

L10 DATA

```

< NLS, L10DATA.NLS;3, >, 29-OCT-76 08:09 KJM ;;;
FILE l10data % L10 <REL-NLS>L10DATA %% (L10,)
(rei-nls,L10DATA.rel,) %

REGISTER %here so DDT will use these on printout%
    r1=1, r2=2, r3=3, r4=4, r5=5, p=7, wa=8, s=9, m=10, rp=11,
    a1=12, a2=13, a3=14, a4=15;
% *** GLOBAL DATA FOR PAGE 0 *** %
% Special stuff declared in page 0, loaded at 140B %
    SET EXTERNAL
        gstksz= 3072, %length of general call stack%
        pstksz= 20;%length of pattern stack%
%...KEEP THESE THINGS TOGETHER -- DONT ADD NEW DATA HERE...%
    DECLARE EXTERNAL %...important stuff...%
    %...user related data...%
        cinit, % current initials of user %
    %...string analysis stuff...%
        flag= 1, %the general flag%
        pstack[pstksz], %pattern stack%
    %...editing global variables...%
        %...text editing global variables...%
            scndir, % analyzer compiler scan direction %
            rplsid, % psid of statement being replaced in sc %
            sptr1[2],
            sptr2[2], % for append T-string %
            nsdbpt, % byte pointer for statement being constructed%
    %...often used work areas...%
        swork, % string reading work area %
        swork1[6],
        stcwrk, % string construction work area %
        stcwr1[6],
    %...set command and aptstr routine...%
        modeset, % option set %
        modeshift, % case set %
        mkrstay, %true if do not move markers%
        clmpty; %reset value for clchng stack%
% accessed in statement construction%
    DECLARE EXTERNAL STACK
        clchng[30];
% general call stack%
    DECLARE EXTERNAL
        gstack[gstksz];
% DECLARE *s %
    DECLARE EXTERNAL STRING
        sar[2000]; % DO NOT USE -used by string primitives%
    DECLARE EXTERNAL
        %...general global variables...%
        sysgnl, %signal value%
        sysmsg, %signal message%
    %...other work areas...%
        ps,
        pe; %cells for BETWEEN construct%
%collector sorter%
    DECLARE EXTERNAL
        ingflg; %true if length is considered befor value in sort%
    DECLARE EXTERNAL STACK

```

02
03
04
05
06
07
037
039
038
08
09
049
050
010
011
012
013
014
034
015
016
067
017
041
042
043
044
045
018
053
054
040
019
020
021
068
022
023
069
024
026
027
028
029
030
031
032
033
070
046
047
048
051

```
    btwstk[10];                                052
%...spec stack...%                            055
  DECLARE EXTERNAL                            060
    spsk= spsk,    % spec stack pointer %     056
    spsk1= spsk,   % initial pointer %       057
    b1[2],        058
    b2[2],        074
    b3[2],        073
    b4[2],        072
    b5[2],        071
    spskt= b5;    % top of spec stack %       059
%...correspondence list...% %PASSED%         061
  DECLARE EXTERNAL                            066
    clhead[4],   %header, see clhdr record%   062
    clistb,      %address of allocated corres. list block% 075
    clistsz,     %current size of clist block% 076
    clstad=clhead; %address of clist header%  065
FINISH of l10data                             035
```

L10 RUNTIME

< NLS, L10RUNTIME.NLS;8, >, 23-DEC-76 12:26 JDH ;;;;

FILE l10runtime % L10 to <rel-nls>l10runtime %% (L10,) to (rel-nls,
L10RUNTIME,) %

02

% NOTES ABOUT RUNNING STANDALONE L10 PROGRAMS % 01527

% the following files should be loaded with your program to generate
a standalone l10 program: 01528

<rel-nls>l10data.rel - l10 data package 01529

<rel-nls>l10runtime - l10 runtime package 01530

<system>stenex - tenex jsys definitions 01531

the following will be undefined unless they are provided by the user: 01532

sysovr 01533

this is a lable that is jumped to when a stack overflow occurs 01534

to see how NLS handles this see <nls, utility, sysovr> 01535

uflow 01536

this is a procedure that gets called when a stack underflow
occurs 01537

to see how NLS handles this see <nls, auxcod, uflow> 01538

err 01539

this procedure is called when some procedure detects an error
condition 01540

to see how NLS handles this see <nls, utility, err> 01541

deferr 01542

this procedure is called when a signal has been propagated to
the top of the stack 01543

to see how NLS handles this see <nls, utility, deferr> 01544

filesc 01545

this procedure is called from procedure esc (end string
construction) when esc detects that the strings being dealt
with do not have the stastr bit set (i.e., they are not
astrings) 01546

if a user is not dealing with any strings other than literal,
local, or global strings (for example, strings within a data
file), then this procedure need not be supplied. 01548

to see how NLS handles this see <nls, utility, filesc> 01547

flcpfse 01549

this procedure is called from procedure cpfse when cpfse
detects that the strings being dealt with do not have the
stastr bit set (i.e., they are not astrings) 01550

if a user is not dealing with any strings other than literal,
local, or global strings (for example, strings within a data
file), then this procedure need not be supplied. 01551

to see how NLS handles this see <nls, utility, flcpfse> 01552

flfechcl 01553

this procedure is called from procedure fechcl when fechcl
detects that the strings being dealt with do not have the
stastr bit set (i.e., they are not astrings) 01554

if a user is not dealing with any strings other than literal,
local, or global strings (for example, strings within a data
file), then this procedure need not be supplied. 01555

to see how NLS handles this see <nls, utility, flfechcl> 01556

since l10 provides no way for specifying the starting point of a
program, it is the user's responsibility to do this. either of the
following methods can be used to accomplish this: 01557

after your load is done and before the program is saved, use the

```

EXEC ENTVEC command                                01558
after the load is done and before saving your program, go into ddt
and start up your program at a label that you have supplied that
will set up the entry vector and optionally do the save for you.
the instructions at this label cannot make use of any procedure
local symbols and can not issue any procedure calls (including any
implicit calls that might be dropped out by the compiler). this
is so because the L10 environment has not been established yet.
it is usually safe to accomplish this by using only assembly
language instructions.                                01559

when you are ready to actually run your program, it is necessary to
establish the L10 environment. this is best accomplished by having
the first thing your program does is to execute the following
assembly language instruction:                        01560
    !JSP 1,l10init                                  01561
no procedure calls (explicit or implicit) can be executed before the
L10 environment has been set up. when your program gets control back
from l10init, the environment will be established.  01562
                                                    01563

if you have problems or comments getting standalone L10 programs to
the point where the bugs are in your program and not associated with
the L10 environment, contact a systems programmer at SRI-ARC  01564
%                                                    01565
% DECLARARIGNTS %                                    065
REGISTER                                             053
    a1= 12, %working register%                       054
    a2= 13, %working register%                       01221
    a3= 14, %working register%                       055
    a4= 15, %working register%                       01222
    r1= 1,  %working register%                       056
    r2= 2,  %working register%                       057
    r3= 3,  %working register%                       058
    r4= 4,  %working register%                       059
    r5= 5,  %working register%                       060
    rp= 11, %record pointer%                         061
    p= 7,   %pattern stack%                          062
    wa= 8,  %pattern stack%                          01223
    s= 9,  %general call stack pointer%              063
    m= 10; %mark stack pointer%                     064
DECLARE EXTERNAL                                    01512
    errmsbase= 140B;                                01604
EXTERNAL l10init;                                  04
EXTERNAL readc;      %entry to READC routine%       0417
DECLARE %pop transfer table% poptab=(baduuo,       01324
    qcall, %1% %no argument call pop%              01325
    qcall1, %2% %single argument call pop%         01326
    qcallm, %3% %up to 13 argument call pop%       01327
    qcallm2, %4% %More than 13 argument call pop%  01328
    %5-37B are not defined %                        01329
    baduuo, baduuo, baduuo, baduuo, baduuo, baduuo, 01330
    baduuo, baduuo, baduuo, baduuo, baduuo, baduuo, 01331
    baduuo, baduuo, baduuo, baduuo, baduuo, baduuo, 01332
    baduuo, baduuo, baduuo, baduuo, baduuo, baduuo, 01333
    baduuo, baduuo, baduuo, baduuo);              01334
DECLARE EXTERNAL                                    01246
    empty= 0,                                       01247

```

```

    endfil= -1,                                01605
    origin= 2 %=fbhdl%,                        01248
    nullch= 0,                                 01607
    chbmty= 440700000001B;                    01606
DECLARE EXTERNAL                               01469
    sdbhdl=5;      %length of sdb header%      01468
DECLARE EXTERNAL                               01510
    forward= 1;                                01511
% RECORDS %                                    01437
    (byteptr)      % field definitions for PDP-10 byte pointer word %
                                                    01520
    RECORD                                                01521
        bpadr[18],                                       01522
        bpindx[6],                                       01610
        bpsize[6],                                       01609
        bpbitpos[6];
                                                    01608
(stidr) RECORD                                         01418
    stpsid[18],                                         01617
    stfile[8],                                         01618
    stastr[1];                                         01619
    % -- (filmnp, lodrng) and (filmnp, lodsdb) cheat and make use of
    the format of an stid -- so please don't change this without
    checking those routines and telling WHP %
                                                    01419
(stbw) RECORD                                         01420
    stwc[9],                                           01615
    stblk[9];                                          01616
(stsdb) RECORD                                         01421
    stpsdb[18];                                        01614
(stast) RECORD                                         01422
    stadr[18];                                        01613
(stsidr) RECORD                                       01423
    stsidf[6],                                         01611
    stsid[30];
                                                    01612
(clistr) RECORD %clist entry%                        01424
    clst1[36],    %original or updated stid%          01425
    clst2[36],    %successor stid%                   01426
    clcc1[12],    %character count for first%         01427
    clcc2[12],    %character count for second%        01428
    clcurmkr[1],  %for CSP use only, to set curmkr%   01629
    clfixed[1];  %for aptstr%
                                                    01429
DECLARE EXTERNAL                                     01430
    cll= 3;
                                                    01620
(clhdr) RECORD %clist header%                        01431
    clcnt[9],     %count of entries in use%           01432
    clfno1[9],    %first file number%                01433
    clfno2[9],    %second file number%               01434
    cltype[6],    %type used to generate clist%      01435
    clbuff[18];  %address of buffer%
                                                    01436
(ring) RECORD % *** ringl is length%                01566
    rsub[18],     %psid of sub of this statment%     01567
    rsuc[18],     %psid of suc of this statement%    01568

```

```

rsdb[18],      %psdb of sdb for this statement%      01569
rinst1[7],     %DEX interpolation string-- scratch space% 01570
rinst2[7],     %DEX interpolation string-- scratch space% 01571
rdummy[1],    %DEX dummy flag-- scratch space%      01572
repet[3],     %DEX repetition-- scratch space%      01573
rhf[1],       %head flag, true if this is head of plex% 01574
rtf[1],       %tail flag, true if tail of plex%     01575
rnamef[1],    %name flag, true if statement has a name% 01576
rtorgin[1],   %inferior tree origin flag, true if origin% 01577
rnull[1],    %unused%                               01578
rnameh[30],   %name hash for this statement%        01579
rsid[30];     %statement identifier%                01580
%although only need four words, use five so that have room
to grow%                                           01581

(sdbhead) RECORD % *** sdbhdl is length%           01582
sgarb[1],     %true if this sdb is garbage%         01584
slength[9],   %number of words in this sdb%        01585
schars[11],   %number of characters in this statement% 01586
slnmdl[7],    %left name delimiter for statement%   01587
srnmdl[7],    %right name delimiter for statement%  01588
spsid[18],    %psid of the statement for this sdb%  01589
sname[11],    %position of character after name%    01590
stime[36],    %date and time when this sdb created% 01591
sinit[21],    %initials of user who created this sdb% 01592
sptype[15],   %property type of this data block%   01593
%                                                    01594
txttyp = text data block (SDB)                    01595
dhdtyp = diagram header block                      01596
segtyp = segment data block                        01597
%                                                    01598
spsdb[18],    %PSDB of the next property data block 0=tail% 01599
sitpsid[18]; %PSID to head of inferior tree if any% 01600
%sgarb and slength must be in the first word of the header
for newsdb%                                       01601
%                                                    01602
%                                                    01603

(xl10init) PROCEDURE; %stack overflow/under flow code% 05
%accessed by a JSP r1, l10init instruction%        06
% initialize the pop transfer instruction%          07
(l10init): a1 _ krummy; !MOVEM a1,41B;              08
% initialize stack pointers%                        09
%general call stack%                               010
s _ -$gstksz; !HRL s,s; !HRR1 s,gstack;            011
m _ s;                                              012
gstack _ $uflow;                                    013
%pattern stack%                                     014
p _ -$pstksz; !HRL p,p; !HRR1 p,pstack;            015
% spec stack %                                       016
spsk _ spsk1;                                       017
% reset clist change stack %                          018
RESET clchng; % before any string constructions are done% 019
clmpty _ clchng;                                    020
GOTO [r1];                                          023
% instruction for pop transfer%                       021
(krummy): !JSP r5,ppaprt;                           022
END.                                                051

```


GAS2, 14-Feb-79 22:36

< NLS, L10RUNTIME.NLS.8, > 5

%%

066

```

% call and pop routines%                                067
EXTERNAL ppaprt, gcall, gcall1, gcallm;                0162
(sysrtn) PROCEDURE; %for returns%                      068
  %get here by a JRST sysrtn%                          069
  !MOVE s,m; %back up the stack%                       070
  !POP s,a4; %save the return location%                071
  !POP s,m; %restore the mark%                         072
  !JRST (a4) END.

                                                                    073
(sys2rt) PROCEDURE; %return with 2 results%            074
  !MOVE s,m;                                           075
  !POP s,a4;                                           076
  !POP s,m;                                           077
  !MOVEM a2,3(s);                                       078
  !JRST (a4) END.

                                                                    079
(sysmrt) PROCEDURE; %for returns with multiple results% 080
  % register a2 contains number-1 of multiple results% 081
  !MOVN a3,a2;                                         082
  !ADD a3,s;                                           083
  !HRL a3,a3; %a3 left points to first on stack%      084
  !HRR a3,m;                                           085
  !ADJ a3,; %a3 right points to destination%          086
  !HRRZ a4,a3;                                         087
  !ADD a4,a2; %a4 points to last loc moved to%        088
  !BLT a3,@a4;                                        089
  !JRST sysrtn END.

                                                                    090
(syssid)PROC;                                         091
  WHILE [m-1] LH = 0 DO                                092
    IF (m .A 18M) <= ([m-1] .A 18M) THEN              093
      GOTO sysovr                                     094
    ELSE                                              095
      IF [m_[m-1]] = $uflow THEN                     096
        deferr();                                     097
      %by here, m points to mark beyond one with signal loc% 098
      [m] _ [m-1] LH; %change return loc to signal code% 099
      RETURN;                                        0100
    END.

                                                                    0101
(repsig)PROC(value);                                  0102
  %Repeat a signal with value as sysgnl%              0103
  SIGNAL(value);                                      0104
  END.                                                0105
(callsig)PROC(value, message);                       0106
  %generate the specified signal with the spcified message. this
  routine is useful if the calling procedure wishes a chance to
  catch the signal itself -- would not be possible if the procedure
  generated the signal itself.%                      0107
  SIGNAL(value, message);                             0108
  END.                                                0109
DECLARE FIELD                                         0110
  popldb=[40B, 9:27],                                01622
  140ac=[40B, 4:23];                                 01623
(cli) PROCEDURE; %dummy procedure for call%           0111
  (ppaprt):                                          0112

```

```

%get here by a JSP r5,ppaprt in location 41B%      0113
% fast code to get UUD number: %                  0114
  !HLRZ r4,40B;                                    0115
  !ROT r4,-11B;                                    0116
  !MOVEI r5,0(r5); % left half zero %            0117
  !JRST @poptab(r4); % goto UUD %                0118
(baduu0): % illegal UUD (see poptab) %           0119
r1 _ $"Illegal UUD at " + 1;                       0120
!HRLI r1,7B2;                                     0121
!JSYS psout;                                       0122
r1 _ 101B;                                         0123
r2 _ (r5 - 1) .A 18M;                              0124
r3 _ 8;                                             0125
!JSYS nout;                                        0126
!JSYS haltf;                                       0127
!JSYS haltf;                                       0128
(qcall):                                           0129
  %get here by a CALL pop%                          0130
  !PUSH s,m; %save the mark%                       0131
  !PUSH s,r5; %save the return location%           0132
  !MOVE m,s; %new mark%                            0133
  !JRST @40B;                                       0134
(qcall1):                                          0135
  %get here by a CALL1 pop%                         0136
  !PUSH s,m; %save the mark%                       0137
  !PUSH s,r5; %save the return location%           0138
  !MOVE m,s; %new mark%                            0139
  !MOVEM a1,1(m); %store the single argument%     0140
  !JRST @40B;                                       0141
(qcallm):                                          0142
  %get here by a CALLM pop%                         0143
  !ROT r4,4; % get AC field in right end %        0144
  !ANDI r4,17B; % only 4 bits %                   0145
  !HRLI r4,0(r4); % copy into left half %         0146
  !SUB s,r4; %now same as before started call%    0147
  !PUSH s,m; %save the mark%                       0148
  !PUSH s,r5; %save the return location%           0149
  !MOVE m,s; %new mark%                            0150
  !JRST @40B;                                       0151
(qcallm2):                                         0152
  %get here by a CALLM pop%                         0153
  !ROT r4,4; % get AC field in right end %        0154
  !ANDI r4,17B; % only 4 bits %                   0155
  !SUB s,0(r4); %now same as before started call% 0156
  !PUSH s,m; %save the mark%                       0157
  !PUSH s,r5; %save the return location%           0158
  !MOVE m,s; %new mark%                            0159
  !JRST @40B;                                       0160
END.

```

**

0161
0163


```

cl,      %location of record for a cl to be updated%      0297
end,     %location of end of cl table%                    0298
cltab,   %location of cl table%                          0299
flhd,    %file header loc%                               0300
ch,      %character read from the tstring%               0301
bfrom,   %source byte pointer%                          0302
lenstat, %number of chars from string%                   0303
lenblank;%number of chars from string%                   0304
REF cl;  0305
IF [tp] # [tp+2] OR [tp+1] <= empty THEN                0306
  BEGIN 0307
  modeset _ 0; %reset global modeset flag %              0308
  err($"illegal string designation");                      0309
  END; 0310
IF [tp+1] >= [tp+3] THEN                                0311
  BEGIN 0312
  modeset _ 0; %reset global modeset flag %              0313
  RETURN; 0314
  END; 0315
ON SIGNAL ELSE modeset _ 0;                              0316
stcwrk _ [tp]; 0317
stcwr1 _ [tp+1]; 0318
fechcl (forward, $stcwrk); 0319
% stcwrk[2] now contains length + 1 of source string %   0320
% calculate number of chars to take from source string % 0321
CASE stcwrk[2] OF 0322
  <= [tp+1] : 0323
    BEGIN 0324
    lenstat _ 0; 0325
    lenblank _ [tp+3] - [tp+1]; 0326
    END; 0327
  < [tp+3] : 0328
    BEGIN 0329
    lenstat _ stcwrk[2] - [tp+1]; 0330
    lenblank _ [tp+3] - stcwrk[2]; 0331
    END; 0332
ENDCASE 0333
  BEGIN 0334
  lenstat _ [tp+3] - [tp+1]; 0335
  lenblank _ 0; 0336
  END; 0337
&cl _ cltab _ [clstad].clbuff; 0338
%address of start of clist% 0339
end _ &cl + [clstad].clcnt * cll; 0340
%end of clist% 0341
ON SIGNAL ELSE; 0342
IF modeset THEN 0343
  BEGIN % mode set now in operation % 0344
  modeset _ 0; %reset the global flag% 0345
  UNTIL (lenstat _ lenstat-1) < 0 DO 0346
    BEGIN 0347
    ch _ READC($stcwrk); 0348
    %now append the character% 0349
    CASE modeshift OF %check for forced case% 0350
      =0: apachr(IF ch IN ['a','z'] THEN ch-40B ELSE ch); 0351
      %forced upper case% 0352

```

```

=1: apachr(IF ch IN ['A','Z'] THEN ch+408 ELSE ch); 0353
    %forced lower case% 0354
    ENDCASE apachr(ch); %no case change% 0355
END; 0356
END 0357
ELSE %no mode change. just copy across% 0358
BEGIN 0359
IF NOT mkrstay THEN 0360
    IF NOT rplsid.stastr THEN 0361
        BEGIN 0362
            IF rplsid.stfile = [tp].stfile THEN 0363
                UNTIL &cl = end DO 0364
                    BEGIN 0365
                        IF cl.clst1 = [tp] AND 0366
                            cl.clcc1 IN [[tp+1],[tp+3]] AND 0367
                            NOT cl.clfixed THEN 0368
                                BEGIN 0369
                                    cl.clfixed _ TRUE; 0370
                                    cl.clcc1 _ 0371
                                        sar.L + 1 + cl.clcc1 - [tp+1]; 0372
                                    PUSH &cl ON clchng; 0373
                                END; 0374
                                    &cl _ &cl + cll; 0375
                                END 0376
                            ELSE cldsr(tp); 0377
                        END; 0378
                    IF (sar.L _ sar.L + lenstat) > sar.M THEN 0379
                        err($"NLS internal error: string too long"); 0380
                    a4 _ stcwrk[4]; %byte pointer made by fechcl% 0381
                    a2 _ lenstat; 0382
                    a3 _ nsdbpt; 0383
                    UNTIL (a2 _ a2-1) < 0 DO ^a3 _ a1 _ ^a4; 0384
                    nsdbpt _ a3; 0385
                END; 0386
            UNTIL (lenblank _ lenblank-1) < 0 DO apachr(SP); 0387
            mkrstay _ FALSE; 0388
        RETURN END.

```

0389

(cldsr) %Called with address of two tpointers. For all clist entries, if between these two pointers, then set to "deleted".%

```

PROCEDURE (tp); 01490
LOCAL end, cl; 01491
REF cl; 01492
&cl _ [clstad].clbuff; 01493
    %address of start of clist% 01494
end _ &cl + [clstad].clcnt * cll; 01495
    %end of clist% 01496
UNTIL &cl >= end DO 01497
    BEGIN 01498
        IF cl.clst1 = [tp] AND 01499
            cl.clcc1 IN [[tp+1],[tp+3]] AND 01500
            NOT cl.clfixed THEN 01501
                BEGIN 01502
                    cl.clst2 _ cl.clst1 := endfil; 01503
                    cl.clcc2 _ [tp+1]; 01504
                END; 01505

```

```

        END;                                01506
        &c1 _ &c1 + c1;                       01507
        END;                                01508
RETURN END.

* Reading characters from SDB%                                01509
(feich1)                                                    0421
  %Documentation%                                          0422
  %This routine is called to initialize a work area for reading
  characters from a statement. Arguments are: direction of
  reading characters (=0 then backwards) and the address of the 7
  word work area (of which the last 2 words are no longer used).
  0424
  If characters are to be read from a statement then      0425
  when calling FECHC1, the first two cells of the work area
  must contain a Tpointer. A character count of one indicates
  the first character of the statement. FECHC1 will initialize
  the rest of the work area. The first word of the word area
  always has the PSID of the statement. The second word has
  the character count. The third word contains a bound on
  characters to be read. ENDCHR's are returned after reach
  this bound. The fourth word has the direction of readout
  for use by readc. A READC(x) actually results in the value
  of x being loaded into register wa followed by a JSP
  a4,readc.                                             0426
  The fifth word of the work area contains a byte pointer to
  the character last read from the statement. Thus an ILDB
  instruction may be used to get the next character if the
  direction is forward. If the direction of reading is
  backward, then the byte pointer is decremented by in-line
  code.                                                 0427
  To read characters from the statement execute a READC(x)
  where the value of x is the address of the work area to be
  used. The character is returned as the value of the READC.
  Subsequent READC's will return the following characters.
  0428
  To change position or direction within the statement the
  work area must be reinitialized by calling FECHC1 again, as
  described above. There may however be more than one work
  area currently in use, and these may be changed
  independently.                                       0429
  If characters are to be read from an A-string then      0430
  the first word of the work area contains the address of the
  A-string instead of a PSID. The second word is 1 if the
  first character of the string is to be read next, two if the
  second, etc. Characters may be read out of an A-string in
  either direction, just like a statment. Endcharacters are
  returned when the string is exhausted.%              0431
PROCEDURE (dir, worka);                                  0432
LOCAL                                                    0433
  wp, %word postion of the starting character%          0434
  cp, %character postion of the starting character%     0435
  stdb, %stdb for statement%                            0436
  rng, %location of ring element%                      0437
  addr; %address of the word containing starting character% 0438
REF rng;                                                0439

```

```

%set work+2 to bound%                                0440
IF [worka] = endfil THEN %set to null string%        0441
  BEGIN                                              0442
    [worka+3] _ 2; %readc will return ENDCHR%       0443
    RETURN;                                          0444
  END                                                0445
ELSE IF [worka].stastr THEN %A-string%              0446
  BEGIN                                              0447
    addr _ [worka].stadr;                            0448
    [worka+2] _ IF pe THEN pe ELSE [addr].L + 1;    0449
    addr _ addr + 1;                                  0450
  END                                                0451
ELSE %SDB%                                           0452
  addr _ flfechc1( dir, worka : stdb );             01526
%find word and character position%                   0462
IF dir THEN %scan forward%                           0463
  BEGIN                                              0464
    [worka+3] _ 1; %direction%                       0465
  END                                                0466
ELSE %scan backwards%                                0467
  BEGIN                                              0468
    [worka+2] _ IF ps THEN ps ELSE 1; %change bound% 0469
    [worka+3] _ 0; %direction%                       0470
  END;                                               0471
%make byte pointer to the previous (forward) or current (backward)
character%                                           0472
DIV ([worka+1]-[worka+3])/5, wp, cp;                 0473
[worka+4] _ 440700B6 - cp * 70000B6 + addr + wp;    0474
RETURN END.                                          0475

(openswork) PROCEDURE;                               0476
  fechc1(1, $swork);                                0477
RETURN END.                                          0478

(savpos) PROCEDURE;                                 0479
  PUSH ps ON btwstk;                                 0480
  PUSH pe ON btwstk;                                 0481
RETURN END.                                          0482

(respos) PROCEDURE;                                 0483
  POP btwstk TO pe;                                  0484
  POP btwstk TO ps;                                  0485
RETURN END.                                          0486

(xhreadc) PROCEDURE; %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.%                                0487
(readc):                                             0488
%cant have locals since is not called by usual procedure linkage.
wa is a register containing the address of the work area.% 0489
!SKIPG a1,3(wa);                                     0490
  !JRST readc1;                                       0491
!SOJN a1,readend;                                    0492
%forward scan%                                       0493
  !AOS a1,1(wa);                                       0494
  !CAMLE a1,2(wa);                                     0495

```



```

        !JRST rdend1;                                0496
        !ILDB a1,4(wa); %get the character%          0497
        !JRST 0(a4);                                  0498
(readc1):                                          0499
!JUMPN a1,readend;                                0500
%backward scan%                                  0501
        !SOS a1,1(wa);                                0502
        !CAMGE a1,2(wa);                              0503
        !JRST rdend2;                                  0504
%back up byte pointer%                            0505
!MOVE a1,4(wa);                                    0506
!ADD a1,=7B10;                                      0507
!CAIG a1,0;                                          0508
!SUB a1,=430000000001B;                             0509
!MOVEM a1,4(wa);                                    0510
!LDB a1,a1; %get the character%                    0511
!JRST 0(a4);                                         0512
(rdend2): !AOSA 1(wa); %adjust for backward overrun, NOTE SKIP% 0513
(rdend1): !SOS 1(wa); %adjust for forward overrun, SKIPPED OVER% 0514
(readend): %out of bounds%                           0515
        a1 _ ENDCHR;                                   0516
        !JRST 0(a4);                                   0517
END.
% A-string routines%                               0518
(asrref) %create a pointer to astring with stastr field set.% 0547
        PROCEDURE (ast);                               0548
        ast.stastr _ TRUE;                             0549
        RETURN (ast) END.                             0550
DECLARE % byte pointers for chbptr %                0551
        chparray=(                                     0553
        350700000001B,                                0554
        260700000001B,                                0555
        170700000001B,                                0556
        100700000001B,                                0557
        107000000001B);
DECLARE                                             0558
        cshift=(7,14,777761B %-15%, 777770B %-8%, 777777B %-1%); 01624
EXTERNAL ldchr, chbptr, apchr;                       0560
DECLARE                                             0561
        ascomm=(0,774B9,77776B7,777777B5,-400B);    01625
(xutilty) PROCEDURE;                                 0562
% a place for several "fast" routines to reside. NOTE: These
routines cannot use locals or make calls since they are lables:
they avoid the stack setup instructions at procedure entry. They
are called, however, as procedures with arguments. % 0563
(ldchr): % (astr, charno) %                          0564
% return a specified character from an astring %      0565
!MOVE a3,@1(m); % string max,,len %                  0566
!MOVE a1,2(m); % char number %                       0567
!CAILE a1,0(a3); % over end? %                       0568

```

```

!JRST ldchr1; % yes, return ENDCHR % 0569
!JUMPE a1,sysrtn; % zeroth char - return zero % 0570
!MOVEI a1,-1(a1); 0571
!IDIVI a1,5; % char-1/5 % 0572
!ADD a1,1(m); % add address of string % 0573
!MOVE a1,1(a1); % get word from string % 0574
!ROT a1,@cshift(a2); % rotate and mask % 0575
!ANDI a1,177B; % faster than LDB % 0576
RETURN; 0577
(ldchr1): 0578
    RETURN(ENDCHR); 0579
(chbptr): % (char) % 0580
    % return a partial byte pointer given a character number. The
    % resulting byte pointer needs to have the address of the astring
    % added to it % 0581
    !SKIPN a1,1(m); %check for zero char count % 0582
    !JRST chbpt1; 0583
    !MOVEI a1,-1(a1); %subtract one % 0584
    !IDIVI a1,5; 0585
    !ADD a1,chparray(a2); %get partial pointer % 0586
    RETURN; 0587
(chbpt1): 0588
    RETURN(chbpty); 0589
(apchr): % (char, astr) % 0590
    % Append the given character to the specified astring. If the
    % string is at it's max length an exceed capacity error results % 0591
    !MOVE a2,@2(m); % get max,,len from string % 0592
    !MOVSI a1,1(a2); % get len+1 % 0593
    !CAMLE a1,a2; % over max ? % 0594
    !JRST apxcd; 0595
    !HLRM a1,@2(m); % store len+1 % 0596
    !MOVEI a2,0(a2); % isolate new length-1 = old len % 0597
    (apstore): 0598
        !IDIVI a2,5; 0599
        !ADD a2,chparray(a3); % get partial pointer % 0600
        !ADD a2,2(m); % plus string address % 0601
        !MOVE a1,1(m); % get character % 0602
        !DPB a1,a2; % and store it % 0603
        RETURN; 0604
    (apxcd): 0605
        err($"NLS internal error: string too long"); 0606
    NULL END. 0607
(stbptr) %character byte pointer% 01513
PROCEDURE (charno); 01514
    %creates a partial byte pointer to the character determined by the
    %character number passed as argument. The resulting byte pointer
    %needs to have the address of the string (NOT ASTRING) added to
    %it.% 01515
    LOCAL wp, cp; 01516
    IF charno = empty THEN RETURN (440700000000B); 01517
    DIV (charno-1) / 5, wp, cp; 01518
    RETURN (chparray[cp] + wp - 1) END. 01519
(ascom) PROCEDURE(astr1, astr2, relation); 0608
    % accepts addresses of 2 a-strings. returns logical value of

```

```

astr1 <relation> astr2 %                                0609
LOCAL reltab;                                          0610
r1 _ -1; % set flag in r1 - means must do all of compare % 0611
r2 _ relation;                                         0612
(comstr): % called by ascom and compas %              0613
% r1 is a flag: =0 means return false if a mismatch (compas
call). <0 means must compare entire string and return reltab
(ascom call). >0 means return the sign of astr1 = astr2 (cmpstr
call) %                                               0614
IF [astr1].L # [astr2].L THEN BEGIN                    0615
    !JUMPE r1,ascom6; % RETURN(FALSE) if called from compas %
                                                         0616
    !SKIPE lngflg; % cmpstr: IF lngflg AND lengths not equal %
                                                         0617
    !JUMPG r1,ascom5; % cmpstr call - compare lengths % 0618
    !CAIN r2,2; % is relation = ? %                    0619
    !JRST ascom6; % yes. RETURN(FALSE) %              0620
    !CAIN r2,6; % if relation # ? %                   0621
    !JRST ascom7 % yes. RETURN(TRUE) %                0622
    END;                                               0623
!IMULI r2,3; % r2 _ r2*3+1 %                          0624
!MOVEI r2,1(r2); % plus one %                         0625
a3 _ [m+1]; % $astr1 %                                0626
a4 _ [m+2]; % $astr2 %                                0627
!HRRZ a1,0(a3); % len of astr1 %                      0628
!HRRZ a2,0(a4); % len of astr2 %                     0629
!CAIG a2,0(a1); % find min len %                     0630
!MOVEI a1,0(a2); % now a1 has the min %               0631
!IDIVI a1,5; % length/5: a1=number of words, a2=mask index %
                                                         0632
(ascom2):                                              0633
    !MOVEI a3,1(a3); % add one to pointer %           0634
    !MOVEI a4,1(a4); % ditto %                        0635
    !MOVE r3,0(a3); % get a word of string %          0636
    !TRZ r3,1; % turn off low order bit %             0637
    !SOJL a1,ascom1; % last word of string ? %       0638
    !MOVE r4,0(a4); % get other word %                0639
    !TRZ r4,1; % and turn of low order bit %         0640
    !SUB r3,r4; % now, take difference %              0641
    !JUMPE r3,ascom2; % if equal, keep going %       0642
    !JUMPE r1,ascom6; % quit here if we can (i.e. from compas)
    %                                                 0643
(ascom3):                                              0644
    !TLNE r3,4B5; % compute sign of result: sign bit on? %
                                                         0645
    !SKIPA r3,=-1; % yes, get -1 %                   0646
    !MOVEI r3,1; % no, get one %                     0647
(ascom4):                                              0648
    !JUMPG r1,ascom8; % cmpstr call - return sign, not reltab %
                                                         0649
    !ADDI r3,0(r2); % add in reltab index %          0650
    RETURN(reltab[r3]); % return result from table % 0651
(ascom1):                                              0652
    !JUMPE a2,ascom5; % last word of str. any chars in it? %
                                                         0653
    !AND r3,ascomm(a2); % and off extraneous chars % 0654

```

```

!MOVE r4,0(a4); % get word from other string %      0655
!AND r4,ascomm(a2); % and it also %                0656
!SUB r3,r4; % take difference %                    0657
!JUMPN r3,ascom3; % check sign if not equal %      0658
(ascom5):                                           0659
!HRRZ r3,@1(m); %last words equal. compare lengths % 0660
!HRRZ r4,@2(m); % get other length %              0661
!SUBI r3,0(r4); % take difference %               0662
!JUMPE r3,ascom4; % equal, return result %        0663
!JRST ascom3; % not equal, compute sign %         0664
(ascom6):                                           0665
RETURN(FALSE);                                     0666
(ascom7):                                           0667
RETURN(TRUE);                                       0668
(ascom8):                                           0669
RETURN(r3); % return sign for cmpstr call %       0670
NULL END.                                           0671
(compas) PROCEDURE(astr1, astr2);                   0672
% compares the contents of two a-strings. Returns true if the
% contents match, false otherwise %                0673
r1 _ 0; r2 _ 2; % relation = true %                0674
GOTO comstr END.                                    0675
(cmpstr) PROCEDURE(astr1, astr2);                   0676
% return -1 if astr1 < astr2, 0 if equal, 1 if astr1 > astr2 %
                                                    0677
r1 _ 1; % set flag >0 means return sign value %  0678
GOTO comstr END. % value of r2 does not matter %  0679
(repchr) %replace a character in an astring%
                                                    0687
PROCEDURE (char, ast, chrno);                       0688
%-----%                                          0689
LOCAL cnt;                                         0690
REF ast;                                           0691
CASE chrno OF                                      0692
> ast.M : err($"NLS internal error: string too long"); %string
overflow%                                         0693
> ast.L : % blank fill if necessary %            0694
BEGIN                                              0695
IF (cnt _ chrno - ast.L) > 1 THEN apblnk(&ast, cnt-1); 0696
ast.L _ chrno;                                    0697
END;                                              0698
<=0: err($"repchr called with chrno<=0");        0699
ENDCASE;                                          0700
!MOVE a2,3(m); % the char number %                0701
!MOVEI a2,-1(a2); % minus one %                  0702
!JRST apstore                                     0703
END.
                                                    0704
(apblnk) PROCEDURE(ast, cnt);                       0705
% Append cnt blanks to the designated astring. %  0706
REF ast;                                           0707
FOR cnt DOWN UNTIL < 1 DO *ast* _ *ast*, SP;      0708
RETURN END.
                                                    0709
(apstr) %The purpose of this routine is to append one A-string onto
another. Arguments are two A-string addresses. The first string is

```



```

        lenblank _ 0;                                0757
        END;                                          0758
    IF ast2.L + lenstr + lenblank > ast2.M THEN      0759
        err($"NLS internal error: string too long"); 0760
    bto _ chbptr(ast2.L) + &ast2;                    0761
    bfrom _ chbptr(lower-1) + &ast1;                 0762
    ast2.L _ ast2.L + lenstr + lenblank;            0763
    a2 _ lenstr;                                     0764
    a3 _ bto;                                         0765
    a4 _ bfrom;                                       0766
    UNTIL (a2 _ a2 - 1) < 0 DO ^a3 _ a1 _ ^a4;      0767
    a2 _ lenblank;                                    0768
    a1 _ SP;                                          0769
    UNTIL (a2 _ a2 - 1) < 0 DO ^a3 _ a1;            0770
    RETURN END.

```

0771

(mvdibf) %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.%

```

        PROCEDURE (bfrom, bto, nw);                  01224
        LOCAL lw;                                    01225
        IF nw = 0 THEN RETURN;                       01226
        lw _ nw + bto - 1; %last word to be transferred to% 01227
        IF bto IN (bfrom, bfrom+nw) THEN             01228
            BEGIN                                     01229
                a1 _ bfrom;                            01230
                a2 _ nw; a3 _ a2;                      01231
                a2 _ a2 + a1; %pointer into source%    01232
                a3 _ a3 + bto; %pointer into destination% 01233
                UNTIL (a2 _ a2-1) < a1 DO              01234
                    BEGIN                              01235
                        a3 _ a3 - 1;                   01236
                        [a3] _ [a2];                   01237
                    END;                               01238
            END;                                       01239
        ELSE %can use block transfer instruction%     01240
            BEGIN                                     01241
                !HRL a1,bfrom; !HRR a1,bto; !BLT a1,@lw 01242
            END;                                       01243
        RETURN (lw) END.                               01244

```

```

        DECLARE                                     01245
        nshmsk=(                                     01318
            774B9,                                    01627
            77776B7,                                    01319
            7777777B5,                                    01320
            777777774B2,                                01321
            77777777776B);                             01322

```

01323

(hash) %An a-string address as argument. The hash code for that string is returned.%

```

        PROCEDURE (ast);                              01299
        LOCAL nw, i, gen1, gen2, old1, cp;           01300

```

01301

```

REF ast;                                01302
IF (oldl _ ast.L) <= empty THEN RETURN(0); 01303
DIV (oldl + 4)/5, nw, cp;                01304
FOR i _ 1 UP UNTIL = nw DO ast[i] _ ast[i] .A hshmsk[4]; 01305
ast[nw] _ ast[nw] .A hshmsk[cp];        01306
gen1 _ 0;                                01307
gen2 _ 1;                                01308
UNTIL nw = 0 DO                          01309
  BEGIN                                  01310
    gen1 _ ast[nw] * gen2 + gen1 / 17;    01311
    gen2 _ gen2 * 43;                    01312
    nw _ nw - 1;                         01313
  END;                                   01314
a1 _ gen1;                                01315
!LSH a1,-6;                              01316
RETURN(a1) END.                          01317

(astruc) PROCEDURE(astring); %a-string to upper case%
% convert a-string (in astring) to upper case %
%-----%                                01250
LOCAL length, bytptr, char;              01251
REF astring;                              01252
IF (length _ astring.L) = empty THEN RETURN(&astring); 01253
bytptr _ chbptr(empty) + &astring;        01254
DO IF (char _ ^bytptr) IN ['a','z'] THEN  01255
  .bytptr _ char - 40B                   01256
UNTIL (length _ length - 1) = empty;     01257
RETURN(&astring);                         01258
END.                                       01259
                                           01260
                                           01261

(singth) PROCEDURE (bp1, bp2); %string length%
%returns the length of the string represented by the passed byte
%pointers, the first of which is assumed to be set so that an
%increment and load byte will get first char of the string and the
%second pointing to the last char in the string. NOTE: the index
%field is ignored and the only the size field of bp1 is used!%
                                           01262
%bpadr, bpsize, bpbitpos%                01263
%-----%                                01264
LOCAL length;                             01265
length _ (bp2.bpadr - bp1.bpadr)*(36 / bp1.bpsize) +
  (bp1.bpbitpos - bp2.bpbitpos) / bp1.bpsize; 01266
RETURN(MAX(length, 0));                   01267
END.                                       01268
                                           01269

(srmake) %make string from value%         01270
PROCEDURE(value,astring,base);           0772
LOCAL power, char, temp;                 0773
REF astring;                              0774
IF value < 0 THEN                         0775
  BEGIN                                  0776
    IF &astring THEN *astring* _ *astring*, '- 0777
    ELSE apachr('-');                     0778
    value _ -value;                       0779
  END;                                    0780
power _ 1;                                0781
                                           0782

```

```

WHILE ((temp _ power*base) <= value) AND (temp > power) DO power _
temp; 0783
UNTIL power < 1 DO 0784
  BEGIN 0785
  DIV value / power, char, value; 0786
  IF &astrng THEN *astrng* _ *astrng*, char + '0 0787
  ELSE apachr(char + '0); 0788
  power _ power / base; 0789
  END; 0790
RETURN END.

(srmk) PROCEDURE(value,base); 0791
%In string construction use apachr to append string to sar.% 0792
srmake(value, 0, base); 0793
RETURN END. 0794

(srval) PROCEDURE(astrng, base); 0795
%convert string to value% 0796
LOCAL value, cnt, char; 0797
REF astrng; 0798
value _ 0; 0799
cnt _ 1; 0800
UNTIL (char _ *astrng*[cnt]) = ENDCHR DO 0801
  BEGIN 0802
  value _ value*base + char - '0; 0803
  BUMP cnt; 0804
  END; 0805
RETURN (value) END. 0806

DECLARE 0807
reltab= (0,0,0, %no 0 relational% 0817
TRUE, FALSE, FALSE, %1: <% 0818
FALSE, TRUE, FALSE, %2: =% 0819
TRUE, TRUE, FALSE, %3: <=% 0820
0, 0, 0, %no 4 relational% 0821
FALSE, TRUE, TRUE, %5: >=% 0822
TRUE, FALSE, TRUE, %6: #% 0823
FALSE, FALSE, TRUE); %7: >%

(repstr) %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.%

PROCEDURE (newast, oldast, charno); 0825
%-----% 0826
LOCAL astrl, repcnt; 0827
REF newast, oldast; 0828
IF (astrl _ newast.L) = empty THEN RETURN (FALSE); 0829
IF charno + newast.L > oldast.L THEN 0830
  BEGIN 0831
  IF charno + newast.L > oldast.M THEN 0832
    RETURN (FALSE); 0833
  oldast.L _ charno + newast.L; 0834
  END; 0835
repcnt _ empty + 1; 0836

```



```

UNTIL repcnt > astr1 DO                                0838
  BEGIN                                                0839
    *oldast*[charno] _ *newast*[repcnt];              0840
    BUMP charno, repcnt;                               0841
  END;                                                 0842
RETURN (TRUE)                                         0843
END.                                                  0844

(srlset) %To set an A-string to a single character, call this
procedure with the character and the address of the string.%
                                                    0858
PROCEDURE (char, ast);                                0859
[ast].L _ empty;                                     0860
apchr (char, ast);                                   0861
RETURN END.                                          0862

(mkbptr) %Make Byte Pointer with the specified position, size and
address. The address is 23 bits wide to allow indexing and
indirection.%
                                                    0863
PROCEDURE (pos, size, addr);                          0864
a1 _ addr; !LSHC a1,-24;                              0865
a1 _ size; !LSHC a1,-6;                               0866
a1 _ pos; !LSHC a1,-6;                               0867
RETURN (a2) END.
                                                    0868

% Statement scanning & content-analysis support%
(pdc) PROCEDURE (pdcnum, pntloc);                      0924
%pointer decrement. arguments are number of positions to
move and address of pointer to be decremented. %
                                                    0932
LOCAL
end; %count at end of statement%                      0933
IF scndir = forward THEN                              0934
  [pntloc+1] _ MAX([pntloc+1]-pdcnum, 1)              0935
ELSE                                                  0936
  BEGIN                                              0937
    cpfse([pntloc]);                                0938
    end _ a2;                                       0939
    [pntloc+1] _ MIN([pntloc+1]+pdcnum, end);        0940
  END;                                              0941
RETURN;                                             0942
END.                                               0943
                                                    0944

(dpstr) PROCEDURE(pntloc);                             0945
pdc(1, pntloc);                                     0946
RETURN END.                                         0947
                                                    0948

(tstsr) PROCEDURE (ast);                              0949
%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.%
                                                    0950
LOCAL                                              0951

```

```

    cnt,      %counter for characters in test string%      0952
    char,     %character from source string%              0953
    tsp;      %pointer into test string%                 0954
REF ast;                                             0955
cnt _ ast.L;                                       0956
tsp _ chbmt + &ast;                               0957
UNTIL (cnt _ cnt-1) < 0 DO                          0958
    IF (char _ READC) = ENDCHR OR ^tsp # char THEN RETURN; 0959
SKIP RETURN END.                                     0960
(tstf) PROCEDURE (ast);                               0961
%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.%
LOCAL
    cnt,      %counter for characters in test string%      0962
    cpos,     %value of swork1 where got match%          0963
    nchar,    %number of characters in test string%      0964
    char,     %character from source string%              0965
    char1,    %first character of test string%           0966
    tsp,      %pointer into the test string%             0967
    tsp1;     %pointer to first character of test string% 0968
REF ast;                                             0969
nchar _ ast.L;                                       0970
IF nchar < 1 THEN SKIP RETURN;                       0971
tsp1 _ chbmt + &ast;                               0972
char1 _ ^tsp1; %first character of string%             0973
LOOP %scan until get match for first character in string% 0974
    CASE READC OF
        = ENDCHR : EXIT; %have reached the end%        0975
        = char1 : %have a match on the first character% 0976
        BEGIN
            cpos _ swork1;                               0977
            cnt _ 1;                                     0978
            tsp _ tsp1;                                  0979
            LOOP
                BEGIN
                    IF (cnt _ cnt+1) > nchar THEN SKIP RETURN; 0980
                    IF (char _ READC) = ENDCHR THEN EXIT 2; 0981
                    IF ^tsp # char THEN EXIT;            0982
                END;
                %failure -- back to scanning for first character% 0983
                swork1 _ cpos;                            0984
                fechcl(scndir, $swork);                  0985
            END;
        ENDCASE;
    RETURN;
    END.

