

File created: 15-Aug-90 13:02:12 {PELE:MV:ENVO}<LISPCORE>LIBRARY>MSPARSE.;5

changes to: (VARS MSPARSECOMS)  
(FNS MSKKNOWN MSDOES MSSETSETTYPE MSJOINSET MSPARSE MSFIXUPTYPES MSVERBTYPE MSSETREP MSSEEKPHRASE  
MSCOMMAND MSSUBJTYPE MSOBJTYPE MSSIMPLESET MSMATCHRATING FIXVERBSETTYPE)

previous date: 12-Jun-90 18:23:48 {PELE:MV:ENVO}<LISPCORE>LIBRARY>MSPARSE.;4

Read Table: INTERLISP

Package: INTERLISP

Format: XCCS

::  
:: Copyright (c) 1984, 1986, 1987, 1988, 1990 by Venue & Xerox Corporation. All rights reserved.

### (RPAQQ MSPARSECOMS

```
[ (FNS MSKKNOWN MSDOES MSSETSETTYPE MS-SMASH-PACKAGE MSFAIL MSJOINSET MSBLOCK MSPATHOPTIONS MSPARSE
MSFIXUPTYPES MSVERBTYPE MSSETREP MSSEEKPHRASE MSTYPEMATCHRATING MSTRYSPELL MSTRYPARSE MSCOMMAND
MSSUBJTYPE MSOBJTYPE MSVERB MSSPLST MSWORD? MSSEEK-WORD MSSEEKTYPE MSRESPELL MSJOIN MSSIMPLESET
MSMATCHRATING MSVERBED MSCANT MSJOINDETS MSNEGATESET MSNEXTWORD MSPRED CHECKSYNONYM MSSETWORDTYPE
MSEAT MSTRYSPELLTYPE MSSETPHRASE MSQUOTED FIXVERBSETTYPE SETSYNONYM MSSETUP)
(DECLARE%: EVAL@COMPILE DONTCOPY (RECORDS * PARSERECORDS)
(MACROS GETWORDTYPE))
(VARS CONJUNCTABLE (MSSPLST))
(FILEVARS INITWORDLIST)
(P (APPLY 'MSSETUP INITWORDLIST))
(FUNCTIONS MSSEEK SEEKWORD)
(COMMANDS ".")
(BLOCKS (MSPARSEBLOCK MSKKNOWN MSDOES MSSETSETTYPE MS-SMASH-PACKAGE MSFAIL MSJOINSET MSBLOCK
MSPATHOPTIONS MSPARSE MSFIXUPTYPES MSVERBTYPE MSSETREP MSSEEKPHRASE MSTYPEMATCHRATING
MSTRYSPELL MSTRYPARSE MSCOMMAND MSSUBJTYPE MSOBJTYPE MSVERB MSSPLST MSWORD? MSSEEKTYPE
MSRESPELL MSJOIN MSSIMPLESET MSMATCHRATING MSVERBED MSCANT MSJOINDETS MSNEGATESET
MSNEXTWORD MSPRED CHECKSYNONYM MSEAT MSTRYSPELLTYPE MSTRYSPELLTYPE MSSETPHRASE MSQUOTED FIXVERBSETTYPE
(ENTRIES MSPARSE MSJOINSET MSTRYSPELL SETSYNONYM MSSETUP)
(REFNS MSTRYPARSE)
(SPECVARS SPELLING)
(LOCALFREEVARS CONJUNCTIONS LASTADVERB LASTPARSED NEXTWORD PARSED S SEEKING SENTENCE
RESETS)
(BLKAPPLYFNS MSDOES MSVERBED MSBLOCK MSPATHOPTIONS MSSETREP MSCOMMAND MSVERB MSPRED
MSSETPHRASE)
(LOCALFREEVARS SETTYPE SETDET)
(NOLINKFNS . T)
(GLOBALVARS MSWORDS MSSPLST CONJUNCTABLE)
MSSETUP MSSETWORDTYPE SETSYNONYM MSSEEK-WORD)
(NIL (LOCALVARS . T)
(GLOBALVARS MSWORDS)))
(PROP FILETYPE MSPARSE)
(DECLARE%: DONTEVAL@LOAD DOEVAL@COMPILE DONTCOPY COMPILERVARS (ADDVARS (NLAMA)
(NLAML)
(LAMA])
```

(DEFINEQ

### (MSKKNOWN

```
[LAMBDA (SET) ; Edited 15-Aug-90 12:53 by jds
(* lmm%: 16-MAR-76 15 34)
( (replace (MSSETPHRASE KNOWN) of SET with T)
' (MSSETSETTYPE SET 'FNS) ; don't set the type, since now things other than FNS can call
(SELECTQ (fetch (SENTENCE ID) of (fetch (MSSETPHRASE REP) of SET))
(NOT (MSKKNOWN (fetch (NOT NEGATED) of (fetch (MSSETPHRASE REP) of SET))))
((AND OR ANDNOT)
(MSKKNOWN (fetch (CSET SET1) of SET))
(MSKKNOWN (fetch (CSET SET2) of SET)))
NIL])
```

### (MSDOES

```
[LAMBDA NIL ; Edited 15-Aug-90 12:48 by jds
(AND (SEEKWORD DOES DO)
(create MSSETPHRASE
REP _ (create THAT
OTHERSET _ (OR (MSSEEK MSSETPHRASE)
(MSCANT))
MSVERB _ (PROGN (COND
((NOT (MSSEEK MSVERB S))
(MSCANT)))
(replace (MSVERB TENSE) of LASTPARSED with 'ED)
LASTPARSED])
```

### (MSSETSETTYPE

```
[LAMBDA (SET TYPE) ; Edited 15-Aug-90 12:53 by jds
[AND SET TYPE (PROG (REP)
;; We no longer care if the default type is matched or not.
```

:: ((NOT (EQUAL (OR (|fetch| (MSSETPHRASE TYPE) |of| SET) (|fetch| (MSSETPHRASE DEFAULTTYPE) |of| SET)) TYPE))  
:: (|printout| T "Expecting " TYPE " but supplied " (OR (|fetch| (MSSETPHRASE TYPE) |of| SET) (|fetch| (MSSETPHRASE  
:: DEFAULTTYPE) |of| SET)) ." T))

```
(COND
  ((AND (NULL (fetch (MSSETPHRASE TYPE) of SET))
        (NULL (fetch (MSSETPHRASE DEFAULTTYPE) of SET))))
  (replace (MSSETPHRASE DEFAULTTYPE) of SET with TYPE)
  (SELECTQ (fetch (SENTENCE ID) of (SETQ REP (fetch (MSSETPHRASE REP) of SET)))
    (THAT (FIXVERBSETTYPE SET))
    (NOT (MSSETSETTYPE (fetch (NOT NEGATED) of (fetch (MSSETPHRASE REP)
                                                         of SET))
                       TYPE))
    ((AND OR ANDNOT)
     (MSSETSETTYPE (fetch (CSET SET1) of REP)
                   TYPE)
     (MSSETSETTYPE (fetch (CSET SET2) of REP)
                   TYPE))
    NIL)
  ;; JRB - Hack to make filenames in non-IL packages work right...
  (if (AND (EQ TYPE 'FILES)
           (EQ (fetch (SENTENCE ID) of REP)
              'QUOTE))
      then (SETQ SET (MS-SMASH-PACKAGE SET (CL:FIND-PACKAGE "INTERLISP")
```

SET])

**(MS-SMASH-PACKAGE**

; Edited 6-Jun-88 20:53 by jrb:

```
[LAMBDA (THING PACKAGE)
  (COND
    ((NULL THING)
     NIL)
    [(CL:CONSP THING)
     (LET ((CARPART (MS-SMASH-PACKAGE (CAR THING)
                                       PACKAGE))
           (CDRPART (MS-SMASH-PACKAGE (CDR THING)
                                       PACKAGE)))
       (if (AND (EQ (CAR THING)
                   CARPART)
              (EQ (CDR THING)
                  CDRPART))
           then THING
           else (CONS CARPART CDRPART))]
    ((CL:SYMBOLP THING)
     (if (EQ (CL:SYMBOL-PACKAGE THING)
            PACKAGE)
         then THING
         else (CL:INTERN (CL:STRING-UPCASE (CL:SYMBOL-NAME THING))
                          PACKAGE)))
    (T THING))
```

