MDK 13-MAY-74 14:37 22976

NIC Mailing Labels and Membership Lists: OLD NLS L10 User Program

```
FILE memlyst % L10 <KUDLICK>MEMLYST %
                                                                        1

% PROGRAM TO GENERATE Output Quickprint LISTINGS FOR GROUP
MEMBERSHIPS
 ... and/or!  an NLS FILE for use with CDC 6600 Mailing Label Program
%
                                                                        2

% global declarations %                                                 3

   DECLARE idstid, hdstid, tlstid, tpstid, new;                         3a

   DECLARE automatic, aflg, dflg;                                       3b

   DECLARE end, outstid, prtstid;                                       3c

   DECLARE labflg, memflg, lnct;                                        3d

   DECLARE STRING nothing[5], grpident[10], id[10], coor[10],
   title[100];                                                          3e

   DECLARE STRING err[300], entry[2000];                                3f

   DECLARE STRING blankcard = "
   ";
                                                                        3g

% main control %                                                        4

   (memlyst) PROCEDURE;                                                 4a

      LOCAL                                                             4a1

         i,   % loop index %                                            4a1a

         j,  % pgct for individuals %                                   4a1b

         orgp;  % TRUE when current ident is orgzn or group %           4a1c

      LOCAL TEXT POINTER tp1, tp2, tp3, tp4,  z1, z2;                   4a2

      LOCAL STRING memf[50], labf[50];                                  4a3

      LOCAL STRING inpstr[400]; % input collection %                    4a4

      LOCAL STRING nxtident[10], exp[300], noexp[300], name[300];       4a5

      LOCAL STRING grplist[6000];                                       4a6
```

```
    % initialization  %                                                   4a7
       automatic _ FALSE;                                                 4a7a
       *nothing* _ NULL;                                                  4a7b
       *inpstr* _ NULL;                                                   4a7c
    % open identfile %                                                    4a8
       IF NOT (idstid,stfile _ open(0, jflname($"identfile")))THEN
       err($" Unable to open IDENT FILE, ");                              4a8a
       ON SIGNAL ELSE                                                     4a8b
          BEGIN                                                           4a8b1
          close(idstid,stfile);                                           4a8b2
          RETURN;                                                         4a8b3
          END;                                                            4a8b4
    % get run parameters %                                                4a9
       % get idents from user %                                           4a9a
          crlf();                                                         4a9a1
          typeas ($" Manual Ident Entry Mode? (Y or N)  ");                4a9a2
          *xlit* _ NULL;                                                  4a9a3
          txtlit($xlit);                                                  4a9a4
          IF *xlit*[1] = 'Y  OR *xlit*[1] = 'y THEN                       4a9a5
             BEGIN % collect ident list %                                 4a9a5a
             crlf();                                                      4a9a5b
             typeas($"Idents =  ");                                        4a9a5c
             identlist($inpstr, idstid,stfile);                           4a9a5d
             END                                                          4a9a5e
          ELSE                                                            4a9a6
          BEGIN                                                    4a9a6a

          crlf();                                                  4a9a6b

          typeas($"Automatic Mode ..."):                           4a9a6c

          % set up the input string to specific values %           4a9a6d

             *inpstr* _ SP, "TIPG", SP, "USERG", SP, "SERVERG",
             SP, "ASSOCG";                                        4a9a6d1

          typeas($inpstr);                                         4a9a6e

          automatic _ TRUE;                                        4a9a6f

          END;
                                                                   4a9a6g

    % determine whether LABELS are desired %                         4a9b

       (what): crlf();                                              4a9b1

       typeas($" Do you want LABELS ?  (Y or N)  ");                4a9b2

       *xlit* _ NULL;                                               4a9b3

       txtlit($xlit);                                               4a9b4

       IF *xlit*[1] = 'Y  OR *xlit*[1] = 'y THEN labfil($labf)      4a9b5

       ELSE labflg _ FALSE;
                                                                    4a9b6

    % determine whether MEMLISTS are desired %                       4a9c

       crlf();                                                      4a9c1

       typeas($" Do you want MEM-LISTS ?  (Y or N)  ");             4a9c2

       *xlit* _ NULL;                                               4a9c3

       txtlit($xlit);                                               4a9c4

       IF *xlit*[1] = 'Y  OR *xlit*[1] = 'y THEN memfil($memf)      4a9c5

       ELSE memflg _ FALSE;
                                                                    4a9c6

    % should we proceed ? %                                          4a9d
```

```
         IF (NOT memflg AND NOT labflg) THEN                         4a9d1
            BEGIN                                                    4a9d1a
            crlf();                                                  4a9d1b
            typeas($" Say YES to LABELS or MEM-LISTS or both !");    4a9d1c
            GOTO what;                                               4a9d1d
            END;
                                                                     4a9d1e
   % initialization before getting idents from identfile %            4a10
      orgp _ TRUE; % for page eject when doing individ, not grp %     4a10a
      crlf();                                                         4a10b
      FIND SF(*inpstr*)^zi;                                           4a10c
      IF (memflg AND automatic) THEN                                  4a10d
         BEGIN  % get ready for title page %                         4a10d1
         *title* _ EOL, "       ", "NETWORK  INDIVIDUALS", EOL,
         EOL;                                                        4a10d2
         prtstid _ cis(prtstid, $blankcard, sucdir);                 4a10d3
         hdstid _ prtstid;                                           4a10d4
         lnct _ 1;                                                   4a10d5
         END;
                                                                     4a10d6
   % main processing loop %                                            4a11
      (mainloop):                                                     4a11a
      LOOP                                                            4a11b
         BEGIN                                                       4a11b1
      % initialization %                                              4a11c
         *entry* _ NULL;  % ident file entry %                       4a11c1
```

```
          *xlit* _ NULL;                                                  4a11c2

          *lit* _ NULL;                                                   4a11c3

          *coor* _ NULL; % grp coordinator ident, if any %
                                                                          4a11c4

       % get next ident from input string %                               4a11d

          IF NOT FIND  z1 > $(SP/',) ^z1 1$(LD/'=) ^z2 THEN EXIT
          LOOP;                                                           4a11d1

          *nxtident* _ z1 z2;                                             4a11d2

          z1[1] _ z2[1];                                                  4a11d3

          *grpident* _ NULL;                                              4a11d4

          *grplist* _ NULL;
                                                                          4a11d5

       % see if ident is valid %                                          4a11e

          IF NOT ckident($nxtident, $entry, idstid,stfile) THEN           4a11e1

             BEGIN                                                        4a11e1a

             *xlit* _ *nxtident*, " ... skipping this ident:
             invalid";                                                    4a11e1b

             typeas($xlit);                                               4a11e1c

             REPEAT LOOP;                                                 4a11e1d

             END;                                                         4a11e1e

          *xlit* _ *nxtident*;                                            4a11e2

          typeas($xlit);                                                  4a11e3

          IF NOT automatic THEN                                           4a11e4

             BEGIN                                                        4a11e4a

             IF deleted($entry) THEN                                      4a11e4b

                BEGIN                                                     4a11e4b1

                typeas($" ... SPECIAL (deleted)");                        4a11e4b2
          REPEAT LOOP;                                             4a11e4b3

          END;                                                     4a11e4b4

       END;                                                        4a11e4c

    crlf();
                                                                   4a11e5

  % see if ident is organization or group or individual %         4a11f

    IF orgrptst($entry,0) THEN   % TRUE if org/group %            4a11f1

       BEGIN                                                       4a11f1a

       *grpident* _ *nxtident*;                                    4a11f1b

       *grplist* _ *nxtident*;                                     4a11f1c

       orgp _ TRUE;                                                4a11f1d

       IF (memflg AND NOT automatic) THEN                          4a11f1e

          BEGIN  % get ready for new page on listing %            4a11f1e1

          getinam($entry, $title, 0, 0);                           4a11f1e2

          astruc($title);                                          4a11f1e3

          *title* _ EOL, "        ", *title*, Sp, "MEMBERSHIP
          LIST", EOL, EOL;                                         4a11f1e4

          prtstid _ cis(prtstid, $blankcard, sucdir);              4a11f1e5

          hdstid _ prtstid;                                        4a11f1e6

          lnct _ 1;                                                4a11f1e7

          % find group coordinator %                               4a11f1e8

             geticord($entry, $coor, 0,0);                        4a11f1e8a

          END;                                                     4a11f1e9

       % expand group or orgzn ident ...
         ... and build records for output file(s) %                4a11f1f

          iexpmdk ($grplist, idstid,stfile);                       4a11f1f1
          crlf();                                                   4a11f1g

          END % of group/org ident processing loop %
                                                                    4a11f1h

       ELSE  % individual ident  %                                  4a11f2

          BEGIN                                                     4a11f2a

          % create label info %                                     4a11f2b

             IF labflg THEN process($nothing);                      4a11f2b1

          % create formatted print line %                           4a11f2c

             IF memflg THEN                                         4a11f2c1

                BEGIN                                              4a11f2c1a

                IF orgp THEN  % start new output page %            4a11f2c1b

                   BEGIN                                          4a11f2c1b1

                   orgp _ FALSE;                                  4a11f2c1b2

                   (donew):                                       4a11f2c1b3

                   *title* _ ', SP, SP, EOL;                      4a11f2c1b4

                   prtstid _ cis(prtstid, $title, sucdir);        4a11f2c1b5

                   *title* _ SP, SP, " individual idents", EOL;
                                                                  4a11f2c1b6
                   prtstid _ cis(prtstid, $title, sucdir);        4a11f2c1b7

                   j _ 5;                                         4a11f2c1b8

                   END;                                           4a11f2c1b9

                j _ j - 1;                                         4a11f2c1c

                IF j < 0 THEN GOTO donew;                          4a11f2c1d

                % format entry for print lines %                   4a11f2c1e

                   list($entry, $nxtident);                       4a11f2c1e1

                END;                                               4a11f2c1f
```

```
            END; % of individual ident processing loop %
                                                                  4a11f2d
         % end of main processing loop %                          4a11g
            END; % of main loop %
                                                                  4a11g1
      % finished--clean up  %                                     4a12
         % sort and set page titles for arpanet directory %       4a12a
            IF (memflg AND automatic) THEN stlist();              4a12a1
         crlf();                                                  4a12b
         typeas($"Processing Finished Normally ");                4a12c
         IF labflg THEN                                           4a12d
            BEGIN                                                 4a12d1
            close(outstid,stfile);                                4a12d2
            astruc($labf);                                        4a12d3
            *labf* _ EOL, " labels file is ", *labf*;             4a12d4
            typeas($labf);                                        4a12d5
            END;                                                  4a12d6
         IF memflg THEN                                           4a12e
            BEGIN                                                 4a12e1
            close(prtstid,stfile);                                4a12e2
            astruc($memf);                                        4a12e3
            *memf* _ EOL, " mem-list file is ", *memf*;           4a12e4
            typeas($memf);                                        4a12e5
            END;                                                  4a12e6
         close(idstid,stfile);                                    4a12f
      RETURN END.
                                                                    4a13
(iexpmdk)PROC(delstr,idfnum); % get next individ ident for current
grp%                                                                   5
   LOCAL count;                                                       5a
   LOCAL TEXT POINTER ptr, z1, z2, s0, s1, s2;                        5b
   REF delstr;                                                        5c
   *delstr* _ *delstr*, ';;                                           5d
   makeptr(asrref(&delstr),$ptr);                                     5e
   intids(0);                                                         5f
   IF automatic THEN aflg _ TRUE ELSE aflg _ FALSE; %upon entry only%  5g
   LOOP                                                               5h
      BEGIN                                                          5h1
      % get next individual ident within current group/orgzn  %      5h2
         *entry* _ NULL;                                            5h2a
      % get full identfile info for this ident %                     5h3
         IF NOT getmdKids($ptr, $entry, 0, idfnum) THEN EXIT LOOP;  5h3a
         IF dflg THEN REPEAT LOOP;  % that ident was deleted %      5h3b
         getiid($entry, 0, $z1, $z2);                               5h3c
         *auxlit* _ " ", z1 z2;                                     5h3d
         typeas($auxlit);                                           5h3e
      % delete all control characters except EOL's %                 5h4
         FIND SF(*entry*) ^s0;                                      5h4a
         LOOP                                                       5h4b
            BEGIN                                                  5h4b1
            IF FIND s0 > ENDCHR THEN EXIT LOOP;                    5h4b2
```

```
          IF FIND s0 > [NLD] ^s2 ^s1 _s1 THEN                          5h4b3
             BEGIN                                                     5h4b3a
             count _ s1[1];                                            5h4b3b
             IF ((*entry*[count] IN [0B, 37B]) AND (*entry*[count]
             # EOL)) THEN ST s1 s2 _ NULL;                             5h4b3c
             s0[1] _ s2[1];                                            5h4b3d
             END                                                       5h4b3e
          ELSE EXIT LOOP;                                              5h4b4
          END;                                                         5h4b5
      % format statements for CDC 6600 mailing labels program %          5h5
         IF labflg THEN process(IF automatic THEN $nothing ELSE
         $grpident);                                                    5h5a
      % format statement for printed mem-list %                          5h6
         IF memflg THEN list($entry, $auxlit);                          5h6a
      END;                                                              5h7
    % sort membership list, but not now if in automatic mode %            5i
      IF (memflg AND NOT automatic) THEN stlist();                       5i1
    RETURN;                                                               5j
    END.                                                                  5k
                                                                          5l
(getmdkids) PROCEDURE (ptr, astr, infotype, idfnum);%expand grp%           6
    LOCAL expchr, gpstid;                                                 6a
    LOCAL TEXT POINTER idf, ide, tmpptr, srcptr, dstptr;                  6b
    LOCAL STRING idstr[20], infostr[500];                                 6c
    REF ptr, astr;                                                        6d
```

```
%reads an ident from pointer ptr into astring idstr, and then
calls ckident to get info on it.                                   6e

   IF infotype = 0, then it returns all of the info in astr, if
   infotype = 1, then the name only is returned.                   6e1

   Uses infostr as a work area%                                    6e2

expchr _ 0;                                                         6f

%first, read ident%                                                 6g

   LOOP                                                             6g1

      BEGIN                                                        6g1a

      CCPOS ptr;                                                   6g1b

      IF FIND SNP ^idf '; THEN                                     6g1c

         BEGIN                                                    6g1c1

         IF NOT popids(&ptr) THEN RETURN(FALSE); %no more idents% 6g1c2

         END                                                      6g1c3

      ELSE EXIT LOOP;                                              6g1d

      END;                                                         6g1e

   IF NOT FIND idf [ NP / '; / '( ]                                 6g2

      ^ptr _ptr ^ide _ide THEN                                     6g2a

         err($"Ident List Format Error");                         6g2a1

   FIND ptr (SNP '( [')] ^ptr);                                     6g3

   IF FIND idf ('&/'") ^idf  < CH > THEN expchr _ READC;            6g4

   *idstr* _ idf ide; %ident%                                       6g5

%Now get info, and check ident%                                     6h

   IF ckident($idstr, $infostr, idfnum : gpstid) THEN %return
   something%                                                      6h1

      BEGIN                                                        6h1a
