

RECORDS

(MLK) SRECORDS
(MLK) SRECORDS

(MLK) SRECORDS
(MLK) SRECORDS

(MLK) SRECORDS
(MLK) SRECORDS

(MLK) SRECORDS
(MLK) SRECORDS

(MLK) SRECORDS
(MLK) SRECORDS

< NLS. SRECORDS.NLS;2, >, 11-OCT-74 10:45 KEV ;;; (MICHAEL,
 SRECORDS.NLS;2,), 16-JUL-74 09:16 EKM ;
 FILE srecords % L10 <REL-NLS>srecords %% (L10,)
 (rel-nls,srecords.rel,) %

% records for use with the directory commands %	0263
(dlinfo) RECORD % information to be listed for Directory commands %	02
	03
dlidlt[2], % undeleted or deleted only or both %	04
% 0 - undeleted only; 1 - deleted only; 2 or 3 - both %	05
dlifrf[1], % list for file group (not directory) %	06
dliacc[1], % list account of file %	07
dliars[1], % list archive status %	08
dliart[1], % list archive tapes %	09
dlidmt[1], % list dump tape %	010
dlidfr[1], % list default number of versions to keep %	011
dlilwr[1], % list last writer %	012
dlibyt[1], % list length and bytesize %	013
dlimis[1], % list miscellaneous information about file %	014
dlinrw[1], % list number of reads and writes %	015
dliprt[1], % list protection %	016
dliisiz[1], % list size in pages %	017
dlinvr[1], % don't list vesion numbers %	018
dlinex[1], % don't list extension names %	019
dlitar[2], % list [time and] date of archiving %	020
% 1 - date only; 2 or 3 - time and date %	021
dlitcr[2], % list [time and] date of creation this version	022
%	
% 1 - date only; 2 or 3 - time and date %	023
dlitdm[2], % list [time and] date of last dump %	024
% 1 - date only; 2 or 3 - time and date %	025
dlitov[2], % list [time and] date original version created	026
%	
% 1 - date only; 2 or 3 - time and date %	027
dlitrd[2], % list [time and] date of last read %	028
% 1 - date only; 2 or 3 - time and date %	029
dlitwr[2], % list [time and] date of last write %	030
% 1 - date only; 2 or 3 - time and date %	031
(dlgrp) RECORD % grouping information for Directory commands %	032
dlgrvr[1], % group in reverse order %	033
dlgacc[1], % group by accounts %	034
dlgars[1], % group by archive status %	035
dlgdar[1], % group by archive date %	036
dlgart[1], % group by archive tapes %	037
dlgbyt[1], % group by bytesize %	038
dlgdcr[1], % group by creation date %	03
dlgdlt[1], % group by deleteion status %	040
dlgddm[1], % group by dump date %	041
dlgdmt[1], % group by dump tapes %	042
dlgdfr[1], % group by file retention specs %	043
dlglwr[1], % group by last writer %	044
dlgdov[1], % group by creation date of original version %	045
	045
dlgprt[1], % group by protection %	046
dlgdrd[1], % group by read date %	047
dlgdwr[1], % group by write date %	048
(dlsort) RECORD % sorting information for Directory commands %	049

dacent[12],	% SEQUENCE character count for dacsp t-pointer%	0115
dacsize[2],	% FORMATTER character size%	0116
dancos[2],	% FORMATTER horizontal increment%	0117
dasgcs[2],	% FORMATTER horizontal increment%	0118
dafrzl[18],	% SEQUENCE a frozen statement chain for this da%	0119
		0120
daind[18],	% FORMATTER indentation per level%	0121
damin[18],	% FORMATTER max indentation%	0122
		0123
dalftjst[18],	% ??? left-justify branch, plex for indentation	
cfi%		0124
dacrow[18],	% FORMATTER current row count%	0125
		0126
damrow[18],	% FORMATTER max row count%	0127
dacol[18],	% FORMATTER current column count%	0128
		0129
damcol[18],	% FORMATTER max column count%	0130
dahinc[18],	% FORMATTER horizontal increment%	0131
		0132
davinc[18],	% FORMATTER vertical increment%	0133
dano[18],	% FORMATTER horizontal increment%	0134
		0135
dasgh[18],	% FORMATTER horizontal increment%	0136
dafont[18],	% FORMATTER default font%	0137
		0138
dalink[18],	% ??? a link-jump stack for this da%	0139
dahandle[9],	% FRONTEND system handle for this display area%	0140
dalssup[9],	% FRONTEND next is number to suppress for literal	
feedback%		0141
		0142
daleft[18],	% FRONTEND left boundry of da (raster units)%	0143
daright[18],	% FRONTEND right boundry of da (raster units)%	0144
		0145
datop[18],	% FRONTEND top boundry of da (raster units)%	0146
dabottom[18],	% FRONTEND bottom boundry of da (raster units)%	0147
% (0,0) is upper left corner%		0148
		0149
dalsrt[18],	% FORMATTER LSRT starting address%	0150
dalsz[9],	% FORMATTER LSRT size for this da%	0151
dapstf[5],	% file id, previous value for this da%	0152
dashrinkent[2],	% FORMATTER LSRT allocation shrink algorithm cnt%	
		0153
daaxis[1],	% this da entry is in use%	0154
daempty[1],	% da-has-no-display flag%	0155
		0156
datap0[36],	% FORMATTER first tab position word%	0157
datap1[36],	% FORMATTER second tab position word%	0158
datap2[36],	% FORMATTER third tab position word%	0159
		0160
dacacode [18],	% SEQUENCE address of content analyzer program	
-- or 0 %		0161
dauseqod [18],	% SEQUENCE address of sequence generator program	
-- or 0 %		0162
		0163
dahandle[18],	% FRONTEND 0 or handle for linked da%	0164

```

dalsadb[18],      % FRONTEND address of lsid pit table %      0165
                                                           0166
daukeycod[18],   % FRONTEND address of sort key extractor      0167
program -- or 0 %
daauxiliary[1], % FRONTEND a da used for other purposes than
displaying.
                (as done in TNL5) Calculator, for example, uses such
an area.%
daprozen[1],    % FRONTEND frozen boundaries flag (calculator)% 0169
daseq[1],       % FRONTEND sequential display area%           0170
dalneighbor[3], % FRONTEND True if window has a left
neighbor%
darneighbor[3], % FRONTEND True if window has a right
neighbor%
datneighbor[3], % FRONTEND True if window has a top neighbor% 0172
                                                           0173
dabneighbor[3], % FRONTEND True if window has a bottom
neighbor%
dasuppress[1];  % FRONTEND the display image is suppressed%   0175
DECLARE EXTERNAL dal = 22, damax = 7;
                %if you enlarge damax, yo must enlarge
                dalneighbor,...,dabneighbor also%
(frrentry) RECORD %file return ring entry%                   0178
frsrring[18],   %address of statement return ring%           0179
frexis[1];     %true if exists%                               0180
DECLARE EXTERNAL frrelen = 1; %length of file return ring
header%
(frheader) RECORD %file return ring header%                   0182
frhexis[1],    %true if valid frr header%                     0183
frhtop[6],     %index to top of ring%                          0184
frnlast[6];    %index to last entry%                           0185
DECLARE EXTERNAL frrhlen = 1; %length of file return ring
header%
(gbnst) RECORD % word returned by getab from HOSTN table %    0187
hstnm1[18],    % index into HSTNAM table %                     0188
hstnmn[3],     % host number for this entry %                 0189
hstnmf[10]     % flags for this host %                         0190
;
(lsrt) RECORD %line segment reference table entry%           0192
rtbps[36],     %starting byte pointer to text - increment and
load byte gets 1st byte of string%                             0193
rtbpe[36],     %ending byte pointer to text - plus 1 char
further%
rtstid[36],    %stid of statement being displayed%           0195
rtxl[18],     %x-cord position at start of line segment%     0196
rtx2[18],     %last character, or last plus 1%                0197
               %rtx2 also used as stop coord for formatter% 0198
rty[18],      %y-cord%                                         0199
rtlev[6],     %level of this statement%                         0200
rtccnt[12],   %ordinal relative to start of stmt of first char
of
line segment%
rthinc[14],   %horizontal increment%                           0202
rtlsid[9],    %system handle (jsys return) %                   0203
rtexis[1],   %entry exists flag %                               0204
rtsrce[4],   %source of display data (code)%                   0205
rtfont[5],   %font of this entry%                               0206

```


STGMBT

STGMGT
STGMGT

STGMGT
STGMGT

STGMGT
STGMGT

STGMGT
STGMGT

STGMGT
STGMGT

STGMGT
STGMGT

STGMGT
STGMGT

STGMGT
STGMGT

```

<NLS>STGMGT.NLS;38, 28-MAR-74 14:44 HGL ;
FILE stgmt % L10 <Rel-NLS>stgmt %% (L10,) (rel-nls,stgmt.rel,) %
                                0294
DECLARE %zone format%
                                0268
    wrdsused = 0, blkmax = 1, blkmin = 2, blkmx1 = 3, blkused = 4,
                                0269
    zonend = 5, zfrelist = 6;
                                0270
%storage allocator%
                                02
(getblk) PROCEDURE %allocate a blk of "size" words in "zone"%
                                03
    (size, %number of words desired in blk%
                                04
    zone); %address of free storage zone%
                                05
%-----%
                                06
%allocates a blk in zone of at least size+bhl words. The blk
consists of a blk header and at least sizee words of zeroes. The
address returned is that of the beginning of the blk (the blk
header). There are at least size usable words at BlockAddress +
bhl. If no blk can be allocated, a zero is returned,%
                                07
%-----%
                                08
LOCAL
                                09
    blk, %address of blk%
                                010
    r, %remainder temp%
                                011
    zero, %temp%
                                012
    end, %tem=%
                                013
    index; %index into zones free lists for blocks of
                                014
        correct size%
REF blk, zone;
                                015
BUMP cgetblk;
                                016
IF zone[blkmin] = zone[blkmax] THEN
                                017
    IF zone[blkmax] >= size+bhl THEN
                                018
        index ← 0
                                019
    ELSE
                                020
        RETURN(0)
                                021
ELSE
                                022
    BEGIN %round up to nearest available size%
                                023
        index ← (size + bhl + zone[blkmin] - 1)/zone[blkmin] - 1;
                                024
        IF index NOT IN [0, zone[blkmx1]] THEN RETURN(0);
                                025
    END;
                                026
IF NOT zone[zfrelist + index] AND (zone[blkmin] = zone[blkmax] OR
NOT replenish(index, &zone)) THEN
                                027
    %no free blocks of that index or larger%
                                028
    RETURN(0);
                                029
&blk ← zone[zfrelist + index];
                                030
%unlink it%
                                031
    unlink(&blk, &zone);
                                032
%zero the blk%
                                033
    zero ← &blk + bhl;
                                034
    end ← &blk + blk.sysblength;
                                035
    DO (zero) ← 0 UNTIL (zero ← zero + 1) >= end;
                                036
    blk.blklength ← size+bhl; %for caller's use only%
                                037
    BUMP zone[blkused];
                                038
    zone[wrdsused] ← zone[wrdsused] + blk.sysblength;
                                039
    RETURN(&blk);
                                040
END.
                                041
                                042
(freeblk) PROCEDURE %de-allocate a block in zone%
                                043
    (blk, %address of block to be de-allocated%
                                044

```

```

zone); %address of free storage zone%                                045
%-----%                                                            046
%the block addressed by blk is marked as free and linked back into
the appropriate free list in zone,                                  047
NOTE: Freeblk will return FALSE if it finds a bad block address.
you may run out of space if you never issue another good free to
replace a bad call.%                                             0385
%-----%                                                            048
LOCAL                                                                049
    previous, %address of previous block in the zone%              050
        %for combining free blocks%                                051
    next; %address of next block in zone%                            052
        %for combining free blocks%                                053
REF blk, previous, next, zone;                                       054
BUMP cfreeblk;                                                       055
IF &blk NOT IN [&zone, zone/zonend/] THEN                             056
    RETURN(FALSE);                                                  058
zone/wrdsused/ < zone/wrdsused/ - blk.sysblength;                   061
IF zone/blkmin/ NOT= zone/blkmax/ THEN                                062
    BEGIN                                                            063
        &next < &blk + blk.sysblength;                             064
        IF &next < zone/zonend/ THEN                                065
            BEGIN                                                    066
                IF next.blkfree AND (blk.sysblength + next.sysblength <=
zone/blkmax/) THEN                                                 067
                    BEGIN %combine them%                             068
                        unlink(&next, &zone);                       069
                        blk.sysblength < blk.sysblength + next.sysblength;
070
                        &next < &blk + blk.sysblength;              071
                    END;                                             072
                END;                                                 073
            IF blk.blkprevfree AND (blk.sysblength + [&blk - 1] <=
zone/blkmax/) THEN                                                 074
                BEGIN %combine them%                                 075
                    %last word of previous block contains the length of that
block%                                                             076
                    &previous < &blk - [&blk - 1];                 077
                    unlink(&previous, &zone);                       078
                    previous.sysblength < previous.sysblength + blk.sysblength;
079
                    &blk < &previous;                                080
                END;                                                 081
            END;                                                     082
        linkup(&blk, &zone);                                         083
        BUMP DOWN zone/blksused/;                                    084
        RETURN(TRUE);                                               085
    END.

(makezone) PROCEDURE %make a zone out of a block%                   086
    (zone, %address of a block to be initialized as a zone%       088
    max, %maximum size of a block -- integral multiple of min%    089
    min, %minimum size block%                                       090
    size); %size of the zone%                                         091
%-----%                                                            092
%This routine takes the address of a free storage block and
constructs an appropriate header so that it will serve as a free

```

storage zone? If min and max are the same, then the zone will contain only fixed sized block and a single free list -- no combining of adjacent free blocks will be attempted by freeblk%

```

%-----%
LOCAL
  blk,      %address of a block within the zone%           093
  lastfreelist, %displacement to the lastfreelist free list 094
  for this zone%                                           095
  frstblk, %address of first block in the zone%           096
  base,     %address of a block within the zone%           097
  length;  %length of rest of block%                       098
REF zone, blk, frstblk, base;                               099
                                                         0100
% zero out this whole block before starting %             0101
  FOR length ← (size-1) DOWN UNTIL < 0 DO [&zone + length] ← 0; 0102
                                                         0387
BUMP cmakezone;                                           0103
max ← ((max+min-1)/min)*min;                               0104
zone/blkmax] ← max;                                       0105
zone/blkmin] ← min;                                       0106
zone/blkmxi] ← (max/min) - 1;                             0107
lastfreelist ← zone/blkmxi] + zfreelist;                  0108
&frstblk ← &base ← &zone + lastfreelist + 1;             0109
length ← ((size - (&frstblk - &zone))/min)*min;         0110
zone/zonend] ← &frstblk + length;                         0111
UNTIL length < max DO                                     0112
  BEGIN                                                  0113
    &blk ← &base + max;                                   0114
    base.sysblength ← max;                               0115
    [&blk-1] ← max;                                     0116
    base.blkfree ← base.blkprevfree ← TRUE;             0117
    base/blksuc] ← zone[lastfreelist] := &base;         0118
    base/blkprd] ← 0;                                    0119
    IF &base ← base/blksuc] THEN                          0120
      base/blkprd] ← zone[lastfreelist];                 0121
    &base ← &blk;                                        0122
    length ← length - max;                               0123
  END;                                                  0124
frstblk.blkprevfree ← FALSE;                             0125
IF length > 0 THEN                                       0126
  BEGIN                                                0127
    blk.sysblength ← length;                             0128
    IF length < min THEN                                0129
      blk.blkfree ← blk.blkprevfree ← FALSE            0130
    ELSE                                               0131
      BEGIN                                           0132
        blk.blkfree ← blk.blkprevfree ← TRUE;         0133
        blk/blksuc] ← zone[zfreelist + length/min - 1] := &blk; 0134
        blk/blkprd] ← 0;                               0135
        IF &base ← blk/blksuc] THEN                    0136
          base/blkprd] ← &blk;                          0137
          [&blk + blk.sysblength -1] ← length;        0138
        END;                                           0139
      END;                                           0140
    zone/blkused] ← zone/wrdsused] ← 0;

```



```

END. 0187
0188
(breakup) PROCEDURE %break up blk and put in index free list% 0189
  (blk, %address of block to be subdivided% 0190
  zone, %address of free storage zone% 0191
  index); %free list index to be replenished% 0192
  %-----% 0193
  %This routine creates a number of blocks (by breaking up blk) of
  appropriate size for the free list indicated by index and links
  them into that free list in zone.% 0194
  %-----% 0195
LOCAL 0196
  length, %length of smaller blocks% 0197
  base, %temp, used in subdividing blk% 0198
  i, %loop control variable% 0199
  r; %temp, used to set up i% 0200
REF blk, base, zone; 0201
length ← (index+1)*zone/blkmin; 0202
DIV blk.sysblength/length, i, r; 0203
IF NOT r THEN BUMP DOWN i; 0204
UNTIL (i ← i -1) < 0 DO 0205
  BEGIN 0206
    &blk ← (&base ← &blk) + length; 0207
    blk.sysblength ← (base.sysblength := length) - length; 0208
    linkup(&base, &zone); 0209
  END; 0210
  &blk ← (&base ← &blk) + blk.sysblength; 0211
  IF base.sysblength >= zone/blkmin THEN 0212
    linkup(&base, &zone) 0213
  ELSE base.blkfree ← FALSE; 0214
RETURN; 0215
END. 0216
0217
(unlink) PROCEDURE %unlink a block from its free list% 0218
  (blk, %address of block to be unlinked from its free list% 0219
  zone); %address of a free storage zone% 0220
  %-----% 0221
  %this routine unlinks blk from the appropriate free list in zone%
  %-----% 0222
LOCAL 0223
  next, %address of next block in zone% 0224
  suc, %address of block pointed to by successo link% 0225
  prd; %address of block pointed to by predecessor link% 0227
REF next, blk, suc, prd, zone; 0228
BUMP cunlink; 0229
blk.blkfree ← FALSE; 0230
IF (&next ← &blk + blk.sysblength) < zone/zonend THEN 0231
  next.blkprevfree ← FALSE; 0232
IF &suc ← blk/blksuc THEN 0233
  suc[blkprd] ← blk[blkprd]; 0234
IF &prd ← blk/blkprd THEN 0235
  prd[blksuc] ← blk[blksuc] := 0 0236
ELSE 0237
  zone[zfrelst + blk.sysblength/zone/blkmin - 1] ← blk/blksuc/
  := 0; 0238

```

```

blk/blkprd/ ← 0;                                0239
RETURN;                                          0240
END.                                             0241
                                                0242
(linkup) PROCEDURE %linkup blk into appropriate free list% 0243
  (blk, %address of block to be linked into free list% 0244
  zone); %address of a free storage zone%        0245
  %-----%                                       0246
  %this routine links blk into the beginning of the appropriate free
  list in zone%                                   0247
  %-----%                                       0248
  LOCAL                                          0249
    next, %address of next block in zone%        0250
    index; %free list index for blk%            0251
  REF next, blk, zone;                          0252
  BUMP clink;                                    0253
  blk.blkfree ← TRUE;                            0254
  index ← blk.SYSBLENGTH/zone[blkmin]-1 + zfirelst; 0255
  blk/blksuc/ ← zone[index] := &blk;            0256
  blk/blkprd/ ← 0;                               0257
  [&blk+blk.sysblength-1] ← blk.sysblength;    0258
  IF (&next ← &blk+blk.sysblength) < zone[zonend] THEN 0259
    next.blkprevfree ← TRUE;                    0260
  IF &blk ← blk/blksuc/ THEN                    0261
    blk/blkprd/ ← zone[index];                 0262
  RETURN;                                       0263
  END.                                          0264
                                                0265
%support routines to control storage allocator% 0298
%string and array handling%                    0328
(getstring) PROCEDURE (length, zone);          0329
  %allocate a block in "zone" of "length" characters plus
  block header plus string header. Initialize the string to
  empty. Return string address.                0330
  length      actual useable length of string desired 0331
  zone        address of zone with free space in it 0332
  SIGNALS if no more room in that block%      0361
  LOCAL blockaddress;                          0334
  REF blockaddress;                             0340
  &blockaddress ← getblk ((length + 4)/5 + bnl
  + %string header% 1, zone);                  0335
  IF NOT &blockaddress THEN SIGNAL (errsig, $"Fatal storage
  shortage error");                             0342
  &blockaddress ← &blockaddress + bnl;          0354
  blockaddress.M ← length;                      0339
  *plockaddress* ← NULL;                       0341
  RETURN (&blockaddress);                      0343
  END.                                          0344
                                                0345
  (getarray) PROCEDURE (length, zone);         0390
  %allocate an array in "zone" of "length" words plus block
  header. Initialize the array to empty. Return array address.
  length      actual useable length of array desired 0391
  zone        address of zone with free space in it 0392
  SIGNALS if no more room in that block%      0393
  0394

```



```

LOCAL blockaddress;                                0395
REF blockaddress;                                  0396
&blockaddress ← getblk (length + bhl, zone);       0397
IF NOT &blockaddress THEN SIGNAL (errsig, S"Fatal storage
shortage error");                                  0398
RETURN (&blockaddress ← &blockaddress + bhl);     0399
END.                                                0400
                                                    0401
(freestring) PROCEDURE (straddr, zone); % also for arrays. % 0347
%Deallocate a string at "straddr" within "zone". Returns FALSE
if no such block in "zone"%                        0348
RETURN(freeblk (straddr - bhl %string header%, zone)); 0349
END.                                                0350
                                                    0351
%special display storage problems%                 0299
(makisrt) PROCEDURE (zone, segs, da);              0362
%allocate a new block in "zone" to be used as a line segment
reference table                                     0363
updates da record                                  0364
returns FALSE if there was no more room in "zone", ELSE address
of first useable location (past header)           0365
                                                    0366
zone          address of start of zone containing free space 0367
segs          number of segs possible in this display area  0368
da           address of display area%              0369
                                                    0370
LOCAL lsrtaddress, words;                          0371
REF da;                                             0372
IF da.dalsrt OR nmode # fulldisplay THEN RETURN (da.dalsrt);
%no-op if already exists%                          0373
words ← MIN(segs * lsrtl, 777B);                  0374
lsrtaddress ← getblk (words, zone );              0375
IF lsrtaddress THEN                                0376
    BEGIN                                          0377
        lsrtaddress ← lsrtaddress + bhl; %block header%    0378
        da.dalsz ← words/lsrtl;                      0379
        da.dalsrt ← lsrtaddress;                    0380
        <DSPGEN, clrall>(&da, FALSE);%clear - no strings to free%
                                                    0381
    END;                                           0382
RETURN (lsrtaddress);                              0383
END.                                                0384
                                                    0384
(freelsrt) PROCEDURE (zone, da):                  0319
%deallocates a line segment reference table and updates da
record                                              0321
zone          zone address                          0322
da           address of display area record%       0323
REF da;                                             0324
IF NOT da.dalsrt OR nmode # fulldisplay THEN RETURN; %no-op if
none exists%                                       0355
freeblk(da.dalsrt - bhl, zone);                   0325
da.dalsz ← da.dalsrt ← 0;                          0326
RETURN END.                                         0327
                                                    0288

```

FINISH

SYNTAX

(MLK) SYNTAX
(MLK) SYNTAX

(MLK) SYNTAX
(MLK) SYNTAX

(MLK) SYNTAX
(MLK) SYNTAX

(MLK) SYNTAX
(MLK) SYNTAX

(MLK) SYNTAX
(MLK) SYNTAX

(M
(M

```

< NLS, SYNTAX.NLS;172, >, 30-OCT-74 17:28 DSM ;;;;
FILE nlslanguage % CML <rel-nls>SYNTAX %% (CML,) (rel-nls,SYNTAX.rel,)
%
% DECLARATIONS %
  DECLARE PARSEFUNCTION
    answ,          % reads answer construct %
    answer,        % for questions - returns 0/1 %
    lookstat,      % looks for s and shows status for substitute
    command%
    sp,            % reads next char, TRUE if space %
    readconfirm,   % reads next char if ca %
    myrdconfirm,   % reads next char if ca; with variations for
    HELP %
    readbug,       % reads next char if BUG %
    readoption,    % TRUE if next char is optchar %
    readrepeat,    % TRUE if next char is repeat %
    lookansw,      % TRUE if next char is Y/CA %
    sublookansw,   % TRUE if next char is Y/CA with variations for
    substitute command%
    lookconfirm,   % TRUE if next char is CA/REPEAT/INSERT %
    lookbug,       % TRUE if next char is BUG %
    mylookbug,     % TRUE if next char is BUG; with variations for
    HELP %
    lookback,      % TRUE if next char is <; for HELP %
    looknum,       % TRUE if next char is a number %
    lookrpt,       % TRUE if next char is rpt; for HELP %
    prmptlssel,    % prompt for an LSEL %
    prmptdssel,    % prompt for an DSEL %
    prmptsssel,    % prompt for an SSEL %
    notca;        % reads next char, TRUE iff not CA char %
  DECLARE EXTERNAL zinsstatement, textent, textl, structure, cshmode,
  nissubs, synsubs, allsubs, protl, devopt, prtgrp, qdhelp, qthelp,
  hchk, jmpcoms;
  DECLARE EXT-KEYWORD % so only one copy exists in system These
  keywords are defined as external strings in CONST. %
    % STRUCTURAL ENTITIES %
      "BRANCH",
      "GROUP",
      "PLEX",
      "STATEMENT",
    % TEXTUAL ENTITIES %
      "CHARACTER",
      "INVISIBLE",
      "LINK",
      "NUMBER",
      "PASSWORD",
      "TEXT",
      "VISIBLE",
      "WORD",
    % MISC. ENTITIES %
      "FILE",
      "OLDFILELINK",
      "NEWFILELINK",
      "NAME",
      "RETURN",
      "FILERETURN",

```

```

"IDENT",                                03639
"IDENTLIST",                             03640
"EDGE",                                   03381
"MARKER",                                 053
"REST", % Used in HELP %                 03983
"BACK", % Used in HELP %                 03987
"MENU", % Used in HELP %                 03984
"HLPCOM", % Used in HELP %               03985
"ONTLQ", % Used in HELP %                03986
DECLARE EXTERNAL % not defined here see (nls,pdata,) % 054
  allsubs,                               03507
  nlssubs;                                055
% COMMON RULES %                           03
% ENTITY DEFINITIONS %                     04
  editentity = textent / structure;       05
% TEXT ENTITY DEFINITIONS %                06
  textent = text1 / "TEXT"!LL! / "LINK"!LL! / "NUMBER"!LL!;
  text1 = "CHARACTER"!LL! / "WORD"!LL! / "VISIBLE"!LL! /
  "INVISIBLE"!LL!;                        08
% STRUCTURE ENTITY DEFINITIONS %           09
  structure = "STATEMENT"!LL! / notstatement; 010
  notstatement = "GROUP"!LL! / "BRANCH"!LL! / "PLEX"!LL! ; 011
% SWITCH %                                 014
  switch = ("ON"!LL 1!/"OFF"!2!);         015
% DEVICE TYPES %                           01832
  devopt =                                 01833
  ( "TASKER"                               0145
  / "TI"!LL! <"Terminal">                 0146
  / "NVT"!LL!                               01839
  / "LINEPROCESSOR"!LL!                     01834
  / "IMLAC"!LL!                              01835
  / "EXECUPORT"!LL!                          0147
  / "33-TTY"!LL 33!                           0148
  / "35-TTY"!LL 35!                           0149
  / "37-TTY"!LL 37! );                       0150
% NLS EDITOR COMMANDS %                    056
SUBSYSTEM nlseditor KEYWORD "BASE"        057
COMMAND %verify%                            058
  zverify =                                 059
  "VERIFY"!LL! "FILE"!LL! CONFIRM xverify( ) ; 060
COMMAND %update%                             061
  zupdate =                                 062
  "UPDATE"!LL! "FILE"!LL!                  063
  namfil ← NULL                             064
  ent ← #"NEW"                              065
  (readconfirm()                             01430
  / ent ←                                     01431
  ( "OLD"!LL! <"version">                   067
  / "COMPACT"!LL!                             068
  / "RENAME"!LL! <"filename">               069
  namfil ← LSEL( #"NEWFILELINK" ) )         070
  CONFIRM)                                   071
  xupdate( ent, namfil ) ;                  072
COMMAND %undelete%                           073
  zundelete =                               074
  "UNDELETE"                                075

```

```

ent ←
  ( ( "FILE"!LL!
    %/ "ARCHIVE"!LL! <"file">%
    ) namfil ← LSEL( #"OLDFILELINK" )
  / "MODIFICATIONS"!LL! <"to file"> namfil ← NULL
  )
CONFIRM
xundelete( ent, namfil ) ;
COMMAND %trim%
ztrim =
  "TRIM" "DIRECTORY"!LL!
  <"no. versions to keep"> param ← LSEL( #"NUMBER" )
  CONFIRM <... "really?">
  CONFIRM
  xtrim( param ) ;
COMMAND %transpose%
ztranspose =
  "TRANPOSE"!LL!
  sent ← editentity
  <"at"> source ← DSEL( sent )
  <"and"> dent ← sent
  dest ← DSEL( dent )
  vs ← NULL filtre ← FALSE
  [filtre ← TRUE <"Filtered:"> vs ← VIEWSPECS]
  CONFIRM
  xtranspose( sent, source, dent, dest, filtre, vs ) ;
COMMAND %substitute%
zsubstitute =
  "SUBSTITUTE"!LL!
  filtre ← FALSE vs ← NULL param ← NULL
  sent ← textent
  <"in">
  ( readoption()
    filtre ← TRUE <"Filtered:"> vs ← VIEWSPECS
    dent ← structure
    / dent ← structure )
  <"at"> dest ← DSEL( dent )
  % collect pairs of entities of type sent %
  subsnit( sent ) % initialize collection stack %
  PERFORM subst1 UNTIL
  ( <"Finished?">
    ( lookstat( sent ) <"Finished?"> sublookansw()
      / answ() )
  )
  CONFIRM
  xsubstitute( sent, dent, dest, param, filtre, vs );
subst1 =
  CLEAR
  subldsp( sent , TRUE ) %display =+new etc > %
  s1 ← LSEL( sent )
  %sub2dsp( sent, s1 ) feedback param if bugged %
  subldsp( sent, FALSE ) %display <for old etc> %
  s2 ← LSEL( sent )
  %sub2dsp( sent, s2 ) feedback param if bugged %
  param ← subpsave( s1, s2 ) % stash the pairs away %;
COMMAND %stop%

```

076
 077
 078
 01845
 079
 01846
 081
 082
 083
 084
 085
 086
 01497
 088
 089
 090
 091
 092
 093
 094
 095
 096
 097
 098
 099
 0100
 0101
 03254
 03255
 03256
 03257
 03258
 03259
 03260
 03261
 03262
 03263
 03264
 03266
 03267
 04256
 04257
 04258
 04259
 03268
 03269
 03270
 03265
 03271
 03272
 03273
 03274
 03275
 03276
 03277
 0124

```

zstop =                                0125
  "STOP" "RECORD"!LL! <"of session"> CONFIRM 0126
  xstop() ;                             0127
COMMAND %sort%                          0134
zsort =                                  0135
  "SORT"                                  0136
  dent ← notstatement                    0137
  dest ← DSEL( dent )                   0138
  CONFIRM                                0139
  xsort( dent, dest ) ;                 0140
COMMAND %simulate%                       0141
zsimulate =                              0142
  "SIMULATE" <"terminal type">         0143
  ent ← devopt                           0144
  CONFIRM                                0151
  xsimulate( ent ) ;                   0152
COMMAND %show%                           0153
zshow =                                  0154
  "SHOW"                                  0155
  dent ← NULL                            01688
  filtre ← NULL                          01689
  param ← NULL                           01703
  ( "FILE"!LL! ent ← #"FILE"           0157
    param ←                               0158
      ( ?\STATUS"!LL!                   0159
        / "DEFAULT"!LL! <"directory for links"> 0160
        / "MODIFICATIONS"!LL! <"status">       0162
        / "RETURN"!LL! <"ring">               0163
        / "SIZE" )                          0164
    / "RETURN"!LL! <"ring">                 03535
    ent ← #"RETURN"                        03536
    / "MARKER"!LL! <"list">                 0161
    param ← #"MARKER" ent ← #"FILE"        03315
  %/ "ARCHIVE"!LL! <"directory"> ent ← #"ARCHIVE" 0165
    param ← (readconfirm() / LSEL( #"NAME" ) )% 0166
    %CAN WE DO THIS??%                    0167
  / "DIRECTORY"!LL! ent ← #"DIRECTORY"     01690
    <"of">                                  01691
    ( readconfirm()                       01704
      / param ← LSEL( #"DIRECTORY" ) CONFIRM 01705
    )                                       01706
    dent ← seqbinit()                     01692
    ( readoption() CLEAR <"opt:">         01694
      PERFORM diropt UNTIL( <"Finished?"> lookansw() ) 01695
    / DUMMY) % dummy case %              01696
  / "DISK" <"space status"> ent ← #"DISK" 0171
    param ← NULL                          0172
  / "NAME"!LL! <"delimiters for statement at"> ent ← #"NAME"
    param ← DSEL( #"STATEMENT" )          0173
  / "VIEWSPECS"!LL! <"status"> ent ← #"VIEWSPECS" 0174
    param ← NULL                          0175
    (param ← "VERBOSE"!LL! )             0176
  CONFIRM                                0177
  xshow( ent, param, dent, filtre ) ;    0178
COMMAND %set%                             0179
                                           0180

```

```

zset = 0161
"SET" 0182
dest ← NULL param ← NULL param2 ← NULL param3 ← NULL 0183
( 0184
    ent ← "CHARACTER"!NOTT! 0191
      <"size for window to"> param ← LSEL( #"NUMBER" ) 0192
/ ent ← "EXTERNAL"!LL! <"Names Link File To:"> 03171
  param ← LSEL( #"LINK" ) 03172
/ ent ← "CONTENT"!LL! <"pattern"> 0195
  param ← 0196
    ( "TO"!LL! 0197
      param2 ← LSEL( #"CHARACTER" ) 0198
    / switch ) 0199
/ ent ← "LINK"!LL! 0200
  <"default for file to directory"> 0201
  param2 ← LSEL( #"NAME" ) 0202
/ ent ← "NAME"!LL! 0203
  <"delimiters in"> param ← structure 0204
  <"at"> dest ← DSEL( param ) 0205
  CLEAR 0206
  <"left delimiter"> param2 ← LSEL( #"CHARACTER" ) 0207
  <"right delimiter"> param3 ← LSEL( #"CHARACTER" ) 0208
/ "NLS" <"protection for file"> 03781
  ent ← ( "PRIVATE"!LL! / "PUBLIC" ) 03064
/ ent ← "TENEX"!LL! <"protection for file named"> 03780
  param ← LSEL( #"OLDFILELINK" ) 01714
  param2 ← seqbinit() 01715
  ( "SET"!LL! <"to"> 01716
    param3 ← #"SET" 02201
    seqbadd( param2, param3 ) 02202
    param3 ← LSEL( #"NUMBER" ) 01717
    seqbadd( param2, param3 ) 01718
  / "ALLOW"!LL! 01719
    param3 ← #"ALLOW" 01720
    seqbadd( param2, param3 ) 01721
    prtgrp 02199
    PERFORM protl UNTIL ( <"Finished?"> lookansw() ) 01723
  / "FORBID"!LL! 01724
    param3 ← #"FORBID" 01725
    seqbadd( param2, param3 ) 01726
    prtgrp 02200
    PERFORM protl UNTIL ( <"Finished?"> lookansw() ) 01728
  / "RESET"!LL! 01729
    param3 ← #"RESET" 01730
    seqbadd( param2, param3 ) 01731
  / "PRIVATE"!LL! 01732
    param3 ← #"PRIVATE" 01733
    seqbadd( param2, param3 ) 01734
    <"for"> prtgrp 01735
  ) 01736
/ ent ← "TEMPORARY" <"modifications for file"> 0213
  CONFIRM <"really?"> 03782
/ ent ← "TTY"!NOTT! <"simulation for window"> 0214
  lookbug() %for correct prompting% 04146

```



```

        param ← DSEL("#EDGE")
        / ent ← "VIEWSPECS"!LL!
        param ← VIEWSPECS )
CONFIRM
xset( ent, param, param2, param3, dest ) ;
prtgrp =
fromwhom ←
( "SELF"!LL!
/ "GROUP"!LL!
/ "PUBLIC"!LL!
)
seqbadd( param, fromwhom );
protl =
( param2 ← "SET"!LL! <"to">
seqbadd( param, param2 )
param2 ← LSEL("#NUMBER")
seqbadd( param, param2 )
/
( param2 ←
( "READ"!LL!
/ "WRITE"!LL!
/ "EXECUTE"!LL!
/ "APPEND"
/ "LIST"!LL!
/ "ALL"!LL!
) <"access"> seqbadd( param, param2)
)
);
%COMMAND retrieve%
%zretrieve =
"RETRIEVE" "FILE"!LL! <"from archive">
namfil ← LSEL("#OLDFILELINK")
CONFIRM
xretrieve( namfil );%
COMMAND %reset%
zreset =
"RESET"
dest ← NULL dent ← NULL
ent ←
( "ARCHIVE"!LL! <"request for file">
dest ← LSEL("#OLDFILELINK")
/ "CASE"!LL! <"mode">
/ "CHARACTER"!NOTT! <"size for window">
/ "CONTENT" <"Pattern">
/ "LINK"!LL! <"default for file">
/ "NAME"!LL! <"delimiters in">
dent ← structure
<"at"> dest ← DSEL(dent)
/ "TEMPORARY"!LL! <"modifications for file">
/ "TTY"!NOTT! <"window">
/ "VIEWSPECS"!LL! )
CONFIRM
xreset( ent, dent, dest );
COMMAND %replace%
zreplace =
"REPLACE"!LL!

```

0215
0216
0217
0218
0219
03786
03787
03788
03789
03790
03791
03792
03793
03794
03795
03796
03797
03798
03799
03800
03801
03802
03803
03804
03805
03806
03807
03808
0220
0221
0222
0223
0224
0225
0226
0227
0228
0229
0230
0231
0232
0233
0234
0236
0237
0238
0239
0240
0243
0244
0245
0246
0247
0248
0249
0250

```

dent ← editentity                                0251
<"at"> dest ← DSEL(dent)                          0252
sent ← dent                                        0253
<"by"> source ← LSEL(sent)                        0254
CONFIRM                                           0255
xreplace( dent, dest, sent, source );             0256
COMMAND %renumber%                                0257
  zrenumber =                                     0258
    "RENUMBER" "SIDS"!LL!                          0259
    <"in file"> CONFIRM                             0260
    xrenumber( );                                  0261
COMMAND %release%                                  0262
  zrelease =                                       0263
    "RELEASE"!NOTT!                                0264
    ent ←                                          0265
      ( "FROZEN"!LL! <"statement at">              0266
        dest ← DSEL("#STATEMENT")                 0267
        / "ALL"!LL! <"frozen statements"> dest ← NULL ) 0268
    CONFIRM                                         0269
    xrelease( ent, dest );                          0270
COMMAND %start%                                    0271
  zstart =                                         0272
    "START" "RECORD"!LL! <"of Session on file"> namfil ←
    LSEL("#NEWFILELINK")                            0273
    CONFIRM                                         0274
    xstart( namfil );                               0275
COMMAND %process dnls%                              01513
  zdprocess =                                      01514
    "PROCESS" !LL NOTT! process;                    01515
  process =                                        04227
    <"Commands from"> dent ← structure              01524
    <"at"> dest ← DSEL(dent)                         01523
    CONFIRM                                         01525
    xprocess( dent, dest );                         01520
COMMAND %process tnls%                              04220
  ztprocess =                                      04221
    "PROCESS" !NOTD! process;                       04222
COMMAND %print%                                     0299
  zprint =                                         0300
    "PRINT"!LL NOTD!                                0301
    ( ent ← "REST"!LL! CONFIRM dest ← NULL vs ← NULL 0302
    / ent ← "FILE"!LL! CONFIRM dest ← NULL vs ← NULL 03534
    / ent ← structure                               0303
      <"at"> dest ← DSEL( ent )                     0304
      vs ← VIEWSPECS                               0305
    / ent ← "JOURNAL"!LL! <"mail"> CONFIRM dest ← NULL vs ←
    NULL)                                           03150
    xprint( ent, dest, vs );                        0306
COMMAND %playback%                                  0307
  zplayback =                                      0308
    "PLAYBACK" "RECORD"!LL!                         0309
    <"of Session from file"> namfil ← LSEL("#OLDFILELINK") 0310
    CLEAR                                           04253
    <"Simulate Recorded timing?"> param2 ← answer() 04252
    CONFIRM                                         0311
    xplayback( namfil , param2 );                  0312

```

```

COMMAND %output%                                03843
zoutput =                                        03844
"OUTPUT"!LL! <"to"> xoutsapf( FALSE ) xoutsnhf( FALSE ) 03845
  ( ( ent ←                                       03846
    ( "JOURNAL"!LL! <"quickprint"> param2 ← FALSE 03847
    / "PRINTER"!LL! param2 ← FALSE                03848
    / "COM"!LL! )                                  03849
    namfil ← NULL                                  03850
    param ← TRUE % use default number of copies % 03851
    param2 ← FALSE % Not test. %                 03852
    ( lookconfirm()                               03853
    / "APPEND"!LL! <"to File">                   03854
      namfil ← LSEL("#OLDFILELINK") xoutsapf( TRUE ) 03855
    / "FILE"!LL! namfil ← LSEL("#NEWFILELINK")    03856
    / "COPIES"!LL! param ← LSEL("#NUMBER")        03857
    / "TEST"!LL! param2 ← #"TEST"                 03858
      ( lookconfirm()                             04147
      / "FILE"!LL! namfil ← LSEL("#NEWFILELINK") 04148
      )                                           04149
    )                                           03859
    CONFIRM                                       03860
    xoutl( ent, namfil, param, param2 )          03861
  / ( ent ←                                       03862
    "QUICKPRINT"!LL!                              03863
    namfil ← NULL                                  03864
    param ← TRUE % use default number of copies % 03865
    param2 ← FALSE % Not test. %                 03866
    ( lookconfirm()                               03867
    / "NO"!LL! <"Headers"> xoutsnhf( TRUE )      03868
      ( lookconfirm()                             03869
      / "APPEND"!LL! <"to File">                 03870
        namfil ← LSEL("#OLDFILELINK") xoutsapf( TRUE ) 03871
      / "FILE"!LL! namfil ← LSEL("#NEWFILELINK") 03872
      / "COPIES"!LL! param ← LSEL("#NUMBER")      03873
      / "TEST"!LL! param2 ← #"TEST"              03874
      )                                           03875
    / "APPEND"!LL! <"to File">                   03876
      namfil ← LSEL("#OLDFILELINK") xoutsapf( TRUE ) 03877
    / "FILE"!LL! namfil ← LSEL("#NEWFILELINK")    03878
    / "COPIES"!LL! param ← LSEL("#NUMBER")        03879
    / "TEST"!LL! param2 ← #"TEST"                 03880
    )                                           03881
    CONFIRM                                       03882
    xoutl( ent, namfil, param, param2 )          03883
    param ← TRUE % use default number of copies % 03884
  / ( ent ←                                       03885
    ("SEQUENTIAL"!LL! /                            03886
    "ASSEMBLER"!LL! )                              03887
    ( "APPEND"!LL! < \TO File"> xoutsapf(TRUE)    03888
      namfil ← LSEL("#OLDFILELINK")              03889
    / "FILE"!LL!                                   03890
      namfil ← LSEL("#NEWFILELINK")              03891
    )                                           03892
  )

```

```

        param ← FALSE                                03893
        ( lookconfirm()                              03894
          / "FORCE"!LL! <"upper case"> param ← TRUE 03895
        )                                            03896
        param2 ← FALSE                               03897
        CONFIRM                                      03898
        xout1( ent, namfil, param, param2 )         03899
/ (                                                03900
  ( ent ← "TERMINAL"!LL! param ← NULL port ← NULL 03901
    ( lookconfirm()                                04261
      / "FILE"!LL! param ← LSEL("#NEWFILELINK")    04262
    )                                              04263
  / ent ← "REMOTE"!LL!                             03902
    <"printer -- TIP"> param ← LSEL("#VISIBLE")    03903
    <"Port #"> port ← LSEL("#NUMBER") )          03904
CONFIRM                                           03905
CLEAR                                             03906
<"Send Form Feeds?">                             03907
  (answ() ff ← TRUE sim ← FALSE                   03908
  / <"Simulate?"> ff ← FALSE sim ← answer()      03909
CLEAR                                             03910
<"Wait at page break?"> pb ← answer()           03911
CLEAR                                             03912
<"Go?">                                           03913
  ( answ()                                         03914
  /                                               03915
    <"Type CA when ready, CD to abort">          03916
    readconfirm()                                 03917
  xout2( ent, param, port, ff, sim, pb)         03918
);                                               03919
COMMAND %move%                                    0360
  zmove =                                         0361
  "MOVE"!LL!                                     0362
  filtre ← FALSE                                 0363
  vs ← NULL                                      0364
  level ← NULL                                   0365
  ( sent ← "LINK"!LL!                            0366
    copy1                                         02952
    dent ← sent                                   02953
    dest ← DSEL("#VISIBLE")                       02954
  / sent ← "NUMBER"!LL!                          04237
    copy1                                         04238
    dent ← sent                                   04239
    dest ← DSEL("#VISIBLE")                       04240
  / sent ← text1                                  02951
    copy1                                         0367
    dent ← sent                                   0368
    dest ← DSEL(dent)                             0369
  / sent ← "TEXT"!LL!                             0370
    copy1                                         0371
    dent ← "#CHARACTER"                          0372
    dest ← DSEL(dent)                             0373
  / sent ← structure                              0374
    copy1                                         0375
    dent ← "#STATEMENT"                          0376
    dest ← DSEL(dent)                             0377

```

```

        [filtre ← TRUE <"Filtered:"> vs ← VIEWSPECS/           0378
        level ← LEVADJ                                         0379
    / dent ← "FILE"!LL!                                       0380
        sent ← dent                                           0381
        <"from old filename"> source ← LSEL("#OLDFILELINK")    0382
        <"to new filename"> dest ← LSEL("#NEWFILELINK")        0383
    / sent ← "EDGE"!LL NOTT!                                   0384
        <"from"> source ← DSEL("#EDGE")                       0386
        <"to">                                                 03638
            (lookbug() dent ← NULL / dent ← "CENTER"!LL! <"of">) 03425
            dest ← DSEL("#EDGE")                               03426
    )                                                         0388
    CONFIRM                                                  0389
    xmove(sent, source, dent, dest, level, filtre, vs);      0390
COMMAND %merge%                                           0391
    zmerge =                                               0392
        "MERGE"                                             0393
        sent ← notstatement                                  0394
        <"at"> source ← SSEL(sent)                            0395
        dent ← sent                                          0396
        <"into"> dest ← DSEL(dent)                           0397
    CONFIRM                                                  0398
    xmerge( sent, source, dent, dest);                       0399
COMMAND %mark%                                           0400
    zmark =                                                 0401
        "MARK" "CHARACTER"!LL! <"at">                       0402
        dest ← DSEL( "#CHARACTER" )                         0403
        <"with marker named"> source ← LSEL("#NAME")        0404
    CONFIRM                                                  0405
    xmark( dest, source);                                    0406
COMMAND %logout%                                          0407
    zlogout =                                               0408
        "LOGOUT"                                            0409
    CONFIRM xlogout();                                       0410
COMMAND %load%                                           0411
    zload =                                                 0412
        "LOAD"!LL!                                         0413
        ent ← "FILE"!LL!                                    0414
        namfil ← LSEL("#OLDFILELINK") CONFIRM              0415
        xload(ent, namfil);                                  0416
%k (unused)%                                             0417
COMMAND %insert%                                          0499
    zinsert =                                               0500
        "INSERT"!LL!                                       0501
        level ← NULL                                        0502
        ( ent ← "LINK"!LL!                                   0503
            <"to follow">                                    0504
            dest ← DSEL("#VISIBLE")                          0505
            param ← LSEL(ent)                                0506
        / ent ← "NUMBER"!LL!                                 04241
            <"to follow">                                    04242
            dest ← DSEL("#VISIBLE")                          04243
            param ← LSEL(ent)                                04244
        / ent ← text1                                        02959
            <"to follow">                                    02960

```

```

        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 ← "DATE"
  <"to follow">
  dest ← DSEL("#VISIBLE")
  param ← NULL
/ ent ← "TIME"
  <"and Date to follow">
  dest ← DSEL("#VISIBLE")
  param ← NULL
/ ent ← "SENDMAIL" <"form">
  <"to follow">
  dest ← DSEL("#STATEMENT")
  level ← LEVADJ
  param ← NULL
/ ent ← "EDGE"!LL NOTT! <"perpendicular to">
  ( lookbug() param ← NULL
  / param ← "CENTER"!LL! <"of"> )
  dest ← DSEL( ?\EDGE" )
)
CONFIRM
xinsert(ent, dest, level, param);
zinsstatement = % INS STATEMENT FUNCTION %
CLEAR
level ← LEVADJ
param ← LSEL("#STATEMENT")
CONFIRM xinsstatement(level, param );
% h (unused) %
COMMAND %freeze%
zfreeze =
  "FREEZE"!NOTT! "STATEMENT"!LL! <"at">
  dest ← DSEL("#STATEMENT")
  vs ← VIEWSPECS CONFIRM
  xfreeze(dest, vs);
% COMMAND format%
COMMAND %force case%
zforce =
  "FORCE"!LL! <"Case">
  ( param ← editentity
  <"at"> dest ← DSEL( param )
  param2 ← NULL {param2 ← cshmode/
/ param ← "MODE"!LL! dest ← NULL
  param2 ← cshmode
  )
CONFIRM
xforce( param, param2, dest ) ;
% CASE SHIFT MODES %

```

```

02961
02962
0507
0508
0509
0510
0511
0512
0513
0514
0515
02191
02192
02193
02194
02195
02196
02197
02198
0516
0517
0518
0519
0520
03382
03383
03423
03424
0521
0522
0523
0524
02913
0525
0526
0527
0528
0529
0530
0531
0532
0533
0534
02212
03309
03310
03311
03319
03320
03321
03322
03323
03324
03325
03326
016

```

```

cshmode =                                017
  ( "UPPER"!ll!                             018
  / "LOWER"!ll!                             019
  / "FIRST"!ll! <"letter upper"> );       020
COMMAND %expunge%                          0535
  xexpunge =                                0536
    "EXPUNGE"                               0537
    ent ←                                    0538
      ( "DIRECTORY"!ll!                     0539
      %/ "ARCHIVE"!ll! <"directory">% )    0540
    CONFIRM                                  0541
    xexpunge(ent);                          0542
COMMAND %edit%                              0543
  zedit =                                    0544
    "EDIT"!NOTD! "STATEMENT"!ll! <"at">   0545
    dest ← DSEL("#STATEMENT")              0546
    xedit(dest);                            0547
COMMAND %disconnect%                        0548
  zdisconnect =                             0549
    "DISCONNECT" "TERMINAL"!ll! CONFIRM    0550
    xdisconnect();                          0551
COMMAND %delete%                            0552
  zdelete =                                  03278
    "DELETE"!ll!                            03279
    filtre ← FALSE dest ← NULL vs ← NULL    03280
    (                                         03281
      ent ← textent                         03282
      <"at"> dest ← DSEL(ent)                 03283
      needconfirm ← TRUE % to force confirmation % 03284
      / ent ← structure                     03285
      <"at"> dest ← DSEL(ent)               03286
      /                                       03287
        filtre ← TRUE <"Filtered:">         03288
        vs ← VIEWSPECS                      03289
        needconfirm ← TRUE % to force confirmation % 03290
      /                                       03291
      ent ← ("FILE"!ll! %/ "ARCHIVE" <"file">%) 03292
      dest ← LSEL("#OLDFILELINK")           03293
      needconfirm ← TRUE % to force confirmation % 03294
      / "MARKER" <"named">                  03295
      ent ← #"MARKER"                       03296
      dest ← LSEL(ent)                      03297
      needconfirm ← TRUE % to force confirmation % 03298
      / "EDGE"!ll NOTT! <"at">              03427
      ent ← #"EDGE"                         03428
      dest ← DSEL(ent)                      03429
      / "ALL"!ll! <"markers">              03299
      ent ← #"ALL"                           03300
      needconfirm ← TRUE % to force confirmation % 03301
      / "MODIFICATIONS"!ll! <"to file">     03302
      ent ← #"MODIFICATIONS"               03303
      CONFIRM <"really?">                  03304
      needconfirm ← TRUE % to force confirmation % 03305
    )                                        03306
  CONFIRM                                    03307
  xdelete(ent, dest, filtre, vs);          03308

