

Set Name Delimiters and Default Directory for Links bug with locked files,

If a file is locked, these modifications do not remain, A bug that DELETES work !!!

Set Name Delimiters and Default Directory for Links bug with locked files,

(J22938) 9-MAY-74 16:43; Title: Author(s): Kirk E. Kelley/KIRK;
Distribution: /BUGS(* action *) ; Sub=Collections: SRI=ARC BUGS;
Clerk: KIRK;

I've been thinking about the ANSWER global variable in th NLS syntax and I remembered an Idea I had long ago which seems so logical that I just about did it, but I thought I'd check with you just in case, The change is to subtutitute Y/N for ANSWER everywhere in the syntax,

1

(J22939) 9-MAY-74 16:50; Title: Author(s): Kirk E. Kelley/KIRK;
Distribution: /JMB(* action *) ; Sub=Collections: SRI=ARC; Clerk:
KIRK;

Laura Goulds thoughts as seen by RWW in her Tutorial Material

Laural Gould of BBN is writing tutorial text for a system to experiment with SCHOLAR teaching people NLS. If you read her tutorial carefully you can see awkward places in the language that she had to gingerly skirt. The ones obvious to me are:
Jump to Content searching from where you are only to end of file,
She thinks this is not intuitive and that it should start searching at the beginning or at least from where you are and wrap around to where you satarted,

Levadj should be either upper or lowercase,
The fina confirmation in substitue bothers her and seems redundant since it already asked if you are finished,
in inset and substitue commands where multiple words or characters can be input the text (s) should appear after word or character (Ithought we had decided to do that may months ago, why didn't we?)
She seems to find prompts necessary but awkward for new users because they force concept explanations earlier or require trying to avoid sliding by. This tends to indicate a need for language subsetting to avoid some concepts. We will invite her out around June 1 to tell us her experiences and thoughts again,

1

Laura Goulds thoughts as seen by RWW in her Tutorial Material

(J22940) 9-MAY-74 17:05; Title; Author(s); Richard W, Watson/RWW;
Distribution: /FDBK(* action *) ; Sub=Collections: SRI=ARC; Clerk:
RWW;

Sendmail 'Send' command suggestion

Instead of the current "Action or Informationonly" barrier in the Send command, the default should be "Information" with an OPTIONAL Action commandword (recognized by a single <="U>),

1

Sendmail 'Send' command suggestion

(J22941) 9=MAY=74 17:29; Title: Author(s): Kirk E. Kelley/KIRK;
Distribution: /FDBK(* action *) ; Sub=Collections: SRI=ARC; Clerk:
KIRK;

Problems with the Insert Statement command

1. L; is not the correct prompt. It should be L;/T;/B;/[A:] 1
2. If you type <^U> at L;, the prompt changes to A; and you get to the address field; if you type in text at L; the prompt changes to T; and the text goes in; however, if you BUG a statement, the prompt changes, ..., but nothing happens. 2
- This used to be the case where ever L; appeared but it was fixed for Copy, Move, Replace, etc. Please fix this bug for Insert as well. 2a
3. If you try to insert a space at L; the prompt changes but nothing happens. 3

Problems with the Insert Statement command

(J22942) 9-MAY-74 17:44; Title: Author(s): Kirk E. Kelley/KIRK;
Distribution: /FDBK(* action *); Sub=Collections; SRI=ARC; Clerk:
KIRK;

Character Size changes should be viewspecs instead of commands,

Character size changes in DNLS should be viewspecs,

Character Size changes should be viewspecs instead of commands.

(J22943) 9-MAY-74 17:46; Title: Author(s): Kirk E. Kelley/KIRK;
Distribution: /FDBK(* action *) ; Sub=Collections: SRI=ARC; Clerk:
KIRK;

Delete Invisible inconsistency

The Delete Invisible command works differently if you bug the beginning of a word surrounded by invisibles than if you bug the end of the word. It should work the same in both cases (delete the tip character of the word as well as the invisible).

1

Delete Invisible inconsistency

(J22944) 9-MAY-74 17:50; Title: Author(s): Kirk E. Kelley/KIRK;
Distribution: /FDBK(* action *) ; Sub=Collections: SRI=ARC; Clerk:
KIRK;

From Yngvar Lundh, Norway,

Information request, 10=MAY=74 01:31

Marcia,

1) May I request a copy, please, of: Arpa Satellite system note 32, NIC 13147, by R. Binder: "Another ALOHA Satellite Protocol", (Referenced by L.G.Roberts at Brighton, Sept 1973),

2) Is "ARPA Satellite System" an organization or group, and if so, could you make available to me further references to objectives, persons and publications?

