

```

<NLS>SYNTAX.NLS;10, 29-OCT-73 13:03 CFD ;
FILE nlslanguage % CML.SAV TO <rel-nls>SYNTAX.REL %
% COMMON RULES %
% ENTITY DEFINITIONS %
  editentity = textent / structure;
% TEXT ENTITY DEFINITIONS %
  textent = text1 / "TEXT"!L1!;
  text1 = "CHARACTER"!L1! / "WORD"!L1! / "VISIBLE"!L1! /
  "INVISIBLE"!L1! / "NUMBER"!L1! / "LINK"!L1!;
% STRUCTURE ENTITY DEFINITIONS %
  structure = "STATEMENT"!L1! / notstatement;
  notstatement = "GROUP"!L1! / "BRANCH"!L1! / "PLEX"!L1! ;
% SUBSYSTEM NAMES %
  nlssubs = "EDITOR"!L1! / "CALCULATOR"!L1! / "FORMS"!L1!
  / "HELP"!L1! / "IDENTIFICATION"!L1! / "JOURNAL"!L1!
  / "MEASUREMENT"!L1! / "PROGRAM"!L1! / "QUERY"!L1!
  / "USEROPTIONS"!L1! ;
% ANSWER %
  answer = ("YES"!L1 L1!/"NO"!L1 O! / "<CA>"!L1 L1!);
% SWITCH %
  switch = ("ON"!L1 L1!/"OFF"!O!);
% RECOGNITION MODES %
  rmode = ("EXPERT"!L1! / "ANTICIPATORY"!L1! / "DEMAND"!L1! /
  "FIXED"!L1! );
% CASE SHIFT MODES %
  cshmode =
  ( "UPPER"!L1!
  / "LOWER"!L1!
  / "INITIAL"!L1! <"upper"> );
% DIRECTORY OPTIONS %
  diropt =
  ( "ACCOUNT"!L1! / "ARCHIVE"!L1! / "CHRONOLOGICAL"!L1! /
  "GRAM" / "DATES"!L1! / "DELETE" / "EVERYTHING"!L1! /
  "LENGTH"!L1! / "PROTECT"!L1! / "REVERSE"!L1! / "SIZE"!L1! /
  "TIME"!L1! / "VERBOSE"!L1!);
% DECLARATIONS %
  DECLARE VARIABLE
    %Journal subsystem%
    authfield,
    clerfield,
    commfield,
    distfield,
    formfield,
    keywfield,
    linkfield,
    numbfield,
    numbtype,
    obsofield,
    subcfield,
    submdest,
    submtype,
    titlfield,
    updafield,
    aheadfilename,
    dent,
    dest,

```

```

ent,
filtre,
filename,
ff,
fromwhom,
level,
literal,
param,
param2,
param3,
pb,
port,
retfilename,
sent,
sim,
source,
subsys,
tip,
vs;
DECLARE PARSEFUNCTION
sp,          % reads next char, TRUE if space %
readca,      % reads next char if ca %
lookca,      % TRUE if next char is CA %
looknum,     % TRUE if next char is a number %
notca;       % reads next char, TRUE iff not CA char %
SUBSYSTEM nlseditor
% NLS EDITOR COMMANDS %
COMMAND %verify%
zverify =
  "VERIFY"!L1! "FILE"!L1! CONFIRM xverify( ) ;
COMMAND %update%
zupdate =
  "UPDATE"!L1! "FILE"!L1!
  filename ← NULL
  ent ← #"NEW"
  /ent ←
    ( "OLD"!L1! <"version">
    / "COMPACT"!L1!
    / "RENAME"!L1! <"filename">
      filename ← LSEL( #"FILE" ) ) )
  CONFIRM
  xupdate( ent, filename ) ;
COMMAND %undelete%
zundelete =
  "UNDELETE"
  ent ←
    ( "FILE"!L1!
    / "ARCHIVE"!L1! <"file">
    / "MODIFICATIONS"!L1! <"to file"> )
  filename ← LSEL( #"FILE" )
  CONFIRM
  xundelete( ent, filename ) ;
COMMAND %trim%
ztrim =
  "TRIM" "DIRECTORY"!L1!
  <"no. versions to keep"> param ← LSEL( #"NUMBER" )

```

```

        CONFIRM <"really?">
        CONFIRM
        xtrim( param ) ;
COMMAND %transpose%
ztranspose =
    "TRANSPOSE"!L1!
    sent ← editentity
    <"at"> source ← DSEL( sent )
    <"and"> dent ← sent
    dest ← DSEL( dent )
    vs ← NULL filter ← FALSE
    / <"Filtered:"> filter ← TRUE vs ← VIEWSPECS /
    CONFIRM
    xtranspose( sent, source, dent, dest, filter, vs ) ;
COMMAND %substitute%
zsubstitute =
    "SUBSTITUTE"!L1!
    filter ← FALSE vs ← NULL
    sent ← textent
    <"in"> dent ← structure
    <"at"> dest ← DSEL(dent)
    %** collect param ← subpairs( sent ) **%
    param ← NULL
    copy2
    CONFIRM
    xsubstitute( sent, dent, dest, param, filter, vs ) ;
COMMAND %stop%
zstop =
    "STOP" "RECORD"!L1! <"of Session"> CONFIRM
    xstop() ;
COMMAND %split%
zsplit =
    "SPLIT"!NOTT! "WINDOW"!L1!
    param ← ("HORIZONTALLY"!L1! / "VERTICALLY"!L1!)
    CONFIRM
    xsplit( param ) ;
COMMAND %sort%
zsort =
    "SORT"
    dent ← notstatement
    dest ← DSEL( dent )
    CONFIRM
    xsimulate( dent, dest ) ;
COMMAND %simulate%
zsimulate =
    "SIMULATE" "TERMINAL"!L1! <"type">
    ent ←
        ( "DISPLAY"!L1!
        / "TI"!L1! <"Terminal">
        / "EXECUPORT"!L1!
        / "33-TTY"!L1 33!
        / "35-TTY"!L1 35!
        / "37-TTY"!L1 37! )
    CONFIRM
    xsimulate( ent ) ;
COMMAND %show%