```

```

COMMAND %create%
zcreate =
  "CREATE" "FILE"!LL!
  namfil ← LSEL("#NEWFILELINK") CONFIRM
  xcreate(namfil);
COMMAND %copy%
zcopy =
  "COPY"!LL!
  vs ← NULL
  level ← NULL
  source ← NULL
  filtre ← NULL
  dest ← NULL
  ( sent ← "LINK"!LL!
    copy1 ← source ← DSEL
    dent ← sent
    dest ← DSEL("#VISIBLE")
  / sent ← "NUMBER"!LL!
    copy1
    dent ← sent
    dest ← DSEL("#VISIBLE")
  / sent ← text1
    copy1
    dent ← sent
    dest ← DSEL(dent)
  / sent ← "TEXT"!LL!
    copy1
    dent ← "#CHARACTER"
    dest ← DSEL(dent)
  / sent ← structure
    copy1
    dent ← "#STATEMENT"
    dest ← DSEL(dent)
    [filtre ← TRUE <"Filtered:"> vs ← VIEWSPECS/
    level ← LEVADJ
  / sent ← "FILE"!LL!
    dent ← sent
    <"from"> source ← LSEL("#OLDFILELINK")
    <"to"> dest ← LSEL("#NEWFILELINK")
  / sent ← "DIRECTORY"!LL! source LSEL*(Directory)
    copy3 <">
    ( readoption() CLEAR <"opt:"> dest DSEL*(Statement)
      PERFORM diropt UNTIL( <"Finished?"> lookansw())
    / DUMMY) % dummy case %
  %/ sent ← "ARCHIVE"!LL! <"directory">
    copy3 <"directory options:">
    ( readoption()
      PERFORM copy4 UNTIL( <"Finished?"> lookansw())
    / DUMMY)% % dummy case %
  / sent ← "SEQUENTIAL"
    <"file from"> source ← LSEL("#OLDFILELINK")
    <" to follow"> dest ← DSEL("#STATEMENT")
    level ← LEVADJ
    dent ← NULL
    CLEAR <"using">
    ("ONE" !LL! <"<CR> to end statement">

```

0573
0574
0575
0576
0577
0578
0579
0580
0581
0582
01707
0583
0584
0585
0586
0587
0588
04245
04246
04247
04248
02955
02956
02957
02958
0589
0590
0591
0592
0593
0594
0595
0596
0597
0598
0599
0600
0601
0602
0603
0604
0605
0606
0607
0608
0609
0610
0611
0612
04264
04265
04266
04267
04268
04269
04270


```

    / dent ← "TWO" !LL! <"<CR>s end statement">                                04271
      (lookconfirm() /                                                         04272
        dent ← "JUSTIFIED" !LL! <"delete extra <SP>">)                       04273
    / dent ← "ASSEMBLER"!LL!)                                                04274
  )                                                                              0619
  CONFIRM                                                                      0620
  xcopy(sent, source, dent, dest, level, filtre, vs);                         0621
copy1 =                                                                        0622
  <"from"> source ← SSEL(sent)                                                0623
  <"to follow">;                                                                0624
copy3 =                                                                        0627
  <"of">                                                                        0628
  ( readconfirm()                                                            01709
    / source ← LSEL( #"DIRECTORY" )                                           01710
  )                                                                              01711
  <"to follow"> dest ← DSEL( #"STATEMENT" )                                  0629
  level ← LEVADJ                                                                0630
  dent ← seqbinit();                                                          0631
copy4 = param2 ← diropt seqbadd( dent, param2 );                             0633
COMMAND %connect%                                                             0634
zconnect =                                                                     0635
  "CONNECT"                                                                     0636
  <"to">                                                                         0637
  (ent ← ("DISPLAY" / "TTY"!LL!)                                             0638
    <"Number"> dest ← LSEL( #"NUMBER" ) <"for">                               0639
    param ← ("INPUT"!LL! <"and Output"> / "OUTPUT"!LL!
    <"Only"> ) /
    ent ← "DIRECTORY"!LL! dest ← LSEL( #"DIRECTORY" )                       0640
    param ← NULL <"Password">                                               01697
    ( lookconfirm()                                                           01698
      / param ← LSEL( #"PASSWORD" ) )                                         01699
  )                                                                              01700
  CONFIRM                                                                      01701
  xconnect(ent, dest, param);                                                 0645
COMMAND %clear%                                                                0646
zclear =                                                                        0647
  "CLEAR"!NOTT! <"TTY Window"> CONFIRM xclear();                            0648
COMMAND %break%                                                                0649
zbreak =                                                                        0652
  "BREAK"!LL!                                                                    0653
  ent ← "STATEMENT"!LL!                                                       0654
  <"at"> dest ← DSEL( #"CHARACTER" )                                         0655
  level ← LEVADJ                                                                0656
  CONFIRM                                                                      0657
  xbreak(ent, dest, level); %should also pass literal%                       0658
COMMAND %archive%                                                             0659
zarchive =                                                                      0660
  "ARCHIVE"                                                                      03189
  ( "FILE"!LL!                                                                  03190
    namfil ← LSEL( #"OLDFILELINK" )                                           03191
    param ← seqbinit()                                                         03192
    [ arcnl /                                                                    03193
  ) CONFIRM xarchive(namfil, param);                                          03194
arch1 =                                                                        03195
  CLEAR <"opt:">                                                                03196
  PERFORM archopts UNTIL (lookansw());                                         03197

```

```

archopts = 03199
  param2 ← 03200
    ( "DELETE"!LL! <"After Archiving"> 03201
      / "DEFERRED" 03202
      % NOT IMPLEMENTED 03526
      / "IMMEDIATE"!LL! 03203
      % 03527
      / "NOT"!LL! <"Allowed"> 03204
      / "PREVENT"!LL! <"Deletion After Archiving"> 03205
      / "RESET"!LL! <"Request Status"> 03206
    ) 03207
  CONFIRM seqbadd( param, param2 ) <"Finished?">; 03208
COMMAND %append% 0672
  zappend = 0673
    "APPEND"!LL! 0674
    sent ← "STATEMENT"!LL! 03528
    zappl 03529
    dent ← #"STATEMENT" 03530
    % REST NOT IMPLEMENTED (dash replaces percent) 03531
    ( sent ← "LINK"!LL! 0675
      zappl 0676
      dent ← #"VISIBLE" 0677
      / sent ← "NUMBER"!LL! 04249
      zappl 04250
      dent ← #"VISIBLE" 04251
      / sent ← text1 - text entities except "TEXT" - 02963
      zappl 02964
      dent ← sent 02965
      / sent ← "TEXT"!LL! 0678
      zappl 0679
      dent ← #"CHARACTER" 0680
      / sent ← structure 0681
      zappl 0682
      dent ← #"STATEMENT" 0683
    ) 0684
    % 03532
    dest ← DSEL(dent) 0685
    <"join with"> literal ← LSEL(#"TEXT") 0686
    CONFIRM 0687
    xappend(sent, source, dent, dest, literal); 0688
  zappl = 0689
    <"at"> 0690
    source ← SSEL(sent) 0691
    <"to">; 0692
COMMAND %accept% 0693
  zaccept = 0694
    "ACCEPT" "CONNECT"!LL! 0695
    <"from terminal number"> dest ← LSEL(#"NUMBER") <"for"> 0696
    param ← ("INPUT"!LL! <"and output"> / "OUTPUT"!LL!
    <"Only">) 03416
    CONFIRM xaccept( param, dest ); 0697
COMMAND %period% 0702
  zperiod = 0703
    ". "!LL NOTD! 0704
    xperiod(); 0705

```

```

COMMAND %tab%                                0706
  ztab =                                      0707
    " !ll!                                    0708
      xtab() ;                                0709
COMMAND %slash%                               0710
  zslash =                                     0711
    "/"!ll! NOTD!                             0712
      xslash() ;                             0713
COMMAND %bslash%                              0714
  zbslash =                                   0715
    "\"!ll! NOTD!                             0716
      xbslash() ;                             0717
COMMAND %uparrow%                             0718
  zuparrow =                                  0719
    "↑"!ll! NOTD!                             0720
      xuparrow() ;                             0721
COMMAND %linefeed%                             0722
  zlinefeed =                                 0723
    "<LF>!ll! NOTD!                             0724
      xlinefeed() ;                             0725
% DIRECTORY OPTIONS %                          01536
% dummy rule to get dent defined properly %    01537
  diropl = dent ← FALSE;                       01538
% options for what is to be listed %          02184
  diropt =                                      01539
    gparam ← FALSE                             01540
    gparam2 ← FALSE                            01541
    gparam3 ← FALSE                            01542
    gparam4 ← FALSE                            01543
    ( "ALL" <"Files">                          01544
      gparam ← #"BOTH"                          01545
    / "DELETE" <"Files Only">                   01546
      gparam ← #"DELETE"                       01547
    / "UNDELETE"!ll! <"Files Only">            01548
      gparam ← #"UNDELETE"                     01549
    / "FOR"!ll! <"File">                        01550
      filtre ← LSEL(#"OLDFILELINK")            01551
      gparam ← #"FOR"                           01552
    / "ARCHIVE"!ll!                             01553
      gparam ← #"ARCHIVE"                       01554
      ( "STATUS"!ll!                             01555
        gparam2 ← #"STATUS"                     01556
      / "TAPE"!ll! <"Numbers">                 01557
        gparam2 ← #"TAPE"                       01558
      )                                          01559
    / "ACCOUNT"                                 01560
      gparam ← #"ACCOUNT"                       01561
    / "DATE"!ll! <"of">                         01562
      gparam ← #"DATE"                          01563
      ( "ARCHIVE"!ll!                             01564
        gparam2 ← #"ARCHIVE"                     01565
      / "CREATION"!ll!                             01566
        gparam2 ← #"CREATION"                   01567
      / "LAST"!ll! <"Dump">                    01568
        gparam2 ← #"LAST"                       01569
      )                                          01570

```

```

/ "FIRST"!L1! <"Version Creation"> 01571
  gparam ← # "FIRST" 01572
/ "READ"!L1! 01573
  gparam ← # "READ" 01574
/ "WRITE"!L1! 01575
  gparam ← # "WRITE" 01576
) 01577
/ "DUMP" <"Tape Number"> 01578
  gparam ← # "DUMP" 01579
/ "EVERYTHING"!L1! 01580
  gparam ← # "EVERYTHING" 01581
/ "LAST"!L1! <"Writer"> 01582
  gparam ← # "LAST" 01583
/ "LENGTH" <"and Bytesize"> 01584
  gparam ← # "LENGTH" 01585
/ "MISCELLANEOUS"!L1! <"Information"> 01586
  gparam ← # "MISCELLANEOUS" 01587
/ "NUMBER"!L1! <"of"> 01588
  gparam ← # "NUMBER" 01589
  ( "VERSIONS"!L1! <"to keep"> 01590
    gparam ← # "VERSIONS" 01591
  / "ACCESSES"!L1! 01592
    gparam ← # "ACCESSES" 01593
  ) 01594
/ "NO" 02431
  gparam ← # "NO" 02432
  ( "VERSIONS"!L1! <"number"> 02433
    gparam ← # "VERSIONS" 02434
  / "EXTENSION"!L1! <"name"> 02435
    gparam ← # "EXTENSION" 02436
  ) 02437
/ "PROTECT"!L1! 01595
  gparam ← # "PROTECT" 01596
/ "SIZE"!L1! <"in Pages"> 01597
  gparam ← # "SIZE" 01598
/ "TIME"!L1! <"and Date of"> 01599
  gparam ← # "TIME" 01600
  ( "ARCHIVE"!L1! 01601
    gparam ← # "ARCHIVE" 01602
  / "CREATION"!L1! 01603
    gparam ← # "CREATION" 01604
  / "LAST"!L1! <"Dump"> 01605
    gparam ← # "LAST" 01606
  / "FIRST"!L1! <"Version Creation"> 01607
    gparam ← # "FIRST" 01608
  / "READ"!L1! 01609
    gparam ← # "READ" 01610
  / "WRITE"!L1! 01611
    gparam ← # "WRITE" 01612
  ) 01613
/ "VERBOSE"!L1! 01614
  gparam ← # "VERBOSE" 01615
/ "GROUP"!L1! <"by"> 01616
  gparam ← # "GROUP" 01617
  ( "REVERSE"!L1! 01618
    gparam ← # "REVERSE" 01619
  )

```

```

        grpopt                                01620
        / grpopt                               01621
        )                                       01622
/ "SORT" <"by">                                01623
  gparam ← #"SORT"                            01624
  ( "REVERSE"!L1!                             01625
    gparam2 ← #"REVERSE"                      01626
    srtopt                                       01627
  / srtopt                                       01628
  )                                             01629
) CONFIRM                                       01630
  seqbadd( dent, gparam)                       01631
  seqbadd( dent, gparam2)                     01632
  seqbadd( dent, gparam3)                     01633
  seqbadd( dent, gparam4)                     01634
;                                               01636
% options for grouping %
grpopt =                                       01637
( "NO"!L1! <"Grouping"> gparam3 ← #"NO"      01638
/ "ACCOUNT" gparam3 ← #"ACCOUNT"              01639
/ "ARCHIVE"!L1! gparam3 ← #"ARCHIVE"          01640
  ( "DATE"!L1! gparam4 ← #"DATE"              01641
  / "STATUS"!L1! gparam4 ← #"STATUS"           01642
  / "TAPE"!L1! gparam4 ← #"TAPE"              01643
  )                                             01644
/ "CREATION"!L1! <"Date"> gparam3 ← #"CREATION" 01645
/ "DELETE"!L1! <"Status"> gparam3 ← #"DELETE" 01646
/ "DUMP" gparam2 ← #"DUMP" gparam3 ← #"DUMP" 01647
  ( "DATE"!L1! gparam4 ← #"DATE"              01648
  / "TAPE"!L1! gparam4 ← #"TAPE"              01649
  )                                             01650
/ "FIRST"!L1! <"Version Creation"> gparam3 ← #"FIRST" 01653
/ "LAST"!L1! <"Writer"> gparam3 ← #"LAST"     01652
/ "NUMBER"!L1! <"of Versions to Keep"> gparam3 ← #"NUMBER" 01651
/ "PROTECT"!L1! gparam3 ← #"PROTECT"          01654
/ "READ"!L1! <"Date"> gparam3 ← #"READ"       01655
/ "WRITE"!L1! <"Date"> gparam3 ← #"WRITE"     01656
);                                             01657
%options for sorting within groups %
srtopt =                                       01658
( "ACCOUNT" gparam3 ← #"ACCOUNT"              01659
/ "ALPHABETICAL"!L1! gparam3 ← #"ALPHABETICAL" 01660
/ "ARCHIVE" gparam3 ← #"ARCHIVE"              01661
  ( "TAPE" gparam4 ← #"TAPE"                  01663
  / "TIME"!L1! <"and Date"> gparam4 ← #"TIME" 01664
  )                                             01665
/ "BYTESIZE"!L1! gparam3 ← #"BYTESIZE"        01666
/ "CREATION"!L1! <"time and Date"> gparam3 ← #"CREATION" 01667
/ "DELETE"!L1! <"Status"> gparam3 ← #"DELETE" 01668
/ "DUMP" gparam2 ← #"DUMP" gparam3 ← #"DUMP" 01669
  ( "TAPE" gparam4 ← #"TAPE"                  01670
  / "TIME"!L1! <"and Date"> gparam4 ← #"TIME" 01671
  )                                             01672
/ "LAST" <"Writer"> gparam3 ← #"LAST"         01674
/ "LENGTH"!L1! <"in Bytes"> gparam3 ← #"LENGTH" 01675
/ "NUMBER"!L1! <"of"> gparam3 ← #"NUMBER"     01676

```

```

( "ACCESSES"!LL! gparm4 ← #"ACCESSES" 01677
/ "READ"!LL! gparm4 ← #"READ" 01678
/ "WRITE"!LL! gparm4 ← #"WRITE" 01679
/ "VERSIONS"!LL! <"to keep"> gparm4 ← #"VERSIONS" 01686
) 01680
/ "FIRST"!LL! <"Version Creation"> gparm3 ← #"FIRST" 01681
/ "READ"!LL! <"Time and Date"> gparm3 ← #"READ" 01682
/ "SIZE"!LL! <"in Pages"> gparm3 ← #"SIZE" 01683
/ "WRITE"!LL! <"Time and Date"> gparm3 ← #"WRITE" 01684
); 01685

END. 0726
% NLS SUPERVISOR COMMANDS % 0727
SUBSYSTEM subsupervisor KEYWORD "SUPERVISOR" 0728
COMMAND % display subsystem stack % 0729
zdsps = 0730
"<"!LL! xsublist(); 0731
COMMAND % display current subsystem stack name % 0732
zdsps = 0733
">"!LL! xsubcurrent(); 0734
COMMAND zksyntax = % show syntax of a command % 04150
"SYNTAX" <"of Command"> 04151
param ← NULL 04152
xkbuild( param ) 04153
syncur ← NULL 04154
synsubs 04155
CONFIRM 04156
xksyntax( syncur ); 04157
COMMAND % quit % 0735
zquit = 0736
"QUIT"!LL! 0737
grammar ← NULL 0738
( readconfirm() param ← NULL xquit(param, grammar) 0739
/ ( "TO"!LL! param ← nissubs 0740
./ param ← "NLS"!LL! 02102
) CONFIRM xquit(param, grammar) 02103
); 02163
% note-- the rule nissubs sets the variable grammar to 0741
contain a pointer to a subsystem dispatch record % 0741
COMMAND %tjump% 03810
ztjump = 03811
"JUMP"!NOTD LL! <"to"> 03812
( 03817
"ADDRESS"!LL! 03631
ent ← #"ITEMNOVS" 03632
dest ← DSEL( #"CHARACTER" ) 03633
vs ← NULL 03635
/ jmpcoms 03813
) 03814
CONFIRM 03815
xjump(ent, dest, vs); 03816
COMMAND %djump% 0423
zdjump = 0424
"JUMP"!NOTT LL! <"to"> 0425
( 0426
lookbug() % look for a bug select % 0427
ent ← #"ITEM" 0428

```

```

        dest ← DSEL( #"STATEMENT" )           0429
        vs ← VIEWSPECS                          01425
    / "ADDRESS"!LL! <"relative to">           01426
        ent ← #"ITEM"                          01533
        dest ← DSEL( #"STATEMENT" )           01428
        dest ← xjdae( dest )                   0430
        vs ← VIEWSPECS                          01429
    / jmpcoms                                  03809
    )                                           0495
    CONFIRM                                    0496
    xjump(ent, dest, vs);                       0497
    jmpring = dest ← xringjump( ent );         0498
    jmpcoms =                                  0431
    ent ←                                       0432
    ( "ITEM"!LL!                               0434
    / "SUCCESSOR"!LL!                          0435
    / "PREDECESSOR"!LL!                       01532
    / "UP"!LL!                                 0436
    / "DOWN"!LL!                               0437
    / "HEAD"!LL!                              0438
    / "TAIL"!LL!                              0439
    / "END"!LL! <"of Branch">                0440
    / "BACK"!LL!                              0441
    / "ORIGIN"!LL!                            0442
    / "NEXT"                                   0443
    ) dest ← DSEL( #"CHARACTER" ) vs ← VIEWSPECS 0444
    /                                           0446
    ( ent ← "LINK"!LL!                         0447
      dest ← LSEL( #"LINK" ) vs ← NULL         03818
    / ent ← "RETURN"!LL!                       0450
      CONFIRM                                  0451
      vs ← NULL                                01765
      %init for moving back through ring%      0452
      jrinit( #"RETURN" )                     0453
      PERFORM jmpring UNTIL (lookansw())       0454
    / ent ← "FILE"!LL!                         0460
      (% sp()                                  0461
        dest ← LSEL( #"OLDFILELINK" )         03819
        vs ← VIEWSPECS /%                      03820
      lookbug()                                0463
        dest ← DSEL( #"OLDFILELINK" )         03821
        vs ← VIEWSPECS                         03822
    / "NAMED"!LL!                              03636
      ent ← #"FILENAME"                       03823
      dest ← LSEL( #"FILE" )                  03637
      vs ← VIEWSPECS                          03824
    / "RETURN"!LL!                              0469
      ent ← #"FILEReturn"                    03825
      CONFIRM                                  0470
      vs ← NULL                                01763
      %init for moving back through ring%      0471
      jrinit( #"FILEReturn" )                 0472
      PERFORM jmpring UNTIL (lookansw())       0473
    )                                           0474
    / ent ← "NAME"!LL!                         0475
      ( lookbug()                              0476

```

```

/ "ANY"!ll!                                03826
/ "FIRST"!ll!                                0477
  ent ← #"FIRSTNAME"                        03829
/ "NEXT"!ll!                                0478
  ent ← #"NEXTNAME"                          03828
/ "EXTERNAL"!ll!                            02310
  ent ← #"EXTNAME"                          03827
)                                             0479
  dest ← LSEL(#"NAME") vs ← VIEWSPECS      03830
/ "CONTENT"!ll!                              0480
  ( "FIRST"!ll!                              0481
    ent ← #"FIRSTCONTENT"                  03831
  / "NEXT"!ll!                              0482
    ent ← #"NEXTCONTENT"                  03832
  )                                          03833
  % display previous content %              03834
  dest ← xjmpcntdisp()                      0483
  ( readrepeat() % accept old option %      0484
  / prmptisel() clearname() dest ← LSEL(#"TEXT") 0485
  )                                          03835
  vs ← VIEWSPECS                            0486
/ "WORD"!ll!                                 0487
  ( "FIRST"!ll!                              0488
    ent ← #"FIRSTWORD"                    03838
  / "NEXT"!ll!                              0489
    ent ← #"NEXTWORD"                     03837
  )                                          03836
  % display previous content %              03839
  dest ← xjmpcntdisp()                      0490
  ( readrepeat() % accept old option %      0491
  / prmptisel() clearname() dest ← LSEL(#"WORD") 0492
  )                                          03840
  vs ← VIEWSPECS                            0493
);                                           0494
COMMAND %tnls help%                          03926
  help =                                     03927
  "HELP" !ll NOTD! (myrdconfirm() param ← NULL / param ←
  LSEL(#"WORD")) hlpinit(#"HLP COM", param) helpt; 03928
  %puts out help message, loads help file if necessary;
  does initialization, allocation of storage. In cleanup,
  backup mode handles setup of data structure for reentry,
  restoration of screen. Initialize moremen to FALSE. (No
  more to be shown)%                          03929
qthelp =                                     03930
  CLEAR <"> hlpinit(#"CNTLQ") helpt;        03979
helpt =                                     03931
  setcutback() trpthelp cutback(); % This is a loop!!! user
  must Command Delete out of this command%    03932
trpthelp =                                  03933
  hchk hshw;                                  03934
hchk =                                       03935
  checkmore() % Check moremen %              03936
  CLEAR <"do you want to see the rest of the menu?"> 03937
  answ() helpshow(#"REST") (hchk / true())    03938
/ rstmor(); % reset moremen (answer was no); close
sequence. Moremen always FALSE, sequence qsw closed upon

```



```

% note-- the rule nlssubs sets the variable grammar to
contain a pointer to a subsystem dispatch record %
grammar ← NULL
param ← nlssubs
xgoto(param, grammar, TRUE);
COMMAND %comment%
zcomment =
";"!ll!
xcomment();
END.
% USER PROGRAMMING SUBSYSTEM COMMANDS %
SUBSYSTEM subprograms KEYWORD "PROGRAMS"
uproctypes =
("CONTENT"!ll! <"analyzer program">/ "SORT"!ll! <"key
extractor program">/ "SEGGEN" <"program"> );
COMMAND %show%
zpslow =
"SHOW"!ll!
( "STATUS"!ll! <"of programs buffer">
CONFIRM xpslow( )
/ "TENEX"!ll! <"Subsystem Status">
CONFIRM xpshtn( )
);
COMMAND %set%
zpsset =
"SET"
ent ←
( "BUFFER"!ll! <"size to">
source ← LSEL("#NUMBER")
/ "NDDT"!ll! <"control-h">
source ← NULL )
CONFIRM xpsset(ent, source );
COMMAND %run%
zprun =
"RUN"!ll!
( "PROGRAM"!ll!
source ← LSEL("#NAME")
CONFIRM xprun( source )
/ "?\TENEX"!ll! <"Subsystem">
source ← LSEL("#OLDFILELINK"); CONFIRM
CLEAR <"Output to File or Teletype?">
( "FILE"!ll! param ← LSEL("#NEWFILELINK?\")
/ "TELETYPE"!ll! param ← FALSE
) CONFIRM
CLEAR <"Input Mode?">
param2 ←
( "FROM"!ll 3! <"FILE:">
param3 ← LSEL("#OLDFILELINK")
/ "INTERACTIVE"!ll 1! <"Termination
Character:">
param3 ← LSEL("#CHARACTER")
/ "TYPEAHEAD"!ll 2!
param3 ← LSEL("#TEXT")
/ "NO"!ll 4! <"Input">
param3 ← FALSE
) CONFIRM

```

```

        CLEAR <"Wait for Completion?">                                03354
        ( "YES"!LL! param4 ← TRUE                                     03355
          / "NO"!LL! param4 ← FALSE                                  03356
        ) CONFIRM                                                    03357
        CLEAR <"Go?">                                                03358
        CONFIRM xprunt( source, param, param2, param3, param4      03359
        )                                                            03360
    );                                                                0775
COMMAND %reset%                                                    0776
    zreset =                                                         0777
        "RESET"                                                      0778
        param ← NULL param2 ← NULL                                    01512
        ent ←                                                         0778
        ( "BUFFER"!LL! <"size">                                       0779
          / "NDDT"!LL! <"control-h">                                   0780
          / "PARSERULE"!LL!                                           01509
            <"named"> param ← LSEL("#NAME")                            01510
            <"in rel-file named"> param2 ←
            LSEL("#OLDFILELINK")                                       01511
        ) CONFIRM xpreset( ent, param, param2 );                    0781
COMMAND %replace%                                                 01498
    zreplace =                                                       01499
        "REPLACE"                                                    01500
        "PARSERULE"!LL!                                             01502
        <"named"> param ← LSEL("#NAME")                               01505
        <"in rel-file named"> param2 ← LSEL("#OLDFILELINK")         01506

        CLEAR <"by new parserule named">                            01507
        param3 ← LSEL("#NAME")                                        01508
        CONFIRM xpreplace( param, param2, param3 );                01504
COMMAND %load%                                                    03417
    zpload =                                                         03418
        "LOAD"!LL! "PROGRAM"!LL!                                     03419
        xpsetf=[]
        source ← LSEL("#OLDFILELINK")                                03421
        CONFIRM xpload( source );                                    03422
COMMAND %kill%                                                    01813
    zkill =                                                         01814
        "KILL"!LL! "TENEX"!LL! <"subsystem">                       01815
        CONFIRM xpkill();                                           01816
COMMAND %institute%                                               0792
    zpinstitute =                                                  0793
        "INSTITUTE"!LL! "PROGRAM"!LL!                               0794
        source ← LSEL("#NAME") <"as">                               0795
        ent ← uprogtypes                                             0796
        CONFIRM xpinstitute( source, ent );                          0797
COMMAND %detach%                                                 03508
    zpdetach =                                                      03509
        "DETACH"                                                     03510
        "SUBSYSTEM"!LL!                                             03511
        grammar ← NULL                                              03512
        param ← nlssubs                                             03513
        CONFIRM xpdetach( param, grammar );                          03514
COMMAND %deinstitute%                                             0798
    zpdeinstitute =                                               0799
        "DEINSTITUTE"                                              0800

```

```

ent ← uprogtypes                                0801
CONFIRM xpdeinstitute( ent );                    0802
COMMAND %delete%                                  0803
  zpdelete =                                      0804
    "DELETE"!LL!                                  0805
  ent ←                                           0806
    ( "ALL"!LL! <"programs in buffer">           0807
      / "LAST"!LL! <"program in buffer"> )       0808
    CONFIRM xpdelete( ent );                      0809
COMMAND %compile%                                  0810
  zpcompile =                                      0811
    "COMPILE"!LL!                                  0812
  sent ←                                           0813
    ( ("FILE"!LL! / "ASSEMBLER"!LL! <"file"> )    0814
      <"at"> source ← DSEL("#STATEMENT")          0815
      <"using"> param ← LSEL("#OLDFILELINK")      0816
      <"to file"> namfil ← LSEL("#NEWFILELINK")   0817
      / "PROCEDURE"!LL! <"at">                  0818
      param ← NULL namfil ← NULL                0819
      source ← DSEL("#STATEMENT")               0820
      / "LLO"!LL! <"user program at">           03151
      param ← NULL namfil ← NULL                03152
      source ← DSEL("#STATEMENT")               03153
      / "CONTENT"!LL! <"pattern">               01759
      param ← NULL namfil ← NULL                01760
      source ← LSEL("#CHARACTER") )             01761
    CONFIRM xpcompile( sent, source, param, namfil ); 0821
COMMAND %attach%                                  03515
  zpattach =                                      03516
    "ATTACH"!LL!                                  03517
    "SUBSYSTEM"!LL!                               03518
    grammar ← NULL                                03519
    param ← allsubs                               03520
    CONFIRM xpattach( param, grammar );          03521
END.                                               0828
% SENDMAIL SUBSYSTEM COMMANDS %                   0829
SUBSYSTEM subsendmail KEYWORD "SENDMAIL"         0830
COMMAND %structure entities%                      02676
  zjstructure =                                    02677
    param ← ("STATEMENT"/"GROUP"/"PLEX"/"BRANCH"!LL!) 0961
    <"at"> param2 ← SSEL(param) CONFIRM jsetfield(param,
    param2);                                       0962
COMMAND %author%                                  01862
  zjauthor =                                       01892
    "AUTHORS"!LL!                                  0846
    param ← LSEL("#IDENTILIST")                   0847
    CONFIRM jsetfield("#AUTHORS", param);         0848
COMMAND %comment%                                  02650
  zjcomment =                                      02651
    "COMMENT"!LL!                                  0851
    param ← LSEL("#TEXT")                          0852
    CONFIRM jsetfield("#COMMENT", param);         0853
COMMAND %distribute%                              02652
  zjdistribute =                                   02653
    "DISTRIBUTE"!LL! <"for">                     03405
    param ← ("ACTION"!LL!/"INFORMATION"!LL! <"Only">) 03406

```

```

        <"to"> param2 ← LSEL("#IDENTLIST") CONFIRM
        jsetfield(param, param2) ;
COMMAND %expedite%
    zjexpedite =
        "EXPEDITE"
        jsetfield("#EXPEDITE", answ()) ;
COMMAND %file%
    zjfile =
        param ← "FILE"!LL! param2 ← DSEL("#CHARACTER?\")
        CONFIRM jsetfield(param, param2) ;
COMMAND %forward mail%
    zjforward =
        "FORWARD" <"item number">
        param ← LSEL("#NUMBER")
        <"for"> param2 ← ("ACTION"!LL! / "INFORMATION"!LL!
        <"only">)
        <"to"> param3 ← LSEL("#IDENTLIST")
        CONFIRM xjforward(param, param2, param3);
COMMAND %initialize specification%
    zjispecs =
        "INITIALIZE"
        <"specifications">
        CONFIRM xjzapworfil= / :
command %insert%
    zjinsert =
        "INSERT"
        ("LINK"!LL!
        <"to follow">
        param ← DSEL("#VISIBLE")
        CONFIRM jsetfield("#INSERT", param) /
        "STATUS"!LL! <"form to follow">
        param ← DSEL("#STATEMENT")
        param2 ← LEVADJ
        CONFIRM xjinsstatus(param, param2) );
COMMAND %interrogate%
    zjinterrogate =
        "INTERROGATE"!LL!
        CONFIRM
        CLEAR <"distribute for action to:">
        param ← LSEL("#IDENTLIST")
        jsetfield("#ACTION", param)
        CLEAR <"distribute for information-only to:">
        param ← LSEL("#IDENTLIST")
        jsetfield("#INFORMATION", param)
        CLEAR <"title:"> param ← LSEL("#TEXT")
        jsetfield("#TITLE", param)
        CLEAR <"type of source:"> (
        "MESSAGE"!LL! param ← "#STATEMENT"
        param2 ← LSEL("#TEXT") /
        param ← "FILE"!LL!
        param2 ← DSEL("#CHARACTER") /
        param ← structure
        <"at"> param2 ← SSEL(param)
        %
        / param ← "OFFLINE"!LL! <"item">
        <"located at"> param2 ← LSEL("#TEXT")

```

03409
03410
02654
02655
02181
02182
02656
02657
0859
0863
02635
02636
02637
02638
02729
02730
02639
02968
02969
02970
02971
02660
02661
0921
02979
01894
0922
0923
02980
02981
02983
02982
02662
02663
0874
01895
0890
03154
01917
03035
03155
03036
0889
01916
0875
0883
0884
0878
0880
0876
0877
03522
0885
0886

```

%
)
jsetfield(param, param2)
CLEAR <"show status?"> (answ() xjstatus() / DUMMY)
CLEAR <"send the mail now?"> (answ() xjdoit() / DUMMY) ;
COMMAND %keywords%
zjkeywords =
"KEYWORDS"!LL!
param ← LSEL("#TEXT")
CONFIRM jsetfield("#KEYWORDS", param) ;
COMMAND %message%
zjmessage =
"MESSAGE"!LL!
param ← LSEL("#TEXT") CONFIRM
jsetfield("#STATEMENT", param) ;
COMMAND %number%
zjnumber =
"NUMBER"!LL!
("ASSIGN"!LL! CONFIRM
param ← xjreserve("#JOURNAL", NULL, FALSE) /
"PREVIOUSLY"!LL! <"Reserved"> param ← LSEL("#NUMBER") )
jsetfield("#NUMBER", param) ;
COMMAND %obsolets%
zjsolets =
"OBSOLETES"!LL!
<"item number(s)"> param ← LSEL("#TEXT")
CONFIRM jsetfield("#OBSOLETES", param) ;
COMMAND %offline item%
zjoffline =
"OFFLINE"
<"item -- located at"> param ← LSEL("#TEXT") CONFIRM
jsetfield("#HARDCOPY", param) ;
COMMAND %process%
zjprocess =
"PROCESS"!LL!
<"sendmail form at">
param ← DSEL("#STATEMENT?\")
CONFIRM
xjprocess(param) ;
COMMAND %private%
zjprivate =
"PRIVATE"
CONFIRM jsetfield("#PRIVATE", TRUE) ;
COMMAND %public%
zjpublic =
"PUBLIC"
CONFIRM jsetfield("#PUBLIC", TRUE) ;
COMMAND %reserve number%
zjreserve =
"RESERVE"!LL! (
param ←
("JOURNAL"!LL! / "XDOC"!LL!)
<"numbers -- how many?"> param2 ← LSEL( "?\NUMBER")
CLEAR <"insert the number list?">

```

```

      (answ() <"to follow"> param3 ← DSEL("#VISIBLE") /
      <"No."> param3 ← NULL)
      CONFIRM xjreserve(param, param2, param3) /
param ← "RFC"!LL! <"number"> CONFIRM
CLEAR <"title"> param ← LSEL("#TEXT")
CLEAR <"author"> param2 ← LSEL("#TEXT")
CLEAR <"distribute to"> param3 ← LSEL("#IDENTLIST")

CLEAR <"online document?"> source ← answer()
CLEAR <"show status?">
      (answ() xjrfcshow(param, param2, param3, source) /
      DUMMY)
CLEAR <"insert the number list?">
      (answ() CLEAR <"to follow"> param4 ←
      DSEL("#VISIBLE") /
      <"No."> param4 ← NULL DUMMY)
CONFIRM
xjrfcreserve(param, param2, param3, source, param4)

) ;
COMMAND %rfc%
  zjrfc =
    "RFC"
    <"number"> param ← LSEL("#NUMBER")
    CONFIRM jsetfield("#RFC", param) ;
COMMAND %send%
  zjsend =
    "SEND"!LL! <"the mail"> CONFIRM xjdoit() ;
COMMAND %show status%
  zjstatus =
    "SHOW"
    ("STATUS"!LL! CONFIRM xjstatus () /
    "RECORD"!LL! <"for ident"> param ← LSEL("#IDENT")
    param2 ← "#ALL" CONFIRM xjistatus(param, param2)) ;
COMMAND %subcollections%
  zjsubcollections =
    "SUBCOLLECTIONS"
    param ← LSEL("#IDENTLIST")
    CONFIRM jsetfield("#SUBCOLLECTIONS", param) ;
COMMAND %title%
  zjtitle =
    "TITLE"!LL!
    param ← LSEL("#TEXT")
    CONFIRM jsetfield("#TITLE", param) ;
COMMAND %update%
  zjupdate =
    "UPDATE"
    <"to item number(s)"> param ← LSEL("#TEXT")
    CONFIRM jsetfield("#UPDATE", param) ;
COMMAND %unrecorded%
  zjunrecorded =
    "UNRECORDED"!LL!
    jsetfield("#UNRECORDED", answ()) ;
INITIALIZATION %get work file%
  zjinit =

```

```

        xwarning() xjloaworfil();                                02967
    END.                                                         0972
% USER-OPTIONS SUBSYSTEM COMMANDS %                             01145
SUBSYSTEM subuseroptions KEYWORD "USEROPTIONS"                 01146
INITIALIZATION                                                  01821
    zuoinit=                                                    01822
        xuoinit();                                             01823
TERMINATION                                                      01824
    zuoterm=                                                    01825
        xuoterm();                                             01826
COMMAND %control characters%                                     01147
    zuocharacters =                                           01148
        "CONTROL" <"characters for terminal">                 01149
        param ← devopt                                         01836
        CONFIRM                                               01840
        CLEAR <"control character">                          01841
        param1 ←                                              01150
            ( "CA"!LL! / "CD" / "RPT"!LL! / "INSERT"!LL! / "BC"!LL!
              / "BW" / "BS" / "LITESC"!LL! / "IGNORE" / "SC"!LL! /
              "SW" / "TAB")                                     01151
        <"character(s)"> param2 ← LSEL("#TEXT")                01152
        <"echo as"> param3 ← LSEL("#TEXT")                    01153
        CONFIRM xuocontchars(param, param1, param2, param3); 01154
        %if the user types n chars in first lit, he should type
        only one char in second lit -- subsequent chars will
        ignored%                                              01155
COMMAND %current content length%                                01156
    zuocurcon =                                                01157
        "CURRENTCONTEXT"!LL! <"length">                       01158
        param ← LSEL("#NUMBER") CONFIRM xuocurcon(param);    01159
        %number of chars to be printed for each side of CMI in
        slash command%                                        01160
COMMAND %entry subsystem%                                       03537
    zuoentry =                                                 03538
        param1 ← "ENTRY"                                       03539
        param ← NULL                                           03540
        grammar ← NULL                                         03541
        (param2 ← "SUBSYSTEM"!LL! param ← allsubs %stores into
          the variable grammar%                                03542
          / param2 ← "PROGRAM"!LL! param ← LSEL("#OLDFILELINK"))
          03543
        CONFIRM xuoinclude( param, param1,param2 );           03544
COMMAND %exclude subsystem/program%                             03545
    zuoexclude =                                               03546
        "EXCLUDE"                                             03547
        (param2 ← "SUBSYSTEM"!LL! param ← allsubs %stores into
          the variable grammar%                                03548
          / param2 ← "PROGRAM"!LL! param ← LSEL("#OLDFILELINK"))
          03549
        CONFIRM xuoexclude( param,param2 );                   03550
COMMAND %external names link file address%                     03175
    zuoextn =                                                  03176
        "EXTERNAL" <"Names Link File Address:">              03177
        param ← LSEL("#LINK")                                   03178
        CONFIRM xuoextn( param );                              03179
COMMAND %feedback%                                             01161

```



```

zuofeedback = 01162
  "FEEDBACK"!L1! 01163
    param2 ← NULL 01164
    param ← 01165
      ("VERBOSE"!L1! / 01166
        "TERSE"!L1! / 01167
        ( ("LENGTH"!L1! / "INDENTING"!L1!)) 01168
        param2 ← LSEL("#NUMBER")) 01169
    CONFIRM xufeedback(param, param2); 01170
COMMAND %hearaId% 01171
zuohearaid = 01172
  "HERALD"!L1! 01173
    param2 ← NULL 01174
    param ← ("TERSE"!L1! / 01175
      "VERBOSE"!L1! / 01176
      "LENGTH"!L1! param2 ← LSEL("#NUMBER")) 01177
    CONFIRM xuohearaid(param, param2); 01178
COMMAND %include subsystem/program% 03551
zuoinclude = 03552
  param1 ← "INCLUDE"!L1! 03553
  grammar ← NULL 03554
  param ← NULL 03555
  (param2 ← "SUBSYSTEM"!L1! param ← allsubs %stores into
    the variable grammar% 03556
  / param2 ← "PROGRAM"!L1! param ← LSEL("#OLDFILELINK"))
  CONFIRM xuoinclude( param, param1,param2 ); 03558
%COMMAND level adjust 03017
zuolevel = 03018
  "LEVELADJUST"!L1! 03019
  param ← switch 03020
  CONFIRM 03022
  xuolevel(param);% 03021
COMMAND %name delimiter defaults% 04228
zuonamed = 04229
  "NAME"!L1! <"delimiter defaults"> 04230
  <"left delimiter"> param1 ← LSEL("#CHARACTER") 04231
  <"right delimiter"> param2 ← LSEL("#CHARACTER") 04232
  xuonamed(param1,param2); 04233
COMMAND %printing parameters% 01186
zuoprint = 01187
  "PRINTOPTIONS" ( 01188
    param ← 01189
      ( 01190
        ("RIGHT"!L1! / "LEFT"!L1!) <"margin is column"> / 01191
        "BOTTOM"!L1! <"margin is line"> / 01192
        "PAGE"!L1! <"size is (lines)"> / 01193
        "INDENTING"!L1! <"per level"> ) 01194
        param2 ← LSEL("#NUMBER") / 01195
      param ← "TAB"!L1! <"stop settings"> param2 ←
      LSEL("#TEXT") ) 01196
    %TAB text is of form
    c c c c c c
    where c stands for any char except SP, or of the form
    8,16,24,... , where the number represents the column

```

```

        position (starting from 1 not 0) for the next tab
        stop% 01197
        CONFIRM xuoprint(param, param2); 01198
COMMAND %prompt mode% 01199
        zuoprompt = 01200
        "PROMPT"!L1! 01201
        param ← ("OFF"!L1! / "PARTIAL"!L1! / "FULL"!L1!) CONFIRM
        01202
        xuoprompt(param); 01203
COMMAND %recognition mode% 01204
        zuorecognition = 01205
        "RECOGNITION"!L1! <"mode"> param ← 01206
        (rmode param2 ← NULL / 01207
        "TERSE"!L1! 01208
        <"secondary mode"> param2 ← (rmode / "TERSE"!L1!))
        01209
        CONFIRM xuorecognition(param, param2); 01210
        rmode = 01211
        ("ANTICIPATORY"!L1! / "DEMAND"!L1! / "FIXED"!L1!); 01212
COMMAND %reset% 03559
        zuoreset = 03560
        "RESET" 03561
        param2 ← NULL 03562
        param ← 03563
        ( "ALL"!L1! 03564
        / "CONTROL" <"characters for terminal"> 03565
        param2 ← 03566
        ( devopt 03567
        / "ALL"!L1! ) 03568
        / "CURRENTCONTEXT"!L1! <"length"> 03569
        / "ENTRY" <"Subsystem"> 03570
        / "EXTERNAL"!L1! <"Names Link File Address"> 03571
        / "FEEDBACK"!L1! 03572
        param2 ← 03573
        ("MODE"!L1! / "LENGTH"!L1! / "INDENTING"!L1! )
        03574
        / "HERALD"!L1! 03575
        param2 ← ("MODE"!L1! / "LENGTH"!L1! ) 03576
        / "DEFAULT"!L1! <"Subsystems and Programs"> 03577
        / "NAME"!L1! <"delimiters"> 04235
        / ("RETURN"!L1! / "FILEReturn") 03578
        <"ring entries"> 03580
        / "PRINTOPTIONS" 03581
        param2 ← 03582
        ( ("RIGHT"!L1! / "LEFT"!L1! / "BOTTOM"!L1! )
        03583
        <"margin"> 03584
        / "PAGE"!L1! <"size"> 03585
        / "INDENTING"!L1! <"per level"> 03586
        / "TAB"!L1! <"stop settings"> 03587
        ) 03588
        / "PROMPT"!L1! <"mode"> 03589
        / "RECOGNITION"!L1! <"mode"> 03590
        / "UNIVERSAL"!L1! <"Subsystem"> 03591
        / "STARTUP"!L1! <"Commands Branch Address"> 03592
        / "VIEWSPECS"!L1! 03593

```

```

)
CONFIRM xuoreset(param, param2);
COMMAND %return/filereturn ring size%
zuoringsize =
  param ← ("RETURN" / "FILERETURN")
  <"ring entries">
  param2 ← LSEL("#NUMBER")
CONFIRM xuoringsize(param, param2);
COMMAND %show%
zuoshow =
  "SHOW"!LL!
  param1 ← NULL
  param ←
    ("ALL"!LL!
    / ( "CONTROL" <"characters for terminal">
      param1 ←
        ("ALL"!LL!
        / DEVOPT)
    )
    / "CURRENTCONTEXT"!LL!
    / "ENTRY" <"subsystem">
    / "EXTERNAL"!LL! <"Names Link File Address">
    / "DEFAULT"!LL! <"Subsystems and Programs">
    / "FEEDBACK"!LL!
    / "HERALD"!LL!
    / "RETURN" <"ring sizes">
    / "LEVELADJUST"!LL!
    / "NAME"!LL! <"delimiter defaults">
    / "PRINTOPTIONS"
    / "PROMPT"!LL!
    / "RECOGNITION"!LL!
    / "STARTUP"!LL! <"Commands Branch Address">
    / "UNIVERSAL"!LL! <"Subsystem">
    / "VIEWSPECS"!LL!
  )
CONFIRM zuoshow(param, param1);
COMMAND %startup%
zuostartup =
  "STARTUP" <"Commands Branch Address">
  param ← LSEL("#LINK")
CONFIRM xuostup( param );
%COMMAND universal subsystem
zuouniversal =
  param1 ← "UNIVERSAL"
  grammar ← NULL
  param ← NULL
  (param2 ← "SUBSYSTEM"!LL! param ← allsubs %%stores into
  the variable grammar%%
  / param2 ← "PROGRAM"!LL! param ← LSEL("#OLDFILELINK"))
CONFIRM xuoinclude( param, param1,param2 );%
COMMAND %viewspecs%
zuoviewspecs =
  "VIEWSPECS"!LL!
  param1 ← NULL
  param ←

```

```

( "DEFAULT"!LL!                                03013
  param1 ← VIEWSPECs                             01216
  %/ switch %) % removed until it functions correctly%
CONFIRM xuoviewspecs(param, param1);            03014
END.                                             01217
% UTILITY SUBSYSTEM COMMANDS %                  01248
SUBSYSTEM subxxx KEYWORD "XXX"                 03068
%INITIALIZATION                                03069
  zxinit=                                       03070
  xutilinit();                                 03071
TERMINATION                                     03072
  zxterm=                                       03073
  xuquit();%                                    03074
COMMAND %run utility functions %               03075
  zxrun =                                       03080
  "RUN"!LL!                                     03081
  ( "JOURNAL"!LL!                               03082
    CONFIRM                                     03083
    xjouutil()                                 03084
  / ("TASKS"!LL!                                03085
    param ← #"FILE"                            03086
    ( readconfirm()                             03087
      / param ←                                  03088
        ( "DETACH"!NCTD LL!                     03089
          / "FILE"!LL!                           03090
          / "TTY"!LL!                             03091
          CONFIRM)                               03092
        xutil (param));                         03093
    CONFIRM)                                     03094
  )
COMMAND %set utility options %                 03095
  zxset =                                       03096
  "SET"!LL!                                     03097
  ( "JOURNAL"!LL! <"options"> setbkup()        03098
    param ←                                     03099
    ( "AUTO"!LL! <"hard copy distribution">    03100
      / "CONTINUE"!LL!                          03101
      / "ON"!LL! <"Line Journal Delivery">     03102
      / "RECOVER"!LL! <"files">                03103
      / "SLINKER"!LL!                           03104
      / "UPDATE"!LL! <"initial files">         03105
      CONFIRM                                    03106
      xsetjou(param)                             03107
    / "PARTIAL"!LL! <"Delivery options">       03108
      param ←                                     03109
      ( "CLEAR"!LL! <"options">                03110
        param1 ← NULL                            03111
        / "IDENTS"!LL! <"for on-line delivery"> 03112
        param1 ← LSEL("#IDENTLIST")             03113
        / "FILES"!LL! <"for input">            03114
        param1 ← LSEL("#TEXT")                  03115
        / "NUMBER"!LL! <"of documents separated by spaces">
          param1 ← LSEL("#TEXT") )              03116
      CONFIRM                                    03117
      xsetpar(param1, param)                     03119
    / "PRIORITY"                                03120
  )

```

```

        <"Value">                                03121
        param ← LSEL("#NUMBER")                  03122
        CONFIRM                                  03123
        xpriority(param)                         03124
    / "LOAD"!ll! <"average">                     03125
        param ← LSEL("#NUMBER")                  03126
        CONFIRM                                  03127
        xsetld(param));                          03128
COMMAND %unlock%                                03129
    zxjunlock =                                  03130
        "UNLOCK"!ll! "JOURNAL"!ll! <"password:"> 03131
        param ← LSEL("#VISIBLE")                 03132
        CONFIRM xjlock(param, FALSE);            03133
% NOT IMPLEMENTED (dash substituted for percent) 03524
COMMAND -print-                                  03134
    zxjprint =                                    03135
        "PRINT"!ll! "HARDCOPY"!ll! -wheel()-    03136
        <"password:"> param ← LSEL("#VISIBLE") CONFIRM 03137
        xjpriharcop(param);                      03138
%                                                 03525
COMMAND %lock%                                   03139
    zxjlock =                                     03140
        "LOCK" "JOURNAL"!ll! <"password:">      03141
        param ← LSEL("#VISIBLE")                 03142
        CONFIRM xjlock(param, TRUE);            03143
COMMAND %load%                                   03920
    zxload =                                      03921
        "LOAD"!ll!                               03922
        ent ← "BUSY"!ll! <"file">               03923
        namfil ← LSEL("#OLDFILELINK") CONFIRM   03924
        xload(ent, namfil);                     03925
COMMAND % enter ddt %                            03250
    zxddt =                                       03251
        "TENEX"!ll! <"Ddt">                     03252
        CONFIRM xxddt();                         03253
COMMAND %detach%                                 03361
    zxdetach =                                    03362
        "DETACH"!ll!                             03363
        param ← FALSE param2 ← FALSE           03364
        <"Infile?">                              03365
        ( "NO"!ll!                               03366
        / "YES"!ll! param ← LSEL( "#OLDFILELINK" ) 03367
        )                                         03368
        <"Outfile?">                             03369
        ( "NO"!ll!                               03370
        / "YES"!ll! param2 ← LSEL( "#NEWFILELINK" ) 03371
        )                                         03372
        CONFIRM xxdetach( param, param2 );      03373
COMMAND %check results of running tasks%        03374
    zxcheck =                                     03375
        "CHECK"!ll! <"Results of Tasks">       03376
        CONFIRM xxcheck();                       03377
END.                                             03144
% CALCULATOR SUBSYSTEM COMMANDS %              03988
SUBSYSTEM subcalculator KEYWORD "CALCULATOR"  03989
INITIALIZATION                                  03990

```

```

zcinit=                                03991
  xcalcinit();                          03992
TERMINATION                              03993
  zcterm=                                03994
  xcquit();                              03995
REENTRY                                  03996
  zclcrent =                             03997
  xscreenter();                          03998
                                          03999
COMMAND zcshow =                         04000
  "SHOW"                                  04001
  ("ACCUMULATOR"!LL! <"Registers">     04002
  CONFIRM                                 04003
  xcshoaccums()                          04004
  /"FILE" !LL! NOTT!<"in window">      04005
  DSEL(#"EDGE")                          04006
  CONFIRM                                 04007
  xcshofil()                             04008
);                                         04009
COMMAND zadd =                            04010
  ("ADD"!LL! / "+"!LL! )                04011
  param2 ← LSEL(#"NUMBER")              04012
  CONFIRM                                 04013
  % execute ADD (param2) %               04014
  xcarith(#"ADD",param2)                04015
  cfeedback();                          04016
COMMAND zcclear =                        04017
  "CLEAR"!LL!                            04018
  ("ACCUMULATOR" !LL!                  04019
  CONFIRM                                 04020
  xclraccum()                            04021
  cfeedback()                            04255
  /"FILE" !LL!                          04022
  CONFIRM                                 04023
  xclrfil()                              04024
);                                         04025
COMMAND zdivide =                        04026
  ("DIVIDE"!LL! / "/"!LL! )             04027
  param2 ← LSEL(#"NUMBER")              04028
  CONFIRM                                 04029
  % execute DIVIDE (param2) %           04030
  xcarith(#"DIVIDE",param2)            04031
  cfeedback();                          04032
                                          04033
COMMAND zevaluate =                      04034
  "EVALUATE"                              04035
  param ← LSEL(#"TEXT")                 04036
  xceval(param)                          04037
  (("ADD"!LL! / "+"!LL! / CONFIRM)      04038
  param ← #"ADD"                          04039
  /("DIVIDE"!LL! / "/"!LL! )            04040
  param ← #"DIVIDE"                      04041
  /("MULTIPLY"!LL! / "*"!LL! / "X"!LL!) 04042
  param ← #"MULTIPLY"                    04043
  /("SUBTRACT"!LL! / "-"!LL! )          04044
  param ← #"SUBTRACT" )                 04045