```

```
         % set flag to show if ident was SPECIAL (i.e., deleted)
         ... but don't test TIPG SERVERG USERG or ASSOCG %              6h1b
            IF (deleted($infostr) AND NOT aflg) THEN                    6h1b1
               BEGIN                                                    6h1b1a
               dflg _ TRUE; % true if ident is deleted %                6h1b1b
               GOTO ea;                                                 6h1b1c
               END                                                      6h1b1d
            ELSE dflg _ FALSE;                                          6h1b2
            aflg _ FALSE;                                               6h1b3
         IF orgrptst($infostr, 0) THEN                                  6h1c
            BEGIN                                                       6h1c1
            expchr _ TRUE;                                              6h1c2
            IF expchr THEN                                              6h1c3
               BEGIN                                                    6h1c3a
               getimem($infostr, 0,  $dstptr, 0);                       6h1c3b
               IF FIND dstptr -EOL %membership list present% THEN       6h1c3c
                  BEGIN                                                 6h1c3c1
                  pushids(&ptr);                                        6h1c3c2
                  dstptr _ gpstid;                                      6h1c3c3
                  FIND dstptr "ptr;                                     6h1c3c4
                  RETURN(getmdkids(&ptr, &astr, infotype, idfnum));     6h1c3c5
                  END;                                                  6h1c3c6
               END;                                                     6h1c3d
            END;                                                        6h1c4
         END;                                                           6h1d
```

```
        %Now edit and append to astr %                                  6h2
           (ea): IF infotype = 1 THEN getifnf($infostr,  $infostr);     6h2a
           *astr* _ *astr*, *infostr*;                                  6h2b
     RETURN(TRUE) END,
                                                                        6i
  (stlist) PROCEDURE;          % sort group membership list %           7
     LOCAL cflg, i, k, m, pgct, pgmax;                                  7a
     LOCAL iprtstid, ihdstid, itlstid, kmax; %for brief ident list%     7b
     LOCAL TEXT POINTER s1,sx, t1,tx;                                   7c
     LOCAL STRING skipout[25], tytle[500], tpstrg[1000], tlstrg[1000];
                                                                        7d
     %  sort membership list  %                                         7e
        tpstid _ hdstid;                                                7e1
        tlstid _ prtstid;                                               7e2
        xgsog($hdstid, $tlstid);
                                                                        7e3
     % set up stid's for brief ident listing when in automatic mode %   7f
        IF automatic THEN                                               7f1
           BEGIN                                                        7f1a
           iprtstid _ getail(tlstid);                                   7f1b
           iprtstid _ cis(iprtstid, $blankcard, sucdir);                7f1c
           ihdstid _ iprtstid;                                          7f1d
           kmax _ 40;                                                   7f1e
           END;
                                                                        7f1f
     % initialize to insert page breaks for output quickprint usage %   7g
        new _ TRUE;                                                     7g1
```

```
      pgmax _ 6;
                                                                         7g2
   % count output entries per page, and insert page breaks %             7h
   % also, eliminate duplicate entries if in AUTOMATIC mode, %           7i
   %  produce the brief ident listing in automatic mode, %               7j
   % and, identify the group coord'tr when NOT in automatic mode %
                                                                         7k
   LOOP  % mainloop %                                                    7l
      BEGIN                                                              7l1
      % set up %                                                         7l2
      *skipout* _ ', "   ";                                              7l3
      tpstid _ cis(tpstid, $skipout, sucdir);    % force new page %      7l4
      tpstid _ cis(tpstid, $title, sucdir); % title is group name %      7l5
      pgct _ pgmax;                                                      7l6
      cflg _ FALSE;
                                                                         7l7
      % print explanation at start of group %                            7l8
      IF new THEN                                                        7l9
         BEGIN                                                           7l9a
         *tytle* _ EOL,  "   ENTRIES ARE IN THE FOLLOWING FORMAT:",
         EOL, EOL, "  Name                        NIC Ident   Network Mail
         Address", EOL, "    U.S. Mail Address
         Phone(s)", EOL,"    City, State, Zip
         Organization Ident", EOL, EOL;                                  7l9b
         tpstid _ cis(tpstid, $tytle, sucdir);                           7l9c
         tpstid _ cis(tpstid, $blankcard, sucdir);                       7l9d
         new _ FALSE;                                                    7l9e
         pgct _ pgct - 1;                                                7l9f
   END;
                                                                    719g
% allow several entries ("pgmax" value) per page %                  7110
   i _ pgct;                                                       7110a
LOOP % inner loop %                                                 7111
% initialization %                                                  7112
   BEGIN                                                           7112a
   tpstid _ getsuc(tpstid);                                        7112b
   IF tpstid = ihdstid THEN EXIT LOOP; %exit inner loop%           7112c
   FIND SF(tpstid) ^si;                                            7112d
   IF NOT FIND si > [EOL] ^sx THEN FIND SE(tpstid) ^sx;
                                                                   7112e
% when in automatic mode, eliminate duplicate entries and ...
output a brief list (idents and names) of each individ %            7113
   IF automatic THEN                                               7113a
      BEGIN                                                       7113a1
      % set up string for comparison with subsequent entry %      7113a2
         *tpstrg* _ si sx;                                       7113a2a
      % if next entry is equal to present one, delete it %        7113a3
      LOOP % innermost loop %                                     7113a4
         BEGIN                                                   7113a4a
         t1stid _ getsuc(tpstid);                                7113a4b
         IF t1stid = ihdstid THEN EXIT LOOP;                     7113a4c
         FIND SF(t1stid) ^t1;                                    7113a4d
         IF NOT FIND t1 > [EOL] ^tx THEN FIND SE(t1stid) ^tx;    7113a4e
         *t1strg* _ t1 tx;                                       7113a4f
```

```
          IF *tpstrg* # *tlstrg* THEN EXIT LOOP;                  7113a4g
          cds(tlstid);                                            7113a4h
          lnct _ lnct - 1;                                        7113a4i
          END;
                                                                  7113a4j
       % output ident and name using already-formatted data %      7113a5
          FIND s1 > 37$37CH "tx; % end of ident field %           7113a5a
          FIND tx < $NP $PT "t1 >; % start of ident field %       7113a5b
          *tlstrg* _ "  ", t1 tx, "         ";                    7113a5c
          *tlstrg* _ *tlstrg*[1 TO 8], s1 t1;                     7113a5d
          iprtstid _ cis(iprtstid, $tlstrg, sucdir);
                                                                  7113a5e
       END;
                                                                   7113a6
    % when NOT in automatic mode, identify group coordinator %       7114
       IF NOT automatic THEN                                        7114a
       BEGIN                                                       7114a1
       IF FIND sx < [2SP] "tx  $PT "t1 > THEN *tpstrg* _ t1 tx     7114a2
       ELSE *tpstrg* _ NULL;                                       7114a3
       FIND s1 > CH "sx;                                           7114a4
       IF *coor* = *tpstrg* THEN                                   7114a5
          BEGIN                                                   7114a5a
          ST s1 sx _ '*;                                          7114a5b
          cflg _ TRUE;                                            7114a5c
          END;                                                    7114a5d
       END;
                                                                   7114a6
```

```
        % keep count for insertion of page breaks %                      7115
           i _ i - 1;                                                    7115a
           IF i > 0 THEN REPEAT LOOP                                     7115b
           ELSE                                                          7115c
              % if a grp coord was found on this page, footnote it %    7115c1
              BEGIN                                                      7115c2
              IF (cflg AND NOT automatic) THEN                           7115c3
                 BEGIN                                                  7115c3a
                 *skipout* _ "* Group Coordinator";                     7115c3b
                 tpstid _ cis(tpstid, $skipout, sucdir);                7115c3c
                 cflg _ FALSE;                                          7115c3d
                 END;                                                   7115c3e
              EXIT LOOP;  % exit inner loop %                            7115c4
              END;                                                       7115c5
           END; % of inner loop %
                                                                         7115d
        % play this till group exhausted %                               7116
           lnct _ lnct - pgct;                                           7116a
           IF lnct > 0 THEN REPEAT LOOP                                  7116b
           ELSE EXIT LOOP; % exit main loop %                            7116c
        END; % of main loop %
                                                                         7117
     % update print stid  to tail of output, for start of next group %     7m
        IF NOT automatic THEN                                              7m1
           BEGIN                                                          7m1a
           prtstid,stpsid _ origin;                                       7m1b
```

```
         prtstid _ getail(getsub(prtstid));                                7m1c
         END;
                                                                           7m1d
   % footnote if a grp coord was found on last pg for this group %          7n
      IF (cflg AND NOT automatic) THEN                                      7n1
         BEGIN                                                             7n1a
         *skipout* _ "* Group Coordinator";                                7n1b
         prtstid _ cis(prtstid, $skipout, sucdir);                         7n1c
         END;
                                                                           7n1d
   % in automatic mode sort brief ident list & insert page breaks %         7o
      IF automatic THEN                                                     7o1
         BEGIN                                                             7o1a
         % sort ident list %                                               7o1b
            itlstid _ iprtstid;                                           7o1b1
            iprtstid _ ihdstid;                                           7o1b2
            xgsog($ihdstid, $itlstid);                                    7o1b3
         % initialize page and line counts %                               7o1c
            k _ 0;                                                        7o1c1
            m _ 0;                                                        7o1c2
            *skipout* _ ', SP, SP, EOL;                                   7o1c3
            *title* _ "   ", "NIC IDENTS", EOL;                           7o1c4
         % LOOP TO INSERT PAGE AND LINE BREAKS IN BRIEF IDENT LISTING
         %                                                                 7o1d
         LOOP                                                              7o1e
            BEGIN                                                         7o1e1
```

```
          % is new page needed %                                       7o1e2
             IF k = 0 THEN                                            7o1e2a
                BEGIN                                                7o1e2a1
                k _ kmax;                                            7o1e2a2
                iprtstid _ cis(iprtstid, $skipout, sucdir);          7o1e2a3
                iprtstid _ cis(iprtstid, $title, sucdir);            7o1e2a4
                END;                                                 7o1e2a5
          % is blank line needed (every five lines) %                  7o1e3
             IF m = 0 THEN                                            7o1e3a
                BEGIN                                                7o1e3a1
                m _ 5;                                               7o1e3a2
                iprtstid _ cis(iprtstid, $"      ", sucdir);         7o1e3a3
                END;                                                 7o1e3a4
          % keep track of where we are %                               7o1e4
             iprtstid _ getsuc(iprtstid);                             7o1e4a
             IF iprtstid.stpsid = origin THEN EXIT LOOP;              7o1e4b
             k _ k=1;                                                 7o1e4c
             m _ m=1;                                                 7o1e4d
          END; % of LOOP for page/line=break insertion %               7o1e5
       prtstid.stpsid _ origin;                                         7o1f
       prtstid _ getail(getsub(prtstid));                               7o1g
       END; % of processing brief ident listing %
                                                                        7o1h
   RETURN;                                                                7p
   END.
```

```
                                                                        7q

(process) PROC (groupid);     % output the cards for the entry in
"entry", inserting the group ident "groupid" if supplied. %              8

   LOCAL i;                                                             8a

   LOCAL TEXT POINTER current, tp1, tp2, t7, t8;                        8b

   LOCAL STRING name[300];                                              8c

   REF groupid;
                                                                        8d

   getinam($entry, $name, 0, 0); % put name in "name" %                 8e

   % DO LINE 1 (really, "card image" #1): NAME and GROUPID %            8f

      IF FIND > (SF(*name*) [',] ^tp1 ^tp2 _tp2) THEN                   8f1

         *name* _ SF(*name*) tp2, SP, tp1 SE(*name*);                   8f1a

      *xlit* _ *blankcard*;                                             8f2

      IF name.L > 24 THEN                                               8f3

         BEGIN                                                          8f3a

         *lit2* _ EOL, *name*, "     name too long, truncated to:";     8f3b

         typeas($lit2);                                                 8f3c

         FIND SF(*name*) ^t7;                                           8f3d

         FIND SE(*name*) ^t8;                                           8f3e

         LOOP                                                           8f3f

            BEGIN                                                       8f3f1

            IF name.L < 25 THEN EXIT LOOP                               8f3f2

            ELSE                                                        8f3f3

               BEGIN                                                    8f3f3a

               IF NOT FIND t8 < $PT 1$NP ^t8 > THEN FIND t7 > 24$24CH
               ^t8;                                                     8f3f3b
```

```
                name.L _ t8[i];                                         8f3f3c
                REPEAT LOOP;                                            8f3f3d
                END;                                                    8f3f3e
             END;                                                       8f3f4
        crlf();                                                         8f3g
        typeas($name);                                                  8f3h
        crlf();                                                         8f3i
        END;                                                            8f3j
     *xlit*[1 TO 1+name.L] _ *name*;                                    8f4
     IF NOT automatic THEN IF groupid.L THEN  *xlit*[30-groupid.L TO
     30] _ SP, *groupid*;                                               8f5
  % DO LINES 2 THRU 8 %                                                 8g
     end _ FALSE;                                                       8g1
     *lit2* _ NULL;                                                     8g2
     laddress($entry, $lit2 %dest%, 0,0, idstid,stfile);                8g3
     *lit2* _ *xlit*[1 TO 30], EOL, *lit2*;                             8g4
     astruc($lit2); % capitalize %                                      8g5
     FIND SF(*lit2*) ^current;                                          8g6
     FOR i _ 1 UP 1 UNTIL > 4 DO                                        8g7
        card($current, lit2.L);                                         8g7a
     IF NOT end THEN                                                    8g8
        BEGIN                                                           8g8a
        crlf();                                                         8g8b
        *xlit* _ *name*, " address label truncated ", EOL;              8g8c
        typeas($xlit);                                                  8g8d
```

```
         END;                                                          8g8e

      RETURN END,
                                                                         8h

(card) PROC (current, leng); % build a card image from input,
starting at textpointer "current", "leng" is length of input string,
Leaves "current" updated, Sets global "end" if applicable, %             9

   LOCAL column, difference;                                             9a

   LOCAL TEXT POINTER t1, t2, t7, t8;                                    9b

   REF current;
                                                                         9c

   *xlit* _ *blankcard*;                                                 9d

   IF NOT end THEN % more input %                                        9e

      BEGIN                                                             9e1

      column _ 1;                                                       9e2

      LOOP                                                              9e3

         BEGIN                                                         9e3a

         IF NOT FIND current > ([EOL] ^t1 ^t2 _t2) THEN                9e3b

            BEGIN                                                     9e3b1

            t2[1] _ leng + 1;                                         9e3b2

            t1[1] _ leng;                                             9e3b3

            end _ TRUE;                                               9e3b4

            END;                                                      9e3b5

         difference _ t2[1] - current[1];
                                                                       9e3c

         LOOP
           % This inner loop truncates the mail address to less than
         30 characters (ending just before a NP char), to fit on a
         label line; it saves the position of truncation to allow
         the address to be continued on the next line %