```

```

(Dis) %branch false and scan %

```

0997

% Resets the character position, then reads another character from the statement. If this is not an ENDCHR the character number is put on the stack and control is transferred to the location specified in the address of the pop.%

0998

```
PROCEDURE (brloc);
swork _ [p-1];
swork1 _ [p];
fechcl(scndir, $swork);
IF READC # ENDCHR THEN
BEGIN
[p] _ swork1;
[cm] _ brloc; %change return location%
RETURN;
END;
p _ p - 2000002B;
RETURN END.
```

0999

01000

01001

01002

01003

01004

01005

01006

01007

01008

01009

01010

```
(fxswork) PROCEDURE;
fechcl(scndir, $swork);
RETURN END.
```

01011

01012

01013

```
(srsup) PROCEDURE; %dummy procedure for string support code%
%these are all called by JSP a4,x and return by JRST (a4)%
```

01014

01015

```
(fcp):
```

01016

```
swork _ a1;
```

01017

```
swork1 _ a2;
```

01018

```
!JRST (a4);
```

01019

```
(pcp):
```

01020

```
!PUSH p,swork;
```

01021

```
!PUSH p,swork1;
```

01022

```
!JRST (a4);
```

01023

```
(sptr):
```

01024

```
!MOVE a2, swork;
```

01025

```
!MOVEM a2,0(a1);
```

01026

```
!MOVE a2, swork1;
```

01027

```
!MOVEM a2,1(a1);
```

01028

```
!JRST (a4);
```

01029

```
(begarb):
```

01030

```
!PUSH p,swork;
```

01031

```
!PUSH p,swork1;
```

01032

```
!HLRZ a2,a1; %lower bound%
```

01033

```
!PUSH p,a2;
```

01034

```
!HRRZ a2,a1; %upper bound%
```

01035

```
!PUSH p,a2;
```

01036

```
a2 _ 0; %count%
```

01037

```
!PUSH p,a2;
```

01038

```
!JRST (a4);
```

01039

```
(fxsp1): %leave a1 and a2 unchanged%
```

01040

```
sptr1 _ a1;
```

01041

```
a3 _ 1;
```

01042

```
sptr1[a3] _ a2;
```

01043

```
!JRST (a4);
```

01044

```
(fxsp2):
```

01045

```
sptr2 _ a1;
```

01046

```

    a3 _ 1;                                01047
    sptr2[a3] _ a2;                         01048
    !JRST (a4);                              01049
(lpr):                                       01050
    a1 _ [a2];                               01051
    a2 _ [a2+1];                             01052
    !JRST (a4);                              01053
END.                                         01054

EXTERNAL fcp, pcp, sptr, begarb, fxsp1, fxsp2, lpr;

(incarb) PROCEDURE;                         01055
    %update position on stack where last succeeded%
    [p-4] _ swork;                           01058
    [p-3] _ swork1;                          01059
    %increment count and compare to upper bound%
    RETURN(([p] _ [p]+1) < [p-1]) END.      01060

(endarb) PROCEDURE;                         01061
    p _ p - 3000003B;                        01063
    scp();                                    01064
    p _ p - 2000002B;                        01065
    RETURN([p+5] IN [[p+3],[p+4]]) END.

(scp) PROCEDURE;                            01066
    swork _ [p-1];                           01068
    swork1 _ [p];                             01069
    fechc1(scndir, $swork);                  01070
    RETURN END.

(scnlft) PROCEDURE;                         01071
    fechc1(scndir _ 0, $swork);              01073
    RETURN END.

(scnrht) PROCEDURE;                         01074
    fechc1(scndir _ 1, $swork);              01076
    RETURN END.

(chrct) PROCEDURE (char, chrcls); %Character class test%
    CASE chrcls OF
        =1: %CH%                              01078
            IF char IN [0,177B] THEN GOTO cctyes; 01081
        =4: %LD%                              01082
            IF char IN ['A','Z] OR
               char IN ['a','z] OR
               char IN ['0','9] THEN GOTO cctyes; 01085
        =11: %NLD%                             01086
            IF char # ENDCHR AND
               char NOT IN ['A','Z] AND
               char NOT IN ['a','z] AND
               char NOT IN ['0','9] THEN GOTO cctyes; 01090
        =7: %L%                                01091
            IF char IN ['a','z] OR
               char IN ['A','Z] THEN GOTO cctyes; 01093
        =8: %D%                                01094

```



```

% storage manipulation.  stacks, rings, buffers%                                01155
DECLARE EXTERNAL FIELD                                                            01156
  systkp= [0(rp), 18:0], %pointer%                                              01157
  systkt= [0(rp), 13:23], %type (1=stack, 2=ring)%                             01158
  systks= [1(rp), 18:18], %size of element%                                     01159
  systke= [1(rp), 18:0]; %end of store%                                         01160
  % rcpsh, rpop, rstsk, and <IDLIBE>gbotids use the fact that the
  stack body starts in word 2. %                                              01161
(rcpsh) %Record push.  Arguments are (1) address of record, and (2)
address of stack or ring buffer.  The record is pushed on the store.%         01162
  PROCEDURE (recadr, st);                                                       01163
  REF st;                                                                        01164
  IF st.systkt NOT IN [1,2] THEN err(0);                                        01165
  IF (st.systkp _ st.systkp + st.systks) = st.systke THEN                      01166
    IF st.systkt = 1 THEN err($"Stack overflow") %stack overflow%              01167
    ELSE st.systkp _ &st + 2; %ring wrap around%                               01168
  IF st.systks = 1 THEN [st.systkp] _ [recadr]                                  01169
  ELSE mvbfbf (recadr, st.systkp, st.systks);                                  01170
  RETURN END.                                                                    01171
(rpop) %Record pop.  Argument is the address of a store.  Pop the top
element.%                                                                        01172
  PROCEDURE (st);                                                                01173
  REF st;                                                                        01174
  IF st.systkt = 1 %stack% THEN                                                 01175
    IF st.systkp < &st+2 THEN err(0) %stack underflow%                       01176
    ELSE NULL                                                                    01177
  ELSE IF st.systkt = 2 %ring% THEN                                             01178
    IF st.systkp <= &st+2 THEN                                                  01179
      st.systkp _ st.systke                                                    01180
    ELSE NULL                                                                    01181
  ELSE err(0); %wrong type%                                                     01182
  st.systkp _ st.systkp - st.systks;                                           01183
  RETURN END.                                                                    01184
(rstsk) %Reset store%                                                           01185
  PROCEDURE (st);                                                                01186
  REF st;                                                                        01187
  IF st.systkt NOT IN [1,2] THEN err(0);                                        01188
  st.systkp _ &st + 2 - st.systks;                                             01189
  RETURN END.                                                                    01190
(rcpto) %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.%                                                           01191
  PROCEDURE (st, recadr);                                                       01192
  REF st;                                                                        01193
  IF st.systkt NOT IN [1,2] THEN err(0);                                        01194
  IF st.systks = 1 THEN [recadr] _ [st.systkp]                                  01195
  ELSE mvbfbf (st.systkp, recadr, st.systks);                                  01196

```

GAS2, 14-Feb-79 22:36

< NLS, L10RUNTIME.NLS.8, > 27

rpop (&st);
RETURN END.

01197

FINISH

01198
052

L10 SYMS


```

< NLS, L10SYMS.NLS;5, >, 16-MAY-75 01:04 DIA ;;;
SYMBOLS ? (XL10,) to (rel-nls,junk,) % 02
% OR (L1011,) to (rel-nls,junk,) % 03
% This file contains initializing symbols for L10 and L1011 compilers.
It should only contain CONSTANT and ADDRESS statements. To make a
compiler or after changing this file do the following: 019
Load compiler. 020
Initialize by starting at label INITLL 021
Then (from EXEC) START and give compiler TXT copy of this file and
JUNK for output. 022
Then save core image as compiler subsystem. 023
% 024
% signal type names % 04
DECLARE CONSTANT 05
    helptype=1, % HELP % 06
    notetype=2, % NOTE % 07
    aborttype=3; % ABORT % 08
% world-wide signal names % 09
% System signal names are greater than 10000 octal. Program-wide
signal names should be less than 10000 octal % 010
DECLARE CONSTANT 011
    nohelp=10000B, % on resume, means no help obtained % 012
    % happens when HELP is not caught % 027
    return=10001B, % a NOTE: procedure returning % 013
    % Whenever a procedure or its coroutine did an INVOKE, and then
    does RETURN % 028
    unwind=10002B, % NOTE, this routine will vanish % 014
    % happens when higher routine does TERMINATE % 026
    % also when runtime package does a recover % 039
    gothelp=10003B, % on resume, means help obtained % 015
    % Should be standard first argument for RESUME % 029
    stringoverflow=10004B, % HELP, string overflowed % 016
    % low level string handlers generate this % 030
    % arg2 is string address % 033
    % they expect resume(gothelp,new-string-address) % 017
    % IF program not able to provide new string, treat as ABORT % 032
    changestring=10005B, % NOTE, arg2=old addr, arg3=new addr % 018
    % NOTE issued by low level string handlers after they got a new
    string address % 031
    saroverflow=10006B, % ABORT, SAR string overflow % 034
    % generated by runtime routines when SAR overflows % 035
    % perhaps later treated like other strings % 036
    stkoverflow=10007B, % ABORT, program defined stack overflow % 037
    stkunderflow=10010B, % ABORT, program defined stack underflow % 038
    listspace=10011B; % ABORT, list allocation zone full % 054
% error type codes (from (nls,xl10runtime,sysrcv) ) % 040
% error type codes are greater than 20000 octal % 041
% this is the first argument passed to recover procedure and
indicates the type of error involved % 042
DECLARE CONSTANT 043
    programbug=20001B, % a program bug involved % 044
    % stkoverflow= general stack overflow -- use same symbol defined
    above % 046
    uncaughtabort=20002B; % uncaught ABORT % 045

```


LOADER

< NLS, LOADER.NLS;2, >, 8-DEC-76 14:00 JDH ;;;;(NLS, LOADER.NLS;2,),
 22-MAR-74 05:32 KEV ;

```
(loader) FILE % L10 <dornbush>loader %% (L10,) (dornbush,loader.rel,) %
02
% GENERAL DOCUMENTATION % 03
% loading sequence: 04
1) load this file with L10RUN starting at 140B and save it. 05
2) using that loader, load this file again with L10RUN as
follows: 06
    /764777S 07
    /757777D 08
    /M 09
    /S 010
    LOADER.REL 011
    L10RUN.REL 012
    $ 013
3) fix symbol table by setting $i-1 _ 116B, and do init$G 014
4) SSAVE 761 thru 764. Delete first loader (1). % 015
% DECLARATIONS % 016
UND-OK 017
DECLARE stable[513]; 018
% symbol table is built from stable downward. Page above
stable is used for pmap-ing input file % 019
% hash table is built from hashtabaddr downward and may be
relocated using the /<address>h switch % 020
SET 021
symloc=116B, l10skz=50, uuo=41B, 022
stadr=120B; 023
DEFINE 024
hashsize=359#, undefinedmask=1B31#, hashtabaddr=700000B#,
definedtype=1#, undefinedtype=2#; 025
% HASHED SYMBOL TABLE: a hash table of length hashsize is
based at location phash. Each element of the hash table is a
pointer to a chained list of single word hash table entries for
all symbols on the same hash chain. Each hash table entry (1
word) contains two pointers. The LH points to the symbol table
entry (in the DDT table or in the undefined table) and the RH
is the pointer to the next entry on the hash chain. Hashbot is
a low water mark for allocating space for hash table entries
and undefned symbol table entries. The space used by undefined
symbol table entries is recovered for reuse by linking on a
chain beginning with ulink % 026
REGISTER 027
a2=13, a1=12, a3=14, r3=3, r4=4, a4=15, r1=1, r2=2, r5=5, r6=6,
pointr=8, rlcwd=7; 028
% pointr points into input page, rlcwd contains relocation
bits for a block. These are kept in registers throughout
loading of a file % 029
% registers zero through six are occasionally used for
executing code - see look, fixup, undef % 030
EXTERNAL sysovr; 031
DECLARE 032
blktyp, blktsiz, wdnun, jfn, lc, rlc, stp, stpst, 033
sflg, mflg, pname, ploc, psym, ulink, 034
ary[6], jfntab=(1B11, 377777377777B, 0,0,0,0,0,0), 035
dsprel="REL", l10sk[l10skz], string[10], sp, altfg, 036
```

```

initlptr, corpg, filpg, word, sr2, sr3, phash, hashbot,
hashtop,                                037
errorn, loadub, symb, nlsflag, nlslimit; 038
SET EXTERNAL                              039
gtjfn=20B, pmap=56B, haltf=170B, rfcoc=112B, sfcoc=113B, 040
sout=53B, bout=51B, nout=224B, nin=225B,          041
bin=50B, reset=147B, openf=21B, sevec=204B,       042
sin=52B, clzff=34B, bkjfn=42B, closf=22B, pbin=73B, pbout=74B; 043
REF phash;                                  044
(rec2) RECORD symb[32], codbit[4];             045
(rec3) RECORD xxx[24], byte[6], bits[6];       046
% INITIALIZATION and PARSING ROUTINES %        047
(main) PROCEDURE;                            048
  (envect): % entry vector %                  049
    GOTO start; GOTO rstart;                   050
  (init): % initialize here before ssave %     051
    jftab[5] _ $dsprel;                        052
    binwd.LH _ $a1; binwd.RH _ $binloc; % setup user JSYS % 053
    reloc.LH _ $a2; reloc.RH _ $reloc1;        054
    corpg _ 760B;                              055
    initlptr.LH _ -1000B; initlptr.RH _ corpg*1000B; 056
    r1 _ 4B5; r2 _ 2B6+$envect;                057
    !JSYS sevec; !JSYS haltf;                  058
  (start): % starting location %               059
    !JSYS reset; r1 _ 4B5; !JSYS clzff;        060
    !MOVNI 9,110skz; !HRL 9,9; !HRR1 9,110sk; 061
    r1 _ 100B; !JSYS rfcoc; r2 _ r2 .A 637777777777777B; %^a% 062
    r3 _ r3 .A 777477777777777B; %^w% !JSYS sfcoc; 063
    uuo _ popjsp;                              064
    command()                                  065
  END.                                         066
(command) PROCEDURE;                           067
  % command parser %                           068
  stp _ stpst _ $stable; mflg _ sflg _ altfg _ 0; 069
  nlsflag _ FALSE; % normal TENLDR entry %     070
  % initialize hash table %                    071
    hashtop _ hashtabaddr;                     072
    &phash _ hashtop-hashsize; hashbot _ &phash-1; 073
    FOR r1_0 UP UNTIL >= hashsize DO phash[r1] _ 0; 074
  rlc _ 140B; ulink _ 0;                       075
  loadub _ 777777B;                            076
  errorn _ 0;                                  077
  (rstart): % reenter location - reset only stack and altfg % 078
  altfg _ 0;                                   079
  !MCVNI 9,110skz; !HRL 9,9; !HRR1 9,110sk; 080
  LOOP BEGIN                                  081
    r1 _ '*'; !JSYS pbout;                     082
    collsr(); % collect string from user %      083
    CASE ^sp OF                                084
      =/: readsw();                             085
      =37B: NULL;                              086
      =0: NULL;                                087
    ENDCASE BEGIN                              088
      getjfn();                                089
      loadfl() END;                            090

```

```

        IF altfg THEN EXIT                                091
        END;                                             092
done();                                                093
nlslimit _ stp+1; %save limit away for NLSLOADER %    094
cr(); !JSYS haltf;                                     095
(sysovr): r1 _ $"L10 STACK OVERFLOW "+1 .V 18M6;     096
!JSYS 76B%psout%; !JSYS haltf                         097
END.                                                  098
(collsr) PROCEDURE;                                    099
% collect string from user - term. on cr, ca, alt %   0100
LOCAL x;                                              0101
sp _ $string .V 4407B8;                                0102
LOOP CASE (x_binchr()) OF                             0103
  =CA: BEGIN cr(); EXIT END;                          0104
  =37B: EXIT;                                         0105
  =12B: EXIT;                                         0106
  =15B: NULL;                                         0107
  =ALT: BEGIN BUMP altfg; cr(); EXIT END;            0108
  =1: %^A% sp_bakstr(sp,$string);                   0109
  =27B: %^W% BEGIN cr(); GOTO rstart END;            0110
  ENDCASE ^sp_x;                                      0111
^sp _ 0; sp _ $string .V 4407B8;                      0112
RETURN END.                                           0113
(bakstr) PROCEDURE(sptr,a);                            0114
% backup sptr one char and echo to tty %              0115
% for 7 bit chars only %                              0116
IF sptr= a .V 4407B8 OR sptr.RH<a THEN BEGIN         0117
  r1 _ 7 %bell%; !JSYS pbout END                     0118
ELSE BEGIN                                            0119
  r1 _ ^\; !JSYS pbout; !LDB 1,sptr; !JSYS pbout;   0120
  IF sptr.bits<29 THEN BEGIN                          0121
    sptr.bits _ sptr.bits + sptr.byte END            0122
  ELSE BEGIN                                          0123
    sptr _ sptr-1; sptr.bits _ 1 END                 0124
  END;                                               0125
RETURN(sptr) END.                                     0126
(binchr) PROCEDURE;                                    0127
!JSYS pbin; RETURN(r1) END.                           0128
(getjfn) PROCEDURE;                                    0129
r1 _ $jfnTAB; r2 _ $string .V 4407B8;                0130
IF NOT SKIP !JSYS gtjfn THEN error(r1);               0131
jfn _ r1; r1 _ r1 .A 18M;                             0132
r2 _ 4400002B5;                                       0133
IF NOT SKIP !JSYS openf THEN error(r1);               0134
wDnum _ filpg _ pointr _ lc _ 0;                      0135
RETURN END.                                           0136
(readsw) PROCEDURE;                                    0137
% parse switch and set flags, etc. %                  0138
LOCAL x, v, count, symval, proc, char;               0139
REF proc;                                             0140
x _ sp;                                               0141
CASE ^sp OF                                           0142
  ='S, ='s: sflg_1;                                    0143
  ='M, ='m: mflg_1;                                    0144
  ='W, ='w: sflg_0;                                    0145
  ='C, ='c: % call a process function %              0146

```

```

BEGIN                                                    0147
symval _ 0;                                           0148
% read proc name and convert to radix 50 value %     0149
  FOR count _ 1 UP UNTIL > 6 DO                       0150
    BEGIN                                              0151
      CASE char _ ^sp OF                              0152
        IN ['A', 'Z']: char _ char - 54;             0153
        IN ['a', 'z']: char _ char - 86;             0154
        IN ['0', '9']: char _ char - 47;             0155
        = SP: REPEAT CASE; % ignore spaces %          0156
      =0: EXIT LOOP;                                  0157
    ENDCASE BEGIN                                     0158
      werr( $"illegal function name");                0159
      RETURN; END;                                    0160
      symval _ symval*50B + char;                     0161
    END;                                               0162
% look up the symbol in the hash table %             0163
  IF NOT (y _ look( symval .V 4B10 ) ) THEN          0164
    BEGIN werr( $"proc not found" ); RETURN; END;    0165
% get addr of proc and call it, passing as arguments the
current bounds on the symbol table %                 0166
  &proc _ [y].LH;                                     0167
  &proc _ proc[1];                                    0168
  proc( stp, stpst );                                 0169
END;                                                  0170
ENDCASE BEGIN                                         0171
r1 _ x; r3 _ 8;                                       0172
IF NOT SKIP !JSYS nin THEN error(r3);                0173
CASE .r1 OF                                           0174
  ='E', ='e: [r2] _ $nlsload; % setup link for NLS%  0175
  ='H', ='h: IF hashbot = &phash -1 THEN % set new hash
table area %                                         0176
    BEGIN                                             0177
      hashtop _ r2;                                   0178
      &phash _ hashtop - hashsize;                   0179
      hashbot _ &phash -1;                           0180
      FOR r2 _ 0 UP UNTIL >= hashsize DO phash[r2] _ 0; 0181
    END;                                              0182
  ='O', ='o: rlc _ r2;                                0183
  ='R', ='r: [r2].LH _ rlc; % next free location %   0184
  ='S', ='s: IF stp=stpst THEN stpst _ stp _ r2;    0185
ENDCASE BEGIN ws($"illegal switch - ignored"); cr()  0186
END;                                                  0187
RETURN END.                                           0188
(done) PROCEDURE;                                     0189
% finish up tasks: undefined symbols printed, map, blt symbol
table if desired %                                   0190
printud();                                           0191
IF stpst=$stable THEN BEGIN                           0192
  r1 _ ((rlc+1000B) .A 9M3); r1.LH _ stp+1;          0193
  symloc _ r1.RH; symloc.LH _ (stp-stpst);           0194
  r2 _ r1.RH + stpst-stp;                             0195
  !BLT 1,0(2) END                                     0196

```

```

ELSE BEGIN                                0197
    symloc _ stp+1; symloc.LH _ (stp-stpst) END; 0198
IF mflg THEN printm();                    0199
RETURN END.                                0200
% SYMBOL TABLE MANIPULATION ROUTINES %    0201
(look) PROCEDURE(s);                       0202
% lookup global symbol s in regular symbol table. Return
address of pointer word in hash table %    0203
% registers r1-r3 are used in the search loop as follows 0204
    r1 pointer to entry in hash table chain 0205
    r2 pointer to symbol table entry        0206
    r3 local copy of symbol value %        0207
% hash the symbol by using integer divide % 0208
    r3 _ s;                                 0209
    r1 _ IF r3 >= 0 THEN r3 ELSE -r3; % make sure its positive % 0210
    !IDIVI r1,hashsize; % remainder to r2 % 0211
% set initial pointer to hash table entry % 0212
    r1 _ phash[r2];                          0213
% loop down the hash chain, looking for the symbol % 0214
(lookuploop):                              0215
    !JUMPE    r1,lookupdone; % jump if hash chain exhausted % 0216
    !HLRZ    r2,0(r1); % fetch pointer do symbol
table entry %                               0217
    !CAMN    r3,0(r2);% compare with symbol in symbol
table %                                       0218
    !JRST    lookupdone; % found the symbol % 0219
    !HRRZ    r1,0(r1); % link to next entry in hash
table %                                       0220
    !JRST    lookuploop; % try again %       0221
(lookupdone):                              0222
    RETURN (r1);                             0223
END.                                         0224
(ddtlookup) PROCEDURE(s);                   0225
% this routine looks up s in the part of the symbol table built
by NLSLOAD. It returns a pointer to the symbol table entry if
found. The current value of stp and the global value nslimit
(saved in routine command when loading the NLS system) are used
to define the current bounds of the symbol table % 0226
LOCAL                                       0227
    ptr; % pointer to the symbol table entry % 0228
ptr _ stp + 1;                              0229
WHILE ptr < nslimit DO                      0230
    BEGIN                                    0231
        IF [ptr] = s THEN RETURN(ptr);      0232
        ptr _ ptr + 2;                       0233
    END;                                      0234
RETURN (FALSE);                             0235
END.                                         0236
(henter) PROCEDURE(s,a,type);               0237
% this routine makes a symbol table entry in the symbol table
and links the entry into the hash table
the leftmost bit of the value word is set to TRUE if the
symbol table entry is undefined %          0238

```



```

LOCAL                                                    0240
  x, % chain link in the hash chain %                    0241
  y; % pointer to the symbol table entry %               0242
% check for an unexpected value %                        0243
  IF a < 0 OR s.codbit = 0 THEN                          0244
    BEGIN % we're in deep shit, leftmost bit already in use % 0245
      werr( $"LOADER ERROR- bad call to henter for " ); 0246
      wsym(s,a); cr();                                   0247
      RETURN;                                           0248
    END;                                                 0249
% set ptr to new entry into y (depends on type) and make new
symbol table entry %                                    0250
  IF type = undefinedtype THEN                          0251
    BEGIN                                                0252
      CASE nlsflag OF                                    0253
        =FALSE:                                         0254
          IF (y _ ulink) = 0 THEN                        0255
            BEGIN                                        0256
              y _ hashbot-1; % allocate a new two word entry % 0257
              hashbot _ hashbot - 2;                    0258
            END                                          0259
          ELSE ulink _ [ulink];                          0260
        =TRUE:                                           0261
          % We must make the undefined symbol table entry in
          the DDT symbol table even though it is in the wrong
          block so that we can find it and fix it up later % 0262
          BEGIN                                          0263
            y _ stp - 1;                                  0264
            stp _ stp - 2;                               0265
          END;                                           0266
        ENDCASE;                                        0267
      % make an entry at address y %                      0268
      [y+1] _ a .V undefinedmask;                       0269
      [y] _ s;                                          0270
    END;                                                 0271
  ELSE                                                    0272
    BEGIN % make entry in normal ddt smb ol table %     0273
      y _ stp - 1;                                       0274
      entr(s,a); % entry is made at location stp %      0275
    END;                                                 0276
% make entry in the hash table if in use %              0277
  IF NOT nlsflag THEN                                    0278
    BEGIN                                                0279
      % hash the symbol %                                 0280
      r1 _ IF s >= 0 THEN s ELSE -s; % make sure its
      positive %                                        0281
      !IDIVI r1,hashsize; % result to r2 %              0282
      % create a link in the hash table %                0283
      x _ phash[ r2 ];                                    0284
      phash[ r2 ] _ hashbot;                             0285
      % make a new entry in the hash table %             0286
      [hashbot].LH _ y;                                   0287
      [hashbot].RH _ x;                                   0288
    END;

```

```

        hashbot _ hashbot - 1;          0289
    END;                                0290
RETURN;                                0291
END.                                    0292
                                        0293
(entr) PROCEDURE(s,a);                 0294
    % enter symbol s in symbol table, value a % 0295
    IF stp<=symlb THEN RETURN;         0296
    [stp-1] _ s; [stp] _ a; stp _ stp-2; 0297
    RETURN END.                         0298
(undef) PROCEDURE(s,a);                0299
    % handles global request symbols - except deferred global
    stuff. IF it's defined, fix it up, else enter in undefined
    table (if possible) %              0300
    LOCAL x,y,p;                        0301
    IF a<0 THEN BEGIN                  0302
        werr($"global request bit zero on -improper linking
        done");                          0303
        cr() END;                       0304
    s.codbit _ 1; % make the undefined global into a global symbol
    %                                    0305
    CASE nlsflag OF                    0306
        =FALSE: % normal TENLDR stuff % 0307
            y _ IF (x_look(s)) THEN [x].LH ELSE 0; 0308
        =TRUE: % nlsloader in use %     0309
            BEGIN                       0310
                x _ 0;                   0311
                IF NOT (y_ddtlook(s)) THEN REPEAT CASE (FALSE); 0312
                END;                     0313
            ENDCASE;                     0314
    % by the time we get to here, y points to the existing symbol
    table entry, and x points to a hash table entry if it is being
    used %                               0315
    IF y THEN                           0316
        BEGIN % symbol already in table % 0317
            IF [y+1] >= 0 THEN fixup(a,[y+1]) 0318
            ELSE                           0319
                BEGIN                     0320
                    % have found an undefined symbol table entry % 0321
                    p _ a; % append to (already existing) undefined link% 0322
                    %WHILE [ p ] .A 18M # 0 DO p_[p];% %find end of chain% 0323
                    % execute this in registers ! % 0324
                    !MOVSI a4,undef1; !BLT a4,3; 0325
                    a2 _ p; !JRST 1; % setup a2 and go % 0326
                    (undef1): !MOVEI a2,0(a1); !HRRZ a1,0(a2); !JUMPN a1,0; 0327
                    !JRST undef2;         0328
                    (undef2):            0329
                    p _ a2;               0330
                    [p].RH _ [y+1];       0331
                    [y+1].RH _ a;         0332
                END                        0333
            END                            0334
        ELSE henter(s,a.RH,undefinedtype); 0335

```

```

RETURN;                                0336
END.                                    0337
(globalok) PROCEDURE(s,a);              0338
% check symbol s for global def: mulitple definition results in
% message on tty (unless same value), undef symbol table gets
% searched for reference to s, fixups are done if necessary.%
LOCAL x, y, ht;                          0339
% look up symbol s in the appropriate tables set ht to TRUE if
% found in the loader's hash symbol table. %
ht _ TRUE;                               0340
CASE nlsflag OF                           0341
  =FALSE: % normal TENLDR stuff %        0342
    y _ IF (x_look(s)) THEN [x].LH ELSE 0; 0343
  =TRUE: % nlsloader in use %            0344
    BEGIN                                 0345
      IF NOT (y_ddtlook(s)) THEN          0346
        REPEAT CASE (FALSE);             0347
      ht _ FALSE;                         0348
    END;                                  0349
ENDCASE;                                  0350
IF y THEN                                  0351
  BEGIN % symbol already in table %      0352
    IF [y+1] >= 0 THEN                    0353
      BEGIN % symbol previously defined % 0354
        % if we are called via the nlsloader, then we will define
        % a new external symbol %        0355
        IF [y+1]#a THEN                   0356
          IF NOT nlsflag THEN              0357
            BEGIN % multiply defined symbol % 0358
              werr($"multiply defined global "); 0359
              wsym(s,a);                   0360
              ws($" (was and is =) "); wn([y+1]); cr(); 0361
            END                             0362
          ELSE                              0363
            henter(s,a,definedtype);      0364
          END                               0365
        ELSE % symbol is in table, but undefined % 0366
          BEGIN                             0367
            fixup([y+1].RH,a);             0368
            CASE nlsflag OF                 0369
              =FALSE:                       0370
                BEGIN % fix up undefined symbols % 0371
                  % the entry must be deleted from the hash table and
                  % moved to the normal DDT symbol table, the space
                  % recovered is linked onto the chain ulink % 0372
                  [x].LH _ stp - 1; % fixup entry in hash table % 0373
                  entr(s,a); % put symbol entry into DDT table % 0374
                  [y] _ ulink; % link freed entry to ulink chain % 0375
                ulink _ y;                  0376
              END;                          0377
            =TRUE:                          0378
              CASE ht OF                    0379
                =FALSE: % symbol found in local ddt table % 0380

```

```

[+1] _ a; % stick real value in ddt symbol
table entry %                                0383
=TRUE: % symbol found in loader's hash table %
                                                0384
        henter(s,a,definedtype);              0385
    ENDCASE;                                  0386
ENDCASE;                                     0387
END;                                          0388
END                                           0389
ELSE henter(s,a,definedtype);                0390
RETURN;                                       0391
END.                                          0392
% LOADER PROCESSING ROUTINES %                0393
(loadf1) PROCEDURE;                           0394
% main file-loading loop. parse block type %  0395
LOCAL w;                                       0396
LOOP BEGIN                                     0397
    w _ !binwd; blktyp _ w.LH; blksiz _ w.RH;  0398
    CASE blktyp OF                             0399
        =1: code();                             0400
        =2: syms();                             0401
        =4: skip();                             0402
        =5: BEGIN endblk(); RETURN END;         0403
        =6: pgname();                           0404
        =7: stblk();                             0405
        =10B: fixblk();                          0406
    ENDCASE BEGIN                              0407
        wdnun _ (filpg-1)*1000B + pointr.RH - corpg*1000B;
                                                0408
        werr($"format error, word "); wn(wdnun);  0409
        ws($" block type "); wn(blktyp); cr();    0410
        RETURN END END                          0411
    END.                                         0412
(code) PROCEDURE;                              0413
% read a code block - load into core %         0414
LOCAL c;                                       0415
LOOP BEGIN                                     0416
    rlcwd _ !binwd;                             0417
    c _ MIN(blksiz,18); blksiz _ blksiz-c;     0418
    lc _ !reloc.RH;                             0419
    IF lc IN [0,17B] OR lc+c >= loadub THEN    0420
        RETURN(werr($"loading outside of allowed area"));
                                                0421
    % LOOP BEGIN                                0422
        IF (c_c-1)<=0 THEN EXIT;                 0423
        [lc] _ !reloc;                           0424
        BUMP lc                                   0425
        END; %                                   0426
    % execute this loop in registers %          0427
    !MOVSI a4,code1; !ADDI a4,2; !BLT a4,5;     0428
    !MOVN r6,c; !MOVSI r6,1(r6); !HRR r6,lc; !JRST 2; % setup r6
and go %                                       0429
    (code1):                                    0430
        !reloc; !MOVEM r1,0(r6); !AOBJN r6,2; !JRST code2;
                                                0431
    (code2):                                    0432
    IF blksiz=0 THEN RETURN                     0433
    END                                           0434

```

```

END. 0435
(syms) PROCEDURE; 0436
% read symbol block - parse symbol types % 0437
LOCAL c, w1, w2; 0438
LOOP BEGIN 0439
  rlcwd _ !binwd; 0440
  c _ MIN(blksiz,18); blksiz _ blksiz-c; 0441
  LOOP BEGIN 0442
    IF (c_c-2)<0 THEN EXIT; 0443
    w1 _ !reloc; 0444
    w2 _ !reloc; 0445
    CASE w1.codbit OF 0446
      =1: %global% globalok(w1,w2); 0447
      =14B: %global req.% udef(w1,w2); 0448
      IN [2,3]: IF sflg THEN henter(w1,w2,definedtype); 0449
    ENDCASE henter(w1,w2,definedtype) 0450
  END; 0451
  IF blksiz=0 THEN RETURN 0452
  END 0453
END. 0454
(fixup) PROCEDURE(chain, a); 0455
% fixup chain with value a (right half) % 0456
LOCAL x; 0457
%DO BEGIN 0458
  x _ [chain]; [chain].RH _ a; 0459
  chain _ x.RH END 0460
  WHILE chain.RH # 0;% 0461
% execute the DO loop in the registers % 0462
!MOVSI a4,f1; !BLT a4,4; % BLT into registers % 0463
a1 _ chain; a2 _ a; !JRST 0; % setup and go % 0464
(f1): 0465
  !MOVE a3,0(a1); !HRRM a2,0(a1); !MOVEI a1,0(a3); !JUMPN 0466
  a1,0; 0467
  !JRST sysrtn; 0468
NULL %return is in above loop % END. 0469
(fixuplh) PROCEDURE(chain, a); 0470
% fixup chain with value a (left half) % 0471
LOCAL x; 0472
DO BEGIN 0473
  x _ [chain]; [chain].LH _ a; 0474
  chain _ x.LH END 0475
  WHILE chain.RH # 0; 0476
RETURN END. 0477
(fixblk) PROCEDURE; 0478
% read fixup block - handles -1 (left half fixup) entries % 0479
LOCAL p, c, w, f; 0480
f _ 0; 0481
LOOP BEGIN 0482
  rlcwd _ !binwd; c _ 0; 0483
  LOOP BEGIN 0484
    IF blksiz=0 THEN RETURN; 0485
    IF c=18 THEN EXIT; 0486
    w _ !reloc; 0487
    blksiz _ blksiz-1; 0488
    IF w=-1 THEN f_1 0488

```

```

        ELSE IF f THEN BEGIN fixuplh(w.LH, w.RH); f_0 END      0489
        ELSE fixup(w.LH, w.RH) END                            0490
    END                                                         0491
END.                                                           0492
(skip) PROCEDURE;                                           0493
% skip over block taking no action %                          0494
LOCAL c;                                                       0495
LOOP BEGIN                                                    0496
    !binwd; c _ 0;                                           0497
    LOOP BEGIN                                                0498
        IF blksiz=0 THEN RETURN;                             0499
        IF c=18 THEN EXIT;                                   0500
        !binwd; BUMP c; blksiz_blksiz-1                     0501
    END                                                         0502
END                                                             0503
END.                                                           0504
(pgname) PROCEDURE;                                          0505
% enter program name and save necessary stuff for endblk %  0506
LOCAL w;                                                       0507
!binwd;                                                         0508
w _ !binwd;                                                    0509
IF blksiz=2 THEN !binwd;                                       0510
entr(w,lc+rlc);                                                 0511
pname_w; ploc_stp+2; psym_stp+1;                               0512
RETURN END.                                                    0513
(stblk) PROCEDURE;                                           0514
% set starting location in 120B %                               0515
rlc _ !binwd;                                                  0516
stadr.RH _ !reloc;                                             0517
RETURN END.                                                    0518
(endblk) PROCEDURE;                                          0519
% end block: correct reloc const. an fixup prog. name info for
DDT%                                                            0520
rlc _ !binwd;                                                  0521
rlc _ !reloc.RH;                                               0522
entr(0,rlc); % program break %                                0523
[ploc].LH _ stp-ploc;                                          0524
IF rlc IN [stp-2,stpst) THEN BEGIN                             0525
    werr($"symbol table and code overlap"); cr() END;         0526
[initlptr] _ 0; % creates private copy of file page so can
close %                                                         0527
IF NOT SKIP !closf(jfn) THEN error(r1);                       0528
RETURN END.                                                    0529
(xbinwd) PROCEDURE;                                          0530
(binloc): % called by !binwd (user JSYS) %                    0531
% return is kept in register a1 %                              0532
!MOVE r1,1(pointr); !AOBJN pointr,0(a1); % return word %    0533
    sr2 _ r2; sr3 _ r3; % save r2, r3 %                       0534
    !pmap(jfn*1B6+filpg,4B11+corpq,100400B6);                0535
    !AOS filpg; pointr _ initlptr;                            0536
    r2 _ sr2; r3 _ sr3;                                       0537
    !MOVE r1,0(pointr); !JRST 0(a1)                           0538
END.                                                           0539
(xreloc) PROCEDURE; % user JSYS reloc to read next word and
relocate it per rlcwd %                                        0540