3) Is there available a manual for the new nls system, and if so, could I have one? (I had been considering starting to give a little private course in nls in my group here soon, and will need to understand what the changes are both of detail and philosophy. Have had no time to try the new version yet, so I am still using the old version as best I can,)

Regards from Yngvar,

1

1a

1b

1c

1d

1e

YL 10-MAY-74 01:46 22945

From Yngvar Lundh, Norway.

(J22945) 10-MAY-74 01:46; Title: Author(s): Yngvar Lundh/YL;
Distribution: /MLK; Sub=Collections: NIC; Clerk: YL;

Directories affected by Privacy Features

Jim, I have a directory <Netinfo> at both SRI=ARC and OFFICE=1. The only person who should have access to these directories is myself. Jeanne Leavitt uses the directories but logs in as me. I also should be able to login to directory <NIC> and <DOCUMENTATION> and <NICWORK>. There is now a GUEST directory at OFFICE=1 which accomodates all Nic users. Anyone should be able to login as GUEST password ARC or ARPA but no one should be able to save files - that is, when a user logs out his files should be deleted from GUEST. I suppose this directory would have to accept every ident in the ident system.

1

JAKE 9=MAY=74 22:14 22946

Directories affected by Privacy Features

(J22946) 9=MAY=74 22:14; Title; Author(s): Elizabeth J, (Jake)
Feinler/JAKE; Distribution: /JEW; sub=Collections: SRI=ARC; Clerk: JAKE;

Maintenance, Tymshare Quote Request, REVISED, PDP-10/11,

This is journal item 22921 revised, it has a updated PDP-11 equipment list,

MEH 10-MAY-74 08:58 22948

Maintenance, Tymshare Quote Request, REVISED, PDP=10/11.

(J22948) 10-MAY-74 08:58; Title: Author(s): Martin E. Hardy/MEH;
Distribution: /MEH(* action *) JCN(* info-only *) ; Sub=Collections:
SRI=ARC; Clerk: MEH;

RLL 10-MAY-74 10:01 22949

A few thoughts on a calendar system

Just some quick ideas on Calendar System

A few thoughts on a calendar system

The Structure of the "Calendar,"

The Calendar file for each person will have several conceptual parts. The decision on how to manage these conceptual entities internal should be left to the programmer=designer,

From the user's point of view each day in the calendar includes sections called "Reminders", "Notes", and "Schedule". The distinction between "notes" and "reminders" is that in the latter a message will appear when he logs on indicating these are reminders that he should read today. We will use the term "item" to mean an entry in any of these sections,

Items may be placed on one person's calendar by someone else, but only after it is approved by the receiving individual. A message will appear when entering into the Calendar system if some other user has sent him an item to be placed on his Calendar. In some cases, for example, a secretary, no approval will be necessary, but a message to the effect that an item was placed on the Calendar by someone will be given,

The capabilities of a user once in the Calendar system will basically include the reading and writing of items in the Calendar file,

Information in the file may be viewed either in summary or complete form. For the display NLS users, the screen will initially display a summary form of the day's calendar. For all users any form maybe called upon for viewing,

Summary displays (for both display NLS and TNLS) can be viewed for a day, a week, a month, or a quarter year,

Complete viewing of Calendar items will be possible for any time period. In addition a user may request the viewing of only "scheduled items", "reminders", or "notes,"

Any of these views maybe requested to be printed out in hardcopy by the line=printer,

Because a display NLS user may not be able to view all of the Calendar information requested at one time, an additional capability will be available so one can "roll" the Calendar backward or forward. These commands will be simple "jump" commands,

Items that are placed into the Calendar file are assigned to one of several classes. Default classification will be made if none is specified,

A few thoughts on a calendar system

A tentative list of classes is the following;

EVENTS

MEETINGS

DEADLINES

NOTES.

1k

1k1

1k2

1k3

1k4

Each class will have its own retention period (after which the item will be deleted automatically from the file) and reminder defaults. This will enable, for example, all items classified as DEADLINE to have reminders automatically placed on the Reminder list for week before and a day before the due date.

1l

It will also be possible to view any one class at one time.

1m

The users, of course, can specify other retention periods or reminder prompts.