                                                                       9e3d
```

```
          BEGIN                                                    9e3d1
          IF difference < 31 THEN EXIT LOOP                        9e3d2
          ELSE                                                     9e3d3
             BEGIN                                                9e3d3a
             IF NOT FIND t2 < SPT 1SNP "t2 > "t1 THEN EXIT LOOP;   9e3d3b
             difference _ t2[1] - current[1];                     9e3d3c
             REPEAT LOOP;                                         9e3d3d
             END;                                                 9e3d3e
          END;
                                                                    9e3d4
       *xlit* [column TO column + difference] _ current t2;          9e3e
       current[1] _ t1[1];                                           9e3f
       column _ column + 30;                                         9e3g
       IF column > 31 OR current[1] >= leng OR end THEN EXIT LOOP;
       % done with a whole card %                                    9e3h
       END;                                                          9e3i
    END;                                                               9e4
 %truncate off any extra info in last field%                            9f
 *xlit*[61 TO 80] _ *blankcard*[61 TO 80];                              9g
 xlit.L _ 60;                                                           9h
 outstid _ cis(outstid, $xlit, sucdir);                                 9i
 RETURN END.
                                                                         9j
(list) PROCEDURE (entr, idnt);  % compose individ print line %          10
 REF entr, idnt;                                                       10a
 LOCAL i, j, diddle, phoneflag, addmor, orgflg;                        10b
LOCAL TEXT POINTER sf, adf, adx, ade, tf, tx, te, ti, hf, he, nf,
nx, ne;                                                           10c

LOCAL STRING out[2000], addr[400], lname[200], netaddr[200],
name[200], phonestr[100], affil[100];
                                                                  10d

% Initialization, %                                               10e

   FIND ^sf;
                                                                  10e1

% Get raw info for formatting, %                                  10f

   % Full name %                                                  10f1

      getinam($entr, $name, 0, 0);                                10f1a

      % Capitalize individual's last name %                       10f1b

         FIND SF(*name*) ^nf;                                     10f1b1

         FIND SE(*name*) ^ne;                                     10f1b2

         IF FIND nf > [","] ^nx THEN                              10f1b3

            *name* _ +nf nx, nx ne                               10f1b3a

         ELSE astruc($name);
                                                                 10f1b4

   % Network address %                                            10f2

      *netaddr* _ NULL;                                           10f2a

      getinma($entr, $netaddr, 0, 0);                             10f2b

      FIND SF(*netaddr*) ^he;                                     10f2c

      WHILE (FIND he [""] ^he ^hf _hf) DO ST hf he _ NULL;        10f2d

      IF getihost($entr, 0, $hf, $he) THEN                        10f2e

         *netaddr* _ *netaddr*, "@", hf he;                      10f2e1

      IF NOT netaddr.L THEN *netaddr* _ "Network Address Not
      Known";
                                                                  10f2f
```

```
% Full address, %                                                       10f3
   addr.L _ 0;                                                          10f3a
   IF laddress($entr, $addr, 0, 0, idstid,stfile) THEN                  10f3b
      BEGIN  % delete all control characters except EOL's %             10f3b1
      FIND SF(*addr*) "adf;                                             10f3b2
      LOOP                                                              10f3b3
         BEGIN                                                          10f3b3a
         IF FIND adf > ENDCHR THEN EXIT LOOP;                           10f3b3b
         IF FIND adf > [NLD] "ade "adx _adx THEN                        10f3b3c
            BEGIN                                                       10f3b3c1
            j _ adx[1];                                                 10f3b3c2
            IF ((*addr*[j] IN [0B, 37B]) AND (*addr*[j] # EOL))
            THEN ST adx ade _ NULL;                                     10f3b3c3
            adf[1] _ ade[1];                                            10f3b3c4
            END                                                         10f3b3c5
         ELSE EXIT LOOP;                                                10f3b3d
         END;                                                           10f3b3e
      END                                                               10f3b4
   ELSE *addr* _ " Address Not Known ";                                 10f3c
   FIND SF(*addr*) "adf SE(*addr*) "ade;
                                                                        10f3d
% Network Organization %                                                10f4
   getiorg ($entr, $affil, 0, 0);                                       10f4a
   astruc($affil);  %capitalize %                                       10f4b
   orgflg _ TRUE;
                                                                        10f4c
   % Build output string. %                                                10g

      % Initialization %                                                  10g1

         diddle _ 0; % USED TO GET CORRECT SPACING %                     10g1a

         *idnt* _ *idnt*, "  ";                                          10g1b

      % Full name and ident in 36 spaces. %                               10g2

         FIND ne ↑nx;                                                    10g2a

         WHILE nx[1] - nf[1] >= 36 - idnt.L DO FIND nx < [SP] ↑nx;       10g2b

            %  Will not work for names with elements longer than
            48-idnt.L characters not separated by spaces! %             10g2b1

         *out* _ *out*, nf nx, SP;                                       10g2c

         FOR i _ out.L + 1 UP UNTIL > diddle + 36 DO *out*[i] _ SP;      10g2d

         *out*[out.L-idnt.L TO out.L] _ *idnt*;                          10g2e

      % Rest of line is network address--  indent three characters on
      next line. %                                                        10g3

         *out* _ "  ", *out*, *netaddr*, EOL, "   "; % room for * %      10g3a

      % Put out rest of name if necessary %                               10g4

         WHILE (FIND > BETWEEN nx ne (SP [PT])) DO                       10g4a

            BEGIN                                                       10g4a1

            FIND nx > SP ↑nf ne ↑nx;                                    10g4a2

            WHILE nx[1] - nf[1] > 28 DO FIND nx < [SP] ↑nx;             10g4a3

            *out* _ *out*, "  ", nf nx, EOL, "   ";                     10g4a4

            END;                                                        10g4a5

      % Now next line:  Beginning of address plus phone number %          10g5
      FIND adf > ([EOL] ^adx _adx/ [ENDCHR] ^adx);                     10g5a
      WHILE adx[1] = adf[1] >= 32 DO FIND adx < [SP] ^adx;              10g5b
      diddle _ out,L;                                                   10g5c
      *out* _ *out*, adf adx;                                           10g5d
      FOR i _ out,L + 1 UP UNTIL > diddle + 34 DO *out*[i] _ SP;        10g5e
      addmor _ TRUE;                                                    10g5f
      % If phone put out first line of phone number %                   10g5g
         IF  phoneflag _ getiphone($entr, 0, $tf, $te) THEN            10g5g1
            BEGIN                                                      10g5g1a
            *phonestr* _ tf te;                                        10g5g1b
            FIND SF(*phonestr*) ^tf SE(*phonestr*) ^te ^tx;            10g5g1c
            WHILE tx[1] = tf[1] > 23 DO FIND tx < [SP] ^tx;            10g5g1d
            *out* _ *out*,  tf tx;                                     10g5g1e
            END;                                                       10g5g1f
         *out* _ *out*, EOL;
                                                                        10g5g2
   % The remaining lines %                                              10g6
      WHILE ((addmor _ (FIND adx > NP)) OR (phoneflag OR orgflg))
      DO                                                                10g6a
         BEGIN                                                         10g6a1
         diddle _ out.L;                                               10g6a2
         IF addmor THEN                                                10g6a3
            BEGIN                                                     10g6a3a
            IF FIND adx > EOL ^adx THEN                               10g6a3b
               BEGIN                                                 10g6a3b1
               *out* _ *out*, "   "; %don't indent any more%         10g6a3b2
            END                                                  10g6a3b3

         ELSE                                                    10g6a3c

            BEGIN                                                10g6a3c1

            *out* _ *out*, "        ";                           10g6a3c2

            FIND adx > NP ^adx;                                  10g6a3c3

            END;                                                 10g6a3c4

         IF (FIND adx ^adf > ( [EOL] ^adx _adx  / ade ^adx ))
         THEN                                                    10g6a3d

            BEGIN                                                10g6a3d1

            WHILE adx[1] = adf[1] > 33-(out,L-diddle) DO FIND
            adx < [SP] ^adx;                                     10g6a3d2

            *out* _ *out*, adf adx;                              10g6a3d3

            FOR i _ out,L + 1 UP UNTIL > diddle + 37 DO
            *out*[i] _ SP;                                       10g6a3d4

            END;                                                 10g6a3d5

         j _ 35;                                                 10g6a3e

         END                                                     10g6a3f

      ELSE j _ 37;  % j controls spacing when no more addr data
      but more phone or org data %                               10g6a4

      IF phoneflag THEN                                          10g6a5

         BEGIN                                                   10g6a5a

         IF FIND > BETWEEN tx te ( SP [PT] ) THEN                10g6a5b

            BEGIN                                                10g6a5b1

            FIND tx > SP ^tf te ^tx;                             10g6a5b2

            WHILE tx[1] = tf[1] > 17 DO FIND tx < [SP] ^tx;      10g6a5b3

            FOR i _ out,L + 1 UP UNTIL > diddle + 41 DO
            *out*[i] _ SP;                                       10g6a5b4
```

```
                    *out* _ *out*,  tf tx;                          10g6a5b5
                    END                                             10g6a5b6
                ELSE                                                10g6a5c
                    BEGIN                                           10g6a5c1
                    phoneflag _ FALSE;                              10g6a5c2
                    orgflg _ FALSE;                                 10g6a5c3
                    FOR i _ out.L + 1 UP UNTIL > diddle + j DO *out*[i]
                    _ SP;                                           10g6a5c4
                    *out* _ *out*, *affil*;                         10g6a5c5
                    END;                                            10g6a5c6
                END;                                                10g6a5d
            IF (orgflg AND NOT phoneflag) THEN                      10g6a6
                BEGIN                                               10g6a6a
                orgflg _ FALSE;                                     10g6a6b
                FOR i _ out.L + 1 UP UNTIL > diddle + j DO *out*[i] _
                SP;                                                 10g6a6c
                *out* _ *out*, *affil*;                             10g6a6d
                END;                                                10g6a6e
            *out* _ *out*, EOL;                                     10g6a7
            END;
                                                                    10g6a8
% output the text for current member of group being processed %     10h
    *out* _ *out*, EOL;                                             10h1
    lnct _ lnct + 1;                                                10h2
    prtstid _ cis(prtstid, $out, sucdir);
                                                                    10h3
RETURN;                                                             10i
```

```
   END.                                                                    10j

(deleted) PROCEDURE (strg);   % is this entry deleted (not active) %      11
   LOCAL TEXT POINTER s1, s2;                                             11a
   REF strg;                                                              11b
   IF getimcmnts($strg, 0, $s1, $s2) THEN                                 11c
      BEGIN                                                               11c1
      IF FIND s1 > ["SPECIAL"] THEN RETURN(TRUE);                         11c2
      END;                                                                11c3
   RETURN(FALSE);                                                         11d
   END.                                                                   11e

(labfil) PROCEDURE (strg);              % get output file for labels %    12
   LOCAL TEXT POINTER tp1, tp2;                                           12a
   REF strg;                                                              12b
   labflg _ TRUE;                                                         12c
   LOOP                                                                   12d
      BEGIN                                                               12d1
      crlf();                                                             12d2
      typeas($" Output file for Labels = ");                              12d3
      *strg* _ NULL;                                                      12d4
      txtlit($strg);                                                      12d5
      outstid _ origin;                                                   12d6
      IF NOT FIND SF(*strg*) >  ['.] THEN *strg* _ *strg*, ".NLS";        12d7
      % Open calls err if it doesn't find the file, %                     12d8
         ON SIGNAL ELSE GO TO newfile;                                    12d8a
```

```
IF outstid,stfile _ open(0, $strg) THEN                              12d9
   BEGIN                                                             12d9a
   typeas($"  (old file) CONFIRM ");                                 12d9b
   IF input() # CA THEN REPEAT LOOP;                                 12d9c
   % using old file - position to tail %                             12d9d
      outstid _ getail(getsub(outstid));                            12d9d1
   EXIT LOOP;                                                        12d9e
   END                                                               12d9f
ELSE                                                                 12d10
   BEGIN                                                            12d10a
   (newfile):                                                       12d10b
   typeas($" (new file) CONFIRM ");                                 12d10c
   IF input() # CA THEN REPEAT LOOP;                                12d10d
   ON SIGNAL ELSE NULL; %disarm all of them%                        12d10e
   % force NLS extension %                                          12d10f
      IF NOT FIND SF(*strg*) > '< 1$(LD/'=) '> 1$(LD/'=) ^tp1
      (['.] ^tp2 _ tp2 / TRUE ^tp2) THEN                           12d10f1
         BEGIN                                                     12d10f1a
         typeas($" cant understand filename:  ");                  12d10f1b
         typeas($strg);                                            12d10f1c
         REPEAT LOOP;                                              12d10f1d
         END;                                                      12d10f1e
      *strg* _ SF(*strg*) tp1, ".NLS", tp2 SE(*strg*);              12d10f2
   IF NOT outstid,stfile _ opwk(0, $strg) THEN                      12d10g
      typeas($" Bad File Name ") % (that's all it could be)  % 12d10g1
```

```
         ELSE EXIT LOOP;                                                 12d10h
         END;                                                            12d10i
      END;                                                               12d11
   RETURN;                                                               12e
   END.                                                                  12f
(memfil) PROCEDURE (strg);              % get output file for
mem-lists%                                                               13
   LOCAL TEXT POINTER tp1, tp2;                                          13a
   REF strg;                                                             13b
   memflg _ TRUE;                                                        13c
   LOOP                                                                  13d
      BEGIN                                                              13d1
      crlf();                                                            13d2
      typeas(s" Output file for Mem-Lists = ");                          13d3
      *strg* _ NULL;                                                     13d4
      txtlit($strg);                                                     13d5
      prtstid _ origin;                                                  13d6
      IF NOT FIND SF(*strg*) >  ['.] THEN *strg* _ *strg*, ".NLS";       13d7
      % Open calls err if it doesn't find the file. %                    13d8
         ON SIGNAL ELSE GO TO newfil;                                    13d8a
      IF prtstid,stfile _ open(0, $strg) THEN                            13d9
         BEGIN                                                           13d9a
         typeas(s"  (old file) CONFIRM ");                               13d9b
         IF input() # CA THEN REPEAT LOOP;                               13d9c
         % using old file - position to tail %                           13d9d
```

```
            prtstid _ getail(getsub(prtstid));                          13d9d1
         EXIT LOOP;                                                     13d9e
         END                                                            13d9f
      ELSE                                                              13d10
         BEGIN                                                          13d10a
         (newfil):                                                      13d10b
         typeas($" (new file) CONFIRM ");                               13d10c
         IF input() # CA THEN REPEAT LOOP;                              13d10d
         ON SIGNAL ELSE NULL; %disarm all of them%                      13d10e
         % force NLS extension %                                        13d10f
            IF NOT FIND SF(*strg*) > '< 1$(LD/'=) '> 1$(LD/'=) ^tp1
            ([';] ^tp2 _ tp2 / TRUE ^tp2) THEN                          13d10f1
               BEGIN                                                    13d10f1a
               typeas($" cant understand filename:  ");                 13d10f1b
               typeas($strg);                                           13d10f1c
               REPEAT LOOP;                                             13d10f1d
               END;                                                     13d10f1e
            *strg* _ SF(*strg*) tp1, ",NLS", tp2 SE(*strg*);            13d10f2
         IF NOT prtstid,stfile _ opwk(0, $strg) THEN                    13d10g
            typeas($" Bad File Name ") % (that's all it could be)  % 13d10g1
         ELSE EXIT LOOP;                                                13d10h
         END;                                                           13d10i
      END;                                                              13d11
   RETURN;                                                              13e
   END.
                                                                        13f
```