```

```

(reloc): % relocate word according to rlcwd and return value
in word%                                0541
% called with !reloc - return kept in a2 % 0542
!binwd;                                  0543
!MOVS a3,rlc;                             0544
!TLNE rlcwd,4B5; % left half relocation % 0545
!ADD r1,a3;                                0546
!MOVSS 0,r1; % exchange halves %          0547
!TLNE rlcwd,200000B; % right half relocation % 0548
!ADD r1,a3;                                0549
!MOVS r1,r1; % get it right side up %     0550
!LSH rlcwd,2; % shift rlcwd for next input word % 0551
!JRST 0(a2); % return %                   0552
END.                                       0553
% I/O UTILITY ROUTINES %                  0554
(cr) PROCEDURE;                            0555
!pbout(15B); !pbout(12B); RETURN END.     0556
(error) PROCEDURE(x);                      0557
r1 _ 101B; r2 _ 4B11+x; r3 _ 0;          0558
!JSYS 11B%erstr%; !JFCL; !JFCL; cr();     0559
BUMP errorn;                               0560
GOTO rstart END.                           0561
(printm) PROCEDURE;                         0562
% print map (start location and length) % 0563
LOCAL x;                                    0564
cr(); ws("$MAP: ");                          0565
x _ stpst+2;                                0566
WHILE (x_x-2)>stp DO                          0567
  IF [x-1] AND [x-1].codbit=0 THEN BEGIN     0568
    wsym([x-1],[x].RH);                      0569
    ws("$ to "); wn( [ [x].LH+x+2] ); cr() END; 0570
  cr();                                       0571
  ws("$symbol table "); wn(symloc.RH);       0572
  ws("$ to "); wn(symloc.RH-1-(symloc.LH .V 18M6)); 0573
  cr();                                       0574
  ws("$last free adr = "); wn(rlc);         0575
  RETURN END.                                0576
(printud) PROCEDURE;                        0577
% list undefined symbols %                  0578
LOCAL ptr, hptr, i, sw;                    0579
% search through the hash symbol table looking for any
undefined globals %                        0580
sw _ FALSE;                                 0581
CASE nlsflag OF                             0582
  =FALSE:                                    0583
    FOR i _ 0 UP UNTIL >= hashsize DO       0584
      BEGIN % link down the i'th hash chain % 0585
        hptr _ phash[ i ];                  0586
        WHILE hptr DO                       0587
          BEGIN                              0588
            ptr _ [hptr].LH; % get pointer to 2 word sym tab
            entry %                          0589
            IF [ptr+1] < 0 AND [ptr].codbit = 1 THEN 0590
              BEGIN                          0591
                IF NOT sw THEN              0592
                  BEGIN                      0593

```

```

        sw _ TRUE;                                0594
        werr("$*** Undefined Globals ***"); cr(); 0595
        END;                                       0596
        wsym([ptr], [ptr+1].RH);                  0597
        END;                                       0598
        hptr _ [hptr].RH; % link to next entry in the hash
        chain %                                    0599
        END;                                       0600
    END;                                           0601
=TRUE:                                           0602
    BEGIN % search only the part of the table used for
    NLSLOADER %                                    0603
    ptr _ stp + 1;                                  0604
    WHILE ptr < nlslimit DO                        0605
        BEGIN                                       0606
            IF [ptr+1] < 0 AND [ptr].codbit = 1 THEN 0607
                BEGIN                               0608
                    IF NOT sw THEN                 0609
                        BEGIN                       0610
                            sw _ TRUE;            0611
                            ws( "$*** Undefined Globals ***"); cr(); 0612
                            END;                 0613
                            wsym([ptr], [ptr+1].RH); 0614
                            END;                 0615
                        ptr _ ptr + 2;            0616
                    END;                           0617
                END;                               0618
            ENDCASE;                               0619
            RETURN;                                0620
        END.                                       0621
(werr) PROCEDURE(s);                              0622
    % write error string and bump errorn %        0623
    BUMP errorn;                                   0624
    ws(s);                                         0625
    RETURN END.                                    0626
(wn) PROCEDURE(n);                                0627
    r1 _ 101B; r2 _ n; r3 _ 8;                   0628
    IF NOT SKIP !JSYS nout THEN error(r3);       0629
    RETURN END.                                    0630
(ws) PROCEDURE(s);                                0631
    % do sout on tty %                            0632
    r1 _ 101B; r2 _ (s+1) .V 18M6; r3 _ 0;      0633
    !JSYS sout; RETURN END.                      0634
(wsym) PROCEDURE(s,a);                            0635
    % write symbol s on tty, followed by value a % 0636
    LOCAL j;                                       0637
    cr();                                          0638
    j _ 6; r1 _ s.symb1;                          0639
    WHILE (j-j-1) >= 0 DO BEGIN                  0640
        !IDIVI 1,50B;                             0641
        ary[j] _ (CASE r2 OF                     0642
            =47B: "%";                            0643
            =46B: "$";                            0644
            =45B: ".";                            0645
            >12B: r2+66B;                          0646

```



```

        >0; r2+57B;                                0647
        ENDCASE SP) END;                            0648
    r1 _ 101B; r2 _ $ary .V 4444B8; r3 _ -6;        0649
    !JSYS sout; r1 _ SP; !JSYS pbout;              0650
    wn(a); RETURN END.                              0651
% NLS LOADER INTERFACE ROUTINES %                  0652
(nlsload) PROCEDURE(njfn, nrlc, nub);              0653
% this procedure links TENLDR with NLS for loading user
% programs. njfn is a open REL file. nrlc is the location to
% start loading in. nub is the upper bound for safe loading.
% the symbol table pointer in 116 is used and changed, and the LH
% of 120 is used for the symbol table bound %      0654
% The switch /<adr>E is used to store the address of this
% routine in location E when NLS is loaded. The loader is then
% saved with NLS %                                0655
LOCAL save1,save2; % save away regs 7 and 8 %      0656
save1 _ rlcwd; save2 _ pointr;                     0657
nlsflag _ TRUE; % indicates entry via NLSLOAD %   0658
mflg _ altfg _ 0; sflg _ 1;                        0659
jfn _ njfn; rlc _ nrlc; loadub _ nub;              0660
stp _ symloc.RH-1;                                  0661
stpst _ (stp - symloc.LH) .A 18M;                  0662
symlb _ stadr.LH;                                   0663
wddnum _ filpg _ pointr _ errorn _ lc _ 0;         0664
loadfl(); done();                                  0665
rlcwd _ save1; pointr _ save2; % restore regs 7 and 8 % 0666
RETURN(errorn, rlc) END.                            0667
% TERMINATING DECLARATIONS %                       0668
DECLARE binwd, reloc; % user JSYS words - must be >= 1000B % 0669
FINISH                                              0670

```

MINISYSGD

< NLS, MINISYSGD.NLS;4, >, 26-JUL-77 17:33 JDH ;;;;

(caddexp) (nls,adrmp,caddexp) 1D1	03157
(conspt) (nls,adrmp,conspt) 1E1	03167
(feedlt) (nls,adrmp,feedlt) 1E7A	03173
(gaddir) (nls,adrmp,gaddir) 1E3	03174
(gadname) (nls,adrmp,gadname) 1E2	03172
(gadnum) (nls,adrmp,gadnum) 1E6	03175
(gdlit2) (nls,adrmp,gdlit2) 1E5	03158
(getpr) (nls,adrmp,getpr) 1E4	03178
(lkupfast) (nls,adrmp,lkupfast) 1E9E	03166
(lnbafn) (nls,adrmp,lnbafn) 1C3	03163
(lnbfls) (nls,adrmp,lnbfls) 1C2	03170
(lnkbody) (nls,adrmp,lnkbody) 1C7	03156
(lnkcom) (nls,adrmp,lnkcom) 1C6	03165
(lnkdae) (nls,adrmp,lnkdae) 1C12	03169
(lnkend) (nls,adrmp,lnkend) 1C14	03181
(lnkfil) (nls,adrmp,lnkfil) 1C11	03182
(lnkfp) (nls,adrmp,lnkfp) 1C8	03155
(lnkhst) (nls,adrmp,lnkhst) 1C9	03164
(lnkprs) (nls,adrmp,lnkprs) 1C4	03154
(lnkpspc) (nls,adrmp,lnkpspc) 1C1	03180
(lnkstr) (nls,adrmp,lnkstr) 1C5	03162
(lnkusr) (nls,adrmp,lnkusr) 1C10	03159
(lnkvws) (nls,adrmp,lnkvws) 1C13	03171
(lookup) (nls,adrmp,lookup) 1E9B	03179
(namelook) (nls,adrmp,namelook) 1E9A	03161
(namingrp) (nls,adrmp,namingrp) 1E9F	03153
(nfstid) (nls,adrmp,nfstid) 1E8A	03160
(nmdr) (nls,adrmp,nmdr) 1E9D	03177
(setit) (nls,adrmp,setit) 1E7B	03176
(specreg) (nls,adrmp,specreg) 1E9C	03168

03869

(brkconnection) (nls,auxcod,brkconnection) 5I	03359
(cetcapp) (nls,auxcod,cetcapp) 3L	03376
(changcom) (nls,auxcod,changcom) 7A	03366
(cnvcrifteol) (nls,auxcod,cnvcrifteol) 3J	03374
(cnveoltcrlf) (nls,auxcod,cnveoltcrlf) 3K	03394
(crit) (nls,auxcod,crlf) 3I	03407
(daofrmt) (nls,auxcod,daofrmt) 9C	03411
(datfrmt) (nls,auxcod,datfrmt) 9D	03409
(delmkn) (nls,auxcod,delmkn) 8A	03365
(delmkr) (nls,auxcod,delmkr) 8B	03372
(dismes) (nls,auxcod,dismes) 3A	03393
(dismsg) (nls,auxcod,dismsg) 3C	03380
(dtfrmt) (nls,auxcod,dtfrmt) 9B	03370
(flgminit) (nls,auxcod,flgminit) 13H	03387
(fstatus) (nls,auxcod,fstatus) 11A	03378
(getdat) (nls,auxcod,getdat) 9A	03379
(gmtfset) (nls,auxcod,gmtfset) 9G	03361
(gofork) (nls,auxcod,gofork) 4L	03367
(goroot) (nls,auxcod,goroot) 4A	03375
(gps) (nls,auxcod,gps) 4B	03396
(gtadcall) (nls,auxcod,gtadcall) 9E	03403
(halt) (nls,auxcod,halt) 4J	03395
(idfrmt) (nls,auxcod,idfrmt) 9H	03397
(insmkr) (nls,auxcod,insmkr) 8C	03369
(iodaterr) (nls,auxcod,iodaterr) 4F	03399
(jctlcres) (nls,auxcod,jctlcres) 5H	03355
(joctlc) (nls,auxcod,joctlc) 5G	03363
(lockpage) (nls,auxcod,lockpage) 12B	03404
(meastart) (nls,auxcod,meastart) 13A	03377
(mscrfile) (nls,auxcod,mscrfile) 13F	03357
(msflname) (nls,auxcod,msflname) 13E	03362
(msgpsi) (nls,auxcod,msgpsi) 3F	03373
(msrecord) (nls,auxcod,msrecord) 13I	03401
(msrtcop) (nls,auxcod,msrtcop) 13B	03353
(mstypcheck) (nls,auxcod,mstypcheck) 13D	03371
(mswdstore) (nls,auxcod,mswdstore) 13G	03398
(nlrst) (nls,auxcod,nlrst) 4C	03356
(nmdlsset) (nls,auxcod,nmdlsset) 10A	03390
(pause) (nls,auxcod,pause) 4I	03384
(rerror) (nls,auxcod,rerror) 4D	03406
(rstconnection) (nls,auxcod,rstconnection) 5J	03368
(rubout) (nls,auxcod,rubout) 5A	03389
(rubsig) (nls,auxcod,rubsig) 5B	03354
(seemkr) (nls,auxcod,seemkr) 8D	03388
(settimer) (nls,auxcod,settimer) 3D	03408
(shutdis) (nls,auxcod,shutdis) 4K	03400
(stopline) (nls,auxcod,stopline) 5C	03412
(stoptimer) (nls,auxcod,stoptimer) 3E	03385
(strtmeasure) (nls,auxcod,strtmeasure) 13C	03413
(tenwheel) (nls,auxcod,tenwheel) 4E	03382
(thwfil) (nls,auxcod,thwfil) 4H	03360
(thwrfp) (nls,auxcod,thwrfp) 4G	03402
(timerpause) (nls,auxcod,timerpause) 3B	03358
(totentad) (nls,auxcod,totentad) 9F	03391
(trapc) (nls,auxcod,trapc) 5D	03386
(trapo) (nls,auxcod,trapo) 5E	03392

GAS2, 14-Feb-79 22:37

< NLS, MINISVSGD.NLS.4, > 3

(traps) (nls,auxcod,traps) 5F	03405
(typeas) (nls,auxcod,typeas) 3G	03383
(typech) (nls,auxcod,typech) 3H	03364
(uflow) (nls,auxcod,uflow) 6A	03381
(xnmdlset) (nls,auxcod,xnmdlset) 10B	03410
	03870

(astr) (nls,bconst,)	01962
(bcstr) (nls,bconst,)	01941
(bstr) (nls,bconst,)	01733
(cdotst) (nls,bconst,)	01661
(cdstr) (nls,bconst,)	01652
(crstr) (nls,bconst,)	01842
(cstr) (nls,bconst,)	01646
(dspjpt) (nls,bconst,)	01906
(dspno) (nls,bconst,)	01866
(dsprfm) (nls,bconst,)	01910
(dsprfs) (nls,bconst,)	01957
(dspstr) (nls,bconst,)	01657
(dspyes) (nls,bconst,)	01720
(dsys6) (nls,bconst,)	01779
(enlstr) (nls,bconst,)	01969
(escstr) (nls,bconst,)	01736
(estr) (nls,bconst,)	01757
(fstr) (nls,bconst,)	01930
(itpsig) (nls,bconst,)	01705
(gstr) (nls,bconst,)	01934
(hstr) (nls,bconst,)	01829
(jnlpw) (nls,bconst,)	01900
(kstr) (nls,bconst,)	01772
(lfstr) (nls,bconst,)	01892
(lstr) (nls,bconst,)	01724
(msavpt) (nls,bconst,)	01759
(msavqn) (nls,bconst,)	01820
(msavrp) (nls,bconst,)	01885
(msavws) (nls,bconst,)	01974
(msbavp) (nls,bconst,)	01796
(msbk21) (nls,bconst,)	01755
(msbuf1) (nls,bconst,)	01901
(mschrt) (nls,bconst,)	01801
(mscoun) (nls,bconst,)	01992
(mscurt) (nls,bconst,)	01878
(msdsys) (nls,bconst,)	01970
(msfini) (nls,bconst,)	01932
(mshdr1) (nls,bconst,)	01907
(mspmt0) (nls,bconst,)	01693
(msreal) (nls,bconst,)	01696
(msstar) (nls,bconst,)	01822
(msstre) (nls,bconst,)	01719
(mssys) (nls,bconst,)	01927
(mstr) (nls,bconst,)	02004
(mstsys) (nls,bconst,)	01944
(mswsiw) (nls,bconst,)	01669
(nstr) (nls,bconst,)	01843
(nulstr) (nls,bconst,)	01981
(ostr) (nls,bconst,)	01923
(pstr) (nls,bconst,)	02010
(qstr) (nls,bconst,)	01950
(rstr) (nls,bconst,)	01729
(sqstkn) (nls,bconst,)	01998
(sstr) (nls,bconst,)	01637
(tbstr) (nls,bconst,)	01966
(tstr) (nls,bconst,)	01915

(ubrstr) (nls,bconst,)		01708
(uslstr) (nls,bconst,)		01668
(ustr) (nls,bconst,)		01806
(ustr) (nls,bconst,)		01818
(vstr) (nls,bconst,)		01648
(werrsi) (nls,bconst,)		01902
(wstr) (nls,bconst,)		01924
(xstr) (nls,bconst,)		01816
(ystr) (nls,bconst,)		01945
(zstr) (nls,bconst,)		01939
(acor2) (nls,bconst,acor2) 30B1A4		01764
(acor4) (nls,bconst,acor4) 30B1A5		01940
(aigtyp) (nls,bconst,afgtyp) 27A4		01649
(alphbase) (nls,bconst,alphbase) 40F		01862
(alphoff) (nls,bconst,alphoff) 40G		01702
(apsda) (nls,bconst,apsda) 29A8		01808
(archost) (nls,bconst,archost) 34A1		01956
(arcistr) (nls,bconst,arcistr) 34B2		01904
(arcstr) (nls,bconst,arcstr) 34B1		01726
(assfil) (nls,bconst,assfil) 10F5		01628
(bbnbacc) (nls,bconst,bbnbacc) 34B4		01762
(bbnbhost) (nls,bconst,bbnbhost) 34A3		01699
(bfile) (nls,bconst,bfile) 8B1		01644
(bfree) (nls,bconst,bfree) 5A		01863
(bhl) (nls,bconst,bhl) 31A3		01734
(bintyp) (nls,bconst,bintyp) 23A1		01988
(blkprd) (nls,bconst,blkprd) 31A2		01723
(blksiz) (nls,bconst,blksiz) 10A14		01955
(blksuc) (nls,bconst,blksuc) 31A1		01883
(braname) (nls,bconst,braname) 24A8		01936
(brnchv) (nls,bconst,brnchv) 22A2		01651
(castr) (nls,bconst,castr) 30C1A1		01871
(ccfda) (nls,bconst,ccfda) 29A11		01639
(ccnda) (nls,bconst,ccnda) 29A10		01711
(ce) (nls,bconst,ce) 16A5		01995
(charv) (nls,bconst,charv) 22A8		01761
(chkpcf) (nls,bconst,chkpcf) 10F12		01953
(chrtyp) (nls,bconst,chrtyp) 23A2		01952
(chtyp) (nls,bconst,chtyp) 10B4		01841
(ckpag) (nls,bconst,ckpag) 5C		01663
(clisterr) (nls,bconst,clisterr) 8B2		01634
(cmptyp) (nls,bconst,cmptyp) 11B2		01916
(cnch2) (nls,bconst,cnch2) 30B1A3		01807
(comdir) (nls,bconst,comdir) 10E2		01745
(comtyp) (nls,bconst,comtyp) 23A4		01912
(contls) (nls,bconst,contls) 24A5		02012
(contnt) (nls,bconst,contnt) 24A3		01789
(corblk) (nls,bconst,corblk) 38B1G		01823
(corexr) (nls,bconst,corexr) 38B1F		01815
(corlqd) (nls,bconst,corlqd) 38B1D		01636
(corlge) (nls,bconst,corlge) 38B1E		01887
(cpcldelim) (nls,bconst,cpcldelim) 35B1		01683
(cpcrdelim) (nls,bconst,cpcrdelim) 35B2		01750
(crpgad) (nls,bconst,crpgad) 10D1		01714
(cs) (nls,bconst,cs) 16A4		01985
(de) (nls,bconst,de) 16A13		01879

(defbmode) (nls,bconst,defbmode) 18E1	01899
(defcolmax) (nls,bconst,defcolmax) 18H3	01914
(defcurcontext) (nls,bconst,defcurcontext) 18C1	01780
(defdbk) (nls,bconst,defdbk) 18E3	01752
(defdcolmax) (nls,bconst,defdcolmax) 18D1	01805
(defdind) (nls,bconst,defdind) 18E2	01732
(defdwrapcol) (nls,bconst,defdwrapcol) 18D2	01642
(defenlf) (nls,bconst,defenlf) 18N1	01680
(deffrrsize) (nls,bconst,deffrrsize) 18G2	01825
(defhmode) (nls,bconst,defhmode) 18F1	01769
(defhsize) (nls,bconst,defhsize) 18F2	01831
(defindcnt) (nls,bconst,defindcnt) 18H4	01903
(deflinmax) (nls,bconst,deflinmax) 18H2	01884
(defoffset) (nls,bconst,defoffset) 18H5	01630
(defpgsize) (nls,bconst,defpgsize) 18H1	01921
(defprompt) (nls,bconst,defprompt) 18I1	01715
(defqcolmax) (nls,bconst,defqcolmax) 18J1	01886
(defre2mode) (nls,bconst,defre2mode) 18K2	01787
(defrecmode) (nls,bconst,defrecmode) 18K1	01983
(defrjtchr) (nls,bconst,defrjtchr) 18L3	01846
(defrtjtab) (nls,bconst,defrtjtab) 18L2	01895
(defspftab) (nls,bconst,defspftab) 18L1	01874
(defsrsize) (nls,bconst,defsrsize) 18G1	01852
(defstu) (nls,bconst,defstu) 18M1	01776
(defsubstr) (nls,bconst,defsubstr) 18O1	01909
(deftb1) (nls,bconst,deftb1) 18H6	01963
(deftb2) (nls,bconst,deftb2) 18H7	01698
(deftb3) (nls,bconst,deftb3) 18H8	01774
(dfltvr) (nls,bconst,dfltvr) 10F15	01749
(dhtyp) (nls,bconst,dhtyp) 10B5	01706
(dirsfname) (nls,bconst,dirsfname) 35D1	01880
(down) (nls,bconst,down) 21B2	01631
(downstr) (nls,bconst,downstr) 21A1	01716
(dpybks) (nls,bconst,dpybks) 7A	01709
(dpycoc) (nls,bconst,dpycoc) 25A2	02011
(ds) (nls,bconst,ds) 16A12	01872
(dspbke) (nls,bconst,dspbke) 4B	01990
(dspblk) (nls,bconst,dspblk) 4A	01791
(dsys1) (nls,bconst,dsys1) 18P1A	01786
(dsys10) (nls,bconst,dsys10) 18P1J	01793
(dsys11) (nls,bconst,dsys11) 18P1K	01802
(dsys12) (nls,bconst,dsys12) 18P1L	01908
(dsys13) (nls,bconst,dsys13) 18P1M	02013
(dsys14) (nls,bconst,dsys14) 18P1N	01694
(dsys15) (nls,bconst,dsys15) 18P1O	01839
(dsys2) (nls,bconst,dsys2) 18P1B	01959
(dsys3) (nls,bconst,dsys3) 18P1C	01725
(dsys4) (nls,bconst,dsys4) 18P1D	01876
(dsys5) (nls,bconst,dsys5) 18P1E	01641
(dsys7) (nls,bconst,dsys7) 18P1G	01666
(dsys8) (nls,bconst,dsys8) 18P1H	01635
(dsys9) (nls,bconst,dsys9) 18P1I	01688
(dtbbas) (nls,bconst,dtbbas) 10A4	01775
(dtbm) (nls,bconst,dtbm) 10A5	01935
(edit) (nls,bconst,edit) 9A2	01919
(efree) (nls,bconst,efree) 5B	01931

(emreset) (nls,bconst,emreset) 36A3	01984
(emrtnfalse) (nls,bconst,emrtnfalse) 36A1	02006
(emsignal) (nls,bconst,emsignal) 36A2	01760
(expadr) (nls,bconst,expadr) 38B1B	01673
(extname) (nls,bconst,extname) 24A9	01898
(fbhdl) (nls,bconst,fbhdl) 10A7	01633
(fcor2) (nls,bconst,fcor2) 30B1A1	01976
(fcor4) (nls,bconst,fcor4) 30B1A2	01671
(fe) (nls,bconst,fe) 16A11	01670
(filev) (nls,bconst,filev) 22A6	01928
(flgname) (nls,bconst,flgfname) 35D2	01722
(fnormal) (nls,bconst,fnormal) 30A2	01654
(fontmax) (nls,bconst,fontmax) 30A1	01865
(forwdv) (nls,bconst,forwdv) 22A16	01643
(fs) (nls,bconst,fs) 16A10	01653
(fvrdchar) (nls,bconst,fvrdchar) 35A1	01993
(groupv) (nls,bconst,groupv) 22A3	02005
(grpadr) (nls,bconst,grpadr) 38B1A	01857
(grpname) (nls,bconst,grpfname) 35D3	01980
(grptyp) (nls,bconst,grptyp) 27A2	01830
(gtftyp) (nls,bconst,gtftyp) 10B2	01870
(hcopyv) (nls,bconst,hcopyv) 22A7	01869
(hdtyp) (nls,bconst,hdtyp) 10A9	01792
(he) (nls,bconst,he) 16A7	01782
(heurfil) (nls,bconst,heurfil) 10F6	01849
(hs) (nls,bconst,hs) 16A6	01771
(idnname) (nls,bconst,idnfname) 26A3	01967
(indtyp) (nls,bconst,indtyp) 27A1	01851
(inremsiteflag) (nls,bconst,inremsiteflag) 34C1	01838
(invisv) (nls,bconst,invisv) 22A12	01922
(irep) (nls,bconst,irep) 30B1A6	01855
(isichost) (nls,bconst,isichost) 34A7	01682
(isidhost) (nls,bconst,isidhost) 34A8	01655
(isiehost) (nls,bconst,isiehost) 34A6	01718
(isirlhost) (nls,bconst,isirlhost) 34A9	01845
(iupcase) (nls,bconst,iupcase) 41C	01707
(jnktyp) (nls,bconst,jnktyp) 10A12	01665
(jnlname) (nls,bconst,jnlname) 26A4	01996
(jnitime) (nls,bconst,jnitime) 33B1	01958
(jump) (nls,bconst,jump) 9A3	01770
(justfil) (nls,bconst,justfil) 10F7	01811
(l10nam) (nls,bconst,l10nam) 26A5	01742
(le) (nls,bconst,le) 16A3	01687
(levdown) (nls,bconst,levdown) 15A3	01859
(levsuc) (nls,bconst,levsuc) 15A1	01656
(levup) (nls,bconst,levup) 15A2	01697
(lfn) (nls,bconst,lfn) 16A1	01710
(linkv) (nls,bconst,linkv) 22A13	01778
(litv) (nls,bconst,litv) 22A5	01629
(lnk2err) (nls,bconst,lnk2err) 8A1E	01864
(lnk5err) (nls,bconst,lnk5err) 8A1E	01781
(lnk6err) (nls,bconst,lnk6err) 8A1F	01773
(lnk7err) (nls,bconst,lnk7err) 8A1G	01712
(lnk9err) (nls,bconst,lnk9err) 8A1I	01645
(lnkds1) (nls,bconst,lnkds1) 16A16	01692
(lowcase) (nls,bconst,lowcase) 41B	01758

(lpbug) (nls,bconst,lpbug) 30B1J	01701
(lpcline) (nls,bconst,lpcline) 30B1G	01893
(lpcoord) (nls,bconst,lpcoord) 30B1R	01856
(lpcscreen) (nls,bconst,lpcscreen) 30B1L	01894
(lpdline) (nls,bconst,lpdline) 30B1H	01690
(lpendstandout) (nls,bconst,lpendstandout) 30E1F	01813
(lpesc) (nls,bconst,lpesc) 30B1B	01860
(lpinline) (nls,bconst,lpinline) 30B1I	01982
(lpintter) (nls,bconst,lpintter) 30B1N	01640
(lpnocoor) (nls,bconst,lpnocoor) 30B1Q	01721
(lppbug) (nls,bconst,lppbug) 30B1K	01977
(lpposition) (nls,bconst,lpposition) 30B1D	01684
(lpresc) (nls,bconst,lpresc) 30B1C	01929
(lpreset) (nls,bconst,lpreset) 30B1M	01978
(lpstandout) (nls,bconst,lpstandout) 30B1O	01844
(lptptclose) (nls,bconst,lptptclose) 30B1T	01632
(lptptopen) (nls,bconst,lptptopen) 30B1S	01737
(lptptsr) (nls,bconst,lptptsr) 30B1U	01925
(lptrack) (nls,bconst,lptrack) 30B1F	02007
(lptty) (nls,bconst,lptty) 30B1E	01685
(lpttype) (nls,bconst,lpttype) 23A5	01809
(ls) (nls,bconst,ls) 16A2	01675
(lwtyp) (nls,bconst,lwtyp) 10B3	02002
(macfil) (nls,bconst,macfil) 10F4	01812
(maxbsize) (nls,bconst,maxbsize) 19B1	01735
(minfpages) (nls,bconst,minfpages) 19B2	01858
(msbk11) (nls,bconst,msbk11) 37B1	01691
(msproc) (nls,bconst,msproc) 37C1	01744
(msrecl) (nls,bconst,msrecl) 37B2	01754
(mssmpsz) (nls,bconst,mssmpsz) 37C2	01889
(mstxte) (nls,bconst,mstxte) 37A1	01968
(n40fil) (nls,bconst,n40fil) 10F2	01979
(nametyp) (nls,bconst,nametyp) 24A1	01766
(netype) (nls,bconst,netype) 23A6	02000
(niltyp) (nls,bconst,niltyp) 10A13	01951
(nmalt) (nls,bconst,nmalt) 43A	01991
(npcodes) (nls,bconst,npcodes) 30E2A	01800
(npsize) (nls,bconst,npsize) 30E7A	01794
(npstrings) (nls,bconst,npstrings) 30E5A	01987
(nsahost) (nls,bconst,nsahost) 34A11	01873
(nsastr) (nls,bconst,nsastr) 34B3	01847
(numbase) (nls,bconst,numbase) 40H	01717
(numbrv) (nls,bconst,numbrv) 22A10	01942
(numoff) (nls,bconst,numoff) 40I	01790
(nvtk) (nls,bconst,nvtk) 10F1	01817
(nwada) (nls,bconst,nwada) 29A7	01704
(nwstrda) (nls,bconst,nwstrda) 29A9	01799
(nxtname) (nls,bconst,nxtname) 24A6	01638
(odtyp) (nls,bconst,odtyp) 11B3	01798
(ofilerr) (nls,bconst,ofilerr) 8B8	01975
(oldpgsz) (nls,bconst,oldpgsz) 32A1	01891
(oldvrsn) (nls,bconst,oldvrsn) 10F14	01961
(opcdrv) (nls,bconst,opcdrv) 28B3	01938
(opdrdv) (nls,bconst,opdrdv) 28B5	01905
(opmcf1) (nls,bconst,opmcf1) 10F10	01997
(opmlf1) (nls,bconst,opmlf1) 10F11	01686

(opname)	(nls,bconst,opname)	26A1	01784
(opprdv)	(nls,bconst,opprdv)	28B4	01677
(opqpf1)	(nls,bconst,opqpf1)	10F9	01678
(oprmdv)	(nls,bconst,oprmdv)	28B8	01850
(opsqdv)	(nls,bconst,opsqdv)	28B1	01695
(opsqf1)	(nls,bconst,opsqf1)	10F8	01730
(optydv)	(nls,bconst,optydv)	28B2	01824
(opxpdv)	(nls,bconst,opxpdv)	28B6	01848
(opxtdv)	(nls,bconst,opxtdv)	28B7	01943
(orgtyp)	(nls,bconst,orgtyp)	27A3	01972
(origft)	(nls,bconst,origff)	10F13	01827
(outremsiteflag)	(nls,bconst,outremsiteflag)	34C2	02001
(pconlen)	(nls,bconst,pconlen)	14A3	01926
(pkeylen)	(nls,bconst,pkeylen)	14A4	01795
(plexv)	(nls,bconst,plexv)	22A4	01877
(plvlen)	(nls,bconst,plvlen)	14A2	02008
(prcadr)	(nls,bconst,prcadr)	11C	01658
(prcerr)	(nls,bconst,prcerr)	8B9	01676
(pred)	(nls,bconst,pred)	40A	01999
(proptab)	(nls,bconst,proptab)	10C1	01660
(prtdir)	(nls,bconst,prtdir)	10E1	01747
(psprivate)	(nls,bconst,psprivate)	10G2	01854
(pspublic)	(nls,bconst,pspublic)	10G1	01785
(ptradr)	(nls,bconst,ptradr)	38B1C	01783
(pvslen)	(nls,bconst,pvslen)	14A1	01659
(random)	(nls,bconst,random)	23A3	01788
(relext)	(nls,bconst,relext)	10E3	01768
(remada)	(nls,bconst,remada)	29A1	01949
(remdda)	(nls,bconst,remdda)	29A2	01753
(reminit)	(nls,bconst,reminit)	29A12	01828
(remscm)	(nls,bconst,remscm)	29A6	01918
(remscsr)	(nls,bconst,remscsr)	29A4	01897
(remssda)	(nls,bconst,remssda)	29A5	01804
(remstr)	(nls,bconst,remstr)	29A3	01743
(rfchost)	(nls,bconst,rfchost)	34A12	01946
(rfpmax)	(nls,bconst,rfpmax)	10A1	01739
(ring1)	(nls,bconst,ring1)	10A6	01868
(rngbas)	(nls,bconst,rngbas)	10A2	01882
(rngm)	(nls,bconst,rngm)	10A3	01674
(rngtyp)	(nls,bconst,rngtyp)	10A11	01973
(rnmddl)	(nls,bconst,rnmddl)	42A	01989
(sdbtyp)	(nls,bconst,sdbtyp)	10A10	01664
(seqname)	(nls,bconst,seqname)	24A7	01740
(sid)	(nls,bconst,sid)	24A4	01835
(sorterr)	(nls,bconst,sorterr)	8B5	01728
(spacerr)	(nls,bconst,spacerr)	8B6	01994
(spec)	(nls,bconst,spec)	9A1	01948
(sqcls)	(nls,bconst,sqcls)	12A3	01971
(sqerr)	(nls,bconst,sqerr)	8B3	01964
(sqgnxt)	(nls,bconst,sqgnxt)	12A4	01765
(sqopn)	(nls,bconst,sqopn)	12A2	01672
(sqstks)	(nls,bconst,sqstks)	5D	01738
(sriai20host)	(nls,bconst,sriai20host)	34A4	01756
(sriaikahost)	(nls,bconst,sriaikahost)	34A5	01834
(stmtv)	(nls,bconst,stmtv)	22A1	01700
(sub)	(nls,bconst,sub)	40B	01826

(subdir) (nls,bconst,subdir) 26A8	01947
(subdr2) (nls,bconst,subdr2) 26A9	01679
(suc) (nls,bconst,suc) 40C	01821
(sucdir) (nls,bconst,sucdir) 21B1	01832
(sucstr) (nls,bconst,sucstr) 21A2	01650
(svmxlev) (nls,bconst,svmxlev) 40E	01837
(swork1) (nls,bconst,swork1) 10A8	01777
(t2fsbegin) (nls,bconst,t2fsbegin) 35C1	01713
(t2fsend) (nls,bconst,t2fsend) 35E1	01741
(tdfltv) (nls,bconst,tdfltv) 22A15	01748
(tenfil) (nls,bconst,tenfil) 10F3	01836
(textv) (nls,bconst,textv) 22A14	01920
(ttycoc) (nls,bconst,ttycoc) 25A1	01861
(txttext) (nls,bconst,txttext) 10E4	01819
(txttyp) (nls,bconst,txttyp) 10B1	01767
(ue) (nls,bconst,ue) 16A9	01763
(up) (nls,bconst,up) 40D	01954
(upcase) (nls,bconst,upcase) 41A	01911
(upgbdf) (nls,bconst,upgbdf) 19A1	01647
(upgerr) (nls,bconst,upgerr) 8B4	01833
(upgtyp) (nls,bconst,upgtyp) 11B1	01875
(us) (nls,bconst,us) 16A8	01751
(usrexp) (nls,bconst,usrexp) 38A1C	01840
(usrina) (nls,bconst,usrina) 38A1E	01810
(usrigd) (nls,bconst,usrigd) 38A1D	01681
(usrnot) (nls,bconst,usrnot) 38A1B	01667
(usroff) (nls,bconst,usroff) 38A1A	01797
(utilhost) (nls,bconst,utilhost) 34A2	01888
(utilistr) (nls,bconst,utilistr) 34B5	01986
(utilstr) (nls,bconst,utilstr) 34B6	01727
(v) (nls,bconst,v) 20A1	01937
(vb) (nls,bconst,vb) 16A14	01853
(vcv) (nls,bconst,vcv) 20A4	01867
(vcvend) (nls,bconst,vcvend) 20B2	01933
(vcvu) (nls,bconst,vcvu) 20A2	02003
(vcvul) (nls,bconst,vcvul) 20A3	01960
(ve) (nls,bconst,ve) 16A15	01890
(vect) (nls,bconst,vect) 9A4	01965
(vexpert) (nls,bconst,vexpert) 13B1	01689
(vfullprompts) (nls,bconst,vfullprompts) 13B2	01881
(visv) (nls,bconst,visv) 22A11	01917
(vmax) (nls,bconst,vmax) 20B1	01913
(vmultchar) (nls,bconst,vmultchar) 13B4	01662
(vtersemode) (nls,bconst,vtersemode) 13B3	01703
(vverbsmode) (nls,bconst,vverbsmode) 13B5	01803
(wordls) (nls,bconst,wordls) 24A10	01731
(wordtyp) (nls,bconst,wordtyp) 24A2	01814
(wordv) (nls,bconst,wordv) 22A9	01746
(xl10nam) (nls,bconst,xl10nam) 26A6	02009
(xopname) (nls,bconst,xopname) 26A2	01896

(a2) (nls,bdata,)	0716
(a3) (nls,bdata,)	0856
(cm2mp) (nls,bdata,)	0802
(cm3tmp) (nls,bdata,)	0799
(cm4tmp) (nls,bdata,)	0778
(cm5tmp) (nls,bdata,)	0929
(cm6tmp) (nls,bdata,)	0909
(jt1) (nls,bdata,)	0673
(jt10) (nls,bdata,)	0672
(jt11) (nls,bdata,)	0824
(jt12) (nls,bdata,)	0637
(jt13) (nls,bdata,)	0836
(jt14) (nls,bdata,)	0876
(jt15) (nls,bdata,)	0795
(jt2) (nls,bdata,)	0670
(jt3) (nls,bdata,)	0668
(jt4) (nls,bdata,)	0788
(jt5) (nls,bdata,)	0664
(jt6) (nls,bdata,)	0784
(jt7) (nls,bdata,)	0662
(jt8) (nls,bdata,)	0661
(jt9) (nls,bdata,)	0660
(p10) (nls,bdata,)	0912
(p4) (nls,bdata,)	0801
(p5) (nls,bdata,)	0967
(p6) (nls,bdata,)	0798
(p7) (nls,bdata,)	0949
(p8) (nls,bdata,)	0797
(p9) (nls,bdata,)	0946
(wa) (nls,bdata,)	0865
(access) (nls,bdata,access) 12H1	0862
(acchecked) (nls,bdata,acchecked) 12A8	0659
(accmask) (nls,bdata,accmask) 12H2	0868
(accum) (nls,bdata,accum) 12B1A	0698
(acstring) (nls,bdata,acstring) 11D1B	0699
(art) (nls,bdata,art) 12N3J	0819
(arte) (nls,bdata,arte) 12N3K	0676
(astrng) (nls,bdata,astrng) 11D1E	0931
(asub) (nls,bdata,asub) 12B1B	0846
(aulsize) (nls,bdata,aulsize) 12N2B	0957
(aulsrt) (nls,bdata,aulsrt) 12N2A	0812
(autostrt) (nls,bdata,autostrt) 12Q5	0840
(bfilno) (nls,bdata,bfilno) 12P1	0958
(bndchk) (nls,bdata,bndchk) 12A6	0701
(cacflg) (nls,bdata,cacflg) 12B3C	0690
(cadflg) (nls,bdata,cadflg) 12B3A	0882
(calflg) (nls,bdata,calflg) 12B3B	0726
(carpos) (nls,bdata,carpos) 12P5	0910
(castid) (nls,bdata,castid) 12B2B	0694
(cbadent) (nls,bdata,cbadent) 12B1E	0731
(cc) (nls,bdata,cc) 12N3G	0952
(cda) (nls,bdata,cda) 12B2A	0654
(cdchct) (nls,bdata,cdchct) 12N1I	0773
(cdchr1) (nls,bdata,cdchr1) 12N1H	0898
(cdpagt) (nls,bdata,cdpagf) 12Q3	0918
(ckiwheel) (nls,bdata,ckiwheel) 11E1V	0809

(cm1tmp)	(nls,bdata,cm1tmp)	7A1	0828
(cmntdl)	(nls,bdata,cmntdl)	11F1E	0732
(colda)	(nls,bdata,colda)	11J1K	0936
(colsw)	(nls,bdata,colsw)	11J1L	0759
(comexflag)	(nls,bdata,comexflag)	12M13	0811
(commands)	(nls,bdata,commands)	12N3M	0969
(corpst)	(nls,bdata,corpst)	5C3	0666
(cppflag)	(nls,bdata,cppflag)	11K1A	0875
(cstid)	(nls,bdata,cstid)	11J1H	0886
(ctlstid)	(nls,bdata,ctlstid)	11E1H	0821
(cversion)	(nls,bdata,cversion)	11J1S	0861
(datesr)	(nls,bdata,datesr)	11E3D	0904
(dblst)	(nls,bdata,dblst)	5F1A	0675
(debugaccess)	(nls,bdata,debugaccess)	12H5	0955
(dfoutm)	(nls,bdata,dfoutm)	12B3D	0738
(dgbufe)	(nls,bdata,dgbufe)	12N3R	0762
(dgstid)	(nls,bdata,dgstid)	12N3Q	0962
(dgstl)	(nls,bdata,dgstl)	12N3Q	0853
(dgstm1)	(nls,bdata,dgstm1)	12N3P	0730
(diststid)	(nls,bdata,diststid)	11E1G	0763
(docjfn)	(nls,bdata,docjfn)	11H1D	0894
(docstrt)	(nls,bdata,docstrt)	11H1F	0781
(docwfn)	(nls,bdata,docwfn)	11H2D	0845
(dpcmb1k)	(nls,bdata,dpcmb1k)	12N3N	0935
(dtbl)	(nls,bdata,dtbl)	12I12	0733
(dtbst)	(nls,bdata,dtbst)	12I15	0640
(ercall)	(nls,bdata,ercall)	12P2	0964
(ermark)	(nls,bdata,ermark)	12P3	0970
(exp)	(nls,bdata,exp)	12A1	0768
(fastname)	(nls,bdata,fastname)	12P4	0848
(fchsct)	(nls,bdata,fchsct)	12M3D	0783
(fcredt)	(nls,bdata,fcredt)	12I3	0847
(filcnt)	(nls,bdata,filcnt)	12F2	0844
(filehead)	(nls,bdata,filehead)	5C4	0863
(filepart)	(nls,bdata,filepart)	5C5	0689
(filnde)	(nls,bdata,filnde)	12I19	0728
(filned)	(nls,bdata,filned)	12I2	0760
(filst)	(nls,bdata,filst)	12F1	0651
(finit)	(nls,bdata,finit)	12I6	0646
(flagjfn)	(nls,bdata,flagjfn)	11E1I	0864
(freemap)	(nls,bdata,freemap)	12L1	0888
(freesz)	(nls,bdata,freesz)	3A2A	0965
(ftpem)	(nls,bdata,ftpem)	11I1B	0741
(ftperm)	(nls,bdata,ftperm)	11I1C	0850
(ftphnd)	(nls,bdata,ftphnd)	11I1A	0734
(ftpoh)	(nls,bdata,ftpoh)	11I1D	0870
(ftpoha)	(nls,bdata,ftpoha)	11I1E	0813
(funo)	(nls,bdata,funo)	12I7	0721
(gapbp)	(nls,bdata,gapbp)	12N3E	0786
(gapcc)	(nls,bdata,gapcc)	12N3F	0706
(gapcnt)	(nls,bdata,gapcnt)	12N3D	0779
(gapcol)	(nls,bdata,gapcol)	12N3C	0914
(groupx)	(nls,bdata,groupx)	3A1A	0769
(gtjint)	(nls,bdata,gtjint)	12M14	0739
(hdrjfn)	(nls,bdata,hdrjfn)	11H1E	0737
(hdrwfn)	(nls,bdata,hdrwfn)	11H2E	0835