```

```

zshow =
  "SHOW"
  ent ←
    ( "FILE"!Ll!
      param ←
        ( "STATUS"!Ll!
          / "LINK"!Ll! <"default directory">
          / "MARKER" <"list">
          / "MODIFICATIONS"!Ll! <"status">
          / "RETURN"!Ll! <"ring">
          / "SIZE"!Ll! )
      / "ARCHIVE"!Ll! <"directory">
      param ← (readca() / LSEL("#NAME") )
      %CAN WE DO THIS???%
      / "DIRECTORY"!Ll!
      param ← (readca() / LSEL("#NAME") $(diropt/ )
      %CAN WE DO THIS???%
      / "DISK" <"space status">
      / "NAME"!Ll! <"delimiters for statement at">
      param ← DSEL( "#STATEMENT" )
      / "VIEWSPECS"!Ll! <"status">
      param ← NULL
      [param ← "VERBOSE"!Ll!/ )
    CONFIRM
    xshow( ent, param ) ;
COMMAND %set%
zset =
  "SET"
  dest ← NULL param ← NULL param2 ← NULL param3 ← NULL
  ent ←
    ( "CASE"!Ll!
      ( param ← editentity
        <"at"> dest ← DSEL( param )
        param2 ← NULL [param2 ← cshmode]
      / param ← "MODE"!Ll!
        param2 ← cshmode )
      / "CHARACTER"!NOTT!
        <"size for window to"> param ← DSEL( "#NUMBER" )
      / "FEEDBACK" <"mode">
        param ← ("VERBOSE"!Ll! / "TERSE"!Ll!)
      / "FILTER"!Ll!
        param ←
          ( "TO"!Ll!
            <"pattern"> param2 ← LSEL( "#CHARACTER" )
            / switch )
      / "LINK"!Ll!
        <"default for file to directory">
        param2 ← LSEL("#NAME")
      / "NAME"!Ll!
        <"delimiters in"> param ← structure
        <"at"> dest ← DSEL( param )
        CLEAR
        <"left delimiter"> param2 ← LSEL("#CHARACTER")
        <"right delimiter"> param3 ← LSEL("#CHARACTER")
      / "PROMPT"!Ll!
        param ← ("OFF"!Ll! / "PARTIAL"!Ll! / "FULL"!Ll!)

```

```

    / "RECOGNITION"!L1! <"mode">
      param ← rmode
    / "TEMPORARY"!L1! <"modifications for file">
    / "TTY"!NOTT! <"window to window">
      param ← LSEL("#WINDOW")
    / "VIEWSPECS"!L1!
      param ← VIEWSPECS )
CONFIRM
xset( ent, param, param2, param3, dest ) ;
COMMAND %retrieve%
zretrieve =
  "RETRIEVE" "FILE"!L1! <"from archive">
    filename ← LSEL("#FILE")
  CONFIRM
  xretrieve( filename );
COMMAND %reset%
zreset =
  "RESET"
  dest ← NULL dent ← NULL
  ent ←
    ( "ARCHIVE"!L1! <"request for file">
      dest ← LSEL("#FILE")
    / "CASE"!L1! <"mode">
    / "CHARACTER"!L1 NOTT! <"size for window">
    / "FEEDBACK" <"mode">
    / "FILTER"!L1!
    / "LINK"!L1! <"default for file">
    / "NAME"!L1! <"delimiters in">
      dent ← structure
      <"at"> dest ← DSEL(dent)
    / "PROMPT"!L1!
    / "RECOGNITION"!L1! <"mode">
    / "TEMPORARY"!L1! <"modifications for file">
    / "TTY"!NOTT! <"window">
    / "VIEWSPECS"!L1! )
  CONFIRM
  xreset( ent, dent, dest );
COMMAND %replace%
zreplace =
  "REPLACE"!L1!
  dent ← editentity
  <"at"> dest ← DSEL(dent)
  sent ← dent
  <"by"> source ← LSEL(sent)
  CONFIRM
  xreplace( dent, dest, sent, source );
COMMAND %renumber%
zrenumber =
  "RENUMBER" "SIDS"!L1!
  <"in file"> CONFIRM
  xrenumber( );
COMMAND %release%
zrelease =
  "RELEASE"!NOTT!
  ent ←
    ( "FROZEN"!L1! <"statement at">

