**

changes to: (FNS Proofreader.New)

previous date: 6-Feb-87 16:02:15 {QV}<DICT>TOOLS>PROOFREADER.;33

Read Table: OLD-INTERLISP-FILE

Package: INTERLISP

Format: XCCS


```
(RPAQQ PROOFREADERCOMS
       [(FILES ANALYZER SPELLINGARRAY)
        (FNS Proofreader.New Proofreader.Open Proofreader.AddEntry Proofreader.Lookup Proofreader.AllForms)
        (FNS Proofreader.CharTable Proofreader.LookupBit Proofreader.SetBit)
        (FNS Proofreader.Correct Proofreader.NextWord)
        (MACROS Proofreader.Hash1 Proofreader.Hash2 \Proofreader.TestCorruption)
        (INITVARS Proofreader Proofreader.AutoLoad Proofreader.Lisp)
        (* Proofreader.AutoLoad is a file or list of files to be loaded whenever a proofreader is opened.)
        (P (Analyzer.Establish (SETQ Proofreader (Proofreader.New (QUOTE Proofreader])

(FILESLOAD ANALYZER SPELLINGARRAY)

(DEFINEQ
```

(Proofreader.New

```
  [LAMBDA (name fileName)                                   (* jtm: "13-Oct-87 11:57")
    (PROG [(analyzer (create Morphalyzer
                             analyzerName _ name
                             openFn _ (FUNCTION Proofreader.Open)
                             lookupFn _ (FUNCTION Proofreader.Lookup)
                             addEntryFn _ (FUNCTION Proofreader.AddEntry]
          (RETURN analyzer])
```

(Proofreader.Open

```
  [LAMBDA (analyzer stream)                                 (* jtm: " 6-Feb-87 15:24")
    (COND
       ((NULL (fetch (Morphalyzer index) of analyzer))
        [replace (Morphalyzer index) of analyzer
           with (PROG [(file (Analyzer.Prop analyzer (QUOTE FileName]
                      [COND
                         ((AND (NULL SpellingArray)
                               (NULL file))
                          (ERROR "No Spelling Array for" analyzer))
                         ((NULL SpellingArray)
                          (COND
                             ((NULL stream)
                              (PROMPTPRINT "initializing Proofreader"))
                             (T (TEDIT.PROMPTPRINT stream "initializing Proofreader" T)))
                          (RESETLST
                              (PROG (LENGTH ALENGTH BLOCK STREAM (START 0)
                                            (HEADERSIZE 6))
                                    [RESETSAVE (SETQ STREAM (OPENSTREAM file (QUOTE INPUT)
                                                                        (QUOTE OLD)))
                                           (QUOTE (PROGN (CLOSEF OLDVALUE]
                                    (SETQ LENGTH (IDIFFERENCE (GETFILEINFO file (QUOTE LENGTH))
                                                              (IPLUS HEADERSIZE 2)))
                                    (for i from 1 to HEADERSIZE do (BIN STREAM))
                                                                        (* skip header)
                                    (while (ILESSP START LENGTH) do (SETQ ALENGTH (MIN 64000 (IDIFFERENCE LENGTH
                                                                                                          START)))
                                                                    (SETQ BLOCK (\ALLOCBLOCK (LRSH (IPLUS 3
                                                                                                          ALENGTH
                                                                                                          )
                                                                                                   2)))
                                                                    (\BINS STREAM BLOCK 0 ALENGTH)
                                                                    (add START ALENGTH)
                                                                    (push SpellingArray (CONS START BLOCK)))
                                    (SETQ SpellingArray (REVERSE SpellingArray)))))]
                      (RETURN (CONS SpellingArray (Proofreader.CharTable]
        (for file inside Proofreader.AutoLoad do (Analyzer.DefaultLoadWordList analyzer file])
```

(Proofreader.AddEntry

```
  [LAMBDA (analyzer lemma entry dontRecord)                 (* jtm: " 6-Feb-87 15:24")
```