(hostni) (nls,bdata,hostni) 12A2	0683
(hostsi) (nls,bdata,hostsi) 12A3	0934
(idcident) (nls,bdata,idcident) 11E1Y	0747
(idcrec) (nls,bdata,idcrec) 11E1W	0890
(idcrtype) (nls,bdata,idcrtype) 11E1X	0717
(identstr) (nls,bdata,identstr) 11F1A	0341
(identwheel) (nls,bdata,identwheel) 11F3A	0903
(idicnt) (nls,bdata,idfcnt) 11E1S	0649
(idfile) (nls,bdata,idfile) 11H1K	0857
(idfno) (nls,bdata,idfno) 11E1T	0933
(idirn) (nls,bdata,idirn) 11E1M	0775
(idmodflag) (nls,bdata,idmodflag) 11E1A@	0643
(idmodified) (nls,bdata,idmodified) 11E1AA	0678
(idnamdel) (nls,bdata,idnamdel) 11F1F	0745
(idntdel) (nls,bdata,idntdel) 11F1D	0650
(idwork) (nls,bdata,idwork) 11F1B	0774
(ijfn) (nls,bdata,ijfn) 11C1E	0679
(infopn) (nls,bdata,infopn) 11J1G	0928
(infork) (nls,bdata,infork) 11C1G	0807
(inpstp) (nls,bdata,inpstp) 5E3	0913
(inptrf) (nls,bdata,inptrf) 5E1	0729
(interflag) (nls,bdata,interflag) 11E1A	0767
(ipassw) (nls,bdata,ipassw) 11E3A	0794
(iswork) (nls,bdata,iswork) 12M11	0961
(jb1) (nls,bdata,jb1) 11E2C	0686
(jb2) (nls,bdata,jb2) 11E2D	0685
(jcatstid) (nls,bdata,jcatstid) 11E1E	0867
(jdebug) (nls,bdata,jdebug) 11E1J	0820
(jdfi) (nls,bdata,jdfi) 11E1R	0893
(jdid) (nls,bdata,jdid) 11E1Q	0776
(jdirn) (nls,bdata,jdirn) 11E1L	0939
(jdno) (nls,bdata,jdno) 11E1P	0814
(jfnfln) (nls,bdata,jfnfln) 8A1	0709
(jidsbot) (nls,bdata,jidsbot) 11F4A	0792
(jidstk) (nls,bdata,jidstk) 11F2A	0854
(jlogjfn) (nls,bdata,jlogjfn) 11E1AF	0757
(jnamstr) (nls,bdata,jnamstr) 11E3E	0727
(jnlprog) (nls,bdata,jnlprog) 12Q4	0687
(jobtbl) (nls,bdata,jobtbl) 12A5	0822
(jokludgeupdtfl) (nls,bdata,jokludgeupdtfl) 11E1AG	0901
(jpasswd) (nls,bdata,jpasswd) 11E3B	0884
(jrnaccess) (nls,bdata,jrnaccess) 12H4	0953
(jrnlstid) (nls,bdata,jrnlstid) 11E1F	0720
(jsavaccess) (nls,bdata,jsavaccess) 11E1AH	0925
(jt0) (nls,bdata,jt0) 11E1AD2	0674
(jtimfg) (nls,bdata,jtimfg) 11E1AD1	0677
(jworkstid) (nls,bdata,jworkstid) 11E1D	0873
(jwp1) (nls,bdata,jwp1) 11E2A	0917
(keypr) (nls,bdata,keypr) 11J1A	0897
(ladj) (nls,bdata,ladj) 12C2	0636
(lastch) (nls,bdata,lastch) 12N1A	0755
(lastfnum) (nls,bdata,lastfnum) 11H1H	0920
(lavtabad) (nls,bdata,lavtabad) 11E1O	0902
(ldrfct) (nls,bdata,ldrfct) 12M3A	0869
(ldrnct) (nls,bdata,ldrnct) 12M3B	0883
(ldsct) (nls,bdata,ldsct) 12M3C	0695

(levind) (nls,bdata,levind)	12K1	0753
(lgngtps) (nls,bdata,lgngtps)	12A10	0941
(libflg) (nls,bdata,libflg)	12Q2	0889
(liblod) (nls,bdata,liblod)	12Q3	0710
(lkupreg) (nls,bdata,lkupreg)	12C5	0796
(lptjfn) (nls,bdata,lptjfn)	11H1B	0724
(lptused) (nls,bdata,lptused)	11H1C	0658
(lstred) (nls,bdata,lstred)	12A7	0766
(lwtim) (nls,bdata,lwtim)	12I8	0892
(mastacprintflag) (nls,bdata,mastacprintflag)	11H1J	0806
(measnt) (nls,bdata,measnt)	12M10	0644
(merrff) (nls,bdata,merrff)	11J1E	0815
(mersiz) (nls,bdata,mersiz)	11J1D	0817
(mkrcnt) (nls,bdata,mkrcnt)	12N1E	0943
(mkrend) (nls,bdata,mkrend)	12N1C	0944
(mkrflg) (nls,bdata,mkrflg)	12N1F	0712
(mkrptr) (nls,bdata,mkrptr)	12N1D	0959
(mkrtb) (nls,bdata,mkrtb)	12I18	0785
(mkrtbl) (nls,bdata,mkrtbl)	12I17	0749
(mkrtxn) (nls,bdata,mkrtxn)	12I16	0911
(modeoff) (nls,bdata,modeoff)	5G1	0764
(msaddr) (nls,bdata,msaddr)	12M4	0657
(msbufe) (nls,bdata,msbufe)	13B2	0831
(msbuff) (nls,bdata,msbuff)	13B1	0711
(msflag) (nls,bdata,msflag)	9C	0895
(mslst) (nls,bdata,mslst)	12M5	0832
(mslste) (nls,bdata,mslste)	12M6	0782
(msrlsv) (nls,bdata,msrlsv)	12M7	0887
(msr2sv) (nls,bdata,msr2sv)	12M8	0851
(msr3sv) (nls,bdata,msr3sv)	12M9	0855
(msstcount) (nls,bdata,msstcount)	9A	0823
(mstxcount) (nls,bdata,mstxcount)	9B	0744
(mswsic) (nls,bdata,mswsic)	12M2	0704
(mswsit) (nls,bdata,mswsit)	12M1	0899
(nambuf) (nls,bdata,nambuf)	11J1P	0842
(namdl1) (nls,bdata,namdl1)	12I9	0896
(namdl2) (nls,bdata,namdl2)	12I10	0932
(namlst) (nls,bdata,namlst)	11J1O	0843
(namndx) (nls,bdata,namndx)	11J1N	0858
(newid) (nls,bdata,newid)	11F1C	0758
(ninfil) (nls,bdata,ninfil)	11J1Q	0756
(nlsvwd) (nls,bdata,nlsvwd)	12I4	0765
(nofeedbk) (nls,bdata,nofeedbk)	12B1F	0635
(normlaccess) (nls,bdata,normlaccess)	12H3	0954
(nsdcktime) (nls,bdata,nsdcktime)	11E1N	0693
(numstid) (nls,bdata,numstid)	11G1A	0924
(nwrecflag) (nls,bdata,nwrecflag)	11E1Z	0692
(nxpg) (nls,bdata,nxpg)	5C2	0645
(ojfn) (nls,bdata,ojfn)	11C1D	0810
(ojnestr) (nls,bdata,ojnestr)	8B1	0930
(ojsqsw) (nls,bdata,ojsqsw)	11E1AE	0880
(oldflg) (nls,bdata,oldflg)	12Q1	0772
(oldlsrt) (nls,bdata,oldlsrt)	12N3L	0834
(oljmlav) (nls,bdata,oljmlav)	11E1AB	0715
(oljsbn) (nls,bdata,oljsbn)	11E1AC	0963
(opbp) (nls,bdata,opbp)	12R2	0725

(opcbp) (nls,bdata,opcbp) 12R3	0808
(opccpos) (nls,bdata,opccpos) 12R4	0638
(opcmax) (nls,bdata,opcmax) 12R5	0684
(opfloat) (nls,bdata,opfloat) 12B1D	0789
(oplev) (nls,bdata,oplev) 12R6	0956
(opnewst) (nls,bdata,opnewst) 12R7	0761
(oprocn) (nls,bdata,oprocn) 11H2A	0787
(oprwrk) (nls,bdata,oprwrk) 12M12	0793
(opstr) (nls,bdata,opstr) 11H2C	0735
(opstring) (nls,bdata,opstring) 11D1A	0665
(oqapfg) (nls,bdata,oqapfg) 12J15	0874
(oqnhfg) (nls,bdata,oqnhfg) 12J14	0736
(outcnt) (nls,bdata,outcnt) 11J1I	0696
(outnam) (nls,bdata,outnam) 11J2A	0682
(p3) (nls,bdata,p3) 12D1	0971
(pageno) (nls,bdata,pageno) 12O4	0878
(patch) (nls,bdata,patch) 12U1	0866
(patche) (nls,bdata,patche) 12U2	0647
(pfilnum) (nls,bdata,pfilnum) 11H1G	0879
(pjfn) (nls,bdata,pjfn) 11C1F	0714
(pjlsw) (nls,bdata,pjlsw) 12T1E	0838
(pjrbbab) (nls,bdata,pjrbbab) 12T1A	0968
(pjrubout) (nls,bdata,pjrubout) 12T1F	0655
(pjsavf) (nls,bdata,pjsavf) 12T1C	0703
(pjstid) (nls,bdata,pjstid) 12T2A	0827
(pjsw) (nls,bdata,pjsw) 12T1D	0669
(pjusqc) (nls,bdata,pjusqc) 12T1B	0872
(pmjfn) (nls,bdata,pmjfn) 12L2	0816
(prbuf) (nls,bdata,prbuf) 12C6	0926
(prmkrf) (nls,bdata,prmkrf) 12O2	0639
(prntfg) (nls,bdata,prntfg) 11H1I	0671
(proc) (nls,bdata,proc) 12B2C	0881
(prseqwk) (nls,bdata,prseqwk) 12R1	0680
(ptstr) (nls,bdata,ptstr) 11H2B	0921
(rfbs) (nls,bdata,rfbs) 12I13	0940
(rfcflg) (nls,bdata,rfcflg) 11E1K	0681
(rfcnum) (nls,bdata,rfcnum) 11E3C	0743
(rfpmin) (nls,bdata,rfpmin) 5C1	0829
(rngl) (nls,bdata,rngl) 12I11	0648
(rnglst) (nls,bdata,rnglst) 5F1B	0906
(rngst) (nls,bdata,rngst) 12I14	0742
(rt) (nls,bdata,rt) 12N3H	0907
(rte) (nls,bdata,rte) 12N3I	0770
(rtsize) (nls,bdata,rtsize) 12N2D	0804
(rttop) (nls,bdata,rttop) 12N2C	0849
(rubabt) (nls,bdata,rubabt) 5E4	0791
(rubmrk) (nls,bdata,rubmrk) 5E2	0947
(rubnocob) (nls,bdata,rubnocob) 5E5	0653
(s2work) (nls,bdata,s2work) 5D1	0923
(sabtfg) (nls,bdata,sabtfg) 11E1C	0826
(setidfno) (nls,bdata,setidfno) 11E1U	0746
(sfbptr) (nls,bdata,sfbptr) 12J12	0718
(sfbufe) (nls,bdata,sfbufe) 13C2	0656
(stbuff) (nls,bdata,sfbuff) 13C1	0771
(sfbufl) (nls,bdata,sfbufl) 12J5	0825
(sfbyte) (nls,bdata,sfbyte) 12J4	0966

(sfhead)	(nls,bdata,sfhead)	12C3	0950
(sflnel)	(nls,bdata,sflnel)	12J8	0652
(sflnpg)	(nls,bdata,sflnpg)	12J13	0790
(sfmxnd)	(nls,bdata,sfmxd)	12J10	0951
(sfndbf)	(nls,bdata,sfndbf)	12J11	0885
(sfndlv)	(nls,bdata,sfndlv)	12J9	0852
(sfpjno)	(nls,bdata,sfpjno)	12J7	0777
(sfpmr1)	(nls,bdata,sfpmr1)	12J1	0752
(sfpmr2)	(nls,bdata,sfpmr2)	12J2	0750
(sfpmr3)	(nls,bdata,sfpmr3)	12J3	0719
(sfstng)	(nls,bdata,sfstng)	12C4	0688
(sfucf)	(nls,bdata,sfucf)	12J6	0891
(sidcnt)	(nls,bdata,sidcnt)	12I5	0722
(signstr)	(nls,bdata,signstr)	11D1C	0740
(skpachk)	(nls,bdata,skpachk)	12A9	0927
(slashflg)	(nls,bdata,slashflg)	12P6	0751
(slnkrv)	(nls,bdata,slnkrv)	12Q6	0705
(slshfg)	(nls,bdata,slshfg)	12O5	0948
(smtmax)	(nls,bdata,smtmax)	11J1J	0960
(sortfg)	(nls,bdata,sortfg)	11J1M	0922
(sqgaend)	(nls,bdata,sqgaend)	6D	0859
(sqqwas)	(nls,bdata,sqqwas)	6C	0837
(sqsavsw)	(nls,bdata,sqsavsw)	6B	0871
(sqsvws)	(nls,bdata,sqsvws)	6E	0830
(srctype)	(nls,bdata,srctype)	12P7	0707
(strpkey)	(nls,bdata,strpkey)	11J1R	0945
(strttm)	(nls,bdata,strttm)	12M3E	0900
(subcnt)	(nls,bdata,subcnt)	12S1	0919
(subhed)	(nls,bdata,subhed)	12G2	0805
(subnm1)	(nls,bdata,subnm1)	3A3B	0713
(subnum)	(nls,bdata,subnum)	3A3A	0915
(subrec)	(nls,bdata,subrec)	12G1	0748
(tiw)	(nls,bdata,tiw)	11C1A	0691
(tmode)	(nls,bdata,tmode)	11C1B	0700
(tmpbug)	(nls,bdata,tmpbug)	11E2B	0702
(tmpit)	(nls,bdata,tmpit)	11E1B	0839
(trmcod)	(nls,bdata,trmcod)	11C1C	0641
(tskerrcnt)	(nls,bdata,tskerrcnt)	11A1A	0833
(tstrng)	(nls,bdata,tstrng)	11D1D	0938
(ttytbl)	(nls,bdata,ttytbl)	12A4	0803
(typend)	(nls,bdata,typend)	12M15	0916
(udpnwrap)	(nls,bdata,udpnwrap)	12N3B	0908
(upgbend)	(nls,bdata,upgbend)	11B1C	0780
(upgbsz)	(nls,bdata,upgbsz)	11B1D	0697
(upgbuf)	(nls,bdata,upgbuf)	11B1A	0708
(upgcacnt)	(nls,bdata,upgcacnt)	11B1E	0860
(upgffbuf)	(nls,bdata,upgffbuf)	11B1B	0663
(upgnms)	(nls,bdata,upgnms)	11B1H1	0905
(upgskix)	(nls,bdata,upgskix)	11B1F	0942
(upgsksz)	(nls,bdata,upgsksz)	3A4A	0818
(upgstk)	(nls,bdata,upgstk)	11B1G	0642
(utiltv)	(nls,bdata,utiltv)	12Q7	0937
(vaccum)	(nls,bdata,vaccum)	12B1C	0723
(vcvtp)	(nls,bdata,vcvtp)	11J1C	0667
(vcvulp)	(nls,bdata,vcvulp)	11J1B	0800
(vrsnno)	(nls,bdata,vrsnno)	12C1	0754

GAS2, 14-Feb-79 22:37

< NLS, MINISYSGD.NLS.4, > 17

(vtop) (nls,bdata,vtop) 11J1F

0877
03872

(backsave) (nls,bintnls,backsave) 6A	02822
(binit) (nls,bintnls,binit) 5A4B	02838
(bpresaveinit) (nls,bintnls,bpresaveinit) 4A	02823
(bqt) (nls,bintnls,bqt) 4A2B	02836
(btypclstr) (nls,bintnls,btypclstr) 3H	02834
(btypestr) (nls,bintnls,btypestr) 3G	02825
(checklib) (nls,bintnls,checklib) 6D	02831
(dex) (nls,bintnls,dex) 3A	02826
(dxin) (nls,bintnls,dxin) 3B	02840
(enabllogin) (nls,bintnls,enabllogin) 6H	02835
(filinit) (nls,bintnls,filinit) 6C	02844
(initback) (nls,bintnls,initback) 5A	02833
(intmeas) (nls,bintnls,intmeas) 6I3	02843
(jbackground) (nls,bintnls,jbackground) 3D	02839
(jnl in) (nls,bintnls,jnl in) 3F	02842
(jnirun) (nls,bintnls,jnlrun) 3E	02841
(killib) (nls,bintnls,killib) 6G	02829
(libstrt) (nls,bintnls,libstrt) 6F	02828
(net) (nls,bintnls,net) 3I	02824
(netdsp) (nls,bintnls,netdsp) 3K	02837
(quin) (nls,bintnls,quin) 3C	02830
(setbvec) (nls,bintnls,setbvec) 6B	02827
(startback) (nls,bintnls,startback) 5B	02832
	03873

(swfill)	(nls,brecords,)		02197
(swstka)	(nls,brecords,)		02318
(swstkd)	(nls,brecords,)		02234
(swstk1)	(nls,brecords,)		02278
(swstks)	(nls,brecords,)		02354
(acctyp)	(nls,brecords,acctyp)	3L3	02361
(b0)	(nls,brecords,b0)	3U5	02388
(bit1)	(nls,brecords,bit1)	3U4	02248
(bit2)	(nls,brecords,bit2)	3U3	02367
(bit3)	(nls,brecords,bit3)	3U2	02206
(bitfiller)	(nls,brecords,bitfiller)	3U1	02199
(canxt)	(nls,brecords,canxt)	3A5	02282
(carbp)	(nls,brecords,carbp)	3A4	02266
(carnc)	(nls,brecords,carnc)	3A2	02256
(catbp)	(nls,brecords,catbp)	3A3	02306
(catnc)	(nls,brecords,catnc)	3A1	02284
(cbhl)	(nls,brecords,cbhl)	3B7A	02228
(cbnxt)	(nls,brecords,cbnxt)	3B2	02355
(cbprev)	(nls,brecords,cbprev)	3B1	02279
(cmda)	(nls,brecords,cmda)	3B5	02352
(cmlst)	(nls,brecords,cmlst)	3B4	02313
(cmnxt)	(nls,brecords,cmnxt)	3B3	02320
(copcode)	(nls,brecords,copcode)	3C3	02309
(corcpy)	(nls,brecords,corcpy)	3D3	02269
(corlck)	(nls,brecords,corlck)	3D1	02262
(cortim)	(nls,brecords,cortim)	3D2	02268
(cparm1)	(nls,brecords,cparm1)	3C1	02200
(cparm2)	(nls,brecords,cparm2)	3C2	02386
(ctfile)	(nls,brecords,ctfile)	3E2	02223
(ctfroz)	(nls,brecords,ctfroz)	3E4	02339
(ctfull)	(nls,brecords,ctfull)	3E1	02329
(ctpnum)	(nls,brecords,ctpnum)	3E3	02275
(ctype)	(nls,brecords,ctype)	3C4	02288
(delhc)	(nls,brecords,delhc)	3F3	02397
(delnet)	(nls,brecords,delnet)	3F2	02263
(delol)	(nls,brecords,delol)	3F1	02358
(dlfbal)	(nls,brecords,dlfbal)	2I5	02312
(dlfbap)	(nls,brecords,dlfbap)	2I10	02196
(dlfbcp)	(nls,brecords,dlfbcp)	2I9	02325
(dlfbff)	(nls,brecords,dlfbff)	2I6	02389
(dlfbfl)	(nls,brecords,dlfbfl)	2I8	02240
(dlfbgk)	(nls,brecords,dlfbgk)	2I3	02399
(dlfbl)	(nls,brecords,dlfbl)	2I12A	02422
(dlfbnb)	(nls,brecords,dlfbnb)	2I1	02384
(dlfbpb)	(nls,brecords,dlfbpb)	2I2	02194
(dlfbpf)	(nls,brecords,dlfbpf)	2I7	02221
(dlfbsk)	(nls,brecords,dlfbsk)	2I4	02298
(dlgbgk)	(nls,brecords,dlgbgk)	2K5	02204
(dlgbl)	(nls,brecords,dlgbl)	2K7A	02305
(dlgbnb)	(nls,brecords,dlgbnb)	2K1	02335
(dlgbnu)	(nls,brecords,dlgbnu)	2K4	02387
(dlgbpb)	(nls,brecords,dlgbpb)	2K2	02381
(dlgbsc)	(nls,brecords,dlgbsc)	2K3	02232
(dlmbal)	(nls,brecords,dlmbal)	2L1B	02257
(dlmbfl)	(nls,brecords,dlmbfl)	2L1E	02406
(dlmbfp)	(nls,brecords,dlmbfp)	2L4	02273

(dlmbfr)	(nls,brecords,dlmbfr)	2L1C	02409
(dlmb1)	(nls,brecords,dlmb1)	2L6A	02283
(dlmb1n)	(nls,brecords,dlmb1n)	2L1A	02218
(dlmbnm)	(nls,brecords,dlmbnm)	2L3	02326
(dlmbpr)	(nls,brecords,dlmbpr)	2L1D	02231
(dlmbzn)	(nls,brecords,dlmbzn)	2L2	02219
(dlstig)	(nls,brecords,dlstig)	2J1	02346
(dlstnl)	(nls,brecords,dlstnl)	2J3	02216
(dlstpc)	(nls,brecords,dlstpc)	2J2	02333
(dstar)	(nls,brecords,dstar)	3K13	02307
(dvstar)	(nls,brecords,dvstar)	3K15	02345
(estar)	(nls,brecords,estar)	3K11	02289
(fbind)	(nls,brecords,fbind)	3H2	02344
(fbnul1)	(nls,brecords,fbnul1)	3H1	02338
(fbpnum)	(nls,brecords,fbpnum)	3H3	02245
(fbtype)	(nls,brecords,fbtype)	3H4	02311
(fdbaar)	(nls,brecords,fdbaar)	2F2	02401
(fdbact)	(nls,brecords,fdbact)	2M6	02292
(fdbad1)	(nls,brecords,fdbad1)	2F3	02207
(fdbarc)	(nls,brecords,fdbarc)	2F7	02276
(fdbart)	(nls,brecords,fdbart)	2M14	02323
(fdbbkt)	(nls,brecords,fdbbkt)	2M13	02341
(fdbbyv)	(nls,brecords,fdbbyv)	2M7	02343
(fdbcnt)	(nls,brecords,fdbcnt)	2M12	02317
(fdbcre)	(nls,brecords,fdbcre)	2M3	02246
(fdbcrv)	(nls,brecords,fdbcrv)	2M9	02254
(fdbctl)	(nls,brecords,fdbctl)	2M1	02243
(fdbctu)	(nls,brecords,fdbctu)	2C1	02398
(fdbdel)	(nls,brecords,fdbdel)	2C6	02321
(fdbdmp)	(nls,brecords,fdbdmp)	2F5	02404
(fdbeph)	(nls,brecords,fdbeph)	2C2	02413
(fdblng)	(nls,brecords,fdblng)	2C4	02274
(fdbmrk)	(nls,brecords,fdbmrk)	2F4	02224
(fdbnar)	(nls,brecords,fdbnar)	2F6	02378
(fdbnex)	(nls,brecords,fdbnex)	2C7	02316
(fdbnun)	(nls,brecords,fdbnun)	2C3	02259
(fdbnxt)	(nls,brecords,fdbnxt)	2C5	02233
(fdbprm)	(nls,brecords,fdbprm)	2C8	02385
(fdbprt)	(nls,brecords,fdbprt)	2M2	02198
(fdbref)	(nls,brecords,fdbref)	2M11	02356
(fdbsiz)	(nls,brecords,fdbsiz)	2M8	02280
(fdbtdm)	(nls,brecords,fdbtdm)	2M15	02393
(fdbtfa)	(nls,brecords,fdbtfa)	2M16	02261
(fdbtmp)	(nls,brecords,fdbtmp)	2C9	02394
(fdbtsa)	(nls,brecords,fdbtsa)	2M17	02265
(fdbunu)	(nls,brecords,fdbunu)	2F1	02377
(fdbuse)	(nls,brecords,fdbuse)	2M4	02347
(fdbusw)	(nls,brecords,fdbusw)	2M18	02308
(fdbver)	(nls,brecords,fdbver)	2M5	02304
(fdbwrt)	(nls,brecords,fdbwrt)	2M10	02349
(fhunused)	(nls,brecords,fhunused)	3G2	02418
(filmax)	(nls,brecords,filmax)	3I15B	02411
(filst1)	(nls,brecords,filst1)	3I15A	02380
(flaccm)	(nls,brecords,flaccm)	3I6	02408
(flarfl)	(nls,brecords,flarfl)	2E2	02211
(flastr)	(nls,brecords,flastr)	3I12	02285

(flbpart) (nls,brecords,flbpart) 3I10	02417
(flbpcst) (nls,brecords,flbpcst) 3I14	02212
(flbrws) (nls,brecords,flbrws) 3I3	02330
(flbyts) (nls,brecords,flbyts) 2H3	02237
(fldfr) (nls,brecords,fldfr) 2H4	02293
(fldirno) (nls,brecords,fldirno) 3I7	02369
(fldmpt) (nls,brecords,fldmpt) 2E1	02331
(flexis) (nls,brecords,flexis) 3I1	02416
(flfill) (nls,brecords,flfill) 2H2	02368
(flfsat) (nls,brecords,flfsat) 2D2	02336
(flhead) (nls,brecords,flhead) 3I2	02241
(fllock) (nls,brecords,fllock) 3I4	02264
(flnmrd) (nls,brecords,flnmrd) 2G1	02299
(flnmwr) (nls,brecords,flnmwr) 2G2	02258
(flnoclos) (nls,brecords,flnoclos) 3I8	02414
(florig) (nls,brecords,florig) 3I11	02405
(flpart) (nls,brecords,flpart) 3I9	02193
(flpcread) (nls,brecords,flpcread) 3I5	02324
(flpcst) (nls,brecords,flpcst) 3I13	02244
(flpgsz) (nls,brecords,flpgsz) 2H1	02239
(flprap) (nls,brecords,flprap) 2B3	02252
(flprex) (nls,brecords,flprex) 2B4	02400
(flprgr) (nls,brecords,flprgr) 2A2	02362
(flprli) (nls,brecords,flprli) 2B2	02334
(flprnu) (nls,brecords,flprnu) 2B1	02391
(flprpu) (nls,brecords,flprpu) 2A1	02376
(flprre) (nls,brecords,flprre) 2B6	02402
(flprse) (nls,brecords,flprse) 2A3	02314
(flprwr) (nls,brecords,flprwr) 2B5	02202
(flsnat) (nls,brecords,flsnat) 2D1	02410
(fstar) (nls,brecords,fstar) 3K12	02253
(isbuff) (nls,brecords,isbuff) 3J2	02348
(isdpth) (nls,brecords,isdpth) 3J8	02366
(isfiltyp) (nls,brecords,isfiltyp) 3J9	02210
(isinfno) (nls,brecords,isinfno) 3J4	02373
(islevb) (nls,brecords,islevb) 3J6	02360
(islevs) (nls,brecords,islevs) 3J3	02319
(islstb) (nls,brecords,islstb) 3J7	02227
(isstid) (nls,brecords,isstid) 3J1	02229
(istatnd) (nls,brecords,istatnd) 3J5	02412
(lcard) (nls,brecords,lcard) 3W1	02407
(lfj1un) (nls,brecords,lfj1un) 3K16	02300
(lhjb10) (nls,brecords,lhjb10) 3K5	02342
(lhjb11) (nls,brecords,lhjb11) 3K4	02420
(lhjb12) (nls,brecords,lhjb12) 3K3	02267
(lhjb13) (nls,brecords,lhjb13) 3K2	02332
(lhjb6) (nls,brecords,lhjb6) 3K9	02383
(lhjb7) (nls,brecords,lhjb7) 3K8	02255
(lhjb8) (nls,brecords,lhjb8) 3K7	02260
(lhjb9) (nls,brecords,lhjb9) 3K6	02322
(lhjrun) (nls,brecords,lhjrun) 3K1	02281
(lkdirn) (nls,brecords,lkdirn) 3L2	02327
(lkinit) (nls,brecords,lkinit) 3L1	02374
(lshed) (nls,brecords,lshed) 3X1	02302
(mkccnt) (nls,brecords,mkccnt) 3N4	02372
(mkexis) (nls,brecords,mkexis) 3N5	02236

(mkfix) (nls,brecords,mkfix) 3N3	02337
(mkname) (nls,brecords,mkname) 3N1	02392
(mkpsid) (nls,brecords,mkpsid) 3N2	02415
(mkrl) (nls,brecords,mkrl) 3N6A	02208
(ms1clk) (nls,brecords,ms1clk) 3M7	02272
(msbegin) (nls,brecords,msbegin) 3M1	02230
(msbinstn) (nls,brecords,msbinstn) 3M3	02359
(mscurr) (nls,brecords,mscurr) 3M6	02203
(mseinstn) (nls,brecords,mseinstn) 3M4	02286
(msend) (nls,brecords,msend) 3M2	02242
(msentl) (nls,brecords,msentl) 3M9A	02371
(msmax) (nls,brecords,msmax) 3M5	02364
(mstclk) (nls,brecords,mstclk) 3M8	02350
(ojdelf) (nls,brecords,ojdelf) 3L4	02271
(ojtiii) (nls,brecords,ojtiii) 3O3	02222
(ojtijjj) (nls,brecords,ojtijjj) 3O4	02290
(ojtinn) (nls,brecords,ojtinn) 3O2	02213
(ojtitt) (nls,brecords,ojtitt) 3O1	02395
(opform) (nls,brecords,opform) 3P1	02291
(opsimff) (nls,brecords,opsimff) 3P2	02379
(opwtpb) (nls,brecords,opwtpb) 3P3	02370
(prvsts) (nls,brecords,prvsts) 3G1	02215
(rfcore) (nls,brecords,rfcore) 3Q6	02310
(rfexis) (nls,brecords,rfexis) 3Q1	02225
(rffree) (nls,brecords,rffree) 3Q5	02287
(rfnull) (nls,brecords,rfnull) 3Q3	02195
(rfpart) (nls,brecords,rfpart) 3Q2	02214
(rfused) (nls,brecords,rfused) 3Q4	02365
(sbas) (nls,brecords,sbas) 3S3	02396
(sbc1cnt) (nls,brecords,sbc1cnt) 3T1	02235
(sbc2cnt) (nls,brecords,sbc2cnt) 3T2	02277
(sbc3cnt) (nls,brecords,sbc3cnt) 3T3	02249
(sbdp) (nls,brecords,sbdp) 3S2	02201
(sbrp) (nls,brecords,sbrp) 3S1	02419
(sbtt) (nls,brecords,sbtt) 3S4	02340
(sqwrkl) (nls,brecords,sqwrkl) 3R21A	02315
(swalloc) (nls,brecords,swalloc) 3R14	02295
(swcacode) (nls,brecords,swcacode) 3R5	02238
(swcall) (nls,brecords,swcall) 3R15	02270
(swclvl) (nls,brecords,swclvl) 3R12	02301
(swcstid) (nls,brecords,swcstid) 3R1	02382
(swkflg) (nls,brecords,swkflg) 3R13	02296
(swlbstid) (nls,brecords,swlbstid) 3R2	02205
(swmrsav) (nls,brecords,swmrsav) 3R9	02353
(swslvl) (nls,brecords,swslvl) 3R11	02303
(swsrsav) (nls,brecords,swsrsav) 3R8	02328
(swstid) (nls,brecords,swstid) 3R3	02297
(swsvw) (nls,brecords,swsvw) 3R10	02351
(swusqcod) (nls,brecords,swusqcod) 3R4	02357
(swvsp2) (nls,brecords,swvsp2) 3R7	02390
(swvspec) (nls,brecords,swvspec) 3R6	02217
(usrflgs) (nls,brecords,usrflgs) 3V4	02403
(usrgrp) (nls,brecords,usrgrp) 3V5	02375
(usrjob) (nls,brecords,usrjob) 3V6	02247
(usrlnk) (nls,brecords,usrlnk) 3V1	02220
(usrno) (nls,brecords,usrno) 3V3	02294

(usrft)	(nls,brecords,usrft)	3V9	02421
(usrftm)	(nls,brecords,usrftm)	3V10	02363
(usrftod)	(nls,brecords,usrftod)	3V8	02226
(usrftty)	(nls,brecords,usrftty)	3V7	02251
(ustar)	(nls,brecords,ustar)	3K14	02209
(vstar)	(nls,brecords,vstar)	3K10	02250
			03874

(r0) (nls,calcsupport,)		03302
(r6) (nls,calcsupport,)		03304
(fone) (nls,calcsupport,fone) 2C1		03296
(nfloat) (nls,calcsupport,nfloat) 3H		03303
(qcadd) (nls,calcsupport,qcadd) 3B		03300
(qcdiv) (nls,calcsupport,qcdiv) 3C		03299
(qcdivw) (nls,calcsupport,qcdivw) 3D		03305
(qcmult) (nls,calcsupport,qcmult) 3A		03306
(qcneg) (nls,calcsupport,qcneg) 3F		03298
(qcsub) (nls,calcsupport,qcsub) 3E		03301
(qfloat) (nls,calcsupport,qfloat) 3G		03297
		03875

(ckcnum) (nls,catnum,ckcnum) 2B	03586
(ckrfcnum) (nls,catnum,ckrfcnum) 2C	03589
(gcatnums) (nls,catnum,gcatnums) 2A	03591
(getcnum) (nls,catnum,getcnum) 2D	03590
(movecnum) (nls,catnum,movecnum) 2E	03593
(numtype) (nls,catnum,numtype) 2F	03592
(rfcex) (nls,catnum,rfcex) 3A	03594
(rfcsyscheck) (nls,catnum,rfcsyscheck) 3B	03588
(usedcnum) (nls,catnum,usedcnum) 2G	03587
	03876

(newnam) (nls,cedit1,)		02893
(newver) (nls,cedit1,)		02864
(notyet) (nls,cedit1,)		02869
(oldver) (nls,cedit1,)		02891
(upcomp) (nls,cedit1,)		02900
(cappsta) (nls,cedit1,cappsta) 2A		02863
(capptex) (nls,cedit1,capptex) 2B		02876
(carcfil) (nls,cedit1,carcfil) 3A		02872
(cbresta) (nls,cedit1,cbresta) 4A		02866
(cconfildir) (nls,cedit1,cconfildir) 5A		02865
(ccoparcdir) (nls,cedit1,ccoparcdir) 6A		02896
(ccopdir) (nls,cedit1,ccopdir) 6B		02860
(ccopfil) (nls,cedit1,ccopfil) 6C		02873
(ccopgro) (nls,cedit1,ccopgro) 6D		02874
(ccopseqfil) (nls,cedit1,ccopseqfil) 6E		02858
(ccopsta) (nls,cedit1,ccopsta) 6F		02853
(ccoptex) (nls,cedit1,ccoptex) 6G		02883
(ccrefil) (nls,cedit1,ccrefil) 7A		02895
(ccurcon) (nls,cedit1,ccurcon) 9A		02888
(ccurloc) (nls,cedit1,ccurloc) 8A		02854
(cdelallmar) (nls,cedit1,cdelallmar) 10A		02861
(cdelfil) (nls,cedit1,cdelfil) 10C		02855
(cdelgro) (nls,cedit1,cdelgro) 10D		02899
(cdelmar) (nls,cedit1,cdelmar) 10E		02871
(cdelmodfil) (nls,cedit1,cdelmodfil) 10F		02890
(cdelsta) (nls,cedit1,cdelsta) 10G		02889
(cdeltex) (nls,cedit1,cdeltex) 10H		02879
(cexparcdir) (nls,cedit1,cexparcdir) 11A		02878
(cexpcondir) (nls,cedit1,cexpcondir) 11B		02892
(cexpdir) (nls,cedit1,cexpdir) 11C		02885
(cinssta) (nls,cedit1,cinssta) 12B		02881
(cinstex) (nls,cedit1,cinstex) 12C		02880
(cis) (nls,cedit1,cis) 12A		02867
(cloafil) (nls,cedit1,cloafil) 13A		02884
(cloamodfil) (nls,cedit1,cloamodfil) 13B		02875
(clogout) (nls,cedit1,clogout) 14A		02898
(cmarcha) (nls,cedit1,cmarcha) 15A		02856
(cmovfil) (nls,cedit1,cmovfil) 17A		02857
(cmovgro) (nls,cedit1,cmovgro) 17B		02887
(cmovsta) (nls,cedit1,cmovsta) 17D		02859
(cmovtex) (nls,cedit1,cmovtex) 17E		02877
(coutassfil) (nls,cedit1,coutassfil) 18A		02886
(coutjouqui) (nls,cedit1,coutjouqui) 18B		02870
(coutproc) (nls,cedit1,coutproc) 18E		02868
(coutqui) (nls,cedit1,coutqui) 18C		02882
(coutseqfil) (nls,cedit1,coutseqfil) 18D		02897
(incspc) (nls,cedit1,incspc) 10I		02894
(movtst) (nls,cedit1,movtst) 17C		02862

03877

(dkbp) (nls,cedit2,bkbp) 11C1	02946
(compactfile) (nls,cedit2,compactfile) 15E	02958
(copfil) (nls,cedit2,copfil) 15F1	02929
(coprng) (nls,cedit2,coprng) 15F2	02933
(cprigro) (nls,cedit2,cprigro) 2A	02925
(cprijou) (nls,cedit2,cprijou) 2B	02912
(cprires) (nls,cedit2,cprires) 2C	02928
(cprista) (nls,cedit2,cprista) 2D	02961
(cprofil) (nls,cedit2,cprofil) 3A	02934
(crensidfil) (nls,cedit2,crensidfil) 4A	02927
(crepgro) (nls,cedit2,crepgro) 5A	02916
(crepsta) (nls,cedit2,crepsta) 5B	02957
(creptex) (nls,cedit2,creptex) 5C	02935
(cresarcfil) (nls,cedit2,cresarcfil) 6A	02936
(crescasmod) (nls,cedit2,crescasmod) 6B	02930
(creslindef) (nls,cedit2,creslindef) 6C	02939
(cresngro) (nls,cedit2,cresngro) 6D	02926
(cresnsta) (nls,cedit2,cresnsta) 6E	02949
(crestemmod) (nls,cedit2,crestemmod) 6F	02917
(cretarcfil) (nls,cedit2,cretarcfil) 7A	02948
(csetcgro) (nls,cedit2,csetcgro) 8A	02941
(csetcmod) (nls,cedit2,csetcmod) 8B	02920
(csetcsta) (nls,cedit2,csetcsta) 8C	02919
(csetctex) (nls,cedit2,csetctex) 8D	02952
(csetextname) (nls,cedit2,csetextname) 8E	02940
(csetlindef) (nls,cedit2,csetlindef) 8F	02931
(csetngro) (nls,cedit2,csetngro) 8G	02921
(csetnsta) (nls,cedit2,csetnsta) 8H	02938
(csettemmod) (nls,cedit2,csettemmod) 8I	02947
(cshoarcdir) (nls,cedit2,cshoarcdir) 9A	02909
(cshodir) (nls,cedit2,cshodir) 9B	02910
(cshodskspa) (nls,cedit2,cshodskspa) 9C	02905
(cshofilsta) (nls,cedit2,cshofilsta) 9D	02924
(cshofrring) (nls,cedit2,cshofrring) 9G	02953
(cshomarfil) (nls,cedit2,cshomarfil) 9E	02923
(cshonsta) (nls,cedit2,cshonsta) 9F	02903
(cshosrring) (nls,cedit2,cshosrring) 9H	02955
(csorgro) (nls,cedit2,csorgro) 10A	02911
(csort) (nls,cedit2,csort) 10B	02901
(csubgro) (nls,cedit2,csubgro) 11A	02906
(csubsta) (nls,cedit2,csubsta) 11B	02951
(ctragro) (nls,cedit2,ctragro) 12A	02943
(ctrasta) (nls,cedit2,ctrasta) 12B	02932
(ctratex) (nls,cedit2,ctratex) 12C	02950
(ctridir) (nls,cedit2,ctridir) 13A	02937
(cundarcfil) (nls,cedit2,cundarcfil) 14A	02915
(cundfil) (nls,cedit2,cundfil) 14B	02962
(cundmodfil) (nls,cedit2,cundmodfil) 14C	02914
(cupdfil) (nls,cedit2,cupdfil) 15A	02942
(cverfil) (nls,cedit2,cverfil) 16A	02908
(iupper) (nls,cedit2,iupper) 8J	02959
(rensorg) (nls,cedit2,rensorg) 4B	02945
(rjrepn) (nls,cedit2,rjrepn) 5E3	02960
(rjshift) (nls,cedit2,rjshift) 5E2	02913
(rpllit) (nls,cedit2,rpllit) 5D	02918
(sbdelck) (nls,cedit2,sbdelck) 11C2	02907

(setrot)	(nls,cedit2,setrot)	15D	02902
(stbpget)	(nls,cedit2,stbpget)	11C3	02944
(trnchk)	(nls,cedit2,trnchk)	5E1	02904
(trntst)	(nls,cedit2,trntst)	12D	02956
(updtfl)	(nls,cedit2,updtfl)	15B	02954
(updtpg)	(nls,cedit2,updtpg)	15C	02922
			03878

(chkdsp) (nls,chelp,chkdsp) 3G	03751
(clonfil) (nls,chelp,clonfil) 3E	03719
(conbck) (nls,chelp,conbck) 5E	03736
(conint) (nls,chelp,conint) 3F	03723
(feedsw) (nls,chelp,feedsw) 6P	03717
(freemenu) (nls,chelp,freemenu) 3H	03729
(neiphlp) (nls,chelp,helphp) 3C	03755
(mencont) (nls,chelp,mencont) 6K	03725
(menustart) (nls,chelp,menustart) 5C	03722
(moreterm) (nls,chelp,moreterm) 3D	03716
(numfst) (nls,chelp,numfst) 5D	03720
(ofstid) (nls,chelp,ofstid) 6O	03733
(pushent) (nls,chelp,pushent) 7A	03724
(qappender) (nls,chelp,qappender) 6D	03745
(qcolumnate) (nls,chelp,qcolumnate) 6M	03734
(qdirparse) (nls,chelp,qdirparse) 6E	03738
(qdisp) (nls,chelp,qdisp) 3A	03726
(qgetup) (nls,chelp,qgetup) 5A	03727
(qinclude) (nls,chelp,qinclude) 6F	03752
(qlnkspec) (nls,chelp,qlnkspec) 6N	03750
(qmenu) (nls,chelp,qmenu) 6J	03737
(qsearch) (nls,chelp,qsearch) 5B	03742
(qsender) (nls,chelp,qsender) 6C	03731
(qsparse) (nls,chelp,qsparse) 6G	03739
(qspcreset) (nls,chelp,qspcreset) 6H	03743
(qstmt) (nls,chelp,qstmt) 6B	03715
(qstrinit) (nls,chelp,qstrinit) 3B	03747
(qstrip) (nls,chelp,qstrip) 6I	03735
(queryseq) (nls,chelp,queryseq) 6A	03741
(qvalid) (nls,chelp,qvalid) 5F	03740
(samename) (nls,chelp,samename) 5L	03749
(samevw) (nls,chelp,samevw) 5M	03746
(srchfndx) (nls,chelp,srchfndx) 5N	03730
(tstmenu) (nls,chelp,tstmenu) 6L	03732
(wbranm) (nls,chelp,wbranm) 5P	03748
(wbrsrh) (nls,chelp,wbrsrh) 5O	03744
(wfnidit) (nls,chelp,wfnidit) 5G	03721
(wgetadr) (nls,chelp,wgetadr) 5J	03753
(wgetfile) (nls,chelp,wgetfile) 5I	03754
(wprsfile) (nls,chelp,wprsfile) 5H	03728
(wstrtnm) (nls,chelp,wstrtnm) 5K	03718
	03879