14

FINISH 15

NIC Mailing Labels and Membership Lists: OLD NLS L10 User Program

(J22976) 13-MAY-74 14:37; Title: Author(s): Michael D. Kudlick/MDK; Distribution: /MDK; Sub-Collections: SRI-ARC; Clerk: MDK; Origin: <KUDLICK>ADDR.NLS;16, 10-MAY-74 13:54 MDK ;

Quirks in the load busy file command

Kirk and Susan, You do not have to do an Update before editing a file which was loaded using the load busy file command. The trcky thing about files loaded with this command is tat they can only be updated if you are enabled or connected to the login-directory of whomever has it locked when you give the update command and that if you close the file (by loading another into the same window, whey yo get it back again you must use load busy file again. 1

Quirks in the load busy file command

(J22977) 13-MAY-74 14:12; Title: Author(s): Charles H. Irby/CHI;
Distribution: /KIRK(* info-only *) SRL(* info-only *) ;
Sub-Collections: SRI-ARC; Clerk: CHI;

People Who Should bee Able to Log in as Userguides and Energy

JMB,dvn,kirk,jhb ought to be able to log in as userguides;dvn,ecw,and ndm ought to be able to log in as energy. 1

People Who Should bee Able to Log in as Userguides and Energy

(J22978) 13-MAY-74 14:44; Title: Author(s): Dirk H. Van Nouhuys/DVN; Distribution: /JEW(* action *) JMB(* info-only *) JHB(* info-only *) ECW(* info-only *) NDM(* info-only *) KIRK(* info-only *) ; Sub-Collections: SRI-ARC DPCS DIRT; Clerk: DVN;

Possible Sources of Support

With MST gone I want to mention again two journal items I sent earlier in the year, 1

(ljournal,22823,) suggests a possible rout to support at NSF, Some one (perhaps me since I've been talking to him,) would have to phone Shuford to nail this down, 1a

(hjournal,7388,) suggests ways to go about selling utilty slots to scholars working in the humanities, It seems unlikey that it would ever amont to more than utitlity support, 1b

(J22980) 13-MAY-74 16:26; Title: Author(s): Dirk H. Van Nouhuys/DVN; Distribution: /DCE(* info-only *) RWW(* info-only *) JCN(* info-only *) JHB(* info-only *) EKM(* info-only *) HGL(* info-only *) ; Sub-Collections: SRI-ARC; Clerk: DVN;

Reminder for Charles about viewspecs ebe

the second e acts as a no-opt. Please let me know when this is fixed or if it cannot be fixed so I change my program accordingly. 1

Reminder for Charles about viewspecs ebe

(J22981) 13-MAY-74 16:35; Title: Author(s): Kirk E. Kelley/KIRK; Distribution: /BUGS(* action *) ; Sub-Collections: SRI-ARC BUGS; Clerk: KIRK;

JEAN and NETHELP

Dave,
The idents JEAN and NETHELP are ridiculous. Whoever put them in, put them in as an individual rather than a group. This puts certain constraints on how I can modify them. For one thing, as an "individual" these idents have to have a first and last name. So for JEAN, the name (as close as I can get it to the one you requested) is Using, Repository -- Repository being the first name, using the last. The same goes for nethelp which is Network Help, Repository. I could not take away hardcopy address as you requested, but did make the delivery online to directory using. (The identfile insists that it have a hardcopy address).
Marcia 1

JEAN and NETHELP

(J22982) 13-MAY-74 16:39; Title: Author(s): Marcia Lynn Keeney/MLK; Distribution: /DHC; Sub-Collections: SRI-ARC; Clerk: MLK;

Praise for the Feedback system

I think the way Susan has been handling the NLS feedback is excellent, This is something we should have had long ago and I hope it becomes an established procedure for the future, It is all too easy to take for granted necessities that aren't apparant because they are working as they should, Let's not let this one go by the wayside after the newness of the new NLS wears off, 1

Praise for the Feedback system

(J22983) 13-MAY-74 17:27; Title: Author(s): Kirk E, Kelley/KIRK;
Distribution: /FDBK(* action *) SRI-ARC(* info-only *) ;
Sub-Collections: SRI-ARC; Clerk: KIRK;

DCE 8-MAY-74 10:38 22785

Letter sending reports to G. Feller, Pepsico Transportation

Augmentation Research Center
Stanford Research Institute
Menlo Park, California 94025

George Feller
Pepsico Transportation
Pepsico Place
525 South Main
Tulsa, Oklahoma 74103

Dear Mr. Feller:

We are sending you the following documents, as requested by Helen Glucksman in a phone call with me on 11 April, 1974: 1

D. C. Engelbart and W. K. English, "A Research Center for Augmenting Human Intellect", AFIPS Proceedings, Fall Joint Computer Conference, 1968, Washington, D. C. (XDOC -- 3954.) 2

J. C. Norton, R. W. Watson, WORKSHOP UTILITY SERVICE FOR THE USE OF KNOWLEDGE WORKSHOP TECHNOLOGY, Technical Proposal to Bell Canada, SRI No. ISC 73-147, October 8, 1973 (Journal -- 19250.) 3

If you have any questions or would like to explore your interests further, please stay in touch. 4

Sincerely,

Douglas C. Engelbart
Augmentation Research Center

DCE/jml

Note for journal submission of this file: phone # for Mr. Feller is (918) 582-1900 x 414

Letter sending reports to G, Feller, Pepsico Transportation

(J22785) 8-MAY-74 10:38; Title: Author(s): Douglas C, Engelbart/DCE; Keywords: feller pepsico transportation; Sub-Collections: SRI-ARC; Clerk: JML;
Origin: <LEAVITT>PEPSI,NLS;3, 22-APR-74 09:12 JML ;

Bug with CR in links

At times, I found it desireable to put the "pretty" part of a link on the first line of a statement and then put the viewspecs on the next line to be hidden by line clipping. However, I now find these links do do not work. For example try taking this link:
<journal
:ebt> 1

Bug with CR in links

(J22987) 14-MAY-74 08:20; Title: Author(s): Kirk E, Kelley/KIRK; Distribution: /BUGS([ACTION]) ; Sub-Collections: SRI-ARC BUGS; Clerk: KIRK;

Small Capacity of Set Case

I recently created an NLS file from a sequential file, The statements were long, whatever length, Insert Sequential creates if nothing in the file indicates statement length, and were all caps, I wanted to change it to caps and lower case, When I tried to apply set casse for plex it replied "exceed capacity", When I tried to set case for a statement it replied exceed capapcity, I eventually discovered I could make set case work by limiting statement size to about half a screenful, This problem was quite reproducible and occured in old NLS as well as new, 1

Small Capacity of Set Case

(J22988) 14-MAY-74 08:32; Title: Author(s): Dirk H. Van Nouhuys/DVN; Distribution: /NEWNLS([ACTION]) ; Sub-Collections: SRI-ARC NEWNLS; Clerk: DVN;

(J22988) 14-MAY-74 08:32; Title: Author(s): Dirk H. Van Nouhuys/DVN; Distribution: /NEWNLS([ACTION]) ; Sub-Collections: SRI-ARC NEWNLS; Clerk: DVN;

brief write=up on arpanet written for Steve Miller (SRI) at his request

About the ARPANET
Advanced Research Projects Agency
Department of Defense
Computer Network 1

The ARPANET is a relatively new kind of digital communication system, a network employing wideband leased lines and message switching, The ARPANET provides for interconnecting dissimilar computers at widely separated sites via high=speed common carrier circuits, The development of the Network was sponsored by the Advanced Research Projects Agency (ARPA) of the U,S, Department of Defense, 1a

Each computer system, called a "Host", on the ARPANET is connected to the network through a communication interface called an "IMP", A maximum of four Hosts may be connected to the network through a single IMP, 1b

IMP's are small general purpose computers whose primary function is to forward data, No storage of data is accomplished at IMP's, only forwarding, Storage is accomplished at Hosts, 1c

Terminals may be connected to the network either through a Host, or directly through a special type of IMP called a "TIP", 1d

TIP's are IMP's to which has been added additional hardware and software that allows them to function as communication interfaces for terminals, Terminals may also be interfaced to the ARPANET through minicomputers with operating systems especialy implemented for this purpose, Two such systems based on PDP=11 computers are in common use, One of these, called "ANTS", was developed at the University of Illinois; the other, called "ELF", was developed at the University of Califoria at Santa Barbara, 1e

Identical "protocols", i,e, rigidly structured and rigidly ordered data communication sequences, are programmed in all IMP's, Similarly, all TIP's have identical protocols programmed in them, In addition, new protocols are continually under development to reflect new data communication needs, 1f

The major objectives in developing the network have been: 2

1) to develop highly reliable and economic digital communications among heterogeneous computer systems and terminals, and 2a

2) to permit sharing of computer resources (hardware, software,

and data) among geographically separated individuals, groups, and organizations. 2b

One of the major technological innovations used in the network is the concept of "packet switching". 3

Information to be transmitted over the ARPANET is segmented for transmission purposes into PACKETS (groups) of up to one thousand bits in length. Information transmitted over the network, and hence each packet transmitted over the network, may contain person-readable or machine-readable information or both. Any amount of data can be transmitted via the network; the data is simply segmented into the required number of packets. 3a

In transmitting packets over the network, transmission paths are not established in advance. Instead, each packet carries sufficient "switching" or routing information with it. This permits each IMP in turn (which as previously mentioned is a small computer) to choose the next leg of the packet's journey from source to destination. Packets are re-assembled at the destination. 3b

Information about the technical details of network technology exists in the professional literature, primarily in: 3c

1) May 1970 Spring Joint Computer Conference Proceedings
2) May 1972 Spring Joint Computer Conference Proceedings 3c1

The network currently consists of about 90 computer systems. 4

About half of these are medium to large scale computer systems referred to above as Hosts. The remaining computers function as direct interfaces to the network for a variety of computer terminals --- typewriter terminals, display terminals, high speed printers, and the like. 4a

More detailed information about the composition of the network -- its computer resources, the projects associated with it, and the uses made of it --- is presently considered to be U.S. proprieta. information. This information is available only to those receiving permission from the ARPA Information Processing Techniques Office. Their address is: 4b

Advanced Research Projects Agency
Information Processing Techniques Office
1400 Wilson Boulevard
Arlington, Virginia 22209 4b1

DR, J,C,R, LICKLIDER is currently the head of this Office, 4c

Foreign (non=U,S,) persons and organizations are not usually given access to network resources or network technological developments, 4d

There are three research computer systems at SRI currently attached to the network === two in the Artificial Intelligence (AI) Laboratory, one in the Augmentation Research Center (ARC), There are also other projects at SRI which are funded by ARPA but for which no separate computer system exists, and there may be other computer systems at SRI attached to the ARPANET in the future, 5

The AI Lab is currently headed by Dr, Peter Hart, The ARC is currently headed by Dr, Douglas Engelbart, Each of these men carries out research for ARPA in conjunction with their use of the ARPA Network, Information about the nature of these research efforts should be directed either to Mr, Bonnar Cox, Executiv Director of the Information Science and Engineering Division, to Dr, Hart or Dr, Engelbart, 5a