```

```

CONFIRM                                04046
xcevend(param)                         04047
cfeedback();                           04048
COMMAND zcformat =                      04049
  "FORMAT"!LL!                          04050
  ("PLACES"!LL! <"to the">             04051
    param ←                             04052
      ("RIGHT"!LL!                     04053
      /"LEFT"!LL!                       04054
      )                                  04055
    param2 ← LSEL("#NUMBER")           04056
      CONFIRM                           04057
      xfdigits(param,param2)           04058
  /"COMMAS"!LL!                         04059
    param ← answer()                   04060
    CONFIRM                              04061
    xcomma(param)                       04062
  /param ←                               04063
    ("LEFT"!LL!                         04064
    /"RIGHT"!LL!)                       04065
    <"justify">                         04066
    CONFIRM                              04067
    xcjust(param)                       04068
  /"TERSE"!LL NOTD! <"output">         04069
    param ← answer()                   04070
    CONFIRM                              04071
    xcfeedb(param)                      04072
  /"DOLLAR"!LL! <"signs">             04073
    param ← answer()                   04074
    CONFIRM                              04075
    xdollar(param)                      04076
  ) cfeedback();                        04077
COMMAND zcinsert =                      04078
  "INSERT"!LL!                          04079
  <"accum following">                   04080
  level ← NULL                           04081
  (ent ← ( text1 / "LINK"!LL! / "NUMBER"!LL! ) 04082
    dest ← DSEL(ent)                    04083
  / ent ← "TEXT"!LL!                     04084
    dest ← DSEL("#CHARACTER")           04085
  / ent ← structure                       04086
    dest ← DSEL("#STATEMENT")           04087
    level ← LEVADJ)                     04088
  CONFIRM                                04089
  % execute INSERT ACCUM (dest, param)  % 04090
  xinsert(ent, dest, level,xcinsac());  04091
COMMAND zmultiply =                     04092
  ("MULTIPLY"!LL! / "*"!LL! / "X"!LL! ) 04093
  param2 ← LSEL("#NUMBER")               04094
  CONFIRM                                04095
  % execute cmult (param2) %             04096
  xcarith("#MULTIPLY",param2)           04097
  cfeedback();                           04098
COMMAND zsubtract =                     04099
  ("SUBTRACT"!LL! / "-"!LL! )           04100
  param2 ← LSEL("#NUMBER")               04101

```

```

CONFIRM                                04102
% execute SUBTRACT (param2)           % 04103
xcarith("#SUBTRACT",param2)           04104
cfeedback();                           04105
COMMAND zcreplace =                    04106
  "REPLACE"!L1!                         04107
  dent ← editentity <"at">              04108
  dest ← DSEL (dent)                   04109
  sent ← #"TEXT"                        04110
  <"by accumulator">                   04111
CONFIRM                                04112
% execute replace                       % 04113
xreplace(dent,dest,sent,xcinsac());    04114
COMMAND ztotal =                        04115
  "TOTAL"!L1!                           04116
CONFIRM                                04117
xctotl()                                04118
;                                        04119
COMMAND zuse =                           04120
  "USE"!L1!                               04121
  ("ACCUMULATOR" !L1! <"number">       04122
   param2 ← LSEL("#NUMBER")             04123
   CONFIRM                               04124
   xcuseaccum(param2)                   04125
  /"SAVED" !L1! <"Accumulators">       04127
   CONFIRM                               04128
   xcusesaved()                         04129
  )cfeedback();                          04130
COMMAND zwrite =                          04131
  "WRITE"!L1! <"new">                   04132
  "FILE"!L1!                              04133
  param ← LSEL("#NEWFILELINK")          04134
CONFIRM                                  04135
xcwritef(param)                          04136
;                                        04137
COMMAND znumber=                          04138
locknum() %see if its another number % 04139
param2 ← LSEL("#NUMBER")                 04140
CONFIRM                                  04141
xcarith("#ADD",param2)                   04142
cfeedback();                             04143
                                         04144
END.                                       04145
% SYNTAX GENERATING SUBSYSTEM %          04158
SUBSYSTEM subsyntax KEYWORD "SYNTAX GENERATOR" 04159
COMMAND zkshow =                          04160
  "SHOW"!L1!                              04161
  grammar ← NULL                          04162
                                         04163
  ( param ← "COMMAND"!L1! <"in subsystem"> 04164
    param1 ← allsubs                       04165
    param1 ← grammar                       04166
    xkbuild( param1 )                      04167
    syncur ← NULL                          04168
    synsubs                                 04169
    param1 ← syncur                        04170
  )

```



```

/ param ← "RULE"!L1!                                04171
  param1 ← LSEL("#WORD")                              04172
/ param ← "SUBSYSTEM"!L1!                            04173
  param1 ← allsubs                                    04174
  param1 ← grammar                                    04175
)                                                      04176
param2 ← #"ALL"                                       04177
%                                                      04215
( lookconfirm()                                       04178
/ param2 ←                                           04179
  ( "DNLS"!L1! <"Commands">                         04180
  / "TNLS"!L1! <"Commands">                         04181
  / "ALL"!L1! <"Commands"> )                       04182
)                                                      04183
%                                                      04216
. CONFIRM                                             04184
  xkshow( param, param1, param2);                    04185
COMMAND xkcopy =                                     04186
"COPY"!L1!                                          04187
grammar ← NULL                                       04188
                                                       04189
  ( param ← "COMMAND"!L1! <"in Subsystem">          04190
    param1 ← allsubs                                  04191
    param1 ← grammar                                  04192
    xkbuild( param1 )                                04193
    syncur ← NULL                                    04194
    synsubs                                         04195
    param1 ← syncur                                  04196
  / param ← "RULE"!L1!                                04197
    param1 ← LSEL("#WORD")                            04198
  / param ← "SUBSYSTEM"!L1!                          04199
    param1 ← allsubs                                  04200
    param1 ← grammar                                  04201
  )                                                    04202
<"to follow"> dest ← DSEL("#STATEMENT")            04203
level ← LEVADJ                                       04204
param2 ← #"ALL"                                       04205
%                                                      04217
( lookconfirm()                                       04206
/ param2 ←                                           04207
  ( "DNLS"!L1! <"Commands">                         04208
  / "TNLS"!L1! <"Commands">                         04209
  / "ALL"!L1! <"Commands"> )                       04210
)                                                      04211
%                                                      04218
CONFIRM                                             04212
  xkcopy( param, param1, param2, dest, level);       04213
END.                                                 04214
FINISH OF NLSLANGUAGE                                01249
FILE nlsident % CML <programs>identification.cml %% (CML,)
(programs,identification.cml,) %
% DECLARATIONS %
  DECLARE PARSEFUNCTION
    answ, % reads answer construct % 03641
    answer, % for questions - returns 0/1 % 03668
    sp, % reads next char, TRUE if space % 03669

```

```

readconfirm, % reads next char if ca % 03673
readpue, % reads next char if BUG % 03674
readoption, % TRUE if next char is optchar % 03675
readrepeat, % TRUE if next char is repeat % 03676
lookansw, % TRUE if next char is Y/CA % 03677
lookconfirm, % TRUE if next char is CA/REPEAT/INSERT % 03678
lookbug, % TRUE if next char is BUG % 03679
locknum, % TRUE if next char is a number % 03680
clearname, % clears the name area % 03681
notca; % reads next char, TRUE iff not CA char
% 03682
DECLARE EXT-KEYWORD % so only one copy exists in system These
keywords are defined as external strings in CONST. % 03684
% TEXTUAL ENTITIES % 03690
"CHARACTER", 03691
"INVISIBLE", 03692
"LINK", 03693
"NUMBER", 03694
"PASSWORD", 03695
"TEXT", 03696
"VISIBLE", 03697
"WORD", 03698
% MISC. ENTITIES % 03699
"FILE", 03700
"OLDFILELINK", 03701
"NEWFILELINK", 03702
"NAME", 03703
"RETURN", 03704
"FILERETURN", 03705
"IDENT", 03706
"IDENTLIST", 03707
"EDGE", 03708
"MARKER"; 03709
DECLARE EXTERNAL % not defined here see (nls,pdata,) % 03710
allsubs, 03711
nissubs; 03712
% IDENTIFICATION SUBSYSTEM COMMANDS % 02313
SUBSYSTEM subident KEYWORD "IDENTIFICATION" 02314
COMMAND %verify (record/ident-file)% 02777
ziverify = 02778
"VERIFY"!L1! <"Master Ident-File"> iwheel() param ←
#"EVERYTHING" param2 ← #"YES" 02779
/ 02781
( 02782
param ← "INDIVIDUAL"!L1! <"branch"> / 02783
param ← "GROUP"!L1! <"branch"> / 02784
param ← "USED"!L1! <"idents"> / 02785
param ← "ORGANIZATION"!L1! <"branch">/ 02786
param ← "EVERYTHING"!L1!/ 02787
"SINGLE"!L1! <"record for"> 02788
param ← LSEL("#IDENT") 0278=
) 02790
CLEAR 03027
<"continue on errors?"> param2 ← answer() 02791
/ 02792
CONFIRM CLEAR <"type Control-O to abort, Control-S

```

```

        for status"> xiverfil(param, param2);                                02793
COMMAND %update ident-file%                                                02758
    ziupdate =                                                                02759
        "UPDATE"!LL! <"master Ident-File"> anychanges()                    02760
        param ← TRUE [<"later"> param ← FALSE/                             02796
        CONFIRM xiupdate(param);                                           02797
COMMAND %type%                                                                02406
    zitype =                                                                  02407
        param ← "TYPE"!LL! <"of organization"> iwheel()                    02408
        param2 ← ( "USER"!LL! / "SERVER"!LL! / "TIP"!LL! /
        "INDEPENDENT"!LL! / "ASSOCIATE"!LL!)                                02409
        CONFIRM setifield(param, param2);                                    02410
COMMAND %subcollections%                                                    02403
    zisubcol =                                                                02404
        param ← "SUBCOLLECTIONS" zinewidlist;                              02405
COMMAND %status of loaded record%                                           02399
    zistatus =                                                                02400
        "STATUS" <"of loaded record">                                       02401
        CONFIRM xistatus(#"LOADED", #"ALL");                                02402
COMMAND %show record%                                                       02394
    zisnow =                                                                  02395
        "SHOW"!LL! "RECORD"!LL! <"for ident"> param ←
        LSEL(#"IDENT") param2 ← #"ALL" [<"field:"> fieldnames/
                                                                 02396
        CONFIRM xistatus(param, param2);                                    02397
    fieldnames =                                                            03028
        param2 ← "APPROVE"!LL! /param2 ← "CAPABILITIES" /param2
        ← "COORDINATOR"!LL! /param2 ← "COMMENTS" /param2 ←
        "DELIVERY"!LL! <"modes"> /param2 ← "EXPAND"!LL!
        <"references"> /param2 ← "FUNCTION"!LL! /param2 ←
        "GROUPS"!LL! /param2 ← "IDENT"!LL! / "NLS" <"mail"> (
        "HOST"!LL! param2 ← #"NLSHOST" / "USER"!LL! <"name">
        param2 ← #"NLSADD") / "NET" <"mail"> ( "HOST"!LL! param2
        ← #"NETHOST" / "USER"!LL! <"name"> param2 ← #"NETADD")
        /param2 ← "HARDCOPY"!LL! <"mail address"> /param2 ←
        "NAME"!LL! /param2 ← "MEMBERSHIP"!LL! /param2 ←
        "ORGANIZATION"!LL! /param2 ← "PHONE"!LL! /param2 ←
        "SECONDARY" <"organization"> /param2 ←
        "SUBCOLLECTIONS"!LL! /param2 ← "TYPE"!LL! <"of
        organization">;                                                    03029
COMMAND %secondary organization%                                            02391
    zisecorg =                                                                02392
        param ← "SECONDARY" <"organization"> zinewtext;                    02393
COMMAND %remove%                                                            02689
    ziremove =                                                                02690
        "REMOVE"!LL!                                                       02691
        ( param ← "RECORD"!LL! <"for"> iwheel() param2 ←
        LSEL("IDENT") /                                                    02696
        param ← "FIELD"!LL! canmodify()                                    02697
        ( param2 ← "COMMENTS"!LL! /                                       02698
        param2 ← "FUNCTION"!LL! /                                         02702
        "MAIL"!LL!                                                         02706
        <"address for delivery type"> param2 ←
        newdelivery /                                                    02714
        param2 ← "PHONE"!LL! /                                             02707
        param2 ← "SECONDARY" <"organization"> /                            02708

```

```

        param2 ← "SUBCOLLECTIONS"!LL! ))      02709
    CONFIRM                                  02716
    xidelete (param, param2);                02715
COMMAND %phone%                             02388
    ziphone =                               02389
        param ← "PHONE"!LL! zinewtext;      02390
COMMAND %organization%                      02385
    ziorg =                                  02386
        param ← "ORGANIZATION"!LL! zinewid; 02387
COMMAND %name%                              02368
    ziname =                                02369
        "NAME"!LL! canmodify() CONFIRM CLEAR 02370
        <"old:"> shoifield("#NAME")          02914
        ( isindividual()                    02371
          CLEAR                              02372
          <"last name:"> param ← LSEL("#TEXT") 02373
          (isnewrec() CLEAR                 02906
            (checkname(param)               03146
              <"is the desired person already in the
              Master Ident-File?">          02838
                (answ() abort()/ DUMMY)     02907
              / DUMMY )                     03147
            / DUMMY )                       03145
          CLEAR                              02374
          <"first name or initial:"> param2 ← LSEL("#TEXT") 02375
        CLEAR                               02376
          <"middle initial or name:"> param3 ← LSEL("#TEXT") 02377
        / CLEAR <"new:"> param ← LSEL("#TEXT") param2 ← NULL 02378
        param3 ← NULL )                     02379
        setifield("#NAME", param, param2, param3);
COMMAND %membership%                        02358
    zimembership =                          02359
        param ← "MEMBERSHIP"!LL! canmodify() <"old:">
        shoifield(param) PERFORM zinewmem UNTIL (<"finished?">
        answ());                            02360
    zinewmem =                               02361
        CLEAR                               02899
        ("ADD"!LL! <"member(s)">           02362
          param2 ← LSEL("#IDENTLIST") CONFIRM 02363
          xiaddnem(param, param2) /         02364
          "DELETE"!LL! <"member(s)">      02365
          param2 ← LSEL("#IDENTLIST") CONFIRM 02366
          xidelmem(param, param2));        02367
COMMAND %mail address%                      02350
    zimail =                                 02351
        "MAIL" <"address type:"> canmodify() ( 02352
          "HARDCOPY"!LL!                   02353
          <"old:"> shoifield("#USADD") CLEAR <"new:"> param
          ← LSEL("#TEXT") CONFIRM setifield("#USADD", param)
          /                                  02917
          "NETWORK"!LL!                    02946
          param3 ← #"NETHOST" param4 ← #"NETADD" onlineadd /
          "NLS"                             02948
          02947

```

```

        param3 ← #"NISHCST" param4 ← #"NLSADD" onlineadd);
onlineadd = %assumes param3 and param4 are set up%
<"old host:"> shoifield (param3) CLEAR
<"new host:"> param2 ← LSEL( #"TEXT") CLEAR
<"old user name:"> shoifield(param4) CLEAR
<"new user name:"> param ← LSEL( #"TEXT")
CONFIRM setifield (param3, param2) setifield (param4,
param);
COMMAND %load record%
ziload =
"LOAD"!LL! <"record for"> param ← LSEL( ?\IDENT")
CONFIRM xiloadrecord(param);
COMMAND %independent%
ziindependent =
param ← "INDEPENDENT" <"(not associated with an
organization)"> canmodify() CONFIRM
CLEAR <"Hardcopy mail address:"> param ← LSEL( #"TEXT")
CONFIRM
setifield( #"ORGANIZATION", #"INDEPENDENT")
setifield( #"USADD", param);
COMMAND %ident%
ziident =
param ← "IDENT"!LL! (
isnewrec()
(
noident() CLEAR <"assigning tentative ident.
Please wait.">
asstentid() <"ident is"> shoifield( #"IDENT")
CLEAR <"is this ident acceptable?">
(answ() / PERFORM getid UNTIL
(okident(param)) /
% ← "current ident:"> shoifield( #"IDENT")
PERFORM getid UNTIL (okident(param))
)
/ iwheel() ipassword
<"current ident:"> shoifield( #"IDENT")
PERFORM getid UNTIL (okident(param))
)
setifield( #"IDENT", param);
getid =
CLEAR <"type desired ident:"> param ← LSEL( #"TEXT");
ipassword =
<"password:"> xipassword(LSEL( #"PASSWORD"));
COMMAND %groups%
zigroups =
param ← "GROUPS" iwheel() canmodify() <"old:">
shoifield(param) PERFORM zinewmem UNTIL (<"finished?">
answ());
COMMAND %function%
zifunction =
param ← "FUNCTION"!LL! zinewtext;
COMMAND %expand references%
ziexpand =

```

```

    param ← "EXPAND" canmodify() setifield("#EXPAND",
    answer());
COMMAND %delivery%
    zidelivery =
        param ← "DELIVERY"!LL! canmodify()
        <"old:"> showfield(param) CLEAR
        <"new:"> param2 ← newdelivery CONFIRM
        setifield(param, param2);
    newdelivery =
        "NETWORK"!LL! / "NLS" / "HARDCOPY"!LL!;
COMMAND %coordinator%
    zicord =
        param ← "COORDINATOR"!LL! zinewid;
COMMAND %comments%
    zimisc =
        param ← "COMMENTS" zinewtext;
COMMAND %capabilities%
    zicapabilities =
        param ← "CAPABILITIES" iwheel() zinewtext;
COMMAND %approve record%
    ziapprove =
        "APPROVE" <"loaded record"> iwheel()
        CONFIRM setifield("#APPROVE", TRUE);
COMMAND %add new record%
    ziadd =
        "ADD"!LL! <"new record for a(n)">
        param ← ( "INDIVIDUAL"!LL! / "ORGANIZATION"!LL! /
        "GROUP"!LL!)
        CONFIRM newrec() setifield("#RTYPE", param);
    zinewtext = %assumes param set up%
        canmodify() showold param2 ← #"TEXT" getnew
        CONFIRM setifield(param, param2);
    zinewidlist = %assumes param set up%
        canmodify() showold param2 ← #"IDENTLIST" getnew
        CONFIRM setifield(param, param2);
    zinewid = %assumes param set up%
        canmodify() showold param2 ← #"IDENT" getnew
        CONFIRM setifield(param, param2);
    showold = CLEAR <"old:"> showfield(param);
    getnew = CLEAR <"new:"> % (
        readoption() <"use field"> fieldnames <"from record">
        param2 ← xistatus ( LSEL("#IDENT"), param2) / %
        param2 ← LSEL(param2)%);
INITIALIZATION %load ident-file%
    ziinit = xiloadfil();
END.
FINISH
% READMAIL SUBSYSTEM COMMANDS %
SUBSYSTEM subreadmail KEYWORD "READMAIL"
COMMAND %verbose view%
    zrverbose =
        "VERBOSE"!LL! <"view for category">
        param ← LSEL("#TEXT")
        CONFIRM xjshwcat(param, #"VERBOSE") ;
COMMAND %undelete%
    zrundelete =

```

```

"UNDELETE"!L1!                                02614
  items                                          02615
  <"in all categories?">                        02616
    param2 ← answer()                          02617
  CONFIRM xjdelitem(param, param2) ;           02618
COMMAND %show%                                  02751
  zrshow =                                      02752
  "SHOW"!L1!                                    02611
    items CONFIRM xjshwitem (param) ;          02612
    %ALL is a keyword%                          02613
COMMAND %resend%                                02745
  zrresend =                                    02746
  "RESEND"!L1!                                  02606
    items                                        02607
    <"when?"> param2 ← LSEL("#TEXT") CONFIRM    02608
    %Will accept days of week or date%         02609
    xjremind(param, param2) ;                  02610
COMMAND %output%                                02735
  zroutput =                                    02736
  "OUTPUT"                                       02741
    items                                        02742
    <"to printer"> CONFIRM xjprinter(param) ;   02743
    %ALL will print whole category%            02744
COMMAND %omit%                                   02737
  zromit =                                       02738
  "OMIT"!L1!                                    02592
    param2 ← NULL                               02593
    param3 ←                                     02594
    ("AUTHORS"!L1! param ← LSEL("#TEXT") /      02595
    %ALL allowed%                               02596
    "TITLEWORDS"!L1! param ← LSEL("#TEXT") /    02597
    %ALL allowed%                               02598
    "DATES"!L1! <"from"> param ← LSEL("#TEXT") <"to">
    param2 ← LSEL("#TEXT") )                    02599
    %null date in from means nolimit, on to means
    today%                                       02600
    xjomit(param, param2, param3) ;             02601
COMMAND %next%                                   02739
  zrnext =                                       02740
  "NEXT"!L1!                                    02590
    <"item"> CONFIRM xjshwitem("#NEXT") ;        02591
COMMAND %interrogate%                           02731
  zrinterrogate =                               02732
  "INTERROGATE"!L1!                             02583
    CONFIRM CLEAR                               02584
    <"category:">                               02585
    param ← LSEL("#LINK")                       02586
    xjsetcurcat(param)                          02587
    PERFORM donext UNTIL (xjlast()) ;           02588
    %xjlast returns TRUE if current item is last in
    category%                                    02589
COMMAND %file%                                   02721
  zrfile =                                       02722
  "FILE"!L1!                                     02575
    items <"under category">                    02576
    param2 ← LSEL("LINK")                       02577

```

```

CONFIRM xjfileitem (param, param2) ; 02578
COMMAND %expunge% 02723
  zrexpunge = 02724
    "EXPUNGE"!LL! <"all deleted items"> 02573
    CONFIRM xjexpunge() ; 02574
COMMAND %delete% 02725
  zrdelete = 02726
    "DELETE"!LL! 02566
    items 02567
    <"from all catagories?"> 02568
    param2 ← answer() 02569
    CONFIRM xjdelitem(param, param2) ; 02570
COMMAND %catagory% 02727
  zrcatagory = 02728
    "CATAGORY"!LL! 02562
    param ← LSEL("#LINK") 02563
    %NEW is built in catagory (default). what about
    AUTHOR!IDENT, CHRON, KEYWORDS!keyword, TITLE!WORD ??% 02564
CONFIRM xjsetcurcat(param) ; 02565
COMMAND %brief view% 02719
  zrbrief = 02720
    "BRIEF"!LL! <"view for catagory"> 02559
    param ← LSEL("#TEXT") 02560
    CONFIRM xjshwcat(param, "#BRIEF") ; 02561
COMMAND %accept% 02717
  zraccept = 02718
    "ACCEPT"!LL! 02549
    param2 ← NULL 02550
    param3 ← 02551
    ("AUTHORS"!LL! param ← LSEL("#TEXT") / 02552
    %ALL allowed% 02553
    "TITELWORDS"!LL! param ← LSEL("#TEXT") / 02554
    %ALL allowed% 02555
    "DATES"!LL! <"from"> param ← LSEL("#TEXT") <"to">
    param2 ← LSEL("#TEXT") ) 02556
    %null date in from means nolimit, on to means
    today% 02557
    xjaccept(param, param2, param3) ; 02558
items = 02623
  <"item number:"> param ← LSEL("#TEXT"); 02624
donext = 02627
  param ← xjshwitem("#NEXT") 02628
  <"file it?"> 02629
    (answ() <"catagory:"> param ← LSEL("#LINK")
    xjfileitem(param, param2) / param ← NULL) 02630
  <"forward it?"> 02631
    (answ() <"to:"> param2 ← LSEL("#TEXT") xjforward(param,
    param2) / param ← NULL) 02632
  <"delete it from this catagory?"> 02633
    (answ() xjdelitem(param) / param ← NULL); 02634
END. 02648

```


SYNTBL

(MLK) SYNTBL
(MLK) SYNTBL

(MLK) SYNTBL
(MLK) SYNTBL

(MLK) SYNTBL
(MLK) SYNTBL

(MLK) SYNTBL
(MLK) SYNTBL

(MLK) SYNTBL
(MLK) SYNTBL

(M
(M

```

< NLS, SYNTBL.NLS;2, >, 3-OCT-74 11:43 KEV ;;;;
FILE syntbl % <SUBSYS>TABLE <rel-nls>syntbl %% (subsys,table,)
(victor,syntbl.rel,)%
(noexpand) TABLE
  structure
  textent
  textl
  synsubs
% each line in the following tables has the following syntax:
  id := value
  id := the name of a CML
  rule, parsefunction, or xroutine
  value := str / lloId
  str := A QUOTED STRING
  which will appear when this rule/parsefunction/xroutine is
  encountered in the CML grammar
  lloId := the name of an llo procedure
  to be called when this rule/parsefunction/xroutine is
  encountered in the CML grammar
the table xoutines if for xoutines
the table prules if for rules
the table prsfuctions if for parsefunctions
%
(xoutines) TABLE
  xcomment = "TYPEIN "
  xjdae = "ADDRESS "
  subldsp = "(STRING) "
(prules) TABLE
  textent = "STRING "
  textl = "STRING "
  structure = "STRUCTURE "
  cshmode = "CASEMODE "
  nlssubs = "SUBSYSTEM "
  allsubs = "SUBSYSTEM "
  prtgrp = "WHOM "
  protl = "DOING "
  devopt = "DEVICES "
  synsubs = "COMMAND "
  jmpcoms = prjmpcmd
  hchk = prhchk
(prsfuctions) TABLE
  answ = "ANSWER "
  answer = "ANSWER "
  sp = "<SPACE> "
  readconfirm = "OK "
  readbug = "BUG "
  readoption = "OPTION "
  readrepeat = "OKREPEAT "
  lookansw = "ANSWER "
  lookconfirm = prconfirmlook
  lookbug = prbuglook
  looknum = prnumlook
FINISH
%%

```

02
03
04
05
06
049
07
08
09
010
011
012
013
014
015
016
017
018
019
020
021
022
023
024
025
026
027
028
029
030
031
032
033
050
051
034
035
036
037
038
039
040
041
042
043
044
045
046
047
048

5436D

(MLK) SYSGD
(MLK) SYSGD

(MLK) SYSGD
(MLK) SYSGD

(MLK) SYSGD
(MLK) SYSGD

(MLK) SYSGD
(MLK) SYSGD

(MLK) SYSGD
(MLK) SYSGD

(MLK) SY
(MLK) SY

```

< NLS, SYSGD.NLS;31, >, 4-OCT-74 22:51 CHI ;;;;<NLS>SYSGD.NLS;30,
7-MAY-74 19:38 JDH :
(al) (nls,resolver,)                                0482
  data declaration                                  03566
(alword) (nls,parser,)                               0124
  data declaration                                  02980
(a2word) (nls,parser,)                               0117
  data declaration                                  02972
(a3ddrec) (nls,colstr,a3ddrec) 70                   02334
  (stid)                                             05933
  add stid into vector                              05932
(a3word) (nls,parser,)                               0119
  data declaration                                  02974
(a4word) (nls,parser,)                               0191
  data declaration                                  03053
(abort) (nls,psident,abort) 14                      0372
  (resultptr,parsemode)                             03392
  abort current command specification                03391
(abortssystem) (nls,psystem,abortssystem) 3E         0400
  (errstrptr)                                        03445
  ptr to diagnostic string                           03444
(acchec) (nls,const,)                                0765
  data declaration                                  03883
(acctyp) (nls,ioexec,)                               02183
  data declaration                                  05712
(accum1) (nls,nddt,)                                 02460
  data declaration                                  06064
(acsave) (nls,pscalc,)                              0451
  data declaration                                  03523
(acsizd) (nls,pscalc,)                              0444
  data declaration                                  03513
(acsizs) (nls,pscalc,)                              0469
  data declaration                                  03547
(adadcs) (nls,records,)                              01329
  data declaration                                  04447
(adadf) (nls,records,)                              01177
  data declaration                                  04295
(adadhi) (nls,records,)                              01274
  data declaration                                  04392
(adalrx) (nls,records,)                              01374
  data declaration                                  04492
(adalry) (nls,records,)                              01104
  data declaration                                  04222
(adaseq) (nls,records,)                              01530
  data declaration                                  04648
(adasiz) (nls,records,)                              01390
  data declaration                                  04508
(adaulx) (nls,records,)                              01116
  data declaration                                  04234
(adauly) (nls,records,)                              01185
  data declaration                                  04303
(addcal) (nls,const,)                                0824
  data declaration                                  03942
(addcha) (nls,const,)                                0931
  data declaration                                  04049
(addeol) (nls,csendmail,addeol) 5B                  02681

```

(string,heading)	06437
support routine for jstatus	06436
(addfil) (nls,ioexec,addfil) 9A	02186
(jfn,astng)	05718
Add a file to file status table. Returns NLS file number if successful; calls err if global filcnt is invalid, abort if too many files are open.	05717
(addhelp) (nls,parser,addhelp) 6D2	0121
(index,helpstr,work,helpcnt)	02977
edits name and adds to helpstr	02976
(addid) (nls,cident,addid) 12C	02563
(idstr,fileno)	06215
add new ident to usedids branch	06214
(addr) (nls,parser,)	0162
data declaration	03022
(addr10) (nls,nddt,)	02416
data declaration	06021
(addrec) (nls,colstr,addrec) 15	02335
(stid)	05935
add stid into vector	05934
(adrmod) (nls,nddt,)	02393
data declaration	05998
(af) (nls,inpfbk,af) 9G2B	0497
Turn the command feedback arrow off	03589
(afgpty) (nls,cident,)	02641
data declaration	06363
(afgtyp) (nls,const,)	0572
data declaration	03690
(aheadf) (nls,pdata,)	09
data declaration	02870
(all) (nls,frontend,)	02010
data declaration	05419
(alldsp) (nls,frontend,alldsp) 5B	01964
recreate display for all display areas	05349
(allver) (nls,cident,)	02598
data declaration	06281
(alocda) (nls,dspgen,alocda) 6D	01889
(da)	05212
allocate a display area	05211
(alpha) (nls,const,)	0833
data declaration	03951
(alphof) (nls,const,)	0635
data declaration	03753
(altern) (nls,parser,)	0129
data declaration	02985
(an) (nls,inpfbk,an) 9G2A	0495
Turn the command feedback arrow on	03587
(answ) (nls,psupport,answ) 3D1	0245
(resultptr,parsemode,stringptr)	03140
ptr to help string	03139
(answer) (nls,psupport,answer) 3D2	0240
(resultptr,parsemode,stringptr)	03130
ptr to help string	03129
(anychanges) (nls,psident,anychanges) 10	0382
(resultptr,parsemode)	03412
has the user made any changes to the current record	03411

(apachr) (nls,utility,apachr) 5D	01611
(ch)	04736
passed character as argument. appends to the statement being constructed in sar.	04735
(apblnk) (nls,utility,apblnk) 8L	01643
(ast.cnt)	04770
Append cnt blanks to the designated astring.	04769
(apchr) (nls,utility,apchr) 8F4	01568
(aplit) (nls,inpfbk,aplit) 9R5	0503
(string)	03599
append contents of string to literal display area, removing file text as needed. Takes care of tabs, etc.	03598
(append) (nls,const,)	0604
data declaration	03722
(apsda) (nls,const,)	0760
data declaration	03878
(apsr) (nls,utility,apsr) 8M	01607
(asfrom,asto)	04729
The purpose of this routine is to append one A-string onto another. Arguments are two A-string addresses. The first string is appended to the second. An ERROR (Exceed Capacity) is generated if the maximum length of the latter string is too short to contain the result.	04728
(aptstr) (nls,utility,aptstr) 5E	01570
(tp)	04690
Append tstring. Argument is pointer to two T-pointers which delimit the substring to be appended to the statement currently being constructed.	04689
(archos) (nls,const,)	0947
data declaration	04065
(bo) (nls,records,)	06692
(arcopr) (nls,const,)	0825
data declaration	03943
(arcstr) (nls,const,)	0662
data declaration	03780
(argcou) (nls,parser,)	0187
data declaration	03049
(arm) (nls,inpfbk,arm) 901	0535
arm cursor	03646
(ascalt) (nls,const,)	0652
data declaration	03770
(ascbst) (nls,const,)	0950
data declaration	04068
(ascom) (nls,utility,ascom) 8G	01588
(astr1,astr2,relation)	04707
accepts addresses of 2 a-strings. returns logical value of astr1	04706
<relation> astr2	0698
(asctsw) (nls,const,)	03816
data declaration	01625
(asrref) (nls,utility,asrref) 8A	04748
(ast)	04747
create a pointer to astring with stastr field set.	0552
(assfil) (nls,const,)	03670
data declaration	01578
(assnbit) (nls,utility,assnbit) 14A1	04699
(table.length)	

(asstentid) (nls,psident,asstentid) 12	0367
(resultptr,parsemode)	03382
assign tentative ident to new record	03381
(astr) (nls,const,)	01006
data declaration	04124
(astruc) (nls,utility,astruc) 8V	01631
(astrng)	04758
a-string to upper case	04757
(auxchr) (nls,frontend,auxchr) 10B	01994
get characters for executable text	05396
(auxinf) (nls,pdata,)	066
data declaration	02927
(auxinterminate) (nls,frontend,auxinterminate) 10C	02013
terminate auxiliary input	05423
(auxstartup) (nls,frontend,auxstartup) 10A	01973
(ptr1,ptr2)	05364
pointer to TP for end of auxiliary text	05363
(b0) (nls,records,)	01489
data declaration	04607
(backup) (nls,const,)	01015
data declaration	04133
(badacc) (nls,pscalc,)	0445
data declaration	03514
(badcfi) (nls,pscalc,)	0463
data declaration	03539
(badfil) (nls,filmp, badfil) 2B2	01788
(fileno)	05022
called when find something screwed up in the file	05021
(basest) (nls,pdata,)	033
data declaration	02894
(bbound) (nls,frontend,)	01991
data declaration	05391
(bcstr) (nls,const,)	0928
data declaration	04046
(begarb) (nls,utility,begarb) 9H5	01596
(begmsg) (nls,const,)	0807
data declaration	03925
(begnod) (nls,parser,)	0163
data declaration	03023
(bfree) (nls,const,)	0835
data declaration	03953
(bfs) (nls,utility,bfs) 9F	01608
(brloc)	04731
branch false and scan	04730
(bhl) (nls,const,)	0605
data declaration	03723
(bildtabwrds) (nls,psuserop,bildtabwrds) 2B8	0402
(array.tbl)	03449
builds 3word tab table	03448
(binjsys) (nls,nddt,binjsys) 9A	02439
(bintyp) (nls,const,)	0987
data declaration	04105
(bit1) (nls,records,)	01143
data declaration	04261
(bit2) (nls,records,)	01450
data declaration	04568

(bit3) (nls,records,)	01057
data declaration	04175
(bitfil) (nls,records,)	01045
data declaration	04163
(bkbp) (nls,cedit,bkbp) 26D	01768
(bp)	04985
back up byte pointer	04984
(bkbc) (nls,utility,bkbc) 8AA	01669
(astrng)	04799
backspace char	04798
(bkjfnjsys) (nls,nddt,bkjfnjsys) 9E	02366
(bkproc) (nls,usrpgm,)	02717
data declaration	06506
(bkpw) (nls,utility,bkw) 8AB	01661
(astr)	04789
backspace word	04788
(blanks) (nls,pscalc,)	0466
data declaration	03543
(blsfnhdr) (nls,seqfil,blsfnhdr) 3J	02235
(da)	05809
This routine builds a page header for output quickprint.	05808
(blkfre) (nls,records,)	01411
data declaration	04529
(blklen) (nls,records,)	01275
data declaration	04393
(blkprd) (nls,const,)	0659
data declaration	03777
(blkpre) (nls,records,)	01256
data declaration	04374
(blksiz) (nls,const,)	0946
data declaration	04064
(blksuc) (nls,const,)	0857
data declaration	03975
(blnk) (nls,ccalc,)	02526
data declaration	06154
(bmooff) (nls,inprbk,bmooff) 9A2	0519
remove all bug marks	03623
(botact) (nls,nddt,)	02365
data declaration	05969
(boundry) (nls,frontend,boundry) 11C	02021
(dacords)	05436
find nearest window boundry	05435
(bpaddr) (nls,nddt,)	02403
data declaration	06007
(bpadr) (nls,records,)	01234
data declaration	04352
(bpbitp) (nls,records,)	01502
data declaration	04620
(bpclall) (nls,nddt,bpclall) 10B1	02382
Clear all breakpoints	05987
(bpclr) (nls,nddt,bpclr) 10B2	02450
(bpa)	06054
(bpindx) (nls,records,)	01322
data declaration	04440
(bpinst) (nls,nddt,)	02354
data declaration	05958

(bppntall) (nls,nddt,bppntall) 10B3	02437
Print all breakpoints	06041
(bpsize) (nls,records,)	01554
data declaration	04672
(bptest) (nls,nddt,)	02385
data declaration	05990
(bpval) (nls,nddt,)	02408
data declaration	06011
(branam) (nls,const,)	0923
data declaration	04041
(branch) (nls,const,)	0906
data declaration	04024
(breakpoint) (nls,nddt,breakpoint) 10Bh	02471
Get here to check whether or not to execute a break...called with a	
!JSP brkchk	06074
(breakup) (nls,stmgt,breakup) 3E	02221
(blk,zone,index)	05785
break up blk and put in index free list	05784
(brkconnection) (nls,auxcod,brkconnection) 5I	02265
break display screen connection	05845
(brnchv) (nls,const,)	0645
data declaration	03763
(bsc) (nls,utility,bsc) 5A	01666
(dest)	04796
Begin string construction. Called with STID (or pointer to string	
with stastr TRUE) of the destination.	04795
(bstr) (nls,const,)	0653
data declaration	03771
(bumpframe) (nls,nddt,bumpframe) 11B5A	02488
(rd)	06091
(bumpstr) (nls,utility,bumpstr) 8S	01627
(astr)	04752
Increments number in astr by one	04751
(c3compare) (nls,colprt,c3compare) 7F	02313
(i1,i2)	05912
(caacti) (nls,pdata,)	061
data declaration	02922
(cacfile) (nls,ccalc,cacfile) 5D	02539
get/create calc file	06170
(cachar) (nls,pdata,)	043
data declaration	02904
(cackf) (nls,records,)	01377
data declaration	04495
(caclear) (nls,ccalc,caclear) 4A	02547
(accx,clrwhat)	06184
clear accum pointed at by accx or all accums	06183
(caddexp) (nls,adrmp,caddexp) 1Q1	02072
(t1,t2,da,ptr)	05510
Core NLS Address Expression evaluation Command	05509
(cafilinitialize) (nls,ccalc,cafilinitialize) 2A	02536
calc file set up	06166
(callo) (nls,pdata,)	015
data declaration	02876
(call1) (nls,pdata,)	030
data declaration	02891
(callm) (nls,pdata,)	056

data declaration	02917
(callsig) (nls,utility,callsig) 4F	01653
(value,message)	04780
Call signal	04779
(calsys) (nls,pscalc,)	0460
data declaration	03534
(canmodify) (nls,psident,canmodify) 9	0364
(resultptr,parsemode)	03376
is the user allowed to modify this record?	03375
(canxt) (nls,records,)	01239
data declaration	04357
(cappsta) (nls,credit,cappsta) 2A	01700
(tostid,fromstid,tp1,tp2)	04856
Core NLS Append Statement Command	04855
(capptex) (nls,credit,capptex) 2B	01724
(stid,t1,t2,l1,l2)	04901
Core NLS Append Text Command	04900
(capsf) (nls,records,)	01301
data declaration	04419
(carbp) (nls,records,)	01208
data declaration	04326
(carcfil) (nls,credit,carcfil) 3A	01717
(rhostn,filename)	04887
Core NLS Archive File Command	04886
(carnc) (nls,records,)	01178
data declaration	04296
(castr) (nls,const,)	0813
data declaration	03931
(catbp) (nls,records,)	01279
data declaration	04397
(catnc) (nls,records,)	01246
data declaration	04364
(caupde) (nls,pscalc,)	0468
data declaration	03546
(cbhl) (nls,records,)	01099
data declaration	04217
(cbnxt) (nls,records,)	01425
data declaration	04543
(cbprev) (nls,records,)	01233
data declaration	04351
(cbresta) (nls,credit,cbresta) 4A	01706
(bug.rlevcnt,tp1,tp2)	04867
Core NLS Break Statement Command	04866
(ccchar) (nls,records,)	01232
data declaration	04350
(cccval) (nls,colsrt,cccval) 14	02320
(stid)	05920
(ccdevi) (nls,records,)	01354
data declaration	04472
(ccecho) (nls,records,)	01443
data declaration	04561
(ccfda) (nls,const,)	0643
data declaration	03761
(ccmpare) (nls,colsrt,ccmpare) 12	02326
(i1,i2)	05926
(ccnda) (nls,const,)	0650

data declaration	03768
(cconfildir) (nls,cedit,cconfildir) 5A	01703
(dir1,dir2,pass1,pass2)	04861
Core NLS Connect to File Directory Command	04860
(ccontrol) (nls,cclsrt,ccontrol) 18	02336
(da)	05936
(ccoparcdir) (nls,cedit,ccoparcdir) 6A	01767
(stid,rlevnt,tp1,tp2,params)	04983
Core NLS Copy Archive File Directory Command	04982
(ccopdir) (nls,cedit,ccopdir) 6B	01690
(stid,rlev,info,gropk,sortk,rhstn,fname)	04837
Core NLS Copy Directory Command	04836
(ccopfil) (nls,cedit,ccopfil) 6C	01718
(rhost1,file1,rhost2,file2,astr)	04889
Core NLS Copy File Command	04888
(ccopgro) (nls,cedit,ccopgro) 6D	01727
(tostid,rlevnt,fromstid,fromstid2,filter,vsptr)	04907
Core NLS Copy Group Command	04906
(ccopseqfil) (nls,cedit,ccopseqfil) 6E	01688
(stid,rlevnt,tp1,tp2,filtyp)	04833
Core NLS Copy Sequential File Command	04832
(ccopsta) (nls,cedit,ccopsta) 6F	01678
(tostid,rlevnt,fromstid,filter,vsptr)	04814
Core NLS Copy Statement Command	04813
(ccoptex) (nls,cedit,ccoptex) 6G	01746
(totptr,fromtptr1,fromtptr2,insertspace)	04944
Core NLS Copy Text Command	04943
(ccrefil) (nls,cedit,ccrefil) 7A	01766
(rhostn,fname)	04981
Core NLS Create File Command	04980
(cct) (nls,utilty,cct) 9P	01617
(chrcls)	04740
(cctbl) (nls,const,)	0924
data declaration	04042
(cctf) (nls,utilty,cctf) 9Q	01567
(chrcls)	04686
(cctype) (nls,records,)	01040
data declaration	04158
(ccurcon) (nls,cedit,ccurcon) 9A	01754
(tptr,string)	04960
Core NLS Current Context in File Command	04959
(ccurloc) (nls,cedit,ccurloc) 8A	01683
(tptr,string)	04823
Core NLS Current Location in File Command	04822
(cdbckc) (nls,dspgen,cdbckc) 3B5	01878
character output routine	05192
(cdelallmar) (nls,cedit,cdelallmar) 10A	01695
(fileno)	04847
Core NLS Delete All Markers Command	04846
(cdelarcfil) (nls,cedit,cdelarcfil) 10B	01738
(f1,f2)	04928
Core NLS Delete Archived File Command	04927
(cdelfil) (nls,cedit,cdelfil) 10C	01684
(rhstn,fname,astr)	04825
delete a file (group) (and its (their) partial copy(ies))	04824
(cdelgro) (nls,cedit,cdelgro) 10D	01777

(stid1,stid2,filter,vsptr)	05001
Core NLS Delete Group Command	05000
(cdelmar) (nls,credit,cdelmar) 10E	01716
(tp1,tp2,fileno)	04885
Core NLS Delete Marker Command	04884
(cdelmodfil) (nls,credit,cdelmodfil) 10F	01757
(fileno)	04966
Core NLS Delete Modifications to File Command	04965
(cdelsta) (nls,credit,cdelsta) 10G	01755
(stid,filter,vsptr)	04962
Core NLS Delete Statement Command	04961
(cdeltex) (nls,credit,cdeltex) 10H	01731
(bug1,bug2,includespace)	04914
Core NLS Delete Text Command	04913
(cdlnk) (nls,pdata,)	028
data declaration	02889
(cdotst) (nls,const,)	0563
data declaration	03701
(cdr) (nls,select,cdr) 10I	0238
(bug.ptr1,ptr2)	03126
(cdstr) (nls,const,)	0575
data declaration	03693
(ce) (nls.const,)	01001
data declaration	04119
(cexparcdir) (nls,credit,cexparcdir) 11A	01728
Core NLS Expunge Archive Directory Command	04908
(cexpcdir) (nls,credit,cexpcdir) 11B	01750
(parameters)	04952
Core expunge connected and login directories	04951
(cfeedback) (nls,pascal,cfeedback) 3E1	0450
(result,parsemode)	03522
do calculator feedback	03521
(cfl11e) (nls.parser,)	0185
data declaration	03046
(cfl12e) (nls.parser,)	0184
data declaration	03045
(cflapc) (nls,execfl,cflapc) 30	01928
(jfn1,jfn2)	05281
copy file from jfn1 to jfn2	05280
(cfldsp) (nls,inpfbk,cfldsp) 9G1F	0521
display CFLSTR in CFL area	03625
(cgtjfn) (nls,colstr,cgtjfn) 21	02329
(astrng)	05929
RETURN jfn of file indicated by astrng	05928
(changcom) (nls,auxcod,changcom) 7A	02270
(string,retparm)	05854
This procedure accepts a string as a parameter, and types out a message (dismes) saying that the command has been changed, and that the user should consult Folklore for details	05853
(charac) (nls,const,)	0699
data declaration	03817
(charv) (nls,const,)	0710
data declaration	03828
(cnbmtv) (nls,utility,)	01603
data declaration	04723
(cnbptr) (nls,utility,cnbptr) 8F3	01599

(checkjfile) (nls,ioexec,checkjfile) 15K	02130
(filename,flags,thorough)	05615
convert filename to absolute form via catalogs	05614
(checklib) (nls,intnls,checklib) 6U	02845
This procedure checks the system flags to see if it should try to start up one of the background processors automatically.	06668
(checkname) (nls,psident,checkname) 17	0373
(resultptr,parsemode,lastname)	03394
check the ident file for individuals with this last name, print brief status for each	03393
(chkbit) (nls,utility,chkbit) 14E1	01562
(table.bit,length)	04681
(chkbp) (nls,nddt,chkbp) 10B5	02481
(loc)	06084
This procedure checks the bkpt num/ addr passed for first a stack manipulation instruction, and secondly an already defined breakpoint.	06083
(chkcaer) (nls,nddt,chkcaer) 7A	02442
(chkdev) (nls,execfl,chkdev) 3D	01936
(fjfn)	05297
returns FALSE if not a supported (disk) file	05296
(chkmode) (nls,nddt,chkmode) 10J1	02395
(caflag)	06000
(chkpcf) (nls,const,)	0944
data declaration	04062
(chkpcs) (nls,execfl,chkpcs) 3E	01933
(fjfn,flags)	05291
check if * for extname and extname = "PC"	05290
(chkprm) (nls,colstr,chkprm) 9	02330
(chnfdb) (nls,ioexec,chnfdb) 13F	02178
(jfn,offset,mask,value)	05703
Return true if file fdb changed properly	05702
(chngadr) (nls,nddt,chngadr) 10A1	02454
(rd,value)	06058
(chnul) (nls,records,)	01553
data declaration	04671
(chpair) (nls,utility,chpair) 80	01600
(ast1,lower,upper,ast2)	04720
extract substring and append to another string	04719
(chprvsts) (nls,ioexec,chprvsts) 14A	02173
(fileno,newprvsts)	05693
change file's privacy status	05692
(chr0) (nls,records,)	01551
data declaration	04669
(chr1) (nls,records,)	01237
data declaration	04355
(chr2) (nls,records,)	01169
data declaration	04287
(chr3) (nls,records,)	01180
data declaration	04298
(chr4) (nls,records,)	01087
data declaration	04205
(chrct) (nls,utility,chrct) 90	01632
(char,chrcls)	04760
Character class test	04759
(chrpos) (nls,coresupport,chrpos) 2A	02223

(char, astrng)	05789
scan string for character	05788
(chrstr) (nls, psuserop, chrstr) 2D2	0427
(cc)	03486
converts internal code for character to string representation	03485
(chrtyp) (nls, const,)	0943
data declaration	04061
(cinssta) (nls, cedit, cinssta) 12B	01735
(stid, rlevcnt, tp1, tp2)	04922
Core NLS Insert Statement Command	04921
(cinstex) (nls, cedit, cinstex) 12C	01732
(tp1, tp2, tp3, insertspace)	04916
Core NLS Insert Text Command	04915
(cint1) (nls, utility,)	01648
data declaration	04773
(cint2) (nls, utility,)	01667
data declaration	04797
(cint3) (nls, utility,)	01573
data declaration	04695
(cint4) (nls, utility,)	01595
data declaration	04716
(cis) (nls, cedit, cis) 12A	01707
(stid, string, levstr)	04869
Core NLS Insert Statement command (interface for old NLS core calls)	04868
(ckbptr) (nls, nddt, ckbptr) 13A	02376
(bp)	05981
Checks the validity of a byte pointer...returns False if no good	05980
(ckcnum) (nls, catnum, ckcnum) 2B	02507
(type, number, idlist, nstid)	06123
This procedure accepts a catalog number in *number*, an identification-list in *idlist*, and searches the preassigne-number list for that number.	06122
(ckdiracc) (nls, ioexec, ckdiracc) 13A	02151
(fileno)	05652
given a file number, this routine checks that the current job has write access to the directory that the file lives in; if not, this routine goes to err with a nasty message	05651
(ckident) (nls, cident#@ckident) 3E	02567
(identsr, retinfo, fileno)	06223
validate an ident and fetch entry if desired	06222
(ckidlist) (nls, cident, ckidlist) 3C	02555
(idlist, badidlist, fileno)	06200
check if list of idents are already being used	06199
(ckilmem) (nls, cident, ckilmem) 13H	02648
(ident, l1, l2, openidfileno)	06376
verify logical membership of an ident in an ident list	06375
(ckipmem) (nls, cident, ckipmem) 13I	02589
(ident, l1, l2)	06266
verify physical membership of an ident in an ident list	06265
(cklsfreestring) (nls, dspgen, cklsfreestring) 3A3	01879
(ls)	05194
if line seg is from string allocated from dspblk then deallocate it	05193
(cknlsid) (nls, cident, cknlsid) 3A	02571
(identsr, infostr, fileno)	06231

validate ident for nls user	06230
(ckbag) (nls,const,)	0587
data declaration	03705
(ckbt1) (nls,records,)	01478
data declaration	04596
(ckbt2) (nls,records,)	01311
data declaration	04429
(ckrfcnum) (nls,catnum,ckrfcnum) 20	02510
(type,rfcnumber,number,idelist,rstid,nstid)	06128
This procedure accepts a catalog number in *number*, an RFC number	
in *rfcnumber*, an identification-list in *idelist*, and searches the	
preassigne-number list for that number.	06127
(ckstrc) (nls,verify,ckstrc) 8A	02225
(fileno)	05792
(clbuff) (nls,records,)	01036
data declaration	04154
(clcc1) (nls,records,)	01361
data declaration	04479
(clcc2) (nls,records,)	01432
data declaration	04550
(clcname) (nls,ccalc,clcname) 5E	02525
(flnam)	06153
set calculator file name to login directory CALC file	06152
(clcnt) (nls,records,)	01384
data declaration	04502
(cldsr) (nls,utility,cldsr) 5F	01665
(tp)	04794
Called with address of two tpointers. For all clist entries, if	
between these two pointers, then set to "deleted".	04793
(cldtbs) (nls,ioexec,cldtbs) 10C	02098
(dt)	05557
Clear data status table; could be combined with almost identical	
clrnsgs	05556
(cldtxt) (nls,utility,cldtxt) 5G	01644
(ptr1,ptr2)	04771
(cleanu) (nls,const,)	0920
data declaration	04038
(clearc) (nls,const,)	01016
data declaration	04134
(clearda) (nls,dspgen,clearda) 6A	01902
(da)	05233
clear display area	05232
(clearlit) (nls,anpfbk,clearlit) 9H2	0540
Clear literal area	03651
(clearname) (nls,psupport,clearname) 6A	0246
(curptr,parsemode,string)	03142
clears the name area and returns TRUE	03141
(clfixe) (nls,records,)	01051
data declaration	04169
(clflup) (nls,ioexec,clflup) 11D	02129
(oldfil,newfil,list)	05613
change clist references to oldfil to newfil (assumes list in sid	
format)	05612
(clfnol) (nls,records,)	01366
data declaration	04484
(clfno2) (nls,records,)	01303

data declaration	04421
(cline) (nls,dspgen,cline) 7H	01910
(x,y,nchar)	05248
write blanks - alphanumeric displays	05247
(clist) (nls,psupport,clist) 7A	0257
(type,fnol,fnol2)	03163
build clist	03162
(c11) (nls,records,)	01558
data declaration	04676
(cllp) (nls,utility,cllp) 4H	01677
dummy procedure for calls	04612
(clmax) (nls,records,)	01272
data declaration	04390
(clnidlst) (nls,cident,clnidlst) 13J	02630
(inlist,outlist)	06343
clean ident list -- of comments, expansion chars, etc.	06342
(cloafil) (nls,cedit,cloafil) 13A	01749
(flnam)	04950
Core NLS Load File command	04949
(cloamodfil) (nls,cedit,cloamodfil) 13B	01719
(filename)	04891
Core NLS Load Modified File command	04890
(clogout) (nls,cedit,clogout) 14A	01772
Core NLS logout command	04992
(close) (nls,ioexec,close) 5A	02137
(fileno)	05627
Close original file and the partial copy and delete from file status table. Calls err if error occurs. writeout called three times! explicitly, in clsfil, and in delfil. Clean this up.	05626
(closeall) (nls,ioexec,closeall) 5D	02150
Close all files in file status table.	05650
(closepc) (nls,ioexec,closepc) 5F	02198
(fileno)	05741
Closes the PC for the file in fileno if it exists. (Currently used only in ident system.)	05740
(closeseq) (nls,seqgen,closeseq) 3C	01853
(sw)	05144
close a sequence	05143
(closeu) (nls,ioexec,closeu) 5B	02166
(fileno)	05680
Close and update file	05679
(clpsid) (nls,ioexec,clpsid) 11C	02158
(list)	05665
change sid's in passed correspondence list to stid's. Uses lookup to find the ring element with the corresponding sid. see clhdr and clistr for format of clist.	05664
(clpsw) (nls,pdata,)	06
data declaration	02867
(clrall) (nls,dspgen,clrall) 5A	01884
(da,free)	05203
reset LSRT entries for the da	05202
(clrbuf) (nls,coresupport,clrbuf) 2B	02222
(outflg)	05787
clear the todas input and (optionally) output	05786
(clrcor) (nls,ioexec,clrcor) 10D	02187
(fileno)	05720

clear core page status table entries and remove pages from core pertaining to this file (except for the header, the address of which will be returned.). Used in initialization of file. Similar to writeout! 05719

(clrfil) (nls,ioexec,clrfil) 10A 02143
(fileno) 05638
Clear the filst entry for the given fileno. 05637

(clrnrgs) (nls,ioexec,clrnrgs) 10B 02155
(rn) 05659
Clear ring status table; could be combined with almost identical
cldtbs 05658

(clrseqda) (nls,inpfbk,clrseqda) 9H3 0486
(daid) 03573
clear sequential display area 03572

(clsfil) (nls,ioexec,clsfil) 5E 02109
(fileno,astrng) 05578
Primitive close file routine. Does not delete file from file status table, but does do writeout. The original file and the partial copy are closed by this routine. If anything goes wrong, it returns false with error message in astrng. Close, closeu, closeall, and clsfil should be examined for possible combination.