(r7) (nls,colsrt,)		03452
(a3ddrec) (nls,colsrt,a3ddrec) 7C		03453
(c3compare) (nls,colsrt,c3compare) 7F		03444
(colgdest) (nls,colsrt,colgdest) 6B		03445
(defkey) (nls,colsrt,defkey) 5B		03455
(gsstid) (nls,colsrt,gsstid) 7E		03450
(kprget) (nls,colsrt,kprget) 6A		03456
(kysrdone) (nls,colsrt,kysrdone) 7G		03449
(lodkey) (nls,colsrt,lodkey) 7D		03457
(s3iftup) (nls,colsrt,s3iftup) 7B		03454
(sort) (nls,colsrt,sort) 7A		03448
(sortbranch) (nls,colsrt,sortbranch) 6C		03446
(sortgrp) (nls,colsrt,sortgrp) 6D		03447
(sortplx) (nls,colsrt,sortplx) 6E		03451
		03880

GAS2, 14-Feb-79 22:37

< NLS, MINISYSGD.NLS.4, > 31

(chrpos) (nls,coresupport,chrpos) 2A
(cirbut) (nls,coresupport,cirbuf) 2B

03295
03294
03881

(bkproc)	(nls,cprograms,bkproc)	21C	03688
(cpcmpfl)	(nls,cprograms,cpcmpfl)	4B	03713
(cpcmproc)	(nls,cprograms,cpcmproc)	4C	03705
(cpconan)	(nls,cprograms,cpconan)	4A	03711
(cpsegg)	(nls,cprograms,cpsegg)	4D1	03689
(dinstupg)	(nls,cprograms,dinstupg)	5	03707
(getuprog)	(nls,cprograms,getuprog)	11	03697
(gpadjbsz)	(nls,cprograms,gpadjbsz)	14	03696
(gpawdsleft)	(nls,cprograms,gpawdsleft)	13	03704
(gpbsz)	(nls,cprograms,gpbsz)	6	03702
(gpconan)	(nls,cprograms,gpconan)	8	03700
(gpexpgm)	(nls,cprograms,gpexpgm)	9	03692
(gpiktrm)	(nls,cprograms,gpiktrm)	22A	03695
(gpget)	(nls,cprograms,gpget)	10	03703
(gpgmrst)	(nls,cprograms,gpgmrst)	7	03693
(gpkill)	(nls,cprograms,gpkill)	19	03699
(gpl10)	(nls,cprograms,gpl10)	15	03690
(gppop)	(nls,cprograms,gppop)	16	03710
(gprnls)	(nls,cprograms,gprnls)	22B	03686
(gprunt)	(nls,cprograms,gprunt)	20	03698
(gpstatus)	(nls,cprograms,gpstatus)	17	03687
(gptxst)	(nls,cprograms,gptxst)	18	03712
(gtctrlbits)	(nls,cprograms,gtctrlbits)	12	03694
(jutval)	(nls,cprograms,jutval)	21B	03708
(marksym)	(nls,cprograms,marksym)	25	03701
(popsym)	(nls,cprograms,popsym)	24	03709
(rplproc)	(nls,cprograms,rplproc)	21A	03714
(upgcnv)	(nls,cprograms,upgcnv)	23	03706
(upgjfn)	(nls,cprograms,upgjfn)	26	03685
(upgsstr)	(nls,cprograms,upgsstr)	17N	03691
			03882

(addcrlf) (nls,csendmail,addcrlf) 5B	03563
(convjdist) (nls,csendmail,convjdist) 8B	03534
(convjsubcol) (nls,csendmail,convjsubcol) 8A	03574
(galjsubcol) (nls,csendmail,galjsubcol) 6X	03579
(getcacc) (nls,csendmail,getcacc) 7A	03539
(getcend) (nls,csendmail,getcend) 7C	03540
(getcfm) (nls,csendmail,getcfm) 7B	03544
(getjaction) (nls,csendmail,getjaction) 6H	03564
(getjauthor) (nls,csendmail,getjauthor) 6I	03543
(getjaxcess) (nls,csendmail,getjaxcess) 6J	03556
(getjclerk) (nls,csendmail,getjclerk) 6K	03581
(getjcomment) (nls,csendmail,getjcomment) 6L	03536
(getjexpedite) (nls,csendmail,getjexpedite) 6M	03546
(getjhcloc) (nls,csendmail,getjhcloc) 6O	03548
(getjinfo) (nls,csendmail,getjinfo) 6P	03560
(getjkeyw) (nls,csendmail,getjkeyw) 6R	03566
(getjlink) (nls,csendmail,getjlink) 6Q	03549
(getjnumber) (nls,csendmail,getjnumber) 6S	03583
(getjobsolates) (nls,csendmail,getjobsolates) 6U	03553
(getjrhc) (nls,csendmail,getjrhc) 6V	03585
(getjsubcol) (nls,csendmail,getjsubcol) 6W	03538
(getjtitle) (nls,csendmail,getjtitle) 6Y	03531
(getjtype) (nls,csendmail,getjtype) 6T	03576
(getjunrecorded) (nls,csendmail,getjunrecorded) 6N	03532
(getjupdates) (nls,csendmail,getjupdates) 6Z	03552
(gtcatent) (nls,csendmail,gtcatent) 7D	03570
(hioco) (nls,csendmail,hioco) 11D	03559
(hiortn) (nls,csendmail,hiortn) 11C	03555
(initjwork) (nls,csendmail,initjwork) 3A	03541
(jlog) (nls,csendmail,jlog) 12B	03557
(jolock) (nls,csendmail,jolock) 9A	03569
(jstatus) (nls,csendmail,jstatus) 5A	03568
(jsubmit) (nls,csendmail,jsubmit) 10A	03547
(jtypchk) (nls,csendmail,jtypchk) 12A	03582
(lockjo) (nls,csendmail,lockjo) 9B	03562
(njs) (nls,csendmail,njs) 11A	03565
(njsverids) (nls,csendmail,njsverids) 11B	03535
(secdist) (nls,csendmail,secdist) 10B	03554
(setcacc) (nls,csendmail,setcacc) 4R	03533
(setjaction) (nls,csendmail,setjaction) 4C	03575
(setjauthor) (nls,csendmail,setjauthor) 4B	03530
(setjaxcess) (nls,csendmail,setjaxcess) 4D	03545
(setjcomment) (nls,csendmail,setjcomment) 4F	03578
(setjexpedite) (nls,csendmail,setjexpedite) 4G	03551
(setjinfo) (nls,csendmail,setjinfo) 4E	03558
(setjkeyword) (nls,csendmail,setjkeyword) 4I	03572
(setjlink) (nls,csendmail,setjlink) 4J	03580
(setjnumber) (nls,csendmail,setjnumber) 4H	03584
(setjobsolates) (nls,csendmail,setjobsolates) 4N	03537
(setjrhc) (nls,csendmail,setjrhc) 4K	03561
(setjsource) (nls,csendmail,setjsource) 4L	03542
(setjsubcol) (nls,csendmail,setjsubcol) 4M	03567
(setjtitle) (nls,csendmail,setjtitle) 4Q	03577
(setjunrecorded) (nls,csendmail,setjunrecorded) 4O	03571
(setjupdates) (nls,csendmail,setjupdates) 4P	03573
(unikjo) (nls,csendmail,unikjo) 9C	03550

GAS2, 14-Feb-79 22:37

< NLS, MINISYSGD.NLS.4, > 34

03883

(appstk) (nls,csyngen,appstk) 14	03761
(ckaltcnt) (nls,csyngen,ckaltcnt) 12	03767
(ckcopcmd) (nls,csyngen,ckcopcmd) 3	03770
(ckcoprul) (nls,csyngen,ckcoprul) 2	03760
(ckcopsub) (nls,csyngen,ckcopsub) 4	03759
(ckoptnode) (nls,csyngen,ckoptnode) 11	03756
(ckshwcmd) (nls,csyngen,ckshwcmd) 6	03771
(ckshwrul) (nls,csyngen,ckshwrul) 5	03765
(ckshwsub) (nls,csyngen,ckshwsub) 7	03757
(gtruln) (nls,csyngen,gtruln) 13	03773
(nprint) (nls,csyngen,nprint) 8	03772
(onode) (nls,csyngen,onode) 10	03766
(pnode) (nls,csyngen,pnode) 9	03764
(prbuglook) (nls,csyngen,prbuglook) 15B	03769
(prconfirmlook) (nls,csyngen,prconfirmlook) 15A	03768
(prhchk) (nls,csyngen,prhchk) 15D	03758
(prjmpcmd) (nls,csyngen,prjmpcmd) 15E	03762
(prnumlook) (nls,csyngen,prnumlook) 15C	03763

03884

(datame) (nls,dspgen,)	03614
(deltad) (nls,dspgen,)	03623
(hazelt) (nls,dspgen,)	03609
(alocda) (nls,dspgen,alocda) 7D	03632
(cdbckc) (nls,dspgen,cdbckc) 4B5	03597
(cklsfreestring) (nls,dspgen,cklsfreestring) 4A4	03598
(clearda) (nls,dspgen,clearda) 7A	03625
(cline) (nls,dspgen,cline) 8H	03635
(clrall) (nls,dspgen,clrall) 6A	03603
(cmdfree) (nls,dspgen,cmdfree) 5D	03610
(cmdinit) (nls,dspgen,cmdinit) 5F	03602
(cnxtaddress) (nls,dspgen,cnxtaddress) 5E	03649
(compisg) (nls,dspgen,compisg) 4B10	03617
(cscreen) (nls,dspgen,cscreen) 8F	03595
(dafrmt) (nls,dspgen,dafrmt) 4A1	03605
(damt) (nls,dspgen,damt) 4A5	03629
(daupdate) (nls,dspgen,daupdate) 4A3	03631
(dealocda) (nls,dspgen,dealocda) 7E	03644
(dline) (nls,dspgen,dline) 8I	03624
(dspace) (nls,dspgen,dspace) 4B4	03633
(endstandout) (nls,dspgen,endstandout) 8P	03622
(frzfrmt) (nls,dspgen,frzfrmt) 4A6	03642
(inline) (nls,dspgen,inline) 8J	03604
(intter) (nls,dspgen,intter) 8M	03627
(ipcmode) (nls,dspgen,ipcmode) 8N	03638
(lpdspstr) (nls,dspgen,lpdspstr) 8A	03621
(lpmarkit) (nls,dspgen,lpmarkit) 8K	03645
(lppclose) (nls,dspgen,lppclose) 8R	03630
(lppint) (nls,dspgen,lppint) 8V	03619
(lppopen) (nls,dspgen,lppopen) 8Q	03637
(lppopmark) (nls,dspgen,lppopmark) 8L	03647
(lppsr) (nls,dspgen,lppsr) 8S	03607
(lppstart) (nls,dspgen,lppstart) 8U	03639
(lppterm) (nls,dspgen,lppterm) 8T	03646
(lprset) (nls,dspgen,lprset) 8E	03600
(lpttywindow) (nls,dspgen,lpttywindow) 8D	03601
(lscopy) (nls,dspgen,lscopy) 4B9	03616
(lsgfrmt) (nls,dspgen,lsgfrmt) 4B2	03596
(lsmove) (nls,dspgen,lsmove) 4B11	03618
(lstidnxt) (nls,dspgen,lstidnxt) 4B8	03628
(mkrset) (nls,dspgen,mkrset) 4D2	03643
(mkrsho) (nls,dspgen,mkrsho) 4D1	03612
(npstrad) (nls,dspgen,npstrad) 4B6	03613
(nxtlsg) (nls,dspgen,nxtlsg) 4B3	03634
(pad) (nls,dspgen,pad) 8B	03615
(position) (nls,dspgen,position) 8C	03636
(prcmds) (nls,dspgen,prcmds) 5B	03626
(pushdc) (nls,dspgen,pushdc) 5A	03599
(resda) (nls,dspgen,resda) 7C	03608
(srch) (nls,dspgen,srch) 4B7	03648
(standout) (nls,dspgen,standout) 80	03640
(stfrmt) (nls,dspgen,stfrmt) 4B1	03641
(supda) (nls,dspgen,supda) 7B	03611
(track) (nls,dspgen,track) 8G	03606
(wrtstr) (nls,dspgen,wrtstr) 5C	03620

(arcflandpc) (nls,execfl,arcflandpc)	3L	03676
(cflapc) (nls,execfl,cflapc)	3P	03653
(chkdev) (nls,execfl,chkdev)	3D	03661
(chkpcs) (nls,execfl,chkpcs)	3E	03658
(coptlandpc) (nls,execfl,coptlandpc)	3N	03660
(delflandpc) (nls,execfl,delflandpc)	3H	03668
(dlbgac) (nls,execfl,dlbgac)	4N	03655
(dlbgas) (nls,execfl,dlbgas)	4R	03664
(dlbglw) (nls,execfl,dlbglw)	4Q	03678
(dlbgpr) (nls,execfl,dlbgpr)	4P	03672
(dlbgtd) (nls,execfl,dlbgtd)	4O	03663
(dlbl dai) (nls,execfl,dlbl dai)	4F	03674
(dlbl di) (nls,execfl,dlbl di)	4G	03650
(dlbl ds) (nls,execfl,dlbl ds)	4M	03680
(dlckdl) (nls,execfl,dlckdl)	4L	03652
(dl driver) (nls,execfl,dl driver)	4A	03665
(dlfmb) (nls,execfl,dlfmb)	4C	03662
(dlgetg) (nls,execfl,dlgetg)	4H	03677
(dlgets) (nls,execfl,dlgets)	4I	03651
(dlgmb) (nls,execfl,dlgmb)	4B	03667
(dlgtblk) (nls,execfl,dlgtblk)	4D	03659
(dlinfb) (nls,execfl,dlinfb)	4J	03671
(dlsnps) (nls,execfl,dlsnps)	4K	03656
(getpcjfn) (nls,execfl,getpcjfn)	3F	03684
(gt desjfn) (nls,execfl,gt desjfn)	3Q	03682
(gthstn) (nls,execfl,gthstn)	3C	03657
(isenabled) (nls,execfl,isenabled)	3G	03670
(mkdirlist) (nls,execfl,mkdirlist)	4E	03681
(movtlandpc) (nls,execfl,movtlandpc)	3M	03666
(parseinput) (nls,execfl,parseinput)	3A	03683
(proflandpc) (nls,execfl,proflandpc)	3K	03679
(proprot) (nls,execfl,proprot)	3Q	03673
(triflandpc) (nls,execfl,triflandpc)	3J	03654
(undtlandpc) (nls,execfl,undtlandpc)	3I	03669
(unprseinpt) (nls,execfl,unprseinpt)	3B	03675

03886

(gtjcpc) (nls,fconst,)	01624
(gtjwrt) (nls,fconst,)	01469
(maddre) (nls,fconst,)	01562
(mbell) (nls,fconst,)	01449
(mbugma) (nls,fconst,)	01479
(mcomba) (nls,fconst,)	01510
(mcomde) (nls,fconst,)	01454
(mcompo) (nls,fconst,)	01481
(mcomwo) (nls,fconst,)	01488
(mdone) (nls,fconst,)	01533
(merrsi) (nls,fconst,)	01542
(minser) (nls,fconst,)	01477
(mlevad) (nls,fconst,)	01592
(mliter) (nls,fconst,)	01496
(mloada) (nls,fconst,)	01512
(mmouse) (nls,fconst,)	01573
(mpopst) (nls,fconst,)	01599
(mquest) (nls,fconst,)	01457
(mrepea) (nls,fconst,)	01462
(mstop) (nls,fconst,)	01608
(msynta) (nls,fconst,)	01459
(mviews) (nls,fconst,)	01600
(addcalit) (nls,fconst,addcalit) 9H1M	01545
(addchar) (nls,fconst,addchar) 9H1E	01584
(append) (nls,fconst,append) 8B2D	01463
(arcoprty) (nls,fconst,arcoprty) 9K2A3	01546
(ascalt) (nls,fconst,ascalt) 6A20	01483
(ascbst) (nls,fconst,ascbst) 6A1	01588
(asctsw) (nls,fconst,asctsw) 6A2	01495
(backup) (nls,fconst,backup) 9E1E	01614
(begmsg) (nls,fconst,begmsg) 9K3B1	01538
(cleanup) (nls,fconst,cleanup) 9E1C	01582
(clearcf1) (nls,fconst,clearcf1) 9H1A	01615
(cmdelete) (nls,fconst,cmdelete) 3A	01532
(cssh2) (nls,fconst,cssh2) 4A	01460
(ctia) (nls,fconst,ctia) 6A3	01589
(ctid) (nls,fconst,ctid) 6A4	01473
(ctle) (nls,fconst,ctle) 6A5	01603
(ctlext) (nls,fconst,ctlext) 8A4	01585
(ctlf) (nls,fconst,ctlf) 6A6	01560
(ctlg) (nls,fconst,ctlg) 6A7	01504
(ctln) (nls,fconst,ctln) 6A8	01610
(ctlo) (nls,fconst,ctlo) 6A9	01563
(ctlp) (nls,fconst,ctlp) 6A10	01619
(ctlr) (nls,fconst,ctlr) 6A11	01522
(ctls) (nls,fconst,ctls) 6A12	01494
(ctlt) (nls,fconst,ctlt) 6A13	01489
(ctlu) (nls,fconst,ctlu) 6A14	01598
(ctlv) (nls,fconst,ctlv) 6A15	01550
(ctlw) (nls,fconst,ctlw) 6A16	01482
(ctlx) (nls,fconst,ctlx) 6A17	01476
(ctly) (nls,fconst,ctly) 6A18	01605
(ctlz) (nls,fconst,ctlz) 6A19	01506
(ctty) (nls,fconst,ctty) 9K2A1	01502
(current) (nls,fconst,current) 9K4B1	01596
(cwacc) (nls,fconst,cwacc) 7B5	01554

(d2sel) (nls,fconst,d2sel) 9I1A	01548
(defvs1) (nls,fconst,defvs1) 9J2A	01561
(defvs2) (nls,fconst,defvs2) 9J2B	01623
(dev33) (nls,fconst,dev33) 9K2A5	01553
(dev35) (nls,fconst,dev35) 9K2A6	01526
(dev37) (nls,fconst,dev37) 9K2A7	01499
(devlproc) (nls,fconst,devlproc) 9K2A14	01551
(devsri) (nls,fconst,devsri) 9K2A9	01572
(devtiex) (nls,fconst,devtiex) 9K2A8	01569
(dntyp) (nls,fconst,dntyp) 9K6A1	01571
(dspalif) (nls,fconst,dspalif) 9K6A4	01559
(eacc) (nls,fconst,eacc) 7B3	01524
(echda) (nls,fconst,echda) 9K3B2	01505
(echostr) (nls,fconst,echostr) 9H1E	01613
(entind) (nls,fconst,entind) 11A2	01471
(errsig) (nls,fconst,errsig) 3F	01485
(exeext) (nls,fconst,exeext) 8A3	01576
(fbaddlit) (nls,fconst,fbaddlit) 9H1L	01455
(fbendlit) (nls,fconst,fbendlit) 9H1O	01517
(fbpop) (nls,fconst,fbpop) 9H1J	01604
(ffacctime) (nls,fconst,ffacctime) 10C1	01484
(ffcnt) (nls,fconst,ffcnt) 10C5	01577
(ffglcnt) (nls,fconst,ffglcnt) 10C7	01541
(ffglmod) (nls,fconst,ffglmod) 10C6	01498
(ffjobitab) (nls,fconst,ffjobitab) 10C9	01472
(ffjobmax) (nls,fconst,ffjobmax) 10C8	01480
(ffprtab) (nls,fconst,ffprtab) 10C2	01535
(ffrftab) (nls,fconst,ffrftab) 10C4	01611
(fftftab) (nls,fconst,fftftab) 10C3	01452
(fix1) (nls,fconst,fix1) 9C1B1	01617
(frrmax) (nls,fconst,frrmax) 9K5A1	01448
(fulldisplay) (nls,fconst,fulldisplay) 9K2B2	01606
(fullprompts) (nls,fconst,fullprompts) 9F1C	01547
(gaderr) (nls,fconst,gaderr) 3D	01450
(gtjid1) (nls,fconst,gtjid1) 8B1H	01475
(gtjigdel) (nls,fconst,gtjigdel) 8B1M	01464
(gtjinfo) (nls,fconst,gtjinfo) 8B1I	01607
(gtjoif) (nls,fconst,gtjoif) 8B1B	01520
(gtjoisf) (nls,fconst,gtjoisf) 8B1E	01514
(gtjoof) (nls,fconst,gtjoof) 8B1A	01490
(gtjoosf) (nls,fconst,gtjoosf) 8B1D	01625
(gtjprf) (nls,fconst,gtjprf) 8B1C	01516
(gtjprv) (nls,fconst,gtjprv) 8B1G	01525
(gtjred) (nls,fconst,gtjred) 8B1L	01453
(gtjstr) (nls,fconst,gtjstr) 8B1F	01458
(imlac0) (nls,fconst,imlac0) 9K2A10	01555
(imlac1) (nls,fconst,imlac1) 9K2A11	01507
(incues) (nls,fconst,incues) 9H1I	01570
(intca) (nls,fconst,intca) 2A	01578
(interperr) (nls,fconst,interperr) 3C	01583
(jinstf) (nls,fconst,jinstf) 8B1P	01521
(keyword) (nls,fconst,keyword) 9H1C	01626
(manticipitory) (nls,fconst,manticipitory) 9C1D	01467
(mdemand) (nls,fconst,mdemand) 9C1C	01515
(mexpert) (nls,fconst,mexpert) 9C1A	01487
(mfixed) (nls,fconst,mfixed) 9C1B	01565

(msetbl)	(nls,fconst,msetbl)	5A	01513
(msimtmax)	(nls,fconst,msimtmax)	10A6	01575
(msstring)	(nls,fconst,msstring)	10A5	01497
(mstg2l)	(nls,fconst,mstg2l)	10A4	01534
(mstgl)	(nls,fconst,mstgl)	10A3	01558
(mstu1)	(nls,fconst,mstu1)	10A1	01593
(mstu2)	(nls,fconst,mstu2)	10A2	01531
(mtruncerr)	(nls,fconst,mtruncerr)	10B1	01587
(multchar)	(nls,fconst,multchar)	9D1B	01478
(nettty)	(nls,fconst,nettty)	9K2A12	01621
(nlsext)	(nls,fconst,nlsext)	8A1	01509
(noprompts)	(nls,fconst,noprompts)	9F1A	01591
(norstcalit)	(nls,fconst,norstcalit)	9H1P	01594
(nsaoprty)	(nls,fconst,nsaoprty)	9K2A4	01508
(nttyp)	(nls,fconst,nttyp)	9K6A3	01503
(nwheecinits)	(nls,fconst,nwheecinits)	12A1	01528
(offline)	(nls,fconst,offline)	9K2A13	01474
(onechar)	(nls,fconst,onechar)	9D1A	01574
(oprty)	(nls,fconst,oprty)	9K2A2	01564
(orgstid)	(nls,fconst,orgstid)	9K6B1	01465
(parsehelp)	(nls,fconst,parsehelp)	9E1G	01544
(parseqmark)	(nls,fconst,parseqmark)	9E1H	01590
(parsing)	(nls,fconst,parsing)	9E1A	01568
(partprompts)	(nls,fconst,partprompts)	9F1B	01527
(pnext)	(nls,fconst,pnext)	9E1B	01518
(popselect)	(nls,fconst,popselect)	9E1F	01595
(popstate)	(nls,fconst,popstate)	3B	01493
(psellen)	(nls,fconst,psellen)	9I1B	01491
(racc)	(nls,fconst,racc)	7B1	01492
(read)	(nls,fconst,read)	8B2B	01537
(readwrite)	(nls,fconst,readwrite)	8B2C	01470
(rechostr)	(nls,fconst,rechostr)	9H1F	01566
(remfudge)	(nls,fconst,remfudge)	9K3A1	01602
(remlcm)	(nls,fconst,remlcm)	9K3B8	01539
(remrdda)	(nls,fconst,remrdda)	9K3B4	01468
(remrsda)	(nls,fconst,remrsda)	9K3B5	01486
(remsdda)	(nls,fconst,remsdda)	9K3B3	01601
(remtsf)	(nls,fconst,remtsf)	9K3B7	01536
(remtsn)	(nls,fconst,remtsn)	9K3B6	01597
(rinistr)	(nls,fconst,rinistr)	9K3C1	01581
(rnglnt)	(nls,fconst,rnglnt)	11A4	01540
(rngmax)	(nls,fconst,rngmax)	11A3	01461
(rptparse)	(nls,fconst,rptparse)	9E1D	01466
(rthawed)	(nls,fconst,rthawed)	8B2E	01552
(rtmax)	(nls,fconst,rtmax)	9K4A2	01529
(rwthawed)	(nls,fconst,rwthawed)	8B2F	01627
(savext)	(nls,fconst,savext)	8A2	01557
(sbfinish)	(nls,fconst,sbfinish)	9B1C	01530
(sbpop)	(nls,fconst,sbpob)	9B1D	01511
(sbrentry)	(nls,fconst,sbrentry)	9B1E	01456
(sbrun)	(nls,fconst,sbrun)	9B1B	01579
(sbstart)	(nls,fconst,sbstart)	9B1A	01586
(sgtjff)	(nls,fconst,sgtjff)	8B1D	01618
(srrmax)	(nls,fconst,srrmax)	9K5A2	01519
(startparams)	(nls,fconst,startparams)	9H1H	01612
(startrec)	(nls,fconst,startrec)	9H1D	01447

(statesig) (nls,fconst,statesig) 3E	01620
(stkmax) (nls,fconst,stkmax) 11A1	01500
(syserr) (nls,fconst,syserr) 9K4A1	01556
(tersemode) (nls,fconst,tersemode) 9G1B	01451
(tntyp) (nls,fconst,tntyp) 9K6A2	01567
(tuacc) (nls,fconst,tuacc) 7B4	01523
(typecalit) (nls,fconst,typecalit) 9H1K	01501
(typelit) (nls,fconst,typelit) 9H1G	01549
(typenulllit) (nls,fconst,typenulllit) 9H1N	01543
(typewriter) (nls,fconst,typewriter) 9K2B1	01622
(verbsmode) (nls,fconst,verbsmode) 9G1A	01616
(wacc) (nls,fconst,wacc) 7B2	01580
(write) (nls,fconst,write) 8B2A	01609
	03887

(bmauxs) (nls,fdata,)	0995
(bmsave) (nls,fdata,)	01235
(inhelb) (nls,fdata,)	01211
(ldavgs) (nls,fdata,)	0981
(ldavgt) (nls,fdata,)	01145
(mlastc) (nls,fdata,)	01128
(mlastf) (nls,fdata,)	01033
(mlastt) (nls,fdata,)	01194
(nocira) (nls,fdata,)	01185
(padstr) (nls,fdata,)	01168
(acursub) (nls,fdata,acursub) 12B1C	01248
(altdfl) (nls,fdata,altdfl) 9G1	01296
(altdir) (nls,fdata,altdir) 9E4	01301
(altext) (nls,fdata,altext) 9E5	01239
(altinp) (nls,fdata,altinp) 9L1A	01009
(analtme) (nls,fdata,analtme) 14A2	01059
(analyzing) (nls,fdata,analyzing) 14A1	01147
(arowon) (nls,fdata,arowon) 9L2H	01098
(asupsub) (nls,fdata,asupsub) 12B1D	01190
(athelp) (nls,fdata,athelp) 9J1C	01075
(auxinput) (nls,fdata,auxinput) 9Q4	01187
(auxmd2) (nls,fdata,auxmd2) 9Q3	01113
(auxmod) (nls,fdata,auxmod) 9Q2	01053
(auxsav) (nls,fdata,auxsav) 9Q1	01196
(auxtp1) (nls,fdata,auxtp1) 9Q5	01074
(auxtp2) (nls,fdata,auxtp2) 9Q6	01200
(auxwrk) (nls,fdata,auxwrk) 9Q7	01278
(bmarg) (nls,fdata,bmarg) 9R5D	01094
(bmcnt) (nls,fdata,bmcnt) 9B6	01006
(buff) (nls,fdata,buff) 9H1	01202
(buffct) (nls,fdata,buffct) 9H4	01230
(buffn) (nls,fdata,buffn) 9H3	01157
(buffs) (nls,fdata,buffs) 9H2	01222
(buffsz) (nls,fdata,buffsz) 9H5	01215
(bugmarks) (nls,fdata,bugmarks) 9I1	01141
(bugreg) (nls,fdata,bugreg) 9L2E	01245
(calcaux) (nls,fdata,calcaux) 9J1B	01032
(cdstd1) (nls,fdata,cdstd1) 9R2A	0994
(cdstd2) (nls,fdata,cdstd2) 9R2B	01306
(cdstop) (nls,fdata,cdstop) 9R2C	01236
(cdstr1) (nls,fdata,cdstr1) 9E12	01134
(cdstr2) (nls,fdata,cdstr2) 9E13	01090
(cdtype) (nls,fdata,cdtype) 9R2D	01002
(cflarw) (nls,fdata,cflarw) 9E8	01135
(cflbase) (nls,fdata,cflbase) 9R5F1	01030
(cflbottom) (nls,fdata,cflbottom) 9R5F5	01302
(cflcsize) (nls,fdata,cflcsize) 9R5F6	01286
(cflda) (nls,fdata,cflda) 9D3	01251
(cflfont) (nls,fdata,cflfont) 9R5F7	0976
(cflhinc) (nls,fdata,cflhinc) 9R5F8	01277
(cflleft) (nls,fdata,cflleft) 9R5F2	01265
(cflpos) (nls,fdata,cflpos) 9L2F	01261
(cflright) (nls,fdata,cflright) 9R5F3	01293
(cflstr) (nls,fdata,cflstr) 9E6	01180
(cfltop) (nls,fdata,cfltop) 9R5F4	01107
(cflvinc) (nls,fdata,cflvinc) 9R5F9	01176

(cflx) (nls,fdata,cflx) 9R5F10	01035
(cfly) (nls,fdata,cfly) 9R5F11	01285
(clsbitables) (nls,fdata,clsbitables) 8C1C	01054
(cmarkflg) (nls,fdata,cmarkflg) 9R4F	01284
(cmdctl) (nls,fdata,cmdctl) 12D1A	01121
(comchr) (nls,fdata,comchr) 9J3A2	01240
(confre) (nls,fdata,confre) 9J4E	01004
(conreg) (nls,fdata,conreg) 9E14	01136
(conrng) (nls,fdata,conrng) 9J4B	01045
(console) (nls,fdata,console) 7C3	01303
(continue) (nls,fdata,continue) 9R4A	01174
(cshift) (nls,fdata,cshift) 9R4J	01273
(csrarmd) (nls,fdata,csrarmd) 9L2G	01066
(ctrichar) (nls,fdata,ctrlchar) 9R4H	01179
(cuno) (nls,fdata,cuno) 7C1	01057
(curchr) (nls,fdata,curchr) 9P1	01100
(curindex) (nls,fdata,curindex) 9J4D	01139
(curmbt) (nls,fdata,curmbt) 9L2B	01049
(curstk) (nls,fdata,curstk) 9J4C	01271
(dabt) (nls,fdata,dabt) 8C1A	01133
(dabtsize) (nls,fdata,dabtsize) 8C1E	01013
(dacnt) (nls,fdata,dacnt) 9B3	01105
(defttysim) (nls,fdata,defttysim) 9A7	0978
(derivspc) (nls,fdata,derivspc) 9J11B	01233
(dpyarea) (nls,fdata,dpyarea) 9B1	01058
(dpyend) (nls,fdata,dpyend) 9B1A	01076
(drmdcr) (nls,fdata,drmdcr) 9E2	01217
(dspjfn) (nls,fdata,dspjfn) 11A	0989
(echoda) (nls,fdata,echoda) 9D4	01237
(echofg) (nls,fdata,echofg) 9R4G	01199
(entcon) (nls,fdata,entcon) 9J6B	01269
(exquery) (nls,fdata,exquery) 9R4D	01072
(failnm) (nls,fdata,failnm) 9J11A	01008
(frzmax) (nls,fdata,frzmax) 9K4	01000
(fvstk) (nls,fdata,fvstk) 9I3	01209
(fwrdr) (nls,fdata,fwrdr) 9J10A	01017
(fzfree) (nls,fdata,fzfree) 9K3	01122
(fzlst) (nls,fdata,fzlst) 9K2A	0975
(fzlst) (nls,fdata,fzlst) 9K2	01046
(gcords) (nls,fdata,gcords) 9L2D	01210
(giblst) (nls,fdata,giblst) 9J10B	01243
(gstks) (nls,fdata,gstks) 5A1A	01242
(hdebug) (nls,fdata,hdebug) 9J9A	0985
(helploc) (nls,fdata,helploc) 9J1A	01082
(hlpfileno) (nls,fdata,hlpfileno) 9J2B	01152
(hseger) (nls,fdata,hseger) 9J2J	01254
(initsr) (nls,fdata,initsr) 9E10	01205
(inpjfn) (nls,fdata,inpjfn) 9N2	01264
(inpwatchjfn) (nls,fdata,inpwatchjfn) 9N1	01191
(intmsf) (nls,fdata,intmsf) 2B	0984
(intmsg) (nls,fdata,intmsg) 3A	01250
(jnlsbn) (nls,fdata,jnlsbn) 9A9	0999
(kwrstate) (nls,fdata,kwrstate) 12A1A	01079
(ldspjfn) (nls,fdata,ldspjfn) 11B	01182
(linkcns1) (nls,fdata,linkcns1) 9A1	01109
(litapflag) (nls,fdata,litapflag) 9L2N	01047

(litbase) (nls,fdata,litbase) 9R5K1	01216
(litbottom) (nls,fdata,litbottom) 9R5K5	01084
(litcolumn) (nls,fdata,litcolumn) 9G10	0997
(litcsize) (nls,fdata,litcsize) 9R5K6	01016
(litda) (nls,fdata,litda) 9D9	01051
(litdahandle) (nls,fdata,litdahandle) 9G2	01188
(litfont) (nls,fdata,litfont) 9R5K7	01070
(lithinc) (nls,fdata,lithinc) 9R5K8	01126
(litleft) (nls,fdata,litleft) 9R5K2	01062
(litline) (nls,fdata,litline) 9G3	01120
(litreset) (nls,fdata,litreset) 9L2L	01106
(litright) (nls,fdata,litright) 9R5K3	01283
(litr marg) (nls,fdata,litr marg) 9G6	01231
(litstore) (nls,fdata,litstore) 9G8	01065
(litstring) (nls,fdata,litstring) 9G9	01083
(littabs) (nls,fdata,littabs) 9E17	01181
(littop) (nls,fdata,littop) 9R5K4	0998
(littyflag) (nls,fdata,littyflag) 9L2M	01158
(litvinc) (nls,fdata,litvinc) 9R5K9	0992
(litx) (nls,fdata,litx) 9R5K10	01056
(lity) (nls,fdata,lity) 9R5K11	01071
(lmarg) (nls,fdata,lmarg) 9R5B	01189
(lpbaudfactor) (nls,fdata,lpbaudfactor) 11C	01096
(lpdipad) (nls,fdata,lpdipad) 11G	01298
(lpperrec) (nls,fdata,lpperrec) 11U1	01068
(lpilpad) (nls,fdata,lpilpad) 11H	01276
(lplitline) (nls,fdata,lplitline) 9G4	01218
(lplitreset) (nls,fdata,lplitreset) 9G5	01085
(lppadchr) (nls,fdata,lppadchr) 11J	01089
(lppch) (nls,fdata,lppch) 11N	01183
(lppcol) (nls,fdata,lppcol) 11O	01289
(lppform) (nls,fdata,lppform) 11Q	01227
(lppjfn) (nls,fdata,lppjfn) 11T	01297
(lpplin) (nls,fdata,lpplin) 11P	01175
(lpppad) (nls,fdata,lpppad) 11S	01219
(lpptimout) (nls,fdata,lpptimout) 11R	0990
(lprng0) (nls,fdata,lprng0) 11U2	01154
(lprng1) (nls,fdata,lprng1) 11U3	01256
(lprng2) (nls,fdata,lprng2) 11U4	01018
(lprng3) (nls,fdata,lprng3) 11U5	01077
(lprng4) (nls,fdata,lprng4) 11U6	01142
(lprng5) (nls,fdata,lprng5) 11U7	01257
(lprng6) (nls,fdata,lprng6) 11U8	01019
(lprng7) (nls,fdata,lprng7) 11U9	01112
(lprng8) (nls,fdata,lprng8) 11U10	0972
(lprng9) (nls,fdata,lprng9) 11U11	01258
(lprngp) (nls,fdata,lprngp) 11U12	01247
(lptab) (nls,fdata,lptab) 11I	01023
(lptype) (nls,fdata,lptype) 11F	01234
(lpxmax) (nls,fdata,lpxmax) 11D	01064
(lpymax) (nls,fdata,lpymax) 11E	01101
(lsbitables) (nls,fdata,lsbitables) 8C1B	01300
(lsbtsize) (nls,fdata,lsbtsize) 8C1D	01138
(ltbase) (nls,fdata,ltbase) 9R5I1	01166
(ltbottom) (nls,fdata,ltbottom) 9R5I5	01097
(lrcsize) (nls,fdata,lrcsize) 9R5I6	0993

(lthinc) (nls,fdata,lthinc) 9R5I7	01163
(lthinc) (nls,fdata,lthinc) 9R5I8	01207
(ltleft) (nls,fdata,ltleft) 9R5I2	01291
(ltright) (nls,fdata,ltright) 9R5I3	01129
(ltsmall) (nls,fdata,ltsmall) 9L2I	01061
(ltop) (nls,fdata,ltop) 9R5I4	01186
(ltvinc) (nls,fdata,ltvinc) 9R5I9	01161
(ltvsda) (nls,fdata,ltvsda) 9D5	01080
(ltx) (nls,fdata,ltx) 9R5I10	01130
(lty) (nls,fdata,lty) 9R5I11	01025
(lvdjnm) (nls,fdata,lvdjnm) 9R1B	01173
(mdshare) (nls,fdata,mdshare) 9010	01221
(menchr) (nls,fdata,menchr) 9J3A1	01193
(mentyp) (nls,fdata,mentyp) 9J3B1	01116
(mnaproc) (nls,fdata,mnaproc) 909	01010
(modeon) (nls,fdata,modeon) 7D2	01148
(moremenu) (nls,fdata,moremenu) 9J4A	01268
(msdbit) (nls,fdata,msdbit) 9L2A	01172
(msgbase) (nls,fdata,msgbase) 9R5L1	01050
(msgbottom) (nls,fdata,msgbottom) 9R5L5	01086
(msgcsize) (nls,fdata,msgcsize) 9R5L6	01212
(msgda) (nls,fdata,msgda) 9D10	01170
(msgfont) (nls,fdata,msgfont) 9R5L7	01131
(msghinc) (nls,fdata,msghinc) 9R5L8	01063
(msgleft) (nls,fdata,msgleft) 9R5L2	01024
(msgreset) (nls,fdata,msgreset) 9L20	01259
(msgright) (nls,fdata,msgright) 9R5L3	01198
(msgtop) (nls,fdata,msgtop) 9R5L4	01241
(msgvinc) (nls,fdata,msgvinc) 9R5L9	01229
(msgx) (nls,fdata,msgx) 9R5L10	01039
(msgy) (nls,fdata,msgy) 9R5L11	01055
(msjfn) (nls,fdata,msjfn) 902	01160
(msrrlp) (nls,fdata,msrrlp) 901	01208
(msrtab) (nls,fdata,msrtab) 903	0987
(msumrap) (nls,fdata,msumrap) 9011	0988
(msumrcnt) (nls,fdata,msumrcnt) 9012	01048
(multiflg) (nls,fdata,multiflg) 9J2K	0991
(multyp) (nls,fdata,multyp) 9J3B2	01125
(nambase) (nls,fdata,nambase) 9R5G1	01197
(nambottom) (nls,fdata,nambottom) 9R5G5	01015
(namcsize) (nls,fdata,namcsize) 9R5G6	01305
(namda) (nls,fdata,namda) 9D7	01144
(namenu11) (nls,fdata,namenu11) 9L2K	01155
(namereset) (nls,fdata,namereset) 9L2P	01151
(namfont) (nls,fdata,namfont) 9R5G7	0983
(namhinc) (nls,fdata,namhinc) 9R5G8	01044
(namleft) (nls,fdata,namleft) 9R5G2	01260
(namreg) (nls,fdata,namreg) 9E16	01088
(namright) (nls,fdata,namright) 9R5G3	01167
(namtop) (nls,fdata,namtop) 9R5G4	01102
(namvinc) (nls,fdata,namvinc) 9R5G9	01103
(namx) (nls,fdata,namx) 9R5G10	01294
(namy) (nls,fdata,namy) 9R5G11	01036
(newstk) (nls,fdata,newstk) 9J5A	01195
(nlcron) (nls,fdata,nlcron) 8A	01274
(nldevice) (nls,fdata,nldevice) 7C2	01149