In addition, the ARC operates the Network Information Center (NIC), The NIC's function is to collect, maintain, publish, and distribute certain summary information about the computer hardware, computer software, and individual persons involved or associated with network=oriented research, 5b

To obtain information about the network from the NIC, one has to be a member of the network community, To be a member, one currently has to have permission from ARPA, or whomever they designate, The NIC does not have the authority to grant thi permission; it is strictly a publishing and distribution cen for ARPA, Once such permission is granted to an organizatio all members of that organization are entitled to receive the summary information that the NIC distributes, 5c

One final aspect of these SRI ARPA=related research efforts should be noted, as it has been the source of some confusion: The ARC has established a separate computer facility that is connected to the ARPA network and physically operated at and by a commercial organization, TYMSHARE Inc, This separate facility, sometimes known simply as the "Utility", exists primarily for the purpose of furthering the research efforts of ARC, and any inquiries about it should be directed to Dr, Engelbart, 5d brief write-up on arpanet written for Steve Miller (SRI) at his request

(J22989) 14-MAY-74 08:41; Title: Author(s): Michael D. Kudlick/MDK; Distribution: /DCE RWW JEW; Sub-Collections: SRI-ARC; Clerk: MDK; Origin: <KUDLICK>ARPANET.NLS;5, 9-MAY-74 13:41 MDK ;

MDK 14-MAY-74 08:50 22990

L10 User Program (OLD-NLS) for creating NLS file of Host Names/Addrs/etc

I've journalized this to make a permanent copy of it,
It's the old-nls version that does the right things,
For the new-nls version, ask David Maynard (DSM),

MDK 14=MAY=74 08:50 22990

L10 User Program (OLD=NLS) for creating NLS file of Host Names/Addrs/etc

```
FILE newhosts % L10 <kudlick>newhosts %
                                                                        1

% PROGRAM TO GENERATE NLS FILE TO BE RUN THROUGH OUTPUT ASSEMBLER FOR
ASCII HOSTNAMES FILE  %
                                                                        2

% default input file = <feinler>hostaddr=master %                       3

% default output file = <netinfo>hosts %
                                                                        4

   REGISTER r2 = 2;                                                     4a

   DECLARE dirno;                                                       4b

   DECLARE                                                              4c

      stid, % input file top plex, current host statement %            4c1

      endstid,   % origin of input file %                              4c2

      filf,      % TRUE if output is NEW FILE %                        4c3

      outstid;   % output file current tail stid %                     4c4

   DECLARE STRING dirname [40];                                         4d

                                                                        5

                                                                        5a

(newhosts) PROCEDURE;                                                   6

   LOCAL TEXT POINTER tp1, tp2, tp3, substid;                           6a

   LOCAL STRING inpstr[100];  %  used to hold attribute names %         6b

   LOCAL STRING statement[150];  % used for formatting output line %    6c

   LOCAL STRING attstr[20];
                                                                        6d

   %   *********   *********   *********   *********   *********   %
                                                                        6e
```

```
  %          INITIALIZATION;  SETTING UP ATTRIBUTE NAMES           %
                                                                          6f
     *inpstr* _ "NICKNAMES,", EOL;
                                                                         6f1
  %        to add a new attribute name, put it within quotes,      %      6g
  %          the final quote to be preceded AND followed by        %      6h
  %            a comma, as with the name NICKNAMES above     o     %
                                                                          6i
  %   *********   *********   *********   *********   *********   %
                                                                          6j
  % initialization, opening files %                                       6k
     % get output file name %                                            6k1
        otfl();                                                         6k1a
     % get input file name %                                             6k2
        infl();
                                                                        6k2a
  % loop to process input %                                               6l
     crlf();                                                             6l1
     typeas ($" Program starting,");                                     6l2
     LOOP                                                                6l3
        BEGIN                                                           6l3a
        % get host name %                                               6l3b
           CCPOS SF(stid);                                             6l3b1
           *statement* _ NULL;                                         6l3b2
           xtrnam($statement, $swork, -1);
                                                                       6l3b3
        % get host address %                                            6l3c
```

```
          substid _ stid; %starting point of search%                  613c1

          substid _ namingrp(substid, substid, $"HOST=ADDR", 10);
                                                                      613c2

          IF substid = endfil THEN                                    613c3

             BEGIN                                                    613c3a

             *statement* _ "skipping ", *statement*, " ... No
             Host=Addr statement";                                    613c3b

             crlf();                                                  613c3c

             typeas($statement);                                      613c3d

             GOTO newsite;                                            613c3e

             END;
                                                                      613c3f

          IF NOT FIND SF(substid) > ['(] [')] $NP ^tp1 1$D ^tp2
          THEN                                                        613c4

             BEGIN                                                    613c4a

             *statement* _ "skipping ", *statement*, " ... No
             Host=Addr value";                                        613c4b

             crlf();                                                  613c4c

             typeas($statement);                                      613c4d

             GOTO newsite;                                            613c4e

             END
                                                                      613c4f

       % first line of entry:  HOST NAME and  HOST ADDR %              613d

          ELSE *statement* _ *statement*, ', , tp1 tp2;               613d1

          outstid _ cis(outstid, $statement, sucdir);
                                                                      613d2

       % second line of entry:  STATUS= %                              613e

          substid _ stid; %starting point of search%                  613e1
```

```
      substid _ namingrp(substid, substid, $"STATUS", 10);        613e2

      *statement* _ "STATUS=";                                     613e3

      IF substid = endfil THEN GOTO nosi                           613e4

      IF NOT FIND SF(substid) > ['(] [')   $NP ^tp1
      (["SERVER"]/["USER"]/["TIP"]) ^tp2 THEN                      613e5

         (nosite);                                                 613e5a

         *statement* _ *statement*, "UNKNOWN"                      613e5b

      ELSE *statement* _ *statement*, tp1 tp2;
                                                                   613e6

   % third and subsequent lines of entry:  %                        613f

      FIND SF(*inpstr*) ^tp1;                                       613f1

      LOOP                                                          613f2

         BEGIN                                                     613f2a

         IF NOT FIND tp1 > $NP ^tp1 [',] ^tp3 ^tp2 _tp2 THEN        613f2b

            BEGIN                                                 613f2b1

            fattr($statement); % output final attr for this
            site %                                                613f2b2

            EXIT LOOP;                                            613f2b3

            END                                                   613f2b4

         ELSE   %  get next attribute name and process it %        613f2c

            BEGIN                                                 613f2c1

            *attstr* _ tp1 tp2;                                   613f2c2

            tp1[1] _ tp3[1];                                      613f2c3

            IF NOT attr($attstr, $statement) THEN                 613f2c4

               BEGIN                                             613f2c4a

               fattr($statement);                                613f2c4b
```

```
                    EXIT LOOP;                                          613f2c4c

                    END;                                                613f2c4d

                 END;                                                   613f2c5

              END;
                                                                        613f2d

        % get next top level site-name %                                613g

           (newsite): IF (stid _ getsuc(stid)) = endstid THEN EXIT
           LOOP;
                                                                        613g1

        END; % of main loop %                                           613h

                                                                        613i

     % close files %                                                    614

        frefint();                                                      614a

     crlf();                                                            615

     typeas($"HOSTS program terminated normally.");                     616

     RETURN;                                                            617

     END.
                                                                        618

(attr) PROCEDURE (atstr, statement); %find and output next attribute
%                                                                       7

  REF atstr, statement;                                                 7a

  LOCAL sbstid;                                                         7b

  LOCAL TEXT POINTER tp1, tp2;                                          7c

  sbstid _ stid; %starting point of search%                             7d

  sbstid _ namingrp(sbstid, sbstid, $atstr, 10);                        7e

  IF sbstid = endfil THEN RETURN (FALSE);                               7f

  IF NOT FIND SF(sbstid) > ['(] [')] $NP ^tp1 1$CH [ENDCHR] ^tp2
```

```
   THEN  RETURN (TRUE)  %  no attribute value for this attribute name
   %                                                                  7g
   ELSE                                                               7h
      BEGIN                                                          7h1
      % get previous attribute out %                                 7h2
         outstid _ cis(outstid, $statement, sucdir);                7h2a
      % prepare this attribute for next output %                     7h3
         *statement* _ *atstr*, '=, tp1 tp2;                        7h3a
      END;                                                           7h4
   RETURN (TRUE);                                                     7i
   END.
                                                                      7j
(fattr) PROCEDURE (stmnt);  % output final attribute for this site%    8
   REF fstr, stmnt;                                                   8a
   % put EOL on final attr, and get it out %                          8b
    *stmnt* _ *stmnt*, EOL;                                           8c
   outstid _ cis(outstid, $stmnt, sucdir);                            8d
   RETURN;                                                            8e
   END.
                                                                      8f
(infl) PROCEDURE;  %  determine input file name  %                     9
   LOCAL TEXT POINTER tp1, tp2;                                       9a
   LOOP                                                               9b
      BEGIN                                                          9b1
      stid _ origin;                                                 9b2
      crlf();                                                        9b3
    typeas($" Input file = ");                                          9b4
    *xlit* _ NULL;                                                      9b5
    txtlit($xlit);                                                      9b6
    IF xlit,L # 0 THEN                                                  9b7
       BEGIN                                                            9b7a
       IF NOT FIND SF(*xlit*) > '< THEN                                 9b7b
          BEGIN                                                         9b7b1
          ! JSYS gjinf;                                                 9b7b2
          dirno _ r2;                                                   9b7b3
          gdname (dirno, $dirname);                                     9b7b4
          *xlit* _ '<, *dirname*, '>, *xlit*;                           9b7b5
          END;                                                          9b7b6
       IF NOT FIND SF(*xlit*) >  [',] THEN *xlit* _ *xlit*, ",NLS";     9b7c
       END                                                              9b7d
    ELSE                                                                9b8
       BEGIN                                                            9b8a
       *xlit* _ "<FEINLER>HOSTADDR=MASTER,NLS";                         9b8b
       typeas($xlit);                                                   9b8c
       END;                                                             9b8d
    IF NOT (stid,stfile _ open(0,$xlit)) THEN                           9b9
       BEGIN                                                            9b9a
       typeas($" Unable to open input file,  Try another ");            9b9b
       REPEAT LOOP;                                                     9b9c
       END;                                                             9b9d
```

```
            endstid _ stid;                                               9b10
            IF (stid := getsub(stid)) = stid THEN                         9b11
               BEGIN                                                      9b11a
               typeas($"Empty input file.  Try another ");               9b11b
               REPEAT LOOP;                                               9b11c
               END;                                                       9b11d
            EXIT LOOP;                                                    9b12
            END;                                                          9b13
         RETURN;                                                          9c
         END.                                                             9d
      (otfl) PROCEDURE;  %  determine output file name %                  10
         LOCAL TEXT POINTER tp1, tp2;                                     10a
         LOOP                                                             10b
            BEGIN                                                         10b1
            freflnt();                                                    10b2
            outstid _ origin;                                             10b3
            crlf();                                                       10b4
            typeas($" Output file = ");                                  10b5
            *xlit* _ NULL;                                                10b6
            txtlit($xlit);                                                10b7
            IF xlit.L # 0 THEN                                            10b8
               BEGIN                                                      10b8a
               IF NOT FIND SF(*xlit*) > '< THEN                           10b8b
            BEGIN                                                    10b8b1
            ! JSYS gjinf;                                            10b8b2
            dirno _ r2;                                              10b8b3
            gdname ( dirno, $dirname);                               10b8b4
            *xlit* _ '<, *dirname*, '>, *xlit*;                      10b8b5
            END;                                                     10b8b6
         IF NOT FIND SF(*xlit*) >  [',] THEN *xlit* _ *xlit*, ",NLS";  10b8c
         END                                                         10b8d
      ELSE                                                           10b9
         BEGIN                                                       10b9a
         *xlit* _ "<NETINFO>HOSTS,NLS";                              10b9b
         typeas($xlit);                                              10b9c
         END;                                                        10b9d
      % Open calls err if it doesn't find the file, %                10b10
         ON SIGNAL ELSE GOTO newfile;                                10b10a
      IF outstid,stfile _ open(0, $xlit) THEN                        10b11
         BEGIN                                                       10b11a
         typeas($"  (old file) CONFIRM ");                           10b11b
         IF input() # CA THEN REPEAT LOOP;                           10b11c
         cdp(getsub(outstid));                                       10b11d
         END                                                         10b11e
      ELSE                                                           10b12
         BEGIN                                                       10b12a
         (newfile):                                                  10b12b
         typeas($" (new file) CONFIRM ");                          10b12c

         IF input() # CA THEN REPEAT LOOP;                         10b12d

         ON SIGNAL ELSE NULL; %disarm all of them%                 10b12e

         IF NOT outstid,stfile _ openull($xlit) THEN               10b12f

            BEGIN                                                 10b12f1

            typeas($" Bad Output File Name "); % (that's all it could
            be)  %                                                10b12f2

            REPEAT LOOP;                                          10b12f3

            END                                                   10b12f4

         ELSE EXIT LOOP;                                           10b12g

         END;                                                      10b12h

      EXIT LOOP;                                                    10b13

      END;
                                                                    10b14

   RETURN;                                                            10c

   END.
                                                                      10d

FINISH                                                                 11
```

L10 User Program (OLD=NLS) for creating NLS file of Host Names/Addrs/etc

(J22990) 14=MAY=74 08:50; Title: Author(s): Michael D, Kudlick/MDK; Distribution: /JAKE; Sub=Collections: SRI=ARC; Clerk: MDK; Origin: <KUDLICK>NEWHOSTS,NLS;8, 10=APR=74 12:16 MDK ;

A Update File Old Bug? and Sendmail Blowup

New NLS bug,,? 1

When I used the Update File Old command in NEW NLS today on the file (NORTON,JCN,NLS;7011,), The message I got back at the top of the display was (PMFDIRO,JOBPMF,;100014,) rather than (NORTON,JCN,NLS;7011,), Harvey seemed to think that this string has come up in funny places before and is associated with some special user space in the system, This could confuse new or old users about what file they are really updating, In addition, when I went to tell this to FEEDBACK, the sendmail subsystem blew up just after I typed the CA after typing in some message like the above, I got Illegal Instruction jsys 56=,,,10400,,56 at LODRFB+217= 54401 Illegal TENEX source/destination designator,, etc etc, So there you are, This is sent from oldnls to try to get it through, 1a