```

```

        dest ← DSEL("#STATEMENT")
        / "ALL"!L1! <"frozen statements"> dest ← NULL )
    CONFIRM
    xrelease( ent, dest );
COMMAND %record%
    zrecord =
        "RECORD" "SESSION"!L1! <"on file"> filename ← LSEL("#FILE")
        CONFIRM
        xrecord( filename );
COMMAND %protect%
    zprotect =
        "PROTECT"!L1! "FILE"!L1! filename ← LSEL("#FILE")
        <"from">
        fromwhom ← ("SELF"!L1! / "GROUP"!L1! / "OTHERS"!L1!)
        param ← #"ALL"
        [
            <"protection"> param ←
                ( "READ"!L1 1!
                  / "WRITE"!L1 2!
                  / "EXECUTE"!L1 4!
                  / "APPEND"!L1 8!
                  / "LIST"!L1 16!
                  / "ALL"!L1 31! <"access"> )
        ]
        CONFIRM
        xprotect( filename, fromwhom, param );
COMMAND %print%
    zprint =
        "PRINT"!L1 NOTD!
        ( readca() ent ← #"REST" dest ← NULL vs ← NULL
        / ent ← structure
        <"at"> dest ← DSEL( ent )
        vs ← VIEWSPECS )
        xprint( ent, dest, vs );
COMMAND %playback%
    zplayback =
        "PLAYBACK" "SESSION"!L1!
        <"from file"> filename ← LSEL("#FILE")
        CONFIRM
        xplayback( filename );
COMMAND %output%
    zoutput =
        "OUTPUT"!L1!
        ( ( ent ←
            ("QUICKPRINT"!L1!
            / "JOURNAL"!L1! <"Quickprint">
            / "PRINTER"!L1!
            / "COM"!L1! )
            filename ← NULL
            param ← TRUE % use default number of copies %
            % construct default file name %
            %** filename ← defilname( ent ) **%
        [
            "FILE"!L1! filename ← LSEL("#FILE") /
            "COPIES"!L1! param ← LSEL("#NUMBER")
        ]
    )

```

```

        CONFIRM
        xout1( ent, filename, param ))
/ ( ent ←
  ("SEQUENTIAL"!L1! /
  "ASSEMBLER"!L1! )
  <"file"> filename ← LSEL("#FILE")
  param ← FALSE
  ["FORCE"!L1! <"upper case"> param ← TRUE/
  CONFIRM
  xout1( ent, filename, param ))
/ (
  ( ent ← "TERMINAL"!L1! tip ← NULL port ← NULL
  / ent ← "REMOTE"!L1!
    <"printer -- TIP"> tip ← LSEL("#VISIBLE")
    <"Port #"> port ← LSEL("#NUMBER") )
  CLEAR
  <"Send Form Feeds?">
  ff ←
    ( "YES"!L1 1! sim ← FALSE
    / "NO"!L1 0! <"Simulate?"> sim ← answer)
  CLEAR
  <"Wait at page break?"> pb ← answer
  CLEAR
  <"Go?">
    ( readca()
    / "YES"!L1!
    / "NO"!L1!
    <"Type CR when ready, CD to abort">
    "
    "!L1!)
  xout2( ent, tip, port, ff, sim, pb)
);
COMMAND %move%
zmove =
  "MOVE"!L1!
  filtre ← FALSE
  vs ← NULL
  level ← NULL
  ( sent ← text1
  copy1
  dent ← sent
  dest ← DSEL(dent)
  / sent ← "TEXT"!L1!
  copy1
  dent ← #"CHARACTER"
  dest ← DSEL(dent)
  / sent ← structure
  copy1
  dent ← #"STATEMENT"
  dest ← DSEL(dent)
  level ← LEVADJ
  copy2
  / dent ← "FILE"!L1!
  sent ← dent
  <"from old filename"> source ← LSEL(sent)
  <"to new filename"> dest ← LSEL(dent)

```

```

    / dent ← "BOUNDARY"!NOTT!
      sent ← dent
      <"from"> source ← LSEL("#WINDOW")
      <"to"> dest ← LSEL("#WINDOW")
    )
    CONFIRM
    xmove(sent, source, dent, dest, level, filtre, vs);
COMMAND %merge%
  zmerge =
    "MERGE"
    sent ← notstatement
    <"at"> source ← SSEL(sent)
    dent ← sent
    <"into"> dest ← DSEL(dent)
    CONFIRM
    xmerge( sent, source, dent, dest);
COMMAND %mark%
  zmark =
    "MARK" "CHARACTER"!Ll! <"at">
    dest ← DSEL( "#CHARACTER" )
    <"with marker named"> source ← LSEL("#NAME")
    CONFIRM
    xmark( dest, source);
COMMAND %logout%
  zlogout =
    "LOGOUT"
    CONFIRM xlogout();
COMMAND %load%
  zload =
    "LOAD"!Ll!
    ent ← ("FILE"!Ll! / "BUSY"!Ll! <"file">)
    filename ← LSEL("#FILE") CONFIRM
    xload(ent, filename);
%k (unused)%
COMMAND %tjump%
  ztjump =
    "JUMP"!NOTD Ll! <"to">
    dest ← DSEL("#CHARACTER")
    xjump("#TITEM", dest, NULL);
COMMAND %djump%
  zdjump =
    "JUMP"!NOTT Ll! <"to">
    (
      lookca() % look for a bug select %
      ent ← "#ITEM"
      dest ← DSEL( "#STATEMENT" )
      dest ← xjdae( dest ) %LSEL( "#ADDR" )% vs ←
      VIEWSPECS
    / ent ←
      (
        "ITEM"!Ll!
        / "SUCCESSOR"!Ll!
        / "PREDECESSOR"!Ll!
        / "UP"!Ll!
        / "DOWN"!Ll!
        / "HEAD"!Ll!