(* * adds "lemma" to the SpellingArray. This procedure is just like Lookup, only it sets the bits rather than just reading them.)

```
    (PROG (char p x1 x2 x3 x4 x5 x6 x7 hash1 hash2 hash3 hash4 hash5 hash6 hash7 hashArray hashArray.CharTable
          start length)                                     (* first save the word on a property list.)
          (COND
             ((NULL dontRecord)
              (Analyzer.PushProp analyzer (QUOTE WordList)
```

```
                lemma)))
          (SETQ hashArray (fetch (Morphalyzer index) of analyzer))
          [COND
             ((NULL entry)
              (SETQ entry (Proofreader.AllForms lemma]
          [COND
             ((NULL hashArray)
              (Proofreader.Open analyzer)
              (SETQ hashArray (fetch (Morphalyzer index) of analyzer]
          (SETQ hashArray.CharTable (CDR hashArray))
          (SETQ hashArray (CAR hashArray))
          (SETQ hash1 953)
          (SETQ hash2 63869)
          (SETQ hash3 2441)
          (SETQ hash4 62265)
          (SETQ hash5 4079)
          (SETQ hash6 60585)
          (SETQ hash7 5807)
          (SETQ p 359)
          (Stream.Init lemma start length)
          (while (SETQ char (Stream.NextChar lemma length start))
             do [COND
                   ((ALPHACHARP char)
                    (SETQ char (ELT hashArray.CharTable (IDIFFERENCE char 64]
                (add p 1009)
                (SETQ x1 (LOGAND 65535 (IDIFFERENCE (LOGXOR char (Proofreader.Hash1 hash2))
                                                    p)))
                [SETQ x2 (LOGAND 65535 (IDIFFERENCE p (IDIFFERENCE (Proofreader.Hash2 hash3)
                                                                   char]
                [SETQ x3 (LOGAND 65535 (LOGXOR p (IDIFFERENCE (Proofreader.Hash1 hash4)
                                                              char]
                (SETQ x4 (LOGAND 65535 (IDIFFERENCE (IDIFFERENCE char (Proofreader.Hash2 hash5))
                                                    p)))
                (SETQ x5 (LOGAND 65535 (IDIFFERENCE (LOGXOR char (Proofreader.Hash1 hash6))
                                                    p)))
                (SETQ x6 (LOGAND 65535 (IDIFFERENCE (IDIFFERENCE (Proofreader.Hash2 hash7)
                                                                 char)
                                                    p)))
                [SETQ x7 (LOGAND 65535 (IDIFFERENCE p (IDIFFERENCE (Proofreader.Hash1 hash1)
                                                                   char]
                (SETQ hash1 x1)
                (SETQ hash2 x2)
                (SETQ hash3 x3)
                (SETQ hash4 x4)
                (SETQ hash5 x5)
                (SETQ hash6 x6)
                (SETQ hash7 x7))

          (* * set the bits.)

          (Proofreader.SetBit hash1 hash7 hashArray)
          (Proofreader.SetBit hash2 hash6 hashArray)
          (Proofreader.SetBit hash3 hash5 hashArray)
          (Proofreader.SetBit hash4 hash4 hashArray)
          (Proofreader.SetBit hash5 hash3 hashArray)
          (Proofreader.SetBit hash6 hash2 hashArray)
          (Proofreader.SetBit hash7 hash1 hashArray)
          (RETURN lemma])
```

(Proofreader.Lookup

```
  [LAMBDA (analyzer stream start length)                    (* jtm: " 6-Feb-87 15:25")
```