**(MSFAIL**

; Edited 25-Nov-86 21:08 by lmm

```
[LAMBDA (FLG)
  (PROG (POS)
    (OR FLG (GO ERR))
    SEARCH
    (OR (SETQ POS (STKPOS (FUNCTION MSTRYPARSE)
                          1
                          (STKNTH -1 POS POS)
                          POS))
        (GO ERR))
    (COND
      ((EQ (STKARG 1 POS)
           'MSCOMMAND)
       (RETFROM POS NIL T)))
    (GO SEARCH)
    ERR (printout T "Sorry, couldn't parse that!" T)
    (ERROR!))
```

**(MSJOINSET**

; Edited 15-Aug-90 12:51 by jds

```
[LAMBDA (C SET1 SET2)
  (PROG (TYPE)
    (COND
      ((AND (EQ C 'AND)
            (EQ (fetch (SENTENCE ID) of (fetch (MSSETPHRASE REP) of SET1))
               'QUOTE)
            (LITATOM (fetch 'QUOTED of (fetch (MSSETPHRASE REP) of SET1))
                     (EQ (fetch (SENTENCE ID) of (fetch (MSSETPHRASE REP) of SET2))
                         'QUOTE)
                     (LITATOM (fetch 'QUOTED of (fetch (MSSETPHRASE REP) of SET2))
                              (printout T (fetch 'QUOTED of (fetch (MSSETPHRASE REP) of SET2))
                                           % C % (fetch 'QUOTED of (fetch (MSSETPHRASE REP) of SET1))
                                           " -> "
                              (fetch 'QUOTED of (fetch (MSSETPHRASE REP) of SET2))
```

```

%
(SETQ C OR)
%
(fetch 'QUOTED of (fetch (MSSETPHRASE REP) of SET1)
".")
(replace (MSSETPHRASE DET) of SET1 with (MSJOINDETS C (fetch (MSSETPHRASE DET) of SET1)
(fetch (MSSETPHRASE DET) of SET2)))
(replace (MSSETPHRASE TYPE) of SET1 with (MSJOIN C (fetch (MSSETPHRASE TYPE) of SET1)
(fetch (MSSETPHRASE TYPE) of SET2)))
[MSSETSETTYPE SET1 (SETQ TYPE (MSJOIN C (fetch (MSSETPHRASE DEFAULTTYPE) of SET1)
(fetch (MSSETPHRASE DEFAULTTYPE) of SET2]
(MSSETSETTYPE SET2 TYPE)
(COND
((fetch (MSSETPHRASE KNOWN) of SET2)
(MSKNOWN SET1))
((fetch (MSSETPHRASE KNOWN) of SET1)
(MSKNOWN SET2)))
(replace (MSSETPHRASE REP) of SET1 with (COND
((AND (EQ C 'AND)
(NULL (fetch (MSSETPHRASE REP) of SET1)))
(fetch (MSSETPHRASE REP) of SET2))
((AND (EQ C 'AND)
(NULL (fetch (MSSETPHRASE REP) of SET2)))
(fetch (MSSETPHRASE REP) of SET1))
(T (create CSET
ID _ C
SET1 _ (create MSSETPHRASE using SET1)
SET2 _ SET2]
SET1))

```

**(MSBLOCK**

; Edited 25-Nov-86 21:08 by lmm

```

[LAMBDA NIL
(SEEKWORD ANY THE A AN)
(PROG (TYPES FNS FILES)
(SETQ TYPES (OR (MSSEEKTYPE 'BLOCKS)
(RETURN)))
[COND
((SEEKWORD OF)
(SETQ FNS (OR (MSSEEK MSSETPHRASE FNS)
(RETURN]
[COND
((SEEKWORD ON)
(SETQ FILES (OR (MSSEEK MSSETPHRASE FILES)
(RETURN)))
(OR FNS (AND (SEEKWORD OF)
(SETQ FNS (OR (MSSEEK MSSETPHRASE FNS)
(RETURN]
(OR FNS FILES (RETURN))
(RETURN (create BLOCKS
TYPES _ TYPES
FNS _ FNS
FILES _ FILES])

```

**(MSPATHOPTIONS**

; Edited 25-Nov-86 21:08 by lmm

```

[LAMBDA (FLG)
(while (MSSEEKTYPE 'PATH) collect (CONS LASTPARSED (OR (SELECTQ LASTPARSED
(OUTPUT LINELENGTH)
(COND
(FLG (printout T LASTPARSED "not
meaningful except in a SHOW
PATHS command!" T)))
(MSEAT))
(FROM (MSSEEK MSSETPHRASE FNS T))
(MSSEEK MSSETPHRASE FNS))
(MSCANT])

```

**(MSPARSE**

; Edited 15-Aug-90 12:49 by jds

```

[LAMBDA (SENTENCE)
(PROG (NEXTWORD LASTADVERB CONJUNCTIONS VAL RESETS LASTPARSED FLG SEEKING SPELLING RESPELL POSS PARSED
(S SENTENCE)
OUTPUTFILE)
PARSE
(OR (SETQ VAL (MSSEEK MSCOMMAND))
(GO MSFAIL))
[COND
((SEEKWORD OUTPUT)
(SETQ OUTPUTFILE (MSEAT]
(AND (MSNEXTWORD)
(GO MSFAIL))
(COND
(LASTADVERB (GO MSFAIL)))
(MSFIXUPTYPES PARSED)
[COND
(CONJUNCTIONS (for X in (DREVERSE CONJUNCTIONS)

```

```

do (SETQ POSS (for Y in (fetch (CONJUNCTION POSSIBLES) of X)
  when [AND (EQ (fetch (PARSED ENDING) of Y)
    (fetch (CONJUNCTION START) of X))
    (EQ (CAR (fetch (PARSED SOUGHT) of Y))
      (CAR (fetch (PARSED SOUGHT)
        of (fetch (CONJUNCTION PARSED) of X)
          collect Y))
      ]
  [COND
    ( (NULL POSS)
      (MSFAIL))
    ( (CDR POSS)
      (SETQ POSS (SORT (for Y in POSS
        collect (CONS (MSMATCHRATING (fetch (CONJUNCTION PARSED)
          of X)
            Y
          (fetch (CONJUNCTION C) of X))
        T))
      (COND
        ((EQ (CAAR POSS)
          (CAADR POSS))
          (printout T "ambiguous conjunction, guessing... "))
        (SETQ POSS (CDAR POSS)))
      (T (SETQ POSS (CAR POSS)
        (SELECTQ (CAR (fetch (PARSED SOUGHT) of POSS))
          (MSSETPHRASE MSPRED MSDOES)
            (MSJOINSET (fetch (CONJUNCTION C) of X)
              (fetch (PARSED ITEM) of POSS)
              (fetch (PARSED ITEM) of (fetch (CONJUNCTION PARSED) of X))))
          (MSVERB [replace (MSVERB VPART) of (fetch (PARSED ITEM) of POSS)
            with (create CVERB
              C _ (fetch (CONJUNCTION C) of X)
              VB1 _ (fetch (MSVERB VPART)
                of (fetch (PARSED ITEM) of POSS))
              VB2 _ (fetch (MSVERB VPART)
                of (fetch (PARSED ITEM)
                  of (fetch (CONJUNCTION PARSED)
                    of X))
                ]
            (SHOULDNT]
      (COND
        (SPELLING
          (* I.e. succeeded after spelling correction)
          (LISPXPIN2 SPELLING T T)
          (LISPXPIN1 "->" T)
          (COND
            ((CAR RESPELL)
              (LISPXPIN2 (CAR RESPELL)
                T T)
              (LISPXSPACES 1 T)))
          (LISPXPIN (CADR RESPELL)
            T T)))
      (RETURN (COND
        (OUTPUTFILE (CONS 'OUTPUT (CONS OUTPUTFILE VAL)))
        (T VAL)))
      MSFAIL
      (COND
        (NOSPPELLFLG (MSFAIL))
        (NULL SPELLING)
        (SETQ SPELLING (LIST (LIST SENTENCE)))
        (GO REPARSE))
        (LISTP SPELLING)
        [for X in SPELLING when (CDR X) do (COND
          ((SETQ RESPELL (MSRESPELL (SETQ NEXTWORD (CAAR X)
            (CAR X)
            (CDR X)))
          (SETQ SPELLING NEXTWORD)
          (GO REPARSE] (* That didn't work, try to respell things as adverbs)
        [for X in SPELLING when (CDR X) do (COND
          ((SETQ RESPELL (MSRESPELL (SETQ SPELLING (CAAR X)
            (CAR X)
            (MSSPLST))))
          (GO REPARSE] (* If can't, fall through)
        (MSFAIL))
      (T (COND
        ((CAR RESPELL)
          (FRPLNODE RESPELL SPELLING (CDDR RESPELL)))
        (T (FRPLACA (CDR RESPELL)
          SPELLING)))
        (* UNDO the spelling correcting since it was wrong)
        (MSFAIL)))
      REPARSE
      [SETQ NEXTWORD (SETQ LASTADVERB (SETQ PARSED (SETQ CONJUNCTIONS (SETQ VAL (SETQ RESETS (SETQ
        LASTPARSED
        (SETQ SEEKING]
      (SETQ S SENTENCE)
      (GO PARSE])