```



```

    / "TAIL"!LL!
    / "END"!LL! <"of Branch">
    / "BACK"!LL!
    / "ORIGIN"!LL!
    / "NEXT"
)
dest ← DSEL("#STATEMENT") vs ← VIEWSPECS

(
ent ← "LINK"!LL! (dest ← LSEL("#LINK")
vs ← VIEWSPECS)
/ ent ← "RETURN"!LL! dest ← NULL vs ← NULL
  %** ca()
  retstat ← NULL
  $[
    retstat ← retext(retstat) %**
    %displays textent from 'return' statement%
    %returns next 'return' statement%
    %** notca() ] %**
/ ent ← "AHEAD"!LL! dest ← NULL vs ← NULL
  %** ca()
  aheadstat ← NULL
  $[
    aheadstat ← aheadtext(aheadstat)
    notca() ] %**
    %displays textent from 'ahead' statement%
    %returns next 'ahead' statement%
/ ent ← "FILE"!LL! dest ← NULL vs ← NULL
  (
    sp() dest ← LSEL("#FILE")
    / lookca() dest ← DSEL("#FILE") vs ← VIEWSPECS
    / "AHEAD"!LL! ent ← "#FILEAHEAD"
    aheadfilename ← NULL %**
    $[
      aheadfilename ← aheadfile(aheadfilename)
      notca() ] %**
      %displays name of 'ahead' file%
      %returns next 'ahead' file%
    / "RETURN"!LL! ent ← "#FILERETURN"
    retfilename ← NULL %**
    $[
      retfilename ← retfile(retfilename)
      notca() ] %**
      %displays name of 'return' file%
      %returns next 'return' file%
  )
)
/ ent ← "NAME"!LL!
  ("FIRST"!LL! (ent ← "#FIRSTNAME") / "NEXT"!LL!
  (ent ← "#NEXTNAME") / "ONLY"!LL!)
  dest ← LSEL("#NAME") vs ← VIEWSPECS
/ "CONTENT"!LL!
  ("FIRST"!LL! ent ← "#FIRSTCONTENT" / "NEXT"!LL!
  ent ← "#NEXTCONTENT")
  (dest ← LSEL("#TEXT") %** / option() %** %Accept
  old content%) vs ← VIEWSPECS
  %textent may contain elipses notation (...)%)

```

```

        %Default IS "next"%
    / "WORD"!LL!
      ("FIRST"!LL! ent ← #"FIRSTWORD" / "NEXT"!LL! ent ←
      #"NEXTWORD")
      (dest ← LSEL("#WORD") %** / option() %** %Accept
      old content%) vs ← VIEWSPECS
      %textent may contain elipses notation (...)%
      %Default is "next"%
    )
  )
  CONFIRM
  xjump(ent, dest, vs);
COMMAND %insert%
zinsert =
  "INSERT"!LL!
  level ← NULL
  ( ent ← text1
    <"to follow">
    dest ← DSEL(ent)
    param ← LSEL(ent)
  / ent ← "TEXT"!LL!
    <"to follow">
    dest ← DSEL("#CHARACTER")
    param ← LSEL(ent)
  / ent ← structure
    <"to follow">
    dest ← DSEL("#STATEMENT")
    level ← LEVADJ
    param ← LSEL(ent)
  / ent ← "JOURNAL"!LL! <"submission form">
    <"to follow">
    dest ← DSEL("#STATEMENT")
    level ← LEVADJ
    param ← NULL
  )
  CONFIRM
  xinsert(ent, dest, level, param);
% h (unused) %
COMMAND %freeze%
zfreeze =
  "FREEZE"!LL! NOTT! "STATEMENT"!LL! <"at">
  dest ← DSEL("#STATEMENT")
  vs ← VIEWSPECS CONFIRM
  xfreeze(dest, vs);
COMMAND %expunge%
zexpunge =
  "EXPUNGE"
  ent ←
    ( "DIRECTORY"!LL!
      / "ARCHIVE"!LL! <"directory"> )
  CONFIRM
  xexpunge(ent);
COMMAND %edit%
zedit =
  "EDIT"!NOTD! "STATEMENT"!LL! <"at">
  dest ← DSEL("#STATEMENT")

```

```

        xedit(dest);
COMMAND %disconnect%
    zdisconnect =
        "DISCONNECT" "TERMINAL"!L1! CONFIRM
        xdisconnect();
COMMAND %delete%
    zdelete =
        "DELETE"!L1!
        filtre ← FALSE dest ← NULL vs ← NULL
        (
            ( ent ← textent
              <"at"> dest ← DSEL(ent)
            / ent ← structure
              <"at"> dest ← DSEL(ent)
              [filtre ← TRUE <"Filtered:"> vs ← VIEWSPECS/ )
            / ( ent ← ("FILE"!L1! / "ARCHIVE" <"file">)
              dest ← LSEL("#FILE" )
            / ( ent ← "MARKER"
              <"named"> dest ← LSEL(ent) )
            / ( ent ← "ALL"!L1! <"markers">)
            / (( ent ← "MODIFICATIONS"!L1!)
              <"to file"> CONFIRM <"really?"> )
        )
        CONFIRM
        xdelete(ent, dest, filtre, vs);
COMMAND %create%
    zcreate =
        "CREATE" "FILE"!L1!
        filename ← LSEL("#FILE") CONFIRM
        xcreate(filename);
COMMAND %copy%
    zcopy =
        "COPY"!L1!
        vs ← NULL
        level ← NULL
        filtre ← NULL
        param ← NULL
        ( sent ← text1
          copy1
          dent ← sent
          dest ← DSEL(dent)
        / sent ← "TEXT"!L1!
          copy1
          dent ← #"CHARACTER"
          dest ← DSEL(dent)
        / sent ← structure
          copy1
          dent ← #"STATEMENT"
          dest ← DSEL(dent)
          level ← LEVADJ
          copy2
        / sent ← "FILE"!L1!
          dent ← sent
          <"from"> source ← LSEL(sent)
          <"to"> dest ← LSEL(dent)
        / sent ← "DIRECTORY"!L1!