1n

This reminder procedure will accept such input as DAILY, WEEKLY, MONTHLY, YEARLY, DAY AHEAD, WEEK AHEAD, n DAYS AHEAD, and n WEEKS AHEAD as well as specifying arbitrary frequency, intervals, or dates.

1o

The insertion of items on the schedule is made by specifying the day and, optionally, the time of the item. This will be done on the display by "bugging" the appropriate place in the summary display or on the teletypewriter (and display) by typing the date and time.

1p

The specification of day and time can be done in a variety of ways. In the absence of the year, month, or day, the current value will be used. The time need not be specified in all cases. Month and day may be given in relative terms, for example, two weeks from next Thursday.

1q

These dating options will also be permitted when asking for information from the Calendar file.

1r

"Reminders" may be inserted independent of a scheduled item.

1s

"Notes" have a day but no time associated with them. They, also, may be inserted and viewed separately.

1t

Other features would include a rescheduling and deletion command. The user may also define new classes with special default settings for retention or reminders.

1u

A few thoughts on a calendar system

It will be possible to have selected calendar items listed on a set of days, for example, "ON VACATION", appearing for two weeks. Tentatively, we can call this "Block item"; one item will be placed on a block of dates,

1v

The "schedule" section will have a specific time associated with each item,

1w

Syntax considerations

2

When inserting an item into the Calendar, after the time is given the system will ask to which class this item belongs. If none is specified the default class will be used. Next, the system will prompt for reminders,

2a

"NO" will mean no reminders

2a1

"YES" will mean use the default reminders for this class

2a2

Other options may be specified as is given above,

2a3

A few thoughts on a calendar system

(J22949) 10-MAY-74 10:01; Title: Author(s): Robert N. Lieberman/RLL;
Distribution: /RWW(* info-only *) SRL(* info-only *) JCN(*
info-only *) ; Sub=Collections: SRI=ARC; Clerk: RLL;
Origin: (LIEBERMAN, CALENDAR=REM,NLS;3,), 10-MAY-74 09:56 RLL
;####;

Jake ... the two files that you'll want for the frontispieces are;

(gjournal, 21326, 1: why) ; How to use the Arpanet Directory
(gjournal, 21327, 1: why) ; How to use the NIC

... Mike

1

MDK 10=MAY=74 11:15 22950

(J22950) 10=MAY=74 11:15; Title: Author(s): Michael D. Kudlick/MDK;
Distribution: /JAKE; Sub=Collections: SRI-ARC; Clerk: MDK;

KIRK 10-MAY-74 11:20 22951

Bug with return stacks after split screens

When I try to JUMP to RETURN after splitting the screen vertically, I get the message "Illegal Return Ring .:. ". I have to reset in order to be able to return.

1

KIRK 10=MAY=74 11:20 22951

Bug with return stacks after split screens

(J22951) 10=MAY=74 11:20; Title; Author(s): Kirk E. Kelley/KIRK;
Distribution: /BUGS(* action *) ; Sub=Collections: SRI=ARC BUGS;
Clerk: KIRK;

SLJ 10-MAY-74 12:34 22953

to susan blue eyed

this is just a pile of blap from fergy to see how this works, if you
are smart, you will ignore all this

1

SLJ 10=MAY=74 12:34 22953

to susan blue eyed

(J22953) 10=MAY=74 12:34; Title: Author(s): Sandy L. Johnson/SLJ;
Distribution: /FDBK(* info=only *) ; Sub=Collections: SRI=ARC; Clerk:
SLJ;

Thanks for the Command Summary

Thank you for the command summary, Its really useful, Its an example of the type of documentation overview tht one would like to automatically get out of the Help database with a user program, How can that be arranged?

RWW 10-MAY=74 14:40 22954

Thanks for the Command Summary

(J22954) 10-MAY=74 14:40; Title: Author(s): Richard W. Watson/RWW;
Distribution: /FDBK(* action *) JMB(* action *) KIRK(* action *)
DVN(* action *) ; Sub=Collections: SRI=ARC; Clerk: RWW;

MEH 10-MAY-74 15:57 22955

Study Request Approval, Change Request For Spring (74) Quarter,

MEH 10-MAY-74 15:57 22955

Study Request Approval, Change Request For Spring (74)
Quarter,

SRI

MEMO

TO: Bob Wing

DATE: 7 MAY 74;

FROM: Martin Hardy

LOCATION: J=2072, X3921

SUBJECT: Study Request Approval Changes (Stanford Spring
quarter)

CC: jcn

Bob, as I have explained in our conversation last week, I was not aware that changing my class enrollment required additional approval. Jane Golet's memo dated Nov 12, 73 (attached) infers it was ok to do so as long as the classes I actually enrolled in did not exceed the number of quarter units approved, and were in line with my degree objective.