```

(MSFIXUPTYPES

```

[LAMBDA (PRSD FLG) ; Edited 15-Aug-90 12:53 by jds
  (for X in (REVERSE PRSD) do (SELECTQ (CAR (fetch (PARSED SOUGHT) of X))
    (MSCOMMAND (SELECTQ (fetch (SENTENCE ID) of (fetch (PARSED ITEM) of X))
      ((? EDIT SHOW)
        [MSSETSETTYPE (fetch (SENTENCE SUBJECT)
          of (fetch (PARSED ITEM) of X))
          (OR (fetch (MSSETPHRASE TYPE)
            of (fetch (SENTENCE MSPRED)
              of (fetch (PARSED ITEM) of X)))
            (fetch (MSSETPHRASE DEFAULTTYPE)
              of (fetch (SENTENCE MSPRED)
                of (fetch (PARSED ITEM) of X]
          [MSSETSETTYPE (fetch (SENTENCE MSPRED)
            of (fetch (PARSED ITEM) of X))
            (OR (fetch (MSSETPHRASE TYPE)
              of (fetch (SENTENCE SUBJECT)
                of (fetch (PARSED ITEM) of X)))
              (fetch (MSSETPHRASE DEFAULTTYPE)
                of (fetch (SENTENCE SUBJECT)
                  of (fetch (PARSED ITEM) of X]
          NIL))
        ((MSPRED MSSETPHRASE MSDOES)
          [COND
            ((type? THAT (fetch (MSSETPHRASE REP) of (fetch (PARSED ITEM)
              of X)))
              (FIXVERBSETTYPE (fetch (PARSED ITEM) of X)))
            ((type? NOT (fetch (MSSETPHRASE REP) of (fetch (PARSED ITEM)
              of X)))
              (PROG [(NS (fetch (NOT NEGATED) of (fetch (MSSETPHRASE REP)
                of (fetch (PARSED ITEM)
                  of X]
                (COND
                  ((type? THAT (fetch (MSSETPHRASE REP) of NS))
                    (FIXVERBSETTYPE NS)
                    (MSSETSETTYPE (fetch (PARSED ITEM) of X)
                      (OR (fetch (MSSETPHRASE TYPE) of NS)
                        (fetch (MSSETPHRASE DEFAULTTYPE) of NS)))
                    (AND (fetch (MSSETPHRASE KNOWN) of NS)
                      (MSKNOWN (fetch (PARSED ITEM) of X]
                  NIL))
          NIL])

```

**(MSVERBTYPE**

```

[LAMBDA (MSVERB SUBJECT OBJECT) ; Edited 15-Aug-90 12:49 by jds
  (COND
    ((type? CVERB (fetch (MSVERB VPART) of MSVERB))
      (MSVERBTYPE (fetch (CVERB VB1) of (fetch (MSVERB VPART) of MSVERB))
        SUBJECT OBJECT)
      (MSVERBTYPE (fetch (CVERB VB2) of (fetch (MSVERB VPART) of MSVERB))
        SUBJECT OBJECT))
    (T (PROG (TYPE)
      [COND
        ((EQ (fetch (MSVERB TENSE) of MSVERB)
          'ED)
          (SETQ OBJECT (PROG1 SUBJECT (SETQ SUBJECT OBJECT]
          (SETQ TYPE (OR (fetch (MSSETPHRASE TYPE) of OBJECT)
            (fetch (MSSETPHRASE DEFAULTTYPE) of OBJECT)))
          (SELECTQ (fetch (MSVERB ROOT) of MSVERB)
            (CONTAIN (MSSETSETTYPE SUBJECT 'FILES))
            ((USE SET REFERENCE SMASH TEST)
              (SELECTQ (fetch (MSVERB MODIFIER) of MSVERB)
                (NIL (AND [EQMEMB (fetch (MSVERB ROOT) of MSVERB)
                  (CDR (GETWORDTYPE TYPE 'AS]
                  (replace (MSVERB MODIFIER) of MSVERB with TYPE)))
                ((FREELY LOCALLY)
                  (MSSETSETTYPE OBJECT 'VARS))
                  (MSSETSETTYPE OBJECT (fetch (MSVERB MODIFIER) of MSVERB)))
                (MSKNOWN SUBJECT))
              (CALL (MSKNOWN SUBJECT)
                (MSSETSETTYPE OBJECT 'FNS))
              (CREATE (MSKNOWN SUBJECT)
                (MSSETSETTYPE OBJECT 'RECORDS))
              ((FETCH REPLACE)
                (MSKNOWN SUBJECT)
                (MSSETSETTYPE OBJECT 'FIELDS))
              (PROGN (MSKNOWN SUBJECT)
                (MSSETSETTYPE OBJECT 'VARS])

```

**(MSSETREP**

```

[LAMBDA (LISTOKFLG) ; Edited 15-Aug-90 12:48 by jds
  (PROG [(NEG (SEEKWORD NOT))
    (VAL (COND
      (NULL S)
      NIL)
    [(SEEKWORD IN)
      (OR (MSSEEK MSBLOCK)

```

```

      (create IN
        EXPRESSION _ (MSEAT]
    [(SEEKWORD %' QUOTE)
     (MSQUOTED))
    [(SEEKWORD @)
     (create APPLY
      PRED _ (PROG ((X (MSEAT)))
        (RETURN (COND
          ((NLISTP X)
           X)
          ((EQ (CAR X)
              'LAMBDA)
           X)
          (T (LIST 'LAMBDA ' (X)
                  X]
        ]
      )
    [(SEEKWORD ONPATH)
     (create PATHS
      MSPATHOPTIONS _ (OR (MSSEEK MSPATHOPTIONS T)
                          (MSCANT]
    [(MSSEEK MSVERB ING)
     (create THAT
      MSVERB _ LASTPARSED
      OTHERSET _ (OR (MSSEEK MSSETPHRASE)
                    (MSCANT]
    ((MSSEEK MSVERBED))
    [(AND (EQ SETTYPE 'FIELDS)
     (SEEKWORD OF))
     (create FIELDS
      RECS _ (OR (MSSEEK MSSETPHRASE RECORDS)
                (MSCANT]
    [(SEEKWORD LIKE)
     (create APPLY
      PRED _ (PROG [(PAT (EDITFPAT (MSEAT]
                        (SETQ EDITQUIETFLG T)
                        (RETURN (LIST 'LAMBDA ' (X)
                                      (LIST 'EDIT4E (KWOTE PAT)
                                      'X]
      )
    ((SEEKWORD THOSE)
     (create IN
      EXPRESSION _ 'MSTHOSE))
    ((SEEKWORD KNOWN)
     (create THAT
      MSVERB _ (create MSVERB
                  ROOT _ 'IS
                  MODIFIER _ 'KNOWN)
      OTHERSET _ (create MSSETPHRASE)))
    ((AND LISTOKFLG (NOTANY S (FUNCTION MSWORD?)))
     (PROG1 (create QUOTE
                QUOTED _ S)
            (SETQ S)))
    ((MSSEEK MSBLOCK))
    [(OR (SEEKWORD THAT)
        (AND (OR SETDET SETTYPE)
             (SEEKWORD WHO WHICH)))
     (fetch (MSSETPHRASE REP) of (OR (MSSEEK MSPRED)
                                     (MSCANT]
    [(COND
     ((EQ SETDET 'THE))
     ((AND (NULL SETDET)
          (OR (NULL SETTYPE)
              (LISTP SETTYPE))))
     (COND
      ((MSWORD? (MSNEXTWORD))
       (AND (LISTP SPELLING)
            (MSTRYSPELL (CONS (CAR S)
                              (KWOTE (CAR S)))
                       S))
      (NIL)
      (T T]
     (MSQUOTED]
    (RETURN (COND
      (NEG (create NOT
              NEGATED _ (create MSSETPHRASE
                              REP _ VAL)))
      (T VAL])

```

**(MSSEEKPHRASE**

```

[LAMBDA (SOUGHT FORCONJUNCTION)
  (PROG ((OS S)
        (OLA LASTADVERB)
        (ORS RESETS)
        (OC CONJUNCTIONS)
        (OP PARSED)
        THIS
        (SEEKING (CONS (create PARSED
                          SOUGHT _ SOUGHT)