```

```

        copy3
        / sent ← "ARCHIVE"!L1! <"directory">
        copy3
        / sent ← "SEQUENTIAL"
          <"file from"> source ← LSEL("#FILE")
          <"to follow"> dest ← DSEL("#STATEMENT")
          level ← LEVADJ
          dent ← NULL
          [dent ← ("HEURISTIC"!L1! / "JUSTIFIED"!L1! /
            "ASSEMBLER"!L1!)]
        )
        CONFIRM
        xcopy(sent, source, dent, dest, level, filtre, vs);
copy1 =
  <"from"> source ← SSEL(sent)
  <"to follow">;
copy2 =
  [filtre ← TRUE <"Filtered:"> vs ← VIEWSPECS];
copy3 =
  <"from"> source ← LSEL( "#VISIBLE" )
  <"to follow"> dest ← DSEL("#STATEMENT")
  level ← LEVADJ
  dent ← $[diropt];
COMMAND %connect%
zconnect =
  "CONNECT"
  <"to">
  (ent ← ("DISPLAY" / "TTY"!L1!)
  <"Number"> dest ← LSEL("#NUMBER")
  param ← ("INPUT"!L1! <"and Output"> / "OUTPUT"!L1!
  <"Only">) /
  ent ← "DIRECTORY"!L1! dest ← LSEL("#NAME")
  param ← NULL
  [ <"Password"> param ← LSEL("#VISIBLE") ]
  )
  CONFIRM
  xconnect(ent, dest, param);
COMMAND %clear%
zclear =
  "CLEAR"!NOTT! "WINDOW"!L1! <"at">
  dest ← DSEL("#WINDOW") CONFIRM
  xclear(dest);
COMMAND %break%
zbreak =
  "BREAK"!L1!
  ent ← "STATEMENT"!L1!
  <"at"> dest ← DSEL("#CHARACTER")
  level ← LEVADJ
  CONFIRM
  xbreak(ent, dest, level); %should also pass literal%
COMMAND %archive%
zarchive =
  "ARCHIVE" "FILE"!L1!
  filename ← LSEL("#FILE") param ← $["DELETE"!L1! / "DO
  NOT DELETE" / "DEFERRED" / "IMMEDIATE"!L1! / "NOT
  ALLOWED"!L1!] CONFIRM

```

```

        xarchive(file, param);
COMMAND %append%
  zappend =
    "APPEND"!LL!
    ( sent ← text1 % text entities except "TEXT" %
      zappl
      dent ← sent
    / sent ← "TEXT"!LL!
      zappl
      dent ← #"CHARACTER"
    / sent ← structure
      zappl
      dent ← #"STATEMENT"
    )
    dest ← DSEL(dent)
    literal ← LSEL(#"TEXT")
    CONFIRM
    xappend(sent, source, dent, dest, literal);
  zappl =
    <"at">
    source ← SSEL(sent)
    <"to">;
COMMAND %accept%
  zaccept =
    "ACCEPT"!NOTT! "CONNECT"!LL!
    <"from display #"> param ← LSEL(#"NUMBER")
    CONFIRM xaccept( param );
COMMAND %comment%
  zcomment =
    ";"!LL!
    %** xcomment() **% ;
COMMAND %period%
  zperiod =
    "."!LL NOTD!
    xperiod() ;
COMMAND %tab%
  ztab =
    " "!LL NOTD!
    xtab() ;
COMMAND %slash%
  zslash =
    "/"!LL NOTD!
    xslash() ;
COMMAND %bslash%
  zbslash =
    "\"!LL NOTD!
    xbslash() ;
COMMAND %uparrow%
  zuparrow =
    "↑"!LL NOTD!
    xuparrow() ;
COMMAND %linefeed%
  zlinefeed =
    "<LF>!LL NOTD!
    xlinefeed() ;
END.

```

```

% NLS SUPERVISOR COMMANDS %
  SUBSYSTEM subsupervisor
    COMMAND % display subsystem stack %
      zdspss =
        "<"!LL! xsublist();
    COMMAND % display current subsystem stack name %
      zdspcs =
        ">"!LL! xsubcurrent();
    COMMAND
      zquit =
        "QUIT"!LL!
          ( readca() subsys ← NULL
            / subsys ← (nlssubs / "NLS"!LL!) CONFIRM )
          xquit(subsys);
    COMMAND
      zgoto =
        "GOTO"!LL! <"subsystem">
          subsys ← (nlssubs / "TENEX"!LL! )
          CONFIRM xgoto(subsys, FALSE);
    COMMAND
      zexecute =
        "EXECUTE"!LL! <"command in">
          subsys ← nlssubs
          xgoto(subsys, TRUE);
  END.
% USER PROGRAMMING SUBSYSTEM COMMANDS %
  SUBSYSTEM subprograms
    uprogtypes =
      ("CONTENT"!LL! <"analyzer program">/ "SORT"!LL! <"key
        extractor program">/ "SEGENERATOR"!LL! <"program"> );
    COMMAND
      zpset =
        "SET"
          ent ←
            ( "BUFFER"!LL! <"size to">
              source ← LSEL("#NUMBER")
              / "NDDT"!LL! <"control-h">
              source ← NULL )
          CONFIRM xpset(ent, source );
    COMMAND
      zpshow =
        "SHOW"!LL! "STATUS"!LL! <"of programs buffer">
          CONFIRM xpshow( );
    COMMAND
      zpreset =
        "RESET"!LL!
          ent ←
            ( "BUFFER"!LL! <"size">
              / "NDDT"!LL! <"control-h"> )
          CONFIRM xpreset( ent );
    COMMAND
      zprun =
        "RUN"!LL! "PROGRAM"!LL!
          source ← LSEL("#NAME")
          CONFIRM xprun( source );
    COMMAND