(clsid) (nls,ioexec,clsid) 11B 05577
(list) 02157
change stid's in passed correspondence list to sid's. see clhdr and clistr for format of clist. 05663

(clst1) (nls,records,) 01133
data declaration 04251

(clst2) (nls,records,) 01485
data declaration 04603

(cltype) (nls,records,) 01227
data declaration 04345

(clupdt) (nls,psupport,clupdt) 7B 0252
unbuild clist 03153

(cmarcha) (nls,cedit,cmarcha) 15A 01685
(bug,tp1,tp2) 04827
Core NLS Mark Character Command 04826

(cmda) (nls,records,) 01421
data declaration 04539

(cmdele) (nls,const,) 0800
data declaration 03916

(cmdfinish) (nls,parser,cmdfinish) 4G 0146

(cmdfree) (nls,dspgen,cmdfree) 4D 01891
(blkadress) 05215
free command blocks 05214

(cmdinit) (nls,dspgen,cmdinit) 4F 01883
(block,da) 05201
initialize display command block 05200

(cmdinterp) (nls,parser,cmdinterp) 5A1 0206
(resultptr,argptr) 03072
pointer to a function state record containing a pointer to a node in the grammar 03071

(cmdmod) (nls,pdata,) 032
data declaration 02893

(cmerge) (nls,cedit,cmerge) 16A 01734
(outlst,inlst,kpr,percnt,copyf) 04920

Core NLS Merge Primitive	04919
(cmergro) (nls,credit,cmergro) 16B	01720
(intobug1,intobug2,frombug1,frombug2)	04893
Core NLS Merge Group Command	04892
(cmlst) (nls,records,)	01302
data declaration	04420
(cmmpare) (nls,colstr,cmmpare) 8C	02341
(cmnxt) (nls,records,)	01326
data declaration	04444
(cmovfil) (nls,credit,cmovfil) 17A	01687
(rhost1,file1,rhost2,file2,astr)	04831
Core NLS Move File Command	04830
(cmovgro) (nls,credit,cmovgro) 17B	01753
(tostid,rlevnt,fromstid1,fromstid2,filter,vsptr)	04958
Core NLS Move Group Command	04957
(cmovsta) (nls,credit,cmovsta) 17D	01689
(tostid,rlevnt,fromstid,filter,vsptr)	04835
Core NLS Move Statement Command	04834
(cmovtex) (nls,credit,cmovtex) 17E	01726
(totptr,fromtptr1,fromtptr2,insertspace)	04905
Core NLS Move Text Command	04904
(cmpstr) (nls,utility,cmpstr) 8I	01601
(astr1,astr2)	04722
return -1 if astr1 < astr2, 0 if equal, 1 if astr1 > astr2	04721
(cmtyp) (nls,const,)	0896
data declaration	04014
(cntbleft) (nls,cident,cntbleft) 14I	02614
(stid)	06313
count the number of branches left in a plex	06312
(cntrlgetchar) (nls,inpfbk,cntrlgetchar) 10A	0512
GETCHAR with CONTROL FILE STUFF	03612
(cnvfndocnum) (nls,ioexec,cnvfndocnum) 15L	02181
(filename,docnumber)	05709
convert filename to document number	05708
(cnxtaddress) (nls,dspgen,cnxtaddress) 4E	01920
(entry=@block,direction)	05265
updates entry and block in direction specified	05264
(colgdest) (nls,colstr,colgdest) 6B	02315
(stid)	05914
(colmax) (nls,const,)	0955
data declaration	04073
(comchr) (nls,help,)	02796
data declaration	06617
(comdir) (nls,const,)	0693
data declaration	03811
(compactfile) (nls,credit,compactfile) 32E	01782
(ofn)	05011
does Compacted update File command	05010
(compas) (nls,utility,compas) 8H	01594
(astr1,astr2)	04715
compares the contents of two a-strings. Returns true if the contents match, false otherwise	04714
(complc) (nls,pdata,)	068
data declaration	02949
(comtyp) (nls,const,)	0580
data declaration	03698

(conbck) (nls, chelp, conbck) 1D3	02775
(index)	06586
back up one context if possible. otherwise go to etrypoint	06585
(conbp) (nls, psedit, conbp) 5H	0292
(base, cn)	03233
construct byte pointer	03232
(condir) (nls, ioexec, condir) 17E	02096
(destname, destpasswd, retname, retpasswd)	05553
connect to destination directory, and return name and password of	
current directory in indicated strings.	05552
(confre) (nls, chelp,)	02741
data declaration	06543
(conidir) (nls, ioexec, conidir) 17D	02118
(cflag)	05593
IF cflag true, connect to Identfile Directory, saving info for	
current one. Flag false means connect back	05592
(conjdir) (nls, ioexec, conjdir) 17C	02190
(cflag)	05725
IF cflag true, connect to Journal Directory, saving info for curren	
one. Flag false means connect back	05724
(conlnk) (nls, chelp,)	02780
data declaration	06594
(conrng) (nls, chelp,)	02750
data declaration	06552
(conspt) (nls, adrmnp, conspt) 1R1	02083
(zl)	05531
used for content and word searches	05530
(constd) (nls, chelp,)	02754
data declaration	06557
(constk) (nls, chelp,)	02753
data declaration	06556
(contls) (nls, const,)	01034
data declaration	04152
(contnt) (nls, const,)	0740
data declaration	03858
(convjdlist) (nls, csendmail, convjdlist) 8E	02661
convert Action Distribution and Information-only Distribution fields	
into a single Distribution field with comments appended to recipient	
idents.	06399
(convjsubcol) (nls, csendmail, convjsubcol) 8A	02700
(idfile)	06474
convert the contents of the Sub-Collections, Author, and	
Distribution field into a real Sub-Collections field for the JWORK	
header	06473
(cobcod) (nls, records,)	01287
data declaration	04405
(copen) (nls, colsrt, copen) 23	02349
(version, shoru)	05952
Open next output file, and return NLS file number	05951
(copfil) (nls, cedit, copfil) 32F1	01745
(oldfil, newfil, list)	04942
Copies and compacts ring and data blocks of files (used by outfile)	
	04941
(copflandpc) (nls, execfl, copflandpc) 3M	01935
(jfn1, pcjfn1, jfn2, pcjfn2, errstring)	05295
do the Copy file stuff	05294

(copgrp)	(nls,filmap,copgrp)	2E1	01802
	(stid,dir,grp1,grp2,upctf)		05049
	Given an stid as the first argument, this routine copies the group bounded by the second and third arguments after the first stid, in the direction specified by the fourth argument. The fifth argument indicates whether the correspondence list should be updated.		
(coprng)	(nls,cedit,coprng)	32F2	01723
	(ostid,nstid)		04899
	Copy name flag, name hash, and SID fields from one ring element to another		
(copsdb)	(nls,filmap,copsdb)	2J1	01807
	(source,dest)		05059
	Copies an SDB. Arguments are		
(copsrc)	(nls,recfil,copsrc)	8	02520
	(fnmstr)		06145
	This procedure accepts address of astring containing a file name. It changes the directory name for the rel file (in the "FILE" statement) to NIC-NLS, and then does an output file to the NIC-NLS directory.		
(copych)	(nls,frontend,copych)	13C	06144
	(numb)		02018
	copy 'numb' characters from old to new		
(copyda)	(nls,frontend,copyda)	11F	05432
	(dafrom,dato)		05431
	copy contents of da excluding file return ring		
(copyrd)	(nls,nddt,copyrd)	13B	02024
	(rd1,rd2)		05441
(copysrring)	(nls,frontend,copysrring)	12X	05440
	(fromsrr,tosrr)		02462
	copy statement return ring		
(corblk)	(nls,const,)		06066
	data declaration		
(corcpy)	(nls,records,)		02015
	data declaration		
(corexr)	(nls,const,)		05426
	data declaration		
(corlck)	(nls,records,)		05425
	data declaration		
(corlge)	(nls,const,)		0785
	data declaration		
(coropnfil)	(nls,icexec,coropnfil)	4I	03903
	(fileno,astng,access)		01215
	Primitive open file-- specified access.		
(cortim)	(nls,records,)		04333
	data declaration		
(coutassfil)	(nls,cedit,coutassfil)	18A	0773
	(ofilnam,da,forceup)		03891
	Core NLS Output to Assembler file command		
(coutjouqui)	(nls,cedit,coutjouqui)	18B	01193
	(ofilnam,da)		04311
	Core NLS Output Journal Quickprint command		
(coutnam)	(nls,colprt,coutnam)	11	0560
	read outfile name		
(coutproc)	(nls,cedit,coutproc)	18E	03678
			0862
			03980
			02192
			05729
			05728
			01211
			04329
			01752
			04956
			04955
			01714
			04881
			04880
			02312
			05911
			01708

(ofilnam,da,device,opflags)	04871
Core NLS Invoke Output Processor	04870
(coutgui) (nls,credit,coutgui) 18C	01739
(ofilnam,da)	04930
Core NLS Output Quickprint command	04929
(coutseqfil) (nls,credit,coutseqfil) 18D	01770
(ofilnam,da,forceup)	04989
Core NLS Output Sequential File command	04988
(cparml) (nls,records,)	01047
data declaration	04165
(cpcmpfil) (nls,cprograms,cpcmpfil) 2A2	02712
(type,cmpnam,ofilnam,da)	06498
Core NLS Compile File command	06497
(cpcnam) (nls,ioexec,cpcnam) 12G	02184
(oldstg,dirnrm)	05676
converts the file name passed in oldstg to a file name for pc in newstng	05675
(cpcconan) (nls,cprograms,cpcconan) 2A1	02711
(tptr,da)	06496
content analyzer pattern compile routine	06495
(cpfse) (nls,utility,cpfse) 9V	01564
(stid)	04683
statement end t-pointer	04682
(cprigro) (nls,credit,cprigro) 19C	01711
(stidl,stid2,vs,da)	04875
Core NLS Print Group command	04874
(cprires) (nls,credit,cprires) 19A	01729
(stid,da)	04910
Core NLS Print Rest command	04909
(cprista) (nls,credit,cprista) 19B	01785
(stid,vs,da)	05016
Core NLS Print Statement command	05015
(cprofil) (nls,credit,cprofil) 20A	01751
(rhstn,fname,mask,prot,astr)	04954
set protection of a file (group) (and its (their) partial copy(ies))	04953
(cpyfile) (nls,ioexec,cpyfile) 9C	02104
(newfno,oldfno)	05569
Copy file status table entry for newfno to oldfno. Used in outfile to force "new" file to use "old" file number and filst.	05568
(cpysr) (nls,utility,cpysr) 8N	01629
(asfrom,asto)	04756
To copy one A-string into another, call this routine with the address of the source string and the address of the destination string (in that order). An ERROR is generated if the maximum length of the destination string is shorter than the source string.	04755
(cpysw) (nls,seqgen,cpysw) 4A	01867
(fromsw,tosw)	05171
copy parts of one sequence work area to another	05170
(crensidfil) (nls,credit,crensidfil) 21A	01779
(fileno)	05005
Core NLS Renumber SID's in File Command	05004
(crepgro) (nls,credit,crepgro) 22B	01679
(bugdit,atbug1,atbug2,bybug1,bybug2)	04816
Core NLS Replace Group Command	04815
(crepsta) (nls,credit,crepsta) 22A	01781

(bugdit,atbug,bybug1,bybug2)	05009
Core NLS Replace Statement Command	05008
(creptex) (nls,cedit,creptex) 22D	01725
(bug1,bug2,bug3,bug4)	04903
Core NLS Replace Text Command	04902
(creror) (nls,colprt,creror) 24	02340
(errcod)	05941
(cresarcfil) (nls,cedit,cresarcfil) 23A	01756
(rhostn,fname)	04964
Core NLS Reset Archive File Request Command	04963
(crescasmod) (nls,cedit,crescasmod) 23B	01682
Core NLS Reset Case Mode Command	04821
(creslindex) (nls,cedit,creslindex) 23C	01733
(fileno,f1,f2)	04918
Core NLS Reset Link Default for File Command	04917
(cresngro) (nls,cedit,cresngro) 23D	01742
(std1,std2,da)	04936
Core NLS Reset Name Delimiters Group	04935
(cresnsta) (nls,cedit,cresnsta) 23E	01721
(std,da)	04895
Core NLS Reset Name Delimiters Statement	04894
(crestemmod) (nls,cedit,crestemmod) 23F	01722
(fileno,lock)	04897
Core NLS Reset Temporary Modifications Command	04896
(cretarcfil) (nls,cedit,cretarcfil) 24A	01771
(rhostn,fname)	04991
Core NLS Retrieve Archived File File Command	04990
(crlf) (nls,auxcod,crlf) 3H	02304
type a carriage return-line feed	05901
(crng) (nls,verify,crng) 6A	02227
(vrfy,fileno)	05796
ring blocks	05795
(crpgad) (nls,const,)	0654
data declaration	03772
(crstr) (nls,const,)	0964
data declaration	04082
(cs) (nls,const,)	0984
data declaration	04102
(cscreen) (nls,dspgen,cscreen) 7F	01890
clear screen - alphanumeric displays	05213
(csdb) (nls,verify,csdb) 6B	02224
(vrfy,fileno)	05791
statement data blocks	05790
(csetcgro) (nls,cedit,csetcgro) 25A	01761
(std1,std2,type)	04972
Core NLS Set Case Group Command	04971
(csetcmod) (nls,cedit,csetcmod) 25B	01730
(type)	04912
Core NLS Set Case Mode Command	04911
(csetcsta) (nls,cedit,csetcsta) 25C	01715
(std,type)	04883
Core NLS Set Case Statement Command	04882
(csetctex) (nls,cedit,csetctex) 25E	01774
(bug1,bug2,type)	04996
Core NLS Set Case Text Command	04995
(csetif) (nls,colprt,csetif) 10	02337

(stmtig)	05938
READ list of file names from keyboard from now	05937
(csetlindex) (nls,cedit,csetlindex) 25F	01747
(fileno,d1,d2)	04946
Core NLS Link Default of file Command	04945
(csetngro) (nls,cedit,csetngro) 25G	01736
(stid1,stid2,dlleft,dlright,da)	04924
Core NLS Set Name Delimiter Group Command	04923
(csetnsta) (nls,cedit,csetnsta) 25H	01759
(stid,dlleft,dlright)	04969
Core NLS Set Name Delimiter Statement Command	04968
(csettemmod) (nls,cedit,csettemmod) 25I	01769
(fileno)	04987
Core NLS Set Temporary Modifications to file	04986
(cshoarcdir) (nls,cedit,cshoarcdir) 26A	01699
(d1,d2,astr)	04854
Core NLS Show Archive Directory Command	04853
(cshodir) (nls,cedit,cshodir) 26B	01701
(info,gropk,sortk,rhstn,fname)	04858
Core NLS Show Directory Command	04857
(cshodskspa) (nls,cedit,cshodskspa) 26C	01686
(astr)	04829
Core NLS Show Disk Space Status Command	04828
(cshofilsta) (nls,cedit,cshofilsta) 26D	01740
(type,fileno,astrng)	04932
Core NLS Show Status of File	04931
(cshofrring) (nls,cedit,cshofrring) 26E	01775
(da,astrng)	04998
Core NLS Show Status of File Return Ring	04997
(cshomarfil) (nls,cedit,cshomarfil) 26F	01694
(fileno,astr)	04845
Core NLS Show Markers in File Command	04844
(cshonsta) (nls,cedit,cshonsta) 26G	01680
(stid,astrng)	04818
Core NLS Show Name Delimiters for Statement	04817
(csndtms) (nls,inpfbk,csndtms) 7B	0543
(messtring,jobno)	03655
core send tty messtring	03654
(csorgro) (nls,cedit,csorgro) 27A	01705
(bug1,bug2)	04865
Core NLS Sort Group Command	04864
(csort) (nls,cedit,csort) 27B	01744
(bugx,bug1,bug2,rlevnt,kpr,copyf)	04940
Core NLS Sort Primitive	04939
(cspcac) (nls,pdata,)	021
data declaration	02882
(cspupd) (nls,pdata,)	059
data declaration	02920
(cspusc) (nls,pdata,)	054
data declaration	02915
(cspvs) (nls,pdata,)	079
data declaration	02940
(csrstr) (nls,inpfbk,csrstr) 903	0487
(string,csize,hinc,font)	03575
set cursor string	03574
(cstr) (nls,const,)	0973

data declaration		04091
(csubgro) (nls,credit,csubgro) 28A		01698
(stid1,stid2,spheadb,da)		04852
Core-NLS substitute group		04851
(csubsta) (nls,credit,csubsta) 28B		01773
(stid,spheadb,da)		04994
Core-NLS substitute statement		04993
(csysbad) (nls,pascal,csysbad) 3L2		0439
(ctfile) (nls,records,)		01090
data declaration		04208
(ctfroz) (nls,records,)		01388
data declaration		04506
(ctfull) (nls,records,)		01344
data declaration		04462
(ctla) (nls,const,)		0952
data declaration		04070
(ctle) (nls,const,)		0989
data declaration		04107
(ctlex) (nls,const,)		0933
data declaration		04051
(ctlf) (nls,const,)		0853
data declaration		03971
(ctlg) (nls,const,)		0730
data declaration		03848
(ctlh) (nls,const,)		0998
data declaration		04116
(ctlo) (nls,const,)		0860
data declaration		03978
(ctlp) (nls,const,)		01025
data declaration		04143
(ctlquit) (nls,frontend,ctlquit) 9A		02011
shut off control stuff		05420
(ctlr) (nls,const,)		0771
data declaration		03889
(ctls) (nls,const,)		0681
data declaration		03799
(ctlt) (nls,const,)		0673
data declaration		03791
(ctlu) (nls,const,)		0970
data declaration		04088
(ctlv) (nls,const,)		0939
data declaration		04057
(ctlw) (nls,const,)		0649
data declaration		03767
(ctlx) (nls,const,)		0638
data declaration		03756
(ctly) (nls,const,)		0993
data declaration		04111
(ctlz) (nls,const,)		0737
data declaration		03855
(ctbnum) (nls,records,)		01226
data declaration		04344
(ctragro) (nls,credit,ctragro) 29A		01764
(first1,first2,secnd1,secnd2)		04977
Core NLS Transpose Group Command		04976
(ctrasta) (nls,credit,ctrasta) 29B		01748

(stid1,stid2)	04948
Core NLS Transpose Statement Command	04947
(ctratex) (nls,credit,ctratex) 29D	01741
(bug1,bug2,bug3,bug4)	04934
Core NLS Transpose Text Command	04933
(ctridir) (nls,credit,ctridir) 30A	01704
(nver,astr)	04863
trim the connected directory	04862
(ctrl) (nls,parser,)	0195
data declaration	03058
(ctrlc) (nls,utility,ctrlc) 11A	01619
save the accumulators	04741
(ctty) (nls,const,)	0685
data declaration	03803
(ctype) (nls,records,)	01493
data declaration	04611
(cueflg) (nls,pdata,)	019
data declaration	02880
(cundarcfil) (nls,credit,cundarcfil) 31A	01713
(f1,f2)	04879
Core NLS Undelete Archived File Command	04878
(cundfil) (nls,credit,cundfil) 31B	01786
(rhstn,iname,astr)	05018
undelete file (group) (and its (their) partial copy(ies))	05017
(cundmodfil) (nls,credit,cundmodfil) 31C	01712
(fileno)	04877
Core NLS Undelete Modifications to File	04876
(cupdate) (nls,credit,cupdate) 16C	01743
(outlst,inlst,kpr,udpr,percnt,initf)	04938
Core NLS Update Primitive	04937
(cupdfil) (nls,credit,cupdfil) 32A	01763
(fileno,type,namstr)	04975
Core NLS Update File command	04974
(curind) (nls,chelp,)	02774
data declaration	06584
(curmkr) (nls,pdata,)	052
data declaration	02913
(curnod) (nls,parser,)	0176
data declaration	03036
(current) (nls,parser,)	0172
data declaration	03032
(curren) (nls,const,)	0961
data declaration	04079
(currin) (nls,parser,)	0158
data declaration	03018
(currpa) (nls,parser,)	0127
data declaration	02983
(curstk) (nls,chelp,)	02804
data declaration	06628
(curvsp) (nls,frontend,curvsp) 6A	02016
(vspec,astrng)	05428
address of string to append string	05427
(cverfil) (nls,credit,cverfil) 33A	01693
(fileno,type)	04843
Core NLS File verify Command	04842
(cv sno) (nls,inpibk,cv sno) 10G	0531

(astr)		03640
convert a-string to number		03639
(cwacc) (nls,const,)		0845
data declaration		03963
(cwrpup) (nls,colprt,cwrpup) 25		02310
(d2sel) (nls,const,)		0834
data declaration		03952
(daauxi) (nls,records,)		01327
data declaration		04445
(dabott) (nls,records,)		01161
data declaration		04279
(dacacc) (nls,records,)		01525
data declaration		04643
(daccnt) (nls,records,)		01063
data declaration		04181
(daccol) (nls,records,)		01309
data declaration		04427
(dacrow) (nls,records,)		01304
data declaration		04422
(dacsiz) (nls,records,)		01538
data declaration		04656
(dacsp) (nls,records,)		01204
data declaration		04322
(daempt) (nls,records,)		01542
data declaration		04660
(daexis) (nls,records,)		01440
data declaration		04558
(dafont) (nls,records,)		01422
data declaration		04540
(dafmt) (nls,dspgen,dafmt) 3A1		01886
(da,oldsw)		05207
display area formatter		05206
(dafroz) (nls,records,)		01431
data declaration		04549
(dafrzl) (nls,records,)		01477
data declaration		04595
(dahand) (nls,records,)		01246
data declaration		04366
(dahinc) (nls,records,)		01348
data declaration		04466
(daidr1) (nls,records,)		01136
data declaration		04254
(daidr2) (nls,records,)		01107
data declaration		04225
(daind) (nls,records,)		01255
data declaration		04373
(dal) (nls,records,)		01170
data declaration		04288
(daleft) (nls,records,)		01121
data declaration		04239
(dalftj) (nls,records,)		01288
data declaration		04406
(dalhan) (nls,records,)		01127
data declaration		04245
(dalink) (nls,records,)		01351
data declaration		04469

(dalsid) (nls,records,)		01213
data declaration		04331
(dalsrt) (nls,records,)		01325
data declaration		04443
(dalssu) (nls,records,)		01267
data declaration		04385
(dalsz) (nls,records,)		01166
data declaration		04284
(damax) (nls,records,)		01452
data declaration		04570
(damcol) (nls,records,)		01298
data declaration		04416
(damind) (nls,records,)		01483
data declaration		04601
(damrow) (nls,records,)		01164
data declaration		04282
(damt) (nls,dspgen,damt) 3A4		01906
(da)		05240
empty da routine		05239
(danocs) (nls,records,)		01151
data declaration		04269
(danohi) (nls,records,)		01095
data declaration		04213
(dapstf) (nls,records,)		01046
data declaration		04164
(dapvs) (nls,records,)		01405
data declaration		04523
(dapvs2) (nls,records,)		01308
data declaration		04426
(darigh) (nls,records,)		01438
data declaration		04556
(daseq) (nls,records,)		01532
data declaration		04650
(dasgcs) (nls,records,)		01168
data declaration		04286
(dasghi) (nls,records,)		01414
data declaration		04532
(dashri) (nls,records,)		01496
data declaration		04614
(dassnbit) (nls,utility,dassnbit) 14D1		01593
(table.bit,length)		04713
(dasupp) (nls,records,)		01042
data declaration		04160
(datab0) (nls,records,)		01546
data declaration		04664
(datab1) (nls,records,)		01214
data declaration		04332
(datab2) (nls,records,)		01306
data declaration		04424
(datfrmt) (nls,auxcod,datfrmt) 90		02306
(tim.frmt,astng)		05905
put date and time in astring passed		05904
(datop) (nls,records,)		01531
data declaration		04649
(daukey) (nls,records,)		01463
data declaration		04581

(daupdate) (nls,dsppgen,daupdate) 3A2	01907
(da)	05242
update display area image	05241
(dausqc) (nls,records,)	01381
data declaration	04499
(davinc) (nls,records,)	01519
data declaration	04637
(davspc) (nls,records,)	01488
data declaration	04606
(davspe) (nls,records,)	01037
data declaration	04155
(ddgtsymbol) (nls,nddt2,ddgtsymbol) 1D1	02496
(astr,value)	06103
generates a symbolic representation for a value	06102
(ddsymget) (nls,nddt2,ddsymget) 1D2	02498
(astr,value)	06107
find a DDT symbol associated with a value	06106
(ddtadr) (nls,nddt,)	02402
data declaration	06006
(ddtcalls) (nls,nddt,ddtcalls) 6A	02477
Various calling routines for ddt	06079
(ddtchkp) (nls,nddt2,ddtchkp) 1D10	02502
(blockptr)	06113
(ddtenter) (nls,nddt2,ddtenter) 1D4	02497
(astr,value,type)	06105
this routine enters a symbol and value into the ddt symbol table	
the current range of the symbol table is contained in "symloc"	
the highest load address is contained in "loadh".	06104
(ddthash) (nls,nddt,ddthash) 12B1	02480
(string)	06082
(ddtlookup) (nls,nddt2,ddtlookup) 1D5	02505
(astr,skiplocals)	06119
IF the second argument is TRUE then the mark stack is ignored and	
the entire DDT symbol table is searched bottom up	06116
(ddtmark) (nls,nddt2,ddtmark) 1D6	02501
(astr)	06112
this routine marks the current position in the ddt symbol table and	
creates a new block with the name given the value saved in the mark	
stack is the block pointer for the marked block	06111
(ddtname) (nls,nddt2,ddtname) 1D7	02503
(astr,type)	06115
this routine returns a radix 50 interpretation of the string and	
incorporates the type information with it to produce a ddt symbol.	
Only Upper or Lower case alphabetic and numeric characters are	
allowed.	06114
(ddtpop) (nls,nddt2,ddtpop) 1D9	02495
this routine pops the current routine from the marked symbol table	
stack	06101
(ddtsm) (nls,nddt,)	02492
data declaration	06096
(ddtsrchblk) (nls,nddt2,ddtsrchblk) 1D8	02494
(blockptr,symbolhash)	06100
this routine searches through a specified block of the symbol	
table, looking for a match with a given symbol value. Inline coding	
has been used to speed up the search loop and registers are used as	
follows:	06099

(ddtsym) (nls,nddt,ddtsym) 12A1	02422
(string)	06027
(de) (nls,const,)	0704
data declaration	03822
(dealocda) (nls,dspgen,dealocda) 6E	01916
(da)	05258
given a system daid, deallocate it	05257
(deblnk) (nls,nddt,deblnk) 7B	02474
(decode) (nls,parser,decode) 5A6	0153
(psptr)	03011
ptr to path stack entry	03010
(defact) (nls,pdata,)	039
data declaration	02900
(defbmo) (nls,const,)	0879
data declaration	03997
(defcol) (nls,const,)	0893
data declaration	04011
(defcur) (nls,const,)	01005
data declaration	04123
(defdbk) (nls,const,)	0700
data declaration	03818
(defdin) (nls,const,)	0674
data declaration	03792
(deferr) (nls,auxcod,deferr) 4E	02261
This procedure gets called when a signaal is done with nothing	
stopping it on the stack.	05842
(defrr) (nls,const,)	0787
data declaration	03905
(defhmo) (nls,const,)	0936
data declaration	04054
(defhsi) (nls,const,)	0794
data declaration	03912
(defind) (nls,const,)	0883
data declaration	04001
(defkey) (nls,colprt,defkey) 5B	02346
(stid,outb,num)	05947
(deflin) (nls,const,)	0858
data declaration	03976
(defoff) (nls,const,)	0555
data declaration	03673
(defpgs) (nls,const,)	0901
data declaration	04019
(defpro) (nls,const,)	0655
data declaration	03773
(defre2) (nls,const,)	0736
data declaration	03854
(defrec) (nls,const,)	0981
data declaration	04099
(defsr) (nls,const,)	0820
data declaration	03938
(deftb1) (nls,const,)	0597
data declaration	03715
(deftb2) (nls,const,)	0630
data declaration	03748
(deftb3) (nls,const,)	0722
data declaration	03840

(defvs1) (nls,const,)	0856
data declaration	03974
(defvs2) (nls,const,)	0871
data declaration	03989
(delbm) (nls,inpfbk,delbm) 9A3	0485
turn the most recent bug mark off	03571
(delcmidents) (nls,cident,delcmidents) 13K	02649
(lhlist,lhdelete,rhlist,rhdelete)	06378
delete idents common to both ident lists	06377
(delda) (nls,frontend,delda) 11N	02008
(da)	05417
delete window	05416
(delfil) (nls,ioexec,delfil) 9B	02121
(fileno)	05597
Delete file (which has been closed) from NLS internal bookkeeping	
including file status table, display area table, and other arrays.	
Write out called here and in several places before it (see close)!	
Clean this up.	05596
(delflandpc) (nls,execfl,delflandpc) 3H	01943
(fjfn,pcjfn,erstring)	05311
delete file (and partial copy) for input jfns	05310
(delgrp) (nls,filmp,delgrp) 2E3	01849
(grp1,grp2,newsid)	05137
This routine destroys all trace of the group bounded by the	
arguments passed it. (Note that it does not remove the group from	
the ring; it assumes that this has been done by, for example,	
REMGRP.)	05136
(delhc) (nls,records,)	01501
data declaration	04619
(delmkn) (nls,auxcod,delmkn) 8A	02269
(fileno,name)	05852
delete the marker whose name is passed from the specified	05851
(delmkr) (nls,auxcod,delmkr) 8B	02276
(tptrs)	05864
delete the markers in T-string specified by arg	05863
(delnet) (nls,records,)	01194
data declaration	04312
(delol) (nls,records,)	01436
data declaration	04554
(delovsrns) (nls,ioexec,delovsrns) 15C	02144
(jfn.number)	05640
Deletes old versions of file if there are more than the number	
passed as the second parameter	05639
(delpc) (nls,ioexec,delpc) 12E	02171
(fileno,astng)	05689
delete partial copy of file from user directory if possible; sets	
lock fields in fdb to zero (unlocked), set other flags referring to	
the existence of the partial copy to zero	05688
(delsubsys) (nls,parser,delsubsys) 4E	0130
(dptptr)	02987
ptr to subsystem dispatch record (this is the symbolic name which	
appears on the SUBSYSTEM statement of the CML definition of the	
subsystem	02986
(dendvw) (nls,inpfbk,dendvw) 4B	0537
handle end of viewspec input	03648
(dent) (nls,pdata,)	027

data declaration	02888
(dest) (nls,pdata,)	086
data declaration	02947
(dev33) (nls,const,)	0844
data declaration	03962
(dev35) (nls,const,)	0762
data declaration	03900
(dev37) (nls,const,)	0708
data declaration	03826
(devlpr) (nls,const,)	0839
data declaration	03957
(devsri) (nls,const,)	0884
data declaration	04002
(devstr) (nls,psuserop,devstr) 2D4	0415
(dev)	03469
converts internal code for device to string representation	03468
(devtie) (nls,const,)	0874
data declaration	03992
(dex) (nls,intnls,dex) 4E	02825
start DEX	06648
(dfiltname) (nls,psedit,dfiltname) 3U4	0279
(str)	03207
construct default output file name	03206
(dfiltr) (nls,const,)	0697
data declaration	03815
(dfnsubsys) (nls,parser,dfnsubsys) 4D	0200
(dptptr,controlbits)	03065
TNLS, DNLS, level 1 options	03064
(dig) (nls,ccalc,)	02529
data declaration	06157
(dinptc) (nls,inpfbk,dinptc) 4A	0492
DNLS single character input routine	03583
(dinstupg) (nls,usrpgm,dinstupg) 9R	02729
(pgmaddr)	06527
deinstitute a program	06526
(disabaccess) (nls,icexec,disabaccess) 13D	02195
(fileno,type)	05735
see enablaccess for comment	05734
(disablw) (nls,utilty,disablw) 15B	01609
(pcap)	04733
Disable wheel status, and restore status to pcap	04732
(disarm) (nls,inpfbk,disarm) 902	0494
disarm cursor	03586
(dismes) (nls,auxcod,dismes) 3A	02295
(type,astrng)	05892
display a message to the user	05891
(dismsg) (nls,auxcod,dismsg) 3B	02282
(astrng)	05872
..display the message for DISMES..	05871
(dlbgac) (nls,execfl,dlbgac) 4N	01930
(fjfn,fdb,adlmb)	05285
get account string or number for a directory list	05284
(dlbgas) (nls,execfl,dlbgas) 4R	01939
(flags,string)	05303
get archive status string	05302
(dlbglw) (nls,execfl,dlbglw) 4Q	01952

(dirno,string)	05328
get last writer string	05327
(dlbgpr) (nls,execfl,dlbgpr) 4F	01947
(prot,string)	05318
get protection string for individuall field	05317
(dlbgtd) (nls,execfl,dlbgtd) 40	01938
(param,time,string)	05301
append /time and/ date string to passed string	05300
(dlbl dai) (nls,execfl,dlbl dai) 4F	01949
(fjfn,pcjfn,jfnfdb,pcfdb,string)	05322
build additional line l information string	05321
(dlbl di) (nls,execfl,dlbl di) 4G	01925
(info,pcjfn,jfnfdb,pcfdb,lead,string)	05275
build information string for this file	05274
(dlbl ds) (nls,execfl,dlbl ds) 4M	01954
(fjfn,pcjfn,string,adlfb)	05332
build string for second pass (and set bits in dlfb	05331
(dlckdl) (nls,execfl,dlckdl) 4L	01927
(fjfn,delbit)	05279
check deletion status of this file	05278
(dl driver) (nls,execfl,dl driver) 4A	01940
(info,gropk,sortk,fjfn,jfnflgs,erstring,adlmb)	05305
main procedure for directory commands	05304
(dlfbal) (nls,records,)	01300
data declaration	04418
(dlfbap) (nls,records,)	01041
data declaration	04159
(dlfbcp) (nls,records,)	01338
data declaration	04456
(dlfbff) (nls,records,)	01490
data declaration	04608
(dlfbfl) (nls,records,)	01131
data declaration	04249
(dlfbgk) (nls,records,)	01506
data declaration	04624
(dlfbl) (nls,records,)	01560
data declaration	04678
(dlfbnb) (nls,records,)	01479
data declaration	04597
(dlfbpb) (nls,records,)	01039
data declaration	04157
(dlfbpf) (nls,records,)	01084
data declaration	04202
(dlfb sk) (nls,records,)	01265
data declaration	04383
(dlfmb) (nls,execfl,dlfmb) 4C	01937
(ablk)	05299
free directory list master blocks	05298
(dlgacc) (nls,records,)	01101
data declaration	04219
(dlgars) (nls,records,)	01220
data declaration	04338
(dlgart) (nls,records,)	01367
data declaration	04485
(dlgbgk) (nls,records,)	01055
data declaration	04173

(dlgbl) (nls,records,)	01278
data declaration	04396
(dlgbnb) (nls,records,)	01376
data declaration	04494
(dlgbnu) (nls,records,)	01486
data declaration	04604
(dlgbpb) (nls,records,)	01475
data declaration	04593
(dlgbsc) (nls,records,)	01108
data declaration	04226
(dlgbyt) (nls,records,)	01503
data declaration	04621
(dlgdar) (nls,records,)	01203
data declaration	04321
(dlgdcr) (nls,records,)	01333
data declaration	04451
(dlgddm) (nls,records,)	01522
data declaration	04640
(dlgdfr) (nls,records,)	01153
data declaration	04271
(dlgdlt) (nls,records,)	01111
data declaration	04229
(dlgdmr) (nls,records,)	01202
data declaration	04320
(dlgdov) (nls,records,)	01316
data declaration	04434
(dlgdrd) (nls,records,)	01420
data declaration	04538
(dlgdwr) (nls,records,)	01270
data declaration	04388
(dlgetg) (nls,execfl,dlgetg) 4H	01951
(gropk,jfnfdb,pcjfn,pcfdb)	05326
return the group key for this file	05325
(dlgets) (nls,execfl,dlgets) 4I	01926
(sortk,jfnfdb,pcjfn,pcfdb)	05277
return the sort key for this file	05276
(dlglwr) (nls,records,)	01465
data declaration	04583
(dlgmb) (nls,execfl,dlgmb) 4B	01942
(blksiz)	05309
get and initialize a directory list master block	05308
(dlgprt) (nls,records,)	01369
data declaration	04487
(dlgrvr) (nls,records,)	01497
data declaration	04615
(dlgtblk) (nls,execfl,dlgtblk) 4D	01934
(ablk,blksiz)	05293
get a directory block of length blksiz	05292
(dliacc) (nls,records,)	01266
data declaration	04384
(dliars) (nls,records,)	01083
data declaration	04201
(dliart) (nls,records,)	01236
data declaration	04354
(dlibyt) (nls,records,)	01295
data declaration	04413

(dlidfr) (nls,records,)	01406
data declaration	04524
(dlidlt) (nls,records,)	01221
data declaration	04339
(dlidmt) (nls,records,)	01299
data declaration	04417
(dlifrf) (nls,records,)	01428
data declaration	04546
(dlilwr) (nls,records,)	01140
data declaration	04258
(dlimis) (nls,records,)	01241
data declaration	04359
(dline) (nls,dspgen,dline) 7I	01901
(x,y)	05231
delete line - alphanumeric displays	05230
(dlinex) (nls,records,)	01396
data declaration	04514
(dlinfb) (nls,execfl,dlinfb) 4J	01946
(chns,adlmb,adlfb,sortk)	05316
insert a dlfb in chain	05315
(dlinrw) (nls,records,)	01339
data declaration	04457
(dlinvr) (nls,records,)	01375
data declaration	04493
(dliprt) (nls,records,)	01197
data declaration	04315
(dlisiz) (nls,records,)	01093
data declaration	04211
(dlitar) (nls,records,)	01378
data declaration	04496
(dlitcr) (nls,records,)	01205
data declaration	04323
(dlitdm) (nls,records,)	01192
data declaration	04310
(dlitov) (nls,records,)	01504
data declaration	04622
(dlitrd) (nls,records,)	01061
data declaration	04179
(dlitwr) (nls,records,)	01462
data declaration	04580
(dlmbal) (nls,records,)	01181
data declaration	04299
(dlmbfl) (nls,records,)	01517
data declaration	04635
(dlmbfp) (nls,records,)	01223
data declaration	04341
(dlmbfr) (nls,records,)	01527
data declaration	04645
(dlmbli) (nls,records,)	01244
data declaration	04362
(dlmbln) (nls,records,)	01078
data declaration	04196
(dlmbnm) (nls,records,)	01340
data declaration	04458
(dlmbpr) (nls,records,)	01105
data declaration	04223

(dlmbzn) (nls,records,)	01079
data declaration	04197
(dlsacc) (nls,records,)	01198
data declaration	04316
(disalp) (nls,records,)	01526
data declaration	04644
(dlsart) (nls,records,)	01114
data declaration	04232
(dlsbyt) (nls,records,)	01342
data declaration	04460
(dlsdfr) (nls,records,)	01373
data declaration	04491
(dlsdlt) (nls,records,)	01173
data declaration	04291
(dlsdmt) (nls,records,)	01252
data declaration	04370
(dlslen) (nls,records,)	01402
data declaration	04520
(dlslwr) (nls,records,)	01064
data declaration	04182
(dlsnac) (nls,records,)	01346
data declaration	04464
(dlsnps) (nls,execfl#@dlsnps) MK	01931
(nlschn,pcchn)	05287
directory command 2 pass - get rid of redundant PCs	05286
(dlsnrd) (nls,records,)	01053
data declaration	04171
(dlsnwr) (nls,records,)	01487
data declaration	04605
(dlsrvr) (nls,records,)	01543
data declaration	04661
(dlssiz) (nls,records,)	01370
data declaration	04488
(dlstar) (nls,records,)	01352
data declaration	04470
(dlstcr) (nls,records,)	01176
data declaration	04294
(dlstdm) (nls,records,)	01167
data declaration	04285
(dlstig) (nls,records,)	01409
data declaration	04527
(dlstnl) (nls,records,)	01074
data declaration	04192
(dlstov) (nls,records,)	01466
data declaration	04584
(dlstpc) (nls,records,)	01368
data declaration	04486
(dlstrd) (nls,records,)	01524
data declaration	04642
(dlstwr) (nls,records,)	01434
data declaration	04552
(dn) (nls.inpfbk,dn) 9E1	0498
(astrng)	03591
display string in name area	03590
(dnls) (nls.intnls,dnls) MB	02856
start display NLS	06683

(dnlscm) (nls,parser,)	0212
data declaration	03060
(dntyp) (nls,const,)	0878
data declaration	03996
(doctrlic) (nls,utility,doctrlic) 11B	01563
(dol) (nls,ccalc,)	02528
data declaration	06156
(dolitr) (nls,parser,)	0193
data declaration	03055
(down) (nls,const,)	0556
data declaration	03674
(downst) (nls,const,)	0656
data declaration	03774
(dpset) (nls,psupport,dpset) 7C	0258
(option,stdlib,stdlib2,stopstid)	03165
set global variables for recreate display routines. In some cases, stdlib and stdout are passed merely to indicate file involvement, not because they will be reformatted.	03164
(dpstp) (nls,psupport,dpstp) 7D	0247
(stdid)	03144
called by text editing control routines to accept the stdid of a bugged statement as inpt and return an appropriate stdid for the display parameter CDSTOP (an stdid on the same level or higher which appears after 'stdid' in the file).	03143
(dpt) (nls,ccalc,)	02531
data declaration	06159
(dptfin) (nls,parser,)	0160
data declaration	03020
(dptini) (nls,parser,)	0183
data declaration	03044
(dptnam) (nls,parser,)	0125
data declaration	02981
(dptnot) (nls,parser,)	0144
data declaration	03002
(dptr) (nls,utility,dptr) 9C	01587
(pntloc)	04705
(dptren) (nls,parser,)	0147
data declaration	03004
(dptrun) (nls,parser,)	0197
data declaration	03061
(dptval) (nls,parser,)	0142
data declaration	03000
(dpybks) (nls,const,)	0647
data declaration	03765
(dpycoc) (nls,const,)	01033
data declaration	04151
(ds) (nls,const,)	0843
data declaration	03961
(dSPACE) (nls,dspgen,dSPACE) 3B4	01908
(addition)	05244
allocate addition more lsrtl entries for table at rttop, updating globals rttop, rtsize, rt, rte	05243
(dspall) (nls,const,)	0852
data declaration	03970
(dsparea) (nls,frontend,dsparea) 11M	01996
(dano)	05400

```

get window record address from window number -- returns zero if
window is not allocated                                05399
(dspbke) (nls,const,)                                  0991
  data declaration                                    04109
(dspblk) (nls,const,)                                  0742
  data declaration                                    03860
(dspjpf) (nls,const,)                                  0559
  data declaration                                    03677
(ds=no) (nls,const,)                                  0797
  data declaration                                    03915
(dsorfm) (nls,const,)                                  0888
  data declaration                                    04006
(dsprfs) (nls,const,)                                  0948
  data declaration                                    04066
(dspstr) (nls,const,)                                  0579
  data declaration                                    03697
(dspvsp) (nls,inpfbk,dspvsp) 9D1                      0510
  (vspecl,vspecl2,which)                              03609
  =1 -- LT "numbers". = 2 -- view specs, =3 -- both 03608
(dspyes) (nls,const,)                                  0658
  data declaration                                    03776
(dstar) (nls,records,)                                 01280
  data declaration                                    04398
(dstart) (nls,frontend,dstart) 14A                   01976
  (gs)                                                 05369
  DNLS startup routine                                05368
(dsttx) (nls,utility,dsttx) 6A                        01672
  (stid)                                               04804
  delete the statement data block (SDB) of the STID passed 04803
(dsubsys) (nls,inpfbk,dsubsys) 9F1                    0504
  (astrng)                                             03601
  display in subsystem area                            03600
(dtbas) (nls,const,)                                  0723
  data declaration                                    03841
(dtbm) (nls,const,)                                   0922
  data declaration                                    04040
(dtfrmt) (nls,auxcod,dtfrmt) 9B                       02274
  (tim.astrng)                                         05860
  out date and time in canonical form in astring passed 05859
(dumpc) (nls,ioexec,dumpc) 12D                        02156
  (fileno,value)                                       05661
  Clears or sets all of the pc bits in the file header ring and data
  block status tables, depending on value. Currently, output and
  update call it after doing all their edits.          05660
(dvstar) (nls,records,)                                 01408
  data declaration                                    04526
(dxstrt) (nls,frontend,dxstrt) 14E                    01985
  (gs)                                                 05380
  Dummy DEX startup routine(starts DNLS!!)            05379
(eacc) (nls,const,)                                   0778
  data declaration                                    03896
(echda) (nls,const,)                                   0735
  data declaration                                    03853
(echo) (nls,inpfbk,echo) 6C                           0516
  (string)                                             03619
  echo a string                                        03618

```



```

(echoff) (nls,inprbk,echoff) 8B 0544
no_echo for any character; break on all characters 03656
(echoimed) (nls,inprbk,echoimed) 8E 0549
echo each character with itself; each character is a break character 03665
(echoit) (nls,inprbk,echoit) 8D 0534
echo all characters(immediate or deferred), break on all but
alpha-numeric 03645
(echon) (nls,inprbk,echon) 8A 0507
echo each character with itself; each character is a break character 03604
(echostr) (nls,const,) 01011
data declaration 04129
(edistr) (nls,parser,edistr) 5C1 0155
(targetptr) 03014
ptr to result string 03013
(edit) (nls,const,) 0899
data declaration 04017
(editx) (nls,frontend,editx) 13A 01993
(ptr) 05395
----- 05394
(edixqt) (nls,frontend,edixqt) 13B 01978
reassign the terminal codes 05371
(efree) (nls,const,) 0914
data declaration 04032
(empty) (nls,utility,) 01610
data declaration 04734
(emrese) (nls,const,) 0982
data declaration 04100
(emrtmf) (nls,const,) 01017
data declaration 04135
(esign) (nls,const,) 0709
data declaration 03827
(enablaccess) (nls,ioexec,enablaccess) 13C 02100
(fileno,type) 05561
Set NLS system access to file or global mask for user. (Used for
special subsystems , e.g., Journal.) Set bit in access mask which
corresponds to type. Set mask in file if fileno # 0, otherwise set
global mask. 05560
(enabllogin) (nls,intnls,enabllogin) 6Y 02849
Enable the Login capability 06675
(enablw) (nls,utility,enablw) 15A 01660
(endarb) (nls,utility,endarb) 9K 01638
(endfil) (nls,utility,) 01597
data declaration 04717
(endopt) (nls,parser,) 0140
data declaration 02998A(end
Send end stand out command - alphanumeric displays 05229
(enlstr) (nls,const,) 0963
data declaration 04081
(ent) (nls,pdata,) 040
data declaration 02901
(entcon) (nls,cheap,) 02803
data declaration 06627
(enthelp) (nls,parser,enthelp) 4C 0113

```

set up to enter the help system after a TQ	02968
(entind) (nls, chelp,)	02748
data declaration	06550
(entyp) (nls, nddt,)	02458
data declaration	06062
(entype) (nls, parser,)	0205
data declaration	03070
(enwheel) (nls, utility, enwheel) 150	01650
logged in user an enabled wheel?	04776
(err) (nls, auxcod, err) 4A	02278
(errno)	05866
(errsig) (nls, const,)	0667
data declaration	03785
(errx) (nls, ccalc, errx) 50	02538
(whence, ar)	06169
JSYS dfout has returned with an error condition. R4 contains error mnemonics. FLOTX1 and FLOTX2 indicate column overflow. All other error returns are either calculator or tenex bugs. If the error occurred on an accumulator and the format is the default, maximum no. of digits, the accum will be set to zero. If the format is for less than the maximum, the format will be changed to the maximum.	
	06168
(esc) (nls, utility, esc) 5B	01671
End string construction.	04802
(escape) (nls, nddt,)	02435
data declaration	06039
(escstr) (nls, const,)	0683
data declaration	03801
(estar) (nls, records,)	01253
data declaration	04371
(estr) (nls, const,)	0705
data declaration	03823
(evaler) (nls, parser, evaler) 5A4	0186
(curptr)	03048
ptr to path stack entry	03047
(evalmo) (nls, parser,)	0166
data declaration	03026
(evalst) (nls, pdata,)	047
data declaration	02908
(evalx) (nls, pdata,)	060
data declaration	02921
(evalxs) (nls, parser,)	0171
data declaration	03031
(exp2) (nls, ccalc,)	02524
data declaration	06151
(expadr) (nls, const,)	0905
data declaration	04023
(expdtst) (nls, cident, expdtst) 80	02586
(string, dstptr)	06260
test expand field and set text-pointer to membership list	06259
(exsign) (nls, ccalc,)	02532
data declaration	06160
(extnam) (nls, const,)	0677
data declaration	03995
(fback2) (nls, const,)	0900
data declaration	04018