```

; Edited 15-Aug-90 12:47 by jds

```

        SEEKING)))
      (SETQ THIS (CAR SEEKING))
      (SETQ LASTPARSED)
LP [COND
  ((LISTP (MSNEXTWORD))
   (COND
    ([PROG ((S (MSNEXTWORD))
             (SEEKING))
            (RETURN (AND [SETQ LASTPARSED (MSSEEKPHRASE (COND
              ((EQ (CAR SOUGHT)
                   'MSSETREP)
                (LIST 'MSSETREP T))
              (T SOUGHT])
              (NOT (MSNEXTWORD])
              (SETQ S (CDR S))
              (GO GOTIT])
            (OR (SETQ LASTPARSED (MSTRYPARSE (CAR SOUGHT)
                                             (CDR SOUGHT)))
              (GO MSFAIL)))
    (GO MSFAIL)))
  (GO GOTIT)
  (replace (PARSED ITEM) of THIS with LASTPARSED)
  (SETQ PARSED (CONS THIS PARSED))
  [COND
    ((AND (NOT FORCONJUNCTION)
          (FMEMB (CAR SOUGHT)
                 CONJUNCTABLE))
     (PROG (LASTPARSED OLDS TEM C (SK SEEKING)
            (MARKER (CONS)))
      LP (MSNEXTWORD) (* So that S won't be reset to something before a adverb)
        (SETQ OLDS S)
        [COND
          ((SETQ C (MSSEEKTYPE 'C))
           [COND
            ((EQ C 'AND)
             (AND (SEEKWORD NOT)
                  (SETQ C ANDNOT])
            (COND
              ((SETQ TEM (MSSEEKPHRASE (COND
                ((EQ (CAR SOUGHT)
                     'MSSETPHRASE)
                 (* kludge. Implicit types should not carry along)
                ' (MSSETPHRASE)
                (T SOUGHT))
                T))
              (SETQ CONJUNCTIONS (CONS (create CONJUNCTION
                C _ C
                PARSED _ (CAR PARSED)
                POSSIBLES _ SK
                START _ MARKER)
                CONJUNCTIONS))
              (GO LP))
              (T (SETQ S OLDS])
              (for X in CONJUNCTIONS when (EQ (fetch (CONJUNCTION START) of X)
                                             MARKER)
                do (replace (CONJUNCTION START) of X with S])
              (replace (PARSED ENDING) of THIS with S)
              (RETURN LASTPARSED))
    (MSFAIL
     (SETQ S OS)
     (SETQ LASTADVERB OLA)
     (SETQ CONJUNCTIONS OC)
     (SETQ PARSED OP)
     (while (NEQ RESETS ORS) do (replace (MSVERB MODIFIER) of (CAR RESETS) with NIL)
            (SETQ RESETS (CDR RESETS]))

```

**(MSTYPEMATCHRATING**

[LAMBDA (X Y)

; Edited 25-Nov-86 21:08 by lmm

(\* Returns the NEGATIVE of a rating of how much two types agree; the values will be sorted and the minimum taken)

```

(COND
  ([OR (NULL X)
        (NULL Y)
        (EQ (COND
              ((LISTP X)
               (SETQ X (CAR X)))
              (T X))
            (COND
              ((LISTP Y)
               (SETQ Y (CAR Y)))
              (T Y))
            (AND (EQ X 'KNOWN)
                 (EQ Y 'FNS))
            (AND (EQ Y 'KNOWN)
                 (EQ X 'FNS])

```

-100)  
(T 0])

(MSTRYSPELL

; Edited 25-Nov-86 21:08 by Imm

```
[LAMBDA (WORD TAIL)
  [AND TAIL (NLISTP (CAR TAIL))
    (LISTP SPELLING)
    (PROG [(PTR (OR (FASSOC TAIL SPELLING)
                    (CAR (SETQ SPELLING (CONS (LIST TAIL)
                                                SPELLING]))
          LP (COND
            ((NULL (CDR PTR))
             (FRPLACD PTR (LIST WORD)))
            ((NEQ (CAR (SETQ PTR (CDR PTR)))
                 WORD)
              (GO LP]
    NIL])
```

(MSTRYPARSE

; Edited 25-Nov-86 21:08 by Imm  
(\* Imm%: 17-FEB-76 20 49)

(\* MSCANT does a RETFROM (MSTRYPARSE)%. The idea is that, no matter how deeply embedded parser is in a particular SEEK, the MSCANT will back up to the current att)

(BLKAPPLY FN ARGS])

(MSCOMMAND

; Edited 15-Aug-90 12:48 by jds

```
[LAMBDA (TEM)
  (COND
    ((SEEKWORD DESCRIBE)
     (CONS LASTPARSED (MSSEEK MSSETPHRASE FNS T)))
    ((SEEKWORD ANALYZE REANALYZE)
     (CONS LASTPARSED (MSSEEK MSSETPHRASE FNS)))
    ((SEEKWORD ERASE FORGET)
     (CONS 'ERASE (MSSEEK MSSETPHRASE FNS)))
    [(SEEKWORD SHOW)
     (COND
       ((SEEKWORD PATHS)
        (CONS 'PATHS (MSSEEK MSPATHOPTIONS)))
       [(MSSEEK MSVERB S)
        (create SENTENCE
          ID _ 'SHOW
          MSPRED _ (create MSSETPHRASE
            REP _ (create THAT
              MSVERB _ LASTPARSED
              OTHERSET _ (PROGN (OR (SEEKWORD OF)
                                   (MSCANT))
                                (OR (MSSEEK MSSETPHRASE)
                                    (MSCANT)]
          (T (SEEKWORD WHERE)
            (create SENTENCE
              ID _ 'SHOW
              SUBJECT _ (OR (MSSEEK MSSETPHRASE)
                           (MSCANT))
              MSPRED _ (MSSEEK MSPRED]
        [(SEEKWORD EDIT)
         (SEEKWORD WHERE)
         (create SENTENCE
           ID _ 'EDIT
           SUBJECT _ (OR (MSSEEK MSSETPHRASE)
                        (MSCANT))
           MSPRED _ (MSSEEK MSPRED)
           OTHERSTUFF _ (COND
             ((SEEKWORD -)
              (PROG1 S (SETQ S]
         [(SEEKWORD FOR)
          (CONS 'FOR (CONS (MSEAT)
            (CONS 'IN (CONS (OR (MSSEEK MSSETPHRASE)
                               (MSCANT))
              (COND
                ((EQ (CAR (GETP (CAR S)
                               'CLISPPWORD))
                     'FORWARD)
                 (PROG1 S (SETQ S NIL)))
                (T (MSCANT]
          ((SEEKWORD CHECK)
           (SEEKWORD BLOCKS)
           (SEEKWORD ON)
           (CONS 'CHECK (MSSEEK MSSETPHRASE FILES)))
        [(SEEKWORD ARE IS)
         (create SENTENCE
           ID _ '?'
           SUBJECT _ (OR (MSSEEK MSSETPHRASE)
            (* IS FOO CALLED BY X)
```



```

(MSCANT)
MSPRED _ (OR (MSSEEKPHRASE (LIST 'MSSETPHRASE (fetch (MSSETPHRASE TYPE) of LASTPARSED)
(fetch (MSSETPHRASE KNOWN) of LASTPARSED)))
(MSCANT]
((PROGN (SETQ TEM (SEEKWORD DOES DO))
(MSSEEK MSSETPHRASE))
(create SENTENCE
ID _ '?
SUBJECT _ LASTPARSED
MSPRED _ (PROG ((TYPE (fetch (MSSETPHRASE TYPE) of LASTPARSED))
(KNOWN (fetch (MSSETPHRASE KNOWN) of LASTPARSED))
SET)
(SETQ SET (OR (MSSEEK MSPRED)
(AND (NOT TEM)
(MSSEEK MSDOES))
(MSCANT)))
(COND
(KNOWN (MSKNOWN SET))
(TYPE (MSSETSETTYPE SET TYPE)))
(RETURN SET])

```

(MSSUBJTYPE

[LAMBDA (MSVERB)

; Edited 15-Aug-90 12:49 by jds
(\* Imm%: 16-DEC-75 1 41)

```

(COND
[(type? CVERB (fetch (MSVERB VPART) of MSVERB))
(MSJOIN (fetch (CVERB C) of (fetch (MSVERB VPART) of MSVERB))
(MSSUBJTYPE (fetch (CVERB VB1) of (fetch (MSVERB VPART) of MSVERB)))
(MSSUBJTYPE (fetch (CVERB VB2) of (fetch (MSVERB VPART) of MSVERB))
((EQ (fetch (MSVERB ROOT) of MSVERB)
'CONTAIN)
'FILES)
(T 'KNOWN])