```

```

zpload =
  "LOAD"!L1! "PROGRAM"!L1!
  source ← LSEL("#FILE")
  CONFIRM xpload( source );
COMMAND
zpinstitute =
  "INSTITUTE"!L1! "PROGRAM"!L1!
  source ← LSEL("#NAME") <"as">
  ent ← uprogtypes
  CONFIRM xpinstitute( source ent );
COMMAND
zpdeinstitute =
  "DEINSTITUTE"
  ent ← uprogtypes
  CONFIRM xpdeinstitute( ent );
COMMAND
zpdelete =
  "DELETE"!L1!
  ent ←
    ( "ALL"!L1! <"programs in buffer">
      / "LAST"!L1! <"program in buffer"> )
  CONFIRM xpdelete( ent );
COMMAND
zpcompile =
  "COMPILE"!L1!
  sent ←
    ( ("FILE"!L1! / "ASSEMBLER"!L1! <"file"> )
      <"at"> source ← DSEL("#STATEMENT")
      <"using"> compiler ← LSEL("#FILE")
      <"to file"> filename ← LSEL("#FILE")
      / "L10"!L1! <"user program at">
      compiler ← NULL filename ← NULL
      source ← DSEL("#STATEMENT") )
  CONFIRM xpcompile( sent, source, compiler,
    filename );
END.
% JOURNAL SUBSYSTEM COMMANDS %
SUBSYSTEM subjournal
INITIALIZATION %possible reenter%
zjinit =
  xjloaworfil() (
    xjsubinc() <"re-enter last submission?">
    ((readca()/ "YES"!L1!) xjgetworfil() / "NO"!L1!
    jouinit ) /
    jouinit );
jouinit =
  xjzapworfil() %sets reenter flag%
  authfield ← NULL
  clerfield ← NULL
  commfield ← NULL
  distfield ← NULL
  formfield ← NULL
  keywfield ← NULL
  linkfield ← NULL
  numbfield ← NULL
  numdtype ← NULL

```

```

    obsofield ← NULL
    subcfield ← NULL
    submdest ← NULL
    submtype ← NULL
    titlfield ← NULL
    updafield ← NULL;
TERMINATION %close workfile%
    zjterm =
        xjsavworfil() %saves state of variables in workfile%
        xjcloworfil(); %close workfile%
COMMAND %assign%
    zjassign =
        "ASSIGN"!Ll! (
            numbtype ←
                ("JOURNAL"!Ll! / "RINS" / "XDOC"!Ll! /
                 "SPECIAL"!Ll! / "NIC"!Ll!)
                <"numbers -- how many?"> numbfield ←
                LSEL("#NUMBER") /
            numbtype ← "RFC"!Ll! <"number"> CONFIRM
                <"title"> titlfield ← LSEL("#TEXT")
                <"author"> authfield ← LSEL("#TEXT")
                <"distribute to"> distfield ← LSEL("#TEXT")
                <"online document?"> submtype ← answer
                <"show status?"> ((readca()/"YES"!Ll!)
                xjrffcshow()/
                "NO"!Ll! )
        )
        CONFIRM
        xjassign();
COMMAND %author%
    zjauthor =
        "AUTHOR"!Ll!
        authfield ← LSEL("#TEXT")
        CONFIRM;
COMMAND %clerk%
    zjclerk =
        "CLERK"
        clerfield ← LSEL("#WORD")
        CONFIRM;
COMMAND %comments%
    zjcomments =
        "COMMENTS"!Ll!
        commfield ← LSEL("#TEXT")
        CONFIRM;
COMMAND %defer%
    zjdefnumber =
        "DEFER"!Ll! "NUMBER"!Ll! <"assignment">
        numbtype ← #"DEFER"
        CONFIRM;
COMMAND %distribute%
    zjdistribute =
        "DISTRIBUTE"!Ll! <"to"> distfield ← LSEL("#TEXT")
        CONFIRM;
COMMAND %finish%
    zjfinish =
        "FINISH"!Ll! CONFIRM xjfinish( ); %resets re-enter flag%