(fbackm) (nls,const,)		0726
data declaration		03844
(fbaddl) (nls,const,)		0585
data declaration		03703
(fbctl) (nls,parser,fbctl) 601		0156
(function,pl)		03016
feedback control routine		03015
(fbhdl) (nls,const,)		0557
data declaration		03675
(fbhelp) (nls,parser,fbhelp) 6D1		0111
(list,cmode)		02966
lists alternatives		02965
(fbind) (nls,records,)		01400
data declaration		04518
(fblen) (nls,parser,)		0168
data declaration		03028
(fbnull) (nls,records,)		01385
data declaration		04503
(fbnum) (nls,records,)		01138
data declaration		04256
(fbpop) (nls,const,)		0622
data declaration		03740
(fbstr) (nls,pdata,)		050
data declaration		02911
(fbstrp) (nls,parser,)		0139
data declaration		02997
(fbtype) (nls,records,)		01289
data declaration		04407
(fchsdb) (nls,filmp, fchsdb) 2I2		01821
(std)		05085
Shortcut for loading sdb given std of statement. See lodsdb for		
status when returns.		05084
(fchtxt) (nls,filmp, fchtxt) 2I3		01830
(stdb)		05102
Called with stdb returns the number of the first character after the		
name of the statement		05101
(fcount) (nls,parser,)		0123
data declaration		02979
(fcp) (nls,utility,fcp) 9H2		01630
(fdbaar) (nls,records,)		01509
data declaration		04627
(fdbact) (nls,records,)		01257
data declaration		04375
(fdbadl) (nls,records,)		01059
data declaration		04177
(fdbarc) (nls,records,)		01229
data declaration		04347
(fdbart) (nls,records,)		01336
data declaration		04454
(fdbbkf) (nls,records,)		01394
data declaration		04512
(fdbbyv) (nls,records,)		01399
data declaration		04517
(fdbcnt) (nls,records,)		01318
data declaration		04436
(fdbcre) (nls,records,)		01141

data declaration	04259
(fdbcrv) (nls,records,)	01162
data declaration	04260
(fdbctl) (nls,records,)	01135
data declaration	04253
(fdbctu) (nls,records,)	01505
data declaration	04623
(fdbdel) (nls,records,)	01330
data declaration	04448
(fdbdmp) (nls,records,)	01513
data declaration	04631
(fdbeph) (nls,records,)	01537
data declaration	04655
(fdblng) (nls,records,)	01225
data declaration	04343
(fdbmrk) (nls,records,)	01092
data declaration	04210
(fdbnar) (nls,records,)	01470
data declaration	04588
(fdbnex) (nls,records,)	01314
data declaration	04432
(fdbnun) (nls,records,)	01186
data declaration	04304
(fdbnxf) (nls,records,)	01113
data declaration	04231
(fdbprm) (nls,records,)	01482
data declaration	04600
(fdbprt) (nls,records,)	01044
data declaration	04162
(fdbref) (nls,records,)	01426
data declaration	04544
(fdbsiz) (nls,records,)	01235
data declaration	04353
(fdbtdm) (nls,records,)	01494
data declaration	04612
(fdbtfa) (nls,records,)	01191
data declaration	04309
(fdbtmp) (nls,records,)	01495
data declaration	04613
(fdbtsa) (nls,records,)	01207
data declaration	04325
(fdbunu) (nls,records,)	01469
data declaration	04587
(fdbuse) (nls,records,)	01410
data declaration	04528
(fdbusw) (nls,records,)	01281
data declaration	04399
(fdbver) (nls,records,)	01277
data declaration	04395
(fdbwrt) (nls,records,)	01413
data declaration	04531
(fdd) (nls,records,)	01156
data declaration	04274
(fds) (nls,records,)	01117
data declaration	04235
(fe) (nls.const,)	0593

data declaration		03711
(fechcl) (nls,utility,fechcl) 7A		01626
(dir,worka)		04750
Documentation		04749
(fechnd) (nls,seqgen,fechnd) 4K		01858
(stid,astr)		05154
This is like FECHNO, but generates the statement number of the sub (whether it exists or not).		05153
(fechnm) (nls,seqgen,fechnm) 4L		01876
(stvwrk,astr)		05189
This routine will append the statement number to the string, given the statement vector. The address of the first word of the statement vector is provided as the first argument, and the address of the A-string for the statement number is given as the second.		05188
(fechno) (nls,seqgen,fechno) 4I		01866
(stid,astr)		05169
Appends statement number of stid to string. Give the STID as the first argument, and the address of the string which is to contain the statement number as the second. The statement number will be built in the string. If the structure is not intact or the statement vector cannot be built, a call to RERROR or an EXCEED CAPACITY ERROR may result.		05168
(fechnp) (nls,seqgen,fechnp) 4J		01875
(stid,astr)		05187
This is like FECHNO, but generates the statement number of the successor (whether it exists or not).		05186
(fechsig) (nls,seqgen,fechsig) 4N		01869
(stid,astrng)		05175
append statement signature to string		05174
(fechux) (nls,seqgen,fechux) 4M		01872
(astr,fileno)		05181
Given the address of an a-string as the first argument and a file number as the second, this routine returns the STID of the statement whose statement number is contained in the A-string.		05180
(fedind) (nls,const,)		0611
data declaration		03729
(feedbk) (nls,const,)		0957
data declaration		04075
(feedlt) (nls,prmspc,feedlt) 9D		02054
(dpa,astrng)		05479
Given the address of a display area, this routine changes		05478
(feedsw) (nls,help,feedsw) 1E15		02737
(sw,astrng)		06538
Given the address of a sequence work area, this routine changes its vspecs in accord with the specifications in the A-string passed it. It passes the characters in the A-string to <PRMSPC, SETLT>, except for content analyzer patterns.		06537
(ff) (nls,pdata,)		053
data declaration		02914
(fhand) (nls,seqfil,)		02244
data declaration		05822
(fhunus) (nls,records,)		01547
data declaration		04665
(fielddesig) (nls,nddt,fielddesig) 10A2		02467
(rd)		06070

(file) (nls,const,)		0999
data declaration		04117
(filere) (nls,const,)		0764
data declaration		03882
(filev) (nls,const,)		0911
data declaration		04029
(filhdr) (nls,filmno,filhdr) 2M1		01845
(fileno)		05129
returns the address of the file header for the file whose number is		
passed		05128
(filinit) (nls,intnls,filinit) 6E		02862
init file handling mechanisms		06691
(filmax) (nls,records,)		01533
data declaration		04651
(filnam) (nls,ioexec,filnam) 9E		02175
(fileno,astrng)		05697
Get file name from filst entry for file whose file number is passd.		
Put into string whose address is passed		05696
(filnum) (nls,ioexec,filnum) 9F		02170
(astrng)		05687
Get NLS file number from filst given file name string. Returns -1		
if error or string not in table.		05686
(filstl) (nls,records,)		01474
data declaration		04592
(filtre) (nls,pdata,)		063
data declaration		02924
(filusd) (nls,frontend,filusd) 3B		02012
(fileno,da)		05422
Given a file number and the address of a display area, this routine		
returns TRUE if the file is used in the frozen list or csp		
associated with the display area; otherwise it returns FALSE.		05421
(filver) (nls,cident,)		02573
data declaration		06234
(findbp) (nls,nddt,findbp) 10B6		02390
(value)		05995
IF value < nbkpts then assume it is a break point number, otherwise		
an address of an instruction		05994
(findda) (nls,frontend,findda) 11L		01959
(dacords)		05342
find the window which contains the passed coordinates		05341
(finsort) (nls,colstr,finsort) 26		02332
(firsti) (nls,parser,)		0177
data declaration		03037
(firstx) (nls,parser,)		0109
data declaration		02962
(fixbnd) (nls,frontend,fixbnd) 11E		02022
(type,value,newvalue)		05438
fix boundaries		05437
(fixbuf) (nls,frontend,fixbuf) 11F		01962
reapportion buffer space among all windows		05347
(fixl) (nls,const,)		01021
data declaration		04139
(fixoutofbounds) (nls,nddt,fixoutofbounds) 10A3		02412
(rd,action,direction)		06016
(flaccm) (nls,records,)		01523
data declaration		04641

(flagut) (nls,utility,flagut) 16A	01574
(flagno,jsysno)	04697
This routine sets up and calls the flag Jsys', returning the appropriate results	04696
(flarf1) (nls,records,)	01065
data declaration	04183
(flastr) (nls,records,)	01249
data declaration	04367
(flbpar) (nls,records,)	01545
data declaration	04663
(flbpcs) (nls,records,)	01066
data declaration	04184
(flbrws) (nls,records,)	01349
data declaration	04467
(flbyts) (nls,records,)	01126
data declaration	04244
(fld1) (nls,ccalc,)	02537
data declaration	06167
(fld2) (nls,ccalc,)	02544
data declaration	06178
(fld3) (nls,ccalc,)	02527
data declaration	06155
(fldfr) (nls,records,)	01258
data declaration	04376
(fldirectory) (nls,ioexec,fldirectory) 9G	02199
(fl)	05743
Get directory number given address of first entry.	05742
(fldirn) (nls,records,)	01453
data declaration	04571
(fldmpt) (nls,records,)	01356
data declaration	04474
(fldsig) (nls,nddt,)	02352
data declaration	05955
(flexis) (nls,records,)	01541
data declaration	04659
(flfill) (nls,records,)	01451
data declaration	04569
(flfsat) (nls,records,)	01380
data declaration	04498
(flhead) (nls,records,)	01132
data declaration	04250
(fllock) (nls,records,)	01200
data declaration	04318
(flndr) (nls,select,flndr) 108	0230
(bug.ptr1,ptr2)	03112
file name -- TENEX syntax	03111
(flnwr) (nls,records,)	01269
data declaration	04387
(flnwrr) (nls,records,)	01184
data declaration	04302
(flnocl) (nls,records,)	01539
data declaration	04657
(flntadr) (nls,ioexec,flntadr) 9D	02142
(fileno)	05636
Returns the address of the file list entry for the file number passed	05635

(florig) (nls,records,)		01516
data declaration		04634
(flpart) (nls,records,)		01038
data declaration		04156
(flpcre) (nls,records,)		01337
data declaration		04455
(flpocst) (nls,records,)		01137
data declaration		04255
(flpogsz) (nls,records,)		01130
data declaration		04248
(flpovp) (nls,records,)		01159
data declaration		04277
(flprex) (nls,records,)		01507
data declaration		04625
(flprgr) (nls,records,)		01441
data declaration		04559
(flprli) (nls,records,)		01187
data declaration		04305
(flprnu) (nls,records,)		01492
data declaration		04610
(flprpu) (nls,records,)		01464
data declaration		04582
(flprre) (nls,records,)		01511
data declaration		04629
(flprse) (nls,records,)		01305
data declaration		04423
(flprwr) (nls,records,)		01416
data declaration		04534
(flsnat) (nls,records,)		01529
data declaration		04647
(fndchr) (nls,inpfof,fndchr) 9B2		0533
(da,xcrdnt,ycrdnt)		03644
find character position		03643
(fndtab) (nls,inpfbk,fndtab) 11A		0551
(da,column)		03669
find new tab position		03668
(fndunqtxt) (nls,utility,fndunqtxt) 16B		01654
(keytext,start,t1,t2)		04782
FIND unquoted text string		04781
(fnorma) (nls,const,)		0612
data declaration		03730
(fontms) (nls,const,)		0838
data declaration		03956
(formdi) (nls,pscal,)		0456
data declaration		03530
(forwar) (nls,const,)		0661
data declaration		03779
(forwdv) (nls,const,)		0565
data declaration		03683
(fpoint) (nls,nddt,)		02398
data declaration		06003
(framec) (nls,pdata,)		035
data declaration		02896
(framedef) (nls,nddt,framedef) 11B5B		02392
(rd)		05997
(framfielddef) (nls,nddt,framfielddef) 11B5C		02478

(rd)		06080
(freeblk) (nls, stgmt, freeblk) 3B		02213
(blk, zone)		05769
de-allocate a block in zone		05768
(freefrring) (nls, frontend, freefrring) 12B		02000
(frr)		05405
free file return ring		05404
(freeisrt) (nls, stgmt, freeisrt) 4E2		02218
(zone, da)		05779
deallocates a line segment reference table and updates da record		05778
(freesrring) (nls, frontend, freesrring) 12H		02017
(srr)		05430
free statement return ring		05429
(freestring) (nls, stgmt, freestring) 4A3		02212
(straddr, zone)		05767
also for arrays.		05766
(freeflnt) (nls, frontend, freeflnt) 3A		01984
free and close files for files in file status table		05378
(frrng) (nls, filmnp, frrng) 2C5		01823
(stid)		05089
free the ring element for STID given as argument		05088
(fresdb) (nls, filmnp, fresdb) 2I6		01820
(stid)		05083
If the statement associated with stid passed as argument has an SDB		
then set it to garbage.		05082
(frexis) (nls, records,)		01481
data declaration		04599
(frhexi) (nls, records,)		01080
data declaration		04198
(frhlas) (nls, records,)		01419
data declaration		04537
(frhtop) (nls, records,)		01075
data declaration		04193
(fromwh) (nls, pdata,)		044
data declaration		02905
(frrcnt) (nls, pdata,)		069
data declaration		02930
(frrle) (nls, records,)		01259
data declaration		04377
(frrhle) (nls, records,)		01060
data declaration		04178
(frrlength) (nls, frontend, frrlength) 12E		01995
(frr)		05398
return number of entries in the file return ring.		05397
(frrmax) (nls, const,)		0558
data declaration		03676
(frrsiz) (nls, const,)		0759
data declaration		03877
(frrri) (nls, records,)		01357
data declaration		04475
(frzblk) (nls, filmnp, frzblk) 2B4		01833
(pgindx, fr)		05107
Freeze block given as arguments		05106
(frzchk) (nls, frontend, frzchk) 5D		01992
(da, file1, file2)		05393

check frozen chain for file membership	05392
(frzfrmt) (nls,dspgen,frzfrmt) 3A5	01914
(da)	05255
display frozen statements	05254
(frzl) (nls,records,)	01043
data declaration	04161
(frzrfb) (nls,filnnp,frzrfb) 2B3	01814
(fileno,blkn,fr)	05072
This routine is called for all freezing and thawing of random file blocks. Arguments are	05071
(fs) (nls,const,)	0577
data declaration	03695
(fstar) (nls,records,)	01160
data declaration	04278
(fstatus) (nls,auxcod,fstatus) 11A	02280
(fileno,astng,type)	05868
(fstr) (nls,const,)	0913
data declaration	04031
(ftpapp) (nls,ioexec,ftpapp) 16B15	02127
(locfile,disfile,xfermode)	05609
append to file at distant system	05608
(ftpbgn) (nls,ioexec,ftpbgn) 16B1	02146
initialize for FTP activity	05643
(ftpcf) (nls,ioexec,ftpcf) 16D3	02128
(op,count,param1,param2)	05611
invoke FTP inferior fork	05610
(ftpcls) (nls,ioexec,ftpcls) 16B4	02106
close distant file system	05572
(ftpert) (nls,ioexec,ftpert) 16D1	02132
create FTP inferior fork	05618
(ftpdel) (nls,ioexec,ftpdel) 16B17	02197
(disfile)	05739
delete distant file	05738
(ftpdir) (nls,ioexec,ftpdir) 16B16	02145
(locfile,remdsg)	05642
retrieve distant directory listing	05641
(ftpdml) (nls,ioexec,ftpdml) 16B6	02184
(locfile,user)	05714
send mail to distant user	05713
(ftpelim) (nls,ioexec,ftpelim) 16B11	02204
(locfile)	05752
eliminate local file	05751
(ftpend) (nls,ioexec,ftpend) 16B2	02160
end of FTP activity	05668
(ftperr) (nls,ioexec,ftperr) 16B1	02102
(msg)	05565
dispatch FTP error	05564
(ftpex) (nls,ioexec,ftpex) 16C1	02123
(op,count,param1,param2,param3,param4,param5,param6)	05601
execute FTPFRK operation	05600
(ftpfml) (nls,ioexec,ftpfml) 16B9	02097
(locfile,author,authorhost,title,header,msg)	05555
format mail	05554
(ftpgts) (nls,ioexec,ftpgts) 16E2	02124
(string,addr)	05603
convert ASCII to NLS string	05602

(ftpkil) (nls,ioexec,ftpkil)	16D2	02167
kill FTP inferior fork		05681
(ftplml) (nls,ioexec,ftplml)	16B7	02179
(locfile,user)		05705
send mail to local user		05704
(ftplog) (nls,ioexec,ftplog)	16B5	02152
(user,pass,acct)		05654
login at distant system		05653
(ftpoptn) (nls,ioexec,ftpoptn)	16B3	02116
(host)		05590
open distant file system		05589
(ftppts) (nls,ioexec,ftppts)	16E3	02147
(string,addr)		05645
convert NLS to ASCII string		05644
(ftpqml) (nls,ioexec,ftpqml)	16B8	02209
(locfile,user,host,stagingdir)		05761
queue mail for user		05760
(ftprem) (nls,ioexec,ftprem)	16B20	02111
read FTP-primitive error mode		05580
(ftpren) (nls,ioexec,ftpren)	16B18	02141
(curfile,newfile)		05634
rename distant file		05633
(ftprph) (nls,ioexec,ftprph)	16B21	02133
(host)		05620
read open host name and address		05619
(ftprtr) (nls,ioexec,ftprtr)	16B14	02126
(locfile,disfile,xfermode)		05607
retrieve file from distant system		05606
(ftpsdd) (nls,ioexec,ftpsdd)	16B10	02112
(directory)		05582
set distant default directory		05581
(ftpsem) (nls,ioexec,ftpsem)	16B19	02168
(mode)		05683
set FTP-primitive error mode		05682
(ftpsig) (nls,const,)		0641
data declaration		03759
(ftpstn) (nls,ioexec,ftpstn)	16B13	02099
(locfile,disfile,xfermode)		05559
send file to distant system		05558
(ftpvhst) (nls,ioexec,ftpvhst)	16B12	02105
(host)		05571
verify host		05570
(ftypmo) (nls,nddt,)		02440
data declaration		06043
(fulldi) (nls,const,)		0994
data declaration		04112
(fullpr) (nls,const,)		0829
data declaration		03947
(fxsp1) (nls,utility,fxsp1)	9H6	01646
(fxsp2) (nls,utility,fxsp2)	9H7	01656
(fxswork) (nls,utility,fxswork)	9G	01577
(fzaxis) (nls,records,)		01358
data declaration		04476
(fznext) (nls,records,)		01091
data declaration		04209
(fzstid) (nls,records,)		01284

data declaration	04402
(fzvspc) (nls,records,)	01535
data declaration	04653
(fzvspe) (nls,records,)	01290
data declaration	04408
(g3nstd) (nls,colstr,g3nstd) 8G	02309
(loc)	05909
(gaddir) (nls,adrmnp,gaddir) 1R3	02089
(ptr,count)	05541
(gaddrlit) (nls,select,gaddrlit) 1E2	0228
(strptr)	03108
address of collection string	03107
(gaddr) (nls,const,)	0568
data declaration	03686
(gadname) (nls,adrmnp,gadname) 1R2	02088
(ptr,ast,type)	05540
(gadnum)* (nls,adrmnp,gadnum) 1R6	02090
extract and evaluate number from current READC source.	05542
(gbotids) (nls,cident,gbotids) 5P	02647
(ptr)	06374
collaps ident pusdown stack	06373
(gcatnums) (nls,catnum,gcatnums) 2A	02512
(retstr,idntsr,substr,count,type)	06132
Get count catalog numbers for subcollection(s) substr. type is	
passed to getcnum. Return numbers (separated by EOL's) in retstr if	
Non-zero	06131
(gcol) (nls,filmp,gcol) 2I5	01805
(sdbolk,fileno)	05055
called by newsdb to garbage collect the sdb block whose block number	
and file number are passed as arguments.	05054
(gdftdir) (nls,ioexec,gdftdir) 15J	02134
(fileno,string)	05622
Given a filenumber, puts owner name in a string	05621
(gdlit2) (nls,adrmnp,gdlit2) 1R5	02073
(ast,stopchar)	05512
append characters to ast from READC until encounter a stopchar or	
the end of input.	05511
(gdmys) (nls,ioexec,gdmys) 7C1	02205
(ptr)	05753
(gdname) (nls,ioexec,gdname) 15I	02193
(dirno,dname)	05731
Convert directory number and into a directory string. Calls err if	
error occurs.	05730
(getaddr) (nls,select,getaddr) 1E1	0215
(ptr)	03085
read and evaluate address expression	03084
(getail) (nls,filmp,getail) 2G1	01797
(stid)	05039
Given an stid, this procedure returns the stid of the tail of the	
current plex	05038
(getarray) (nls,stmgt,getarray) 4A2	02215
(length,zone)	05773
allocate an array in "zone" of "length" words plus block header.	
Initialize the array to empty. Return array address.	05772
(getbck) (nls,filmp,getbck) 2G7	01813
(stid)	05070

This procedure finds the back of the stid of the statement passed it. It does not observe viewspecs. 05069

(getblk) (nls,stgmt,getblk) 3A 02219
 (size,zone) 05781
 allocate a blk of "size" words in "zone" 05780

(getbug) (nls,select,getbug) 1B1C 0219
 (parsemode,tptr) 03093
 address of result tptr 03092

(getcacc) (nls,csendmail,getcacc) 7A 02666
 (entrystid,fldstr,ptr1,ptr2) 06409
 fetch access field from catalog entry 06408

(getcend) (nls,csendmail,getcend) 7C 02667
 (entrystid,ptr1,ptr2) 06411
 fetch end of catalog entry 06410

(getcfn) (nls,csendmail,getcfn) 7B 02671
 (entrystid,fldstr,ptr1,ptr2) 06418
 fetch filename from catalog entry 06417

(getchar) (nls,inpfbk,getchar) 3D 0539
 get an untranslated character from user 03650

(getcnum) (nls,catnum,getcnum) 2D 02511
 (identr,numbr,typesr,destination,stid) 06130
 This is the routine which is used for reserving catalogue numbers 06129

(getdae) (nls,select,getdae) 1B1D 0225
 (parsemode,tptr) 03102
 address of result tptr 03101

(getdat) (nls,auxcod,getdat) 9A 02281
 (astng) 05870
 put date and time in astring passed 05869

(getdev) (nls,intnls,getdev) 6G 02817
 (lineno,device) 06640
 get user's device 06639

(getend) (nls,filmap,getend) 2G5 01816
 (stid) 05075
 This procedure returns the end of the branch headed by the stid passed it. 05074

(getenf) (nls,ioexec,getenf) 9I 02107
 (fn,astr) 05574
 get EXTERNAL NAME LINK FILE name 05573

(getfacc) (nls,ioexec,getfacc) 1HJ 02131
 (originstid,fldstr,ptr1,ptr2) 05617
 fetch access field from file origin statement 05616

(getfhd) (nls,filmap,getfhd) 2H4 01841
 (stid) 05123
 The logical value of the head flag is returned. 05122

(getftl) (nls,filmap,getftl) 2H3 01825
 (stid) 05093
 The logical value of the tail flag is returned. 05092

(getgpids) (nls,cident,getgpids) 4C 02600
 (ptr,astr,idfnum) 06285
 given an identlist, return list of only the group idents 06284

(getgtjflg) (nls,ioexec,getgtjflg) 8G 02122
 (accessmode,chkpc,version) 05599
 get flags for gtjfn JSYS 05598

(gethed) (nls,filmap,gethed) 2G3 01806
 (stid) 05057

```

Given an stid, this routine returns the head of the current plex; if
it is passed the origin statement, it returns the origin          05056
(getiadd) (nls,cident,getiadd) 7B                                02638
  (entrystr, fldstr, ptr1, ptr2)                                06358
  get MAIL ADDRESS field from an ident entry                    06357
(geticapability) (nls,cident,geticapability) 7U                02584
  (entrystr, fldstr, ptr1, ptr2)                                06256
  get CAPABILITIES field from an ident entry                    06255
(geticord) (nls,cident,geticord) 7J                             02605
  (entrystr, fldstr, ptr1, ptr2)                                06295
  get COORDINATOR field from a group/organization ident entry  06294
(getidelivery) (nls,cident,getidelivery) 7W                     02601
  (entrystr, fldstr, ptr1, ptr2)                                06287
  get DELIVERY-TYPES field from an ident entry                  06286
(getids) (nls,cident,getids) 4B                                  02611
  (ptr.astr, infotype, idfile)                                  06307
  expand list of idents into list of individuals                06306
(getiend) (nls,cident,getiend) 8A                                02640
  (entrystr, ptr1, ptr2)                                        06362
  find the end of an ident entry                                06361
(getiexp) (nls,cident,getiexp) 7G                                02633
  (entrystr, fldstr, ptr1, ptr2)                                06348
  get EXPAND field from an ident entry                          06347
(getifnf) (nls,cident,getifnf) 7D                               02553
  (entrystr, fldstr)                                           06196
  get NAME field (FIRST NAME FIRST) field from an individual ident
  entry                                                         06195
(getifunction) (nls,cident,getifunction) 7S                     02564
  (entrystr, fldstr, ptr1, ptr2)                                06217
  get FUNCTION field from an ident entry                        06216
(getigrps) (nls,cident,getigrps) 7N                             02635
  (entrystr, fldstr, ptr1, ptr2)                                06352
  get GROUPS field from an ident entry                          06351
(getihost) (nls,cident,getihost) 7P                             02619
  (entrystr, fldstr, ptr1, ptr2)                                06321
  get NETWORK-DELIVERY HOST field from an ident entry          06320
(getiid) (nls,cident,getiid) 7A                                 02550
  (entrystr, fldstr, ptr1, ptr2)                                06190
  get IDENT field from an ident entry                           06189
(getilname) (nls,cident,getilname) 7E                           02570
  (entrystr, fldstr, ptr1, ptr2)                                06229
  get LAST NAME field from an individual ident entry            06228
(getimcmnts) (nls,cident,getimcmnts) 7X                         02596
  (entrystr, fldstr, ptr1, ptr2)                                06278
  get COMMENTS field from an ident entry                       06277
(getimem) (nls,cident,getimem) 7H                               02593
  (entrystr, fldstr, ptr1, ptr2)                                06273
  get MEMBERSHIP field from a group/organization ident entry  06272
(getinam) (nls,cident,getinam) 7C                               02578
  (entrystr, fldstr, ptr1, ptr2)                                06244
  get NAME field from an ident entry                            06243
(getinit) (nls,inthls,getinit) 6P                               02851
  (infostr)                                                    06678
  get user's id                                                06677
(getinlhost) (nls,cident,getinlhost) 7Q                         02620
  (entrystr, fldstr, ptr1, ptr2)                                06323

```

get NLS-DELIVERY HOST field from an ident entry	06322
(getinma) (nls,cident,getinma) 7R	02618
(entrystr, fldstr, ptr1, ptr2)	06319
get NETWORK MAILBOX Address field from an ident entry	06310
(getins) (nls,nddt2,getins) 1D2R	02499
(astr,instruction)	06100
(getint) (nls,filmap,getint) 2K1	01839
(stid)	05119
Called with STID, returns initials of associated statement (left justified in word zero filled).	05118
(getiorg) (nls,cident,getiorg) 7F	02575
(entrystr, fldstr, ptr1, ptr2)	06238
get ORGANIZATION field from an individual ident entry	06237
(getiphone) (nls,cident,getiphone) 7O	02554
(entrystr, fldstr, ptr1, ptr2)	06198
get PHONE field from an ident entry	06197
(getiprevmem) (nls,cident,getiprevmem) 7I	02565
(entrystr, fldstr, ptr1, ptr2)	06219
get MEMBERSHIP PREVIOUS field from an ident entry	06218
(getisorg) (nls,cident,getisorg) 7T	02588
(entrystr, fldstr, ptr1, ptr2)	06264
get SECONDARY ORGANIZATION field from an ident entry	06263
(getisubcol) (nls,cident,getisubcol) 7V	02636
(entrystr, fldstr, ptr1, ptr2)	06354
get SUBCOLLECTIONS field from an ident entry	06353
(getityp) (nls,cident,getityp) 7K	02656
(entrystr, fldstr, ptr1, ptr2)	06390
get TYPE-OF-ORGANIZATION field from an organization ident entry	06389
(getiuser) (nls,cident,getiuser) 7M	02652
(entrystr, fldstr, ptr1, ptr2)	06382
get USER (NLS delivery) field from an ident entry	06381
(getiverify) (nls,cident,getiverify) 7L	02645
(entrystr, fldstr, ptr1, ptr2)	06370
get VERIFIED-BY-NIC-PERSONNEL field from an ident entry	06369
(getjaction) (nls.csendmail,getjaction) 6H	02690
(string)	06455
get the contents of the Action Distribution field from the JWORK header	06454
(getjauthor) (nls.csendmail,getjauthor) 6I	02670
(string)	06416
get the contents of the Author field from the JWORK header	06415
(getjaxcess) (nls.csendmail,getjaxcess) 6J	02683
(string)	06441
get the contents of the Access List field from the JWORK header	06440
(getjclerk) (nls.csendmail,getjclerk) 6K	02706
(string)	06486
get the contents of the Clerk field from the JWORK header	06485
(getjcomment) (nls.csendmail,getjcomment) 6L	02663
(string)	06403
get the contents of the Comment field from the JWORK header	06402
(getjexpedite) (nls.csendmail,getjexpedite) 6M	02673
(string)	06422
get the Expedite attribute from the JWORK header	06421
(getjhcloc) (nls.csendmail,getjhcloc) 6O	02675
(string)	06425
get the contents of the Hard Copy Location field from the JWORK	

header		06424
(getjinfo) (nls,csendmail,getjinfo) 6P		02687
(string)		06449
get the contents of the Information-only Distribution field from the		
JWORK header		06448
(getjkeyw) (nls,csendmail,getjkeyw) 6R		02692
(string)		06459
get the contents of the Keywords field from the JWORK header		06458
(getjlink) (nls,csendmail,getjlink) 6Q		02676
(string)		06427
get the contents of the Insert Link field from the JWORK header		06426
(getjnumber) (nls,csendmail,getjnumber) 6S		02708
(string)		06490
get the contents of the Number field from the JWORK header		06489
(getjobsobletes) (nls,csendmail,getjobsobletes) 6U		02680
(string)		06435
get the contents of the Obsoletes Documents field from the JWORK		
header		06434
(getjrfo) (nls,csendmail,getjrfo) 6V		02710
(string)		06494
get the contents of the RFC field from the JWORK header		06493
(getjsubcol) (nls,csendmail,getjsubcol) 6W		02665
(string)		06407
get the contents of the Sub-Collections field from the JWORK header		
(getjtitle) (nls,csendmail,getjtitle) 6X		06406
(string)		02658
get the contents of the Title field from the JWORK header		06394
(getjtype) (nls,csendmail,getjtype) 6T		06393
(string)		02702
get the TYPE of the item from the JWORK header		06478
(getjunrecorded) (nls,csendmail,getjunrecorded) 6N		06477
(string)		02659
get the Unrecorded attribute from the JWORK header		06396
(getjupdates) (nls,csendmail,getjupdates) 6Y		06395
(string)		02679
get the contents of the Updates Documents field from the JWORK		06433
header		06432
(getlev) (nls,seqgen,getlev) 4H		01856
(stid)		05150
Called with STID, returns level of that statement.		05149
(getlit) (nls,select,getlit) 1B1B		0236
(parsemode,tptr1,tptr2)		03123
address of finishing tptr		03122
(getnam) (nls,filmp,getnam) 2H7		01842
(stid)		05125
The hash word for the statement name is returned.		05124
(getnmf) (nls,filmp,getnmf) 2H5		01827
(stid)		05097
The logical value of the name flag is returned.		05096
(getnxt) (nls,filmp,getnxt) 2G8		01826
(stid)		05095
This procedure finds the sequentially "next" statement, i.e. the		
substatement, successor, or successor of up, etc, of the stid passed		
as argument. Ignores all viewspecs.		05094
(getop) (nls,nddt,getop) 10A4		02427

(getpcjfn) (nls,execfil,getpcjfn) 3F	01958
(fjfn,pcjfn,errstring,isok)	05340
get jfn for pc of passed jfn	05339
(getpint) (nls,psupport,getpint) 8A	0244
(tpl.tp2)	03138
convert 2 text pointers to integer	03137
(getpr) (nls,adrmnp,getpr) 1R4	02091
(proc,count,ptr)	05543
(getprd) (nls,filmnp,getprd) 2G4	01840
(stid)	05121
Given an stid, this routine returns the predecessor; if the psid heads a plex, the stid itself is returned	05120
(getsbn) (nls,seqfil,getsbn) 6P	02258
(astrng)	05841
convert string passed to sixbit name	05840
(getsdb) (nls,filmnp,getsdb) 2H6	01828
(stid)	05099
The STDB in the specified ring element is returned.	05098
(getsgw) (nls,seqgen,getsgw) 3K	01860
get a sequence generator work area	05157
(getsid) (nls,filmnp,getsid) 2H8	01824
(stid)	05091
The statement identifier for the statement name is returned.	05090
(getstring) (nls,stgmt,getstring) 4A1	02211
(length,zone)	05765
allocate a block in "zone" of "length" characters plus block header plus string header. Initialize the string to empty. Return string address.	05764
(getstsize) (nls,filmnp,getstsize) 2K4	01843
(stid)	05127
Called with STID, returns size in characters for associated statement.	05126
(getsub) (nls,filmnp,getsub) 2H2	01817
(stid)	05077
The STID in the sub field is returned.	05076
(getsuc) (nls,filmnp,getsuc) 2H1	01803
(stid)	05051
The stid for the successor field is returned. If there is no successor, the stid of the up is returned.	05050
(getsym) (nls,ndat,getsym) 10A5	02465
(gettim) (nls,filmnp,gettim) 2K3	01794
(stid)	05033
Called with STID; returns time of last write for associated statement.	05032
(getup) (nls,filmnp,getup) 2G2	01819
(stid)	05081
Given an stid, this routine returns the stid of the source of the original stid	05080
(getversn) (nls,ioexec,getversn) 15D	02177
(string)	05701
returns the version number associated with the file name passed it	05700
(getvnd) (nls,filmnp,getvnd) 2G6	01837
(stid,level)	05115
find end of branch	05114
(getwindow) (nls,select,getwindow) 1B1E	0216

(parsemode, tptr)		03087
address of result tptr		03086
(gnstid) (nls, colsr, gnstid) 16		02331
Return next stid in sequence		05930
(gnxtid) (nls, cident, gnxtid) 4A		02561
(idstr, filenc)		06212
given an ident, generate a new one by adding a number		06211
(gofork) (nls, auxcod, gofork) 4M		02275
(prenam, injfn, outjfn, ntiw)		05862
goto lower fork		05861
(goodrng) (nls, filmp, goodrng) 202		01798
(stid)		05041
Returns TRUE iff stid points to a ring element which is in use		05040
(goroot) (nls, auxcod, goroot) 4B		02279
Goto root		05867
(gostid) (nls, colsr, gostid) 8F		02314
(loc)		05913
(gotohelp) (nls, parser, gotohelp) 4E		0175
save the accumulators		03035
(gparam) (nls, pdata,)		02
data declaration		02863
(gparm2) (nls, pdata,)		031
data declaration		02892
(gparm3) (nls, pdata,)		07
data declaration		02868
(gparm4) (nls, pdata,)		067
data declaration		02928
(gpbsz) (nls, usrpgm, gpbsz) 3		02726
(size)		06521
Goto User program Buffer Size Set		06520
(gpconan) (nls, usrpgm, gpconan) 5		02724
(tpb, da)		06517
does Goto Program Content analyzer		06516
(gpexpgm) (nls, usrpgm, gpexpgm) 6		02719
(str)		06510
does Goto Program Execute a program		06509
(gpfktrm) (nls, usrpgm, gpfktrm) 15A		02721
print tenex subsystem status from within fork termination psi		
routine		06512
(gpget) (nls, usrpgm, gpget) 7		02727
(adstr)		06523
does Goto Programs Get Rel File command		06522
(gpgmrst) (nls, usrpgm, gpgmrst) 4		02720
Goto User program reset		06511
(gpkill) (nls, usrpgm, gpkill) 12		02723
does: KILL TENEX SUBSYSTEM		06515
(gp110) (nls, usrpgm, gp110) 8		02718
(stid, da)		06508
does Goto Program L10 program compile		06507
(gppop) (nls, usrpgm, gppop) 9		02732
does Goto Program Pop a program		06530
(gprcname) (nls, nddt, gprcname) 10H1		02475
(gprnls) (nls, usrpgm, gprnls) 15B		02715
(ttymode)		06503
restore nls display		06502
(gprunt) (nls, usrpgm, gprunt) 13		02722

(adstr,rhosto,outfil,mode,rhostt,tand)	06514
does: RUN TENEX SUBSYSTEM	06513
(gps) (nls,auxcod,gps) 4C	02297
GOTO STATE routine	05894
(gpstatus) (nls,usrpgrm,gpstatus) 10	02716
(str.da)	06505
does goto Program Status display	06504
(gptxst) (nls,usrpgrm,gptxst) 11	02733
(fork,astr)	06532
does: SHOW TENEX SUBSYSTEM STATUS	06531
(gramms) (nls,pdata,)	072
data declaration	02933
(grdtyp) (nls,records,)	01391
data declaration	04509
(group) (nls,const,)	0954
data declaration	04072
(groupss) (nls,cident,groupss) 16A1C	02616
(ident)	06315
(groupv) (nls,const,)	0600
data declaration	03718
(grpadr) (nls,const,)	0826
data declaration	03944
(grpdr) (nls,select,grpdr) 1B2B	0224
(tpr1,tpr2,ret1,ret2)	03100
address of the second result text pointer	03099
(grpst) (nls,filmap,grpst) 2N1	01822
(stid1,stid2)	05087
Given two stid's, this routine checks that they specify a legal	
group; it also returns them ordered (GRP1,GRP2). If the stid's do	
not form a legal group, an err("\$illegal group") is issued.	05086
(grptyp) (nls,const,)	0792
data declaration	03910
(gsstid) (nls,colstr,gsstid) 7E	02323
(st1,st2)	05923
(gstr) (nls,const,)	0921
data declaration	04039
(gtcatent) (nls,csendmail,gtcatent) 7D	02696
(docnumber,catfilename,catfileno,openformodify)	06466
locate document in catalogs	06465
(gtdesjfn) (nls,execfl,gtdesjfn) 3P	01956
(jfn1,pcjfn1,udir2,ufil2,uext2,uver2,ulft2,flag2,errstring)	05336
get destination jfns	05335
(gthstn) (nls,execfl,gthstn) 3C	01932
(hp1,hp2)	05289
returns host number for input string text pointers	05288
(gtintmsg) (nls,intnls,gtintmsg) 6AB	02813
no locals/arguments	06636
(gtjerr) (nls,ioexec,gtjerr) 8F	02194
(error,string,flags)	05733
Convert error word from gtjfn into string	05732
(gtjid1) (nls,const,)	0634
data declaration	03752
(gtinfo) (nls,const,)	0995
data declaration	04113
(gtjoif) (nls,const,)	0767
data declaration	03885

(gtjois) (nls,const,)		0748
data declaration		03866
(gtjoof) (nls,const,)		0677
data declaration		03795
(gtjoos) (nls,const,)		01032
data declaration		04150
(gtjprf) (nls,const,)		0750
data declaration		03868
(gtjprv) (nls,const,)		0779
data declaration		03897
(gtjstr) (nls,const,)		0594
data declaration		03712
(halt) (nls,auxcod,halt) 4E		02296
terminate NLS		05893
(hash) (nls,utility,hash) 8A@		01670
(ast)		04801
An a-string address as argument. The hash code for that string is returned.		04800
(hashdump) (nls,nddt,hashdump) 12B2		02423
Type out the contents of the hash table		06028
(hashenter) (nls,nddt,hashenter) 12B3		02448
(string,value,record,rcdlength)		06052
enter string if not already entered		06051
(hashlookup) (nls,nddt,hashlookup) 12B4		02381
(string)		05986
(hashtbinit) (nls,nddt,hashtbinit) 12B5		02400
(hcopyv) (nls,const,)		0842
data declaration		03960
(hdtyp) (nls,const,)		0743
data declaration		03861
(he) (nls,const,)		0566
data declaration		03684
(helphp) (nls,help,helphp) 1B5		02808
(hinc0) (nls,frontend,)		01982
data declaration		05376
(hinc1) (nls,frontend,)		01975
data declaration		05367
(hinc2) (nls,frontend,)		01970
data declaration		05358
(hinc3) (nls,frontend,)		01963
data declaration		05348
(hioco) (nls,csendmail,hioco) 11D		02686
(outcome,acs)		06447
co-routine return to calling fork		06446
(hiop) (nls,seqfil,hiop) 4E		02254
(acs)		05835
NLS Host Input/Output Processor		05834
(hiortn) (nls,csendmail,hiortn) 11C		02682
(outcome,hdr,body,ur1r)		06439
return to calling fork		06438
(hlpcmd) (nls,pdata,)		083
data declaration		02944
(hlpstrt) (nls,help,hlpstrt) 1B2		02756
(filptr,namarray)		06560
(hlpsub) (nls,pdata,)		011
data declaration		02872

(hmapin) (nls,seqfil,hmapin) 6D	02256
(hmapout) (nls,seqfil,hmapout) 6C	02259
(hndacc) (nls,seqfil,)	02252
data declaration	05832
(hostnumber) (nls,intnls,hostnumber) 6A@	02810
get logical host number from tenex	06634
(hrldmo) (nls,const,)	01003
data declaration	04121
(hrldsi) (nls,const,)	0668
data declaration	03786
(hrldst) (nls,pdata,)	057
data declaration	02918
(hs) (nls.const,)	0719
data declaration	03837
(hseger) (nls,help,)	02799
data declaration	06621
(hsplit) (nls,frontend,hsplit) 11B	01979
(top,cords1,cords2)	05373
split window horizontally	05372
(hstnmf) (nls,records,)	01557
data declaration	04675
(hstnmi) (nls,records,)	01321
data declaration	04439
(hstnmn) (nls,records,)	01134
data declaration	04252
(hstr) (nls,const,)	0791
data declaration	03909
(ibotto) (nls,intnls,)	02859
data declaration	06687
(icsize) (nls,intnls,)	02041
data declaration	06665
(idchr) (nls,cident,)	02632
data declaration	06346
(idd) (nls,records,)	01165
data declaration	04283
(idfsearch) (nls,cident,idfsearch) 13F	02627
(type,idstr,infostr,fileno)	06337
search ident-file for last name or content-in-name	06336
(idfirmuser) (nls,intnls,idfirmuser) 6R	02838
(str)	06663
collect ident from user in tty mode	06662
(idgrporg) (nls,cident,idgrporg) 13E	02628
(string,idstr)	06339
format abbreviated group or organization entry	06338
(idpind) (nls,cident,idpind) 13D	02613
(string,idstr)	06311
format and print abbreviated individual's ident entry	06310
(idplname) (nls,cident, IDPLNAME) 13C	02579
(stid,idstr)	06246
print a last name branch of individuals	06245
(idr) (nls,select,idr) 1C2	0221
(bug.ptr1,ptr2)	03095
(ids) (nls,records,)	01328
data declaration	04446
(idstatus) (nls,cident,idstatus) 10A	02602
(idstr.entrystr,typid,statstr)	06289

generate status string for an entry	06288
(idtst) (nls,utility,idtst) 9T	01575
(astrng)	04698
(ifont) (nls,intnls,)	02830
data declaration	06654
(ihinc) (nls,intnls,)	02812
data declaration	06635
(ileft) (nls,intnls,)	02839
data declaration	06664
(ilsdefault) (nls,utility,ilsdefault) 13A2A	01576
(ilspci) (nls,utility,ilspci) 13A1A	01565
(imlaco) (nls,const,)	0848
data declaration	03966
(imlacl) (nls,const,)	0739
data declaration	03857
(inOstca) (nls,prmspc,inOstca) 7A	02028
(astrng)	05448
(inOtca) (nls,prmspc,inOtca) 5A	02044
(astrng)	05465
(inlca) (nls,prmspc,inlca) 4A	02034
(bgl)	05455
(inlsadj) (nls,prmspc,inlsadj) 8A	02056
(bgl.astrng)	05482
(inlsca) (nls,prmspc,inlsca) 6A	02049
(bgl)	05470
(inlstca) (nls,prmspc,inlstca) 7B	02058
(bgl.astrng)	05484
(inltca) (nls,prmspc,inltca) 5B	02039
(bgl.astrng)	05460
(in2ca) (nls,prmspc,in2ca) 4B	02029
(bgl.bg2)	05449
(in2sadj) (nls,prmspc,in2sadj) 8B	02046
(bgl.bg2.astrng)	05467
(in2sca) (nls,prmspc,in2sca) 6B	02057
(bgl.bg2)	05483
(in2stca) (nls,prmspc,in2stca) 7C	02048
(bgl.bg2.astrng)	05469
(in2tca) (nls,prmspc,in2tca) 5C	02031
(bgl.bg2.astrng)	05451
(in3ca) (nls,prmspc,in3ca) 4C	02061
(bgl.bg2.bg3)	05489
(in3sadj) (nls,prmspc,in3sadj) 8C	02038
(bgl.bg2.bg3.astrng)	05459
(in3sca) (nls,prmspc,in3sca) 6C	02030
(bgl.bg2.bg3)	05450
(in3stca) (nls,prmspc,in3stca) 7D	02036
(bgl.bg2.bg3.astrng)	05457
(in3tca) (nls,prmspc,in3tca) 5D	02047
(bgl.bg2.bg3.astrng)	05468
(inlca) (nls,prmspc,inlca) 4D	02050
(bgl.bg2.bg3.bg4)	05471
(inlsca) (nls,prmspc,inlsca) 6D	02063
(bgl.bg2.bg3.bg4)	05492
(inbug) (nls,prmspc,inbug) 3A	02045
(bg)	05466
(incarb) (nls,utility,incarb) 9J	01590

update position on stack where last succeeded	04710
(inch) (nls,prmspc,inch) 3K	02040
(char)	05461
(incont) (nls,prmspc,incont) 3F	02051
(string)	05473
puts bugged text entity in string passed it	05472
(incrversn) (nls,ioexec,incrversn) 15H	02176
(string)	05699
Given the address of a string containing a file name, this routine will increment the number in the string representing the version number (if the string has no ";nn" in it, the routine returns FALSE.	05698
(incspc) (nls,credit,incspc) 10I	01762
(ptr1,ptr2)	04973
(incues) (nls,const,)	01022
data declaration	04140
(indcnt) (nls,const,)	0818
data declaration	03936
(index1) (nls,nddt,)	02459
data declaration	06063
(indir1) (nls,nddt,)	02384
data declaration	05989
(inds) (nls,cident,)	02562
data declaration	06213
(indtyp) (nls,const,)	01020
data declaration	04138
(infilename) (nls,prmspc,infilename) 3M	02066
(string)	05498
input a file name (DNLS)	05497
(init) (nls,intnls,init) 5A	02826
initialize the stacks	06649
(initbtbl) (nls,inpfbk,initbtbl) 10B	0525
initializes the global array SYSBREAKTBL to contain appropriate addresses for RDLIT. This routine should be called after modifying the input translation table TRNSLI.	03630
(initch) (nls,intnls,initch) 6K	02836
(device)	06660
initialize translation tables	06659
(initcl) (nls,colsrt,initcl) 22	02327
(initda) (nls,intnls,initda) 60	02818
(da,base,x,y,str1,str2,type)	06641
(initdis) (nls,intnls,initdis) 6L	02835
(initframe) (nls,nddt,initframe) 11B5D	02438
This procedure sets up the relevant globals for the stack frame stuff.	06042
(initht) (nls,nddt,initht) 12B6	02404
Initialize DDT internal Hash Table	06008
(initjwork) (nls,csendmail,initjwork) 3A	02668
Initialize JWORK file	06412
(initrd) (nls,nddt,initrd) 11A1	02493
(rd)	06098
Initialises an RD...Assumes that the following fields are valid when called:	06097
(initsubsystems) (nls,parser,initsubsystems) 4A	0207
defines the "normal" subsystems available from NLS	03073
(inlevadj) (nls,prmspc,inlevadj) 30	02055

(astrng)		05481
this procedure gets characters for LEVADJ. It collects characters until it encounters something other than a U or D (or backspace). It puts these characters into the string passed it and displays them in the name register. Note that the string will be empty upon RETURN if the user does no level adjust.		
(inline) (nls,dspgen,inline)	7J	01885
(x,y,string,standoutflag)		05205
insert line - alphanumeric displays		
(iname) (nls,prmspc,inname)	3I	02042
(astrng)		05463
(innmwd) (nls,prmspc,innmwd)	3J	02062
(astrng,wordflag,delimproc)		05491
This routine reads characters from the work station, appends them to a register, and displays them in the nameregister. Alphabetic characters are forced to upper case before insertion into the A-string if the wordflag is false. The argument should be the address of the register into which the A-string is to be put. (Note that this routine does not clear the A-string before appending characters to it.)		
(innum) (nls,prmspc,innum)	3D	02032
(astrng)		05453
This routine reads digits from the work station, appends		
(inpcuc) (nls,inpfbk,inpcuc)	3A	0524
input a character, force upper case		
(inpcur) (nls,records,)		03629
data declaration		
(inoptr) (nls,records,)		01514
data declaration		
(inprom) (nls,const,)		04632
data declaration		
(inpsav) (nls,parser,)		01372
data declaration		
(inpsel) (nls,records,)		04490
data declaration		
(inpt) (nls,pdata,)		0676
data declaration		
(input) (nls,inpfbk,input)	3B	03794
character input (TYPEWRITER or DISPLAY)		
(inpvsp) (nls,inpfbk,inpvsp)	10G	0148
(char)		03005
process view specs		
(inrems) (nls,const,)		0148
data declaration		
(inscha) (nls,pdata,)		03005
data declaration		
(insd) (nls,filmp,insd)	2F5	01458
(stid,grp1,grp2)		04576
Given an stid, this routine inserts the group defined by the second and third arguments down from the first argument.		
(inseq) (nls,seqfil,inseq)	4A	042
(stid,rlevcnt,ifilnam,filtyp)		02903
does the Input Sequential command		
(inseq1) (nls,seqfil,inseq1)	4D	0514
(jfn,line,justified)		03615
gets a line for inseqn		
		048
		03579
		03578
		0805
		03923
		087
		02948
		01831
		05104
		05103
		02237
		05811
		05810
		02241
		05817
		05816

(inseqn) (nls,seqfil,inseqn) 4C	02257
(stid,rlvcnt,ifilnam,just)	05839
does the input sequential command (new way): double CR means end of st.	05838
(insgrp) (nls,filmp,insgrp) 2F3	01793
(target,dir,src1,src2)	05031
Insert SRC1, SRC2 at TARGET according to DIR. If DIR is true, then the group is inserted as the successor; otherwise it is inserted as the sub. (SRC1 and SRC2 are assumed to define a legal, ordered group.)	05030
(insize) (nls,pscalc,)	0459
data declaration	03533
(insmkr) (nls,auxcod,insmkr) 8C	02273
(bug.name)	05858
insert marker	05857
(insptr) (nls,parser,)	0108
data declaration	02961
(insr) (nls,prmspc,insr) 3E'	02067
(astrng)	05500
This routine reads characters from the work station,	05499
(inss) (nls,filmp,inss) 2F4	01799
(stid,grp1,grp2)	05043
Given an stid, this routine inserts a group, defined by the second and third arguments, as the suc of the first argument. First it makes the suc of STID the suc of GRP2, and updates the tail flag; then it makes GRP1 the suc of STID and updates the head and tail flags.	05042
(instid) (nls,prmspc,instid) 3B	02037
(bg)	05458
(insym) (nls,nddt,insym) 10A7N	02447
(symbol)	06050
this routine identifies the given symbol as a PDP10 instruction and returns the skeleton for the instruction if found, FALSE otherwise	06049
(intdaf1) (nls,intnls,intdaf1) 6D	02846
(da)	06670
initialize da record to standard values for a text area	06669
(intdex) (nls,intnls,intdex) 6S	02834
initialize DEX	06658
(intdspda) (nls,intnls,intdspda) 6I	02855
(tat,tal,tar)	06682
set up display areas for display terminal (relative to position of text area)	06681
(interp) (nls,const,)	0930
data declaration	04048
(intext) (nls,prmspc,intext) 3C	02065
(astrng)	05496
This routine reads literal input from a display terminal, appends it to an A-string, and displays it in the literal feedback area. The argument passed this routine should be the address of the A-string to which the literal input is to be appended. (Note that this routine does not clear the A-string before it begins reading characters.) TEXT = arbitrary number of characters up to but not including a CA or Center dot. DOES A SKIP RETURN IF A BC OR BW IS INPUT AND THE STRING IS EMPTY	05495
(intfil) (nls,ioexec,intfil) 7A	02185