```

(MSOBJTYPE

[LAMBDA (MSVERB)

; Edited 15-Aug-90 12:49 by jds

```

(COND
[(type? CVERB (fetch (MSVERB VPART) of MSVERB))
(MSJOIN (fetch (CVERB C) of (fetch (MSVERB VPART) of MSVERB))
(MSOBJTYPE (fetch (CVERB VB1) of (fetch (MSVERB VPART) of MSVERB)))
(MSOBJTYPE (fetch (CVERB VB2) of (fetch (MSVERB VPART) of MSVERB))
((EQ (fetch (MSVERB ROOT) of MSVERB)
'CALL)
' (FNS))
((EQ (fetch (MSVERB ROOT) of MSVERB)
'CONTAIN)
NIL)
((EQ (fetch (MSVERB ROOT) of MSVERB)
'USE)
(SELECTQ (fetch (MSVERB MODIFIER) of MSVERB)
(FREELY LOCALLY NIL)
' (VARS))
((RECORDS FIELDS PROPNames I.S.OPRS)
(LIST (fetch (MSVERB MODIFIER) of MSVERB)))
(SHOULDNT))
(T ' (VARS])

```

(MSVERB

[LAMBDA (TENSE)

; Edited 25-Nov-86 21:08 by Imm

```

(PROG ((ROOT (MSSEEKTYPE TENSE))
C)
[COND
(NULL ROOT)
(COND
((AND (EQ TENSE 'ED)
(SEEKWORD ON))
(SETQ S (CONS 'BY S))
(SETQ ROOT CONTAIN))
(T (RETURN)
(RETURN (create MSVERB
ROOT _ ROOT
TENSE _ TENSE
MODIFIER _ (COND
((AND LASTADVERB (FMEMB ROOT (CDR LASTADVERB)))
(PROG1 (CAR LASTADVERB)
(SETQ LASTADVERB NIL]))

```

(MSSPLST

[LAMBDA NIL

; Edited 25-Nov-86 21:08 by Imm

```

(OR MSSPLST (PROGN (SETQ MSSPLST ' (AS FOR))
[MAPHASH MSWORDS (FUNCTION (LAMBDA (MEANING WORD)
(AND [find Y in MEANING
suchthat (FMEMB (CAR Y)

```

' (V PHRASE]  
(SETQ MSSPLST (CONS WORD MSSPLST])

MSSPLST])

(MSWORD?

[LAMBDA (X)  
  
(OR (NOT (LITATOM X))  
(GETHASH X MSWORDS])

; Edited 25-Nov-86 21:08 by Imm  
(\* Imm%: 24-JAN-76 6 14)

(MSSEEK-WORD

[LAMBDA (LST)  
(LET (FND)  
(COND  
((SETQ FND (CL:MEMBER (MSNEXTWORD)  
LST :TEST 'STRING.EQUAL))  
(SETQ LASTPARSED (CAR FND))  
(SETQ S (CDR S))  
LASTPARSED)  
([AND (LISTP SPELLING)  
(LISTP S)  
(NOT (MSWORD? (CAR S))  
(MAPC LST (FUNCTION (LAMBDA (WORD)  
(MSTRYSPELL WORD S]))

; Edited 25-Nov-86 21:08 by Imm

(MSSEEKTYPE

[LAMBDA (TYPE)  
  
;; This was a garbage edit, since it didn't have anything to do with parsing  
;; Check here to see if the current word could be an executable thingy.  
;; (|if| (AND (EQ TYPE 'TYPE) (CL:SYMBOLP CURWORD)) |then| (LET ((CURATOM (|if| (EQ CURWORD 'QUOTE) |then| (CADR S) |else|  
;; CURWORD)) RESULT) ;; This should actually collect them all and ask you if there's more than one. (|for| FPTYPE |in| MSFNTYPES |when|  
;; (HASDEF CURATOM (|fetch| (MSANALYZABLE FILEPKGNAME) |of| FPTYPE)) |do| (SETQ RESULT (|fetch| (MSANALYZABLE SETNAME)  
;; |of| FPTYPE)) (RETURN) ;; (COND ((HASDEF CURATOM 'METHOD-FNS) (SETQ RESULT 'METHODS)) ((HASDEF CURATOM 'CLASS)  
;; (SETQ RESULT 'CLASSES))) (|if| RESULT |then| ;; For some unintelligible reason, we DON'T remove quoted names... (|if| (NOT (EQ  
;; CURWORD 'QUOTE)) |then| (SETQ S (CDR S))) (RETURN RESULT))))

; Edited 1-Apr-88 13:59 by jrb:

(COND  
((SETQ LASTPARSED (GETWORDTYPE (MSNEXTWORD)  
TYPE))  
(SETQ S (CDR S))  
LASTPARSED)  
( (LISTP SPELLING)  
(MSTRYSPELLTYPE TYPE])

(MSRESPELL

[LAMBDA (WORD TAIL SPLST)  
  
(AND (LITATOM WORD)  
WORD  
(SETQ SPLST (CHOOZ WORD 70 SPLST TAIL))  
[COND  
[(LITATOM SPLST)  
(COND  
( (NEQ SPLST WORD)  
(SETQ SPLST (CONS NIL (FRPLACA TAIL SPLST)  
[ (LISTP (CDR SPLST)) (\* Alias)  
;; (SETQ SPLST (CONS NIL (FRPLACA TAIL (CAR (FLAST SPLST))  
(CAR SPLST)  
(SETQ SPLST (FRPLNODE TAIL (CAR SPLST)  
(CONS (CDR SPLST)  
(CDR TAIL]  
SPLST])

; Edited 25-Nov-86 21:08 by Imm  
(\* Imm%: 22-FEB-76 3 6)

(MSJOIN

[LAMBDA (C X Y)  
(COND  
( (OR (EQ X Y)  
(NULL Y))  
X)  
( (NULL X)  
Y)  
(T (UNION (MKLIST X)  
(MKLIST Y])

; Edited 25-Nov-86 21:08 by Imm

(MSSIMPLESET

[LAMBDA (X)  
(OR (NULL X)  
(AND (EQ (fetch (SENTENCE ID) of X)  
'QUOTE)  
(OR (NLISTP (fetch 'QUOTED of X))

; Edited 15-Aug-90 12:52 by jds

(NULL (CDR (fetch 'QUOTED of X))

(MSMATCHRATING

; Edited 15-Aug-90 12:48 by jds

```

[LAMBDA (X Y CONJ)
  (IPLUS (COND
    ((EQUAL (fetch (PARSED SOUGHT) of X)
             (fetch (PARSED SOUGHT) of Y))
      -200)
    (T 0))
  (SELECTQ (CAR (fetch (PARSED SOUGHT) of X))
    (MSVERB (COND
      [(EQUAL (CDR (fetch (PARSED SOUGHT) of X))
              (CDR (fetch (PARSED SOUGHT) of Y)))
        (IPLUS (MSTYPEMATCHRATING (MSSUBJTYPE (fetch (PARSED ITEM) of X))
                                   (MSSUBJTYPE (fetch (PARSED ITEM) of Y)))
              (MSTYPEMATCHRATING (MSOBJTYPE (fetch (PARSED ITEM) of X))
                                   (MSOBJTYPE (fetch (PARSED ITEM) of Y))
              (T 100)))
      (MSSETPHRASE (IPLUS (COND
        ((EQUAL (fetch (MSSETPHRASE TYPE) of (fetch (PARSED ITEM) of X))
                 (fetch (MSSETPHRASE TYPE) of (fetch (PARSED ITEM) of Y)))
          -100)
        (T 0))
      (COND
        ([OR (MSSIMPLESET (fetch (MSSETPHRASE REP) of (fetch (PARSED ITEM) of X))
                          (MSSIMPLESET (fetch (MSSETPHRASE REP) of (fetch (PARSED ITEM) of Y))
          (SELECTQ CONJ
            (OR 50)
            600))
          (T 0))))))
    T])

```

(MSVERBED

; Edited 12-Jun-90 17:58 by teruuchi  
(\* Handled separately SOLELY for the verb SET, which is both present and past tense)

```

[LAMBDA NIL
  (AND (MSSEEK MSVERB ED)
    (create THAT
      MSVERB _ LASTPARSED
      OTHERSET _ (COND
        ((SEEKWORD BY IN)
          (OR (MSSEEK MSSETPHRASE)
              (MSCANT)))
        ((EQUAL (MSNEXTWORD)
                 'OUTPUT)
          (create MSSETPHRASE
            DET _ 'ANY))
        ((MSNEXTWORD)
          (MSCANT))
        (T
          (create MSSETPHRASE
            DET _ 'ANY])