```



```

        %performs secondary distribution and journal
        submission%
COMMAND %interrogate%
    zjinterrogate =
        "INTERROGATE"!L1! <"for submission"> CONFIRM
        <"select"> (
            subtype ← structure
            <"at"> submdest ← DSEL(subdtype) /
            subtype ← "FILE" (
                ["NAMED" submdest ← LSEL("#FILE")] /
                subtype ← #"WINDOW" submdest ← LSEL("#WINDOW"))
            /
            "MESSAGE" subtype ← #"STATEMENT"
            submdest ← LSEL("#TEXT") /
            subtype ← "HARDCOPY"!L1!
            <"located at"> submdest ← LSEL("#TEXT")
        )
        <"title"> titlfield ← LSEL("#TEXT")
        <"distribute to"> distfield ← LSEL("#TEXT")
        <"show status?"> (("YES"!L1!/CA) xjshow() / "NO"!L1!)
        <"finished?">
            (("YES"!L1!/CONFIRM) xjfinish() / "NO"!L1!);
COMMAND %keywords%
    zjkeywords =
        "KEYWORDS"!L1!
        keywfield ← LSEL("#TEXT")
        CONFIRM;
COMMAND %lock%
    zjlock =
        "LOCK"!L1! "JOURNAL"!L1! <"password">
        keywfield ← LSEL("#VISIBLE")
        CONFIRM xjlock(keywfield, TRUE);
COMMAND %number%
    zjnumber =
        "NUMBER"!L1!
        numdtype ← #"NUMBER"
        numbfield ← LSEL("#TEXT")
        CONFIRM;
COMMAND %obsoletes%
    zjobsobletes =
        "OBSOLETES"!L1! <"documents">
        obsofield ← LSEL("#TEXT")
        CONFIRM;
COMMAND %place%
    zjplace =
        "PLACE" "LINK"!L1! <"at">
        linkfield ← LSEL("#VISIBLE")
        CONFIRM;
COMMAND %print%
    zjprint =
        "PRINT" "HARDCOPY"!L1! %wheel()%
        <"password"> updafield ← LSEL("#VISIBLE") CONFIRM
        xjpriharcop(updafield);
COMMAND %process%
    zjprocess =
        "PROCESS"!L1! "SUBMISSION"!L1! <"form at">

```

```

    formfield ← LSEL("#STATEMENT")
    CONFIRM
    xjprocess(formfield);
COMMAND %rfc number%
    zjrfcnumber =
        "RFC"!L1! <"number">
        numbtype ← #"RFC"
        numbfield ← LSEL("#TEXT")
        CONFIRM;
COMMAND %show%
    zjshow =
        "SHOW"!L1! "STATUS"!L1!
        CONFIRM xjshow( );
COMMAND %subcollections%
    zjsubcollections =
        "SUBCOLLECTIONS"!L1!
        subcfield ← LSEL("#TEXT")
        CONFIRM;
COMMAND %submit%
    zjselect =
        "SELECT"!L1!
        (
            subtype ← structure
            <"at"> submdest ← DSEL(subdtype) /
            subtype ← "FILE" (
                ["NAMED" submdest ← LSEL("#FILE")] /
                subtype ← #"WINDOW" submdest ← LSEL("#WINDOW"))
            /
            "MESSAGE" subtype ← #"STATEMENT"
            submdest ← LSEL("#TEXT") /
            subtype ← "HARDCOPY"!L1!
            <"located at"> submdest ← LSEL("#TEXT") /
            subtype ← "JOURNAL"!L1! <"document">
            submdest ← LSEL("#NUMBER")
        )
        CONFIRM;
COMMAND %title%
    zjtitle =
        "TITLE"!L1!
        titlfield ← LSEL("#TEXT")
        CONFIRM;
COMMAND %unlock%
    zjunlock =
        "UNLOCK"!L1! "JOURNAL"!L1! <"password">
        keywfield ← LSEL("#VISIBLE")
        CONFIRM xjlock(keywfield, FALSE);
COMMAND %updates%
    zjupdates =
        "UPDATES"!L1! <"document(s)"> updafield ← LSEL("#TEXT")
        CONFIRM;

```

END.

```

SUBSYSTEM subidentification
INITIALIZATION %not yet implemented%
    ziinit =
        xsubnotimp();

```

END.

```

SUBSYSTEM subhelp
  INITIALIZATION %not yet implemented%
    zhelpinit =
      xsubnotimp();
  END.
SUBSYSTEM subcalculator
  INITIALIZATION %not yet implemented%
    zcalcinit =
      xsubnotimp();
  END.
FINISH OF NLSLANGUAGE
% IDENTIFICATION SUBSYSTEM COMMANDS %
SUBSYSTEM subjournal
  INITIALIZATION %possible reenter%
    ideinit =
      xiloaidefil() %sets reenter flag%
      capafield ← NULL %capabilities%
      commfield ← NULL %comments%
      cordfield ← NULL %cordinator%
      delifield ← NULL %delivery%
      expafield ← NULL %expand%
      funcfield ← NULL %function%
      identype ← NULL %ident%
      membfield ← NULL %membership%
      hmaidest ← NULL %hardcopy mail address%
      nmaifield ← NULL %NLS mail address%
      smaifield ← NULL %sequential mail address%
      namefield ← NULL %name%
      nhosfield ← NULL %NLS host name%
      orgafield ← NULL %organization%
      phonfield ← NULL %phone%
      rtypfield ← NULL
      shosfield ← NULL
      sorgfield ← NULL
      subcfield ← NULL
      updafield ← NULL;
  TERMINATION %close workfile%
    zjterm =
      xjsavworfil() %saves state of variables in workfile%
      xjcloworfil(); %close workfile%
  COMMAND %assign%
    zjassign =
      "ASSIGN"!LL! (
        numbtype ←
          ("JOURNAL"!LL! / "RINS" / "XDOC"!LL! /
           "SPECIAL"!LL! / "NIC"!LL!)
          <"numbers -- how many?"> numbfield ←
            LSEL("#NUMBER") /
        numbtype ← "RFC"!LL! <"number"> CONFIRM
          <"title"> titlfield ← LSEL("#TEXT")
          <"author"> authfield ← LSEL("#TEXT")
          <"distribute to"> distfield ← LSEL("#TEXT")
          <"online document?"> submtype ← answer
          <"show status?"> (("YES"!LL!/CA) xjrfcshow() /
            "NO"!LL!)
      )