(* * hashes the string into the array using a probabalistic technique.
This may produce a false positive.)

```
    (PROG (char word p x1 x2 x3 x4 x5 x6 x7 hash1 hash2 hash3 hash4 hash5 hash6 hash7 hashArray
                hashArray.CharTable)
          (SETQ hashArray (fetch (Morphalyzer index) of analyzer))
          [COND
             ((NULL hashArray)
              (Proofreader.Open analyzer)
              (SETQ hashArray (fetch (Morphalyzer index) of analyzer]
          (SETQ hashArray.CharTable (CDR hashArray))
          (SETQ hashArray (CAR hashArray))
          (SETQ hash1 953)
          (SETQ hash2 63869)
          (SETQ hash3 2441)
          (SETQ hash4 62265)
          (SETQ hash5 4079)
          (SETQ hash6 60585)
          (SETQ hash7 5807)
          (SETQ p 359)
          (Stream.Init stream start length)
          (while (SETQ char (Stream.NextChar stream length start))
             do [COND
                   ((IGREATERP char 255)
```

```
                (SETQ char (IMOD char 256]
             [COND
                ((ALPHACHARP char)
                 (SETQ char (ELT hashArray.CharTable (IDIFFERENCE char 64]
             (add p 1009)
             (SETQ x1 (LOGAND 65535 (IDIFFERENCE (LOGXOR char (Proofreader.Hash1 hash2))
                                                 p)))
             [SETQ x2 (LOGAND 65535 (IDIFFERENCE p (IDIFFERENCE (Proofreader.Hash2 hash3)
                                                                char]
             [SETQ x3 (LOGAND 65535 (LOGXOR p (IDIFFERENCE (Proofreader.Hash1 hash4)
                                                           char]
             (SETQ x4 (LOGAND 65535 (IDIFFERENCE (IDIFFERENCE char (Proofreader.Hash2 hash5))
                                                 p)))
             (SETQ x5 (LOGAND 65535 (IDIFFERENCE (LOGXOR char (Proofreader.Hash1 hash6))
                                                 p)))
             (SETQ x6 (LOGAND 65535 (IDIFFERENCE (IDIFFERENCE (Proofreader.Hash2 hash7)
                                                              char)
                                                 p)))
             [SETQ x7 (LOGAND 65535 (IDIFFERENCE p (IDIFFERENCE (Proofreader.Hash1 hash1)
                                                                char]
             (SETQ hash1 x1)
             (SETQ hash2 x2)
             (SETQ hash3 x3)
             (SETQ hash4 x4)
             (SETQ hash5 x5)
             (SETQ hash6 x6)
             (SETQ hash7 x7))
          (COND
             ((AND (Proofreader.LookupBit hash1 hash7 hashArray)
                   (Proofreader.LookupBit hash2 hash6 hashArray)
                   (Proofreader.LookupBit hash3 hash5 hashArray)
                   (Proofreader.LookupBit hash4 hash4 hashArray)
                   (Proofreader.LookupBit hash5 hash3 hashArray)
                   (Proofreader.LookupBit hash6 hash2 hashArray)
                   (Proofreader.LookupBit hash7 hash1 hashArray))
              (RETURN T])
```

(Proofreader.AllForms

```
  [LAMBDA (lemma)                                                    (* jtm: " 6-Feb-87 15:25")
```

(* * ask the user for the forms to fill out this word.)

```
    (PROG (forms form newForms menuPos)
          (SETQ forms (LIST (QUOTE NOUN)
                            (QUOTE VERB)
                            (QUOTE ADJ)
                            (English.Suffix lemma "s")
                            (English.Suffix lemma "s")
                            (English.Suffix lemma "er")
                            " "
                            (English.Suffix lemma "ed")
                            (English.Suffix lemma "est")
                            " "
                            (English.Suffix lemma "ing")
                            " " " " (QUOTE *OTHER*)))
          (while [SETQ form
                  (MENU (create MENU
                                TITLE _ "parts of speech"
                                CENTERFLG _ T
                                ITEMS _ forms
                                MENUCOLUMNS _ 3
                                CHANGEOFFSETFLG _ T
                                MENUPOSITION _ (COND
                                                  (menuPos)
                                                  (T (GETMOUSESTATE)
                                                     (SETQ menuPos (CONS LASTMOUSEX LASTMOUSEY]
             do (pushnew newForms form))
          (RETURN newForms])

)

(DEFINEQ
```

(Proofreader.CharTable

```
  [LAMBDA NIL                                                        (* jtm: " 6-Feb-87 15:27")
```