```

; Edited by TT (30-May-1990) This Modification is for parsing the  
; command with "OUTPUT <filename>" [USAR#10196]

(\* Setphrase of ANY)

(MSCANT

; Edited 25-Nov-86 21:08 by lmm

```

[LAMBDA NIL
  (RETFROM 'MSTRYPARSE)]

```

(MSJOINDETS

; Edited 25-Nov-86 21:08 by lmm

```

[LAMBDA (C X Y)
  (COND
    ((NULL X)
     Y)
    ((NULL Y)
     X)
    ((EQ X Y)
     X)
    ((OR (EQ X 'WHICH)
         (EQ Y 'WHICH))
     'WHICH)
    (T (SHOULDNT])

```

(MSNEGATESET

; Edited 25-Nov-86 21:08 by lmm

```

[LAMBDA (X)
  (replace (MSSETPHRASE REP) of X with (create NOT
    NEGATED _ (create MSSETPHRASE using X REP _ (fetch (MSSETPHRASE
    REP)
    of X))

```

**(MSNEXTWORD**

```

[LAMBDA NIL
  (PROG (TEM ADV)
    LPX [COND
      ((EQ NEXTWORD S)
        (RETURN (CAR NEXTWORD])
      (COND
        ((NULL (CAR S))
          (RETURN NIL)))
      (COND
        ((NEQ S SENTENCE)
          (COND
            ((SETQ TEM (GETWORDTYPE (CAR S)
                                   'V))
              (SETQ ADV (CAR TEM))
              (GO ADV)))
            (SELECTQ (SETQ TEM (CAR S))
              ((AS FOR)
                (SELECTQ (CAR (SETQ S (CDR S)))
                  ((A AN)
                    (SETQ S (CDR S)))
                  NIL)
              (CHECKSYNONYM)
                [SETQ TEM (OR (GETWORDTYPE (CAR S)
                                           TEM)
                  (COND
                    ((AND (EQ TEM 'AS)
                          [SOME PARSED (FUNCTION (LAMBDA (X)
                                                    (EQ (CAR (fetch (PARSED SOUGHT)
                                                                    of X))
                                                                    'MSVERB)
                                                    (EQ (fetch (MSVERB ROOT)
                                                                    of (fetch (PARSED ITEM)
                                                                    of X))
                                                                    'CONTAIN])
                          (SETQ TEM (GETFILEPKGTYPE (CAR S)
                                                    NIL T)))
                    [MSETWORDTYPE (CAR S)
                      'AS
                      (SETQ TEM (CONS TEM ' (CONTAIN]
                      TEM)
                      (T (MSFAIL T]
                    (SETQ ADV (CAR TEM))
                    (GO ADV))
                    NIL)))
          (COND
            ((CHECKSYNONYM)
              (GO LPX)))
            (RETURN (CAR (SETQ NEXTWORD S)))
          ADV (SETQ TEM (CDR TEM))
          (SETQ S (CDR S))
          (for X in PARSED when (AND (EQ (CAR (fetch (PARSED SOUGHT) of X))
                                       'MSVERB)
            (NULL (fetch (MSVERB MODIFIER) of (fetch (PARSED ITEM) of X)))
            (FMEMB (fetch (MSVERB ROOT) of (fetch (PARSED ITEM) of X))
                  TEM))
          do (replace (MSVERB MODIFIER) of (fetch (PARSED ITEM) of X) with ADV)
            (SETQ RESETS (CONS (fetch (PARSED ITEM) of X)
                              RESETS))
            (GO LPX))
          [COND
            (LASTADVERB (MSFAIL T])
            (T (SETQ LASTADVERB (CONS ADV TEM)
              (GO LPX]))

```

; Edited 25-Nov-86 21:08 by Imm  
(\* Imm%: 16-MAR-76 15 8)

(\* i.e. either at end of sentence or at a NIL)

(\* No adverbs or AS words allowed at beginning of sentence)

(\* V means ADVERB)

(\* Skip over A and AN)

**(MSPRED**

```

[LAMBDA (FLG)
  (AND (SEEKWORD DOES DO)
    (SETQ FLG T))
  (COND
    [(SEEKWORD NOT)
      (create MSSETPHRASE
        REP _ (create NOT
          NEGATED _ (create MSSETPHRASE
            REP _ (create THAT
              MSVERB _ (OR (MSSEEK MSVERB S)
                (MSCANT))
              OTHERSET _ (OR (MSSEEK MSSETPHRASE)
                (MSCANT))
            [ (MSSEEK MSVERB S)
              (create MSSETPHRASE
                REP _ (create THAT
                  MSVERB _ LASTPARSED

```

; Edited 25-Nov-86 21:08 by Imm  
(\* FLG is used in calls from EDIT/SHOW where passive is not allowed)

```

OTHERSET _ (OR (MSSEEK MSSETPHRASE)
              (MSCANT])
((AND (NULL FLG)
      (SEEKWORD ARE IS))
 (MSSEEK MSSETPHRASE])

```

**(CHECKSYNONYM**

[LAMBDA NIL

; Edited 25-Nov-86 21:08 by Imm  
(\* Imm%: 22-FEB-76 3 10)

(\* Returns T if a synonym is found. Resets S to be sentence with synonym substituted)

```

(PROG (KEYLST (STAIL S)
      (NXT
       (CNT 0))
      (COND
       ([NOT (SETQ KEYLST (GETWORDTYPE (CAR S)
                                       'PHRASE)]
              (RETURN)))
      (PHLOOP
       (SETQ STAIL (CDR STAIL))
      (LPX (COND
           ((SETQ NXT (FASSOC (CAR STAIL)
                             (CDR KEYLST)))
            (SETQ KEYLST (CDR NXT))
            (GO PHLOOP))
          [(CAR KEYLST)
           (RETURN (NOT (EQUAL S (SETQ S (APPEND (CAR KEYLST)
                                                  STAIL]
                               (SETQ NXT (FASSOC '& (CDR KEYLST))))))

```

(\* See SETSYNONYM for def of synonym format)

(\* A & in a synonym will match any word; the word will be substituted for N in the "new" phrase where N is the number of &'s which have been matched -  
e.g. SETSYNONYM ((FOO & &) (IN 1 OR ON 2)) will take FOO FIE FUM into IN FIE OR ON FUM)

```

(SETQ KEYLST (SUBST (CAR STAIL)
                   (SETQ CNT (ADD1 CNT))
                   (CDR NXT)))
(GO PHLOOP))
(SETQ NXT (FASSOC '* (CDR KEYLST)))

```

(\* Just to make a lot of things easier, a \* will match either (NAME) (NAMES) or nothing)

```

(SETQ KEYLST (CDR NXT))
(COND
 ((OR (EQ (CAR STAIL)
          'NAME)
      (EQ (CAR STAIL)
          'NAMES))
  (SETQ STAIL (CDR STAIL]
 (GO LPX))
 (T
  [AND (LISTP STAIL)
        (LISTP SPELLING)
        (NOT (MSWORD? (CAR STAIL)))
        (MAPC (CDR KEYLST)
              (FUNCTION (LAMBDA (X)
                        (MSTRYSPELL X STAIL]
        (RETURN]))

```

(\* If some words match and others don't, try to correct on the rest)

**(MSSETWORDTYPE**

[LAMBDA (WORD TYPE SYN)
(PUTHASH WORD (NCONC1 (GETHASH WORD MSWORDS)
 (CONS TYPE SYN))
MSWORDS])

; Edited 25-Nov-86 21:08 by Imm

**(MSEAT**

[LAMBDA NIL
(PROG1 (CAR S)
 (SETQ S (CDR S))))]

; Edited 25-Nov-86 21:08 by Imm  
(\* Imm%: 5-DEC-75 11 24)

**(MSTRYSPELLTYPE**

[LAMBDA (TYPE)
(DECLARE (SPECVARS TYPE))
[AND (LISTP SPELLING)
 (LITATOM (CAR S))
 (NOT (MSWORD? (CAR S)))
 (MAPHASH MSWORDS (FUNCTION (LAMBDA (INTERP WORD)
 (AND (FASSOC TYPE INTERP)
 (MSTRYSPELL WORD S]

; Edited 25-Nov-86 21:08 by Imm

NIL])