(fileno)		05716
initialize file-- initialize header and create dummy origin statement		05715
(intids) (nls,cident,intids) 5A		02606
(ptr)		06297
initialize ident pushdown stack		06296
(intmeas) (nls,intnls,intmeas) 6AA		02857
measurement initialization		06684
(intmsf) (nls,intnls,)		02814
data declaration		06637
(intmsg) (nls,intnls,)		02854
data declaration		06680
(intter) (nls,dspgen,intter) 7M		01904
interrogate alphanumeric displays		05236
(invisi) (nls,const,)		0868
data declaration		03986
(invisv) (nls,const,)		0902
data declaration		04020
(invsbl) (nls,prmspc,invsbl) 3G		02041
(astrng)		05462
(invspc) (nls,prmspc,invspc) 3M		02027
(srcdpa)		05447
This routine reads characters from the work station and changes viewspecs. The original viewspecs are taken from the appropriate display area and stored in sysvspec. The viewspecs changed are those in sysvspec. That's where the new viewspecs will be on return.. It sets the viewspecs large upon entry, and small upon exit.		
A CD does a GOTO STATE. A CA or Centerdot causes a return of the display area where the cursor is. No need to call dpset.		05446
(inword) (nls,prmspc,inword) 3R		02035
(astrng)		05456
(iodaterr) (nls,auxcod,iodaterr) 4H		02298
Come here on an IO data Error (channel 11)		05895
(ipchar) (nls,records,)		01109
data declaration		04227
(ipcoor) (nls,records,)		01548
data declaration		04666
(ipstid) (nls,records,)		01332
data declaration		04450
(iptime) (nls,records,)		01171
data declaration		04289
(iright) (nls,intnls,)		02844
data declaration		06667
(isbuff) (nls,records,)		01412
data declaration		04530
(isdpth) (nls,records,)		01449
data declaration		04567
(isechon) (nls,inpfbk,isechon) 8C		0522
returns true if echoing on		03626
(isenabled) (nls,execfl,isenabled) 3G		01945
determine if this user is an enabled wheel		05314
(isfilt) (nls,records,)		01240
data declaration		04358
(isindividual) (nls,psident,isindividual) 15		0369
(resultptr,parsemode)		03386

is the current record for an individual	03385
(isinfn) (nls,records,)	01461
data declaration	04579
(islevb) (nls,records,)	01510
data declaration	04628
(islevs) (nls,records,)	01323
data declaration	04441
(islstb) (nls,records,)	01098
data declaration	04216
(isnewrec) (nls,psident,isnewrec) 8	0375
(resultptr,parsemode)	03398
is the current record a new record	03397
(isread) (nls,seqfil,isread) 4B	02243
(jfn,inbuff,lstchr)	05821
This procedure is used during insert sequential to read another character from the input file. It expects the input jfn, the address of a string, and a terminating character, as arguments, in that order. It puts the characters read into the string; it returns true until it has an empty input buffer and encounters an EOF.	05820
(isstid) (nls,records,)	01100
data declaration	04218
(istatn) (nls,records,)	01534
data declaration	04652
(itop) (nls,intnls,)	02822
data declaration	06645
(iupcas) (nls,const,)	0644
data declaration	03762
(iupper) (nls,cedit,iupper) 25D	01784
(ptr1,ptr2)	05014
iupper edits the text string between text pointers ptr1 and ptr2, forcing the first alpha of each word to upper case	05013
(ivinc) (nls,intnls,)	02821
data declaration	06644
(iwheel) (nls,psident,iwheel) 11	0366
(resultptr,parsemode)	03380
is the user an ident-wheel	03379
(jclosfl) (nls,ioexec,jclosfl) 17A	02138
Close any journal files which may be open	05628
(jctlcres) (nls,auxcod,jctlcres) 5H	02263
(jflname) (nls,ioexec,jflname) 17B	02101
(astr)	05563
The PROCEDURE ACCEPTS THE NAME PART OF A Journal file name as a parameter, and constructs a full name from it.	05562
(jfnflink) (nls,ioexec,jfnflink) 15G	02149
(fjfn,jfnname,flags)	05649
converts jfn to file link string	05648
(jfnstr) (nls,ioexec,jfnstr) 15E	02200
(jfn.string)	05745
puts full file name for jfn passed into string	05744
(jfnstr) (nls,ioexec,jfnstr) 15F	02140
(jfn,string,format)	05632
converts jfn to a full file name	05631
(jgrptst) (nls,cident,jgrptst) 9A	02574
(string,dstptr)	06236
test if a group entry	06235
(jlog) (nls,csendmail,jlog) 12B	02684

(number, type, identsr)	06443
Type Message to Journal Logging tty.	06442
(jnktyp) (nls, const,)	0589
data declaration	03707
(jnljn) (nls, pssystem, jnljn) 5D	0389
Roll in JNLDEL file	03424
(jnljnm) (nls, const,)	01002
data declaration	04120
(jnlout) (nls, pssystem, jnlout) 5E	0387
Roll out JNLDEL file	03421
(jnlpsw) (nls, const,)	0880
data declaration	03998
(jnltim) (nls, const,)	0949
data declaration	04067
(jocctlc) (nls, auxcod, jocctlc) 5G	02267
Come here on a control c	05848
(jolock) (nls, csendmail, jolock) 9A	02695
This procedure checks the global flags which tell the current status of the Journal:	06464
(jrinit) (nls, frontend, jrinit) 4B2	0275
(resultptr, parsemode, entity)	03199
ptr to entity record	03198
(jsetfield) (nls, pssendmail, jsetfield) 2N	0337
(resultptr, parsemode, field, value)	03322
set the specified field to the specified value	03321
(jstatus) (nls, csendmail, jstatus) 5A	02694
(status)	06463
generate a submission status string	06462
(jsubmit) (nls, csendmail, jsubmit) 10A	02674
do the Journal submission or secondary distribution	06423
(jsys) (nls, pdata,)	026
data declaration	02887
(jtypchk) (nls, csendmail, jtypchk) 12A	02707
(maxchars, jstid)	06488
check number of chars in a jworkfile	06487
(jump) (nls, const,)	0718
data declaration	03836
(just) (nls, ccalc,)	02530
data declaration	06158
(jutval) (nls, usrpgm,)	02730
data declaration	06528
(keydone) (nls, colsrt, keydone) 8D	02351
(loc)	05954
(keyinit) (nls, parser, keyinit) 6B3	0106
(curptr, work, lcmdmode)	02959
keyinit sets up a static work record for use by nextkey to sequence through the alternative keyword lists	02958
(keyinl) (nls, parser,)	0164
data declaration	03024
(keyinp) (nls, pdata,)	070
data declaration	02931
(keyrpt) (nls, pdata,)	020
data declaration	02881
(keysav) (nls, pdata,)	078
data declaration	02939
(keywrec) (nls, parser, keywrec) 6A1	0210

(resultptr, parsemode, curptr)	03078
ptr to path stack entry	03077
(kidrec) (nls, nddt,)	02491
data declaration	06095
(killib) (nls, intnls, killib) 6X	02840
(kprget) (nls, colsrt, kprget) 6A	02348
get key procedure address	05950
(kps) (nls, utility, kps) 5C	01633
(asirom)	04762
argument points to string to be appended to the statement being constructed	04761
(ksavef) (nls, parser,)	0136
data declaration	02994
(kstr) (nls, const,)	0720
data declaration	03838
(kysrdone) (nls, colsrt, kysrdone) 7G	02322
(stid, loc, skent)	05922
(llonam) (nls, const,)	0690
data declaration	03808
(llcmd) (nls, parser,)	0135
data declaration	02993
(laddress) (nls, cident, laddress) 6D	02646
(entrystr, fldstr, ptr1, ptr2, fileno)	06372
get LOGICAL U.S. MAIL ADDRESS for an ident entry	06371
(lastse) (nls, pdata,)	065
data declaration	02926
(lbound) (nls, frontend,)	02009
data declaration	05418
(lcard) (nls, records,)	01335
data declaration	04453
(lccords) (nls, frontend, lccords) 11K	01997
get coordinates associated with last control character input	05401
(lccsp) (nls, frontend, lccsp) 11H	01980
find Current Statement Pointer (CSP) for the window containing the cursor during the last input control char	05374
(lcta) (nls, frontend, lcta) 11I	01969
find the window containing the cursor during the last input control character	05357
(lcfiler) (nls, frontend, lcfiler) 11G	01998
find the file represented in the window containing the cursor during the last input control char	05402
(lda) (nls, frontend, lda) 11J	01977
returns address of Display Area descriptor for the display area where the cursor was the last time a character was input	05370
(ldchr) (nls, utility, ldchr) 8F2	01664
(ldelivery) (nls, cident, ldelivery) 6A	02655
(stid)	06388
get LOGICAL DELIVERY TYPE(S) for an ident entry	06387
(ldr) (nls, select, ldr) 1C9	0223
(bug, ptr1, ptr2)	03098
(le) (nls, const,)	0609
data declaration	03727
(levdow) (nls, const,)	0828
data declaration	03946
(level) (nls, pdata,)	08
data declaration	02869

(levnam) (nls, chelp, levnam) 1D11	02801
(stid, namstr, nhash, levels)	06625
lookup name in file starting at stid and terminating when through with its branch headed by lbstid, looking down levels levels checking names only on the last level.	06624
(levset) (nls, filmno, levset) 2F7	01847
(stid, levstg)	05133
Determines target stid and direction for inserting statements. Given an stid and the address of a string containing u's and d's (a levadj string), returns an stid and -1 if to be inserted down, 0 if to be inserted as successor to returned stid.	05132
(levsuc) (nls, const,)	0578
data declaration	03696
(levup) (nls, const,)	0627
data declaration	03745
(lfdstr) (nls, const,)	0869
data declaration	03987
(lfjln) (nls, records,)	01271
data declaration	04389
(lfn) (nls, const,)	0640
data declaration	03766
(lgetjfn) (nls, icexec, lgetjfn) 8C	02154
(dirnam, flnam, extnsn, flags, errstr)	05657
long gtjfn jsys-- set up registers and execute long gtjfn. Returns jfn for file whose name is specified in the parameters if successful, FALSE with error message in string whose address is passed if not. If &dirnam is 0, then the file name is checked to see if it is a Journal item and the default directory is taken as the one the user is connected to.	05656
(lgetsubcoll) (nls, cident, lgetsubcoll) 6C	02626
(entrystr, fldstr, ptr1, ptr2)	06335
get LOGICAL SUBCOLLECTIONS for an ident entry	06334
(lgngrp) (nls, const,)	0992
data declaration	04110
(lh) (nls, records,)	01371
data declaration	04489
(lhjb10) (nls, records,)	01398
data declaration	04516
(lhjb11) (nls, records,)	01555
data declaration	04673
(lhjb12) (nls, records,)	01210
data declaration	04328
(lhjb13) (nls, records,)	01359
data declaration	04477
(lhjb6) (nls, records,)	01243
data declaration	04361
(lhjb7) (nls, records,)	01174
data declaration	04292
(lhjb8) (nls, records,)	01189
data declaration	04307
(lhjb9) (nls, records,)	01334
data declaration	04452
(lhjrun) (nls, records,)	01238
data declaration	04356
(lhword) (nls, records,)	01353
data declaration	04471

(libstrt) (nls,intnls,libstrt) 6W	02828
(libno)	06652
This procedure tries to login and start a background processor. It may do a directory connect too depending on the processor	06651
(linebreak) (nls,inpfbk,linebreak) 9H8	0541
break the last line in the literal feedback area	03652
(link) (nls,const,)	0988
data declaration	04106
(linkup) (nls,stgmt,linkup) 3G	02220
(blk.zone)	05783
linkup blk into appropriate free list	05782
(linkv) (nls,const,)	0727
data declaration	03845
(linmax) (nls,const,)	0996
data declaration	04114
(litbc) (nls,inpfbk,litbc) 5E3	0513
(char,prevchar)	03614
backspace character in literal area	03613
(litchr) (nls,inpfbk,litchr) 5E2	0550
(char)	03667
append char to literal display area	03666
(litdpy) (nls,inpfbk,litdpy) 9H6	0517
(astrng)	03621
clear file text display windows and display contents of string in literal area	03620
(litera) (nls,pdata,)	081
data declaration	02942
(litpbout) (nls,inpfbk,litpbout) 5E1	0532
(char)	03642
append char to literal display area	03641
(litptr) (nls,parser,)	0100
data declaration	02950
(litsup) (nls,inpfbk,litsup) 9H9	0527
(ycoord)	03634
suppress display of any line segments above ycoord on the screen	03633
(littak) (nls,pdata,)	071
data declaration	02932
(littolnk) (nls,select,littolnk) 1E3	0226
(type,tp1,tp2)	03104
turn a lit into a link maybe	03103
(littty) (nls,inpfbk,littty) 9H7	0493
(astrng)	03585
clear file text display windows and make lit area a tty window	03584
(litv) (nls,const,)	0554
data declaration	03672
(lkdirn) (nls,ioexec,)	02110
data declaration	05579
(lkfile) (nls,filmp,lkfile) 2L6	01809
(fl)	05062
(lkinit) (nls,ioexec,)	02188
data declaration	05721
(lkmkr) (nls,inpfbk,lkmkr) 10D	0530
(astrng,fileno)	03638
translate marker name to t-ptr	03637
(lkun) (nls,filmp,lkun) 2L4	01834

(fil, astrng)	05109
Resets user setable word to unlock	05108
(lkupfast) (nls, adrmnp, lkupfast) 1R&D	02082
(stid, start, value, field)	05529
(lmemlist) (nls, cident, lmemlist) 6E	02612
(entrystr, outlist, explist, noexplist, errorlist, idfile)	06309
get LOGICAL MEMBERSHIP for a group/organization ident	06308
(lname) (nls, cident,)	02643
data declaration	06366
(lnbafn) (nls, adrmnp, lnbafn) 1E	02079
(astr, fileno)	05524
append file name string to passed string	05523
(lnbfls) (nls, adrmnp, lnbfls) 1D	02066
(stp, adstr, astr)	05537
create file name string from link parse block	05536
(lnk2er) (nls, const,)	0637
data declaration	03955
(lnk5er) (nls, const,)	0729
data declaration	03847
(lnk6er) (nls, const,)	0721
data declaration	03839
(lnk7er) (nls, const,)	0651
data declaration	03769
(lnk9er) (nls, const,)	0567
data declaration	03685
(lnkbdy) (nls, adrmnp, lnkbdy) 1I	02071
(adstr)	05508
find the body of a link	05507
(lnkcom) (nls, adrmnp, lnkcom) 1H	02081
(adstr)	05528
find the optional comment in a link	05527
(lnkdae) (nls, adrmnp, lnkdae) 1M	02085
(adstr, stp)	05535
find the optional dae in a link	05534
(lnkds1) (nls, const,)	0619
data declaration	03737
(lnkend) (nls, adrmnp, lnkend) 1P	02094
(adstr)	05549
find the right delimiter of a link	05548
(lnkfil) (nls, adrmnp, lnkfil) 1N	02095
(adstr, stp)	05551
find the optional filename in a link	05550
(lnkfp) (nls, adrmnp, lnkfp) 1J	02070
(adstr, tpl)	05506
find the optional filspc of a link	05505
(lnkhst) (nls, adrmnp, lnkhst) 1K	02080
(adstr, stp)	05526
find the optional hostname in a link	05525
(lnkprs) (nls, adrmnp, lnkprs) 1F	02069
(stp, adstr)	05504
detailed link parser	05503
(lnkpspc) (nls, adrmnp, lnkpspc) 1C	02093
(lnknum, tptadr, usrstg, frmstg, addexpstg, vspstg, adstr)	05547
extract fields from link and setup link data block	05546
(lnkstr) (nls, adrmnp, lnkstr) 1G	02077
(stp, adstr)	05520

find the start of a link	05519
(lnkusr) (nls,adrmno,lnkusr) 1L	02074
(adstr,stp)	05514
find the optional username in a link	05513
(lnkvws) (nls,adrmno,lnkvws) 10	02087
(adstr,stp)	05539
find the optional vwspc in a link	05536
(localdef) (nls,nddt,localdef) 11E2A	02429
(rd)	06033
(lockfile) (nls,filmp,lockfile) 2L5	01815
(fileno)	05073
(lockid) (nls,filmp,lockid) 2L7	01838
(astr,dirnum,initials)	05117
append "Being Modified By dirnumnam (init)" to astr	05116
(lockjo) (nls,csendmail,lockjo) 9E	02669
(type)	06453
Type = 1 for bad file lock (flag 1), and 0 for jlock (flag 0);	06452
(lockmes) (nls,icexec,lockmes) 12A	02161
(fileno)	05670
Type out a message indicating that the file is locked, and by whom	
(lockpage) (nls,auxcod,lockpage) 12B	05669
(core)	02301
lock the data page	05899
(lodkey) (nls,colart,lodkey) 7D	05898
(n,stad)	02350
(lodrfb) (nls,filmp,lodrfb) 2B1	05953
(nblk,btype,fileno)	01818
Load random file block. Arguments are	05079
(lodrng) (nls,filmp,lodrng) 2C1	05078
(stad)	01846
This routine loads a ring block into core. The STID is provided as	05135
argument. Returns two results:	05134
(lodsdb) (nls,filmp,lodsdb) 2I1	01801
(stdb)	05047
This routine loads an sdb into core. The STDB is provided as	
argument. Returns	05046
(longop) (nls,nddt,)	02380
data declaration	05965
(lookansw) (nls,psupport,lookansw) 3E1	0259
(resultptr,parsemode,stringptr)	03167
ptr to help string	03166
(lookbug) (nls,psupport,lookbug) 3H2	0239
(curptr,parsemode,string)	03126
lookbug looks at the next inpt character, if it is a CA, then a	
true return is taken else FALSE is returned	03127
(lookc) (nls,inpfbk,lookc) 3C	0542
look at a character	03653
(lookconfirm) (nls,psupport,lookconfirm) 3H1	0242
(curptr,parsemode,string)	03134
lookconfirm looks at the next inpt character, if it is a	
CA/REPEAT/INSERT, then a true return is taken else FALSE is returned	03133
(looknum) (nls,psupport,looknum) 3J1	0253
(curptr,parsemode,string)	03155
looknum looks at the next inpt character, if it is a digit, then a	

data declaration	03717
(lscopy) (nls,dspgen,lscopy) 3B9	01896
(option,da)	05223
copy LSRT entries	05222
(lsgfmt) (nls,dspgen,lsgfmt) 3B2	01877
(da)	05191
line segment formatter	05190
(lshed) (nls,records,)	01273
data declaration	04391
(lsmove) (nls,dspgen,lsmove) 3B10	01897
move line segment entry from &art to &rt, leaving only &art updated	
	05224
(lsrtl) (nls,records,)	01096
data declaration	04214
(lstidnxt) (nls,dspgen,lstidnxt) 3B8	01905
(start)	05238
next stid's LSRT address	05237
(lstr) (nls,const,)	0660
data declaration	03778
(ltl) (nls,inpfbk,ltl) 9D2	0501
set LT viewspec area large	03595
(lts) (nls,inpfbk,lts) 9D3	0499
set LT viewspec area small	03592
(luser) (nls,cident,luser) 6B	02603
(entrystr,fdstr,ptr1,ptr2)	06291
get LOGICAL USER-NAME (for NLS delivery) for an ident entry	06290
(m) (nls,parser,)	0165
data declaration	03025
(macfil) (nls,const,)	0768
data declaration	03886
(makemf) (nls,inthls,makemf) 6F	02816
create the message fork	06638
(makepc) (nls,filmap,makepc) 2L8	01832
(fl,fileno,astr)	05105
(makeptr) (nls,utility,makeptr) 9A	01604
(stid,ptr)	04725
Make a pointer pointing to the first character of stid, and store it	
into ptr	04724
(makezone) (nls,stgmt,makezone) 3C	02217
(zone,max,min,size)	05777
make a zone out of a block	05776
(makgid) (nls,cident,makgid) 13G	02559
(namestr,idstr)	06208
generate an ident from group name string	06207
(maklsrt) (nls,stgmt,maklsrt) 4B1	02216
(zone,segs,da)	05775
allocate a new block in "zone" to be used as a line segment	
reference table	05774
(mantic) (nls,const,)	0626
data declaration	03744
(marker) (nls,const,)	0827
data declaration	03945
(markit) (nls,inpfbk,markit) 9A1	0505
(da,xcrd,ycrd)	03602
(markpt) (nls,parser,)	0116
data declaration	02971

(marksym) (nls,usrpgrm,marksym) 18	02725
(blocknm)	06519
marks DDT's symbol table	06518
(maxsel) (nls,parser,)	0204
data declaration	03069
(maywrt) (nls,ioexec,maywrt) 13E	02148
(fileno,fl)	05647
Uses chkac JSYS to see if user has access to the file whose file status table entry address or file number is passed.	05646
(mdeman) (nls,const,)	0749
data declaration	03867
(menchr) (nls,help,)	02786
data declaration	06603
(mencont) (nls,help,mencont) 1E11	02750
(menumax,menusize,menusw,destsw,oldview)	06563
(mentyp) (nls,help,)	02766
data declaration	06574
(mergbranch) (nls.colstr,mergbranch) 6F	02325
(bug1,bug2)	05925
(merge) (nls.colstr,merge) 8A	02333
(stidx,inlst,kpr,percnt,copyf)	05931
(mergrp) (nls.colstr,mergrp) 6G	02311
(bug1,bug2,bug3,bug4)	05910
(mergplx) (nls.colstr,mergplx) 6H	02345
(bug1,bug2)	05946
(merorder) (nls.colstr,merorder) 8B	02319
(n)	05919
(mesfre) (nls.frontend,mesfre) 4A	02023
(fl,ptr)	05439
(mexper) (nls,const,)	0671
data declaration	03789
(mfixed) (nls,const,)	0866
data declaration	03984
(mkbptr) (nls.utility,mkbptr) 8X	01673
(pos.size,addr)	04806
Make Byte Pointer with the specified position, size and address. The address is 23 bits wide to allow indexing and indirection.	04805
(mkcnt) (nls.records,)	01460
data declaration	04578
(mkdirlist) (nls.execfl,mkdirlist) 4E	01955
(fjfn,pcjfn,info,grpok,sortk,adlmb)	05334
make an entry for a directory list	05333
(mkexis) (nls.records,)	01125
data declaration	04243
(mkfix) (nls.records,)	01383
data declaration	04501
(mkname) (nls.records,)	01307
data declaration	04425
(mkosid) (nls.records,)	01540
data declaration	04658
(mkrl) (nls.records,)	01062
data declaration	04180
(mkrset) (nls.dspgen,mkrset) 3D2	01915
col # is da.daccol	05256
(mkrsho) (nls.dspgen,mkrsho) 3D1	01893
(mlf) (nls.inpfbk,mlf) 9G1D	0547

(astrng)		03662
Move literal to Feedback line		03661
(mlfl) (nls,inpfbk,mfl) 9G1C		0523
(astrng)		03628
move a-string to 1st pos of CFL		03627
(mlfr) (nls,inpfbk,mifr) 9G1E		0511
(astrng)		03611
replace last literal in CFL		03610
(mnstd) (nls,cheap,)		02789
data declaration		06606
(modidfil) (nls,cident,modidfil) 12B		02642
(idstr,infostr,fileno,doit)		06365
modify old record in the master ident-file		06364
(modrefs) (nls,cident,modrefs) 12E		02624
(oidstr,nidstr,entrystr,fileno)		06331
change all references to old-ident to new-ident		06330
(momrec) (nls,nddt,)		02420
data declaration		06024
(moreme) (nls,cheap,)		02740
data declaration		06542
(movbdry) (nls,frontend,movbdry) 11D		01966
(dal,ocords,ncords,type)		05353
move window boundary		05352
(movecnum) (nls,catnum,movecnum) 2E		02514
(brname,numbsr,identsr,typesr,stid)		06136
Move designated number under designated list (branch name). If the		
list doesn't exist, create it		06135
(movflandpc) (nls,execfl,movflandpc) 3L		01941
(jfn1,pcjfn1,jfn2,pcjfn2,errstring)		05307
do the move file stuff		05306
(movsym) (nls,inhls,movsym) 6C		02850
location 116B contains -ln,,slcc where ln is length of symbol table		
and slcc is lowest location used in symbol table. This procedure		
moves the symbol table so that highest loc used is 377777B and		
updates 116B appropriately. Symbols can then be defined at runtime		
using the space between new slcc and end of program in low segment.		
		06676
(movtst) (nls,cedit,movtst) 17C		01696
(stid,grp1,grp2)		04849
Given an stid as the first argument, this routine makes sure that		
the stid is not in the group bounded by the second and third		
arguments passed it. If it is, it does an err.		04848
(mrk) (nls,inpfbk,mrk) 9G1B		0491
put CFL prompt in next position		03582
(mrk1) (nls,inpfbk,mrk1) 9G1A		0538
put CFL up arrow in first pos		03649
(mslclk) (nls,records,)		01444
data declaration		04562
(msavpf) (nls,const,)		0707
data declaration		03825
(msavqn) (nls,const,)		0781
data declaration		03899
(msavrp) (nls,const,)		0859
data declaration		03977
(msavws) (nls,const,)		0969
data declaration		04087

(msbavp) (nls,const,)	0809
data declaration	03927
(msbegi) (nls,records,)	01230
data declaration	04348
(msbins) (nls,records,)	01437
data declaration	04555
(msbk11) (nls,const,)	0891
data declaration	04009
(msbk21) (nls,const,)	0618
data declaration	03736
(msbufl) (nls,const,)	0881
data declaration	03999
(mschrt) (nls,const,)	0754
data declaration	03872
(mscoun) (nls,const,)	0706
data declaration	03824
(mscurr) (nls,records,)	01054
data declaration	04172
(mscirt) (nls,const,)	0684
data declaration	03802
(msdsys) (nls,const,)	0965
data declaration	04083
(mseins) (nls,records,)	01499
data declaration	04617
(msend) (nls,records,)	01250
data declaration	04368
(msent1) (nls,records,)	01457
data declaration	04575
(msfini) (nls,const,)	0915
data declaration	04033
(msgpsi) (nls,auxcod,msgpsi) 3E	02277
pseudo interrupt from the timer fork	05665
(mshdrl) (nls,const,)	0885
data declaration	04003
(msmax) (nls,records,)	01446
data declaration	04564
(msomto) (nls,const,)	0620
data declaration	03738
(msproc) (nls,const,)	0801
data declaration	03919
(msreal) (nls,const,)	0623
data declaration	03741
(msrecl) (nls,const,)	0702
data declaration	03820
(mssmps) (nls,const,)	0864
data declaration	03982
(msstar) (nls,const,)	0784
data declaration	03902
(msstre) (nls,const,)	01030
data declaration	04148
(mssys) (nls,const,)	0910
data declaration	04028
(mstclk) (nls,records,)	01276
data declaration	04394
(mstr) (nls,const,)	01013
data declaration	04131

(mstsys) (nls,const,)		0932
data declaration		04050
(mstxte) (nls,const,)		0962
data declaration		04080
(mswsiw) (nls,const,)		0592
data declaration		03710
(multch) (nls,const,)		0640
data declaration		03758
(multif) (nls,help,)		02736
data declaration		06536
(multyp) (nls,help,)		02769
data declaration		06576
(mvbfbf) (nls,utility,mvbfbf) 10F		01566
(bfrom,bto,nw)		04665
This routine moves a buffer of words. Arguments are address of source, address of destination, and number of words to transfer. It returns the address of the last word into which data was moved.		04684
(myrtn) (nls,resolver,myrtn) 1F		0481
for returns		03565
(n4ofil) (nls,const,)		0977
data declaration		04095
(name) (nls,const,)		0732
data declaration		03850
(namelook) (nls,adrmnp,namelook) 1R8A		02076
(stid,astr)		05518
This routine accepts an stid and astring, and finds the named statement in the file indicated by the stid, returning the stid of the statement or endfil. Uses non-sequential lookup. Copies name astring into local string so a literal may be used in the call		05517
(namesearch) (nls,cident,namesearch) 13B		02591
(string,idstr,type,fileno)		06270
search and print routine		06269
(namety) (nls,const,)		0714
data declaration		03832
(namfil) (nls,pdata,)		084
data declaration		02945
(namfst) (nls,help,namfst) 1D7		02760
(testr,conad,stid,index)		06566
a name is first list item. Look through menu for one context stack. Look through all context stacks. Look through data base sequentially from the top.		06565
(namingrp) (nls,adrmnp,namingrp) 1R8E		02068
(stid,lbstid,namstr,levels)		05502
lookup name in file starting at stid and terminating when through with the branch headed by lbstid, looking down levels levels.		05501
(nddtarm) (nls,nddt,nddtarm) 6B		02406
called from NLS		06009
(nddt disarm) (nls,nddt,nddt disarm) 6C		02411
called from NLS		06015
(ndr) (nls,select,ndr) 1C4		0232
(bug,ptr1,ptr2)		03115
verify number		03114
(needco) (nls,pdata,)		049
data declaration		02910
(neighb) (nls,help,neighb) 1D10		02800
(testr,stid,levent)		06623

Neighbor search for a name. The search will end when the number of levels down specified by levcnt from stid has been searched. If stid is the origin the whole file (down to levcnt) will be searched. If tid is a statement other than the origin the search will be confined to that branch.

(net) (nls,intnls,net) 4F	06622
Start NLS from the Net	02819
(netdsp) (nls,intnls,netdsp) 4G	06642
Dispatch to HIOP or NJS	02853
(nettty) (nls,const,)	06679
data declaration	01027
(netype) (nls,const,)	04145
data declaration	01008
(newda) (nls,frontend,newda) 110	04126
allocate new window	01981
(newfil) (nls,const,)	05375
data declaration	0733
(newfrring) (nls,frontend,newfrring) 12A	03851
(size)	01961
allocate a new file return ring	05346
(newintdspda) (nls,intnls,newintdspda) 6J	05345
(tat,tap,tal,tar)	02858
set up display areas for display terminal (relative to position of text area)	06688
(newnam) (nls,credit,)	01760
data declaration	04970
(newpag) (nls,tsprt,newpag) 3D	01924
(da)	05273
page eject	05272
(newrec) (nls,psident,newrec) 7	0370
(resultptr,parsemode)	03388
initialize current record, for adding new records to ident-file	03387
(newrng) (nls,filmap,newrng) 203	01835
(fileno)	05111
find room for a new ring element and allocate it. Called with file number of file where want the new element. Returns	05110
(newsdb) (nls,filmap,newsdb) 214	01850
(room,fileno)	05139
get a new SDB. Arguments are	05138
(newsrring) (nls,frontend,newsrring) 12G	01968
(size)	05356
allocate a new statement return ring	05355
(newstk) (nls,cheap,)	02787
data declaration	06604
(newsut) (nls,filmap,newsut) 2F2	01851
(stid)	05141
This routine gets a new stid and then inserts it as the sub of the stid passed it. It returns the new stid.	05140
(newsuc) (nls,filmap,newsuc) 2F1	01808
(stid)	05061
This routine gets a new stid and then inserts it as the suc of the stid passed it. It returns the new stid.	05060
(newver) (nls,credit,)	01702
data declaration	04859
(nextkey) (nls,parser,nextkey) 6B2	0126
(work,nextchar)	02982

(nextx) (nls,parser,)	0180
data declaration	03050
(nfield) (nls,nddt,)	02371
data declaration	05974
(nfloat) (nls,ccalc,nfloat) 5B	02546
(instrng,fl1,fl2)	06182
convert char - dp	06181
(nfstid) (nls,adrmap,nfstid) 1R7A	02075
(fnmstng,stnstg,vspstg,da)	05516
set t-ptr to new file and address expression	05515
(niltyp) (nls,const,)	0942
data declaration	04060
(ninjsys) (nls,nddt,ninjsys) 9C	02479
(inppbp,base)	06081
(njs) (nls,csendmail,njs) 11A	02691
(acs)	06457
network journal submission	06456
(njsverids) (nls,csendmail,njsverids) 11B	02662
(clerk,authlist,distlist,errlst)	06401
verify ident lists	05400
(nlcrms) (nls,inpfbk,nlcrms) 3E	0508
(ndofcom)	03606
Issue NLSOR JSYS	03605
(nls) (nls,intnls,nls) 5A4	02852
(nlsave) (nls,intnls,nlsave) 6A	02842
save nls as disk file	06666
(nlsemi) (nlslan,nlsemi) **Couldn't open <NLS>nlslan<↑>NLS**	095
(nlsext) (nls,const,)	0744
data declaration	03862
(nlrst) (nls,auxcod,nlrst) 4D	02264
RESET NLS to recover from catastrophic problems	05844
(nlssub) (nls,pdata,)	064
data declaration	02925
(nlsutilty) (nls,recfil,nlsutilty) 3	02521
(modeflg)	06147
Do utility type of things for NLS	06146
(nmdlbset) (nls,auxcod,nmdlbset) 10C	02292
(stidl,dleft,dright,da)	05887
set name delimiters for a branch	05886
(nmdlgset) (nls,auxcod,nmdlgset) 10E	02287
(stidl,stad2,dleft,dright,da)	05880
set name delimiters for a group	05879
(nmdlpset) (nls,auxcod,nmdlpset) 10D	02289
(stidl,dleft,dright,da)	05883
set name delimiters for a plex	05882
(nmdlsset) (nls,auxcod,nmdlsset) 10A	02293
(stid,dleft,dright)	05889
set name delimiters for a statement	05888
(nmdr) (nls,select,nmdr) 106	0234
(bug,ptr1,ptr2)	03119
statement name (and statement number) delimiter	03118
(nofail) (nls,pdata,)	068
data declaration	02929
(noident) (nls,psident,noident) 13	0384
(resultptr,parsemode)	03416
is there no ident assigned to current record	03415

(noleva) (nls,const,)	0610
data declaration	03728
(noprom) (nls,const,)	0887
data declaration	04005
(notca) (nls,psupport,notca) 3F1	0248
(curbtr,parsemode,string)	03146
notca looks at the next inpt character, if it is not a CA, then the character is read and a true return is taken. If the next character is a CA, then it is not read, and FALSE is returned	03145
(notrapcc) (nls,utility,notrapcc) 12B	01623
(notuse) (nls,records,)	01261
data declaration	04379
(notyet) (nls,credit,)	01710
data declaration	04873
(novspe) (nls,const,)	0903
data declaration	04021
(npcode) (nls,const,)	0916
data declaration	04034
(npsize) (nls,const,)	0745
data declaration	03863
(npstrad) (nls,dspgen,npstrad) 3B6	01894
(npchar)	05219
get symbol for non-printing	05218
(npstri) (nls,const,)	0966
data declaration	04104
(nselec) (nls,parser,)	0115
data declaration	02970
(nstr) (nls,const,)	0812
data declaration	03930
(nsucce) (nls,parser,)	0114
data declaration	02969
(nttyp) (nls,const,)	0725
data declaration	03843
(nulich) (nls,utility,)	01616
data declaration	04739
(nullfi) (nls,cident,)	02595
data declaration	06276
(nulstr) (nls,const,)	0978
data declaration	04096
(numbas) (nls,const,)	0657
data declaration	03775
(number) (nls,const,)	01009
data declaration	04127
(numbrv) (nls,const,)	0929
data declaration	04047
(numfst) (nls,help,numfst) 1D5	02751
(conad,stid,number,index)	06554
The first item in the list is a menu number. Count down through menu stack "n" items. It is an error if the number is too high	06553
(numoff) (nls,const,)	0741
data declaration	03659
(numsym) (nls,nddt,)	02472
data declaration	06075
(numtype) (nls,cathum,numtype) 2F	02513
(stid,numstr,typesr)	06134
Compares name of branch containing numstr to string typesr, and	

```

returns true if same, false if not. Returns stid of number
statement as second result                                06133
(nvtk) (nls,const,.)                                   0777
  data declaration                                       03895
(nwada) (nls,const,.)                                   0637
  data declaration                                       03755
(nwidst) (nls,cident,.)                                 02592
  data declaration                                       06271
(nwlnk) (nls,pdata,.)                                   045
  data declaration                                       02906
(nwrngb) (nls,filemp,nwrngb) 204                       01791
  (fileh.rngblk,pgindx,fileno)                          05027
  Used by newrng to actually allocate the ring element. Generates new
  SIP and stores it in the ring element. Called with    05026
(nwstrd) (nls,const,.)                                   0752
  data declaration                                       03870
(nxckpt) (nls,records,.)                                 01147
  data declaration                                       04265
(nxtbit) (nls,utility,nxtbit) 1401                     01651
  (table.bit,length)                                     04777
(nxtlsg) (nls,dsplen,nxtlsg) 3B3                       01909
  (da)                                                   05246
  line segment termination                               05245
(nxtnam) (nls,const,.)                                   0562
  data declaration                                       03680
(octnum) (nls,nddt,octnum) 7C                           02470
  (value)                                               06073
(odtyp) (nls,const,.)                                   0751
  data declaration                                       03869
(offlin) (nls,const,.)                                   0633
  data declaration                                       03751
(ofiler) (nls,const,.)                                   0972
  data declaration                                       04090
(oflo) (nls,ccalc,.)                                    02534
  data declaration                                       06163
(ofstid) (nls,adrmp,ofstid) 1R7B                        02078
  (dps.stnstrg)                                         05522
  evaluate intra-file address expression                 05521
(ojdelf) (nls,ioexec,.)                                 02135
  data declaration                                       05623
(ojtiii) (nls,records,.)                                01088
  data declaration                                       04206
(ojtijj) (nls,records,.)                                01254
  data declaration                                       04372
(ojtinn) (nls,records,.)                                01068
  data declaration                                       04186
(ojtitt) (nls,records,.)                                01498
  data declaration                                       04616
(okident) (nls,psident,okident) 16                     0368
  (resultptr,parsemode,identstr)                       03384
  is the passed ident ok in syntax and uniqueness       03383
(oldanswer) (nls,prmspc,oldanswer) 3L                   02043
  this procedure accepts a yes or no answer from the
  keyboard, and returns with a 0 if the answer was negative, and a 1
  if it was positive                                     05464
(oldfil) (nls,const,.)                                   0886

```

data declaration	04004
(oldid) (nls,cident,oldid) 3D	02607
(identstr,fileno)	06299
check if an ident is already being used	06298
(oldint) (nls,utilty,)	01652
data declaration	04778
(oldpgs) (nls,const,)	0867
data declaration	03985
(oldver) (nls,credit,)	01758
data declaration	04967
(oldvrs) (nls,const,)	0953
data declaration	04071
(onecha) (nls,const,)	0889
data declaration	04007
(opcmdv) (nls,const,)	0926
data declaration	04044
(opcod1) (nls,ndat,)	02415
data declaration	06020
(opcode) (nls,parser,)	0161
data declaration	03021
(opdrdv) (nls,const,)	0615
data declaration	03733
(open) (nls,ioexec,open) 4A	02189
(jfn,astrng)	05723
Open file routine. If the value of jfn passed to this procedure is zero, it tries to get a jfn (using short get jfn with the file name contained in the string whose address is passed as the second parameter.) If it cannot get the jfn, err is called. The full file name is put into astrng. If not already open, the original file whose jfn is passed or has been obtained is opened along with its partial copy (if it exists); the file is entered in the file status table and the NLS file number returned. If the file is already open, the old file number is returned. If the file cannot be successfully opened, the file is removed from the file status table and err is called	05722
(openall) (nls,ioexec,openall) 4G	02117
Open all files in file status table. Returns FALSE if cannot open one of the files in the file list-- used when user continues from EXEC after quit (files were closed but not deleted from filst)	05591
(openid) (nls,ioexec,openid) 6A	02203
open initial file; constructs file name from login directory name and current initials. creates the file if it does not exist. Returns file number.	05750
(openlock) (nls,ioexec,openlock) 4E	02191
(jfn,astr)	05727
Open and lock file indicated by astr after waiting for it to be free. (Used primarily for systems files in Journal, Ident systems, etc.) (Uses OPLCK to set and reset openlock lock flag-- this is the only flag used to channel locks by everyone using the journal, ident, and measurement systems(!!!)) Expunges directory in effort to try to get file opened. Also does some other wierd things. This is a very poorly coded procedure which is used in many important subsystems. Returns stid of origin statement of file. SIGNALS or calls err if something wrong.	05726
(openpc) (nls,ioexec,openpc) 4K	02172
(fileno,astr,acc)	05691

Opens the PC for the file in fileno if it exists with access specified in acc (if possible). Returns True if file does not have PC or PC opened ok. Returns False if PC open fail, astr contains message. 05690

(openseq) (nls,seqgen,openseq) 3B 01864
(fstid,lbstid,vspec,vspc2,usqcod,cacode) 05165
open a sequence 05164

(openswork) (nls,utilty,openswork) 7B 01635

(openull) (nls,ioexec,openull) 6B 02108
(nfname) 05576
create file with given name. Returns file number. (Almost identical with opendir except for name creation and flags sent to sgt,jfn; they should be combined (add flag parameter).) 05575

(openwk) (nls,ioexec,openwk) 4B 02182
(jfn,astr) 05711
Opens and resets work file specified by astr: creates it with a default extension of .NLS if it does not exist. It returns the stid of the origin statement of the file 05710

(oper) (nls,nddt,oper) 10A6 02486
(value,value1,op) 06089

(opform) (nls,records,) 01484
data declaration 04602

(opinit) (nls,seqfil,opinit) 60 02238
(da,jfn,devtyp,opflags) 05812

(oplockfail) (nls,ioexec,oplockfail) 4E8C1E3C3 02120

(opmcfil) (nls,const,) 01004
data declaration 04122

(opmlfil) (nls,const,) 0608
data declaration 03726

(opname) (nls,const,) 0602
data declaration 03720

(opnctfile) (nls,ioexec,opnctfile) 4F 02162
(initstr,userstr) 05672
Open initial file (in any directory) specified by the strings whose addresses are passed in initstr and userstr. CALLED ONLY IN (JNLDEL, OLDDIST). Return stid of origin if opened, and FALSE if not. In addition to opening it, it also locks it by creating a partial copy. Enables to permit the operator through this procedure to get to the initial file of any user. Calls rawopen. 05671

(opnfil) (nls,ioexec,opnfil) 4H 02113
(fileno,astring) 05584
Primitive open file routine-- read/write access. The original file and the partial copy for the given file number are opened by this routine. Does not allow pc to be opened for read unless sysopen fails on pc, (disk allocation exceeded, for example). Returns FALSE if original file cannot be opened. See coropnfil. 05583

(opninf) (nls,colsrt,opninf) 20 02338
l--open next input file in list 05939

(opprdv) (nls,const,) 0815
data declaration 03933

(oppqfil) (nls,const,) 0601
data declaration 03719

(oprchr) (nls,seqfil,oprchr) 6F 02247
get a character for OP 05825

(oprgps) (nls,seqfil,oprgps) 6I 02246

(oprhdf) (nls,seqfil,oprhdf) 6N 02236

(oprlev) (nls,seqfil,oprlev) 6K	02233
(oprmdv) (nls,const,)	0816
data declaration	03934
(oprnst) (nls,seqfil,oprnst) 6G	02239
get next statement for OP	05813
(oprrst) (nls,seqfil,oprrst) 6J	02229
(pos)	05799
set to passed char position	05798
(oprsig) (nls,seqfil,oprsig) 6M	02260
(oprsvc) (nls,seqfil,oprsvc) 6L	02250
(oprtxt) (nls,seqfil,oprtxt) 6H	02228
get char pos of 1st text after "name"	05797
(opseqf) (nls,seqfil,opseqf) 3E	02245
(ofilnam,da,upflg,pgflg,typfil)	05824
type of file	05823
(opsimr) (nls,records,)	01471
data declaration	04589
(opsgdv) (nls,const,)	0621
data declaration	03739
(opsqfl) (nls,const,)	0669
data declaration	03787
(opstmt) (nls,colstr,opstmt) 17	02318
(stid,short)	05918
copy next input statemnt to file	05917
(optact) (nls,pdata,)	017
data declaration	02878
(optcha) (nls,pdata,)	051
data declaration	02912
(optcou) (nls,parser,)	0101
data declaration	02951
(optfla) (nls,parser,)	0211
data declaration	03079
(optydv) (nls,const,)	0786
data declaration	03904
(opwk) (nls,ioexec,opwk) 4D	02159
(jfn,astr)	05667
The same as opwknr, but permits a file to already be open!	
(accomplished through code which exists in open. Should be combined	
with opwknr with parameter added to check if this code need be	
executed.)	05666
(opwknr) (nls,ioexec,opwknr) 4C	02139
(jfn,astr)	05630
Opens (or creates) without resetting work file specified in astr.	
Returns the NLS fileno. The file cannot be already open! (opwk	
handles this case; the two should be combined.)	05629
(opwtpb) (nls,records,)	01456
data declaration	04574
(opxtdv) (nls,const,)	0775
data declaration	03893
(orgrptst) (nls,cident,orgrptst) 9C	02585
(string,dstptr)	06258
test if a group or organization entry	06257
(orgs) (nls,cident,)	02615
data declaration	06314
(orgsti) (nls,const,)	0614
data declaration	03732

(orgtst) (nls,cident,orgtst) 95	02558
(string,dstptr)	06206
test if a organization entry	06205
(orgtyp) (nls,const,)	0967
data declaration	04085
(origfil) (nls,const,)	0789
data declaration	03907
(origin) (nls,utility,)	01663
data declaration	04792
(ostr) (nls,const,)	0904
data declaration	04022
(outasm) (nls,seqfil,outasm) 3D	02240
(ofilnam,da,forceup)	05815
does the output to Assembler File command	05814
(outjm) (nls,seqfil,outjm) 3C	02230
(str,da)	05801
does Output Journal Mail command	05800
(outptr) (nls,recfil,outptr) 7	02518
(fjmstr,da,copies,lpt)	06141
(outqp) (nls,seqfil,outqp) 3B	02242
(ofilnam,da)	05819
does the output quickprint command	05818
(outrem) (nls,const,)	01010
data declaration	04128
(outseq) (nls,seqfil,outseq) 3A	02232
(ofilnam,da,forceup)	05805
does the output Sequential File command	05804
(oversc) (nls,chelp,)	02792
data declaration	06610
(p) (nls,parser,)	0170
data declaration	03030
(pad) (nls,dspgen,pad) 7B	01895
(count)	05221
pad with count null characters	05220
(pager) (nls,seqfil,pager) 3C13	02234
(sw)	05807
FORMAL PARAMETERS	05806
(param) (nls,pdata,)	04
data declaration	02865
(param1) (nls,pdata,)	029
data declaration	02890
(param2) (nls,pdata,)	055
data declaration	02916
(param3) (nls,pdata,)	085
data declaration	02946
(param4) (nls,pdata,)	016
data declaration	02877
(parmdef) (nls,nddt,parmdef) 11B3A	02362
(rd)	05965
(parseh) (nls,const,)	0823
data declaration	03941
(parseinput) (nls,execfl,parseinput) 3A	01957
(fname,udirname,ufilename,uextname,uverno,ulitovr,uname,flags)	05338
parse input file link from user	05337
(parsenter) (nls,nddt,parsenter) 10A7	02369
this routine parses the current input stream looking for a value	

being typed in -	05972
(parsevalue) (nls,nddt,parsevalue) 10A8	02446
(rd,symtype,symval)	06048
(parsin) (nls,const,)	0872
data declaration	03990
(partpr) (nls,const,)	0783
data declaration	03901
(passwo) (nls,const,)	0766
data declaration	03884
(pathba) (nls,pdata,)	077
data declaration	02938
(pathst) (nls,pdata,)	034
data declaration	02895
(pathx) (nls,pdata,)	013
data declaration	02874
(pause) (nls,auxcod,pause) 4K	02285
(tim)	05877
Pause for specified number of msec	05876
(pb) (nls,pdata,)	041
data declaration	02902
(pbug) (nls,inpfbk,pbug) 9B1	0515
(ordinates)	03617
process a bug selection	03616
(pcomco) (nls,parser,)	0134
data declaration	02992
(pconle) (nls,const,)	0908
data declaration	04026
(pcp) (nls,utility,pcp) 9H3	01613
(pdc) (nls,utility,pdc) 9B	01561
(pgcnum,ptloc)	04680
pointer decrement. arguments are number of postions to	04679
(pdummy) (nls,nddt,)	02489
data declaration	06092
(pfunct) (nls,parser,)	0152
data declaration	03009
(pgsize) (nls,const,)	0570
data declaration	03688
(pjsep) (nls,seqgen,pjsep) 4C	01873
(sw,title)	05183
Print Journal Header/Trailer	05182
(pjseqg) (nls,seqgen,pjseqg) 4B	01857
(sw,which)	05152
Print Journal Seq Generator	05151
(pkeyle) (nls,const,)	0746
data declaration	03864
(pkstrp) (nls,parser,)	0159
data declaration	03019
(plex) (nls,const,)	0639
data declaration	03757
(plexv) (nls,const,)	0849
data declaration	03967
(plvcou) (nls,parser,)	0132
data declaration	02989
(plvlen) (nls,const,)	01024
data declaration	04142
(plxdr) (nls,select,plxdr) 1B2C	0233

(tptr1,ret1,ret2)	03117
address of the second result text pointer	03116
(plxset) (nls,select,plxset) 1D1	0216
(stid)	03091
Given a stid, this routine will return the stid's of the head and tail, respectively, of the plex in which the stid appears.	03090
(pmode) (nls,parser,)	0182
data declaration	03043
(pnext) (nls,const,)	0753
data declaration	03871
(popids) (nls,cident,popids) 5B	02580
(ptr)	06248
pop the ident pushdown stack	06247
(popsel) (nls,const,)	0958
data declaration	04076
(popsta) (nls,const,)	0680
data declaration	03798
(popsym) (nls,usrpgm,popsym) 17	02731
pops block of symbols from DDT's symbol table	06529
(port) (nls,pdata,)	018
data declaration	02879
(position) (nls,dspgen,position) 7C	01911
(xpos,ypos)	05250
position cursor - alphanumeric displays	05249
(ppaprt) (nls,utility,ppaprt) 4H1	01614
(pr) (nls,utility,pr) 13B	01572
(astr,val)	04694
print register values for illegal instruction trap	
pseudo-interrupts	04693
(praddr) (nls,records,)	01429
data declaration	04547
(prcadr) (nls,const,)	0728
data declaration	03846
(prcerr) (nls,const,)	0689
data declaration	03807
(prcmds) (nls,dspgen,prcmds) 4B	01903
(da)	05235
post processor	05234
(pred) (nls,const,)	0569
data declaration	03687
(preframe) (nls,nddt,preframe) 11B5E	02487
(rd)	06090
(prexis) (nls,records,)	01183
data declaration	04301
(prgetst) (nls,seqfil,prgetst) 6E	02253
returns byte pointer + count for next statement	05833
(printg) (nls,tsprt,printg) 3A	01921
(da,stid1,stid2,type,oldsw)	05267
print the structure (type) addresssed by stid1,stid2; oldsw is the address of an already open sequence work area or zero. If zero, a sequence work area is opened by printg and closed; if non-zero, the sequence is not closed by printg.	05266
(processor) (nls,seqfil,processor) 6A	02251
(prcnam,tp,da,type,outdsg,prcjfn)	05831
start and run a processor	05830
(proflandpc) (nls,execfl,proflandpc) 3K	01953

(fjfn,pcjfn,mask,prot,astr)	05330
set protection of a file (and its pc)	05329
(progsy) (nls,ndat,)	02407
data declaration	06010
(promps) (nls,pdata,)	022
data declaration	02883
(prompt) (nls,inpfbk,prompt) 6D	0502
(astr)	03597
Prompt user by typing astr	03596
(propco) (nls,parser,)	0173
data declaration	03033
(proprot) (nls,execfl,proprot) 3N	01948
(jfn1,jfn2)	05320
propagate protection from jfn1 to jfn2	05319
(prsave) (nls,pdata,)	037
data declaration	02898
(prsavx) (nls,pdata,)	036
data declaration	02897
(prtdir) (nls,const,)	0695
data declaration	03813
(prtype) (nls,tsprt,prtype) 3B	01923
(da,number)	05271
type 'number' characters from prbuf	05270
(prvsts) (nls,records,)	01073
data declaration	04191
(prwrđ1) (nls,records,)	01224
data declaration	04342
(prwrđ2) (nls,records,)	01528
data declaration	04646
(pselle) (nls,const,)	0678
data declaration	03796
(psi) (nls,utility,psi) 13A	01612
deal with illegal instruction pseudo-interrupt	04737
(pspriv) (nls,const,)	0822
data declaration	03940
(pspubl) (nls,const,)	0734
data declaration	03852
(pstr) (nls,const,)	01031
data declaration	04149
(ptradr) (nls,const,)	0731
data declaration	03849
(ptrstk) (nls,pdata,)	012
data declaration	02873
(ptrx) (nls,pdata,)	010
data declaration	02871
(ptrxsa) (nls,parser,)	0143
data declaration	03001
(pushdc) (nls,dspgen,pushdc) 4A	01880
(op,param1)	05196
enter commands into the display commands list	05195
(pushent) (nls,help,pushent) 1F1	02757
(std,type)	06562
Makes use of the global curstk as the context to be filled in	06561
(pushfrring) (nls,frontend,pushfrring) 12C	01960
(frr,filename,fileno)	05344
push file return ring	05343