This is the premise on which I acted. My original approval for this spring quarter was for 7 units; one 3 unit (EES=236) and one 4 unit (CS=106) class. My actual Stanford enrollment was in one 3 unit class (LAC=10).

LAC=10 is a less demanding class than either EES=236 or CS=106. I chose to enroll in it instead because of the uncertainty of the need for my skills in any renewal of our ARPA contract, and because it is a class that will enhance my daily work and assist me in achieving my degree objective.

LAC=10 is offered by Stanford's Learning Assistance Center, and fully sanctioned by the University. You must be a matriculated Stanford graduate or undergraduate student to enroll. The class and Center's objective is to enhance concepts of learning. To this end the class teaches critical and analytical reading and in addition develops high speed reading capabilities (such as skimming and scanning technical articles and journals).

Because of the class objective and because it is taken only by Stanford students I feel this class will be extremely beneficial to myself and SRI.

Martin Hardy

PS:

I do not plan to enroll in a class this summer. And, in addition, because of conflicts with my work and family

MEH 10-MAY-74 15:57 22955

Study Request Approval, Change Request For Spring (74)
Quarter.

life, I do not plan to enroll in more than one 3 or 4
unit class in subsequent quarter.

MEH 10-MAY-74 15:57 22955

Study Request Approval, Change Request For Spring (74) Quarter.

(J22955) 10-MAY-74 15:57; Title: Author(s): Martin E. Hardy/MEH;
Distribution: /JCN(* info-only *) ; Sub=Collections: SRI=ARC; Clerk:
MEH;
Origin: (HARDY, SCH,NLS;27,), 10-MAY-74 15:35 MEH ;####;

Jim ... Here are a few more SRI=ARC directory names that may have multiple users: USING HELP ENERGY CATALOG BCPL BBN=ORG BBN=SOURCES,

The first one is Dave Crocker/Nancy Neigus et al,

The second is Jean Iseli, possibly others,

Dirk van Nouhuys knows about the ENERGY directory, Beau about the CATALOG one of course. A message to these persons should provide the appropriate ident lists,

As for BCPL, BBN=ORG, and BBN=SOURCES, I don't have any clues but I include them here in case you know of them, and overlooked them,

... Mike

MDK 10-MAY=74 16:16 22956

(J22956) 10-MAY=74 16:16; Title; Author(s): Michael D. Kudlick/MDK;
Distribution: /JEW; Sub=Collections: SRI=ARC; Clerk: MDK;

Some bugs in New NLS

- 7 May: Load File, gives no chance for setting VSPECS 1
- 7 May: Load File, doesn't seem to work by bugging name=word of file you want to load 2
- 7 May: Split Window == doesn't wait for you to indicate which window you want the to-be-broken window's contents to be in; just arbitrarily assigns a window. 3
- 7 May: VSPECs on a Jump == erratic behaviour. Not sure of course that keyset works every time, but it has been too consistently erratic not to raise suspicion. 4
- Then several times, when I have commanded, for instance, Jump Up Bug tz CA, it has done a weird thing in creating the new display: seemed to use tz VSPEC for while, then reverted back to the wy VSPEC that existed just before the Jump. (Note: might be a funny, residual=window problem. I seemed to have trouble getting rid of a horizontal boundary earlier == but maybe that problem really was the VSPEC=shift problem and I just thought that the window was still there because of the inconsistent VSPEC zones, and of course if there wasn't a window there the Move Boundary Command would tell me it couldn't.) 4a
- 10 May: Jump Word First, next prompt is "RPT", and "?" shows alternatives of onlyr "RPT" and "RPT". User has to discover that he can enter LIT. Same for Jump Content. (Am I missing what the "RPT" stands for?) Anyway, seems to me that something is wrong here. 5
- 10 May: Compile Contentanalyzer doosn't work for a bugged pattern. 6

DCE 10-MAY-74 16:26 22958

Some bugs in New NLS

(J22958) 10-MAY-74 16:26; Title; Author(s): Douglas C.
Engelbart/DCE; Distribution: /FDBK(* action *) ; Sub-Collections:
SRI=ARC; Clerk: DCE;

KIRK 10-MAY-74 16:37 22959

Hardcopy reference documentation for NEWNLS

I have placed a hardcopy version of the journalized HELP command summary in a black cardboard binder on the round table in the console area. This is for reference and should be returned to the round table when you are through with it.