A Update File Old Bug? and Sendmail Blowup

(J22991) 14-MAY-74 09:07; Title: Author(s): James C. Norton/JCN; Distribution: /FDBK; Sub-Collections: SRI-ARC; Clerk: JCN;

DVN KIRK 14-MAY-74 09:53 22992

Four HELP Show command alternatives necessary for NLS to be self-teaching

The problem of teaching a new user of Help, while dealing with a variety of recognition modes, is serious. In fact unless one of the following four things are done, we believe that NLS cannot be self teaching because Help itself cannot be self teaching. 1

1. Commandwords are eliminated from the help subsystem. (Commands without commandwords are possible with CML. Kirk has written a Whole Universe Catalog subsystem with a CML front-end that does not use commandwords.) 1a

2. A special "Show mode" is created where the show command is automatically recognized and repeated. This has the disadvantages of (1.) requireing some special procedure to get out of the Show mode in order to specify some other command like "Quit". (2.) failing to explain how to use commands in HELP other than "Show", and (3.) unnecessary typing out of the word "Show" everytime, 1b

3. A default recognition mode is defined in the Help subsystem independant of the user's useroptions setting. We don't know how hard it would be to code such a mode. The default recognition mode could be either the default recognition for NLS (EXPERT, FIXED?) or else ANTICIPATORY. What should that default be? 1c

4. The help software and the help database are made smart enough to be able to determine the user's recognition mode and give him the welcome messages, response to questionmark, help subsystem descriptions, etc. appropriate for that recognition mode. This would by far be the hardest for the Database to implement and we suspect it would also be the hardest for the Software as well. 1d

Four HELP Show command alternatives necessary for NLS to be self-teaching

(J22992) 14-MAY-74 09:53; Title: Author(s): Dirk H. Van Nouhuys, Kirk E. Kelley/DVN KIRK; Distribution: /RWW([ACTION]) DIRT([INFO-ONLY]) ; Sub-Collections: SRI-ARC DIRT; Clerk: KIRK;

DCE 14-MAY-74 10:17 22993

My OK for circulating 22989 within SRI

Steve Miller: You have my full approval for circulating Mike Kudlick's note about the ARPANET (MJournal, 22989,) within SRI as you see fit. (Feel free to retype it if you prefer a different format -- but I would prefer having author and date listed.)
Regards, Doug 1

My OK for circulating 22989 within SRI

(J22993) 14-MAY-74 10:17; Title: Author(s): Douglas C. Engelbart/DCE ; Distribution: /mdk swm ; Sub-Collections: SRI-ARC; Clerk: DCE ;

LEG 14-MAY-74 13:29 22994

Bad Slash, String Addresses,Pompous Greeting from HELP

10-MAY-74 1054-EDT GOULD at BBN-TENEX: XNLS
Distribution: VANNOUHUYS
Received at: 10-MAY-74 07:56:44 1

DIRK -

I'VE JUST BEEN PLAYING WITH THE NEW XNLS AND FIND THAT NEITHER THE FRONTSLASH NOR THE BACKSLASH COMMAND WORKS - OR PERHAPS THEY'VE BEEN CHANGED. EACH CHARACTER IS SIMPLY ABSORBED BY THE SYSTEM. CAN'T GET THE SINGLE QUOTE FOR CHARACTER ADDRESS TO WORK EITHER AND JUST FOUND OUT THAT CONTENT ADDRESSING IS NOW EXPRESSED WITH DOUBLE QUOTES. HAS THE CONTENT ADDRESS THAT USED TO BE REPRESENTED WITH ANGLE BRACKETS BEEN ELIMINATED?

TRIED OUT THE HELP SYSTEM TO GET AID IN THESE MATTERS AND FOUND IT RELATIVELY TERRIFYING. THE INITIAL ASSAULT DEALT WHEN YOU ASK FOR HELP WITH HELP SHOULD BE ENOUGH TO DISCOURAGE ALL BUT THE HARDIEST - "IT IS AN ONLINE TIME-SHARED MULTI-CONSOLE SYSTEM WITH A COMMAND SPACE DIVIDED INTO SUBSYSTEMS FOR SPECIFIC TASKS IN INFORMATION SPACE"! WHAT AN ARRAY OF IMPOSING WORDS! (HOWEVER, I JUST TRIED IT ON OUR SECRETARY AND SHE SAID IT DIDN'T SCARE HER A BIT. IT DIDN'T MAKE ANY PICTURES IN HER
ND, BUT SHE FIGURED ALL WOULD COME CLEAR EVENTUALLY. MAYBE I'M OVER-REACTING.)

HOPE TO TRY OUT THE PRIMER ON SOME SECRETARY-TYPE SUBJECTS NEXT WEEK AND WILL LET YOU KNOW WHAT TRANSPIRES.

REGARDS,
LAURA 1a

Bad Slash, String Addresses,Pompous Greeting from HELP

(J22994) 14-MAY-74 13:29; Title: Author(s): Laura E, Gould/LEG; Distribution: /NEWNLS([ACTION]) LEG([INFO-ONLY] responses to your questionsshould come through our feedback system, I willcheck out about the nagle-breacket content search) ; Sub-Collections: NIC NEWNLS; Clerk: DVN;

MDK 14-MAY-74 08:41 22989
brief write-up on arpanet written for Steve Miller (SRI) at his request
Location: (MJOURNAL, 22989, 1:w)
*****Note: Author Copy***** 1

(J22995) 14-MAY-74 14:50; Title: Author(s): Michael D. Kudlick/MDK; Distribution: /JHB; Sub-Collections: SRI-ARC; Clerk: MDK;

L10 Answers; Response to 30732

Mike: As an example of CASE and LOOP constructs, I put a copy of INMES,NLS in the <USER=PROGS> directory at Office=1; There is a lot of special stuff in there which you don't need at this point, but you can pick out what might be of help, 1

Part Two, Section 4 of the L10 Users' Guide begins with a discussion of assignment statements (to initialize a variable, for instance), It also includes a discussion of WHILE, UNTIL, and FOR statements (condition loops), Let me know which parts of that section are not clear and I will try to elaborate, 2

Your looking at JFORM may have confused the issue, but you will find an explanation of string designators in Part Two, Section 6, on age 68, Yes *var*[i] refers to the i'th character in the string variable "var", If you wish to locate the i'th visible, you will have to setpointers around it with a FIND statement, e,g, your CCPOS will be left after the end of the i'th visible after: 3

```
    WHILE i>0 AND FIND $NP $PT DO i_i=1 ;                                  3a
```

If you are finding parts of the L10 Users' Guide confusing, I would appreciate your comments so that I might improve it, Let me know how I can be of further sevice, ==Dean 4

L10 Answers: Response to 30732

(J22996) 14-MAY-74 15:23; Title: Author(s): N. Dean Meyer/NDM; Distribution: /MIKE JCN; Sub-Collections: SRI-ARC; Clerk: NDM;

talk without slide marks

A COMMAND META LANGUAGE FOR AN NLS FRONTEND 1

INTRODUCTION 2

In the first part of this discussion, I will explain ARC's goals, motivations, and plans for splitting NLS into two parts: 2a

A frontend to interact with the user, and 2a1

A backend to carry out the commands specified by the user in the frontend. 2a2

In the second part of this discussion, I will go into detail about our concept of a frontend system which centers around a Control Meta Language for the specification of user interactions. 2b

In my concluding remarks, I will go into where we are currently with repsect to accomplishing the ideas talked about. 2c

2d

TERMINOLOGY 3

Before getting started however, I would like to define the terminology I will be using. 3a

NLS 3b

Over the past 10 years at the Augmentation Research Center (ARC) of SRI, we have been developing a computer and communications system, called NLS, to enhance the intellectual effectiveness of people by enhancing their ability to write, study and publish documents, correspondences, and notes; file and retrieve material; plan, organize and coordinate activities; and communicate with others through various media. NLS is a highly interactive system designed around well human-engineered display-based workstations. 3b1

FRONTEND SYSTEM 3c

A frontend system is a LOGICAL configuration of terminals, processing capability, and programs through which a user has access to, and interacts with, various subsystems. 3c1

FRONTEND PROCESS 3d

A frontend process is a program that is running as part of a frontend system. 3d1

BACKEND SYSTEM 3e

A backend system is a LOGICAL configuration of processing capability, and programs which perform functions specified by the user during her interactions with the frontend system, 3e1

BACKEND PROCESS 3f

A backend process is a program that is running as part of a backend system, 3f1

WORKSTATION 3g

A workstation is a well human engineered combination of desk, display(s), input and pointing devices, (perhaps integrated telephone and intercom systems, audio input/output devices, micro=film readers,) and so forth, 3g1

GRAMMAR 3h

A grammar is a tree structured data structure that represents allowed user interactions, 3h1

SUBSYSTEM 3i

A subsystem is a coherent set of functions or tools, with its own command language (described by a grammar), and its own set of backend execution processes, Examples of subsystems would be a text editor, a mail subsystem, a numerical calculator, etc, 3i1

CONTROL META LANGUAGE = CML 3j

Control Meta Language (or CML) is a formal language developed at ARC for describing the command language and interaction of a subsystem (or an application program) with its human user, A program written in CML is compiled by the CML compiler and the object code produced is a grammar, 3j1

CML INTERPRETTER 3k

A CML interpreter is a program that interprets grammars produced by the CML compiler, However, it could just as easily be a machine whose instruction set corresponds to the grammars produced by the CML compiler, 3k1

USER=PROFILE 3l

A user=profile is a data structure used by a command interpreter while interacting with the user, which describes to the interpreter how the system should appear to this user. 3l1

3m

A TWO PART NLS 4

Now for our plans, goals, and motivations. 4a

NLS is evolving into a two part system. 4b

The frontend of the system will collect commands from, and in general interact with, the user. 4c

Among the responsibilites of the frontend system are: 4c1

Prompting the user as to what state she is in at any time (e.g. which subsytem is currently being used) 4c1a

Prompting the user as to what commands are available at any time 4c1b

Prompting the user as to what action is required at any time (e.g. type something in, make a data selection, etc.) 4c1c

Providing help to the user when requested to do so (e.g. providing a list of available commands when the user issues the ? command) 4c1d

Collecting (and parsing) commands, including arguments, from the user, according to the currently active grammar 4c1e

Passing (through a well defined protocol) a complete command specification on to the backend system 4c1f

Loading new subsystem grammars from remote or local file systems 4c1g

Passing error and other state information received from the backend on to the user, and 4c1h

Manipulating, and displaying, the display data base in response to control information from the backend system 4c1i

4d

The backend of NLS will contain the core execution routines that will perform the commands specified by the user in her interactions with the frontend. 4e

Among the responsibilites of the backend system are: 4e1

Receiving a completed command specification from the frontend and manipulating the information data base in response to these commands 4e1a

Detecting and passing error and other state information to the frontend as a result of the above manipulations, and 4e1b

Passing control information to the frontend needed to manipulate the display data base in response to the above manipulations 4e1c

4f

We expect the frontend and backend processes of NLS to be subsets of generalized frontend and backend systems which provide tools to users. 4g

In particular, NLS consists of many subsytems, each of which has its own grammar and related backend processes. However, even though there may be many subsystems, each having its own command language vocabulary, the way in which the user gets Help, is prompted, makes choices between alternatives, supplies parameters to commands, and so forth, is uniform thoughout all subsystems. 4g1

Some of the subsystems currently (or soon to be) available in NLS are: 4g2

A 2-dimensional editor subsystem for the compostion, editing, and formatting of textual information. 4g2a

An identification subsystem for maintaining information about users, e.g. where they are to receive mail, what groups and/or organizations they belong to, etc. 4g2b

A calculator subsystem, for performing arithmetic operations, that allows selection from and insertion into text files 4g2c

A sendmail subsystem for distributing (and keeping track of) correspondences 4g2d

A readmail subsystem fo assisting users in processing correspondences received from other users 4g2e

A help subsystem to assist the user in learning NLS 4g2f

A "programs" subsystem for helping programmers to implement and debug programs and grammars, and 4g2g

A useroption subsystem to allow the user to customize the system for her needs or preferences 4g2h

We expect these subsytems to be a starting point from which to provide other tools to users through the concept of frontend and backend systems. 4g3

4h

We expect many benefits from this split (otherwise we would not be doing it). Among the expected benefits are: 4i

When the system is divided into its logical frontend and backend processes, and a protocol has been established for communication between these processes, it becomes possible to run frontend and backend processes on separate machines (possibly separated (or connected) by the ARPA Network). 4i1

Frontend processes will most likely be run on a satellite/frontend machine; backend processes will most likely be run on a large timesharing machine. 4i1a

For the sake of efficiency, and if there is room, it may be desirable to (and will be possible to) run some or all parts of one or more backend processes on the satellite machine. 4i1b

Users will see an increase in responsiveness. 4i2

We have acquired many hours of experience with NLS and have formed certain strong feelings about the responsiveness requirements of such a system. The responsiveness we have been attaining from our loaded TENEX system is inadaquate, especially when being used from display terminals through the network. 4i2a

The reponsiveness problems that we have observed are due in large measure to the fact that we are trying to run a program with very frequent activation and typically short computation per activation in a loaded timesharing system. Approximately 30% of the computing that NLS does is associated with command specification and display formating. We hope to remove much of this portion of NLS from the general time-sharing environment by moving it into a satellite machine. Thus, the user will profit through adaquate responsiveness and the portion of NLS left in the timesharing environment will recieve infrequent, command-at-a-time activations with significant computation per activation. This can be further enhanced by moving frequently used execution processes into the frontend system. 4i2b

Users will be able to specify commands asynchronously with respect to their execution. 4i3

A user will be able to specify new commands, and receive proper prompting, feedback, etc., without having to await the completion of previous commands. This is much more than merely being able to "type-ahead". 4i3a