(pushids) (nls,cident,pushids) 5C	02583
(ptr)	06254
push the ident pushdown stack	06253
(pushsrring) (nls,frontend,pushsrring) 12J	01980
(srr,stdio,cc,vs1,vs2)	05300
push contents onto statement return ring	05385
(putchr) (nls,tsprt,putchr) 3C	01922
(da,char,column,astrng)	05269
put char into an a-string or onto tty	05268
(putvspc) (nls,prmspc,putvspc) 9B	02060
(dpa)	05488
Given the address of a display area, this routine will put the viewspecs saved in sysvspec and sysvspec/1/ in the display area.	
	05487
(pvslen) (nls,const,)	0501
data declaration	03699
(qappender) (nls,chelp,qappender) 1E4	02795
(string,appendstr,front,end,truncate,dirflg,lines)	06610
Appends contents of appendstr to front of string; puts contents delimited by front and end at the end of string. (Does not try to append appendstr if NULL; does not try to append front through end if front is zero.) Does not reset original contents of the string.	
	06615
(qcadd) (nls,ccalc,qcadd) 3B	02542
(ar,ma)	06176
floating add ar + ma, result to ar	06175
(qcall) (nls,utility,qcall) 4H2	01610
(qcall1) (nls,utility,qcall1) 4H3	01659
(qcallm) (nls,utility,qcallm) 4H4	01647
(qcdiv) (nls,ccalc,qcdiv) 3C	02540
(ar,ma)	06172
call divide and replace accum with quo	06171
(qcdivw) (nls,ccalc,qcdivw) 3D	02540
(ar,ma,quo)	06186
divide ar by ma, result in quo	06185
(qcins) (nls,ccalc,qcins) 5F	02522
(cfloat,opstrng)	06149
insert value	06148
(qcmult) (nls,ccalc,qcmult) 3A	02549
(ar,ma)	06180
floating multiply ar * ma, result to ar	06107
(qcneg) (nls,ccalc,qcneg) 3F	02535
(ar)	06165
floating negate double precision of ar, result in ar	06164
(qcolumnate) (nls,chelp,qcolumnate) 1E13	02768
(sw,stdio,columnatestr,selection,flushflg)	06577
query columnation	06576
(qcolwi) (nls,chelp,)	02809
data declaration	06633
(qcsup) (nls,ccalc,qcsup) 3E	02545
(ar,ma)	06180
floating subtract ar - ma, result to ar	06179
(qda) (nls,chelp,)	02783
data declaration	06599
(qdavs2) (nls,chelp,)	02745
data declaration	06547

```

(qdavsp) (nls, chelp,) 02767
    data declaration 06575
(qdirparse) (nls, chelp, qdirparse) 1E5 02779
    (start, end) 06593
    parse query directives 06592
(qdisp) (nls, chelp, qdisp) 1B1 02759
    (stid, entry, contmen) 06564
(qfloat) (nls, ccalc, qfloat) 5A 02533
    (ar, ch) 06162
    qfloat ch (ascii char from 0-9) 06161
(qfloutp) (nls, ccalc, qfloutp) 5G 02541
    (ar, os, whence) 06174
    qfloating output a value at address ar to string at address os 06173
(qinclude) (nls, chelp, qinclude) 1E6 02805
    (sw, stringu@qlkleft, qlkright, topflag) 06630
    process query link list (includes) 06629
(qlnkspec) (nls, chelp, qlnkspec) 1E14 02798
    (tptadr, usrstg, fnmstg, stnstg, vspstg) 06620
    Makes use of the new lnkprs 06619
(qm) (nls, inpfbk, qm) 9G3A 0528
(qmenu) (nls, chelp, qmenu) 1E10 02777
    (stid, endstd, destsw) 06589
    output a query menu 06588
(qmenuc) (nls, chelp,) 02806
    data declaration 06631
(qmenum) (nls, chelp,) 02746
    data declaration 06548
(qmoff) (nls, inpfbk, qmoff) 9G3B 0520
    turn the '?' off. 03624
(qnewst) (nls, chelp,) 02749
    data declaration 06551
(qnum) (nls, chelp, qnum) 1D6 02739
    (stid, number) 06541
    A menu number found under a superior list item. Scan first level
    substructure under the current stid for n-1 successors. If invalid
    (i.e., getsuc encounters an 'up' before it reaches n-1 give error
    message, return TRUE or FALSE (if number invalid) and "current "
    location. 06540
(qpspec) (nls, chelp,) 02802
    data declaration 06626
(qsearch) (nls, chelp, qsearch) 1D1 02790
    (list, index) 06608
    Given the address of a string containing the node list of an item
    to be found and an index to the context ring element form which the
    search will start, search for the specified node. Returns the stid
    of the found node (or -1 if not found) and an index to a new context
    ring element (or the old index if the requested item is not found.)
    SRCHTYP specifies the search algorithm to be used if a context
    search (over the context nodes-- very fast) fails. 06607
(qsender) (nls, chelp, qsender) 1E3 02763
    (string, workarea, topflag) 06570
    sends the contents of string out to sequence workarea with level
    controlled by topflag. Resets contents of string after send. 06569
(qsparse) (nls, chelp, qsparse) 1E7 02781
    (qspstr, end, topflag) 06596
    set global queryspecs 06595

```

(qspcreset) (nls, chelp, qspcreset) 1E8	02793
(topflag)	06612
reset default queryspecs values	06611
(qspecs) (nls, chelp,)	02762
data declaration	06568
(qstmt) (nls, chelp, qstmt) 1E2	02735
(sw, prevsw, topflag, appendstr)	06535
process one QUERY source statement	06534
(qstorb) (nls, chelp,)	02742
data declaration	06544
(qstr) (nls, const,)	0941
data declaration	04059
(qstrinit) (nls, chelp, qstrinit) 1B4	02794
(entstd, topstd)	06614
Called to initialize the stack area as a storage zone for help/query. Makes use of stgmt routines to set up blocks in the display area block pool. Also responsible for initializing some global cells as well as the context ring	06613
(qstrip) (nls, chelp, qstrip) 1E9	02771
(stid, destsw, topflag, string)	06581
decrud a data base stmt	06580
(qsw) (nls, chelp,)	02743
data declaration	06545
(qt) (nls, intnls, qt) 5B	02815
(querco) (nls, chelp,)	02772
data declaration	06582
(querin) (nls, chelp,)	02784
data declaration	06600
(querop) (nls, chelp,)	02797
data declaration	06618
(querpr) (nls, chelp,)	02791
data declaration	06609
(query) (nls, intnls, query) 4D	02860
start Typewriter NLS	06688
(queryseq) (nls, chelp, queryseq) 1E1	02785
(sw, phase)	06602
query sequence generator program	06601
(qvalid) (nls, chelp, qvalid) 1D4	02782
(conad, index)	06598
Check if this stack has any entries; back up if not	06597
(r0) (nls, colsrt,)	02344
data declaration	05945
(r1) (nls, parser,)	0122
data declaration	02978
(r2) (nls, parser,)	0209
data declaration	03076
(r3) (nls, parser,)	0120
data declaration	02975
(r4) (nls, parser,)	0199
data declaration	03083
(r50toa) (nls, nddt2, r50toa) 1D3	02500
(value, astr)	06110
this procedure converts a RADIX50 symbol representation into a string	NLS 06109
(r6) (nls, nddt,)	02466
data declaration	06069

(r7) (nls,colstr,)		02328
data declaration		05927
(racc) (nls,const,)		0679
data declaration		03797
(raddr) (nls,nddt,)		02428
data declaration		06032
(random) (nls,const,)		0738
data declaration		03856
(rawopen) (nls,ioexec,rawopen) 4J		02207
(astr,killpc)		05757
Primitive open at any cost-- Open the file specified in astr,		
enter it into the file list. Open the partial copy whether the file		
is locked by another or not. Used only for special subsystem files		
(journal, etc.) Return fileno if successful, 0 otherwise		05756
(rbound) (nls,frontend,)		02019
data declaration		05433
(rcpsh) (nls,utility,rcpsh) 10B		01639
(recadr,st)		04764
Record push. Arguments are (1) address of record, and (2) address		
of stack or ring buffer. The record is pushed on the store.		04763
(rcpto) (nls,utility,rcpto) 10E		01662
(st,recadr)		04791
Record Pop To. Arguments are (1) address of store, and (2) address		
of record. Pops the top record on the store to the designated		
location.		04790
(rdef) (nls,nddt,)		02424
data declaration		06029
(rdhdr) (nls,ioexec,rdhdr) 15B		02163
(fileno,astrng)		05674
read file header		05673
(rdinit) (nls,nddt,)		02391
data declaration		05996
(rdlit) (nls,inpfbk,rdlit) 5C		0548
(astrng,chrst)		03664
read literal from keyboard		03663
(rdlutl) (nls,inpfbk,rdlutl) 5D		0546
(astrng,char,display)		03660
utility for rdlit		03659
(rdnum) (nls,nddt,rdnum) 7D		02441
(base)		06044
(rdproc) (nls,nddt,)		02361
data declaration		05964
(rdprvsts) (nls,ioexec,rdprvsts) 14E		02165
(fileno)		05678
read file's privacy status		05677
(rdsym) (nls,nddt,rdsym) 7E		02374
(astr)		05977
(rdummy) (nls,records,)		01386
data declaration		04504
(read) (nls,const,)		0804
data declaration		03922
(readbug) (nls,psupport,readbug) 3G2		0256
(curptr,parsemode,string)		03161
readbug looks at the next inpt character, if it is a CA, then it is		
read and a true return is taken else FALSE is returned		03160
(readc) (nls,utility,readc) 7E1		01637

(readconfirm) (nls,psupport,readconfirm) 3G1	0249
(curptr,parsemode,string)	03148
readconfirm looks at the next inpt character, if it is a CA/REPEAT/INSERT, then it is read and a true return is taken else FALSE is returned	03147
(readfrring) (nls,frontend,readfrring) 12D	02002
(frr,index)	05409
Read file return ring entry.	05408
(readoption) (nls,psupport,readoption) 3B1	0260
(resultptr,parsemode,string)	03169
ptr to help string	03168
(readrepeat) (nls,psupport,readrepeat) 3C1	0243
(resultptr,parsemode,string)	03136
ptr to help string	03135
(readsrring) (nls,frontend,readsrring) 12L	01965
(srr,index)	05351
Read statement return ring entry.	05350
(readwr) (nls,const,)	0629
data declaration	03747
(rechos) (nls,const,)	0628
data declaration	03746
(recog2) (nls,const,)	0717
data declaration	03835
(recogn) (nls,const,)	0670
data declaration	03788
(recpdef) (nls,nddt,recpdef) 11B4A	02394
(rd)	05999
(recprepl) (nls,nddt,recprepl) 11B4E	02482
(rd)	06085
(recpt1) (nls,nddt,recpt1) 11B4C	02409
(bp,rd)	06012
(recpt2) (nls,nddt,recpt2) 11B4D	02436
(bp,rd)	06040
(recred) (nls,frontend,recred) 5A	02007
(relext) (nls,const,)	0716
data declaration	03834
(reljfn) (nls,ioexec,reljfn) 8E	02119
(jfn)	05595
release the jfn	05594
(relsgw) (nls,seqgen,relsgw) 3L	01861
(sw)	05159
release a Sequence generator work area	05158
(remada) (nls,const,)	0564
data declaration	03682
(remdda) (nls,const,)	0701
data declaration	03819
(remfud) (nls,const,)	0980
data declaration	04098
(remgrp) (nls,filmp,remgrp) 2E2	01787
(grp1,grp2)	05020
Removes group whose bounds are passed it from position in ring but does not delete either ring elements or SDB's.	05019
(remini) (nls,const,)	0790
data declaration	03908
(remlcm) (nls,const,)	0608
data declaration	03926

(remrdd) (nls,const,)		0682
data declaration		03800
(remrsd) (nls,const,)		0811
data declaration		03929
(remscm) (nls,const,)		0898
data declaration		04016
(remscs) (nls,const,)		0876
data declaration		03994
(remstd) (nls,const,)		0979
data declaration		04097
(remssd) (nls,const,)		0757
data declaration		03875
(remstr) (nls,const,)		0691
data declaration		03809
(remtsf) (nls,const,)		0803
data declaration		03921
(remtsn) (nls,const,)		0711
data declaration		03829
(reentr) (nls,intnls,reentr) 4A		02823
use buffct as a temp variable		06646
(repchr) (nls,utility,repchr) 8K		01628
(char,ast,chrno)		04754
replace a character in an astring		04753
(repet) (nls,records,)		01291
data declaration		04409
(replac) (nls,nddt,)		02377
data declaration		05982
(replenish) (nls,stmgt,replenish) 3D		02210
(index,zone)		05763
replenish supply of free blocks in zone		05762
(replfield) (nls,nddt,replfield) 10A9A		02421
(rd)		06026
This procedure accepts an expression, and replaces the contents of the field indicated by rd with its value. It expects the expression to be terminated with a CA.		06025
(replrecord) (nls,nddt,replrecord) 10A9B		02445
(rd)		06047
Replace the record indicated by rd with a series of values::		06046
(replstring) (nls,nddt,replstring) 11B6A		02389
(rd)		05993
(repsig) (nls,utility,repsig) 4E		01622
(value)		04744
Repeat a signal with value as sysgn1		04743
(repstr) (nls,utility,repstr) 8U		01586
(newast,oldast,chrno)		04704
The routine replaces the characters in one a-string, OLDAST, with those in another, NEWAST, beginning at CHARNO. If the new a-string is empty, the routine returns FALSE; otherwise it returns TRUE.		04703
(repval) (nls,nddt,repval) 10A9C		02373
(rd)		05976
(rerror) (nls,auxgod,rerror) 4G		02303
(resda) (nls,dspgen,resda) 6C		01888
(da)		05210
restore display area		05209
(resetb) (nls,inpfbk,resetb) 10F		0545
(astrng)		03658

resets the character input buffer	03657
(resetf) (nls,ioexec,resetf) 6C	02196
(fileno)	05737
Reset file-- null partial file	05736
(reslev) (nls,seqgen,reslev) 4D	01874
(vspec,clevel)	05185
Given a viewspec word and a level, this routine will perform any relative level adjustment required by the viewspec word and return the updated viewspec word.	05184
(resolver) (nls,resolver,resolver) 1D	0484
(stp,stpst)	03570
upper bound for symbol table	03569
(respos) (nls,utility,respos) 7D	01636
(restbp) (nls,nddt,restbp) 10B7	02368
Restore the original instructions	05971
(retcc) (nls,records,)	01315
data declaration	04433
(retfil) (nls,pdata,)	038
data declaration	02899
(retfn) (nls,records,)	01293
data declaration	04411
(retsrr) (nls,records,)	01324
data declaration	04442
(retsti) (nls,records,)	01085
data declaration	04203
(return) (nls,const,)	0595
data declaration	03713
(retval) (nls,parser,)	0154
data declaration	03012
(retvs1) (nls,records,)	01331
data declaration	04449
(retvs2) (nls,records,)	01544
data declaration	04662
(rfcex) (nls,catnum,rfcex) 3A	02515
(authorstr,titlestr,sendstr,onlineflag,rfcnum,catnum)	06138
reserve an RFC (Request For Comments) and a Journal number given the currently available info contained in the string pointed to by STID.	
Reserve it on behave of the user whose ident is in authrsr	06137
(rfchos) (nls,const,)	0935
data declaration	04053
(rfcore) (nls,records,)	01467
data declaration	04585
(rfcsyscheck) (nls,catnum,rfcsyscheck) 3B	02509
see if the RFC system is enabled for this host	06126
(rfexis) (nls,records,)	01094
data declaration	04212
(rffree) (nls,records,)	01247
data declaration	04365
(rfnull) (nls,records,)	01515
data declaration	04633
(rfpart) (nls,records,)	01070
data declaration	04188
(rfpmax) (nls,const,)	0687
data declaration	03805
(rfused) (nls,records,)	01447
data declaration	04565

(rh) (nls,records,)	data declaration	01313
(rhf) (nls,records,)	data declaration	04431
(rhword) (nls,records,)	data declaration	01195
(rindex) (nls,nddt,)	data declaration	04313
(ringl) (nls,const,)	data declaration	01118
(rinist) (nls,const,)	data declaration	04236
(rinst1) (nls,records,)	data declaration	02418
(rinst2) (nls,records,)	data declaration	06023
(rjrepn) (nls,credit,rjrepn) 22E3	(ptr1,ptr2,aststring)	0841
(rjshift) (nls,credit,rjshift) 22E2	(ptr,deit)	03959
(rlevset) (nls,filmp,rlevset) 2F6	(stid,dir)	0919
	Determines target stid and direction for inserting statements. Given an stid and a relative levadj count, returns an stid and levdown if to be inserted down, levsuc if to be inserted as successor to returned stid.	04037
(rlookup) (nls,resolver,rlookup) 1E	(astr,bot,top)	01119
	upper bound of symbol table	04237
(rnamef) (nls,records,)	data declaration	01199
(rnameh) (nls,records,)	data declaration	04317
(rngbas) (nls,const,)	data declaration	01776
(rnglnt) (nls,help,)	data declaration	04999
(rngm) (nls,const,)	data declaration	01709
(rngmax) (nls,help,)	data declaration	04872
(rngtyp) (nls,const,)	data declaration	01811
(rnmcl) (nls,const,)	data declaration	05066
(rnull) (nls,records,)	data declaration	05065
(round) (nls,ccalc,)	data declaration	0483
(rp1lit) (nls,credit,rp1lit) 220	(grp1,grp2,t1,t2)	03568
	This routine replaces the group defined by the first two arguments passed it with a new statement containing the text in the astrng passed it. It returns the value of the new stid.	03567
(rp1pro) (nls,usrpgrm,)	data declaration	01231
		04349
		01296
		04414
		0855
		03973
		02776
		06587
		0598
		03716
		02744
		06546
		0968
		04086
		0990
		04108
		01454
		04572
		02543
		06177
		01691
		04839
		04838
		02734
		06533

(rpop) (nls,utility,rpop) 100	01605
(st)	04727
Record pop. Argument is the address of a store. Pop the top element.	04726
(rptcha) (nls,pdata,)	082
data declaration	02943
(rptpar) (nls,const,)	0624
data declaration	03742
(rsdb) (nls,records,)	01228
data declaration	04346
(rsid) (nls,records,)	01427
data declaration	04545
(rsize) (nls,nddt,)	02449
data declaration	06053
(rstconnection) (nls,auxcod,rstconnection) 5J	02271
break display screen connection	05855
(rstlit) (nls,inpfof,rstlit) 9H1	0506
reset the literal display area	03603
(rstr) (nls,const,)	0666
data declaration	03784
(rstsk) (nls,utility,rstsk) 10D	01642
(st)	04768
Reset store	04767
(rsub) (nls,records,)	01149
data declaration	04267
(rsuc) (nls,records,)	01292
data declaration	04410
(rtllsg) (nls,records,)	01312
data declaration	04430
(rtbpe) (nls,records,)	01355
data declaration	04473
(rtbps) (nls,records,)	01549
data declaration	04667
(rtcbug) (nls,records,)	01145
data declaration	04263
(rtccnt) (nls,records,)	01206
data declaration	04324
(rtcsiz) (nls,records,)	01077
data declaration	04195
(rtexis) (nls,records,)	01347
data declaration	04465
(rtf) (nls,records,)	01242
data declaration	04360
(rtfont) (nls,records,)	01251
data declaration	04369
(rthinc) (nls,records,)	01175
data declaration	04293
(rtlast) (nls,frontend,)	02020
data declaration	05434
(rtlev) (nls,records,)	01283
data declaration	04401
(rtlsid) (nls,records,)	01219
data declaration	04337
(rtmax) (nls,const,)	0788
data declaration	03906
(rtnew) (nls,records,)	01435

data declaration	04553
(rtbart) (nls,records,)	01552
data declaration	04670
(rtsrce) (nls,records,)	01182
data declaration	04300
(rtstid) (nls,records,)	01350
data declaration	04468
(rtxl) (nls,records,)	01433
data declaration	04551
(rtx2) (nls,records,)	01201
data declaration	04319
(rty) (nls,records,)	01144
data declaration	04262
(rubout) (nls,auxcod,rubout) 5A	02291
rubout interrupt routine	05885
(rubsig) (nls,auxcod,rubsig) 5B	02262
simply calls signal for rubout	05843
(runcmp) (nls,recfil,runcmp) 6	02517
(fmsstr,da)	06140
(runddt) (nls,ndat,runddt) 6D	02357
(breakadr)	05961
(rwthaw) (nls,const,)	01035
data declaration	04153
(s) (nls,parser,)	0189
data declaration	03051
(s1) (nls,pdata,)	03
data declaration	02864
(s2) (nls,pdata,)	048
data declaration	02909
(s3iftup) (nls,colsrt,s3iftup) 7B	02343
(i,n)	05944
sort tree into vector...from filcydd	05943
(savex) (nls,parser,)	0180
data declaration	03041
(savext) (nls,const,)	0851
data declaration	03969
(savhel) (nls,pdata,)	024
data declaration	02885
(savpos) (nls,utility,savpos) 7C	01668
(savstk) (nls,parser,)	0107
data declaration	02960
(savvspc) (nls,prmspc,savvspc) 9A	02064
(dpa)	05494
Given the address of a display area, this routine will save the viewspecs associated with the area in sysvspec and sysvspec/1/ (for massaging by setit).	05493
(sbas) (nls,records,)	01468
data declaration	04586
(sbc1cn) (nls,records,)	01124
data declaration	04242
(sbc2cn) (nls,records,)	01052
data declaration	04170
(sbc3cn) (nls,records,)	01360
data declaration	04478
(sbcoun) (nls,parser,)	0105
data declaration	02957

(sbdelck) (nls,credit,sbdelck) 28E	01692
(ch, ttype)	04841
check delimiter	04840
(sbdp) (nls,records,)	01050
data declaration	04168
(sbfini) (nls,const,)	0793
data declaration	03911
(sbinit) (nls,psedit,sbinit) 5B	0270
(sbnhed,sbrec,sbdsp,sbastr,sbttype)	03189
initialize substitute data structure	03188
(sbmode) (nls,parser,)	0118
data declaration	02973
(sbnptr) (nls,parser,)	0137
data declaration	02995
(sbpmod) (nls,parser,)	0151
data declaration	03008
(sbpop) (nls,const,)	0747
data declaration	03865
(sbptr) (nls,parser,)	0102
data declaration	02952
(sbpush) (nls,psedit,sbpush) 5F	0307
(str1,str2,hed)	03262
.....Documentation.....	03261
(sbrent) (nls,const,)	0586
data declaration	03704
(sbrp) (nls,records,)	01550
data declaration	04668
(sbrun) (nls,const,)	0912
data declaration	04030
(sbstac) (nls,pdata,)	074
data declaration	02935
(sbstar) (nls,const,)	0940
data declaration	04058
(sbstkx) (nls,pdata,)	062
data declaration	02923
(sbtt) (nls,records,)	01392
data declaration	04510
(scanit) (nls,help,scanit) 1D2	02773
(testr,qswork)	06583
(sccsiz) (nls,records,)	01294
data declaration	04412
(scfont) (nls,records,)	01106
data declaration	04224
(schars) (nls,records,)	01089
data declaration	04207
(schinc) (nls,records,)	01320
data declaration	04438
(scnleft) (nls,utility,scnleft) 9M	01580
(scnrht) (nls,utility,scnrht) 9N	01585
(scp) (nls,utility,scp) 9L	01583
(scsize) (nls,records,)	01341
data declaration	04459
(sdaid) (nls,records,)	01102
data declaration	04220
(sdbhdl) (nls,const,)	0665
data declaration	03783

(sdbtyp) (nls,const,)		0588
data declaration		03706
(sdd) (nls,records,)		01049
data declaration		04167
(sdbtx) (nls,poeta,)		014
data declaration		02875
(sds) (nls,records,)		01069
data declaration		04187
(search) (nls,frontend,search) 13D		02025
(target)		05443
search old statement for character in target. return numb of		
characters away from current swrk setting if found, 0 if not found		
		05442
(seemkr) (nls,auxcod,seemkr) 8D		02290
(fileno,astr)		05884
(sellfu) (nls,parser,)		0213
data declaration		03081
(sel2fu) (nls,parser,)		0150
data declaration		03007
(seldsp) (nls,frontend,seldsp) 5C		02003
selective display recreate control		05410
(send) (nls,seqgen,send) 3H		01868
(sw,str)		05173
an ENDFIL or address of a string		05172
(sent) (nls,pdata,)		058
data declaration		02919
(seqbadd) (nls,psupport,seqbadd) 5B		0250
(resultptr,parsemode,seqbstrptr,strptr)		03150
ptr to keyword string		03149
(seqbinit) (nls,psupport,seqbinit) 5A		0251
(resultptr,parsemode)		03152
parsing mode		03151
(seqend) (nls,seqgen,seqend) 3G		01862
(stid,vsl,vs2)		05161
This procedure may be used to find the last branch to be handled by		
the sequence generator depending on the branch or plex only		
viewspecs; the sequence generator itself no longer checks the		
viewspecs. Thus, the stid of the last branch must be passed as a		
parameter to openseq!		05160
(seqgen) (nls,seqgen,seqgen) 3D		01854
(sw)		05146
address of a sequence generator work area		05145
(seqnam) (nls,const,)		0688
data declaration		03806
(seqnxt) (nls,seqgen,seqnxt) 3F		01870
(sw)		05177
address of sequence work area		05176
(seqsrch) (nls,help,seqsrch) 1D8		02755
(testr,stid)		06559
Do a name-in-group search sequentially under the current location -		
if this is the top of a file the whole file will be searched		06558
(secsym) (nls,nddt,)		02457
data declaration		06061
(sequence) (nls,nddt,sequence) 10A10		02353
(rd,symtype,symval)		05957
Standard parms...Parse Sequence		05956

(setabs) (nls,intnls,setabs) 6AC	02848
(tabtbl)	06674
new bit table version of settabs	06673
(setaccess) (nls,ioexec,setaccess) 13B	02201
(fileno,type)	05747
Set the NLS system access for the file indicated by fileno to type	
if access type is legal. Calls err if anything wrong.	05746
(setbackup) (nls,parser,setbackup) 5A2	0178
(firstptr,curptr)	03039
ptr to current path stack entry	03038
(setbit) (nls,utility,setbit) 14B1	01591
(table,bit,length)	04711
(setbkpt) (nls,nddt,setbkpt) 10B8	02363
(addr,test,testval,bpa,bpn)	05967
Create a breakpoint	05966
(setbkup) (nls,psupport,setbkup) 4A	0254
(resultptr,parsemode)	03157
parsing mode	03156
(setcacc) (nls,csendmail,setcacc) 4R	02660
(entrystid,repstr,ptr1,ptr2)	06398
set access field in catalog entry	06397
(setcharsize) (nls,psedit,setcharsize) 3AE1F	0325
(da,charsize)	03298
sets all fields in the da for specified char size	03297
(setcinit) (nls,intnls,setcinit) 6Q	02837
(strng)	06661
(setdev) (nls,intnls,setdev) 6R	02861
(device)	06690
set up device dependent data for passed device	06689
(setdis) (nls,intnls,setdis) 6M	02832
set up display areas	06656
(setfacc) (nls,ioexec,setfacc) 14E	02136
(originstid=@repstr,ptr1,ptr2)	05625
set access field in file origin statement	05624
(setfdb) (nls,ioexec,setfdb) 12C	02206
(dirno,initls,fl)	05755
Set the lock fields in an FDB. Return TRUE if access permitted by	
maywrt. FALSE if not	05754
(setfield) (nls,nddt,setfield) 10A9D	02410
(rd,value)	06014
Accepts a rdecord descripton, and a value. Sets the defined field	
to the value	06013
(setfil) (nls,ioexec,setfil) 7B	02114
(fileno)	05586
Initialize file header and core page status table. The header of	
the file whose number is passed is initialized by this routine. All	
status blocks are set to empty, the marker table is set to empty,	
and the CORPST(core-page-status-table) entries pertaining to this	
file (other than the header) are set to indicate that no pages are	
loaded.	05585
(sethint) (nls,nddt,sethint) 13C	02463
Set up control H as interrupt	06067
(sethrld) (nls,parser,sethrld) 4H	0110
(strptr)	02964
ptr to hearaid name string	02963
(setiaad) (nls,cident,setiaad) 11H	02639

(entrystr,repstr,ptr1,ptr2)	06360
set MAIL ADDRESS field for an ident entry	06359
(seticapability) (nls,cident,seticapability) 11S	02577
(entrystr,repstr,ptr1,ptr2)	06242
set CAPABILITIES field for an ident entry	06241
(seticord) (nls,cident,seticord) 11F	02634
(entrystr,repstr,ptr1,ptr2)	06350
set COORDINATOR field for a group/organization ident entry	06349
(setidelivery) (nls,cident,setidelivery) 11U	02622
(entrystr,repstr,ptr1,ptr2)	06327
set DELIVERY-TYPES field for an ident entry	06326
(setiexp) (nls,cident,setiexp) 11E	02576
(entrystr,repstr,ptr1,ptr2)	06240
set EXPAND field for an ident entry	06239
(setifield) (nls,psident,setifield) 24	0362
(resultptr,parsemode.fieldname.value.value2,value3)	03372
set specified fieldname to specified value	03371
(setifunction) (nls,cident,setifunction) 11Q	02609
(entrystr,repstr,ptr1,ptr2)	06303
set FUNCTION field for an ident entry	06302
(setigrps) (nls,cident,setigrps) 11K	02572
(entrystr,repstr,ptr1,ptr2)	06233
set GROUPS field for an ident entry	06232
(setihost) (nls,cident,setihost) 11L	02625
(entrystr,repstr,ptr1,ptr2)	06333
set NET-DELIVERY HOST field for an ident entry	06332
(setiid) (nls,cident,setiid) 11A	02653
(entrystr,repstr,ptr1,ptr2)	06384
set IDENT field for an ident entry	06383
(setimcmnts) (nls,cident,setimcmnts) 11V	02569
(entrystr,repstr,ptr1,ptr2)	06227
set COMMENTS field for an ident entry	06226
(setimem) (nls,cident,setimem) 11C	02604
(entrystr,repstr,ptr1,ptr2)	06293
set MEMBERSHIP field for a group/organization ident entry	06292
(setinam) (nls,cident,setinam) 11B	02651
(entrystr,repstr,ptr1,ptr2)	06380
set NAME field for an ident entry	06379
(setinlhost) (nls,cident,setinlhost) 11M	02621
(entrystr,repstr,ptr1,ptr2)	06325
set NLS-DELIVERY HOST field for an ident entry	06324
(setinma) (nls,cident,setinma) 11N	02552
(entrystr,repstr,ptr1,ptr2)	06194
set NETWORK MAILBOX address field for an ident entry	06193
(setinterrupts) (nls,intnls,setinterrupts) 6T	02833
set up rubout, command delete, and page write interrupt	06657
(setiorg) (nls,cident,setiorg) 11I	02581
(entrystr,repstr,ptr1,ptr2)	06250
set ORGANIZATION field for an individual ident entry	06249
(setiphone) (nls,cident,setiphone) 11P	02629
(entrystr,repstr,ptr1,ptr2)	06341
set PHONE field for an ident entry	06340
(setiprevmem) (nls,cident,setiprevmem) 11D	02590
(entrystr,repstr,ptr1,ptr2)	06268
set MEMBERSHIPPREVIOUS field for an ident entry	06267
(setisorg) (nls,cident,setisorg) 11R	02560

(entrystr,repstr,ptr1,ptr2)	06210
set SECONDARY ORGANIZATION field for an ident entry	06209
(setisubcol) (nls,cident,setisubcol) 11T	02597
(entrystr,repstr,ptr1,ptr2)	06280
set SUBCOLLECTIONS field for an ident entry	06279
(setityp) (nls,cident,setityp) 11G	02587
(entrystr,repstr,ptr1,ptr2)	06262
set TYPE-OF-ORGANIZATION field for an organization ident entry	06261
(setiuser) (nls,cident,setiuser) 11J	02608
(entrystr,repstr,ptr1,ptr2)	06301
set USER (NLS delivery) field for an ident entry	06300
(setiverify) (nls,cident,setiverify) 11O	02568
(entrystr,repstr,ptr1,ptr2)	06225
set VERIFIED-BY-NIC-PERSONNEL field for an ident entry	06224
(setjaction) (nls,csendmail,setjaction) 4C	02701
(recipientstr)	06476
set Action Distribution field in JWORK header	06475
(setjauthor) (nls,csendmail,setjauthor) 4B	02657
(authorstr)	06392
set the author field of JWORK file	06391
(setjaxaccess) (nls,csendmail,setjaxaccess) 4D	02672
(accessstr)	06420
set the access list field in JWORK header	06419
(setjcomment) (nls,csendmail,setjcomment) 4F	02704
(commentstr)	06482
set Comment field in JWORK header	06481
(setjexpedite) (nls,csendmail,setjexpedite) 4G	02678
(onoff)	06431
set or clear expedite attribute in JWORK header	06430
(setjinfo) (nls,csendmail,setjinfo) 4E	02685
(recipientstr)	06445
set Information Distribution field in JWORK header	06444
(setjkeyword) (nls,csendmail,setjkeyword) 4I	02698
(keywordstr)	06470
set keywords field in JWORK header	06469
(setjlink) (nls,csendmail,setjlink) 4J	02705
(keywordstr)	06484
set INSERT LINK (***FIX THIS***) field in JWORK header	06483
(setjnumber) (nls,csendmail,setjnumber) 4H	02709
(numberstr)	06492
set the contents of the Number field in the JWORK header	06491
(setjobsolates) (nls,csendmail,setjobsolates) 4N	02664
(obsoletestr)	06405
set Obsoletes Document(s) field in JWORK header	06404
(setjrffc) (nls,csendmail,setjrffc) 4K	02688
(rffcstr)	06451
set RFC field in JWORK header	06450
(setjsource) (nls,csendmail,setjsource) 4L	02669
(sourcetype,s1,s2)	06414
set source type and content in JWORK header	06413
(setjsubcol) (nls,csendmail,setjsubcol) 4M	02693
(subcollstr)	06461
set Sub-Collections field in JWORK header	06460
(setjtitle) (nls,csendmail,setjtitle) 4Q	02703
(titlestr)	06480
set Title field in JWORK header	06479

(setjunrecorded) (nls,csendmail,setjunrecorded) 40	02697
(cnooff)	06468
set or clear UNRECORDED attribute in JWORK heade	06467
(setjupdates) (nls,csendmail,setjupdates) 4P	02699
(updatestr)	06472
set Updates Document(s) field in JWORK header	06471
(setkey) (nls,colstr,setkey) 13	02342
(ptr1,ptr2)	05942
(setlit) (nls,inprfbk,setlit) 9H4	0496
set up for literal collection	03588
(setlt) (nls,prmspc,setlt) 9E	02059
(setchr,vsl,vs2)	05486
This routine adjusts the viewspecs in accord with characters entered during view specification. Saves the viewspec words on the stack savevspec for each character input. When a BC is input the stack is popped. The string being displayed in the name area is updated accordingly.	05485
(setpriority) (nls,recfil,setpriority) 4	02510
(qcode)	06139
(setrd) (nls,nddt,setrd) 11A2	02359
(rd)	05962
(setrot) (nls,cedit,setrot) 32D1	01697
(ptr.fnstng,dtstng)	04850
(settimer) (nls,auxcod,settimer) 3C	02305
(milliseconds,proc,arg1,arg2,arg3,arg4)	05903
set the system timer -- after MILLISECONDS milliseconds, PROC will be called with ARG1 thru ARG4	05902
(setupop) (nls,recfil,setupop) 5	02519
(jfn,filnamstr,modelg,TTY)	06143
This procedure opens a file with the name in filnamstr if jfn = 0, otherwise uses file identified by jfn	06142
(setvec) (nls,intrnl,setvec) 6B	02831
Sets entry vector for NLS	06655
(sflg) (nls,Chelp,)	02807
data declaration	06632
(sfmacpt) (nls,seqfil,sfmacpt) 3H	02255
(da,sw,toplev)	05837
This copies the contents of READC to a local string, for output to the assembler. It works like sputst, except it inserts tabs at the beginning of statements that are of a lower level than the first statement in the series. In addition a statement consisting of only one space will be output as a null line.	05836
(sfnextst) (nls,seqfil,sfnextst) 3F	02248
(sw)	05827
This procedure calls the sequence generator to find the next statement. If there are no more statements in the sequence, it returns FALSE. Otherwise, it initializes COPOS for reading the next statement, which may involve skipping the statement name.	05826
(sfont) (nls,records,)	01154
data declaration	04272
(sfputst) (nls,seqfil,sfputst) 3G	02249
(da,sw,typfil)	05829
This copies the contents of READC to a local string. It expects READC to be set up to begin reading. When it completes processing this statement it performs the necessary cleanup to begin processing the next statement (by calling SFENDST) and returns.	05828

```

(sftrln) (nls,seqfil,sxtrln) 3I 02231
  (begbp,bytptr) 05803
  This routine copies characters bounded by begbp and bytptr (both
  byte pointers) into sbuf and pmaps sbuf into the print file when
  the buffer is full. 05802
(sgarb) (nls,records,) 01196
  data declaration 04314
(sgtjfn) (nls,ioexec,sgtjfn) 8D 02169
  (flags,flnam,errstr) 05665
  short gtjfn jsys-- set up registers and execute short gtjfn.
  Returns jfn for file whose name is specified in flnam if successful,
  FALSE with error message in string whose address is passed if not.
  The file name is checked to see if it is a Journal item. 05684
(shinc) (nls,records,) 01123
  data declaration 04241
("hl) (nls,nddt2,shl) 1D2R19 02504
  (value,shiftcount) 06117
  this routine performs the logical left shift of the specified
  number 06116
(showfield) (nls,psident,showfield) 25 0365
  (resultptr,parsemode,fieldname) 03378
  show the specified fieldname of current record 03377
(showaddr) (nls,nddt,showaddr) 10A11 02433
  (rd) 06037
  Parns are: 06036
(shr) (nls,nddt2,shr) 1D2R20 02506
  (value,shiftcount) 06121
  this routine performs the logical right shift of the specified
  number 06120
(shutdis) (nls,auxcod,shutdis) 4M 02299
  shut down display before leave NLS 05896
(sid) (nls,const,) 0798
  data declaration 03916
(siftup) (nls,colsrt,siftup) 27 02347
  (i,n) 05949
  sort tree into vector...from floydd 05948
(sigclose) (nls,ioexec,sigclose) 5C 02208
  (fileno) 05759
  Close file after SIGNAL. Called in subsystems with NLS system
  files; If journal file, lock journal system. Close the file
  specified by fileno. 05758
(sign) (nls,ccalc,) 02523
  data declaration 06150
(sim) (nls,pdata,) 046
  data declaration 02907
(sinit) (nls,records,) 01317
  data declaration 04435
(skip) (nls,frontend,skip) 13E 01971
  (numb) 05360
  skip a numb of characters in old and type percent signs 05359
(skbach) (nls,const,) 0971
  data declaration 04089
(slengt) (nls,records,) 01058
  data declaration 04176
(sleuth) (nls,parser,sleuth) 5A3 0103
  (psptr,optptr,captr,defaultptr) 02954

```

ptr to default action ptr	02953
(slink) (nls,pdata,)	076
data declaration	02937
(slngth) (nls,utility,slngth) 8Y	01569
(bp1,bp2)	04709
string length	04708
(slnmdl) (nls,records,)	01128
data declaration	04246
(sltp11) (nls,parser,)	0203
data declaration	03068
(sltp12) (nls,parser,)	0141
data declaration	02999
(sltp21) (nls,parser,)	0128
data declaration	02984
(sltp22) (nls,parser,)	0190
data declaration	03052
(sname) (nls,records,)	01072
data declaration	04190
(softcd) (nls,prmspc,softcd) 10	02033
This routine is generally used by the bug accepting routines. It is called when they encounter a backspace character while waiting for a bug mark. This routine goes to state if there are no lower entries on the stack; otherwise, it pops the return stack, uses DELBM to erase the most recent bug mark, and returns.	
(sort) (nls,colsrt,sort) 7A	05454
(stidx,stidl,stid2,rlevcnt,kpr,copyf)	02321
(sortbranch) (nls,colsrt,sortbranch) 6C	05921
(stidx)	02316
(sorter) (nls,const,)	05915
data declaration	0664
(sortgrp) (nls,colsrt,sortgrp) 6D	03782
(bug1,bug2)	02317
(sortplx) (nls,colsrt,sortplx) 6E	05916
(bug1)	02324
(source) (nls,pdata,)	05924
data declaration	080
(sp) (nls,psupport,sp) 3A1	02941
(curptr,parsemode,string)	0241
sp looks at the next inpt character, if it is a space, then the space is read and a true return is taken. If the next character is not a space, then it is not read, and FALSE is returned	
(spacer) (nls,const,)	03131
data declaration	01000
(spaddr) (nls,nddt,)	04118
data declaration	02396
(span) (nls,utility,span) 9R	06001
(chrcls)	01615
(spec) (nls,const,)	04738
data declaration	0938
(specreg) (nls,adrmp,specreg) 1R8C	04056
(astrng,type,ptr)	02084
spec a register. This procedure converts a string or a statement identifier to a t-pointer. The conversion algorithm depends on the first character in the register, and on the parameter passed as the second argument. If a string is being matched, the routine expects as a third argument the stid at which the search should begin,	
	05533
	05532

(specttyout) (nls,inpfbk,specttyout) 10H	0526
(ttyno, astr)	03632
This procedure accepts a teletype numbr and a string, and types the string as a message on the tty.	03631
(splite) (nls,pscalc,)	0454
data declaration	03528
(sport) (nls,seggen,sport) 3I	01863
(sw)	05163
send-port mechanism	05162
(spsid) (nls,records,)	01556
data declaration	04674
(sptr) (nls,utility,sptr) 9H4	01621
(sqcls) (nls,const,)	0966
data declaration	04084
(sqerr) (nls,const,)	0956
data declaration	04074
(sqgnxt) (nls,const,)	0713
data declaration	03831
(sqinit) (nls,seggen,sqinit) 3J	01852
sequence generator init	05142
(sqopn) (nls,const,)	0596
data declaration	03714
(sqstks) (nls,const,)	0686
data declaration	03804
(sqwrkl) (nls,records,)	01310
data declaration	04426
(srlset) (nls,utility,srlset) 8W	01675
(char,ast)	04809
To set an A-string to a single character, call this procedure with the character and the address of the string.	04808
(srcc) (nls,records,)	01455
data declaration	04573
(srch) (nls,dspgen,srch) 3B7	01919
(stid,level,match,top)	05263
find already formatted stid	05262
(srexis) (nls,records,)	01480
data declaration	04598
(srhexi) (nls,records,)	01500
data declaration	04618
(srhfil) (nls,records,)	01472
data declaration	04590
(srhfina) (nls,records,)	01110
data declaration	04228
(srhlas) (nls,records,)	01048
data declaration	04166
(srhtop) (nls,records,)	01190
data declaration	04308
(srmake) (nls,utility,srmake) 8P	01676
(value, astrng, base)	04811
make string from value	04810
(srmk) (nls,utility,srmk) 8Q	01571
(value, base)	04692
In string construction use apachr to append string to sar.	04691
(srnmdl) (nls,records,)	01146
data declaration	04264
(srpsid) (nls,records,)	01139

data declaration	04257
(srrcnt) (nls,pdata,)	075
data declaration	02936
(srrele) (nls,records,)	01285
data declaration	04403
(srrhle) (nls,records,)	01387
data declaration	04505
(srrlength) (nls,frontend,srrlength) 12M	01989
(srr)	05388
return number of entries in statement return ring.	05387
(srrmax) (nls,const,)	0761
data declaration	03879
(srrsiz) (nls,const,)	0937
data declaration	04055
(srsup) (nls,utility,srsup) 9H	01640
dummy procedure for string support code	04765
(srval) (nls,utility,srval) 8R	01657
(astrng,base)	04785
convert string to value	04784
(srvs1) (nls,records,)	01188
data declaration	04306
(srvs2) (nls,records,)	01081
data declaration	04199
(sseqgen) (nls,seqgen,sseqgen) 3E	01871
(sw)	05179
NOBODY BUT SEQGEN SHOULD CALL THIS ROUTINE	05178
(sstr) (nls,const,)	0561
data declaration	03679
(sstrid) (nls,records,)	01218
data declaration	04336
(ssysna) (nls,pdata,)	025
data declaration	02886
(stadr) (nls,records,)	01521
data declaration	04639
(standout) (nls,dspgen,standout) 7N	01912
Send stand out following characters command - alphanumeric displays	05251
(startp) (nls,const,)	01007
data declaration	04125
(startr) (nls,const,)	0553
data declaration	03671
(startup) (nls,intnls,startup) 5C	02824
initialize the world	06647
(stastr) (nls,records,)	01401
data declaration	04519
(statem) (nls,const,)	0703
data declaration	03821
(states) (nls,const,)	01026
data declaration	04144
(stblk) (nls,records,)	01393
data declaration	04511
(stbpget) (nls,cedit,stbpget) 28C	01765
(stid,cnt)	04979
statement byte pointer get	04978
(stbptr) (nls,utility,stbptr) 8J	01649
(charno)	04775

character byte pointer	04774
(stdr) (nls,select,stdr) 1B2A	0227
(bug,tptr1,tptr2)	03106
second text pointer	03105
(stdtab) (nls,const,)	0865
data declaration	03983
(stdvsp) (nls,const,)	0983
data declaration	04101
(stfile) (nls,records,)	01122
data declaration	04240
(stfrmt) (nls,dspgen,stfrmt) 3B1	01913
(da,stid,charct,level,origlev,stvec,viewspec1,vwspc2)	05253
statement format	05252
(stime) (nls,records,)	01418
data declaration	04536
(stkcnt) (nls,help,)	02752
data declaration	06555
(stkhsh) (nls,help,)	02765
data declaration	06573
(stklit) (nls,prmspc,stklit) 9C	02052
(setchr,vsl,vs2)	05475
puts input viewspec characters on stack, appends them to string vspstr, and calls setlt to activate them.	05474
(stkmax) (nls,help,)	02761
data declaration	06567
(stkovr) (nls,intnls,stkovr) 5A10A	02811
(stksym) (nls,nddt,)	02401
data declaration	06005
(stktyp) (nls,help,)	02738
data declaration	06539
(stmtv) (nls,const,)	0631
data declaration	03749
(stnamcap) (nls,cident,stnamcap) 13A	02617
(string)	06317
Capitalize first letter of each word in a string	06316
(stofhd) (nls,filmp,stofhd) 2D4	01796
(stid,head)	05037
store head flag of specified statement.	05036
(stoftl) (nls,filmp,stoftl) 2D3	01800
(stid,tail)	05045
store tail flag of specified statement.	05044
(stonam) (nls,filmp,stonam) 2D7	01812
(stid,nameh)	05068
store the name hash of specified statement.	05067
(stonmf) (nls,filmp,stonmf) 2D5	01810
(stid,name)	05064
store name flag of specified statement.	05063
(stopline) (nls,auxcod,stopline) 5C	02308
↑S interrupt routine	05908
(stoptimer) (nls,auxcod,stoptimer) 3D	02286
stop the system timer	05878
(storbp) (nls,nddt,storbp) 10B9	02413
store the breakpoints in the table	06017
(storesrring) (nls,frontend,storesrring) 12K	01987
(srr.index,stid,cc,vsl,vs2)	05384
store contents onto statement return ring. INDEX has the same	

semantics as in readsrring.	05383
(stosdb) (nls,filmp,stosdb) 2D6	01804
(stid,stdb)	05053
Store the psdb of specified statement.	05052
(stosid) (nls,filmp,stosid) 2D8	01846
(stid,sid)	05131
Store the name hash of specified statement.	05130
(stosub) (nls,filmp,stosub) 2D2	01792
(stid,substid)	05029
Store sub of specified statement.	05028
(stosuc) (nls,filmp,stosuc) 2D1	01836
(stid,sucstid)	05113
Store successor of specified statement.	05112
(stpos) (nls,seqgen,stpos) 4G	01855
(stid)	05148
Given a STID, this routine returns an integer which is the position of that statement within its plex.	05147
(stpsdb) (nls,records,)	01155
data declaration	04273
(stpsid) (nls,records,)	01112
data declaration	04230
(stptset) (nls,cident,stptset) 8B	02631
(string,nptrl,nptr2,optrl,optr2)	06345
set string and/or text-pointers for get routines	06344
(strbug) (nls,prmspc,strbug) 11	02053
(ptr)	05477
this routine will store the current bug mark in the pointer passed it.	05476
(strchr) (nls,psuserop,strchr) 2D1	0437
(cc)	03503
converts string representation for character to internal code	03502
(strdev) (nls,psuserop,strdev) 2D3	0410
(dev)	03461
converts string representation for device to internal code	03460
(strngrec) (nls,nddt,strngrec) 11B6B	02383
(rd)	05988
(strngunderflow) (nls,nddt,strngunderflow) 11B6C	02356
(rd)	05960
(stsid) (nls,records,)	01120
data declaration	04238
(stsidf) (nls,records,)	01439
data declaration	04557
(stvect) (nls,seqgen,stvect) 4E	01865
(stid,stvwrk)	05167
This routine generates a statement vector. It is called with a STID as the first argument and the address of the first word of a 17 word work area as the second. The number of words in the work area determine how many levels down in structure the routine can go (currently 17). ERR(5) is called if the work area is not long enough to contain the vector for the given STID. STVECT uses routine STPOS.STVECT returns an integer, which is the level of the statement. It also creates a statement vector. Upon completion, the first word of the work area contains the level of the STID and subsequent cells contain the position within the respective plexes. Thus for statement 1D2, the vector would contain (3, 1, 4, 2) in the first four cells.	05166

(stvmcd) (nls,seggen,stvmcd) 4F	01859
(stvwrk,type)	05156
This routine modifies a statement vector. It is called with the address of a statement vector as the first argument, and a type parameter as the second. The statement vector is assumed to be initialized, and is modified as specified by TYPE.	
(stwc) (nls,records,)	05155
data declaration	01148
(sub) (nls,const,)	04266
data declaration	0974
(subldsp) (nls,psedit,subldsp) 5C	04092
(resultptr,parsemode,textent,newflag)	0312
TRUE if new text entity is being collected	03272
(sub2dsp) (nls,psedit,sub2dsp) 5D	03271
(resultptr,parsemode,type,tptr)	0269
ptr to string tptr record	03187
(subcal) (nls,subcal) **Couldn't open <NLS>nls,subcal<↑F>NLS**	03186
(subdir) (nls,const,)	098
data declaration	0909
(subhel) (nls,subhel) **Couldn't open <NLS>nls,subhel<↑F>NLS**	04027
(subide) (nls,subide) **Couldn't open <NLS>nls,subide<↑F>NLS**	090
(subidstr) (nls,cident,subidstr) 12F	091
(string,oidstr,nidstr)	02610
support routine for MODREFS	06305
(subpairs) (nls,psupport,subpairs) 311	06304
(curptr,parsemode,string)	0255
collects a number of pairs for the substitute command and leaves a pointer to the work area in the record pointed to by curptr. It also leaves data about what must be removed from the display	03159
(subpro) (nls,subpro) **Couldn't open <NLS>nls,subpro<↑F>NLS**	03158
(subpsave) (nls,psedit,subpsave) 5E	094
(resultptr,parsemode,newptr,oldptr)	0296
ptr to the old entity	03240
(subque) (nls,subque) **Couldn't open <NLS>nls,subque<↑F>NLS**	03239
(subsen) (nls,subsen) **Couldn't open <NLS>nls,subsen<↑F>NLS**	097
(subsinit) (nls,psedit,subsinit) 5A	099
(resultptr,parsemode,textent)	0262
text entity type for the substitute	03173
(subsup) (nls,subsup) **Couldn't open <NLS>nls,subsup<↑F>NLS**	03172
(subuse) (nls,subuse) **Couldn't open <NLS>nls,subuse<↑F>NLS**	092
(subxxx) (nls,subxxx) **Couldn't open <NLS>nls,subxxx<↑F>NLS**	069
(suc) (nls,const,)	093
data declaration	0817
(succdir) (nls,const,)	03935
data declaration	0795
(sucstr) (nls,const,)	03913
data declaration	0573
(supda) (nls,dspgen,supda) 6B	03691
(da)	01892
suppress display area	05217
(supervisor) (nls,parser,supervisor) 4F	05216
main control routine	0145
(subloc) (nls,nddt,)	03003
data declaration	02370
(svmxle) (nls,const,)	05973
data declaration	0802
	03920