(nimode) (nls,fdata,nimode) 9R4B	01213
(nlparse) (nls,fdata,nlparse) 9R4C	01232
(nlsdae) (nls,fdata,nlsdae) 9B2B	01164
(nlsdas) (nls,fdata,nlsdas) 9B2	01137
(nlsnam) (nls,fdata,nlsnam) 9E11	01238
(nlssbn) (nls,fdata,nlssbn) 9A8	01263
(notetime) (nls,fdata,notetime) 14A3	01069
(nwheelf) (nls,fdata,nwheelf) 7C4	01037
(ordmbt) (nls,fdata,ordmbt) 9L2C	01022
(oshift) (nls,fdata,oshift) 9R4I	01117
(overscreen) (nls,fdata,overscreen) 9J2I	01220
(pbugpt) (nls,fdata,pbugpt) 9C3	01223
(pbugtb) (nls,fdata,pbugtb) 9C1	01012
(pbugte) (nls,fdata,pbugte) 9C2	01060
(psisksz) (nls,fdata,psisksz) 5A2A	01095
(psistk) (nls,fdata,psistk) 8B1A	01119
(pstk) (nls,fdata,pstk) 12C1B	01150
(pstkp) (nls,fdata,pstkp) 12C1A	01042
(pstksz) (nls,fdata,pstksz) 5A1B	0974
(putbackchar) (nls,fdata,putbackchar) 11K	01031
(qagain) (nls,fdata,qagain) 9J2A	01005
(qcolwidth) (nls,fdata,qcolwidth) 9J8C	01295
(qda) (nls,fdata,qda) 9J2C	01177
(qdavs2) (nls,fdata,qdavs2) 9J2H	01014
(qdavspc) (nls,fdata,qdavspc) 9J2G	01123
(qmenucnt) (nls,fdata,qmenucnt) 9J8B	01275
(qmenumax) (nls,fdata,qmenumax) 9J8A	01028
(qmrkon) (nls,fdata,qmrkon) 9L2J	01099
(qnewstmt) (nls,fdata,qnewstmt) 9J2F	01146
(qspecs) (nls,fdata,qspecs) 9J7B	01262
(qspecs) (nls,fdata,qspecs) 9J7A	01110
(qstorblk) (nls,fdata,qstorblk) 9J2E	01007
(qsw) (nls,fdata,qsw) 9J2D	01011
(rawchr) (nls,fdata,rawchr) 9P2	01287
(recrunfil) (nls,fdata,recrunfil) 13B1A	01304
(rmarg) (nls,fdata,rmarg) 9R5A	01091
(rmdcr) (nls,fdata,rmdcr) 9E1	01165
(rtfree) (nls,fdata,rtfree) 9B4	01078
(savcfl) (nls,fdata,savcfl) 9E7	01169
(savedda) (nls,fdata,savedda) 9A3	01132
(savetda) (nls,fdata,savetda) 9A4	01226
(savevspec) (nls,fdata,savevspec) 7E1A	01280
(savnldvice) (nls,fdata,savnldvice) 9A2	01267
(sflg) (nls,fdata,sflg) 9J2L	01281
(spacecol) (nls,fdata,spacecol) 9G7	01225
(spacestr) (nls,fdata,spacestr) 9E3	01124
(subbase) (nls,fdata,subbase) 9R5H1	0996
(subbottom) (nls,fdata,subbottom) 9R5H5	0986
(subcsize) (nls,fdata,subcsize) 9R5H6	01299
(subda) (nls,fdata,subda) 9D8	0980
(subfont) (nls,fdata,subfont) 9R5H7	01292
(subhinc) (nls,fdata,subhinc) 9R5H8	01114
(subleft) (nls,fdata,subleft) 9R5H2	01203
(subright) (nls,fdata,subright) 9R5H3	01214
(subtop) (nls,fdata,subtop) 9R5H4	01178
(subvinc) (nls,fdata,subvinc) 9R5H9	01184

(subx) (nls,fdata,subx) 9R5H10	01201
(suby) (nls,fdata,suby) 9R5H11	0979
(symtflg) (nls,fdata,symtflg) 13A1A	01171
(syncur) (nls,fdata,syncur) 12B1E	01040
(synstore) (nls,fdata,synstore) 12B1B	01255
(synsubs) (nls,fdata,synsubs) 12B1A	01115
(sysbreak) (nls,fdata,sysbreak) 9F1	01270
(sysvspec) (nls,fdata,sysvspec) 9R1A1	01003
(tabase) (nls,fdata,tabase) 9R5E1	01081
(tabottom) (nls,fdata,tabottom) 9R5E5	01143
(tabstore) (nls,fdata,tabstore) 9G11	01162
(tacsiz) (nls,fdata,tacsiz) 9R5E6	01228
(tada) (nls,fdata,tada) 9D2	01204
(tahinc) (nls,fdata,tahinc) 9R5E8	01246
(taleft) (nls,fdata,taleft) 9R5E2	01043
(tamind) (nls,fdata,tamind) 9R5E12	01282
(taright) (nls,fdata,taright) 9R5E3	01224
(tatop) (nls,fdata,tatop) 9R5E4	01067
(tavinc) (nls,fdata,tavinc) 9R5E9	01021
(tax) (nls,fdata,tax) 9R5E10	01041
(tay) (nls,fdata,tay) 9R5E11	01140
(tda) (nls,fdata,tda) 9D1	01118
(tmarg) (nls,fdata,tmarg) 9R5C	01029
(todlit) (nls,fdata,todlit) 9R4M	0982
(topcon) (nls,fdata,topcon) 9J6A	01206
(tracking) (nls,fdata,tracking) 11L	01253
(tracksend) (nls,fdata,tracksend) 11M	01272
(trnsli) (nls,fdata,trnsli) 9F3	01159
(trnslo) (nls,fdata,trnslo) 9F2	01290
(ttyda) (nls,fdata,ttyda) 9A6	01052
(ttysim) (nls,fdata,ttysim) 9A5	01249
(typefg) (nls,fdata,typefg) 9R4E	01093
(untyp) (nls,fdata,untyp) 9J3B3	01127
(uojfn) (nls,fdata,uojfn) 8D1A	01244
(vsbase) (nls,fdata,vsbase) 9R5J1	01111
(vsbottom) (nls,fdata,vsbottom) 9R5J5	01192
(vscsize) (nls,fdata,vscsize) 9R5J6	01092
(vsfont) (nls,fdata,vsfont) 9R5J7	01087
(vshinc) (nls,fdata,vshinc) 9R5J8	01001
(vsleft) (nls,fdata,vsleft) 9R5J2	01288
(vspcda) (nls,fdata,vspcda) 9D6	01279
(vspsav) (nls,fdata,vspsav) 9B5	01266
(vspstr) (nls,fdata,vspstr) 9E9	01252
(vsright) (nls,fdata,vsright) 9R5J3	01104
(vsstr1) (nls,fdata,vsstr1) 10A	0977
(vsstr2) (nls,fdata,vsstr2) 10B	01038
(vstop) (nls,fdata,vstop) 9R5J4	01026
(vsvinc) (nls,fdata,vsvinc) 9R5J9	01108
(vsx) (nls,fdata,vsx) 9R5J10	01156
(vsy) (nls,fdata,vsy) 9R5J11	01153
(wordshift) (nls,fdata,wordshift) 9R4L	01034
(wrddreg) (nls,fdata,wrddreg) 9E15	01073
(wshift) (nls,fdata,wshift) 9R4K	01027
(xacs) (nls,fdata,xacs) 2A	01020
(xsmode) (nls,fdata,xsmode) 7D1	0973

(badfil) (nls,filmp, badfil) 7B1	02964
(copfgrp) (nls,filmp, copfgrp) 5B3	03053
(copgrp) (nls,filmp, copgrp) 5B2	02988
(copit) (nls,filmp, copit) 5E2	03044
(copnam) (nls,filmp, copnam) 6C5	03000
(copplist) (nls,filmp, copplist) 5C1	03032
(copprop) (nls,filmp, copprop) 5D2	02995
(creit) (nls,filmp, creit) 5E1	02967
(crenod) (nls,filmp, crenod) 5B1	03040
(crepc) (nls,filmp, crepc) 8C1	02993
(crepr2) (nls,filmp, crepr2) 6A2	03007
(creprop) (nls,filmp, creprop) 5D1	02997
(delgrp) (nls,filmp, delgrp) 5B6	03056
(delit) (nls,filmp, delit) 5E4	03058
(delprop) (nls,filmp, delprop) 5D4	03057
(fchtxt) (nls,filmp, fchtxt) 4F10	03030
(filhdr) (nls,filmp, filhdr) 7D1	03054
(freintree) (nls,filmp, freintree) 6A4	02971
(freplist) (nls,filmp, freplist) 5C2	03035
(freprop) (nls,filmp, freprop) 6A1	03039
(frerng) (nls,filmp, frerng) 6E3	03020
(frzblk) (nls,filmp, frzblk) 7A2	03034
(frzrib) (nls,filmp, frzrib) 7A1	03028
(gcol) (nls,filmp, gcol) 6F3	02991
(getail) (nls,filmp, getail) 4A3	02982
(getbck) (nls,filmp, getbck) 4A9	03005
(getdblen) (nls,filmp, getdblen) 4F8	03015
(getend) (nls,filmp, getend) 4A7	03011
(getfhd) (nls,filmp, getfhd) 4B2	03048
(getftl) (nls,filmp, getftl) 4B1	03024
(gethed) (nls,filmp, gethed) 4A5	02992
(getint) (nls,filmp, getint) 4F5	03046
(getitree) (nls,filmp, getitree) 4E1	02974
(getnam) (nls,filmp, getnam) 4B4	03049
(getnmdl) (nls,filmp, getnmdl) 4F3	03003
(getnmf) (nls,filmp, getnmf) 4B3	03026
(getnxt) (nls,filmp, getnxt) 4A10	03025
(getorf) (nls,filmp, getorf) 4F2	02972
(getorg) (nls,filmp, getorg) 4F1	02973
(getphed) (nls,filmp, getphed) 4F4	03016
(getprd) (nls,filmp, getprd) 4A6	03047
(getpsdb) (nls,filmp, getpsdb) 4D1	03052
(getptab) (nls,filmp, getptab) 6C2	03010
(getsdb) (nls,filmp, getsdb) 4C1	03027
(getsid) (nls,filmp, getsid) 4B5	03021
(getstsize) (nls,filmp, getstsize) 4F9	03050
(getsub) (nls,filmp, getsub) 4A2	03012
(getsuc) (nls,filmp, getsuc) 4A1	02989
(gettim) (nls,filmp, gettim) 4F7	02978
(getup) (nls,filmp, getup) 4A4	03014
(getvnd) (nls,filmp, getvnd) 4A8	03042
(goodrng) (nls,filmp, goodrng) 6E4	02983
(grptst) (nls,filmp, grptst) 7E1	03018
(initdb) (nls,filmp, initdb) 6F2	03038
(insd) (nls,filmp, insd) 6B5	03031
(insgrp) (nls,filmp, insgrp) 6B3	02969

(insitree) (nls,filmp,insitree) 6A3	02980
(inss) (nls,filmp,inss) 6B4	02985
(levset) (nls,filmp,levset) 6B7	03055
(lkfile) (nls,filmp,lkfile) 8B3	02998
(lkun) (nls,filmp,lkun) 8B1	03036
(lnkprop) (nls,filmp,lnkprop) 6C3	02970
(lockfile) (nls,filmp,lockfile) 8B2	03009
(lockid) (nls,filmp,lockid) 8B4	03043
(locprop) (nls,filmp,locprop) 6C1	03023
(lodent) (nls,filmp,lodent) 5A1	02986
(lodprop) (nls,filmp,lodprop) 5A2	03019
(lodrfb) (nls,filmp,lodrfb) 5A3	03013
(makepc) (nls,filmp,makepc) 8C2	03033
(movit) (nls,filmp,movit) 5E3	03008
(movprop) (nls,filmp,movprop) 5D3	02987
(mvdifgrp) (nls,filmp,mvdifgrp) 5B4	02984
(newdb) (nls,filmp,newdb) 6F1	02994
(newrng) (nls,filmp,newrng) 6E1	03037
(newsb) (nls,filmp,newsb) 6B2	03059
(newsuc) (nls,filmp,newsuc) 6B1	02996
(nwrngb) (nls,filmp,nwrngb) 6E2	02968
(remgrp) (nls,filmp,remgrp) 5B5	02963
(reprop) (nls,filmp,reprop) 5D5	03017
(rlevset) (nls,filmp,rlevset) 6B6	03002
(stofhd) (nls,filmp,stofhd) 4G4	02981
(stoftl) (nls,filmp,stoftl) 4G3	03006
(stoitree) (nls,filmp,stoitree) 4G10	02977
(stonam) (nls,filmp,stonam) 4G7	03004
(stonmf) (nls,filmp,stonmf) 4G5	02999
(stoorg) (nls,filmp,stoorg) 4G9	03045
(stosdb) (nls,filmp,stosdb) 4G6	02990
(stosid) (nls,filmp,stosid) 4G8	03022
(stosub) (nls,filmp,stosub) 4G2	02975
(stosuc) (nls,filmp,stosuc) 4G1	03041
(trecop) (nls,filmp,trecop) 6D1	02976
(trnsint) (nls,filmp,trnsint) 4F6	02979
(unlnkprop) (nls,filmp,unlnkprop) 6C4	03001
(upctbl) (nls,filmp,upctbl) 7C1	02966
(wpiabt) (nls,filmp,wpiabt) 8A3	03051
(wrpi) (nls,filmp,wrpi) 8A2	03029
(wrtpsi) (nls,filmp,wrtpsi) 8A1	02965
	03889

(a1) (nls,fintnls,)	0526
(a4) (nls,fintnls,)	0509
(r5) (nls,fintnls,)	0506
(rp) (nls,fintnls,)	0527
(checkdevline) (nls,fintnls,checkdevline) 7D	0513
(ckdvin) (nls,fintnls,ckdvlm) 7C	0532
(condevline) (nls,fintnls,condevline) 7E	0523
(dnls) (nls,fintnls,dnls) 4A	0529
(frentr) (nls,fintnls,frentr) 5B	0499
(getdev) (nls,fintnls,getdev) 7F	0502
(getuidnt) (nls,fintnls,getuidnt) 7Q	0501
(init) (nls,fintnls,init) 5A4	0507
(initcn) (nls,fintnls,initch) 7J	0517
(initda) (nls,fintnls,initda) 7N	0503
(initdis) (nls,fintnls,initdis) 7K	0515
(initnls) (nls,fintnls,initnls) 5A	0519
(intdafl) (nls,fintnls,intdafl) 7B	0521
(intdspda) (nls,fintnls,intdspda) 7H	0528
(loderrm) (nls,fintnls,loderrm) 6C	0510
(newintdspda) (nls,fintnls,newintdspda) 7I	0530
(nls) (nls,fintnls,nls) 5A5	0524
(nlsave) (nls,fintnls,nlsave) 6D	0520
(qt) (nls,fintnls,qt) 6B2	0500
(query) (nls,fintnls,query) 4C	0531
(rentr) (nls,fintnls,rentr) 5B2	0504
(saveit) (nls,fintnls,saveit) 6B	0525
(setcinit) (nls,fintnls,setcinit) 7P	0518
(setdev) (nls,fintnls,setdev) 7G	0511
(setdis) (nls,fintnls,setdis) 7L	0514
(setvec) (nls,fintnls,setvec) 7A	0512
(st) (nls,fintnls,st) 6B1	0516
(start) (nls,fintnls,start) 5A6	0498
(startup) (nls,fintnls,startup) 5C	0505
(t2fnsupdate) (nls,fintnls,t2fnsupdate) 6E	0533
(tnls) (nls,fintnls,tnls) 4B	0508
(ttywindow) (nls,fintnls,ttywindow) 7M	0522

03890

(jrinit) (nls, fonly, jrinit) 4B2	0488
(xaccept) (nls, fonly, xaccept) 3C1	0497
(xclear) (nls, fonly, xclear) 3A1	0489
(xcomment) (nls, fonly, xcomment) 3H1	0496
(xconnect) (nls, fonly, xconnect) 3B1	0495
(xconterm) (nls, fonly, xconterm) 3B2	0485
(xdisconnect) (nls, fonly, xdisconnect) 3D1	0492
(xfreeze) (nls, fonly, xfreeze) 3E1	0486
(xjdae) (nls, fonly, xjdae) 4A1	0494
(xjpcntdisp) (nls, fonly, xjpcntdisp) 4B3	0491
(xrelease) (nls, fonly, xrelease) 3F1	0490
(xringjump) (nls, fonly, xringjump) 4B1	0493
(xsimulate) (nls, fonly, xsimulate) 3G1	0487
	03891

(alword) (nls,records,alword) 201A	02055
(a2word) (nls,records,a2word) 201E	02036
(a3word) (nls,records,a3word) 201C	02046
(a4word) (nls,records,a4word) 201D	02163
(adadcs) (nls,records,adadcs) 3B4	02120
(adadf) (nls,records,adadf) 3B6	02065
(adadhi) (nls,records,adadhi) 3B5	02099
(adalrx) (nls,records,adalrx) 3B3	02074
(adalry) (nls,records,adalry) 3B2	02100
(adaseq) (nls,records,adaseq) 3A5	02186
(adasize) (nls,records,adasize) 3A4	02144
(adaulx) (nls,records,adaulx) 3A3	02038
(adauly) (nls,records,adauly) 3A2	02173
(addr) (nls,records,addr) 2M1C	02185
(alternative) (nls,records,alternative) 2M1A	02172
(argcount) (nls,records,argcount) 2P1E	02160
(begnodeptr) (nls,records,begnodeptr) 2P1B	02121
(ccchar) (nls,records,ccchar) 3C3	02081
(ccdevice) (nls,records,ccdevice) 3C1	02131
(ccecho) (nls,records,ccecho) 3C4	02162
(cctype) (nls,records,cctype) 3C2	02017
(cfl1len) (nls,records,cfl1len) 2I1D	02158
(cfl2len) (nls,records,cfl2len) 2I1E	02157
(conlnk) (nls,records,conlnk) 5B1	02049
(constd) (nls,records,constd) 5A1	02063
(constk) (nls,records,constk) 5A2	02062
(ctrl) (nls,records,ctrl) 2M1E	02167
(curnodeptr) (nls,records,curnodeptr) 2P1C	02143
(currcm) (nls,records,currcm) 2J1A	02135
(currinst) (nls,records,currinst) 2J1F	02112
(currpath) (nls,records,currpath) 2J1G	02058
(daidr1) (nls,records,daidr1) 3E2	02048
(daidr2) (nls,records,daidr2) 3E1	02110
(dnlscmd) (nls,records,dnlscmd) 2L1B	02192
(dolitreset) (nls,records,dolitreset) 2I1I	02165
(dptfinish) (nls,records,dptfinish) 2B1E	02115
(dptinit) (nls,records,dptinit) 2B1D	02156
(dptname) (nls,records,dptname) 2B1A	02056
(dptnotused) (nls,records,dptnotused) 2B1F	02088
(dptreentry) (nls,records,dptreentry) 2B1G	02091
(dptrun) (nls,records,dptrun) 2B1E	02168
(dptvalid) (nls,records,dptvalid) 2B1C	02085
(endopt) (nls,records,endopt) 2J1C	02083
(entype) (nls,records,entype) 2I1A	02178
(evalmod) (nls,records,evalmod) 2P1I	02183
(evalxsav) (nls,records,evalxsav) 2P1H	02134
(fbh1p1) (nls,records,fbh1p1) 2A1D1A	02113
(fblen) (nls,records,fblen) 2K1D	02130
(fbstrptr) (nls,records,fbstrptr) 2N1C	02082
(fcounter) (nls,records,fcounter) 2P1J	02051
(fdd) (nls,records,fdd) 3F6	02057
(fds) (nls,records,fds) 3F7	02042
(firstinst) (nls,records,firstinst) 2J1E	02145
(firstx) (nls,records,firstx) 2J1J	02026
(fmodcol) (nls,records,fmodcol) 3N2	02041
(fmodright) (nls,records,fmodright) 3N1	02066

(frzl) (nls,records,frzl) 3G6A	02020
(fzaxis) (nls,records,fzaxis) 3G5	02142
(fznext) (nls,records,fznext) 3G4	02107
(fzstid) (nls,records,fzstid) 3G3	02102
(fzvspc2) (nls,records,fzvspc2) 3G2	02187
(fzvspec) (nls,records,fzvspec) 3G1	02105
(grdtyp) (nls,records,grdtyp) 3M2	02071
(hbastr) (nls,records,hbastr) 2A1D	02079
(hbbck) (nls,records,hbbck) 2A1B	02022
(hbfor) (nls,records,hbfor) 2A1A	02137
(hbval) (nls,records,hbval) 2A1C	02119
(ibottom) (nls,records,ibottom) 1A4	02180
(icsize) (nls,records,icsize) 1A5	02128
(idd) (nls,records,idd) 3F4	02060
(ids) (nls,records,ids) 3F5	02188
(ifont) (nls,records,ifont) 1A6	02068
(ihinc) (nls,records,ihinc) 1A7	02015
(ileft) (nls,records,ileft) 1A1	02098
(inpsav) (nls,records,inpsav) 2K1C	02092
(insptr) (nls,records,insptr) 2N1B	02024
(iright) (nls,records,iright) 1A2	02126
(itop) (nls,records,itop) 1A3	02044
(ivinc) (nls,records,ivinc) 1A8	02040
(keyinlength) (nls,records,keyinlength) 2K1E	02125
(ksaveflag) (nls,records,ksaveflag) 2K1F	02072
(l1cmd) (nls,records,l1cmd) 2L1C	02070
(litptr) (nls,records,litptr) 2I1H	02014
(markptr) (nls,records,markptr) 2P1D	02034
(maxselect) (nls,records,maxselect) 2I1G	02177
(mnstd) (nls,records,mnstd) 5C1	02153
(msffno) (nls,records,msffno) 4A4	02159
(msfltime) (nls,records,msfltime) 4A1	02171
(msfmflag) (nls,records,msfmflag) 4A2	02090
(msfmtyp) (nls,records,msfmtyp) 4A3	02086
(msrmod) (nls,records,msrmod) 4C2	02077
(msrr0) (nls,records,msrr0) 4B1	02078
(msrr1) (nls,records,msrr1) 4B2	02039
(msrr2) (nls,records,msrr2) 4B3	02018
(msrr3) (nls,records,msrr3) 4B4	02181
(msrtl) (nls,records,msrtl) 4C1	02064
(nextx) (nls,records,nextx) 2J1K	02161
(notused) (nls,records,notused) 3M3	02093
(nselects) (nls,records,nselects) 2I1F	02033
(nsuccessor) (nls,records,nsuccessor) 2M1B	02031
(opcode) (nls,records,opcode) 2M1F	02117
(optcount) (nls,records,optcount) 2J1B	02032
(optflag) (nls,records,optflag) 2J1D	02189
(pcomcode) (nls,records,pcomcode) 2F1A	02069
(pfunction) (nls,records,pfunction) 2P1A	02097
(pkstrptr) (nls,records,pkstrptr) 2G1A	02114
(plvcount) (nls,records,plvcount) 2D1A	02025
(pmode) (nls,records,pmode) 2P1F	02150
(praddr) (nls,records,praddr) 3I3	02155
(prexists) (nls,records,prexists) 3I4	02067
(propcode) (nls,records,propcode) 2N1A	02136
(prwrd1) (nls,records,prwrd1) 3I1	02080

(prwrd2) (nls,frecords,prwrd2) 3I2	02124
(ptrxsav) (nls,frecords,ptrxsav) 2P1G	02087
(quercol) (nls,frecords,quercol) 5D2	02101
(querinc) (nls,frecords,querinc) 5D1	02139
(queropts) (nls,frecords,queropts) 5D4	02170
(querprt) (nls,frecords,querprt) 5D3	02053
(retcc) (nls,frecords,retcc) 3H3	02166
(retfn) (nls,frecords,retfn) 3H2	02106
(retrc1) (nls,frecords,retrc1) 2K1A1	02104
(retrc2) (nls,frecords,retrc2) 2K1A2	02151
(retrc3) (nls,frecords,retrc3) 2K1A3	02050
(retrc4) (nls,frecords,retrc4) 2K1A4	02108
(retsrr) (nls,frecords,retsrr) 3H6	02118
(retstid) (nls,frecords,retstid) 3H1	02191
(retval) (nls,frecords,retval) 2K1B	02103
(retvs1) (nls,frecords,retvs1) 3H4	02122
(retvs2) (nls,frecords,retvs2) 3H5	02182
(savex) (nls,frecords,savex) 2J1I	02147
(savstk) (nls,frecords,savstk) 2J1H	02023
(sbcount) (nls,frecords,sbcount) 2H1C	02019
(sbmode) (nls,frecords,sbmode) 2H1D	02043
(sbnptr) (nls,frecords,sbnptr) 2H1B	02073
(sbpmode) (nls,frecords,sbpmode) 2H1E	02096
(sbptr) (nls,frecords,sbptr) 2H1A	02016
(sccsiz) (nls,frecords,sccsiz) 3J1	02133
(scfont) (nls,frecords,scfont) 3J3	02037
(schinc) (nls,frecords,schinc) 3J2	02116
(scsize) (nls,frecords,scsize) 3L3	02127
(sdaid) (nls,frecords,sdaid) 3K2	02035
(sdd) (nls,frecords,sdd) 3F2	02021
(sds) (nls,frecords,sds) 3F3	02190
(sellfun) (nls,frecords,sellfun) 2I1B	02111
(sel2fun) (nls,frecords,sel2fun) 2I1C	02095
(sfont) (nls,frecords,sfont) 3L5	02179
(shinc) (nls,frecords,shinc) 3L4	02045
(sltp11) (nls,frecords,sltp11) 2E1A	02176
(sltp12) (nls,frecords,sltp12) 2E1B	02084
(sltp21) (nls,frecords,sltp21) 2E1C	02059
(sltp22) (nls,frecords,sltp22) 2E1D	02123
(sstrid) (nls,frecords,sstrid) 3K1	02076
(stkcnt) (nls,frecords,stkcnt) 5B2	02061
(stkhsh) (nls,frecords,stkhsh) 5C2	02089
(stktop) (nls,frecords,stktop) 5A4	02028
(sxcord) (nls,frecords,sxcord) 3L2	02027
(sycord) (nls,frecords,sycord) 3L1	02141
(t1word) (nls,frecords,t1word) 2O1E	02146
(t2word) (nls,frecords,t2word) 2O1G	02129
(t3word) (nls,frecords,t3word) 2O1H	02138
(t4word) (nls,frecords,t4word) 2O1I	02109
(t5word) (nls,frecords,t5word) 2O1J	02132
(tnlscmd) (nls,frecords,tnlscmd) 2L1A	02164
(totcnt) (nls,frecords,totcnt) 5A3	02047
(trmtyp) (nls,frecords,trmtyp) 3M1	02154
(ttylno) (nls,frecords,ttylno) 3M4	02030
(usrmode) (nls,frecords,usrmode) 2J1L	02029
(val2) (nls,frecords,val2) 2M1D	02094

(vs1) (nls,frecords,vs1) 2C1A		02175
(vs2) (nls,frecords,vs2) 2C1B		02174
(vscacode) (nls,frecords,vscacode) 2C1C		02148
(vsusqcod) (nls,frecords,vsusqcod) 2C1D		02169
(xc1) (nls,frecords,xc1) 3D2		02054
(xc2) (nls,frecords,xc2) 3D1		02052
(xcord) (nls,frecords,xcord) 3D1		02149
(xyds) (nls,frecords,xyds) 3F1		02152
(yc1) (nls,frecords,yc1) 3P2		02184
(yc2) (nls,frecords,yc2) 3P1		02075
(ycord) (nls,frecords,ycord) 3D2		02140
		03892

(bbound) (nls,frontend,)	03118
(ninc0) (nls,frontend,)	03109
(ninc1) (nls,frontend,)	03102
(ninc2) (nls,frontend,)	03097
(ninc3) (nls,frontend,)	03090
(lbound) (nls,frontend,)	03136
(rbound) (nls,frontend,)	03146
(rtlast) (nls,frontend,)	03147
(tbound) (nls,frontend,)	03111
(vinc0) (nls,frontend,)	03141
(vinc1) (nls,frontend,)	03131
(vinc2) (nls,frontend,)	03094
(vinc3) (nls,frontend,)	03126
(alldsp) (nls,frontend,alldsp) 5B	03091
(auxchr) (nls,frontend,auxchr) 10B	03121
(auxinterminate) (nls,frontend,auxinterminate) 10C	03140
(auxstartup) (nls,frontend,auxstartup) 10A	03100
(boundry) (nls,frontend,boundry) 11C	03148
(copych) (nls,frontend,copych) 13C	03145
(copyda) (nls,frontend,copyda) 11P	03150
(copysrring) (nls,frontend,copysrring) 12J	03142
(ctlquit) (nls,frontend,ctlquit) 9A	03137
(curvsp) (nls,frontend,curvsp) 6A	03143
(delda) (nls,frontend,delda) 11N	03135
(dsparea) (nls,frontend,dsparea) 11M	03123
(dstart) (nls,frontend,dstart) 14A	03103
(editx) (nls,frontend,editx) 13A	03120
(edixqt) (nls,frontend,edixqt) 13B	03105
(filusd) (nls,frontend,filusd) 3B	03138
(findda) (nls,frontend,findda) 11L	03085
(fixbnd) (nls,frontend,fixbnd) 11E	03086
(fixbuf) (nls,frontend,fixbuf) 11F	03089
(freefrring) (nls,frontend,freefrring) 12B	03127
(freefrring) (nls,frontend,freefrring) 12I	03144
(freflnt) (nls,frontend,freflnt) 3A	03112
(frrlength) (nls,frontend,frrlength) 12E	03122
(frzchk) (nls,frontend,frzchk) 5E	03119
(howformat) (nls,frontend,howformat) 5D	03110
(hsplit) (nls,frontend,hsplit) 11B	03106
(lccords) (nls,frontend,lccords) 11K	03124
(lccsp) (nls,frontend,lccsp) 11H	03107
(lcda) (nls,frontend,lcda) 11I	03096
(lcfile) (nls,frontend,lcfile) 11G	03125
(lda) (nls,frontend,lda) 11J	03104
(mesfre) (nls,frontend,mesfre) 4A	03149
(mkdelfrr) (nls,frontend,mkdelfrr) 12G	03139
(movbndry) (nls,frontend,movbndry) 11D	03093
(newda) (nls,frontend,newda) 11O	03108
(newfrring) (nls,frontend,newfrring) 12A	03088
(newsrring) (nls,frontend,newsrring) 12H	03095
(pushfrring) (nls,frontend,pushfrring) 12C	03087
(pushsrring) (nls,frontend,pushsrring) 12K	03115
(readfrring) (nls,frontend,readfrring) 12D	03129
(readsrring) (nls,frontend,readsrring) 12M	03092
(recred) (nls,frontend,recred) 5A	03134
(search) (nls,frontend,search) 13D	03151

(seldsp) (nls,frontend,seldsp) 5C	03130
(skip) (nls,frontend,skip) 13E	03098
(srlength) (nls,frontend,srlength) 12N	03116
(storesrring) (nls,frontend,storesrring) 12L	03114
(tstart) (nls,frontend,tstart) 14B	03099
(uplsinm) (nls,frontend,uplsfnm) 12F	03117
(vsplit) (nls,frontend,vsplit) 11A	03133
(xfresta) (nls,frontend,xfresta) 7C	03101
(xrecplasup) (nls,frontend,xrecplasup) 9B	03128
(xrelallsta) (nls,frontend,xrelallsta) 7B	03113
(xrelsta) (nls,frontend,xrelsta) 7A	03132
(xsimdev) (nls,frontend,xsimdev) 8A	03152
	03893

(aigpty) (nls,identsupport,)		03518
(idchr) (nls,identsupport,)		03512
(iname) (nls,identsupport,)		03521
(nullfi) (nls,identsupport,)		03494
(nwidst) (nls,identsupport,)		03492
(strinn) (nls,identsupport,)		03466
(ckident) (nls,identsupport,ckident)	4B	03473
(ckidlist) (nls,identsupport,ckidlist)	4C	03469
(ckiimem) (nls,identsupport,ckilmem)	11I	03525
(ckipmem) (nls,identsupport,ckipmem)	11K	03490
(cknlsid) (nls,identsupport,cknlsid)	4A	03475
(clnidlst) (nls,identsupport,clnidlst)	11L	03510
(delcmidents) (nls,identsupport,delcmidents)	11M	03526
(expdtst) (nls,identsupport,expdtst)	7C	03486
(gbotids) (nls,identsupport,gbotids)	10D	03524
(getgpids) (nls,identsupport,getgpids)	11A	03495
(getiadd) (nls,identsupport,getiadd)	6B	03516
(geticapability) (nls,identsupport,geticapability)	6V	03483
(geticord) (nls,identsupport,geticord)	6J	03498
(getidelivery) (nls,identsupport,getidelivery)	6X	03496
(getids) (nls,identsupport,getids)	9A	03500
(getiend) (nls,identsupport,getiend)	7A	03517
(getiexp) (nls,identsupport,getiexp)	6G	03513
(getifnf) (nls,identsupport,getifnf)	6D	03467
(getifunction) (nls,identsupport,getifunction)	6T	03471
(getigrps) (nls,identsupport,getigrps)	6N	03514
(getihost) (nls,identsupport,getihost)	6Q	03505
(getiid) (nls,identsupport,getiid)	6A	03465
(getilname) (nls,identsupport,getilname)	6E	03474
(getimcmnts) (nls,identsupport,getimcmnts)	6Y	03488
(getimem) (nls,identsupport,getimem)	6H	03493
(getinam) (nls,identsupport,getinam)	6C	03480
(getinlhost) (nls,identsupport,getinlhost)	6R	03506
(getinma) (nls,identsupport,getinma)	6S	03504
(getiorg) (nls,identsupport,getiorg)	6F	03479
(getiphone) (nls,identsupport,getiphone)	6P	03468
(getiprevmem) (nls,identsupport,getiprevmem)	6I	03472
(getisorg) (nls,identsupport,getisorg)	6U	03489
(getisubcol) (nls,identsupport,getisubcol)	6W	03515
(getityp) (nls,identsupport,getityp)	6K	03529
(getiuser) (nls,identsupport,getiuser)	6M	03527
(getiverify) (nls,identsupport,getiverify)	6L	03522
(getpointer) (nls,identsupport,getpointer)	6D	03485
(idflsearch) (nls,identsupport,idflsearch)	11G	03508
(idpgrporg) (nls,identsupport,idpgrporg)	11F	03509
(idpind) (nls,identsupport,idpind)	11E	03502
(idplname) (nls,identsupport,idplname)	11D	03519
(intids) (nls,identsupport,intids)	10A	03478
(jgrptst) (nls,identsupport,jgrptst)	8A	03477
(laddress) (nls,identsupport,laddress)	5D	03523
(ldelivery) (nls,identsupport,ldelivery)	5A	03528
(lgetsubcoll) (nls,identsupport,lgetsubcoll)	5C	03507
(lmemlist) (nls,identsupport,lmemlist)	5E	03501
(loadfil) (nls,identsupport,loadfil)	3A	03487
(luser) (nls,identsupport,luser)	5B	03497
(makgid) (nls,identsupport,makgid)	11H	03503

(namesearch) (nls,identsupport,namesearch) 11C	03491
(ognxt) (nls,identsupport,ognxt) 11J	03520
(oldid) (nls,identsupport,oldid) 4D	03499
(orgrptst) (nls,identsupport,orgrptst) 8C	03484
(orgtst) (nls,identsupport,orgtst) 8B	03470
(popids) (nls,identsupport,popids) 10B	03481
(pushids) (nls,identsupport,pushids) 10C	03482
(stnamcap) (nls,identsupport,stnamcap) 11B	03476
(stptset) (nls,identsupport,stptset) 7B	03511
	03894

(dinchr) (nls,inpfbk,)	03846
(mslww) (nls,inpfbk,)	03797
(msmrkr) (nls,inpfbk,)	03776
(msupvw) (nls,inpfbk,)	03801
(af) (nls,inpfbk,af) 9G2B	03789
(an) (nls,inpfbk,an) 9G2A	03844
(aplit) (nls,inpfbk,aplit) 9H5	03796
(arm) (nls,inpfbk,arm) 9C1	03829
(bmooff) (nls,inpfbk,bmooff) 9A2	03814
(cfldsp) (nls,inpfbk,cfldsp) 9G1F	03816
(clearlit) (nls,inpfbk,clearlit) 9H2	03837
(clrseqda) (nls,inpfbk,clrseqda) 9H3	03777
(cntrlgetchar) (nls,inpfbk,cntrlgetchar) 10A	03832
(csndtms) (nls,inpfbk,csndtms) 7B	03841
(csrstr) (nls,inpfbk,csrstr) 9C3	03778
(cvsno) (nls,inpfbk,cvsno) 10G	03835
(delbm) (nls,inpfbk,delbm) 9A3	03775
(dendvw) (nls,inpfbk,dendvw) 4C	03831
(dinbc) (nls,inpfbk,dinbc) 4B	03848
(dinptc) (nls,inpfbk,dinptc) 4A	03785
(disarm) (nls,inpfbk,disarm) 9C2	03787
(dn) (nls,inpfbk,dn) 9E1	03790
(dspvsp) (nls,inpfbk,dspvsp) 9D1	03803
(dsubsys) (nls,inpfbk,dsubsys) 9F1	03851
(echo) (nls,inpfbk,echo) 6C	03809
(echoff) (nls,inpfbk,echoff) 8B	03842
(echoimed) (nls,inpfbk,echoimed) 8E	03850
(echoit) (nls,inpfbk,echoit) 8D	03828
(echon) (nls,inpfbk,echon) 8A	03800
(errfill) (nls,inpfbk,errfill) 3G	03782
(fndchr) (nls,inpfbk,fndchr) 9B2	03827
(fndtab) (nls,inpfbk,fndtab) 11A	03852
(formln) (nls,inpfbk,formln) 5E	03793
(getchar) (nls,inpfbk,getchar) 3D	03836
(gtchwait) (nls,inpfbk,gtchwait) 3D8	03838
(initbtbl) (nls,inpfbk,initbtbl) 10B	03820
(inpcuc) (nls,inpfbk,inpcuc) 3A	03819
(input) (nls,inpfbk,input) 3B	03807
(inpvsp) (nls,inpfbk,inpvsp) 10C	03781
(isechon) (nls,inpfbk,isechon) 8C	03817
(linebreak) (nls,inpfbk,linebreak) 9H8	03839
(litbc) (nls,inpfbk,litbc) 5H3	03806
(litchr) (nls,inpfbk,litchr) 5H2	03795
(litdpy) (nls,inpfbk,litdpy) 9H6	03810
(litpbout) (nls,inpfbk,litpbout) 5H1	03826
(litsup) (nls,inpfbk,litsup) 9H9	03822
(littty) (nls,inpfbk,littty) 9H7	03786
(lkmkr) (nls,inpfbk,lkmkr) 10D	03825
(lookc) (nls,inpfbk,lookc) 3C	03840
(lpaltgetchar) (nls,inpfbk,lpaltgetchar) 3E	03798
(lpgetchar) (nls,inpfbk,lpgetchar) 3F	03788
(ltl) (nls,inpfbk,ltl) 9D2	03794
(lts) (nls,inpfbk,lts) 9D3	03791
(markit) (nls,inpfbk,markit) 9A1	03799
(mlf) (nls,inpfbk,mlf) 9G1D	03847
(mlf1) (nls,inpfbk,mlf1) 9G1C	03818

(mlfr) (nls,inpfbk,mlfr) 9G1E	03805
(mrk) (nls,inpfbk,mrk) 9G1B	03784
(mrk1) (nls,inpfbk,mrk1) 9G1A	03833
(outerrec) (nls,inpfbk,outerrec) 3H	03812
(pbug) (nls,inpfbk,pbug) 9B1	03808
(prompt) (nls,inpfbk,prompt) 6D	03813
(qm) (nls,inpfbk,qm) 9G3A	03823
(qloff) (nls,inpfbk,qloff) 9G3B	03815
(rdlit) (nls,inpfbk,rdlit) 5C	03849
(rdlutl) (nls,inpfbk,rdlutl) 5D	03804
(resetb) (nls,inpfbk,resetb) 10F	03843
(rstlit) (nls,inpfbk,rstlit) 9H1	03845
(setlit) (nls,inpfbk,setlit) 9H4	03774
(specttyout) (nls,inpfbk,specttyout) 10H	03821
(sprjtb) (nls,inpfbk,sprjtb) 5G	03780
(sptbbc) (nls,inpfbk,sptbbc) 5F	03834
(tinptc) (nls,inpfbk,tinptc) 5A	03802
(todco) (nls,inpfbk,todco) 6A	03783
(txtlit) (nls,inpfbk,txtlit) 5B	03779
(typebuff) (nls,inpfbk,typebuff) 6E	03811
(typectl) (nls,inpfbk,typectl) 6B	03792
(unput) (nls,inpfbk,unput) 10E	03830
(xsndtmsg) (nls,inpfbk,xsndtmsg) 7A	03824
	03895

(addfil) (nls,ioexec,addfil) 9A	03270
(checkjfile) (nls,ioexec,checkjfile) 15L	03216
(chnfdb) (nls,ioexec,chnfdb) 13E	03264
(chprvsts) (nls,ioexec,chprvsts) 14A	03257
(cldtbs) (nls,ioexec,cldtbs) 10C	03186
(clflup) (nls,ioexec,clflup) 11D	03215
(close) (nls,ioexec,close) 5A	03223
(closeall) (nls,ioexec,closeall) 5D	03236
(closepc) (nls,ioexec,closepc) 5F	03282
(closeu) (nls,ioexec,closeu) 5B	03250
(clpsid) (nls,ioexec,clpsid) 11C	03242
(clrcor) (nls,ioexec,clrcor) 10D	03272
(clrtfil) (nls,ioexec,clrtfil) 10A	03229
(clrnrgs) (nls,ioexec,clrnrgs) 10B	03239
(clstfil) (nls,ioexec,clstfil) 5E	03196
(clsid) (nls,ioexec,clsid) 11B	03241
(cnvfndocnum) (nls,ioexec,cnvfndocnum) 15M	03266
(condir) (nls,ioexec,condir) 17E	03184
(conidir) (nls,ioexec,conidir) 17D	03193
(conjdir) (nls,ioexec,conjdir) 17C	03274
(coropnfil) (nls,ioexec,coropnfil) 4I	03276
(cpcnam) (nls,ioexec,cpcnam) 12G	03248
(cpyfle) (nls,ioexec,cpyfle) 9C	03191
(delfil) (nls,ioexec,delfil) 9B	03206
(delovsrns) (nls,ioexec,delovsrns) 15D	03230
(delpc) (nls,ioexec,delpc) 12E	03255
(disabaccess) (nls,ioexec,disabaccess) 13C	03279
(dumpc) (nls,ioexec,dumpc) 12D	03240
(enablaccess) (nls,ioexec,enablaccess) 13B	03188
(fetchfile) (nls,ioexec,fetchfile) 16B1	03271
(filnam) (nls,ioexec,filnam) 9E	03260
(filnum) (nls,ioexec,filnum) 9F	03254
(fldirectory) (nls,ioexec, fldirectory) 9G	03283
(flntadr) (nls,ioexec,flntadr) 9D	03228
(ftpapp) (nls,ioexec,ftpapp) 16C15	03212
(ftpbgn) (nls,ioexec,ftpbgn) 16C1	03232
(ftpfcf) (nls,ioexec,ftpfcf) 16E3	03214
(ftpcls) (nls,ioexec,ftpcls) 16C4	03192
(ftpert) (nls,ioexec,ftpert) 16E1	03218
(ftpdel) (nls,ioexec,ftpdel) 16C17	03281
(ftpdir) (nls,ioexec,ftpdir) 16C16	03231
(ftpdml) (nls,ioexec,ftpdml) 16C6	03268
(ftpelim) (nls,ioexec,ftpelim) 16C11	03288
(ftpend) (nls,ioexec,ftpend) 16C2	03244
(ftperr) (nls,ioexec,ftperr) 16F1	03213
(ftpex) (nls,ioexec,ftpex) 16D1	03208
(ftpfml) (nls,ioexec,ftpfml) 16C9	03185
(ftpgts) (nls,ioexec,ftpgts) 16F2	03209
(ftpkil) (nls,ioexec,ftpkil) 16E2	03251
(ftplml) (nls,ioexec,ftplml) 16C7	03222
(ftplog) (nls,ioexec,ftplog) 16C5	03237
(ftpogn) (nls,ioexec,ftpogn) 16C3	03202
(ftppts) (nls,ioexec,ftppts) 16F3	03233
(ftpqml) (nls,ioexec,ftpqml) 16C8	03293
(ftpren) (nls,ioexec,ftpren) 16C20	03197
(ftppren) (nls,ioexec,ftppren) 16C18	03227