4i4

The overall cost of a system will be reduced, 4i5

by reducing the backend costs associated with very frequent activations, 4i5a

by reducing network costs by transfering larger quantities of data at one time, rather than many small packets of data, and 4i5b

by removing the 30% of NLS execution code that does command specification from the backend machine and moving it out to a dedicated satellite machine. 4i5c

We expect that the additional equipment cost will be easily offset by the reduction in backend machine processing. 4i5d

We will be able to provide for well human engineered command specifcation. 4i6

We feel that from a human engineering standpoint it is important to be able to give the user as many prompts and cues as she deems necessary during the specification of commands. This has been done to date through character-at-a-time interaction with the main timeshared computer. Clearly, line-at-a-time interaction is considerably more efficient, but lacks the prompting capability. We hope through a frontend system to achieve efficiency via command-at-a-time interaction with the main time-sharing backend system while still being able to give the user help during command specification. 4i6a

We will have a beneficial modularity forced upon us. 4i7

A by-product of this frontend-backend approach is that it forces the separation of command language from basic functions of a subsystem or application program. It forces us to describe through a protocol how to perform the basic operations that a subsystem makes available. This forced modularity will allow not only new command languages to easily make use of old funtions but also the development of new functions that make use of old functions through the protocols they support. This standardized application program interface should greatly facilitate future development. 4i7a

User-specific data localized. 4i8

The frontend system provides an ideal place to localize and utilize user-specific data. This data can be fetched from a remote or local file system when the frontend finds out who the human is and can not only influence how the system appears to her, but also can accomodate many generic functions in a way specifically tailored to her. It might for example provide her with simplified file naming, allowing her to use short names which the frontend will translate into full path names, if necesssary. 4i8a

4i9

We will be able to provide a uniform user interface. 4i10

we feel that it is very important for a user to access her computer-based tools through a uniform, coherant interface. We expect this to be the largest single payoff of the frontend concept. The frontend system ALWAYS stands between the user and her tools and provides a uniform, well human engineered interface to these tools. Thus, although particular command languages may change to allow the user to refer to the functions a particular subsystem performs, the way in which the user gets Help, is prompted, makes choices between alternatives, supplies parameters to commands, and so forth, is uniform thoughout all subsystems. 4i10a

4j

To accomplish the above goals we have outlined a number of tasks, and have recognized a number of problems. 4k

We must complete the logical split of NLS into frontend and backend processes. 4k1

We must decide on an initial mapping of logical frontend and backend processes onto physical satellite and backend machines. 4k2

We must choose a satellite machine for the frontend system, and choose an operating system for the satellite machine and a language for all frontend software, 4k3

We must rewrite (or hopefully transliterate) the frontend programs so they can run on the satellite machine, 4k4

4k5

We must develop the necessary protocols for communication between the frontend and backend processes, 4k6

We intend to try a Call-by-name protocol which will allow logical procedure calls from the frontend to backend systems (via the network), 4k6a

There are problems involved here as to notifying the frontend system where the execution modules for individual commands live, We expect that at the same time that a grammar for a subsystem is loaded into the frontend system, some sort of binding will occur that binds commands to the appropriate backend machines, 4k6b

We must decide from where, and how, grammars for subsystems are loaded, 4k7

We must get a better understanding of where Network Graphics Protocol (NGP) fits into the picture, 4k8

we expect to actually drive the workstation displays by using NGP, 4k8a

We must address the major problem of synchronization, especially with regards to error recovery, 4k9

This becomes extremely difficult when we reach the point of the user specifying commands asynchroneously with their execution, 4k9a

We must address the problem of where the file system lives, 4k10

Should a frontend system include a file system? Is it necessary that the frontend system have a file system? or can we get by with using only the file system of the backend system? or only a file system on the frontend system? 4k10a

4l

THE SOFTWARE PART OF A FRONTEND SYSTEM 5

Now that I have discussed where we think we are going and how we hope to get there, I'd like to go into more detail about the frontend system, As I said before, a frontend system is a logical configuration of terminals, processing capability, and programs, For the rest of this discussion, I will be concerned only with the software aspects of the frontend system, 5a

Some likely software components (in addition to the operating system) of the frontend system are: 5b

A command processor 5b1

A user-profile 5b2

a NGP package 5b3

a Network Control Program (NCP) package 5b4

and possibly a file system, and some backend processes 5b5

5c

THE COMMAND PROCESSOR 5d

For specifying subsystem user interfaces (command languages, prompting, help facilities, etc,) we have developed a Control Meta Language (CML), (The CML compiler was written using TREE-META, a compiler compiler,) 5d1

A machine was hypothesized which had primitive operations which interacted with the user (for example, to have her choose one of several alternatives in a command or select some text from the screen as a parameter to a command), 5d1a

This hypothetical machine is a two address machine, The two addresses (in each instruction) are used to address the alternative(s) to this instruction and to address the successor instruction, At any point, any of the set of alternative instructions may be executed (based on user action) and the program counter moves to that instructions successor, Then, that instruction or any of its alternatives may be executed, The particular action(s) the user must take to execute one of the set of alternatives is dependent on the CML interpreter and the user-profile, 5d1a1

A formal language and compiler were developed for this machine that allows one to describe a desired command language and interaction sequence, This language is CML, 5d1a2

(DEMO OF CML FROM SLIDES) 5d1a2a

The program (or object code) produced by the CML compiler is a tree structured grammar. 5d1a3

An interpreter has been written to simulate this hypothetical machine on a PDP-10 for several types of display and typewriter terminals. (The command language specification is independent of the terminal type being supported with the exception of commands that only make sense for certain classes of terminals). 5d1b

It is this interpreter that serves as a standard user interface, and is in fact the command parser or processor! 5d1c

5e

THE USER PROFILE 5f

The user-profile is used by the command interpreter while interacting with the user. This data structure describes to the interpreter how the system should appear to this user (what actions she must take to disambiguate alternatives in commands, how much prompting to give her, which commands to make available to her, etc). 5f1

5g

WHERE WE ARE NOW 6

Well, that covers what we hope to accomplish; now to let you know where we currently stand. 6a

We have a running CML compiler and interpretter. 6a1

expecting pdp 11-40 5/1 6a2

logical spilt almost complete 6a3

talk without slide marks

(J22997) 10-MAY-74 09:31; Title: Author(s): Kenneth E. (Ken) Victor/KEV; Distribution: /KEV([INFO-ONLY]) ; Sub-Collections: SRI-ARC; Clerk: KEV;
Origin: (VICTOR, TALK.NLS;6,), 8-MAY-74 11:18 KEV ;####;

copy of letter i sent to David E, Rice, National Bureau of Economic Research, Inc,, in response to his request for more information about CML

14 MAY 74 7:57PM

Augmentation Research Center
Stanford Research Institute
Menlo Park, California 94025

David E, Rice
National Bureau of Economic Research, Inc,
Computer Research Center for Economic and Management Science
575 Technology Square
Cambridge, Mass, 02139

Dear Mr, Rice:

I've enclosed a copy of the notes on which I based my talk at the ACM=SIGGRAPH/NBS Workshop on Machine Independent Graphics, I've also enclosed a copy of our most recent (although somewhat out of date) documentation on CML,

CML is currently fairly specific for our NLS needs, but we have plans to make it more general in the near future,

We would appreciate seeing a copy of your description of ACOL when it becomes available,

If I can be of further assistance to you, please feel free to get in touch with me,

Sincerely,

Kenneth E, (Ken) Victor
Augmentation Research Center

(J22998) 10-MAY-74 10:35; Title: Author(s): Kenneth E. (Ken) Victor/KEV; Distribution: /RWW([INFO-ONLY]) DCE([INFO-ONLY]) CHI([INFO-ONLY]) ; Sub-Collections: SRI-ARC; Clerk: KEV; Origin: (VICTOR, LETTER,NLS;1,), 10-MAY-74 10:22 KEV ;
####;

NDM 14-MAY-74 17:01 22999

Response to Questions

Liz: Nice to hear from you, I'll try to answer your questions, 1

Yes, there is documentation on DNLS available online. The way to find it is through <USERGUIDES>ARCLOCATOR.NLS Let me know if you need help there. Of course you won't have access to DNLS without a line processor (the box beside the display terminal which allows network DNLS). If you are looking for a general discussion of DNLS, I'm afraid the online userguide may be too technical. I wish I had had time to give you a demonstration of DNLS. Perhaps you can arrange to visit me sometime? 2

Two: There is no way (currently) to set up different default viewspecs. As you learn to use them, you will probably find yourself changing your viewspecs frequently, so defaults are not all that important. There IS an easy way to set up default control characters, e.g. other than Carriage Return for Command Accept. Let me know if you would like to learn how to do this. 3

Three: The way to print on letterhead stationery is as follows: 4

1) Postion your CM to the top of the file. 4a

2) do an Output Device Printer and, instead of accepting the default file to which it offers to send the resulting print file (in the <PRINTER> directory), specify a file in your directory by giving a filename (usually the same name as the file but with the extension ".PRINT"). 4b

3) Go to TENEX and issue the REFUSE command. This prevents people from linking to you while you are printing. When you are done printing, issue the TENEX command RECEIVE LINKS. 4c

4) Then from TENEX, call the subsystem "SENDPRINT" (as you would call NLS). 4d

It asks for a printfile; give the name of the print file you just created (ALTMODE works here). 4d1

It says "Output to" and you say "s" for "Self" followed by a Carriage Return (<CR>). 4d2

It asks if it should "send form feeds?" If your terminal is set up to handle form feeds, type "y<CR>", otherwise type "n<CR>" then "y<CR>" for simulated form feeds (using line feeds). 4d3

Then it asks "Wait at page end?" and you say "y<CR>". 4d4

Then it says "go?" and you type "y<CR>". 4d5

Then you have a chance to position your stationery. When you have it lined up to the top line, type a <CR>. Between each page it waits for a <CR> giving you a chance to position a new page. When it is done it asks "More files?" and you may respond "y" or "n" followed by a <CR>. 4d6

Good luck, and feel free to call or link or whatever any time.
--Dean 5

Response to Questions

(J22999) 14-MAY-74 17:01; Title: Author(s): N, Dean Meyer/NDM; Distribution: /EAR LAC(for your information); Sub-Collections: SRI-ARC; Clerk: NDM;

DVN 14-MAY-74 17:27 23000

Contact with Questions about Documentation Support to DENDRAL

Bob Englemore (RSE) is with the heuristic DENDRAL Program at Stanford and is a neighbor and friend of mine. He called today on behalf of the Lederberg group at Stanford Medical Center investigating whether we could offer any help in "some kind of automatic online documentation". 1

Their machine supported by NIH is supposed to be a national resource and they are trying to prepare for distant users. Their desing work is not very far along in documentation and he and I groped around for somet ime trying to figure out what would be useful to say to one another. 2

I told him briefly about Sysguide, Locator, the operation of of our user maanuals online, query, Kirk's Whole Univers Catalog and HELP. He seemed for the moment most interested in HELP. 3

> He was also very interested in the possiblity of programs that would interrogate a documentation writer, which I said we did not have. It's an interesting thought. 3a

I plan to send him our most recent report with applicable parts on content filters andLlocator marked, and some design documents on HELP ad Query with the understanding that they are design documents and the subjects either don't work or work differently. 4

I mentioned the existence of the utility to him; of course they could run NLS on their 10 if they wanted to. 5

He is going to chew over the paper and perhaps try to log in and run HELP (I warned him about it's state) then call again and, presumably come over perhaps with others and see the features that interest him.. 6

I predict a lot of groping around before we figure out what if anything we might do for them and how it might be arranged (paid for), but something might come of it. 7

Contact with questions about Documentation Support to DENDRAL

(J23000) 14-MAY-74 17:27; Title: Author(s): Dirk H. Van Nouhuys/DVN; Distribution: /DCE([INFO-ONLY]) JCN([INFO-ONLY]) RWW([INFO-ONLY]) JHB([INFO-ONLY]) HGL([INFO-ONLY]) EKM([INFO-ONLY]) KIRK([INFO-ONLY]) JMB([INFO-ONLY]) ; Sub-Collections: SRI-ARC; Clerk: DVN;

MEH 15-MAY-74 08:03 23001

Maintenance, Tymshare Quote Request, Changes to PDP-10/11 List,

This is a letter to Mr Swarbrick informing him of changes to the 10/11 equipment list, MEH 15-MAY-74 08:03 23001

Maintenance, Tymshare Quote Request, Changes to PDP-10/11 List,

Augmentation Research Center
Menlo Park, California 94025
Stanford Research Institute
13 MAY 74

Mr John Swarbrick
Service Manager
Tymshare
Data Service Division
10261 Bubb Road
Cupertino, CA 95014

Dear Mr Swarbrick:

As an oversight, my letter dated 10 MAY 74 did not include our BB&N pager in the PDP-10 equipment list,

Please include it in your quote,

Sorry for this oversight, Please call if you wish to discuss further,

PDP-10 equipment (REV-1)

1,	KA10	- arithmetic processor
2,	KM10	- fast memory
3,	KT10A	- dual memory protect relocate
4,	TM10A	- mag tape control
5,	TD10	- dectape control
6,	DC10A	- data line scanner
7,	TU30B (2)	- 7 channel mag tape
8,	TU55 (2)	- DEC tape transport
9,	DC10B	- 8 line communication group
10,	MA10 (8)	- 16K core memory
11,	ME10 (2)	- 16K core memory
12,	MC10 (40)	- memory parts
13,	DF10 (2)	- data channel
14,	RF10	- disk controller
15,	RF10C	- disk controller
16,	RP02 (6)	- disk drive
17,	PAGER	- BB&N memory pager

PDP-11 equipment

1,	11/40-BA	- cpu with 16K memory

2, KE11=E = extended instructions
3, KW11=P = programmable clock
4, MF11=U (2) = 16K memory w/control for additional 16K
5, MF11=L (3) = 16K memory
6, KT11=D = memory management control
7, DH11=AA = 16 line communication multiplexor
8, H960=DA = expansion cabinet
9, DD11=A = Peripherial mounting module

Sincerely,

Augmentation Research Center

Martin Hardy
Supervisor, Computer Facility
(415) 326=6200 X3921

MEH/jcn
rww
O, Courtney

(J23001) 15=MAY=74 08:03; Title: Author(s): Martin E, Hardy/MEH;
Sub=Collections: SRI=ARC; Clerk: MEH;
Origin: (HARDY, TYM=3,NLS;34,), 14=MAY=74 09:17 MEH ;####;

Maintenance, Request to extend our PDP-10 contract.