```

```

CONFIRM
  xjassign();
COMMAND %author%
  zjauthor =
    "AUTHOR"!L1!
    authfield ← LSEL("#TEXT")
    CONFIRM;
COMMAND %clerk%
  zjclerk =
    "CLERK"
    clerfield ← LSEL("#WORD")
    CONFIRM;
COMMAND %comments%
  zjcomments =
    "COMMENTS"!L1!
    commfield ← LSEL("#TEXT")
    CONFIRM;
COMMAND %defer%
  zjdefnumber =
    "DEFER"!L1! "NUMBER"!L1! <"assignment">
    numbtype ← #"DEFER"
    CONFIRM;
COMMAND %distribute%
  zjdistribute =
    "DISTRIBUTE"!L1! <"to"> distfield ← LSEL("#TEXT")
    CONFIRM;
COMMAND %finish%
  zjfinish =
    "FINISH"!L1! CONFIRM xjfinish( ); %resets re-enter flag%
    %performs secondary distribution and journal
    submission%
COMMAND %interrogate%
  zjinterrogate =
    "INTERROGATE"!L1! <"for submission"> CONFIRM
    <"select"> (
      subtype ← structure
      <"at"> submdest ← DSEL(subtype) /
      subtype ← "FILE" (
        ["NAMED" submdest ← LSEL("#FILE")] /
        subtype ← #"WINDOW" submdest ← LSEL("#WINDOW"))
      /
      "MESSAGE" subtype ← #"STATEMENT"
      submdest ← LSEL("#TEXT") /
      subtype ← "HARDCOPY"!L1!
      <"located at"> submdest ← LSEL("#TEXT")
    )
    <"title"> titlfield ← LSEL("#TEXT")
    <"distribute to"> distfield ← LSEL("#TEXT")
    <"show status?"> (("YES"!L1!/CA) xjshow() / "NO"!L1!)
    <"finished?">
      (("YES"!L1!/CONFIRM) xjfinish() / "NO"!L1!);
COMMAND %keywords%
  zjkeywords =
    "KEYWORDS"!L1!
    keywfield ← LSEL("#TEXT")
    CONFIRM;

```

```

COMMAND %lock%
  zjlock =
    "LOCK"!L1! "JOURNAL"!L1! <"password">
    keywfield ← LSEL("#"VISIBLE")
    CONFIRM xjlock(keywfield);
COMMAND %number%
  zjnumber =
    "NUMBER"!L1!
    numbtype ← #"NUMBER"
    numbfld ← LSEL("#"TEXT")
    CONFIRM;
COMMAND %obsoletes%
  zjobsobletes =
    "OBSOLETES"!L1! <"documents">
    obsofield ← LSEL("#"TEXT")
    CONFIRM;
COMMAND %place%
  zjplace =
    "PLACE" "LINK"!L1! <"at">
    linkfield ← LSEL("#"VISIBLE")
    CONFIRM;
COMMAND %print%
  zjprint =
    "PRINT" "HARDCOPY"!L1! %wheel()%
    <"password"> updafield ← LSEL("#"VISIBLE") CONFIRM
    xjpriharcop(updafield);
COMMAND %process%
  zjprocess =
    "PROCESS"!L1! "SUBMISSION"!L1! <"form at">
    formfield ← LSEL("#"STATEMENT")
    CONFIRM
    xjprocess(formfield);
COMMAND %rfc number%
  zjrfcnnumber =
    "RFC"!L1! <"number">
    numbtype ← #"RFC"
    numbfld ← LSEL("#"TEXT")
    CONFIRM;
COMMAND %show%
  zjshow =
    "SHOW"!L1! "STATUS"!L1!
    CONFIRM xjshow( );
COMMAND %subcollections%
  zjsubcollections =
    "SUBCOLLECTIONS"!L1!
    subcfield ← LSEL("#"TEXT")
    CONFIRM;
COMMAND %submit%
  zjselect =
    "SELECT"!L1!
    (
      submtype ← structure
      <"at"> submdest ← DSEL(submtype) /
      submtype ← "FILE" (
        ["NAMED" submdest ← LSEL("#"FILE)] /
        submtype ← #"WINDOW" submdest ← LSEL("#"WINDOW"))
    )

```

```
    /
    "MESSAGE" subtype ← #"STATEMENT"
      submdest ← LSEL(#"TEXT") /
    subtype ← "HARDCOPY"!L1!
      <"located at"> submdest ← LSEL(#"TEXT") /
    subtype ← "JOURNAL"!L1! <"document">
      submdest ← LSEL(#"NUMBER")
  )
  CONFIRM;
COMMAND %title%
  zjtitle =
    "TITLE"!L1!
    titlfield ← LSEL(#"TEXT")
  CONFIRM;
COMMAND %unlock%
  zjunlock =
    "UNLOCK"!L1! "JOURNAL"!L1! <"password">
    keywfield ← LSEL(#"VISIBLE")
    CONFIRM xjunlock(keywfield);
COMMAND %updates%
  zjupdates =
    "UPDATES"!L1! <"document(s)"> updafield ← LSEL(#"TEXT")
  CONFIRM;
END.
```