(ftpron)	(nls,ioexec,ftpron)	16C21	03219
(ftprtr)	(nls,ioexec,ftprtr)	16C14	03211
(ftpsdd)	(nls,ioexec,ftpsdd)	16C10	03198
(ftpsem)	(nls,ioexec,ftpsem)	16C19	03252
(ftpsto)	(nls,ioexec,ftpsto)	16C13	03187
(ftpvhst)	(nls,ioexec,ftpvhst)	16C12	03263
(gdftdir)	(nls,ioexec,gdftdir)	15K	03220
(gdmys)	(nls,ioexec,gdmys)	7C1	03289
(gdname)	(nls,ioexec,gdname)	15J	03277
(getenf)	(nls,ioexec,getenf)	9I	03194
(getfacc)	(nls,ioexec,getfacc)	14D	03217
(getgtjflg)	(nls,ioexec,getgtjflg)	8G	03207
(getversn)	(nls,ioexec,getversn)	15E	03262
(gtjerr)	(nls,ioexec,gtjerr)	8F	03278
(incrversn)	(nls,ioexec,incrversn)	15I	03261
(intfil)	(nls,ioexec,intfil)	7A	03269
(iszeropage)	(nls,ioexec,iszeropage)	15C	03259
(jclosfl)	(nls,ioexec,jclosfl)	17A	03224
(jflname)	(nls,ioexec,jflname)	17B	03189
(jfnlink)	(nls,ioexec,jfnlink)	15H	03235
(jfnstr)	(nls,ioexec,jfnstr)	15F	03284
(jfnstr)	(nls,ioexec,jfnstr)	15G	03226
(lgetjfn)	(nls,ioexec,lgetjfn)	8C	03238
(lockmes)	(nls,ioexec,lockmes)	12A	03245
(maywrt)	(nls,ioexec,maywrt)	13D	03234
(open)	(nls,ioexec,open)	4A	03273
(openall)	(nls,ioexec,openall)	4G	03203
(openid)	(nls,ioexec,openid)	6A	03287
(openlock)	(nls,ioexec,openlock)	4E	03275
(openpc)	(nls,ioexec,openpc)	4K	03256
(openull)	(nls,ioexec,openull)	6B	03195
(openwk)	(nls,ioexec,openwk)	4B	03267
(oplockfail)	(nls,ioexec,oplockfail)	4E8C1E3C3	03205
(opnctfile)	(nls,ioexec,opnctfile)	4F	03246
(opnfil)	(nls,ioexec,opnfil)	4H	03199
(opwk)	(nls,ioexec,opwk)	4D	03243
(opwknr)	(nls,ioexec,opwknr)	4C	03225
(rawopen)	(nls,ioexec,rawopen)	4J	03291
(rdhdr)	(nls,ioexec,rdhdr)	15B	03247
(rdprvsts)	(nls,ioexec,rdprvsts)	14B	03249
(reljfn)	(nls,ioexec,reljfn)	8E	03204
(resetf)	(nls,ioexec,resetf)	6C	03280
(setaccess)	(nls,ioexec,setaccess)	13A	03285
(setfacc)	(nls,ioexec,setfacc)	14E	03221
(setfdb)	(nls,ioexec,setfdb)	12C	03290
(setfil)	(nls,ioexec,setfil)	7B	03200
(sgtjfn)	(nls,ioexec,sgtjfn)	8D	03253
(sigclose)	(nls,ioexec,sigclose)	5C	03292
(sysclose)	(nls,ioexec,sysclose)	8B	03286
(sysopen)	(nls,ioexec,sysopen)	8A	03210
(transdir)	(nls,ioexec,transdir)	9H	03201
(unlkclist)	(nls,ioexec,unlkclist)	11A	03183
(unlkfile)	(nls,ioexec,unlkfile)	12F	03190
(vrprvacc)	(nls,ioexec,vrprvacc)	14C	03258
(writeout)	(nls,ioexec,writeout)	15A	03265

(b1) (nls,l10data,b1) 8F1C	01374
(b2) (nls,l10data,b2) 8F1D	01383
(b3) (nls,l10data,b3) 8F1E	01373
(b4) (nls,l10data,b4) 8F1F	01388
(b5) (nls,l10data,b5) 8F1G	01372
(btwstk) (nls,l10data,btwstk) 8E1	01363
(cinit) (nls,l10data,cinit) 5B1	01377
(clchnng) (nls,l10data,clchnng) 6A1	01389
(clhead) (nls,l10data,clhead) 8G1A	01368
(clistb) (nls,l10data,clistb) 8G1B	01380
(clistsz) (nls,l10data,clistsz) 8G1C	01361
(clmpty) (nls,l10data,clmpty) 5F4	01360
(clstad) (nls,l10data,clstad) 8G1D	01357
(flag) (nls,l10data,flag) 5C1	01386
(gstack) (nls,l10data,gstack) 7A1	01370
(lngflg) (nls,l10data,lngflg) 8D1A	01369
(mkrstay) (nls,l10data,mkrstay) 5F3	01391
(modeset) (nls,l10data,modeset) 5F1	01379
(modeshift) (nls,l10data,modeshift) 5F2	01366
(nsdbpt) (nls,l10data,nsdbpt) 5D1E	01356
(pe) (nls,l10data,pe) 8C2	01381
(ps) (nls,l10data,ps) 8C1	01384
(pstack) (nls,l10data,pstack) 5C2	01367
(rp1sid) (nls,l10data,rp1sid) 5D1B	01362
(sar) (nls,l10data,sar) 8A1	01375
(scndir) (nls,l10data,scndir) 5D1A	01387
(spsk) (nls,l10data,spsk) 8F1A	01358
(spsk1) (nls,l10data,spsk1) 8F1B	01392
(spskt) (nls,l10data,spskt) 8F1H	01382
(sptr1) (nls,l10data,sptr1) 5D1C	01376
(sptr2) (nls,l10data,sptr2) 5D1D	01385
(stcwr1) (nls,l10data,stcwr1) 5E2A	01365
(stcwrk) (nls,l10data,stcwrk) 5E2	01390
(swork) (nls,l10data,swork) 5E1	01359
(swork1) (nls,l10data,swork1) 5E1A	01378
(sysgn1) (nls,l10data,sysgn1) 8B2	01364
(sysmsg) (nls,l10data,sysmsg) 8B3	01371
	03897

(clcurm) (nls,l10runtime,)		02683
(clip) (nls,l10runtime,)		02779
(errmsb) (nls,l10runtime,)		02723
(apachr) (nls,l10runtime,apachr) 7D		02701
(apblnk) (nls,l10runtime,apblnk) 9L		02744
(apchr) (nls,l10runtime,apchr) 9F4		02647
(apsr) (nls,l10runtime,apsr) 9M		02697
(aptstr) (nls,l10runtime,aptstr) 7E		02650
(ascom) (nls,l10runtime,ascom) 9H		02667
(asrref) (nls,l10runtime,asrref) 9A		02721
(astruc) (nls,l10runtime,astruc) 9S		02732
(begarb) (nls,l10runtime,begarb) 10G5		02677
(bfs) (nls,l10runtime,bfs) 10E		02675
(bpadr) (nls,l10runtime,bpadr) 4A1A		02692
(bpbitpos) (nls,l10runtime,bpbitpos) 4A1D		02763
(bpindx) (nls,l10runtime,bpindx) 4A1B		02719
(bpsize) (nls,l10runtime,bpsize) 4A1C		02776
(bsc) (nls,l10runtime,bsc) 7A		02766
(callsig) (nls,l10runtime,callsig) 6G		02751
(cct) (nls,l10runtime,cct) 100		02706
(cctf) (nls,l10runtime,cctf) 10P		02646
(chbmtty) (nls,l10runtime,chbmtty) 3F5		02690
(chbptr) (nls,l10runtime,chbptr) 9F3		02684
(chpair) (nls,l10runtime,chpair) 90		02768
(chrct) (nls,l10runtime,chrct) 10N		02733
(cibuff) (nls,l10runtime,clbuff) 4D5		02641
(clcc1) (nls,l10runtime,clcc1) 4C3		02725
(clcc2) (nls,l10runtime,clcc2) 4C4		02749
(clcnt) (nls,l10runtime,clcnt) 4D1		02730
(cldsr) (nls,l10runtime,cldsr) 7F		02764
(clfixed) (nls,l10runtime,clfixed) 4C6		02643
(clfno1) (nls,l10runtime,clfno1) 4D2		02726
(clfno2) (nls,l10runtime,clfno2) 4D3		02712
(cll) (nls,l10runtime,cll) 4C7A		02780
(clst1) (nls,l10runtime,clst1) 4C1		02704
(clst2) (nls,l10runtime,clst2) 4C2		02656
(cltype) (nls,l10runtime,cltype) 4D4		02688
(cmpstr) (nls,l10runtime,cmpstr) 9J		02686
(compas) (nls,l10runtime,compas) 9I		02674
(cpfse) (nls,l10runtime,cpfse) 10S		02642
(cpysr) (nls,l10runtime,cpysr) 9N		02727
(dptr) (nls,l10runtime,dptr) 10B		02666
(empty) (nls,l10runtime,empty) 3F1		02700
(endarb) (nls,l10runtime,endarb) 10J		02739
(endfil) (nls,l10runtime,endfil) 3F2		02678
(esc) (nls,l10runtime,esc) 7B		02771
(fcp) (nls,l10runtime,fcp) 10G2		02729
(fechcl) (nls,l10runtime,fechcl) 8A		02722
(forward) (nls,l10runtime,forward) 3H1		02672
(fxsp1) (nls,l10runtime,fxsp1) 10G6		02747
(fxsp2) (nls,l10runtime,fxsp2) 10G7		02711
(fxswork) (nls,l10runtime,fxswork) 10F		02654
(hash) (nls,l10runtime,hash) 9R		02685
(incarb) (nls,l10runtime,incarb) 10I		02669
(kps) (nls,l10runtime,kps) 7C		02734
(l10init) (nls,l10runtime,l10init) 5B1		02773

(ldchr) (nls,l10runtime,ldchr) 9F2	02714
(lpr) (nls,l10runtime,lpr) 10G8	02657
(mkbptr) (nls,l10runtime,mkbptr) 9A@	02772
(mvbfbf) (nls,l10runtime,mvbfbf) 9P	02698
(nullch) (nls,l10runtime,nullch) 3F4	02705
(openswork) (nls,l10runtime,openswork) 8E	02735
(origin) (nls,l10runtime,origin) 3F3	02762
(pcp) (nls,l10runtime,pcp) 10G3	02696
(pdc) (nls,l10runtime,pdc) 10A	02640
(ppaprt) (nls,l10runtime,ppaprt) 6I1	02702
(qcall) (nls,l10runtime,qcall) 3E1	02707
(qcall1) (nls,l10runtime,qcall1) 3E2	02760
(qcallm) (nls,l10runtime,qcallm) 3E3	02748
(rcpsh) (nls,l10runtime,rcpsh) 11B	02740
(rcpto) (nls,l10runtime,rcpto) 11E	02761
(rdummy) (nls,l10runtime,rdummy) 4E6	02731
(readc) (nls,l10runtime,readc) 8E1	02738
(repchr) (nls,l10runtime,repchr) 9K	02724
(repet) (nls,l10runtime,repet) 4E7	02737
(repsig) (nls,l10runtime,repsig) 6F	02717
(repstr) (nls,l10runtime,repstr) 9Y	02755
(respos) (nls,l10runtime,respos) 8D	02687
(rhf) (nls,l10runtime,rhf) 4E8	02679
(rinst1) (nls,l10runtime,rinst1) 4E4	02774
(rinst2) (nls,l10runtime,rinst2) 4E5	02680
(rnamef) (nls,l10runtime,rnamef) 4E10	02769
(rnameh) (nls,l10runtime,rnameh) 4E13	02709
(rnull) (nls,l10runtime,rnull) 4E12	02652
(rpop) (nls,l10runtime,rpop) 11C	02691
(rsdb) (nls,l10runtime,rsdb) 4E3	02689
(rsid) (nls,l10runtime,rsid) 4E14	02745
(rstsk) (nls,l10runtime,rstsk) 11D	02756
(rsub) (nls,l10runtime,rsub) 4E1	02663
(rsuc) (nls,l10runtime,rsuc) 4E2	02708
(rtf) (nls,l10runtime,rtf) 4E9	02699
(rtorgin) (nls,l10runtime,rtorgin) 4E11	02682
(savpos) (nls,l10runtime,savpos) 8C	02770
(schars) (nls,l10runtime,schars) 4F3	02750
(scnlft) (nls,l10runtime,scnlft) 10L	02658
(scnrht) (nls,l10runtime,scnrht) 10M	02665
(scp) (nls,l10runtime,scp) 10K	02664
(sdbhdl) (nls,l10runtime,sdbhdl) 3G1	02676
(sgarb) (nls,l10runtime,sgarb) 4F1	02695
(sinit) (nls,l10runtime,sinit) 4F9	02715
(sitpsid) (nls,l10runtime,sitpsid) 4F12	02752
(slength) (nls,l10runtime,slength) 4F2	02644
(slngth) (nls,l10runtime,slngth) 9T	02668
(slnmdl) (nls,l10runtime,slnmdl) 4F4	02655
(sname) (nls,l10runtime,sname) 4F7	02694
(span) (nls,l10runtime,span) 10Q	02703
(spsdb) (nls,l10runtime,spsdb) 4F11	02673
(spsid) (nls,l10runtime,spsid) 4F6	02648
(sptr) (nls,l10runtime,sptr) 10G4	02716
(sptype) (nls,l10runtime,sptype) 4F10	02720
(srisset) (nls,l10runtime,srisset) 9Z	02777
(srmake) (nls,l10runtime,srmake) 9U	02778

(srmk) (nls,l10runtime,srmk) 9V	02651
(srnmdl) (nls,l10runtime,srnmdl) 4F5	02661
(srsup) (nls,l10runtime,srsup) 10G	02741
(srval) (nls,l10runtime,srval) 9W	02758
(stadr) (nls,l10runtime,stadr) 4B7A	02765
(stastr) (nls,l10runtime,stastr) 4B3	02713
(stblk) (nls,l10runtime,stblk) 4B5B	02736
(stbptr) (nls,l10runtime,stbptr) 9G	02767
(stfile) (nls,l10runtime,stfile) 4B2	02728
(stime) (nls,l10runtime,stime) 4F8	02743
(stpsdb) (nls,l10runtime,stpsdb) 4B6A	02718
(stpsid) (nls,l10runtime,stpsid) 4B1	02653
(stsid) (nls,l10runtime,stsid) 4B8B	02753
(stsidf) (nls,l10runtime,stsidf) 4B8A	02757
(stwc) (nls,l10runtime,stwc) 4B5A	02662
(sys2rt) (nls,l10runtime,sys2rt) 6C	02670
(sysmrt) (nls,l10runtime,sysmrt) 6D	02710
(sysrtn) (nls,l10runtime,sysrtn) 6B	02754
(syssig) (nls,l10runtime,syssig) 6E	02693
(systke) (nls,l10runtime,systke) 11A4	02775
(systkp) (nls,l10runtime,systkp) 11A1	02660
(systks) (nls,l10runtime,systks) 11A3	02681
(systkt) (nls,l10runtime,systkt) 11A2	02746
(tcht) (nls,l10runtime,tchf) 10R	02659
(tstf) (nls,l10runtime,tstf) 10D	02759
(tstsr) (nls,l10runtime,tstsr) 10C	02649
(xl10init) (nls,l10runtime,xl10init) 5	02671
(xutility) (nls,l10runtime,xutility) 9F	02742
(xxreadc) (nls,l10runtime,xxreadc) 8E	02645

03898

(accum1)	(nls,nddt2,)		03438
(addr10)	(nls,nddt2,)		03430
(calddt)	(nls,nddt2,)		03425
(index1)	(nls,nddt2,)		03437
(indir1)	(nls,nddt2,)		03421
(longop)	(nls,nddt2,)		03420
(opcod1)	(nls,nddt2,)		03429
(pdummy)	(nls,nddt2,)		03443
(ddgtsymbol)	(nls,nddt2,ddgtsymbol)	1F1	03416
(ddsymget)	(nls,nddt2,ddsymget)	1F2	03418
(ddtchkp)	(nls,nddt2,ddtchkp)	1F11	03432
(ddtenter)	(nls,nddt2,ddtenter)	1F4	03417
(ddtlookup)	(nls,nddt2,ddtlookup)	1F5	03441
(ddtmark)	(nls,nddt2,ddtmark)	1F7	03426
(ddtname)	(nls,nddt2,ddtname)	1F8	03433
(ddtpop)	(nls,nddt2,ddtpop)	1F10	03415
(ddtsrchblk)	(nls,nddt2,ddtsrchblk)	1F9	03414
(findbp)	(nls,nddt2,findbp)	1E7	03423
(getins)	(nls,nddt2,getins)	1F2R	03419
(mrkglookup)	(nls,nddt2,mrkglookup)	1F6	03436
(nddtarm)	(nls,nddt2,nddtarm)	1E1	03424
(nddt disarm)	(nls,nddt2,nddt disarm)	1E2	03427
(octnum)	(nls,nddt2,octnum)	1E5	03440
(r50toa)	(nls,nddt2,r50toa)	1F3	03422
(sethint)	(nls,nddt2,sethint)	1E3	03439
(shl)	(nls,nddt2,shl)	1F2R19	03435
(shr)	(nls,nddt2,shr)	1F2R20	03442
(tblsearch)	(nls,nddt2,tblsearch)	1E6	03434
(typval)	(nls,nddt2,typval)	1E8	03428
(unsethint)	(nls,nddt2,unsethint)	1E4	03431

03899

(acalddt) (nls,nddtdata,acalddt) 3E1	01396
(bptbl) (nls,nddtdata,bptbl) 3B1	01405
(db) (nls,nddtdata,db) 3A4	01395
(ddmrk) (nls,nddtdata,ddmrk) 3A1	01413
(ddmrkm) (nls,nddtdata,ddmrkm) 3A2	01397
(ddmrkx) (nls,nddtdata,ddmrkx) 3A3	01406
(ddtinst) (nls,nddtdata,ddtinst) 3B4	01409
(ddtjunk) (nls,nddtdata,ddtjunk) 3B5	01404
(ddtlit) (nls,nddtdata,ddtlit) 3F1	01393
(ddtrg1) (nls,nddtdata,ddtrg1) 3B6	01403
(ddtrg2) (nls,nddtdata,ddtrg2) 3B7	01410
(ddtrg3) (nls,nddtdata,ddtrg3) 3B8	01394
(ddtrloc) (nls,nddtdata,ddtrloc) 3B2	01399
(ddtrp) (nls,nddtdata,ddtrp) 3B3	01407
(displayflag) (nls,nddtdata,displayflag) 3E3	01401
(dnstr) (nls,nddtdata,dnstr) 3G1	01402
(hstkhed) (nls,nddtdata,hstkhed) 3C2	01412
(hsymptr) (nls,nddtdata,hsymptr) 3C3	01398
(hsymb) (nls,nddtdata,hsymb) 3C1	01411
(levtbl) (nls,nddtdata,levtbl) 3E2	01400
(proctbl) (nls,nddtdata,proctbl) 3D1	01408
	03900

(m) (nls,parser,)		0179
(p) (nls,parser,)		0180
(r1) (nls,parser,)		0162
(r2) (nls,parser,)		0192
(r3) (nls,parser,)		0160
(r4) (nls,parser,)		0187
(s) (nls,parser,)		0184
(add2help) (nls,parser,add2help) 6D5		0171
(addhelp) (nls,parser,addhelp) 6D3		0161
(ap2cistr) (nls,parser,ap2clstr) 6D2		0193
(bugtrap) (nls,parser,bugtrap) 4I		0177
(cmdfinish) (nls,parser,cmdfinish) 4H		0185
(cmdinterp) (nls,parser,cmdinterp) 5A1		0168
(cshelp) (nls,parser,cshelp) 6E1		0172
(decode) (nls,parser,decode) 5A6		0174
(delsubsys) (nls,parser,delsubsys) 4F		0165
(dfnsubsys) (nls,parser,dfnsubsys) 4E		0188
(edistr) (nls,parser,edistr) 5C1		0175
(enthelp) (nls,parser,enthelp) 4D		0158
(evaler) (nls,parser,evaler) 5A4		0183
(ibbidhlp) (nls,parser,fbldhlp) 6D4		0159
(ibctl) (nls,parser,fbctl) 6C1		0176
(ibhelp) (nls,parser,fbhelp) 6D1		0157
(indwhere) (nls,parser,fnwhere) 6E3		0152
(indxec) (nls,parser,fnxec) 6E4		0166
(getsdptr) (nls,parser,getsdptr) 4B		0164
(gotohelp) (nls,parser,gotohelp) 4C		0181
(initsubsystems) (nls,parser,initsubsystems) 4A		0190
(keyinit) (nls,parser,keyinit) 6B3		0155
(keywrec) (nls,parser,keywrec) 6A1		0194
(nextkey) (nls,parser,nextkey) 6B2		0163
(prspathstk) (nls,parser,prspathstk) 6E2		0169
(setbackup) (nls,parser,setbackup) 5A2		0182
(sethrlid) (nls,parser,sethrlid) 4J		0156
(sleuth) (nls,parser,sleuth) 5A3		0153
(supervisor) (nls,parser,supervisor) 4G		0173
(testselect) (nls,parser,testselect) 5A5		0195
(vocab) (nls,parser,vocab) 6B1		0167
(xfeedback) (nls,parser,xfeedback) 5D1		0191
(xpop) (nls,parser,xpop) 5B2		0189
(xpush) (nls,parser,xpush) 5B1		0178
(xread) (nls,parser,xread) 5B4		0170
(xstor) (nls,parser,xstor) 5D2		0154
(xstore) (nls,parser,xstore) 5B3		0186

03901

(call0) (nls,pdata,)		018
(call1) (nls,pdata,)		040
(callm) (nls,pdata,)		085
(jsys) (nls,pdata,)		032
(aheadfilename) (nls,pdata,aheadfilename)	5A2	012
(allsubs) (nls,pdata,allsubs)	7B6A4	020
(anyof) (nls,pdata,anyof)	2E1B2	0118
(auxintlag) (nls,pdata,auxinflag)	7B11A5	097
(basestateflag) (nls,pdata,basestateflag)	7B11A6	043
(caaction) (nls,pdata,caaction)	7B10A2	092
(cachar) (nls,pdata,cachar)	7A5A1	062
(call) (nls,pdata,call)	2E1C3	0101
(cdlnk) (nls,pdata,cdlnk)	7B11A3	035
(clpsw) (nls,pdata,clpsw)	7B11A4	08
(cmdmode) (nls,pdata,cmdmode)	7B7A1	042
(compicode) (nls,pdata,complcode)	7B9A1	0132
(confirm) (nls,pdata,confirm)	2E1A2	051
(cspcacode) (nls,pdata,cspcacode)	6A4	027
(cspupdate) (nls,pdata,cspupdate)	6A2	089
(cspusqcod) (nls,pdata,cspusqcod)	6A5	081
(cspvs) (nls,pdata,cspvs)	6A6	0119
(cueflg) (nls,pdata,cueflg)	7B11B1	024
(curmkr) (nls,pdata,curmkr)	6A1	079
(cutstop) (nls,pdata,cutstop)	7A4A1	053
(defaction) (nls,pdata,defaction)	7B10A1	055
(dent) (nls,pdata,dent)	5A3	033
(dest) (nls,pdata,dest)	5A4	0130
(dptvldationcode) (nls,pdata,dptvldationcode)	2B2A	082
(dsel) (nls,pdata,dsel)	2E1A4	0115
(ent) (nls,pdata,ent)	5A5	056
(enter) (nls,pdata,enter)	2E1E3	037
(eschar) (nls,pdata,eschar)	2C1A	0124
(evalsize) (nls,pdata,evalsize)	2A1I	011
(evalstk) (nls,pdata,evalstk)	7B14A2	069
(evalx) (nls,pdata,evalx)	7B14A1	090
(fbclear) (nls,pdata,fbclear)	2E1D1	073
(fbstr) (nls,pdata,fbstr)	7B7A3	076
(fbstrmax) (nls,pdata,fbstrmax)	2A1N	017
(ff) (nls,pdata,ff)	5A8	080
(filtre) (nls,pdata,filtre)	5A6	094
(framecounter) (nls,pdata,framecounter)	7A3A1	046
(framesize) (nls,pdata,framesize)	2A1L	0102
(fromwhom) (nls,pdata,fromwhom)	5A9	063
(frrcnt) (nls,pdata,frrcnt)	9A1	0103
(funrecsize) (nls,pdata,funrecsize)	2A1B	052
(gparam) (nls,pdata,gparam)	5A1A	03
(gparm2) (nls,pdata,gparm2)	5A1B	041
(gparm3) (nls,pdata,gparm3)	5A1C	010
(gparm4) (nls,pdata,gparm4)	5A1D	098
(grammar) (nls,pdata,grammar)	7A2A1	0109
(hlpcmdstk) (nls,pdata,hlpcmdstk)	8A1	067
(npcmdmax) (nls,pdata,hpcmdmax)	8A2	074
(hrlidstr) (nls,pdata,hrlidstr)	7B2A1	087
(inpt) (nls,pdata,inpt)	7A1A1	060
(inschar) (nls,pdata,inschar)	7A5A4	0131
(keyinpstr) (nls,pdata,keyinpstr)	4A1	0107

(keyop) (nls,pdata,keyop) 2E1A1	04
(keyrptstr) (nls,pdata,keyrptstr) 4B1	026
(keysaveflag) (nls,pdata,keysaveflag) 4C1	0117
(lastsel) (nls,pdata,lastsel) 7B11A8	083
(leftdelim) (nls,pdata,leftdelim) 2C1B	025
(levadj) (nls,pdata,levadj) 2E1A8	059
(level) (nls,pdata,level) 5A10	02
(literal) (nls,pdata,literal) 5A11	0122
(littakedown) (nls,pdata,littakedown) 3B	0108
(lssel) (nls,pdata,lssel) 2E1A5	068
(namri1) (nls,pdata,namfil) 5A7	0126
(necho) (nls,pdata,necho) 2E1D2	019
(needconfirm) (nls,pdata,needconfirm) 7B11A7	072
(nlssubs) (nls,pdata,nlssubs) 7B6A2	095
(nofail) (nls,pdata,nofail) 7B7A2	099
(nwink) (nls,pdata,nwink) 7B11A2	065
(optaction) (nls,pdata,optaction) 7B10A3	022
(optchar) (nls,pdata,optchar) 7A5A2	078
(option) (nls,pdata,option) 2E1B1	048
(parallel) (nls,pdata,parallel) 2D1B	034
(param) (nls,pdata,param) 5A12	06
(param1) (nls,pdata,param1) 5A13	038
(param2) (nls,pdata,param2) 5A14	084
(param3) (nls,pdata,param3) 5A15	0127
(param4) (nls,pdata,param4) 5A16	021
(parsefunction) (nls,pdata,parsefunction) 2D1D	0106
(pathbase) (nls,pdata,pathbase) 7B12D3	0116
(pathrecsize) (nls,pdata,pathrecsize) 2A1C	09
(pathstk) (nls,pdata,pathstk) 7B12D2	045
(pathx) (nls,pdata,pathx) 7B12D1	015
(pb) (nls,pdata,pb) 5A17	057
(pca) (nls,pdata,pca) 2E1A6	070
(pfcall) (nls,pdata,pfcall) 2E1C1	044
(pload) (nls,pdata,pload) 2E1E2	0121
(port) (nls,pdata,port) 5A18	023
(promptstr) (nls,pdata,promptstr) 7B1A1	028
(prsavearea) (nls,pdata,prsavearea) 7B8A2	050
(prsavevx) (nls,pdata,prsavevx) 7B8A1	049
(prsvsize) (nls,pdata,prsvsize) 2A1A	0128
(psdepth) (nls,pdata,psdepth) 2A1E	039
(pssize) (nls,pdata,pssize) 2A1F	061
(ptrssize) (nls,pdata,ptrssize) 2A1H	047
(ptrstk) (nls,pdata,ptrstk) 7B13B2	014
(ptrx) (nls,pdata,ptrx) 7B13B1	013
(recho) (nls,pdata,recho) 2E1D3	091
(retfilename) (nls,pdata,retfilename) 5A19	054
(rptchar) (nls,pdata,rptchar) 7A5A3	0123
(s1) (nls,pdata,s1) 5A20	05
(s2) (nls,pdata,s2) 5A21	071
(savhelpstr) (nls,pdata,savhelpstr) 7B3A1	030
(sbentsize) (nls,pdata,sbentsize) 2A1K	096
(sbstack) (nls,pdata,sbstack) 7B5B2	0111
(sbstksize) (nls,pdata,sbstksize) 2A1J	075
(sbstkx) (nls,pdata,sbstkx) 7B5B1	093
(sdptsize) (nls,pdata,sdptsize) 2A1M	0129
(sdptx) (nls,pdata,sdptx) 7B6A1	016

(sdptxa) (nls,pdata,sdptxa) 7B6A3	077
(selstatesize) (nls,pdata,selstatesize) 2A1G	086
(sent) (nls,pdata,sent) 5A22	088
(serial) (nls,pdata,serial) 2D1C	0112
(sim) (nls,pdata,sim) 5A23	066
(slink) (nls,pdata,slink) 7B11A1	0114
(source) (nls,pdata,source) 5A24	0120
(srrcnt) (nls,pdata,srrcnt) 9A2	0113
(sse1) (nls,pdata,sse1) 2E1A3	036
(ssysname) (nls,pdata,ssysname) 7B4A1	031
(store) (nls,pdata,store) 2E1E1	0100
(totalrecsize) (nls,pdata,totalrecsize) 2A1D	058
(unknown) (nls,pdata,unknown) 2D1A	0125
(usesrr) (nls,pdata,usesrr) 6A3	0110
(valueof) (nls,pdata,valueof) 2E1E4	0105
(vs) (nls,pdata,vs) 5A25	07
(vwspecs) (nls,pdata,vwspecs) 2E1A7	064
(wasize) (nls,pdata,wasize) 2A10	0104
(xecute) (nls,pdata,xecute) 2E1C2	029

03902

(in0stca) (nls,prmspc,in0stca) 7A	0539
(in0tca) (nls,prmspc,in0tca) 5A	0555
(in1ca) (nls,prmspc,in1ca) 4A	0545
(in1sadj) (nls,prmspc,in1sadj) 8A	0566
(in1sca) (nls,prmspc,in1sca) 6A	0560
(in1stca) (nls,prmspc,in1stca) 7B	0568
(in1tca) (nls,prmspc,in1tca) 5B	0550
(in2ca) (nls,prmspc,in2ca) 4B	0540
(in2sadj) (nls,prmspc,in2sadj) 8B	0557
(in2sca) (nls,prmspc,in2sca) 6B	0567
(in2stca) (nls,prmspc,in2stca) 7C	0559
(in2tca) (nls,prmspc,in2tca) 5C	0542
(in3ca) (nls,prmspc,in3ca) 4C	0570
(in3sadj) (nls,prmspc,in3sadj) 8C	0549
(in3sca) (nls,prmspc,in3sca) 6C	0541
(in3stca) (nls,prmspc,in3stca) 7D	0547
(in3tca) (nls,prmspc,in3tca) 5D	0558
(in4ca) (nls,prmspc,in4ca) 4D	0561
(in4sca) (nls,prmspc,in4sca) 6D	0572
(inbug) (nls,prmspc,inbug) 3A	0556
(inch) (nls,prmspc,inch) 3K	0551
(incont) (nls,prmspc,incont) 3F	0562
(infile) (nls,prmspc,infilename) 3M	0575
(inlevadj) (nls,prmspc,inlevadj) 3D	0565
(iname) (nls,prmspc,inname) 3I	0553
(innmwd) (nls,prmspc,innmwd) 3J	0571
(innum) (nls,prmspc,innum) 3D	0543
(insr) (nls,prmspc,insr) 3E	0576
(instid) (nls,prmspc,instid) 3B	0548
(intext) (nls,prmspc,intext) 3C	0574
(invsbl) (nls,prmspc,invsbl) 3G	0552
(invspc) (nls,prmspc,invspc) 3N	0538
(inword) (nls,prmspc,inword) 3H	0546
(oldanswer) (nls,prmspc,oldanswer) 3L	0554
(putvspc) (nls,prmspc,putvspc) 9B	0569
(savvspc) (nls,prmspc,savvspc) 9A	0573
(softcd) (nls,prmspc,softcd) 10	0544
(stkit) (nls,prmspc,stkit) 9C	0563
(strbug) (nls,prmspc,strbug) 11	0564

03903

(copyfl) (nls,psedit,)	0296
(deltfl) (nls,psedit,)	0266
(movefl) (nls,psedit,)	0273
(trnsfl) (nls,psedit,)	0285
(conbp) (nls,psedit,conbp) 4I	0284
(dfltname) (nls,psedit,dfltname) 3P5	0281
(sbinit) (nls,psedit,sbinit) 4B	0268
(sbpush) (nls,psedit,sbpush) 4G	0275
(subldsp) (nls,psedit,subldsp) 4C	0272
(sub2dsp) (nls,psedit,sub2dsp) 4D	0265
(subpsave) (nls,psedit,subpsave) 4F	0314
(subsinit) (nls,psedit,subsinit) 4A	0261
(substatus) (nls,psedit,substatus) 4E	0294
(xappend) (nls,psedit,xappend) 3A1	0298
(xarchive) (nls,psedit,xarchive) 3B1	0270
(xbreak) (nls,psedit,xbreak) 3C1	0309
(xbslash) (nls,psedit,xbslash) 3A01	0289
(xcmgrp) (nls,psedit,xcmgrp) 3D2C	0301
(xcmst) (nls,psedit,xcmst) 3D2B	0300
(xcopy) (nls,psedit,xcopy) 3D1	0286
(xcreate) (nls,psedit,xcreate) 3E1	0304
(xdelete) (nls,psedit,xdelete) 3F1	0312
(xdiopt) (nls,psedit,xdiopt) 3D2A	0279
(xdlpnt) (nls,psedit,xdlpnt) 3A@3A	0262
(xedit) (nls,psedit,xedit) 3G1	0276
(xexpunge) (nls,psedit,xexpunge) 3H1	0318
(xforce) (nls,psedit,xforce) 3I1	0267
(xinsert) (nls,psedit,xinsert) 3J1	0317
(xinsstatement) (nls,psedit,xinsstatement) 3J2	0280
(xlinefeed) (nls,psedit,xlinefeed) 3AK1	0271
(xload) (nls,psedit,xload) 3K1	0290
(xlogout) (nls,psedit,xlogout) 3L1	0263
(xmark) (nls,psedit,xmark) 3M1	0288
(xmove) (nls,psedit,xmove) 3O1	0319
(xout1) (nls,psedit,xout1) 3P3	0313
(xout2) (nls,psedit,xout2) 3P4	0305
(xoutsapf) (nls,psedit,xoutsapf) 3P1	0277
(xoutsnhf) (nls,psedit,xoutsnhf) 3P2	0299
(xperiod) (nls,psedit,xperiod) 3AM1	0307
(xplayback) (nls,psedit,xplayback) 3Q1	0278
(xprint) (nls,psedit,xprint) 3R1	0297
(xprocess) (nls,psedit,xprocess) 3S1	0282
(xrenumber) (nls,psedit,xrenumber) 3V1	0303
(xreplace) (nls,psedit,xreplace) 3W1	0321
(xreset) (nls,psedit,xreset) 3X1	0311
(xretrieve) (nls,psedit,xretrieve) 3Y1	0316
(xset) (nls,psedit,xset) 3Z1	0302
(xshoviespe) (nls,psedit,xshoviespe) 3A@2	0295
(xshow) (nls,psedit,xshow) 3A@1	0320
(xslash) (nls,psedit,xslash) 3AN1	0269
(xsort) (nls,psedit,xsort) 3AA1	0292
(xstart) (nls,psedit,xstart) 3U1	0287
(xstop) (nls,psedit,xstop) 3AC1	0315
(xsubstitute) (nls,psedit,xsubstitute) 3AD1	0264
(xtab) (nls,psedit,xtab) 3AJ1	0293
(xtranspose) (nls,psedit,xtranspose) 3AE1	0291

(xtrim) (nls,psedit,xtrim) 3AF1	0310
(xundelete) (nls,psedit,xundelete) 3AG1	0274
(xuparrow) (nls,psedit,xuparrow) 3AL1	0283
(xupdate) (nls,psedit,xupdate) 3AH1	0308
(xverify) (nls,psedit,xverify) 3AI1	0306
	03904

(helpfl) (nls,pshelp,)		0475
(menusw) (nls,pshelp,)		0479
(checkmore) (nls,pshelp,checkmore)	3H	0466
(dumprompt) (nls,pshelp,dumprompt)	3C	0478
(helpshow) (nls,pshelp,helpshow)	4B	0469
(hlpinit) (nls,pshelp,hlpinit)	4A	0476
(lkup) (nls,pshelp,lkup)	3B	0477
(lookback) (nls,pshelp,lookback)	3A	0468
(looknxt) (nls,pshelp,looknxt)	3D	0471
(mylookbug) (nls,pshelp,mylookbug)	3E	0465
(myrdconfirm) (nls,pshelp,myrdconfirm)	3F	0474
(qbkint) (nls,pshelp,qbkint)	4D	0472
(rstmor) (nls,pshelp,rstmor)	3I	0467
(sethflg) (nls,pshelp,sethflg)	3G	0470
(xhlpriing) (nls,pshelp,xhlpriing)	4C	0473

03905

(xpattach) (nls,psprogs,xpattach) 3A	0376
(xpcompile) (nls,psprogs,xpcompile) 3B	0375
(xpdeinstitute) (nls,psprogs,xpdeinstitute) 3D	0363
(xpdelete) (nls,psprogs,xpdelete) 3C	0369
(xpdetach) (nls,psprogs,xpdetach) 3E	0373
(xp institute) (nls,psprogs,xp institute) 3F	0365
(xpkill) (nls,psprogs,xpkill) 3G	0368
(xpload) (nls,psprogs,xpload) 3H	0374
(xpreset) (nls,psprogs,xpreset) 3L	0367
(xprun) (nls,psprogs,xprun) 3I	0371
(xprunt) (nls,psprogs,xprunt) 3M	0364
(xpset) (nls,psprogs,xpset) 3N	0362
(xpsetf) (nls,psprogs,xpsetf) 3O	0366
(xpsnow) (nls,psprogs,xpsnow) 3P	0372
(xpshtn) (nls,psprogs,xpshtn) 3Q	0370

03906

(xjcopy) (nls,pssendmail,)		0327
(jidstatus) (nls,pssendmail,jidstatus) 4V		0347
(jsetfield) (nls,pssendmail,jsetfield) 4U		0336
(sjaccess) (nls,pssendmail,sjaccess) 3N		0360
(sjaction) (nls,pssendmail,sjaction) 3E		0337
(sjauthor) (nls,pssendmail,sjauthor) 3D		0345
(sjbacksendit) (nls,pssendmail,sjbacksendit) 3X		0338
(sjbranch) (nls,pssendmail,sjbranch) 3S		0340
(sjcomment) (nls,pssendmail,sjcomment) 3C		0334
(sjfile) (nls,pssendmail,sjfile) 3V		0325
(sjforward) (nls,pssendmail,sjforward) 3Q		0332
(sjgroup) (nls,pssendmail,sjgroup) 3U		0346
(sjhandling) (nls,pssendmail,sjhandling) 3I		0355
(sjhardcopy) (nls,pssendmail,sjhardcopy) 3K		0361
(sjinfo) (nls,pssendmail,sjinfo) 3F		0339
(sjkeywords) (nls,pssendmail,sjkeywords) 3H		0350
(sjlink) (nls,pssendmail,sjlink) 3P		0322
(sjmessage) (nls,pssendmail,sjmessage) 3R		0352
(sjnumber) (nls,pssendmail,sjnumber) 3A		0341
(sjobsolete) (nls,pssendmail,sjobsolete) 3M		0333
(sjplex) (nls,pssendmail,sjplex) 3T		0357
(sjrecording) (nls,pssendmail,sjrecording) 3J		0326
(sjrtc) (nls,pssendmail,sjrtc) 3L		0349
(sjsendit) (nls,pssendmail,sjsendit) 3W		0335
(sjsubcol) (nls,pssendmail,sjsubcol) 3G		0324
(sjtitle) (nls,pssendmail,sjtitle) 3B		0358
(sjupdates) (nls,pssendmail,sjupdates) 3O		0348
(xjdoit) (nls,pssendmail,xjdoit) 4I		0331
(xjforward) (nls,pssendmail,xjforward) 4T		0343
(xjinsrecord) (nls,pssendmail,xjinsrecord) 4P		0330
(xjinsstatus) (nls,pssendmail,xjinsstatus) 4O		0323
(xjlistatus) (nls,pssendmail,xjlistatus) 4S		0356
(xjloaworfil) (nls,pssendmail,xjloaworfil) 4B		0328
(xjlock) (nls,pssendmail,xjlock) 4J		0353
(xjprocess) (nls,pssendmail,xjprocess) 4N		0351
(xjreserve) (nls,pssendmail,xjreserve) 4F		0344
(xjrtcreserve) (nls,pssendmail,xjrtcreserve) 4G		0342
(xjrtcscho) (nls,pssendmail,xjrtcscho) 4H		0359
(xjstatus) (nls,pssendmail,xjstatus) 4R		0354
(xjzapworfil) (nls,pssendmail,xjzapworfil) 4A		0329

03907

(bldsynsub) (nls,pssyngen,bldsynsub)	7	0484
(gtads) (nls,pssyngen,gtads)	6	0483
(xkbuild) (nls,pssyngen,xkbuild)	5	0480
(xkcopy) (nls,pssyngen,xkcopy)	4	0481
(xkshow) (nls,pssyngen,xkshow)	3	0482
		03908

(abortssystem) (nls,pssystem,abortssystem) 3G	0458
(brkhere) (nls,pssystem,brkhere) 503	0452
(ddt) (nls,pssystem,ddt) 50	0459
(jniout) (nls,pssystem,jniout) 5F	0441
(updcsp) (nls,pssystem,updcsp) 3H2	0454
(xcutil) (nls,pssystem,xcutil) 5G	0440
(xdummy) (nls,pssystem,xdummy) 11C	0446
(xgoexec) (nls,pssystem,xgoexec) 4A	0451
(xgoto) (nls,pssystem,xgoto) 3A	0444
(xgtcrsb) (nls,pssystem,xgtcrsb) 3F	0462
(xjnidelete) (nls,pssystem,xjnidelete) 5J	0449
(xjnload) (nls,pssystem,xjnload) 5I	0460
(xjoutil) (nls,pssystem,xjoutil) 5E	0448
(xjump) (nls,pssystem,xjump) 3H1	0463
(xksyntax) (nls,pssystem,xksyntax) 3C	0461
(xpriority) (nls,pssystem,xpriority) 5M	0445
(xquit) (nls,pssystem,xquit) 3B	0457
(xsetjou) (nls,pssystem,xsetjou) 5H	0456
(xsetld) (nls,pssystem,xsetld) 5L	0455
(xsetpar) (nls,pssystem,xsetpar) 5K	0442
(xsubcurrent) (nls,pssystem,xsubcurrent) 3E	0453
(xsublist) (nls,pssystem,xsublist) 3D	0450
(xxcheck) (nls,pssystem,xxcheck) 5P	0447
(xxddt) (nls,pssystem,xxddt) 5N	0464
(xxdetach) (nls,pssystem,xxdetach) 5Q	0443