(* * comment)

```
    (PROG (SpellingArray.CharTable)
          (SETQ SpellingArray.CharTable (ARRAY 58))
          (for i in (QUOTE (0 32)) do (SETA SpellingArray.CharTable (IPLUS i 1)
                                            65325)
                                      (SETA SpellingArray.CharTable (IPLUS i 2)
                                            65204)
                                      (SETA SpellingArray.CharTable (IPLUS i 3)
                                            449)
```

```
                                    (SETA SpellingArray.CharTable (IPLUS i 4)
                                          588)
                                    (SETA SpellingArray.CharTable (IPLUS i 5)
                                          7102)
                                    (SETA SpellingArray.CharTable (IPLUS i 6)
                                          64682)
                                    (SETA SpellingArray.CharTable (IPLUS i 7)
                                          64545)
                                    (SETA SpellingArray.CharTable (IPLUS i 8)
                                          64418)
                                    (SETA SpellingArray.CharTable (IPLUS i 9)
                                          1278)
                                    (SETA SpellingArray.CharTable (IPLUS i 10)
                                          1433)
                                    (SETA SpellingArray.CharTable (IPLUS i 11)
                                          63968)
                                    (SETA SpellingArray.CharTable (IPLUS i 12)
                                          63827)
                                    (SETA SpellingArray.CharTable (IPLUS i 13)
                                          1874)
                                    (SETA SpellingArray.CharTable (IPLUS i 14)
                                          2027)
                                    (SETA SpellingArray.CharTable (IPLUS i 15)
                                          2180)
                                    (SETA SpellingArray.CharTable (IPLUS i 16)
                                          63195)
                                    (SETA SpellingArray.CharTable (IPLUS i 17)
                                          63058)
                                    (SETA SpellingArray.CharTable (IPLUS i 18)
                                          62865)
                                    (SETA SpellingArray.CharTable (IPLUS i 19)
                                          2798)
                                    (SETA SpellingArray.CharTable (IPLUS i 20)
                                          2963)
                                    (SETA SpellingArray.CharTable (IPLUS i 21)
                                          62372)
                                    (SETA SpellingArray.CharTable (IPLUS i 22)
                                          62216)
                                    (SETA SpellingArray.CharTable (IPLUS i 23)
                                          62067)
                                    (SETA SpellingArray.CharTable (IPLUS i 24)
                                          3624)
                                    (SETA SpellingArray.CharTable (IPLUS i 25)
                                          3793)
                                    (SETA SpellingArray.CharTable (IPLUS i 26)
                                          3944))
          (RETURN SpellingArray.CharTable])
```

(Proofreader.LookupBit

```
  [LAMBDA (row column SpellingArray)                    (* jtm: " 6-Feb-87 15:27")

          (* * There are 4096 bits per row, but only 4093 of them are used.)

    (PROG (byte (startByte 0))
          (SETQ row (IMOD row 199))
          (SETQ column (IMOD column 4093))
          (SETQ byte (IPLUS (LLSH row 9)
                            (LRSH column 3)))
          (for block in SpellingArray do (COND
                                             ((ILESSP byte (CAR block))
                                              (SETQ byte (\GETBASEBYTE (CDR block)
                                                                       (IDIFFERENCE byte startByte)))
                                             (RETURN)))
                                         (SETQ startByte (CAR block)))
          (RETURN (BITTEST byte (MASK.1'S (IDIFFERENCE 7 (LOGAND column 7))
                                          1])
```

(Proofreader.SetBit

```
  [LAMBDA (row column SpellingArray)                    (* jtm: " 6-Feb-87 15:28")

          (* * There are 4096 bits per row, but only 4093 of them are used.)

    (PROG (address (startByte 0))
          (SETQ row (IMOD row 199))
          (SETQ column (IMOD column 4093))
          (SETQ address (IPLUS (LLSH row 9)
                               (LRSH column 3)))
          (for block byte in SpellingArray do (COND
                                                  ((ILESSP address (CAR block))
                                                   (SETQ byte (\GETBASEBYTE (CDR block)
                                                                            (IDIFFERENCE address startByte)))
                                                   (SETQ byte (BITSET byte (MASK.1'S (IDIFFERENCE 7
                                                                                                  (LOGAND column 7))
                                                                                     1)))
                                                   (\PUTBASEBYTE (CDR block)
```

```
                                                    (IDIFFERENCE address startByte)
                                                   byte)
                                              (RETURN)))
                                         (SETQ startByte (CAR block])

)

(DEFINEQ
```

(Proofreader.Correct

```
  [LAMBDA (analyzer stream start length)                    (* jtm: " 6-Feb-87 15:28")
```

(* * returns a list of possible spelling corrections for the given word.)

```
    (PROG (form word wordList caps periods)
          [COND
             ((NOT (LISTP stream))
              (SETFILEPTR stream start)
              (SETQ word (for i from 1 to length collect (READC stream]
          (SETQ caps (Analyzer.Capitalization word))
          (SETQ periods (FMEMB (QUOTE %.)
                               word))
```