**(MSSETPHRASE**

```
[LAMBDA (TYPE KNOWN) ; Edited 8-Apr-88 11:37 by jrb:
  (PROG (SETDET REP NOTYP SETTYPE VAL (NEG (SEEKWORD NOT)))
    (SETQ VAL (MSSETSETTYPE (create MSSETPHRASE
      DET _ [SETQ SETDET (COND
        ((SETQ NOTYP (SEEKWORD WHO WHOM))
          'WHICH)
        ((SEEKWORD WHAT WHICH)
          'WHICH)
        (T (SEEKWORD ANY THE A)
          'WHICH)
        TYPE _ [SETQ SETTYPE (AND (NOT NOTYP)
          (MSSEEKTYPE 'TYPE]
        REP _ (SETQ REP (MSSEEK MSSETREP))
        KNOWN _ KNOWN)
      TYPE))
    (OR SETDET SETTYPE REP (MSCANT))
    (AND NEG (MSNEGATESET VAL))
    (RETURN VAL))
```

**(MSQUOTED**

```
[LAMBDA NIL ; Edited 25-Nov-86 21:08 by lmm
  (create QUOTE
    QUOTED _ (MKLIST (MSEAT]))
```

**(FIXVERBSETTYPE**

```
[LAMBDA (SETPHRSE) ; Edited 15-Aug-90 12:51 by jds
  (COND (* lmm%: 16-MAR-76 15 31)
    ((NULL (fetch (MSSETPHRASE TYPE) of SETPHRSE))
      (MSVERBTYPE (fetch (THAT MSVERB) of (fetch (MSSETPHRASE REP) of SETPHRSE))
        SETPHRSE
        (fetch (THAT OTHERSET) of (fetch (MSSETPHRASE REP) of SETPHRSE))
```

**(SETSYNONYM**

```
[LAMBDA (OLDPHRASE NEWPHRASE FLG) ; Edited 25-Nov-86 21:08 by lmm
  (OR (LISTP OLDPHRASE)
    (SETQ OLDPHRASE (LIST OLDPHRASE)))
  [AND NEWPHRASE (OR (LISTP NEWPHRASE)
    (SETQ NEWPHRASE (LIST NEWPHRASE)
      (SETQ NEXT (FASSOC (CAR OLDPHRASE)
        KEYLST))
      (SETQ KEYLST (CDR NEXT)))
    (T (FRPLACD KEYLST (CONS (CONS (CAR OLDPHRASE)
      (SETQ NEXT (LIST NIL)))
      (CDR KEYLST)))
      (SETQ KEYLST NEXT)))
    (GO LP)]
  LP (COND
    ((NLISTP KEYLST)
      (SHOULDNT))
    (NULL (SETQ OLDPHRASE (CDR OLDPHRASE)))
    (FRPLACA KEYLST NEWPHRASE)
    (RETURN))
    ((SETQ NEXT (FASSOC (CAR OLDPHRASE)
      KEYLST))
      (SETQ KEYLST (CDR NEXT)))
    (T (FRPLACD KEYLST (CONS (CONS (CAR OLDPHRASE)
      (SETQ NEXT (LIST NIL)))
      (CDR KEYLST)))
      (SETQ KEYLST NEXT)))
    (GO LP)]
  (* The keylist is of the form (VALUE
  (WORD . KEYLST) (WORD . KEYLST) [...]))
```

**(MSSETUP**

```
[LAMBDA (V A AS F SYN TYPES OTHERS INITFLG) ; Edited 25-Nov-86 21:08 by lmm
  [OR (AND (NULL INITFLG)
    (HASHARRAYP MSWORDS))
    (SETQ MSWORDS (HASHARRAY 150 NIL 'STRING-EQUAL-HASHBITS 'STRING.EQUAL)]
  [MAPC V (FUNCTION (LAMBDA (X)
    (MSSETWORDTYPE (CAR X)
      'S
      (CAR X))
    (MSSETWORDTYPE (CADR X)
      'S
      (CAR X))
    (MSSETWORDTYPE (CADDR X)
      'ING
      (CAR X))
```

```

      (MSSETWORDTYPE (CADDR X)
        'ED
        (CAR X])
[MAPC A (FUNCTION (LAMBDA (X P)
  (SETQ P (CONS (CAAR X)
    (CDR X)))
  (MAPC (CAR X)
    (FUNCTION (LAMBDA (Y)
      (MSSETWORDTYPE Y 'V P])
[MAPC AS (FUNCTION (LAMBDA (X P)
  (SETQ P (CONS (CAAR X)
    (CDR X)))
  (MAPC (CAR X)
    (FUNCTION (LAMBDA (Y)
      (MSSETWORDTYPE Y 'AS P])
[MAPC F (FUNCTION (LAMBDA (X P)
  (SETQ P (CONS (CAAR X)
    (CDR X)))
  (MAPC (CAR X)
    (FUNCTION (LAMBDA (Y)
      (MSSETWORDTYPE Y 'FOR P])
[MAPC SYN (FUNCTION (LAMBDA (X)
  (SETSYNONYM (CAR X)
    (CADR X)
    T])
[MAPC TYPES (FUNCTION (LAMBDA (L)
  (MAPC L (FUNCTION (LAMBDA (X)
    (MSSETWORDTYPE X 'TYPE (CAR L])
[MAPC OTHERS (FUNCTION (LAMBDA (X)
  (MSSETWORDTYPE (CAR X)
    (CADR X)
    (OR (CADDR X)
    (CAR X])

```

(\* I.e. all the things that can occur almost anywhere - synonyms, adverbs, and the word AS)

NIL])

)

(DECLARE%: EVAL@COMPILE DONTCOPY

(RPAQQ **PARSERRECORDS** (SENTENCE CONJUNCTION PARSED MSSETPHRASE MSVERB VPART CVERB THAT QUOTE IN NOT PATHS APPLY CSET BLOCKS FIELDS))

(DECLARE%: EVAL@COMPILE

(RECORD SENTENCE (ID SUBJECT MSPRED . OTHERSTUFF))

(RECORD CONJUNCTION (C START PARSED . POSSIBLES))

(RECORD PARSED (SOUGHT ITEM . ENDING))

(RECORD MSSETPHRASE (DET (TYPE KNOWN . DEFAULTTYPE) . REP) (\* represents a set of things)
(TYPE? (EQLLENGTH (CADR DATUM) 2)))

(RECORD MSVERB (TENSE . VPART)
(SUBRECORD VPART))

(RECORD VPART (ROOT . MODIFIER))

[RECORD CVERB (C VB1 . VB2)
(TYPE? (FMEMB (CAR DATUM) ' (OR AND ANDNOT])

[RECORD THAT (ID MSVERB . OTHERSET)
ID \_ 'THAT (TYPE? (EQ (CAR DATUM) 'THAT])

(RECORD QUOTE (ID . QUOTED)
ID \_ 'QUOTE)

(RECORD IN (ID . EXPRESSION)
ID \_ 'IN)

[RECORD NOT (ID . NEGATED)
ID \_ 'NOT (TYPE? (EQ (CAR DATUM) 'NOT])

(RECORD PATHS (ID . MSPATHOPTIONS)
ID \_ 'PATHS)

(RECORD APPLY (ID . PRED)
ID \_ 'APPLY)

```

{MEDLEY}<library>MSPARSE.;1

(RECORD CSET (ID SET1 . SET2))

(RECORD BLOCKS (ID TYPES FNS . FILES)
  ID _ 'BLOCKS                                (* e.g. ENTRIES ON FOO))

(RECORD FIELDS (ID . RECS)
  ID _ 'FIELDS)
)

(DECLARE%: EVAL@COMPILE

(PUTPROPS GETWORDTYPE MACRO [(WORD TYPE)
                               (CDR (FASSOC TYPE (GETHASH WORD MSWORDS))]
)
)

(RPAQQ CONJUNCTABLE (MSSETPHRASE MSPRED MSDOES MSVERB))

(RPAQQ MSSPLST NIL)

(RPAQQ INITWORDLIST
  (( (CONTAIN CONTAINS CONTAINING CONTAINED)
    (BIND BINDS BINDING BOUND)
    (CALL CALLS CALLING CALLED)
    (SET SETS SETTING SET)
    (SMASH SMASHES SMASHING SMASHED)
    (TEST TESTS TESTING TESTED)
    (USE USES USING USED)
    (REFERENCE REFERENCES REFERENCING REFERENCED)
    (CREATE CREATES CREATING CREATED)
    (FETCH FETCHES FETCHING FETCHED)
    (REPLACE REPLACES REPLACING REPLACED)
    (DECLARE DECLARES DECLARING DECLARED))
  (( (FREELY FREE)
    USE SET SMASH TEST)
  (( (LOCALLY LOCAL)
    USE SET SMASH BIND DECLARE REFERENCE)
  (( (SOMEHOW)
    CALL)
  (( (DIRECTLY)
    CALL)
  (( (INDIRECTLY)
    CALL USE))
  (( (LOCALVARS LOCALVAR)
    DECLARE)
  (( (SPECVARS SPECVAR)
    DECLARE)
  (( (RECORDS RECORD)
    USE)
  (( (FIELDS FIELD)
    USE SET SMASH REFERENCE)
  (( (PROPNames PROPERTIES PROPERTY PROP PROPS)
    USE)
  (( (I.S.OPRS I.S.OPR CLISPWORD CLISPWORDS)
    USE)
  (( (MSPRED)
    CALL USE)
  (( (TYPE)
    USE))
  (( (VALUE)
    CALL USE)
  (( (TESTING)
    CALL USE)
  (( (EFFECT)
    CALL))
  (( (RECORD FIELD *)
    FIELD)
  (( (RECORD FIELDS)
    FIELDS)
  (( (FIELD *)
    FIELD)
  (( (RECORD *)
    RECORD)
  (( (CLISP WORD)
    CLISPWORD)
  (( (CLISP WORDS)
    CLISPWORD)
  (( (PROP *)
    PROPNames)
  (( (PROPERTY *)
    PROPNames)
  (( (ON PATH)
    ONPATH)
  (( (ON THE PATH)
    ONPATH)
  (( (ON A PATH)
    ONPATH)
)
)