(swallo) (nls,records,)		01262
data declaration		04380
(swcaco) (nls,records,)		01129
data declaration		04247
(swcall) (nls,records,)		01217
data declaration		04335
(swclvl) (nls,records,)		01442
data declaration		04560
(swcsti) (nls,records,)		01476
data declaration		04594
(swkflg) (nls,records,)		01263
data declaration		04381
(swlbst) (nls,records,)		01056
data declaration		04174
(swmrsa) (nls,records,)		01115
data declaration		04233
(swork1) (nls,const,)		0724
data declaration		03842
(swslvl) (nls,records,)		01518
data declaration		04636
(swsrse) (nls,records,)		01343
data declaration		04461
(swstid) (nls,records,)		01264
data declaration		04382
(swsvw) (nls,records,)		01417
data declaration		04535
(swusqc) (nls,records,)		01157
data declaration		04275
(swvsp2) (nls,records,)		01491
data declaration		04609
(swvspe) (nls,records,)		01076
data declaration		04194
(sxcord) (nls,records,)		01071
data declaration		04189
(sycord) (nls,records,)		01382
data declaration		04500
(sys2rt) (nls,utility,sys2rt) 4B		01592
return with 2 results		04712
(sysble) (nls,records,)		01216
data declaration		04334
(sysclose) (nls,ioexec,sysclose) 8B		02202
(jfn,astng)		05749
Issue close jsys-- Return TRUE if file closed, FALSE if not with		
error message in string whose address was passed.		05748
(syserr) (nls,const,)		0850
data declaration		03968
(sysmrt) (nls,utility,sysmrt) 4C		01620
for returns with multiple results		04742
(sysopen) (nls,ioexec,sysopen) 8A		02125
(jfn,accessmode,type,astng)		05605
Open jsys-- set up registers for and execute openf JSYS. Return		
TRUE if successful, FALSE if failure with error message in string		
whose address was passed as fourth parameter.		05604
(sysovr) (nls,invnls,sysovr) 5A9A		02843
(sysrtn) (nls,utility,sysrtn) 4A		01655
for returns		04783

(syssig) (nls,utility,syssig) 4D	01606
(systke) (nls,utility,) data declaration	01674 04807
(systkp) (nls,utility,) data declaration	01582 04701
(systks) (nls,utility,) data declaration	01598 04718
(systkt) (nls,utility,) data declaration	01645 04772
(t1word) (nls,parser,) data declaration	0179 03040
(t2word) (nls,parser,) data declaration	0167 03027
(t3word) (nls,parser,) data declaration	0174 03034
(t4word) (nls,parser,) data declaration	0157 03017
(t5word) (nls,parser,) data declaration	0169 03029
(tblsearch) (nls,nddt,tblsearch) 13E (tbl.tblsize,entsize,value) Generalised table searcher...return TRUE, address of entry and entry number if ok, FALSE otherwise	02453 06057 06056
(tbound) (nls,frontend,) data declaration	01983 05377
(tbnstr) (nls,const,) data declaration	0960 04078
(tchf) (nls,utility,tchf) 9S (char)	01581 04700
(tdfltv) (nls,const,) data declaration	0696 03814
(tdr) (nls,select,tdr) 103 (bug1,bug2,ptr1,ptr2)	0251 03113
(tenexverno) (nls,intnls,tenexverno) 6Z get version number of current TENEX	02829 06653
(tenfil) (nls,const,) data declaration	0799 03917
(tersem) (nls,const,) data declaration	0574 03692
(testselect) (nls,parser,testselect) 5A5 (curptr) ptr to path stack entry	0214 03083 03082
(text) (nls,const,) data declaration	0613 03731
(textv) (nls,const,) data declaration	0847 03965
(thwfil) (nls,auxcod,thwfil) 4J (filno) thaw file pages of file	02266 05847 05846
(thwrfp) (nls,auxcod,thwrfp) 4I thaw random file pages	02300 05897
(tinptc) (nls,inpfbk,tinptc) 5A return a translated, shifted char	0509 03607
(tip) (nls,pdata,) data declaration	023 02884
(tmchr1) (nls,records,)	01268

data declaration	04386
(tmchr2) (nls,records,)	01363
data declaration	04481
(tmchr3) (nls,records,)	01520
data declaration	04638
(tmchr4) (nls,records,)	01158
data declaration	04276
(tmchr5) (nls,records,)	01364
data declaration	04482
(tmchr6) (nls,records,)	01389
data declaration	04507
(tnls) (nls,intnls,tnls) 4C	02827
start Typewriter NLS	06650
(tnlscm) (nls,parser,)	0192
data declaration	03054
(tnltyp) (nls,const,)	0715
data declaration	03833
(todco) (nls,inpfbk,todco) 6A	0490
(char)	03581
output a character to tty if feedbk > 0	03580
(topact) (nls,nddt,)	02425
data declaration	06030
(topcon) (nls,chelp,)	02788
data declaration	06605
(topsch) (nls,chelp,topsch) 1D9	02778
(testr)	06591
Set current location to top of data base. Call seqsrch to search the file. Check to see if there are more files in the data base.	
	06590
(totent) (nls,chelp,)	02747
data declaration	06549
(tpoffs) (nls,const,)	0755
data declaration	03873
(track) (nls,dspgen,track) 7G	01887
resume tracking - alphanumeric displays	05208
(transdir) (nls.ioexec,transdir) 9H	02115
(string)	05588
Translates string to directory number using stdir jsys. Accepts address of a string, appends 0 to it as required by TENEX. If not a match calls err. Else returns directory number.	05587
(trapc) (nls,auxcod,trapc) 5D	02288
no-op control c	05881
(trapcc) (nls,utility,trapcc) 12A	01634
(trapo) (nls,auxcod,trapo) 5E	02294
no-op control o	05890
(traps) (nls,auxcod,traps) 5F	02302
no-op control s	05900
(trfname) (nls,nddt,trfname) 8A	02476
(bp,rd)	06078
Types out a field of a record with the appropriate name	06077
(triflandpc) (nls,execfl,triflandpc) 3J	01929
(fifn,pcjfn,nver,erstring)	05283
trim file (and partial copy) for input jfns	05282
(trmtyp) (nls,records,)	01424
data declaration	04542
(trnchk) (nls,credit,trnchk) 22E1	01681

(ptr1,ptr2,ptr3,ptr4)		04820
moves pointers around for transpose number		04819
(trnsint) (nls,filmap,trnsint) 2K2		01795
(initials,string)		05035
Given a word of 5-bit initials, this routine translates them and appends them to the string whose address is passed.		05034
(trntst) (nls,credit,trntst) 29C		01780
(first1,first2,secnd1,secnd2)		05007
This routine checks that the groups specified by the transpose commands have legal bounds. It does this by calling MOVST to see if FIRST1,2 are in the group bounded by SECND1,2 or SECND1,2 in the group bounded by FIRST1,2.		05006
(tslshc) (nls,const,)		0772
data declaration		03890
(tstart) (nls,frontend,tstart) 14C		01972
(gs)		05362
TNLS startup routine		05361
(tstf) (nls,utility,tstf) 9E		01658
(ast)		04787
Like tst except looks for any occurrence of the test string in the remaining portion of the T-string. This is done by first scanning thru looking for the first character of the string, then if find it the rest of the string is compared. If get a mismatch SWORK is reset to character after the start of the current partial match and the process is repeated. Failure occurs only when the T-string is exhausted. Returns with SKIP RETURN on success, normal return on failure.		04786
(tstmenu) (nls,help,tstmenu) 1812		02764
(stid)		06572
test if query menu stmt		06571
(tstr) (nls,const,)		0895
data declaration		04013
(tstsr) (nls,utility,tstsr) 9D		01569
(ast)		04688
Test string. Compare the T-string specified by SWORK with the string whose address is passed as arg. Fetches characters from string by calling READC. On a successful match does SKIP RETURN and SWORK is left pointing to the character after the pattern. If the match fails, normal return.		04687
(ttycoc) (nls,const,)		0832
data declaration		03950
(ttylno) (nls,records,)		01086
data declaration		04204
(ttywindow) (nls,intnls,ttywindow) 6N		02847
(da)		06672
make a da a tty window		06671
(tuacc) (nls,const,)		0776
data declaration		03894
(txtdat) (nls,utility,txtdat) 9U		01602
(txttext) (nls,const,)		0780
data declaration		03898
(txtlit) (nls,inpfbk,txtlit) 5B		0488
(astrng)		03577
passed the address of a string, appends text from keyboard input buffer to string		03576
(tybbrk) (nls,ndat,tybbrk) 10B10		02432

(bb)		06035
(typeas) (nls,auxcod,typeas) 3F		02284
(astrng)		05875
Type a-string on tty		05874
(typebuff) (nls,inpfbk,typebuff) 6E		0518
Type out the contents of the lock ahead input buffer		03622
(typeca) (nls,const,)		0861
data declaration		03979
(typech) (nls,auxcod,typech) 3G		02268
(char)		05850
type (translated) character on tty		05849
(typectl) (nls,inpfbk,typectl) 6B		0500
(char)		03594
type (translated) character on tty - prints special chars		03593
(typeli) (nls,const,)		0836
data declaration		03954
(typentity) (nls,nddt,typentity) 8B		02483
(rd,curmode)		06087
Types out the entity described by rd in appropriate manner		06086
(typenu) (nls,const,)		0819
data declaration		03937
(typewr) (nls,const,)		01028
data declaration		04146
(typfield) (nls,nddt,typfield) 8C		02490
(rd,curmode)		06094
rd = address of Record Descriptor, curmode = default mode		06093
(typframe) (nls,nddt,typframe) 11B5F		02464
(bp,rd)		06068
(typical) (nls,nddt,typical) 11B2B		02399
(bp,rd)		06004
(typloc) (nls,nddt,typloc) 8D		02367
(rd)		05970
(typparm) (nls,nddt,typparm) 11B3B		02456
(bp,rd)		06060
(typstmx) (nls,nddt,typstmx) 11B6D		02417
(bp,rd)		06022
(typstsz) (nls,nddt,typstsz) 11B6E		02397
(bp,rd)		06002
(typtrace) (nls,nddt,typtrace) 10B11		02414
(addr)		06019
Types out a trace message for address		06018
(typval) (nls,nddt,typval) 8E		02461
(value,mode)		06065
(ubrstr) (nls,const,)		0646
data declaration		03764
(ue) (nls,const,)		0712
data declaration		03830
(uflow) (nls,auxcod,uflow) 6A		02283
general stack underflow routine		05873
(undflandpc) (nls,execrl,undflandpc) 3I		01944
(jfn,pcjfn,erstrng)		05313
undelete file (and partial copy) for input jfns		05312
(unlink) (nls,stmgt,unlink) 3F		02214
(blk.zone)		05771
unlink a block from its free list		05770
(unlkclist) (nls,ioexec,unlkclist) 11A		02153

```

after unlock file check clist entries for validity; clean up if
necessary
(unlkfile) (nls,loexec,unlkfile) 12F 05655
(fileno) 02103
Unlock file execution procedure 05567
(unlkjo) (nls,csendmail,unlkjo) 9C 05566
(type) 02677
Type indates flag to be unlocked... -1 means all 06429
(unprseinnt) (nls,execfl,unprseinnt) 3B 06428
(fifn,ifnflgs,udirname,ufilename,uextname,uverno,ulftovr,uname,flags) 01950
parse jfn into individual strings 05324
(unput) (nls,inpfbk,unput) 10E 05323
undo the effect of the last call to input 0536
(unsethint) (nls,nddt,unsethint) 13D 03647
remove control H as interrupt 02434
(untyp) (nls,chehp,) 06038
data declaration 02770
(uoaccess) (nls,psuserop,uoaccess) 2B4 06579
(pagno,access) 0412
sets the access to user options file 03464
(uoclose) (nls,psuserop,uoclose) 2B3 03463
close the user options file 0403
(uoget) (nls,psuserop,uoget) 2B2 03450
get the user profile page(s) 0422
(uorescont) (nls,psuserop,uorescont) 2B6 03476
reset nonstandard ctrichar table 0430
(uoreset) (nls,psuserop,uoreset) 2B1 03490
reset all user options 0420
(uoshcc) (nls,psuserop,uoshcc) 2C3 03475
(device) 0434
display control chars for specific terminal 03498
(uoshctrl) (nls,psuserop,uoshctrl) 2C2 03497
displays U-OP ctrichar definitions 0424
(uoshcure) (nls,psuserop,uoshcure) 2C4 03481
displays U-OP current context length 0409
(uoshfeed) (nls,psuserop,uoshfeed) 2C5 03459
displays U-OP feedback paramters 0411
(uoshjmp) (nls,psuserop,uoshjmp) 2C7 03462
displays U-OP jump ring paramters 0416
(uoshlev) (nls,psuserop,uoshlev) 2C8 03470
displays U-OP level adjust prompting status 0407
(uoshow) (nls,psuserop,uoshow) 2C1 03456
show entire user options data 0404
(uoshpmt) (nls,psuserop,uoshpmt) 2C10 03451
displays U-OP prompt parameters 0426
(uoshprint) (nls,psuserop,uoshprint) 2C9 03484
displays U-OP print option parameters 0413
(uoshreco) (nls,psuserop,uoshreco) 2C11 03465
displays U-OP recognition parameters 0418
(uoshrlid) (nls,psuserop,uoshrlid) 2C6 03472
displays U-OP herald parameters 0417
(uoshview) (nls,psuserop,uoshview) 2C12 03471
displays U-OP viewspecs 0435
(utabs) (nls,psuserop,utabs) 2B7 03499
(tabstr) 0405
03453

```

parse user string and set tabs		03452
(uotblget) (nls,psuserop,uotblget) 2B5		0429
get a "free" entry in "cctbl"		03469
(uovers) (nls,const,)		0625
data declaration		03743
(up) (nls,const,)		0945
data declaration		04063
(upcase) (nls,const,)		0890
data declaration		04008
(upcomp) (nls,credit,)		01783
data declaration		05012
(upctbl) (nls,filmap,upctbl) 2E4A		01790
(oldsid,rplsid,newsid)		05025
given three stid's, this routine will update occurrences of the first stid, using rplsid as the stid that contains the text corresponding to oldsid, and newsid as the stid that is the replacement for oldsid. (rplsid should be endfil if the original text has been eliminated.)		05024
(update) (nls,colsrt,update) 8E		02339
(stidx.inlst,kpr,udpr,percent,initf)		05940
(updtfl) (nls,credit,updtfl) 32B		01778
(fileno,vertype,filename)		05003
does Update File command		05002
(updtpg) (nls,credit,updtpg) 32C		01737
(rpage,pgindx,newjfn)		04926
used in update to map a page from the destination file into the update workarea page (beginning at ckpag), then block transfers a page from the original file or the PC to the workarea; finally puts this page back into the destination file page by pmapping it back out (TENEX page tables keep track of this); page removed from NLS core page status table		04925
(upfchgrps) (nls,cident,upfchgrps) 12G		02623
(entry.openidfileno)		06329
propagate membership change to other idents		06328
(upgcnv) (nls,usrpgm,upgcnv) 16		02728
(str)		06525
get user program address from name or number		06524
(upgerr) (nls,const,)		0796
data declaration		03914
(upgjfn) (nls,usrpgm,upgjfn) 19		02714
(adstr)		06501
get a jfn for parsed link data structure		06500
(upgsstr) (nls,usrpgm,upgsstr) 10N		02713
(pgmaddr,str)		06499
(upgtyp) (nls,const,)		0846
data declaration		03964
(upidfil) (nls,cident,upidfil) 12A		02582
(idstr,infostr,fileno,doit)		06252
update add a new record to the master ident-file		06251
(uplsfnm) (nls,frontend,uplsfnm) 12F		01990
(ofn.nfn)		05390
change file name ofn to nfn in all file return rings		05389
(us) (nls,Const,)		0831
data declaration		03949
(usedcnum) (nls,cathum,usedcnum) 2G		02508
(numbsr,nstid)		06125


```

This routine moves the number in number from the in-use or
pre-assigned list to the used list
(usedids) (nls,cident,usedids) 16A1A
(useda) (nls,const,)
data declaration
(usesrr) (nls,pdata,)
data declaration
(uslstr) (nls,const,)
data declaration
(usrexp) (nls,const,)
data declaration
(usrflg) (nls,records,)
data declaration
(usrgrp) (nls,records,)
data declaration
(usrina) (nls,const,)
data declaration
(usrjob) (nls,records,)
data declaration
(usrigd) (nls,const,)
data declaration
(usrlnk) (nls,records,)
data declaration
(usrmod) (nls,parser,)
data declaration
(usrno) (nls,records,)
data declaration
(usrnot) (nls,const,)
data declaration
(usroff) (nls,const,)
data declaration
(usrrt) (nls,records,)
data declaration
(usrtim) (nls,records,)
data declaration
(usrtod) (nls,records,)
data declaration
(usrtty) (nls,records,)
data declaration
(ustar) (nls,records,)
data declaration
(ustr) (nls,const,)
data declaration
(utilho) (nls,const,)
data declaration
(utilop) (nls,const,)
data declaration
(utilst) (nls,const,)
data declaration
(uustr) (nls,const,)
data declaration
(v) (nls,const,)
data declaration
(val2) (nls,parser,)
data declaration
(vb) (nls.const,)

```

```

06124
02650
0985
04103
073
02934
0591
03709
0806
03924
01512
04630
01282
04400
0783
03881
01142
04260
01023
04141
01082
04200
0112
02967
01260
04378
0590
03708
0675
03793
01559
04677
01445
04563
01097
04215
01152
04270
01395
04513
0758
03876
0863
03981
01014
04132
0663
03781
0642
03760
0925
04043
0149
03006
0821

```

data declaration	03939
(vcv) (nls,const,)	0840
data declaration	03958
(vcvend) (nls,const,)	0917
data declaration	04035
(vcvu) (nls,const,)	01012
data declaration	04130
(vcvul) (nls,const,)	0951
data declaration	04069
(vdr) (nls,select,vdr) 1C7	0220
(bug,ptr1,ptr2)	03094
(ve) (nls,const,)	0582
data declaration	03700
(vect) (nls,const,)	0959
data declaration	04077
(verbsm) (nls,const,)	01018
data declaration	04136
(verga) (nls,cident,verga) 14G	02637
(stid,tstproc,contonerr,identstid,useckident)	06356
support routine for 'groups' or 'organizations' branch	06355
(vergroups) (nls,cident,vergroups) 14E	02594
(fileno,contonerr,identstid,useckident)	06275
verify 'groups' branch	06274
(veridfile) (nls,cident,veridfile) 14A	02644
(fileno,which,contonerr,idstr)	06368
verify Master Ident-file	06367
(verinds) (nls,cident,verinds) 14C	02557
(fileno,contonerr,identstid)	06204
verify 'individuals' branch	06203
(verorgs) (nls,cident,verorgs) 14F	02599
(fileno,contonerr,identstid,useckident)	06283
verify 'organizations' branch	06282
(verusedids) (nls,cident,verusedids) 14B	02551
(fileno,contonerr,identsr)	06192
verify 'usedids' branch	06191
(vexper) (nls,const,)	0616
data declaration	03734
(vfmmain) (nls,verify,vfmmain) 4A	02226
(fileno,vrfy)	05794
values of vrfy	05793
(vfullp) (nls,const,)	0854
data declaration	03972
(vinc0) (nls,frontend,)	02014
data declaration	05424
(vinc1) (nls,frontend,)	02004
data declaration	05411
(vinc2) (nls,frontend,)	01967
data declaration	05354
(vinc3) (nls,frontend,)	01999
data declaration	05403
(visibl) (nls,const,)	0894
data declaration	04012
(visv) (nls,const,)	0897
data declaration	04015
(vmax) (nls,const,)	0892
data declaration	04010

(vmultc) (nls,const,)	0584
data declaration	03702
(vocab) (nls,parser,vocab) 6B1	0133
(index,astr)	02991
returns vocabulary string, given index and address of astring	02990
(vrprvacc) (nls,ioexec,vrprvacc) 14C	02174
(fileno,filename,catstid,ident)	05695
verify ident's privacy-status access to file	05694
(vs) (nls,pdata,)	05
data declaration	02866
(vs1) (nls,parser,)	0202
data declaration	03067
(vs2) (nls,parser,)	0201
data declaration	03066
(vsaf1f) (nls,records,)	01403
data declaration	04521
(vsblkf) (nls,records,)	01415
data declaration	04533
(vsbrof) (nls,records,)	01297
data declaration	04415
(vscaco) (nls,parser,)	0181
data declaration	03042
(vscakf) (nls,records,)	01163
data declaration	04281
(vscapf) (nls,records,)	01172
data declaration	04290
(vsdaft) (nls,records,)	01459
data declaration	04577
(vsfrzf) (nls,records,)	01448
data declaration	04566
(vsidtf) (nls,records,)	01423
data declaration	04541
(vsindf) (nls,records,)	01319
data declaration	04437
(vslev) (nls,records,)	01179
data declaration	04297
(vslevd) (nls,records,)	01103
data declaration	04221
(vsmkrf) (nls,records,)	01345
data declaration	04463
(vsnamf) (nls,records,)	01508
data declaration	04626
(vspagf) (nls,records,)	01222
data declaration	04340
(vsplit) (nls,frontend,vsplit) 11A	02006
(left,cords1,cords2)	05415
split window vertically	05414
(vsplx1f) (nls,records,)	01430
data declaration	04548
(vsrlev) (nls,records,)	01209
data declaration	04327
(vssidf) (nls,records,)	01286
data declaration	04404
(vsstnf) (nls,records,)	01067
data declaration	04185
(vsstnr) (nls,records,)	01536

data declaration	04654
(vstar) (nls,records,)	01150
data declaration	04268
(vstr) (nls,const,)	0571
data declaration	03689
(vstrnc) (nls,records,)	01245
data declaration	04363
(vsusc) (nls,parser,)	0198
data declaration	03062
(vsuscf) (nls,records,)	01397
data declaration	04515
(vterse) (nls,const,)	0636
data declaration	03754
(vverbs) (nls,const,)	0756
data declaration	03874
(wacc) (nls,const,)	0918
data declaration	04036
(wdr) (nls,select,wdr) 1C5	0222
(bug,ptr1,ptr2)	03097
This procedure delimits a "WORD" given a pointer to a character in	
a string. The pointers PTR1 and PTR2 are set to the first and last	
characters, respectively, of the word.	
(werr) (nls,auxcod,werr) 4F	03096
(errno)	02272
(werrsi) (nls,const,)	05856
data declaration	0682
(window) (nls,const,)	04000
data declaration	0810
(word) (nls,const,)	03928
data declaration	01019
(wordls) (nls,const,)	04137
data declaration	0672
(wordrec) (nls,ndat,wordrec) 11B7A	03790
(rd)	02364
(wordty) (nls,const,)	05968
data declaration	0770
(wordv) (nls,const,)	03888
data declaration	0694
(wpiabt) (nls,filmnp,wpiabt) 2L3	03812
(write) (nls,const,)	01844
data declaration	0997
(writeout) (nls,ioexec,writeout) 15A	04115
(fileno,astrng)	02180
write out all pages of a file and PC; all pages associated with the	
file are removed from the user's map--written back onto the file.	
Returns TRUE is successful, FALSE if error with message in string	
whose address, is passed. Similar to clrcor. collect common code.	
(wrpi) (nls,filmnp,wrpi) 2L2	05706
write pseudo interrupt routine	01829
(wrtpsi) (nls,filmnp,wrtpsi) 2L1	05100
save the accumulators	01789
(wrtstr) (nls,dspgen,wrtstr) 4C	05023
(da,ls)	01898
write string in display area	05226
(wstr) (nls,const,)	05225
	0907

data declaration	04025
(xaccept) (nls,psedit,xaccept) 3A1	0316
(result,parsemode,console)	03260
Execute Accept Connection Command	03279
(xacs) (nls,intnls,)	02820
data declaration	06643
(xappend) (nls,psedit,xappend) 3B1	0299
(result,parsemode,sourcentity,source,destentity,destination,literal)	03246
Execute Append Command	03245
(xarchive) (nls,psedit,xarchive) 3C1	0273
(result,parsemode,filename,parameters)	03195
Execute Archive Command	03194
(xbackp) (nls,nddt,xbackp) 10H2	02452
(xbpclear) (nls,nddt,xbpclear) 10B12	02464
Clear a breakpoint	06088
(xbpprint) (nls,nddt,xbpprint) 10B13	02469
Print breakpoint	06072
(xbpset) (nls,nddt,xbpset) 10B14	02367
This is the procedure for setting breakpoints	05991
(xbreak) (nls,psedit,xbreak) 3D1	0314
(result,parsemode,entity,destination,level)	03276
Execute Break Command	03275
(xbrkpt) (nls,nddt,xbrkpt) 10B15	02426
This the breakpoint control routine	06031
(xbslash) (nls,psedit,xbslash) 3AW1	0286
(resultptr,parsemode)	03221
interpreter parsing mode	03220
(xbytsz) (nls,nddt,)	02379
data declaration	05984
(xc1) (nls,records,)	01365
data declaration	04483
(xc2) (nls,records,)	01362
data declaration	04480
(xcalcinit) (nls,pscalc,xcalcinit) 3A1	0447
(resultptr,parse)	03516
(xcallp) (nls,nddt,xcallp) 10H3	02375
(pread,frm,fp)	05979
Fake a call to pread	05978
(xcalprc) (nls,nddt,xcalprc) 10H4	02366
(xcarith) (nls,pscalc,xcarith) 3C1	0440
(result,parsemode,operation,numptr)	03506
(xceval) (nls,pscalc,xceval) 3F1	0448
(result,parsemode,param)	03516
Calculate value of an expression	03517
(xcevend) (nls,pscalc,xcevend) 3F2	0453
(result,parsemode,operation)	03527
operate on accum use temp accum, vaccum, as operand	03526
(xcfeedb) (nls,pscalc,xcfeedb) 3G7	0462
(result,parsemode,param)	03538
set flag to abbreviate feedback in TNLS	03537
(xchange) (nls,pshelp,xchange) 3H	0475
(resultptr,parsemode,basename)	03556
basename is a pointer to a string containing the name of the data	
base (a file name for now, though later it may be a number.	03555
(xcinsac) (nls,pscalc,xcinsac) 3N1	0467

(result,parsemode)		03545
Insert or replace		03544
(xcjust) (nls,pscalc,xcjust) 3G2		0443
(result,parsemode,param)		03512
right or left justify number		03511
(xclear) (nls,frontend,xclear) 3E1		0326
(result,parsemode>window)		03300
Execute Clear Command		03299
(xclraccum) (nls,pscalc,xclraccum) 3D1		0457
(resultptr,parsemode)		03531
(xclrfil) (nls,pscalc,xclrfil) 3E1		0455
(resultptr,parsemode)		03529
(xcmgrp) (nls,psedit,xcmgrp) 3G2C		0302
(destination,level,source,type,filterflag,vs)		03252
copy or move group		03251
(xcmst) (nls,psedit,xcmst) 3G2B		0301
(destination,level,source,type,filterflag,vs)		03250
copy or move statement		03249
(xcomma) (nls,pscalc,xcomma) 3G3		0441
(result,parsemode,param)		03508
set flag to insert commas in formatted number		03507
(xcomment) (nls,frontend,xcomment) 3A1		0272
(resultptr,parsemode)		03193
interpreter parsing mode		03192
(xconfirm) (nls,select,xconfirm) 1B5A		0229
(resultptr,parsemode,lastopcode)		03110
opcode of previous interpreter inst.		03109
(xconnect) (nls,frontend,xconnect) 3F1		0311
(result,parsemode,entity,destination,parameters)		03270
Execute Connect Command		03269
(xconterm) (nls,frontend,xconterm) 3F2		0261
(console,incut)		03171
Connect TTY Command		03170
(xcontin) (nls,nudt,xcontin) 10G1		02419
(xcopy) (nls,psedit,xcopy) 3G1		0282
(result,parsemode,sourcentity,source,destentity,destination,level,filterflag,vs)		03213
Execute Copy Command		03212
(xcord) (nls,records,)		01404
data declaration		04522
(xcquit) (nls,pscalc,xcquit) 3P1		0449
(resultptr,parsemode)		03520
calculator TERMINATION CODE		03519
(xcreate) (nls,psedit,xcreate) 3H1		0306
(result,parsemode,filename)		03260
Execute Create Command		03259
(xcreenter) (nls,pscalc,xcreenter) 3O1		0452
(resultptr,parsemode)		03525
calc reenter		03524
(xcshoaccums) (nls,pscalc,xcshoaccums) 3H1		0470
(resultptr,parsemode)		03549
show accumulators		03548
(xcshofil) (nls,pscalc,xcshofil) 3I1		0458
(resultptr,parsemode)		03532
(xctotl) (nls,pscalc,xctotl) 3J1		0446
(resultptr,parsemode)		03515

(xcuseaccum) (nls,pscalc,xcuseaccum) 3K1	0464
(resultptr,parsemode,numptr)	03540
(xcusesaved) (nls,pscalc,xcusesaved) 3L1	0471
(resultptr,parsemode)	03550
(xcutil) (nls,pssystem,xcutil) 5F	0385
(resultptr,pmode,umode)	03418
flags to be passed to nlsutility	03417
(xcwritef) (nls,pscalc,xcwritef) 3M1	0442
(resultptr,parsemode,fnamptr)	03510
calculator write file	03509
(xdddef) (nls,nddt,xdddef) 10D1	02388
this routine parses and processes the DEFINE symbol command	05992
(xddfind) (nls,nddt,xddfind) 10E1	02488
this routine parses and processes the FIND command	06071
(xddmark) (nls,nddt,xddmark) 10G3	02431
this routine parses and processes the MARK SYMBOL TABLE command	06034
(xddpop) (nls,nddt,xddpop) 10G4	02378
this routine parses and processes the POP SYMBOL TABLE command	05983
(xdelete) (nls,psedit,xdelete) 3I1	0319
(result,parsemode,entity,destination,filterflag,vs)	03286
Execute Delete Command	03285
(xdiropt) (nls,psedit,xdiropt) 3G2A	0278
(source,dent,frfile,info,gropk,sortk,rhostn,deffil)	03205
parse input for directory commands	03204
(xdisconnect) (nls,frontend,xdisconnect) 3J1	0300
(result,parsemode)	03248
Execute Disconnect Command	03247
(xdlpnt) (nls,psedit,xdlpnt) 3AF3A	0263
(function,astr)	03175
print a file for show directory	03174
(xdollar) (nls,pscalc,xdollar) 3G5	0465
(result,parsemode,param)	03542
set flag to insert dollar sign in formatted number	03541
(xedit) (nls,psedit,xedit) 3K1	0276
(result,parsemode,destination)	03201
Execute Edit Command	03200
(xent) (nls,pshelp,xent) 3C	0480
(resultptr,parsemode)	03564
(xexpunge) (nls,psedit,xexpunge) 3L1	0328
(result,parsemode,entity)	03304
Execute Expunge Command	03303
(xfdigits) (nls,pscalc,xfdigits) 3G1	0461
(result,parsemode,param,value)	03536
set number of digits after the decimal	03535
(xfeedback) (nls,parser,xfeedback) 5D1	0194
(resultptr,parsemode,fbmode,fbstring)	03057
pointer to feedback string	03056
(xfreeze) (nls,frontend,xfreeze) 3M1	0264
(result,parsemode,destination,vs)	03177
Execute Freeze Command	03176
(xfresta) (nls,frontend,xfresta) 7C	01974
(dpa,ssid,vspecl,vspec2)	05366
Given the address of a display are, an ssid, and two	05365
(xgo) (nls,nddt,xgo) 10F1	02485

(xgoexec) (nls, pssystem, xgoexec) 4A	0395
goto exec command	03435
(xgoto) (nls, pssystem, xgoto) 3A	0390
(resultptr, parsemode, subsysptr, dispatchptr, exflagptr)	03426
ptr to EXECUTE flag record	03425
(xhlpinit) (nls, pshep, xhlpinit) 3A	0474
(resultptr, parsemode, helpmode)	03554
(xhquit) (nls, pshep, xhquit) 3B	0477
(resultptr, parsemode)	03559
(xiaddmem) (nls, psident, xiaddmem) 18	0363
(resultptr, parsemode, fieldname, idlist)	03374
add members to membership list	03373
(xicloidfil) (nls, psident, xicloidfil) 4	0380
(resultptr, parsemode)	03408
close Master Ident-File file	03407
(xidelete) (nls, psident, xidelete) 20	0377
(resultptr, parsemode, recfield, which)	03402
delete members to membership list	03401
(xidelmem) (nls, psident, xidelmem) 19	0378
(resultptr, parsemode, fieldname, idlist)	03404
delete members to membership list	03403
(xiloadrecord) (nls, psident, xiloadrecord) 23	0374
(resultptr, parsemode, param)	03396
load an old record	03395
(xiloaidfil) (nls, psident, xiloaidfil) 3	0376
(resultptr, parsemode)	03400
load Master Ident-File file	03399
(xinsert) (nls, psedit, xinsert) 3N1	0327
(result, parsemode, entity, destination, level, parameter)	03302
Execute Insert Command	03301
(xinsstatement) (nls, psedit, xinsstatement) 3N2	0318
(result, parsemode, level, source)	03284
Execute repeat Insert Statement	03283
(xipassword) (nls, psident, xipassword) 22	0379
(resultptr, parsemode, param)	03406
check ident password	03405
(xistatus) (nls, psident, xistatus) 6	0383
(resultptr, parsemode, identptr, fieldname)	03414
show status of record	03413
(xiupdate) (nls, psident, xiupdate) 5	0371
(resultptr, parsemode, now)	03390
update modified record	03389
(xiverfil) (nls, psident, xiverfil) 21	0381
(resultptr, parsemode, what, errorstop)	03410
verify the contents of the Master Ident-File	03409
(xjcloworfil) (nls, pssendmail, xjcloworfil) 20	0341
(resultptr, parsemode)	03330
close work file	03329
(xjdae) (nls, frontend, xjdae) 4A1	0305
(resultptr, pmode, sourceptr)	03258
ptr to the bug select record	03257
(xjdoit) (nls, pssendmail, xjdoit) 2G	0336
(resultptr, parsemode)	03320
finish submission or forward request	03319
(xjforward) (nls, pssendmail, xjforward) 2M	0338
(resultptr, parsemode, jnumber, ids)	03324

Forward J doc to new ids		03323
(xjinsstatus) (nls,psendmail,xjinsstatus)	2K	0333
(resultptr,parsemode,destination,level)		03314
Insert Status of submission form		03313
(xjloaworfil) (nls,psendmail,xjloaworfil)	2B	0334
(resultptr,parsemode)		03316
load Journal work file		03315
(xjlock) (nls,psendmail,xjlock)	2H	0343
(resultptr,parsemode,password,lockflag)		03334
lock or unlock the journal		03333
(xjmpcntdisp) (nls,frontend,xjmpcntdisp)	4B3	0288
(resultptr,parsemode)		03225
parsing mode		03224
(xjoutil) (nls,pssystem,xjoutil)	50	0392
(resultptr,pmode)		03430
parsing mode		03429
(xjpriharcop) (nls,psendmail,xjpriharcop)	2I	0339
(resultptr,parsemode)		03326
print hard copy		03325
(xjprocess) (nls,psendmail,xjprocess)	2J	0342
(resultptr,parsemode,destination)		03332
process command form		03331
(xjreserve) (nls,psendmail,xjreserve)	2D	0340
(resultptr,parsemode,type,numo,insertloc)		03328
reserve numbers		03327
(xjrforcreserve) (nls,psendmail,xjrforcreserve)	2E	0345
(resultptr,parsemode,titlestr,authstr,sendto,onlineflag,insertloc)		03338
reserve an RFC and a Catalog numstr		03337
(xjrforcsho) (nls,psendmail,xjrforcsho)	2F	0346
(resultptr,parsemode,titlestr,authstr,sendto,onlineflag)		03340
show status of RFC reserve request		03339
(xjstatus) (nls,psendmail,xjstatus)	2L	0344
(resultptr,parsemode)		03336
Show Status of submission request		03335
(xjump) (nls,psedit,xjump)	30I	0324
(result,parsemode,entity,destination,vs)		03296
Execute Jump Command		03295
(xjzapworfil) (nls,psendmail,xjzapworfil)	2A	0335
(resultptr,parsemode)		03318
initialize work file		03317
(xll0na) (nls,const,)		01029
data declaration		04147
(xlevadj) (nls,select,xlevadj)	1B3A	0217
(resultptr,parsemode)		03089
parsing mode		03088
(xlinefeed) (nls,psedit,xlinefeed)	3AS1	0274
(resultptr,parsemode)		03197
interpreter parsing mode		03196
(xlist) (nls,pshelp,xlist)	3F	0478
(resultptr,parsemode)		03561
A dummy for now		03560
(xload) (nls,psedit,xload)	3P1	0287
(result,parsemode,entity,filename)		03223
Execute Load Command		03222
(xlogout) (nls,psedit,xlogout)	3Q1	0285

(result,parsemode)	03179
Execute Logout Command	03178
(xmark) (nls,psedit,xmark) 3R1	0284
(result,parsemode,destination,mkrname)	03217
Execute Mark Command	03216
(xmerge) (nls,psedit,xmerge) 3S1	0297
(result,parsemode,entity,source,destination)	03242
Execute Merge Command	03241
(xmkprint) (nls,nddt,xmkprint) 10G2	02473
this routine parses and progresses the MARK SYMBOL TABLE PRINT command	06076
(xmove) (nls,psedit,xmove) 3T1	0329
(result,parsemode,sourcentity,source,destentity,destination,level,fi lterflag,vs)	03306
Execute Move Command	03305
(xnbyte) (nls,nddt,)	02360
data declaration	05963
(xnmark) (nls,nddt,xnmark) 10G1	02451
This the mark symbol table control routine	06055
(xnmdlset) (nls,auxcod,xnmdlset) 10B	02307
(stidl,stad2,dleft,dright,da)	05907
Execute set name deilimiters for a group	05906
(xnshow) (nls,nddt,xnshow) 10J2	02455
(oldrd,echflg)	06059
(xopnam) (nls,const,)	0875
data declaration	03993
(xout1) (nls,psedit,xout1) 3U2	0321
(result,parsemode,entity,filename,parameter)	03290
Execute Output (quickprint, journa, printer, com) Command	03289
(xout2) (nls,psedit,xout2) 3U3	0308
(result,parsemode,entity,tip,tipport,formfeed,simif,waitpb)	03264
Output (Terminal, Remote printer) Command	03263
(xoutsapf) (nls,psedit,xoutsapf) 3U1	0289
(result,parsemode,parameter)	03227
set odapfg for output commands	03226
(xpattach) (nls,psprogs,xpattach) 3A	0361
(resultptr,parsemode,subname)	03370
attach subsystem to NLS	03369
(xpcompile) (nls,psprogs,xpcompile) 3B	0360
(resultptr,parsemode,entity,location,compiler,fname)	03368
compile program	03367
(xpdeinstitute) (nls,psprogs,xpdeinstitute) 3D	0348
(resultptr,parsemode,type)	03344
Deinstitute a user program	03343
(xpdelete) (nls,psprogs,xpdelete) 3C	0353
(resultptr,parsemode,howmany)	03354
Delete a user program from stack	03353
(xpdetach) (nls,psprogs,xpdetach) 3E	0357
(resultptr,parsemode,subname)	03362
detach subsystem to NLS	03361
(xperiod) (nls,psedit,xperiod) 3AU1	0310
(resultptr,parsemode)	03268
interpreter parsing mode	03267
(xp institute) (nls,psprogs,xp institute) 3F	0350
(resultptr,parsemode,pgmnam,type)	03348
Institute a user program	03347

(xpcill) (nls,psprogs,xpcill) 3G	0352
(resultptr,parsemode)	03352
kill tenex subsystem	03351
(xplayback) (nls,psedit,xplayback) 3V1	0277
(result,parsemode,filename)	03203
Execute Playback Command	03202
(xpload) (nls,psprogs,xpload) 3H	0358
(resultptr,parsemode,filnm)	03364
load a user program	03363
(xpop) (nls,parser,xpop) 5B2	0131
pops the address of an argument record from the eval stack	02988
(xpreplace) (nls,psprogs,xpreplace) 3J	0359
(resultptr,parsemode,oldrule,filename,newrule)	03366
Replace Parserule command	03365
(xpreset) (nls,psprogs,xpreset) 3K	0351
(resultptr,parsemode,entity,rulename,fname)	03350
Reset command	03349
(xprint) (nls,psedit,xprint) 3W1	0298
(result,parsemode,entity,destination,vs)	03244
Execute TNLS Print Command	03243
(xpriority) (nls,psystem,xpriority) 5J	0391
(resultptr,parsemode,priorval)	03428
set priority only if person can enable	03427
(xprocc) (nls,nddt,xprocc) 10H5	02444
(xprocess) (nls,psedit,xprocess) 3Y1	0280
(result,parsemode,entity,destentity,destination)	03209
Execute Process Command	03208
(xprotect) (nls,psedit,xprotect) 3X1	0295
(result,parsemode,filename,params)	03238
Execute Protect Command	03237
(xprun) (nls,psprogs,xprun) 3I	0355
(resultptr,parsemode,pgmnam)	03358
Run a user program	03357
(xprunt) (nls,psprogs,xprunt) 3L	0349
(resultptr,parsemode,subnam,outfil,inmode,intext)	03346
Run tenex subsystem	03345
(xpset) (nls,psprogs,xpset) 3M	0347
(resultptr,parsemode,entity,param)	03342
Set command	03341
(xpshow) (nls,psprogs,xpshow) 3N	0356
(resultptr,parsemode)	03360
Show Status of user programs display	03359
(xpshtn) (nls,psprogs,xpshtn) 3O	0354
(resultptr,parsemode)	03356
Show Tenex subsystem Status display	03355
(xptadd) (nls,nddt,)	02443
data declaration	06045
(xptr) (nls,nddt,)	02355
data declaration	05959
(xpush) (nls,parser,xpush) 5B1	0208
(addrofvalue)	03075
address of value	03074
(xqmore) (nls,pshelp,xqmore) 3I	0476
(resultptr,parsemode)	03558

Get here if more command typed. Only valid if qsw sequence work area is non-zero; if zero, display message about last menu having

```

been complete. Otherwise, call qdisp in continue mode.
(xqshow) (nls,pshelp,xqshow) 3E
    (resultptr,parsemode,listptr)
(xquit) (nls,pssystem,xquit) 3B
    (resultptr,parsemode,subsysptr)
    ptr to subsys name record
(xread) (nls,parser=@xread) 5Bh
    Reads the address of an argument record from the eval stack
(xrecord) (nls,psedit,xrecord) 3Z1
    (result,parsemode,filename)
    Execute Record Command
(xrecplasup) (nls,frontend,xrecplasup) 9B
    (record,fl,f2,rhostn)
    record or playback from file
(xrelallsta) (nls,frontend,xrelallsta) 7B
    (dpa)
    Given the address of a display area, this routine will
(xrelease) (nls,frontend,xrelease) 3A@1
    (result,parsemode,entity,destination)
    Execute Release Command
(xrelsta) (nls,frontend,xrelsta) 7A
    (dpa,stad)
    This routine searches the chain of frozen statements.
(xrenumber) (nls,psedit,xrenumber) 3AA1
    (result,parsemode)
    Execute Renumber Command
(xreplace) (nls,psedit,xreplace) 3AB1
    (result,parsemode,destentity,destination,sourcentity,source)
    Execute Replace Command
(xreplp) (nls,nddt,xreplp) 10H6
(xreset) (nls,psedit,xreset) 3AO1
    (result,parsemode,entity,destentity,destination)
    Execute Reset Command
(xretrieve) (nls,psedit,xretrieve) 3AD1
    (result,parsemode,filename)
    Execute Retrieve Command
(xringjump) (nls,frontend,xringjump) 4B1
    (resultptr,parsemode,entity)
    entity type for the jump
(xrpsset) (nls,nddt,xrpsset) 10I1
    Set up the record pointer RECP
(xsearch) (nls,pshelp,xsearch) 3D
    (resultptr,parsemode,value)
    Search types:
(xselect) (nls,select,xselect) 1B1A
    (resultptr,parsemode,type,curptr)
    ptr to path stack entry
(xset) (nls,psedit,xset) 3AE1
    (result,parsemode,entity,param1,param2,param3,destination)
    Execute Set Command
(xsetjou) (nls,pssystem,xsetjou) 5G
    (resultptr,pmode,option)
    pointer to option specified
(xsetld) (nls,pssystem,xsetld) 5I
    (resultptr,pmode,param)
    pointer to selection record

```

```

03557
0472
03551
0399
03443
03442
0138
02996
0309
03266
03265
02001
05407
05406
01986
05382
05381
0285
03219
03218
02005
05413
05412
0304
03256
03255
0331
03310
03309
02430
0317
03282
03281
0323
03294
03293
0303
03254
03253
02372
05975
0479
03563
03562
0235
03121
03120
0267
03183
03182
0398
03441
03440
0397
03439
03438

```

(xsetpar) (nls,pssystem,xsetpar) 5H	0388
(resultptr,pmode,entity,option)	03423
pointer to option specified	03422
(xshoviespe) (nls,psedit,xshoviespe) 3AF2	0294
(da,string)	03236
(xshow) (nls,psedit,xshow) 3AF1	0330
(result,parsemode,entity,param,dopt,dfile)	03308
Execute show Command	03307
(xsimdev) (nls,frontend,xsimdev) 8A	02026
(da,devicetype,modetype)	05445
Execute simulate Device Command	05444
(xsimulate) (nls,frontend,xsimulate) 3AG1	0268
(result,parsemode,devicetype)	03185
Execute Simulate Command	03184
(xslash) (nls,psedit,xslash) 3AV1	0271
(resultptr,parsemode)	03191
interpreter parsing mode	03190
(xsndtmsg) (nls,inpfbk,xsndtmsg) 7A	0529
(messtring,userst)	03636
given a message and user directory name, xsndtmsg finds all the	
places that user is currently logged in and attached and sends the	
message. The job MUST BE enabled for this to work. Otherwise you	
get an illegal instruction trap. It returns TRUE/FALSE.	03635
(xsort) (nls,psedit,xsort) 3AK1	0291
(result,parsemode,entity,destination)	03231
Execute Sort Command	03230
(xsplit) (nls,psedit,xsplit) 3A11	0283
(result,parsemode,parameter)	03215
Execute Split Command	03214
(xstop) (nls,psedit,xstop) 3AJ1	0322
(result,parsemode)	03292
Execute Stop Command	03291
(xstor) (nls,parser,xstor) 5D2	0104
(resultptr,parsemode,location)	02956
ptr to store location	02955
(xstore) (nls,parser,xstore) 5B3	0196
(addressofvalue)	03060
address of value	03059
(xstr) (nls,const,)	0774
data declaration	03892
(xsubcurrent) (nls,pssystem,xsubcurrent) 3D	0396
(resultptr,parsemode)	03437
parsing mode	03436
(xsublist) (nls,pssystem,xsublist) 3C	0393
(resultptr,parsemode)	03432
parsing mode	03431
(xsubnotimp) (nls,pssystem,xsubnotimp) 6	0394
(resultptr,parsemode)	03434
subsystem not yet implemented	03433
(xsubstitute) (nls,psedit,xsubstitute) 3AK1	0266
(result,parsemode,textentity,structentity,destination,pairs,filterfl	
ag.vs)	03181
Execute substitute Command	03180
(xtab) (nls,psedit,xtab) 3AQ1	0293
(resultptr,parsemode)	03235
interpreter parsing mode	03234

(xtop) (nls,pshelp,xtop) 3G	0473
(resultptr,parsemode)	03553
A dummy for now	03552
(xtrace) (nls,nddt,xtrace) 1OK1	02405
(xtranspose) (nls,psedit,xtranspose) 3A11	0290
(result,parsemode,sourcentity,source,destentity,destination,filterfl ag.vs)	03229
Execute Transpose Command	03228
(xtrim) (nls,psedit,xtrim) 3A11	0315
(result,parsemode,parameter)	03278
Execute Trim Command	03277
(xtrnam) (nls,utility,xtrnam) 7F	01624
(ast.worka,dleft,dright)	04746
Arguments are 1: address of A-string, 2: address of work area for READC, 3: left name delimiter, 4: right name delimiter. If instead of passing the delimiters you want xtrnam to use the delimiters given in the SDB header, pass a minus one as the third argument. If finds a name in the statement it is placed in the A-string, otherwise the A-string is set to empty. Uses (TXTEXT, nmdr) to determine the name.	
(xtval) (nls,nddt,xtval) 1OLL	02358
(xundelete) (nls,psedit,xundelete) 3A11	0320
(result,parsemode,entity,filename)	03288
Execute Undelete Command	03287
(xuocontchars) (nls,psuserop,xuocontchars) 2A4	0438
(resultptr,parsemode,dev,cc,char,echo)	03505
set control characters for specified device	03504
(xuoconcur) (nls,psuserop,xuoconcur) 2A5	0419
(resultptr,parsemode,numbr)	03474
set current context length	03473
(xuofeedback) (nls,psuserop,xuofeedback) 2A6	0433
(resultptr,parsemode,type,param)	03496
set feedback mode, etc	03495
(xuohearauld) (nls,psuserop,xuohearauld) 2A7	0421
(resultptr,parsemode,type,param)	03477
set herald	03476
(xuoinit) (nls,psuserop,xuoinit) 2A2	0425
(resultptr,parsemode)	03483
initialize user-options subsystem	03482
(xuolevel) (nls,psuserop,xuolevel) 2A8	0432
(resultptr,parsemode,param)	03494
set ask for level adjust ON/OFF	03493
(xuoprint) (nls,psuserop,xuoprint) 2A10	0408
(resultptr,parsemode,type,param)	03458
set print parameters	03457
(xuoiprompt) (nls,psuserop,xuoiprompt) 2A11	0428
(resultptr,parsemode,type)	03488
set prompt mode	03487
(xuoirecognition) (nls,psuserop,xuoirecognition) 2A12	0436
(resultptr,parsemode,param,param2)	03501
set recognition mode	03500
(xuoireset) (nls,psuserop,xuoireset) 2A1k	0414
(resultptr,parsemode,param1,param2)	03467
reset user options	03466
(xuoiringssize) (nls,psuserop,xuoiringssize) 2A9	0423
(resultptr,parsemode,type,param)	03480

set return ring size		03479
(xuoshow) (nls,psuserop,xuoshow) 2A15		0401
(resultptr,parsemode,param1,param2)		03447
show user options		03446
(xuoterm) (nls,psuserop,xuoterm) 2A3		0431
(resultptr,parsemode)		03492
terminate user-options subsystem		03491
(xuvviewspecs) (nls,psuserop,xuvviewspecs) 2A13		0406
(resultptr,parsemode,param,vs)		03455
set standard viewspecs		03454
(xuparrow) (nls,psedit,xuparrow) 3A11		0281
(resultptr,parsemode)		03211
interpreter parsing mode		03210
(xupdate) (nls,psedit,xupdate) 3A01		0313
(result,parsemode,entity,filename)		03274
Execute Update Command		03273
(xupidfil) (nls,cident,xupidfil) 12B		02654
(idstr,infostr,fileno)		06386
support routine for UPIDFIL		06385
(xutilinit) (nls,psystem,xutilinit) 5B		0386
(resultptr,pmode)		03420
parsing mode		03419
(xutilty) (nls,utility,xutilty) 8F		01641
a place for several "fast" routines to reside. NOTE: These		
routines cannot use locals or make calls since they are labels:		
they avoid the stack setup instructions at procedure entry. They		
are called, however, as procedures with arguments.		04766
(xverga) (nls,cident,xverga) 14H		02556
(stid,tstproc,contonerr,useckident)		06202
support routine for verga		06201
(xverify) (nls,psedit,xverify) 3AP1		0332
(result,parsemode)		03312
Execute Verify Command		03311
(xviewspecs) (nls,select,xviewspecs) 1B4A		0237
(resultptr,parsemode)		03125
parsing mode		03124
(xvinds) (nls,cident,xvinds) 14D		02566
(stid,uplname,contonerr)		06221
auxiliary routine for verinds		06220
(xxreadc) (nls,utility,xxreadc) 7E		01584
This is the routine that is called to read a character by the READC		
construct. Code to call is JSP a4,readc so return loc is in A4.		
(xyds) (nls,records,)		04702
data declaration		01473
(ycl) (nls,records,)		04591
data declaration		01212
(yc2) (nls,records,)		04330
data declaration		01407
(ycord) (nls,records,)		04525
data declaration		01379
(ystr) (nls,const,)		04497
data declaration		0934
(zinsst) (nlslan,zinsst) **Couldn't open <NLS>nlslan<↑F>NLS**		04052
(zstr) (nls,const,)		096
data declaration		0927
		04045