(* * first try transpositions)

```
          (for tail temp on word while (CDR tail) do (SETQ temp (CAR tail))
                                                     (RPLACA tail (CADR tail))
                                                     (RPLACA (CDR tail)
                                                            temp)
                                                     (COND
                                                        ((AND (EQ caps (QUOTE FIRST))
                                                              (EQ tail word))
                                                             (* don't transpose the first letters of a capitalized word.)
                                                         NIL)
                                                        (T (\Proofreader.TestCorruption analyzer word wordList)))
                                                     (RPLACA (CDR tail)
                                                            (CAR tail))
                                                     (RPLACA tail temp))
```

(* * next try deletions)

```
          (COND
             ((CDR word)
              (\Proofreader.TestCorruption analyzer (CDR word)
                     wordList)))
          (for tail temp on word while (CDR tail) do (SETQ temp (CDR tail))
                                                     (RPLACD tail (CDDR tail))
                                                     (\Proofreader.TestCorruption analyzer word wordList)
                                                     (RPLACD tail temp))
```

(* * prepend a character.)

```
          (SETQ word (CONS (QUOTE A)
                           word))
          (SELECTQ caps
               (FIRST                                       (* don't prepend a character before a capitalized word.)
                      NIL)
               (ALL                                         (* prepend a capital letter.)
                    (for c from (CHARCODE A) to (CHARCODE Z) do (RPLACA word (CHARACTER c))
                                                                (\Proofreader.TestCorruption analyzer word
                                                                       wordList)))
               (for c from (CHARCODE a) to (CHARCODE z) do (RPLACA word (CHARACTER c))
                                                           (\Proofreader.TestCorruption analyzer word wordList)))
          (SETQ word (CDR word))
```

(* * insert characters.)

```
          (for tail on word do (RPLACD tail (CONS (QUOTE A)
                                                  (CDR tail)))
                               [COND
                                  ((EQ caps (QUOTE ALL))
                                   (for c from (CHARCODE A) to (CHARCODE Z) do (RPLACA (CDR tail)
                                                                                      (CHARACTER c))
                                                                              (\Proofreader.TestCorruption
                                                                                     analyzer word wordList)))
                                  (T (for c from (CHARCODE a) to (CHARCODE z) do (RPLACA (CDR tail)
                                                                                        (CHARACTER c))
                                                                                 (\Proofreader.TestCorruption
                                                                                  analyzer word wordList]
                               (COND
                                  (periods (RPLACA (CDR tail)
                                                   (QUOTE %.))
                                           (\Proofreader.TestCorruption analyzer word wordList)))
                               (RPLACD tail (CDDR tail)))
```

(* * replace characters)

```
        (for tail temp on word do (SETQ temp (CAR tail))
                                  [COND
                                     ((OR (EQ caps (QUOTE ALL))
                                          (AND (EQ caps (QUOTE FIRST))
                                               (EQ tail word)))
                                      (for c from (CHARCODE A) to (CHARCODE Z)
                                         do (COND
                                               ((NEQ temp (CHARACTER c))
                                                (RPLACA tail (CHARACTER c))
                                                (\Proofreader.TestCorruption analyzer word wordList]
                                  [COND
                                     ((OR (EQ caps NIL)
                                          (NOT (ALPHACHARP (CHCON1 temp)))
                                          (AND (EQ caps (QUOTE FIRST))
                                               (NEQ tail word)))
                                      (for c from (CHARCODE a) to (CHARCODE z)
                                         do (COND
                                               ((NEQ temp (CHARACTER c))
                                                (RPLACA tail (CHARACTER c))
                                                (\Proofreader.TestCorruption analyzer word wordList]
                                  (COND
                                     (periods (RPLACA tail (QUOTE %.))
                                              (\Proofreader.TestCorruption analyzer word wordList)))
                                  (RPLACA tail temp))
        (SETQ wordList (SORT wordList))
        [for i on wordList do (while (STREQUAL (CAR i)
                                               (CADR i))
                                  do (RPLACD i (CDDR i]
        (RETURN wordList])
```

(Proofreader.NextWord

```
  [LAMBDA (analyzer stream startPtr searchLength NWFn)          (* jtm: " 6-Feb-87 15:29")
```