This is a memo to Otis Courtney requesting he extend PDP-10 maintenance contract. It requests extention 10 days past our ARPA contract. By doing this we keep the bookeeping easy, and Otis says there will be no problem in canceling concurrently with ARPA contract.

MEH 15=MAY=74 08:28 23002

SRI MEMO

TO: Otis Courtney DATE: 14 MAY 74

FROM: Martin Hardy LOCATION: J=2072, X3921

SUBJECT: PDP=10 Maintenance CC: jcn

Otis:

Please extend our weekday (16 hours 5 day a week) PDP=10 equipment maintenance contract from 10 FEB 74 to 10 JULY 74, Do not extend our add=on weekend (saturday/Sunday) coverage,

Martin Hardy

Maintenance, Request to extend our PDP-10 contract.

(J23002) 15-MAY-74 08:28; Title: Author(s): Martin E. Hardy/MEH;
Sub-Collections: SRI-ARC; Clerk: MEH;
Origin: (HARDY, MAINT/10.NLS;2,), 14-MAY-74 14:28 MEH ;####;

Scenarios for running the NIC L10 Programs for Mailing Labels and/or Membership Lists

BRIEF DESCRIPTION 1

These scenarios are for using an L10 user program that lives in the following places: 1a

L10 Source Program 1a1

for New NLS: (mjournal, 22975, 1: wz)
for Old NLS: (mjournal, 22976, 1: wz) 1a1a

REL FILE for L10 Program 1a2

for New NLS: (kudlick, NICLYST.REL,)
for Old NLS: (kudlick, MEMLYST.REL,) 1a2a

NOTE ::::: This is a relatively large user program; it loads o.k. with a buffer size greater than 10. 1a2b

The program creates mailing labels and/or formatted name-and-address membership lists for any designated groups or individuals, or for the entire network community, using information in the Identfile. 1b

THESE SCENARIOS ASSUME THAT THE USER HAS ALREADY SET THE BUFFER SIZE TO AT LEAST 10, AND LOADED AND "RUN" THE PROGRAM VIA THE PROGRAMS SUBSYSTEM OF NLS. 1c

After it starts, the program automatically types out the first question and waits for a yes/no answer, then it types out subsequent questions, always waiting for an answer to one question before proceeding to the next question. When it has all the info it needs, it then starts processing and types out the group and individual idents as they are processed. It tells you when it's finished. 1d

After the program is finished, you must then perform a few more operations to produce the output membership listing and/or produce the BCD tape file. 1e

SCENARIOS 2

1) scenario showing how to create the membership lists for two groups, INWG and NGG 2a

a) DURING PROGRAM EXECUTION 2a1

```
 Manual Ident Entry Mode? (Y or N)     Y <cr>
Idents:  INWG <space> NGG <cr>
```

```
         Do you want LABELS?  (Y or N)    N <cr>
         Do you want MEM-LISTS?  (Y or N)    Y <cr>
         Output File for Mem-Lists =  filename <cr>  (new file)
        CONFIRM <cr>

        INWG
         xxx xxx xxx xxx ...
        NGG
         zzz zzz zzz ...

        Processing Finished Normally
         mem-list file is <DIRECTORY>FILENAME.NLS;1
```
2a1a

b) TO PRODUCE THE OUTPUT MEMBERSHIP LISTING 2a2

```
        * QUIT PROGRAMS <cr>
        * LOAD FILE filename <cr>
        * UPDATE FILE <cr>      (not strictly necessary)
        * SET VIEWSPECS wzn <cr>
        * OUTPUT QUICKPRINT <cr>
```
2a2a

2) scenario showing how to create the mailing labels for three groups, PRG, SURG, and USING, 2b

a) DURING PROGRAM EXECUTION 2b1

```
         Manual Ident Entry Mode? (Y or N)     Y <cr>
        Idents:  PRG <space> SURG <space> USING <cr>
         Do you want LABELS?  (Y or N)  Y <cr>
         Output File for Labels =  filename <cr>  (new file)
        CONFIRM <cr>
         Do you want MEM-LISTS?  (Y or N)    N <cr>

        PRG
         xxx xxx xxx xxx ...
        SURG
         zzz zzz zzz ...
        USING
         www www ...

        Processing Finished Normally
         labels file is <DIRECTORY>FILENAME.NLS;1
```
2b1a

b) TO PRODUCE THE BCD TAPE FILE FOR USE BY THE CDC-6600 MAILING LABELS PROGRAM 2b2

```
        * QUIT PROGRAMS <cr>
        * LOAD FILE filename <cr>
        * UPDATE FILE <cr>           (not strictly necessary)
        * JUMP TO 1 <cr>
        * SET VIEWSPECS wzn <cr>
        * OUTPUT SEQUENTIAL FILE seqfile <cr> <cr>
        * GOTO TENEX <cr>
        @ BCDTAP <cr>
          ...
          ... Here you must answer BCDTAP's questions.
          ... Especially: select 800 bpi.
          ... Be sure to have a tape reel mounted before you use
        BCDTAP,
          ... and type the name of your output sequential file for
        the
          ... input file to BCDTAP.
          ...
```
2b2a

3) other scenarios 2c

a) To run in automatic mode, answer "No" to the "manual-entry-mode" question. 2c1

b) To create labels and membership lists simultaneously, answer "yes" to both the "do-you-want-labels" question and the "do-you-want-memlists" question. You can do this in either automatic mode or manual mode. 2c2

Scenarios for running the NIC L10 Programs for Mailing Labels and/or Membership Lists

(J23004) 15-MAY-74 11:56; Title: Author(s): Michael D, Kudlick/MDK; Distribution: /MDK; Sub-Collections: SRI-ARC; Clerk: MDK; Origin: <KUDLICK>SCENARIOS,NLS;4, 15-MAY-74 11:53 MDK ;

HGL 15-MAY-74 15:10 23006

Help Additions and Fixes Currently Being Implemented

Introduction 1

The following fixes and additions to the Help system are currently under implementation, They include items which were scheduled before the system was brought up as well as items which are being done because of user bug reports and suggestions, other items are not included on this list because they involve more basic recosideration of the design of the system or because of implementation difficulties in the CML environment, They will be discussed with a more detailed presentation of the further planned evolution of the Help/Query system in a later document, 1a

The following are new commands, features, or conceptual modifications: 2

Create a new entity for help system, "node", (Text is not sufficient,) 2a

Permit bugging of nodes, 2b

New back command to replace Show <, etc, This command would offer the user a view of the beginnig of the first line of the text/menu portrayed in much the same manner as Jump to Return in NLS itself and is an attempt to meet the criticisms that it is difficult for the user to tell his context and that the angle bracket is confusing, 2c

Feedback path name in teletype simulation window, 2d

Go to proper subcommand level regardless of when control-Q is typed, This was left out in the first pass, 2e

Have question mark in help give a short description of the system rather than just the command options, 2f

Implement a "Show mode", While in this mode the user need not retype the "Show" command, A special character will exit this mode and permit the execution of other Help system commands, 2g

The following satisfy additions and fixes requested by the data base builders: 3

Entry messages should be in the data base and under the control of the database builder, 3a

Entry messages should be changed as requested if the preceding is not done soon, 3b

Viewspecs will work in included links, 3c

Unmenued statements will, by default, display all line, all levels. 3d

Should be able to have unmenued, named statements (i.e.,] could be a delimiter.) 3e

Should be able to have comment, named statements (i.e., % could be a delimiter.) 3f

The following are bug fixes: 4

Counting with comment or unmenued characters in MORE command should not lead to an EMPTY screen. (Are we seeing if there iare "more" without checking if there are more MENUED statements to show?) 4a

Do not make general subsystem commands other than quit (e.g., execute, etc) available in help if possible. Could the commands be hidden when the question mark message is typed out? 4b

Recycle used menu storage area when it is full. (This is currently not done and leads to the subsystem being aborted with the message "Data Base Portrayal Trouble. Call ARC!" It is not fatal and Help may be reentered, but it is inconvenient. We jsut hadn't gotten around to putting the code in on the first pass. 4c

Make the error messages a bit more enlightening. 4d

Look into problem of not resetting the user's command state in certain subsystems after going to and returning from help. (Is this possible? Perhaps CML limitation; bad for systems like sendmail.) 4e

Help Additions and Fixes Currently Being Implemented

(J23006) 15-MAY-74 15:10; Title: Author(s): Harvey G, Lehtman/HGL;
Distribution: /SRI-ARC([INFO-ONLY]) ; Sub-Collections: SRI-ARC;
Clerk: HGL;
Origin: (LEHTMAN, MOREHELP.NLS;4,), 15-MAY-74 15:07 HGL ;####;

MY OK for circulating 22989 within SRI

Steve Miller: You have my full approval for circulating Mike Kudlick's note about the ARPANET (MJournal, 22989,) within SRI as you see fit, (Feel free to retype it if you prefer a different format -- but I would prefer having author and date listed,)
Regards, Doug 1

My OK for circulating 22989 within SRI

(J23007) 15-MAY-74 16:00; Title: Author(s): Douglas C. Engelbart/DCE ; Distribution: /mdk swm ; Sub-Collections: SRI-ARC; Clerk: DCE ;

Announcement of Changes in NIC Services

This announcement was mailed to all Principal Investigators, Technical Liaisons, Station Agents, Network Associates, and Group Coordinators. Also, a log-in message at OFFICE-1 points NIC on-line users to a copy of the announcement.

Announcement of Changes in NIC Services

NIC # ...
Author: M.D. Kudlick
Title: Announcement of Changes in NIC Services
To: PI, NLG, NSAG, NAG, and All Group Coordinators 1

TO ALL USERS OF NIC SERVICES: 2

EFFECTIVE JULY 1, 1974, THE SCOPE AND KINDS OF SERVICES PROVIDED BY THE NETWORK INFORMATION CENTER WILL BE REDUCED AT ARPA'S REQUEST. 3

THIS NOTE DESCRIBES THE NIC SERVICES SRI=ARC HAS PROPOSED THAT ARPA SUPPORT AFTER JULY 1, 1974. IT ALSO DESCRIBES THOSE CURRENT NIC SERVICES THAT WILL BE DISCONTINUED ON THAT DATE. 4

IF YOU HAVE ANY QUESTIONS REGARDING THESE CHANGES, PLEASE CONTACT E.J.FEINLER (FEINLER@SRI=ARC). BEGINNING JULY 1st, MS. FEINLER WILL SUPERVISE ALL NIC OPERATIONS.

5

I. NIC Services that SRI=ARC is proposing to continue after July 1, 1974

6

ON=LINE NIC SERVICES AT OFFICE=1 6a

1) There will be one "NIC" login directory maintained at OFFICE=1 for use by all NIC users, for the purpose of accessing the Resource Notebook and other NIC files through the NIC/QUERY system. 6a1

2) The ASCII file of official Network hostnames and addresses, as described in RFC# 608, will be maintained at OFFICE=1 and continue to be accessible from that host via FTP, using the pathname OFFICE=1 <NETINFO>HOSTS.TXT 6a2

3) No other on=line services will be provided. In particular, no directories will be provided to NIC users for on=line use at OFFICE=1 unless separate contractual arrangements are made by interested users with J.C. Norton (NORTON@OFFICE=1). 6a3

OFF=LINE NIC SERVICES 6b

1) The NIC will continue to maintain, publish, and distribute the Arpanet Directory in its present format. 6b1

2) The NIC will maintain, publish, and distribute the Arpanet Resource Notebook in a new, compact "handbook" format. 6b2

3) The NIC will publish and distribute a modified version of the Arpanet News, as time and funds permit. 6b3

OTHER SERVICES 6c

1) The role of Station Agent will be discontinued. 6c1

2) The role of Technical Liaison will be continued in order to provide a mechanism for the NIC to obtain information about Hosts for inclusion in the Resource Notebook, and to provide users a contact at each Host. 6c2

3) The role of the NIC's staff in maintaining the NIC identfile will be continued. The mechanism by which name/address and other changes are to be sent to the NIC will be announced. 6c3

6d

II. NIC Services that will be discontinued after July 1, 1974

7

ON-LINE NIC SERVICES AT OFFICE-1 7a

1) Use of NLS at OFFICE-1 by NIC users will be discontinued. 7a1

2) Use by NIC users of the NIC Journal via NLS will be discontinued. 7a2

3) Support of special interest groups such as INWG, SUR, PRG, USING, etc. will be discontinued. Discontinued services include publication and distribution of group notes and membership lists. 7a3

4) Support of the Network protocol developers, through publication and distribution of RFC's (and RFC indexes), and publication and distribution of the Protocol Notebook, will be discontinued. 7a4

5) Auxiliary services, such as the "NIC Locator", NIC Journal "indexes", and on-line "HELP" for NIC users, will be discontinued. 7a5

OFF-LINE NIC SERVICES 7b

1) The Station Agent Collection of NIC Functional Documents will be discontinued. No further loose-leaf updates to the Station Agent Collection will be provided; and no replacements

for these Functional Documents will be provided, other than the Arpanet Directory and Resource Notebook as discussed above, 7b1

2) All off-line services that supported or were supplementary to the discontinued on-line services listed above, will themselves be discontinued. These discontinued off-line services are primarily the hardcopy distribution of NIC Journal documents, including personal and group correspondence and RFC's, 7b2

OTHER SERVICES 7c

1) The availability of Enterprise and Zenith phone numbers for placing toll-free calls to the NIC from certain areas of the country will be discontinued, 7c1

(J23008) 15=MAY=74 16;10; Title; Author(s); Michael D, Kudlick/MDK;
Distribution; /SRI=ARC([INFO=ONLY]) ; Sub=Collections; SRI=ARC;
Clerk; MDK;
Origin; <KUDLICK>D,NLS;14, 14=MAY=74 11;40 MDK ;####;