```



```

((LOOK AT)
ANALYZE)
[(CHANGED ON &)
 (IN (PROGN (UPDATEFILES)
 (CDR (GETPROP '1 'FILE)
 (AMONG (AVOIDING NOT))
 (DOESN'T (DOES NOT))
 (DON'T (DO NOT))
 (ISN'T (IS NOT))
 (AREN'T (ARE NOT))
 ((ALL PATHS)
 PATHS)
 (SOME ANY)
 (ALL ANY)
 (ANYONE ANY)
 ((ALL PATHS)
 PATHS)
 (UNDEFINED (FNS NOT @ GETD)))
 (FNS FN FUNCTIONS FUNCTION)
 (VARS VAR VARIABLES VARIABLE)
 (RECORDS RECORD)
 (FILES FILE)
 (I.S.OPRS I.S.OPR CLISPCWORD CLISPCWORDS)
 (FIELDS FIELD)
 (PROPNAME PROPERTIES PROPS PROP))
 (FROM PATH)
 (TO PATH)
 (AVOIDING PATH)
 (NOTRACE PATH)
 (MARKING PATH)
 (OUTPUT PATH)
 (LINELENGTH PATH)
 (DOES X DO)
 (DO X)
 (IS X)
 (ARE X IS)
 (AND C)
 (OR C)
 (SEPARATE PATH)
 (LOCALFREEVARS BLOCKS)
 (SPECVARS BLOCKS)
 (LOCALVARS BLOCKS)
 (GLOBALVARS BLOCKS)
 (BLOCKS BLOCKS BLKFNS)
 (BLOCK BLOCKS BLKFNS)
 (ENTRY BLOCKS ENTRIES)
 (ENTRIES BLOCKS)
 (BLKAPPLYFNS BLOCKS)
 (BLKLIBRARY BLOCKS)
 (BLKFNS BLOCKS))
T)

```

(APPLY 'MSSETUP INITWORDLIST)

```

(DEFMACRO MSSEEK (&REST SOUGHT)
 '(MSSEEKPHRASE ' ,SOUGHT))

```

```

(DEFMACRO SEEKWORD (&REST SOUGHT)
 '(MSSEEK-WORD ' ,SOUGHT))

```

```

(DEFCOMMAND "." (&REST LINE) "Masterscope query"
 (MASTERSCOPE LINE))

```

(DECLARE%: DONTEVAL@LOAD DOEVAL@COMPILE DONTCOPY

```

(BLOCK%: MSPARSEBLOCK MSKNOWN MSDOES MSSETSETTYPE MS-SMASH-PACKAGE MSFAIL MSJOINSET MSBLOCK MSPATHOPTIONS
MSPARSE MSFIXUPTYPES MSVERBTYPE MSSETREP MSSEEKPHRASE MSTYPEMATCHRATING MSTRYSPELL MSTRYPARSE MSCOMMAND
MSSUBJTYPE MSOBJTYPE MSVERB MSSPLST MSWORD? MSSEEKTYPE MSRESPELL MSJOIN MSSIMPLESET MSMATCHRATING
MSVERBED MSCANT MSJOINDETS MSNEGATESET MSNEXTWORD MSPRED CHECKSYNONYM MSEAT MSTRYSPELLTYPE MSSETPHRASE
MSQUOTED FIXVERBSETTYPE (ENTRIES MSPARSE MSJOINSET MSTRYSPELL SETSYNONYM MSSETUP)
(REFNS MSTRYPARSE)
(SPECVARS SPELLING)
(LOCALFREEVARS CONJUNCTIONS LASTADVERB LASTPARSED NEXTWORD PARSED S SEEKING SENTENCE RESETS)
(BLKAPPLYFNS MSDOES MSVERBED MSBLOCK MSPATHOPTIONS MSSETREP MSCOMMAND MSVERB MSPRED MSSETPHRASE)
(LOCALFREEVARS SETTYPE SETDET)
(NOLINKFNS . T)
(GLOBALVARS MSWORDS MSSPLST CONJUNCTABLE)
MSSETUP MSSETWORDTYPE SETSYNONYM MSSEEK-WORD)

```

```

(BLOCK%: NIL (LOCALVARS . T)
 (GLOBALVARS MSWORDS))
)

```

(PUTPROPS **MSPARSE FILETYPE** :COMPILE-FILE)

{MEDLEY}<library>MSPARSE.;1

Page 18

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

(ADDTOVAR **NLAMA** )

(ADDTOVAR **NLAML** )

(ADDTOVAR **LAMA** )

)

(PUTPROPS **MSPARSE COPYRIGHT** ("Venue & Xerox Corporation" 1984 1986 1987 1988 1990))

---

**FUNCTION INDEX**

CHECKSYNONYM . . . . .13	MSFIXUPTYPES . . . . .4	MSPARSE . . . . .3	MSSETREP . . . . .5	MSTRYSPELLTYPE . . .13
FIXVERBSETTYPE . . .14	MSJOIN . . . . .10	MSPATHOPTIONS . . . .3	MSSETSETTYPE . . . .1	MSTYPEMATCHRATING .7
MS-SMASH-PACKAGE . .2	MSJOINDETS . . . . .11	MSPRED . . . . .12	MSSETUP . . . . .14	MSVERB . . . . .9
MSBLOCK . . . . .3	MSJOINSET . . . . .2	MSQUOTED . . . . .14	MSSETWORDTYPE . . .13	MSVERBED . . . . .11
MSCANT . . . . .11	MSKNOWN . . . . .1	MSRESPELL . . . . .10	MSSIMPLESET . . . . .10	MSVERBTYPE . . . . .5
MSCOMMAND . . . . .8	MSMATCHRATING . . .11	MSSEEK-WORD . . . . .10	MSSPLST . . . . .9	MSWORD? . . . . .10
MSDOES . . . . .1	MSNEGATESET . . . . .11	MSSEEKPHRASE . . . . .6	MSSUBJTYPE . . . . .9	SETSYNONYM . . . . .14
MSEAT . . . . .13	MSNEXTWORD . . . . .12	MSSEEKTYPE . . . . .10	MSTRYPARSE . . . . .8	
MSFAIL . . . . .2	MSOBTJTYPE . . . . .9	MSSETPHRASE . . . . .14	MSTRYSPELL . . . . .8	

---

**RECORD INDEX**

APPLY . . . . .15	CSET . . . . .16	IN . . . . .15	NOT . . . . .15	QUOTE . . . . .15	VPART . . . . .15
BLOCKS . . . . .16	CVERB . . . . .15	MSSETPHRASE . . .15	PARSED . . . . .15	SENTENCE . . . . .15	
CONJUNCTION . .15	FIELDS . . . . .16	MSVERB . . . . .15	PATHS . . . . .15	THAT . . . . .15	

---

**VARIABLE INDEX**

CONJUNCTABLE .16	INITWORDLIST .16	MSSPLST . . . . .16	PARSERRECORDS 15
------------------	------------------	---------------------	------------------

---

**MACRO INDEX**

GETWORDTYPE . .16	MSSEEK . . . . .17	SEEKWORD . . . . .17
-------------------	--------------------	----------------------

---

**PROPERTY INDEX**

MSPARSE . . . . .17
---------------------

---

**COMMAND INDEX**

". " . . . . .17
------------------

---