(* * Scans the stream looking for a word, i.e. a sequence of alphabetic charqacters.
If the file ptr is already in the middle of such a sequence, it backs up to the beginning of that sequence.
The function applies NWFn to (stream start stop) for each such word.)

```
    (SETFILEPTR stream (OR startPtr (SETQ startPtr 0)))
    (bind char end endPtr word length start value quote period number (filePtr _ (GETFILEPTR stream))
          (EOFPtr _ (GETEOFPTR stream)) first (SETQ endPtr (COND
                                                              (searchLength (IMIN EOFPtr (IPLUS startPtr
                                                                                                searchLength)))
                                                              (T EOFPtr)))
       do (SETQ char (AND (ILESSP (GETFILEPTR stream)
                                  endPtr)
                          (BIN stream)))
          (COND
             [(AND char (AND (NUMBERP char)
                             (ILESSP char 128)
                             (Analyzer.AlphaCharP char)))
              [OR start (SETQ start (SUB1 (GETFILEPTR stream]
              (COND
                 (number
```

(* we have a number followed by some characters. (e.g. 7th, 21st, etc.) Take in the last digit of the number.)

```
                        (add start -1)
                        (SETQ number NIL)))
              (COND
                 (quote (COND
                           ((EQ quote T)                        (* don't make a list until you need to.)
                            (SETQ quote NIL)))
                        (push quote char)))
              (COND
                 (period (COND
                            ((EQ period T)
                             (SETQ period NIL)))
                         (push period char]
             ((AND start char (EQUAL char (CHARCODE ')))        (* if the quote is in the middle of a word, leave it in.)
              (SETQ quote T))
             ((AND start char (EQUAL char (CHARCODE %.)))       (* look for e.g., i.e.)
              (OR period (SETQ period T)))
             (start (SETQ end (GETFILEPTR stream))
                    (SETQ length (IDIFFERENCE end start))
                    (AND char (add length -1))                  (* back up to the last legal char.)
                    (COND
                       ((EQ quote T)                            (* delete final quotes)
                        (add length -1))
                       ([OR (EQUAL quote (QUOTE (115)))
                            (EQUAL quote (QUOTE (83]            (* delete 's)
                        (add length -2)))
                    (SETQ quote NIL)
                    (COND
                       ((EQ period T)                           (* delete final periods)
                        (add length -1)))
                    (SETQ period NIL)
```

```
                    (COND
                       ((AND (EQ length 1)
                             (EQ char (CHARCODE %))))            (* letters used for outlines.)
                        (add length 1)))
                    [COND
                       [NWFn (SETQ value (APPLY* NWFn analyzer stream start length))
                             (COND
                                ((EQ value T)
                                 (RETURN (CONS start length)))
                                (value (RETURN value]
                       (T (RETURN (CONS start length]
                    (SETFILEPTR stream end)
                    (SETQ start NIL))
                ((AND char (NUMBERP char)
                      (IGEQ char 48)
                      (ILEQ char 57))                                (* a number)
                 (SETQ number char))
                (T (SETQ number NIL)))
          (OR char (RETURN])

)

(DECLARE: EVAL@COMPILE

(PUTPROPS Proofreader.Hash1 MACRO ((X)
                                   (IPLUS (LLSH (LOGAND X 2047)
                                                5)
                                          (LRSH X 11))))

(PUTPROPS Proofreader.Hash2 MACRO ((X)
                                   (IPLUS (LLSH (LOGAND X 8191)
                                                3)
                                          (LRSH X 13))))

(PUTPROPS \Proofreader.TestCorruption MACRO [(analyzer word wordList)
                                             (COND
                                                ((Proofreader.Lookup analyzer word NIL NIL)
                                                 (push wordList (CONCATLIST word])
)

(RPAQ? Proofreader NIL)

(RPAQ? Proofreader.AutoLoad NIL)

(RPAQ? Proofreader.Lisp NIL)

          (* * Proofreader.AutoLoad is a file or list of files to be loaded whenever a proofreader is opened.)

[Analyzer.Establish (SETQ Proofreader (Proofreader.New (QUOTE Proofreader]

(PUTPROPS PROOFREADER COPYRIGHT ("Xerox Corporation" 1985 1986 1987))
```

FUNCTION INDEX

VARIABLE INDEX

MACRO INDEX