03909

(cutpat) (nls,psupport,)	0253
(abort) (nls,psupport,abort) 6A	0234
(answ) (nls,psupport,answ) 3D1	0232
(answer) (nls,psupport,answer) 3D2	0226
(clearname) (nls,psupport,clearname) 8A	0242
(clist) (nls,psupport,clist) 9A	0252
(clupdt) (nls,psupport,clupdt) 9C	0245
(cutback) (nls,psupport,cutback) 7A	0244
(dpset) (nls,psupport,dpset) 9D	0255
(dpstp) (nls,psupport,dpstp) 9E	0235
(false) (nls,psupport,false) 3L1	0251
(getpint) (nls,psupport,getpint) 10A	0231
(isdnl) (nls,psupport,isdnl) 3K1	0246
(istnl) (nls,psupport,istnl) 3K2	0258
(lookansw) (nls,psupport,lookansw) 3F1	0257
(lookbug) (nls,psupport,lookbug) 3I2	0225
(lookconfirm) (nls,psupport,lookconfirm) 3I1	0229
(looknum) (nls,psupport,looknum) 3J1	0247
(lookstat) (nls,psupport,lookstat) 3E1	0233
(notca) (nls,psupport,notca) 3G1	0239
(prmptdsel) (nls,psupport,prmptdsel) 3M2	0237
(prmptsel) (nls,psupport,prmptsel) 3M1	0227
(prmptsel) (nls,psupport,prmptsel) 3M3	0250
(readbug) (nls,psupport,readbug) 3H2	0249
(readconfirm) (nls,psupport,readconfirm) 3H1	0240
(readoption) (nls,psupport,readoption) 3B1	0259
(readrepeat) (nls,psupport,readrepeat) 3C1	0230
(seqbadd) (nls,psupport,seqbadd) 5B	0241
(seqbinit) (nls,psupport,seqbinit) 5A	0243
(setbkup) (nls,psupport,setbkup) 4A	0248
(setcutback) (nls,psupport,setcutback) 7B	0260
(sp) (nls,psupport,sp) 3A1	0228
(sublookansw) (nls,psupport,sublookansw) 3F2	0256
(true) (nls,psupport,true) 3L2	0254
(upclptr) (nls,psupport,upclptr) 9B	0236
(xwarning) (nls,psupport,xwarning) 11A	0238

03910

(bildtabwrds) (nls,psuserop,bildtabwrds) 2B11	0379
(chrstr) (nls,psuserop,chrstr) 2D2	0418
(devstr) (nls,psuserop,devstr) 2D4	0396
(getpstring) (nls,psuserop,getpstring) 2B12	0420
(strchr) (nls,psuserop,strchr) 2D1	0438
(strdev) (nls,psuserop,strdev) 2D3	0388
(uoaccess) (nls,psuserop,uoaccess) 2B7	0392
(uoclose) (nls,psuserop,uoclose) 2B6	0380
(uoget) (nls,psuserop,uoget) 2B4	0405
(uoincnam) (nls,psuserop,uoincnam) 2B5	0402
(uolchap) (nls,psuserop,uolchap) 2C2	0426
(uolcrap) (nls,psuserop,uolcrap) 2C3	0397
(uolitap) (nls,psuserop,uolitap) 2C1	0390
(uolstrap) (nls,psuserop,uolstrap) 2C4	0416
(uorescont) (nls,psuserop,uorescont) 2B9	0424
(uoreset) (nls,psuserop,uoreset) 2B1	0431
(uoresspfortab) (nls,psuserop,uoresspfortab) 2B3	0437
(uoresub) (nls,psuserop,uoresub) 2B2	0400
(uoshcc) (nls,psuserop,uoshcc) 2C8	0429
(uoshctrl) (nls,psuserop,uoshctrl) 2C6	0410
(uoshcurc) (nls,psuserop,uoshcurc) 2C9	0391
(uoshdisplay) (nls,psuserop,uoshdisplay) 2C10	0419
(uoshesub) (nls,psuserop,uoshesub) 2C11	0408
(uoshextn) (nls,psuserop,uoshextn) 2C26	0407
(uoshfeed) (nls,psuserop,uoshfeed) 2C14	0389
(uoshisubs) (nls,psuserop,uoshisubs) 2C13	0411
(uoshjmp) (nls,psuserop,uoshjmp) 2C16	0398
(uoshlev) (nls,psuserop,uoshlev) 2C17	0386
(uoshnmd) (nls,psuserop,uoshnmd) 2C18	0413
(uoshoutput) (nls,psuserop,uoshoutput) 2C19	0436
(uoshow) (nls,psuserop,uoshow) 2C5	0381
(uoshpmt) (nls,psuserop,uoshpmt) 2C22	0417
(uoshprint) (nls,psuserop,uoshprint) 2C20	0393
(uoshreco) (nls,psuserop,uoshreco) 2C23	0401
(uoshrid) (nls,psuserop,uoshrid) 2C15	0399
(uoshsc) (nls,psuserop,uoshsc) 2C7	0394
(uoshspfortab) (nls,psuserop,uoshspfortab) 2C21	0406
(uoshssub) (nls,psuserop,uoshssub) 2C12	0430
(uoshstup) (nls,psuserop,uoshstup) 2C25	0382
(uoshview) (nls,psuserop,uoshview) 2C24	0434
(uotabs) (nls,psuserop,uotabs) 2B10	0383
(uotblget) (nls,psuserop,uotblget) 2B8	0423
(xuocontchars) (nls,psuserop,xuocontchars) 2A4	0439
(xuocurcon) (nls,psuserop,xuocurcon) 2A5	0403
(xuodisplay) (nls,psuserop,xuodisplay) 2A6	0421
(xuoexclude) (nls,psuserop,xuoexclude) 2A7	0384
(xuoextn) (nls,psuserop,xuoextn) 2A15	0427
(xuofeedback) (nls,psuserop,xuofeedback) 2A8	0428
(xuoheerald) (nls,psuserop,xuoheerald) 2A9	0404
(xuoinclude) (nls,psuserop,xuoinclude) 2A10	0414
(xuoinit) (nls,psuserop,xuoinit) 2A2	0415
(xuonamed) (nls,psuserop,xuonamed) 2A14	0432
(xuooutput) (nls,psuserop,xuooutput) 2A16	0412
(xuoprint) (nls,psuserop,xuoprint) 2A17	0387
(xuoprompt) (nls,psuserop,xuoprompt) 2A18	0422
(xuorecognition) (nls,psuserop,xuorecognition) 2A19	0435

(xuroreset) (nls,psuserop,xuroreset) 2A20	0395
(xuringsize) (nls,psuserop,xuringsize) 2A21	0409
(xuoshow) (nls,psuserop,xuoshow) 2A22	0377
(xuospfortab) (nls,psuserop,xuospfortab) 2A23	0433
(xuostup) (nls,psuserop,xuostup) 2A24	0378
(xuoterm) (nls,psuserop,xuoterm) 2A3	0425
(xuoviewspecs) (nls,psuserop,xuoviewspecs) 2A25	0385
	03911

(copsrc) (nls,recfil,copsrc) 9	03463
(cxdetach) (nls,recfil,cxdetach) 3	03461
(nlsutility) (nls,recfil,nlsutility) 4	03464
(outptr) (nls,recfil,outptr) 8	03460
(runcmp) (nls,recfil,runcmp) 7	03459
(setpriority) (nls,recfil,setpriority) 5	03458
(setupop) (nls,recfil,setupop) 6	03462
	03912

(gmtfla) (nls,sdata,)		01342
(lcltim) (nls,sdata,)		01351
(ccignore) (nls,sdata,ccignore) 4F10		01329
(cfreeblk) (nls,sdata,cfreeblk) 4E2		01310
(cgetblk) (nls,sdata,cgetblk) 4E1		01318
(chntab) (nls,sdata,chntab) 4F7		01340
(cinstid) (nls,sdata,cinstid) 6B1B		01319
(clink) (nls,sdata,clink) 4E6		01353
(cmakezone) (nls,sdata,cmakezone) 4E3		01322
(creplenish) (nls,sdata,creplenish) 4E4		01316
(csstk) (nls,sdata,csstk) 6A1C		01326
(csstkb) (nls,sdata,csstkb) 6A1B		01321
(csstkx) (nls,sdata,csstkx) 6A1A		01355
(cunlink) (nls,sdata,cunlink) 4E5		01331
(entvec) (nls,sdata,entvec) 4A1		01308
(ilevel) (nls,sdata,ilevel) 6B1C		01313
(ilsdsp) (nls,sdata,ilsdsp) 4B1		01312
(lev1lc) (nls,sdata,lev1lc) 4F4		01328
(lev2lc) (nls,sdata,lev2lc) 4F5		01348
(lev3lc) (nls,sdata,lev3lc) 4F6		01341
(levt2) (nls,sdata,levt2) 4F2		01323
(levt3) (nls,sdata,levt3) 4F3		01315
(levtab) (nls,sdata,levtab) 4F1		01352
(lhostn) (nls,sdata,lhostn) 4A3		01320
(msgfrk) (nls,sdata,msgfrk) 5A3		01338
(msglck) (nls,sdata,msglck) 5A1		01354
(msgtim) (nls,sdata,msgtim) 5A2		01343
(nojournalf) (nls,sdata,nojournalf) 4A4		01333
(nonetworkf) (nls,sdata,nonetworkf) 4A6		01327
(omode) (nls,sdata,omode) 6B1A		01330
(p1) (nls,sdata,p1) 4C1		01307
(p2) (nls,sdata,p2) 4C2		01335
(psireg) (nls,sdata,psireg) 4B3		01317
(psirglast) (nls,sdata,psirglast) 4B4		01339
(sav40) (nls,sdata,sav40) 4G1		01346
(savchntab) (nls,sdata,savchntab) 4F8		01350
(svacl) (nls,sdata,svacl) 4G2		01344
(svacs) (nls,sdata,svacs) 4G3		01314
(svacse) (nls,sdata,svacse) 4G4		01309
(tenex) (nls,sdata,tenex) 4A2		01325
(timra1) (nls,sdata,timra1) 4H3		01349
(timra2) (nls,sdata,timra2) 4H4		01311
(timra3) (nls,sdata,timra3) 4H5		01334
(timra4) (nls,sdata,timra4) 4H6		01345
(timrproc) (nls,sdata,timrproc) 4H2		01337
(timrset) (nls,sdata,timrset) 4H1		01324
(tops20flag) (nls,sdata,tops20flag) 4A5		01336
(trpcnt) (nls,sdata,trpcnt) 4F9		01347
(userstr) (nls,sdata,userstr) 3A1		01332
		03913

(cdr) (nls,select,cdr) 1C1	0224
(flndr) (nls,select,flndr) 1C11	0215
(gaddrlit) (nls,select,gaddrlit) 1E2	0213
(getlid) (nls,select,getlid) 1B1D	0220
(getaddr) (nls,select,getaddr) 1E1	0196
(getbug) (nls,select,getbug) 1B1G	0202
(getdae) (nls,select,getdae) 1B1H	0209
(getid) (nls,select,getid) 1B1C	0207
(getidlist) (nls,select,getidlist) 1B1E	0221
(getlit) (nls,select,getlit) 1B1B	0222
(getwindow) (nls,select,getwindow) 1B1I	0197
(grpdr) (nls,select,grpdr) 1B2B	0208
(iddr) (nls,select,iddr) 1C8	0199
(identquery) (nls,select,identquery) 1B1F	0211
(idldr) (nls,select,idldr) 1C9	0200
(idr) (nls,select,idr) 1C2	0204
(ldr) (nls,select,ldr) 1C12	0206
(littolnk) (nls,select,littolnk) 1E3	0210
(ndr) (nls,select,ndr) 1C4	0217
(plxdr) (nls,select,plxdr) 1B2C	0218
(plxset) (nls,select,plxset) 1D1	0201
(stdr) (nls,select,stdr) 1B2A	0212
(tdr) (nls,select,tdr) 1C3	0216
(vdr) (nls,select,vdr) 1C10	0203
(wdr) (nls,select,wdr) 1C5	0205
(xconfirm) (nls,select,xconfirm) 1B5A	0214
(xlevadj) (nls,select,xlevadj) 1B3A	0198
(xselect) (nls,select,xselect) 1B1A	0219
(xviewspecs) (nls,select,xviewspecs) 1B4A	0223

03914

(ihand) (nls,seqfil,)		03333
(hndacc) (nls,seqfil,)		03341
(blsfhdr) (nls,seqfil,blsfhdr) 3L		03321
(getsbn) (nls,seqfil,getsbn) 6W		03348
(hiop) (nls,seqfil,hiop) 4E		03343
(hmapin) (nls,seqfil,hmapin) 6E		03346
(hmapout) (nls,seqfil,hmapout) 6C		03349
(inseq) (nls,seqfil,inseq) 4A		03325
(inseq1) (nls,seqfil,inseq1) 4D		03329
(inseqn) (nls,seqfil,inseqn) 4C		03347
(isread) (nls,seqfil,isread) 4B		03331
(opinit) (nls,seqfil,opinit) 6V		03326
(oprchr) (nls,seqfil,oprchr) 6G		03336
(oprqid) (nls,seqfil,oprqid) 6S		03312
(oprqc) (nls,seqfil,oprqc) 6R		03344
(oprqt) (nls,seqfil,oprqt) 6U		03350
(oprqtc) (nls,seqfil,oprqtc) 6Q		03322
(oprqps) (nls,seqfil,oprqps) 6J		03335
(oprhdf) (nls,seqfil,oprhdf) 6Q		03324
(oprint) (nls,seqfil,oprint) 6T		03323
(oprlev) (nls,seqfil,oprlev) 6L		03319
(oprnst) (nls,seqfil,oprnst) 6H		03327
(oprrst) (nls,seqfil,oprrst) 6K		03313
(oprsid) (nls,seqfil,oprsid) 6P		03351
(oprsig) (nls,seqfil,oprsig) 6N		03352
(oprsvc) (nls,seqfil,oprsvc) 6M		03339
(oprtxt) (nls,seqfil,oprtxt) 6I		03314
(opseqf) (nls,seqfil,opseqf) 3F		03334
(opssig) (nls,seqfil,opssig) 3G		03318
(outasm) (nls,seqfil,outasm) 3E		03328
(outjm) (nls,seqfil,outjm) 3C		03315
(outqp) (nls,seqfil,outqp) 3B		03330
(outseq) (nls,seqfil,outseq) 3A		03317
(pager) (nls,seqfil,pager) 3D		03320
(pmfmap) (nls,seqfil,pmfmap) 6D		03332
(prgetst) (nls,seqfil,prgetst) 6F		03342
(processor) (nls,seqfil,processor) 6A		03340
(sfmacpt) (nls,seqfil,sfmacpt) 3J		03345
(sfnextst) (nls,seqfil,sfnextst) 3H		03337
(sfputst) (nls,seqfil,sfputst) 3I		03338
(sftrln) (nls,seqfil,sftrln) 3K		03316
		03915

(closeseq) (nls,seqgen,closeseq) 3C	03061
(cpysw) (nls,seqgen,cpysw) 4A	03075
(fechnd) (nls,seqgen,fechnd) 4K	03066
(fechnm) (nls,seqgen,fechnm) 4L	03084
(fechno) (nls,seqgen,fechno) 4I	03074
(fechnp) (nls,seqgen,fechnp) 4J	03083
(fechsig) (nls,seqgen,fechsig) 4N	03077
(fechux) (nls,seqgen,fechux) 4M	03080
(getlev) (nls,seqgen,getlev) 4H	03064
(getsgw) (nls,seqgen,getsgw) 3K	03068
(openseq) (nls,seqgen,openseq) 3B	03072
(pjsep) (nls,seqgen,pjsep) 4C	03081
(pjseqq) (nls,seqgen,pjseqq) 4B	03065
(relsgw) (nls,seqgen,relsgw) 3L	03069
(reslev) (nls,seqgen,reslev) 4D	03082
(send) (nls,seqgen,send) 3H	03076
(seqend) (nls,seqgen,seqend) 3G	03070
(seqgen) (nls,seqgen,seqgen) 3D	03062
(seqnxt) (nls,seqgen,seqnxt) 3F	03078
(sport) (nls,seqgen,sport) 3I	03071
(sqinit) (nls,seqgen,sqinit) 3J	03060
(sseqgen) (nls,seqgen,sseqgen) 3E	03079
(stpos) (nls,seqgen,stpos) 4G	03063
(stvect) (nls,seqgen,stvect) 4E	03073
(stvmod) (nls,seqgen,stvmod) 4F	03067
	03916

(gtintmsg) (nls,sintnls,gtintmsg) 2B	02852
(hostnumber) (nls,sintnls,hostnumber) 1C	02845
(idfirmuser) (nls,sintnls,idfirmuser) 2E	02851
(inittimer) (nls,sintnls,inittimer) 2D	02846
(movsym) (nls,sintnls,movsym) 3	02850
(setabs) (nls,sintnls,setabs) 2C	02849
(setinterrupts) (nls,sintnls,setinterrupts) 2A	02848
(tenexverno) (nls,sintnls,tenexverno) 1D	02847
	03917

(blkfree) (nls,srecords,blkfree) 3A5	02579
(blklength) (nls,srecords,blklength) 3A3	02600
(blkprevtree) (nls,srecords,blkprevtree) 3A6	02507
(cackf) (nls,srecords,cackf) 3B2	02562
(capst) (nls,srecords,capst) 3B3	02523
(chnul) (nls,srecords,chnul) 3C1	02639
(chr0) (nls,srecords,chr0) 3C6	02637
(chr1) (nls,srecords,chr1) 3C5	02500
(chr2) (nls,srecords,chr2) 3C4	02464
(chr3) (nls,srecords,chr3) 3C3	02472
(chr4) (nls,srecords,chr4) 3C2	02439
(daauxiliary) (nls,srecords,daauxiliary) 3D64	02535
(dabneighbor) (nls,srecords,dabneighbor) 3D70	02557
(dabottom) (nls,srecords,dabottom) 3D44	02460
(dacacode) (nls,srecords,dacacode) 3D57	02622
(dacnt) (nls,srecords,dacnt) 3D12	02432
(dacol) (nls,srecords,dacol) 3D25	02527
(dacrow) (nls,srecords,dacrow) 3D22	02524
(dacsiz) (nls,srecords,dacsiz) 3D13	02630
(dacsp) (nls,srecords,dacsp) 3D10	02481
(daempty) (nls,srecords,daempty) 3D51	02631
(daaxis) (nls,srecords,daaxis) 3D50	02592
(dafont) (nls,srecords,dafont) 3D34	02584
(dafrozen) (nls,srecords,dafrozen) 3D65	02589
(dafrz1) (nls,srecords,dafrz1) 3D16	02605
(dahandle) (nls,srecords,dahandle) 3D37	02502
(dahinc) (nls,srecords,dahinc) 3D28	02546
(daind) (nls,srecords,daind) 3D18	02506
(dal) (nls,srecords,dal) 3D72A1	02465
(daleft) (nls,srecords,daleft) 3D40	02447
(dalftjst) (nls,srecords,dalftjst) 3D21	02485
(dalhandle) (nls,srecords,dalhandle) 3D60	02448
(dalink) (nls,srecords,dalink) 3D36	02548
(dalneighbor) (nls,srecords,dalneighbor) 3D67	02612
(dalsidb) (nls,srecords,dalsidb) 3D61	02488
(dalsrt) (nls,srecords,dalsrt) 3D46	02534
(dalssup) (nls,srecords,dalssup) 3D38	02511
(dalsz) (nls,srecords,dalsz) 3D47	02461
(damax) (nls,srecords,damax) 3D72A2	02596
(damcol) (nls,srecords,damcol) 3D27	02521
(damind) (nls,srecords,damind) 3D19	02606
(damrow) (nls,srecords,damrow) 3D24	02564
(danocs) (nls,srecords,danocs) 3D14	02457
(danohi) (nls,srecords,danohi) 3D31	02624
(dapstf) (nls,srecords,dapstf) 3D48	02426
(dapvs) (nls,srecords,dapvs) 3D6	02577
(dapvs2) (nls,srecords,dapvs2) 3D8	02526
(daright) (nls,srecords,daright) 3D41	02542
(darneighbor) (nls,srecords,darneighbor) 3D68	02487
(daseq) (nls,srecords,daseq) 3D66	02627
(dasgcs) (nls,srecords,dasgcs) 3D15	02463
(dasghi) (nls,srecords,dasghi) 3D33	02580
(dashrinkcnt) (nls,srecords,dashrinkcnt) 3D49	02609
(dasuppress) (nls,srecords,dasuppress) 3D71	02425
(datab0) (nls,srecords,datab0) 3D53	02633
(datab1) (nls,srecords,datab1) 3D54	02489

(datab2) (nls,srecords,datab2) 3D55	02525
(datneighbor) (nls,srecords,datneighbor) 3D69	02450
(datop) (nls,srecords,datop) 3D43	02626
(daukeycod) (nls,srecords,daukeycod) 3D63	02436
(dausqcod) (nls,srecords,dausqcod) 3D58	02565
(davinc) (nls,srecords,davinc) 3D30	02617
(davspc2) (nls,srecords,davspc2) 3D4	02608
(davspec) (nls,srecords,davspec) 3D2	02423
(dlgacc) (nls,srecords,dlgacc) 2B2	02441
(dlgars) (nls,srecords,dlgars) 2B3	02493
(dlgart) (nls,srecords,dlgart) 2B5	02556
(dlgbyt) (nls,srecords,dlgbyt) 2B6	02613
(dlgdar) (nls,srecords,dlgdar) 2B4	02480
(dlgdcr) (nls,srecords,dlgdcr) 2B7	02536
(dlgddm) (nls,srecords,dlgddm) 2B9	02619
(dlgdfr) (nls,srecords,dlgdfr) 2B11	02458
(dlgdlt) (nls,srecords,dlgdlt) 2B8	02443
(dlgdm) (nls,srecords,dlgdm) 2B10	02479
(dlgdov) (nls,srecords,dlgdov) 2B13	02530
(dlgdrd) (nls,srecords,dlgdrd) 2B15	02583
(dlgdwr) (nls,srecords,dlgdwr) 2B16	02513
(dlglwr) (nls,srecords,dlglwr) 2B12	02601
(dlgprt) (nls,srecords,dlgprt) 2B14	02558
(dlgrvr) (nls,srecords,dlgrvr) 2B1	02610
(dliacc) (nls,srecords,dliacc) 2A3	02510
(dliars) (nls,srecords,dliars) 2A4	02437
(dliart) (nls,srecords,dliart) 2A5	02499
(dlibyt) (nls,srecords,dlibyt) 2A9	02519
(dlidfr) (nls,srecords,dlidfr) 2A7	02578
(dlidlt) (nls,srecords,dlidlt) 2A1	02494
(dlidmt) (nls,srecords,dlidmt) 2A6	02522
(dlifrf) (nls,srecords,dlifrf) 2A2	02587
(dlilwr) (nls,srecords,dlilwr) 2A8	02453
(dlimis) (nls,srecords,dlimis) 2A10	02501
(dlinex) (nls,srecords,dlinex) 2A15	02572
(dlinrw) (nls,srecords,dlinrw) 2A11	02538
(dlinvr) (nls,srecords,dlinvr) 2A14	02561
(dliprt) (nls,srecords,dliprt) 2A12	02476
(dliisiz) (nls,srecords,dliisiz) 2A13	02440
(dlitar) (nls,srecords,dlitar) 2A16	02563
(dlitcr) (nls,srecords,dlitcr) 2A17	02482
(dlitdm) (nls,srecords,dlitdm) 2A18	02475
(dlitov) (nls,srecords,dlitov) 2A19	02614
(dlitrd) (nls,srecords,dlitrd) 2A20	02431
(dlitwr) (nls,srecords,dlitwr) 2A21	02599
(dlsacc) (nls,srecords,dlsacc) 2C2	02477
(dlsalp) (nls,srecords,dlsalp) 2C3	02623
(dlsart) (nls,srecords,dlsart) 2C4	02445
(dlsbyt) (nls,srecords,dlsbyt) 2C6	02541
(dlsdfr) (nls,srecords,dlsdfr) 2C11	02560
(dlsdlt) (nls,srecords,dlsdlt) 2C8	02466
(dlsdmt) (nls,srecords,dlsdmt) 2C10	02505
(dlslen) (nls,srecords,dlslen) 2C13	02575
(dlslwr) (nls,srecords,dlslwr) 2C12	02433
(dlsnac) (nls,srecords,dlsnac) 2C14	02544
(dlsnrd) (nls,srecords,dlsnrd) 2C15	02427

(dlsnwr)	(nls,srecords,dlsnwr)	2C16	02607
(dlsvr)	(nls,srecords,dlsvr)	2C1	02632
(dlssiz)	(nls,srecords,dlssiz)	2C19	02559
(dlstar)	(nls,srecords,dlstar)	2C5	02549
(dlstcr)	(nls,srecords,dlstcr)	2C7	02469
(dlstdm)	(nls,srecords,dlstdm)	2C9	02462
(dlstov)	(nls,srecords,dlstov)	2C17	02602
(dlstrd)	(nls,srecords,dlstrd)	2C18	02621
(dlstwr)	(nls,srecords,dlstwr)	2C20	02590
(ercode)	(nls,srecords,ercode)	3N1	02539
(ercondir)	(nls,srecords,ercondir)	3N4	02566
(erdata)	(nls,srecords,erdata)	3N19	02430
(erdatime)	(nls,srecords,erdatime)	3N8	02595
(erdlen)	(nls,srecords,erdlen)	3N18	02514
(erfns1)	(nls,srecords,erfns1)	3N12	02537
(erfns2)	(nls,srecords,erfns2)	3N13	02498
(erfns3)	(nls,srecords,erfns3)	3N14	02446
(erfns4)	(nls,srecords,erfns4)	3N15	02597
(erfns5)	(nls,srecords,erfns5)	3N16	02533
(erfns6)	(nls,srecords,erfns6)	3N17	02497
(erhost)	(nls,srecords,erhost)	3N2	02428
(erjobn)	(nls,srecords,erjobn)	3N6	02518
(erline)	(nls,srecords,erline)	3N5	02569
(erlpbaudf)	(nls,srecords,erlpbaudf)	3N9	02554
(erlptype)	(nls,srecords,erlptype)	3N10	02567
(erprom)	(nls,srecords,erprom)	3N11	02570
(ertipn)	(nls,srecords,ertipn)	3N7	02586
(eruser)	(nls,srecords,eruser)	3N3	02529
(frexis)	(nls,srecords,frexis)	3E2	02444
(frhexis)	(nls,srecords,frhexis)	3F3	02435
(frhlast)	(nls,srecords,frhlast)	3F2	02582
(frhtop)	(nls,srecords,frhtop)	3F1	02571
(frrelen)	(nls,srecords,frrelen)	3E3A	02509
(frrhlen)	(nls,srecords,frrhlen)	3F4A	02574
(frsrring)	(nls,srecords,frsrring)	3E1	02551
(hstnmf)	(nls,srecords,hstnmf)	3G3	02470
(hstnmi)	(nls,srecords,hstnmi)	3G1	02552
(hstnmn)	(nls,srecords,hstnmn)	3G2	02449
(lsrtl)	(nls,srecords,lsrtl)	3H22A	02508
(oc2qtr)	(nls,srecords,oc2qtr)	3O2	02553
(ochar1)	(nls,srecords,ochar1)	3O6	02540
(ochar2)	(nls,srecords,ochar2)	3O5	02504
(ochar3)	(nls,srecords,ochar3)	3O4	02620
(ocjnk1)	(nls,srecords,ocjnk1)	3O1	02635
(ocjnk2)	(nls,srecords,ocjnk2)	3O3	02603
(rtl1sg)	(nls,srecords,rtl1sg)	3H21	02484
(rtbpe)	(nls,srecords,rtbpe)	3H2	02550
(rtbps)	(nls,srecords,rtbps)	3H1	02634
(rtcbug)	(nls,srecords,rtcbug)	3H15	02456
(rtccnt)	(nls,srecords,rtccnt)	3H9	02483
(rtcsiz)	(nls,srecords,rtcsiz)	3H20	02451
(rtexis)	(nls,srecords,rtexis)	3H12	02545
(rtfont)	(nls,srecords,rtfont)	3H14	02503
(rthinc)	(nls,srecords,rthinc)	3H10	02467
(rtlev)	(nls,srecords,rtlev)	3H8	02515
(rtlplink)	(nls,srecords,rtlplink)	3H17	02468

(rtlsid) (nls,srecords,rtlsid) 3H11	02492
(rtnew) (nls,srecords,rtnew) 3H19	02591
(rtpartst) (nls,srecords,rtpartst) 3H18	02638
(rtsrce) (nls,srecords,rtsrce) 3H13	02473
(rtstid) (nls,srecords,rtstid) 3H3	02547
(rtx1) (nls,srecords,rtx1) 3H4	02496
(rtx2) (nls,srecords,rtx2) 3H5	02478
(rty) (nls,srecords,rty) 3H7	02455
(srcc) (nls,srecords,srcc) 3I2	02452
(srexix) (nls,srecords,srexix) 3I3	02636
(srhexix) (nls,srecords,srhexix) 3J3	02611
(srhfileno) (nls,srecords,srhfileno) 3J2	02604
(srhfname) (nls,srecords,srhfname) 3J1	02625
(srhlast) (nls,srecords,srhlast) 3J5	02593
(srhtop) (nls,srecords,srhtop) 3J4	02616
(srpsid) (nls,srecords,srpsid) 3I1	02628
(srrelen) (nls,srecords,srrelen) 3I6A	02438
(srrhlen) (nls,srecords,srrhlen) 3J6A	02491
(srvs1) (nls,srecords,srvs1) 3I4	02474
(srvs2) (nls,srecords,srvs2) 3I5	02454
(sysblength) (nls,srecords,sysblength) 3A4	02490
(tmchr1) (nls,srecords,tmchr1) 3M6	02512
(tmchr2) (nls,srecords,tmchr2) 3M5	02555
(tmchr3) (nls,srecords,tmchr3) 3M4	02618
(tmchr4) (nls,srecords,tmchr4) 3M3	02459
(tmchr5) (nls,srecords,tmchr5) 3M2	02517
(tmchr6) (nls,srecords,tmchr6) 3M1	02568
(vsatlf) (nls,srecords,vsatlf) 3K15	02576
(vsblkf) (nls,srecords,vsblkf) 3K10	02581
(vsbrof) (nls,srecords,vsbrof) 3K8	02520
(vscakf) (nls,srecords,vscakf) 3K18	02429
(vscapf) (nls,srecords,vscapf) 3K6	02531
(vsdaft) (nls,srecords,vsdaft) 3K21	02598
(vsfrzf) (nls,srecords,vsfrzf) 3K17	02594
(vsidtf) (nls,srecords,vsidtf) 3K19	02585
(vsindf) (nls,srecords,vsindf) 3K11	02532
(vslev) (nls,srecords,vslev) 3K2	02471
(vslevd) (nls,srecords,vslevd) 3K5	02442
(vsmkrf) (nls,srecords,vsmkrf) 3L1	02543
(vsnamf) (nls,srecords,vsnamf) 3K12	02615
(vspagf) (nls,srecords,vspagf) 3K20	02495
(vspixf) (nls,srecords,vspixf) 3K9	02588
(vsrind) (nls,srecords,vsrind) 3K16	02424
(vsrlev) (nls,srecords,vsrlev) 3K4	02486
(vssidf) (nls,srecords,vssidf) 3K22	02516
(vsstnf) (nls,srecords,vsstnf) 3K13	02434
(vsstnr) (nls,srecords,vsstnr) 3K14	02629
(vstrnc) (nls,srecords,vstrnc) 3K3	02528
(vsusqf) (nls,srecords,vsusqf) 3K7	02573

03918

(breakup) (nls,stgmt,breakup) 3E	03864
(freeblk) (nls,stgmt,freeblk) 3B	03856
(freelsrt) (nls,stgmt,freelsrt) 4E2	03861
(freestring) (nls,stgmt,freestring) 4A3	03855
(getarray) (nls,stgmt,getarray) 4A2	03858
(getblk) (nls,stgmt,getblk) 3A	03862
(getstring) (nls,stgmt,getstring) 4A1	03854
(linkup) (nls,stgmt,linkup) 3G	03863
(makezone) (nls,stgmt,makezone) 3C	03860
(maklsrt) (nls,stgmt,maklsrt) 4B1	03859
(replenish) (nls,stgmt,replenish) 3D	03853
(unlink) (nls,stgmt,unlink) 3F	03857
	03919

(nchk) (nls,syntax,)	0149
(nisedi) (nls,syntax,)	0144
(prtgrp) (nls,syntax,)	0133
(qdhelp) (nls,syntax,)	0137
(qthelp) (nls,syntax,)	0141
(subpro) (nls,syntax,)	0150
(subsen) (nls,syntax,)	0151
(subsup) (nls,syntax,)	0138
(subsyn) (nls,syntax,)	0147
(subuse) (nls,syntax,)	0136
(subxxx) (nls,syntax,)	0139
(cshmode) (nls,syntax,cshmode) 4AG2A	0134
(devopt) (nls,syntax,devopt) 3E1	0140
(jmpcoms) (nls,syntax,jmpcoms) 5A6C	0148
(prot1) (nls,syntax,prot1) 4L3	0143
(structure) (nls,syntax,structure) 3C1	0142
(text1) (nls,syntax,text1) 3B2	0135
(textent) (nls,syntax,textent) 3B1	0146
(zinsstatement) (nls,syntax,zinsstatement) 4AC2	0145

03920

(noexpand) (nls,syntbl,noexpand) 1A	03866
(prstunctions) (nls,syntbl,prsfuctions) 1I	03867
(prules) (nls,syntbl,prules) 1H	03868
(xroutines) (nls,syntbl,xroutines) 1G	03865
	03921

(newpag)	(nls,tsprt,newpag)	3D	0537
(printg)	(nls,tsprt,printg)	3A	0534
(prtype)	(nls,tsprt,prtype)	3B	0536
(putchr)	(nls,tsprt,putchr)	3C	0535
			03922

(cdfill)	(nls,undata,)		01438
(cred6)	(nls,undata,)		01417
(cred7)	(nls,undata,)		01435
(curnch)	(nls,undata,)		01418
(ec)	(nls,undata,)		01430
(rpib)	(nls,undata,)		01443
(abtframe)	(nls,undata,abtframe)	4A7	01441
(abtgt)	(nls,undata,abtgt)	4A6	01433
(cfnst1)	(nls,undata,cfnst1)	4B8	01442
(cfnst2)	(nls,undata,cfnst2)	4B9	01436
(cfnst3)	(nls,undata,cfnst3)	4B10	01434
(cfnst4)	(nls,undata,cfnst4)	4B11	01419
(comrot)	(nls,undata,comrot)	4A4	01422
(crdlay)	(nls,undata,crdlay)	4I3	01446
(curgrp)	(nls,undata,curgrp)	4E1	01420
(errwrk)	(nls,undata,errwrk)	4B7	01444
(es)	(nls,undata,es)	4B1	01428
(lfdlay)	(nls,undata,lfdlay)	4I4	01429
(lit)	(nls,undata,lit)	4B6	01437
(nlcret)	(nls,undata,nlcret)	4D2	01431
(nlcrlst)	(nls,undata,nlcrlst)	4D3	01440
(nlcrrt)	(nls,undata,nlcrrt)	4D1	01415
(nlstyp)	(nls,undata,nlstyp)	4D4	01424
(num)	(nls,undata,num)	4B2	01423
(rootframe)	(nls,undata,rootframe)	4A5	01416
(srchstid)	(nls,undata,srchstid)	4I5	01414
(state)	(nls,undata,state)	4A2	01432
(stateframe)	(nls,undata,stateframe)	4A3	01427
(stn)	(nls,undata,stn)	4B3	01445
(stn2)	(nls,undata,stn2)	4B4	01425
(stno)	(nls,undata,stno)	4B5	01439
(sysfs)	(nls,undata,sysfs)	3A1A	01426
(typnwa)	(nls,undata,typnwa)	4F1	01421
			03923

(rjtchr) (nls,update,)		0607
(cctbl) (nls,update,cctbl) 6A1		0621
(colmax) (nls,update,colmax) 11A3		0625
(dfnmdl) (nls,update,dfnmdl) 19A1		0588
(dfnmdr) (nls,update,dfnmdr) 19A2		0634
(enduop) (nls,update,enduop) 24A1		0627
(enlfstr) (nls,update,enlfstr) 17A1		0597
(rback2mode) (nls,update,rback2mode) 8A2		0617
(rbackmode) (nls,update,rbackmode) 8A1		0603
(fedind) (nls,update,fedind) 8A3		0587
(feedbk) (nls,update,feedbk) 8A4		0626
(frrsize) (nls,update,frrsize) 10A1		0606
(hrlldmode) (nls,update,hrlldmode) 9A1		0633
(hrlldsize) (nls,update,hrlldsize) 9A2		0596
(indcnt) (nls,update,indcnt) 11A4		0616
(inprompt) (nls,update,inprompt) 12A1		0598
(linmax) (nls,update,linmax) 11A2		0632
(nolevadj) (nls,update,nolevadj) 15A1		0586
(novspec) (nls,update,novspec) 15A2		0618
(pgssize) (nls,update,pgssize) 11A1		0579
(recog2mode) (nls,update,recog2mode) 14A2		0601
(recogmode) (nls,update,recogmode) 14A1		0631
(rtjtab) (nls,update,rtjtab) 22A2		0578
(spftab) (nls,update,spftab) 22A1		0592
(srrsize) (nls,update,srrsize) 10A2		0623
(stdtab) (nls,update,stdtab) 11A6		0615
(stdvsp) (nls,update,stdvsp) 13A1		0629
(stupstr) (nls,update,stupstr) 16A1		0608
(tpoffset) (nls,update,tpoffset) 11A5		0620
(tslshchars) (nls,update,tslshchars) 7A1		0583
(udpcolmax) (nls,update,udpcolmax) 21A2		0595
(udpsp1) (nls,update,udpsp1) 21A4		0602
(udpspace) (nls,update,udpspace) 21A1		0585
(udpwrapcol) (nls,update,udpwrapcol) 21A3		0614
(uoversion) (nls,update,uoversion) 5A2		0589
(uqpcolmax) (nls,update,uqpcolmax) 20A3		0594
(uqpindcnt) (nls,update,uqpindcnt) 20A4		0605
(uqplinmax) (nls,update,uqplinmax) 20A2		0622
(uqppgssize) (nls,update,uqppgssize) 20A1		0590
(uqpstdtab) (nls,update,uqpstdtab) 20A6		0610
(uqptpoffset) (nls,update,uqptpoffset) 20A5		0619
(userdata) (nls,update,userdata) 5A1		0630
(usys1) (nls,update,usys1) 18A1		0612
(usys10) (nls,update,usys10) 18A10		0584
(usys11) (nls,update,usys11) 18A11		0628
(usys12) (nls,update,usys12) 18A12		0613
(usys13) (nls,update,usys13) 18A13		0599
(usys14) (nls,update,usys14) 18A14		0581
(usys15) (nls,update,usys15) 18A15		0624
(usys2) (nls,update,usys2) 18A2		0611
(usys3) (nls,update,usys3) 18A3		0609
(usys4) (nls,update,usys4) 18A4		0604
(usys5) (nls,update,usys5) 18A5		0600
(usys6) (nls,update,usys6) 18A6		0591
(usys7) (nls,update,usys7) 18A7		0580
(usys8) (nls,update,usys8) 18A8		0593

GAS2, 14-Feb-79 22:37

< NLS, MINISYSGD.NLS.4, > 101

(usys9) (nls,updata,usys9) 18A9

0582
03924

GAS2, 14-Feb-79 22:37

< NLS, MINISYSGD.NLS.4, > 102

(uopsta) (nls,upover,)

0577
03925

(cint1) (nls,utilty,)	02812
(cint2) (nls,utilty,)	02819
(cint3) (nls,utilty,)	02790
(cint4) (nls,utilty,)	02795
(oldint) (nls,utilty,)	02815
(assnbit) (nls,utilty,assnbit) 8A1	02794
(bkc) (nls,utilty,bkc) 10G	02820
(bkw) (nls,utilty,bkw) 10H	02818
(bumpstr) (nls,utilty,bumpstr) 10E	02808
(chkbit) (nls,utilty,chkbit) 8E1	02782
(cltxt) (nls,utilty,cltxt) 10D	02802
(ctrlc) (nls,utilty,ctrlc) 5A	02789
(dassnbit) (nls,utilty,dassnbit) 8D1	02804
(deferr) (nls,utilty,deferr) 12A	02785
(disablw) (nls,utilty,disablw) 9B	02800
(doctrlc) (nls,utilty,doctrlc) 5B	02783
(enablw) (nls,utilty,enablw) 9A	02817
(enwheel) (nls,utilty,enwheel) 9C	02813
(err) (nls,utilty,err) 12C	02799
(filesc) (nls,utilty,filesc) 10I	02821
(flagut) (nls,utilty,flagut) 10A	02791
(flcpfse) (nls,utilty,flcpfse) 10L	02788
(flfehcl) (nls,utilty,flfehcl) 10M	02809
(flgpage) (nls,utilty,flgpage) 10B	02810
(fundngtxt) (nls,utilty,fundngtxt) 10C	02816
(idtst) (nls,utilty,idtst) 10W	02792
(ilsdefault) (nls,utilty,ilsdefault) 7A2A	02793
(ilspsi) (nls,utilty,ilspsi) 7A1A	02786
(makeptr) (nls,utilty,makeptr) 10F	02797
(notrapcc) (nls,utilty,notrapcc) 6B	02805
(nxtbit) (nls,utilty,nxtbit) 8C1	02814
(pr) (nls,utilty,pr) 7B	02798
(psi) (nls,utilty,psi) 7A	02801
(setbit) (nls,utilty,setbit) 8B1	02787
(stkovr) (nls,utilty,stkovr) 11B2A	02781
(sysovr) (nls,utilty,sysovr) 11B1A	02807
(trapcc) (nls,utilty,trapcc) 6A	02811
(txtdat) (nls,utilty,txtdat) 10K	02784
(werr) (nls,utilty,werr) 12B	02796
(xtrnam) (nls,utilty,xtrnam) 10J	02806
(xyzgwe) (nls,utilty,xyzgwe) 11B	02803
	03926

(ckdgm) (nls,verify,)		03309
(ckstrc) (nls,verify,ckstrc) 7A		03308
(crng) (nls,verify,crng) 6A		03311
(csdb) (nls,verify,csdb) 6B		03307
(vfmain) (nls,verify,vfmain) 4A		03310