1

KIRK 10-MAY-74 16:37 22959

Hardcopy reference documentation for NEWNLS

(J22959) 10-MAY-74 16:37; Title: Author(s): Kirk E. Kelley/KIRK;
Distribution: /SRI=ARC(* info-only *); Sub=Collections: SRI=ARC;
Clerk: KIRK;

blueyedsusan II

well, susan, it took me all day (at least 30 minutes) to figure out how to do this, but here is one comment: when you go to help and show sendmail subsystem and ask for #7 you get #6 so yyou then ask for # 8 which isnt there and you get #7 which is really what you want to know simplyy the (god, my typing) sendmail commands, love flew out the door when money came inuendo, xxxhave good wknd, carmen,

1

SLJ 10-MAY-74 16:49 22960

blueyedsusan II

(J22960) 10-MAY-74 16:49; Title: Author(s): Sandy L. Johnson/SLJ;
Distribution: /FDBK(* info-only *) ; Sub=Collections; SRI=ARC; Clerk:
SLJ;

well, susan it took me forever . . .

1

SLJ 10-MAY-74 16:54 22961

(J22961) 10-MAY-74 16:54; Title: Author(s): Sandy L. Johnson/SLJ;
Sub=Collections: SRI=ARC; Clerk: SLJ;

Creating the hardcopy NLS command summary from the HELP database via a user program,

I don't think that would be too hard to do. It would just be a matter of allocating the time for it. I could probably do it but it would take me at least three times as long as say HGL and I am very busy with the online documentation right now. However, I do have some ideas for creating a useable Hardcopy Help Database (HHD) by doing such things as replacing all of the links with statement numbers and providing a comprehensive index, table of contents, etc. CHI talked about automatically generating the Syntax and Examples automatically from the CML code. This would obviously be the ultimate way to go.

KIRK 10-MAY-74 16:59 22962

Creating the hardcopy NLS command summary from the HELP database via
a user program.

(J22962) 10-MAY-74 16:59; Title: Author(S): Kirk E. Kelley/KIRK;
Sub=Collections; SRI=ARC; Clerk: KIRK;

Creating the hardCopy NLS command summary from the HELP database via a user program as suggested by RWV's <22954,>.

I don't think that would be too hard to do. It would just be a matter of allocating the time for it. I could probably do it but it would take me at least three times as long as say HGL and I am very busy with the online documentation right now. CHI has ideas for generating the Syntax and Examples automatically from the CML code. This would obviously be the ultimate way to go.

1

KIRK 10-MAY-74 17:31 22963

Creating the hardcopy NLS command summary from the HELP database via
a user program as suggested by RWW's <22954,>.

(J22963) 10-MAY-74 17:31; Title: Author(s): Kirk E. Kelley/KIRK;
Distribution: /RWW(* info-only *) DVN(* info-only *) JMB(*
info-only *) HGL(* info-only *) CHI(* info-only *) ;
Sub=Collections: SRI=ARC; Clerk: KIRK;

extra prompt in new nls

In a few commands after one has a choice of input (e.g, "OK:/ C:") the system response with the selected choice prompt, E.G, update C: File OK:/C C: Compact OK:, It is nice to know what I picked (I think) but in TNLS I feel it is unnecessary and adds to the overabundance of feedback characters. True one can turn the whole thing off via useroption but why have it at all. Just added work for the system, the network and the terminal (as well as the human trying to watch every character), OUT DAMN C.

RLL 10-MAY=74 18:19 22964

extra prompt in new nls

(J22964) 10-MAY=74 18:19; Title: Author(s): Robert N.
Lieberman/RLL; Distribution: /FDBK(* action *) ; Sub=Collections:
SRI-ARC; Clerk; RLL;

SENDMAIL BUG

Don't send anything via the new SENDMAIL that is less than 2000 characters but more than one paragraph or statement long, SENDMAIL will only deliver the first paragraph,

1

KIRK 12-MAY-74 13:41 22965

SENDMAIL BUG

(J22965) 12-MAY-74 13:41; Title: Author(s); Kirk E. Kelley/KIRK;
Distribution: /BUGS(* action *) SRI=ARC(* info-only *) ;
Sub-Collections: SRI=ARC BUGS; Clerk: KIRK;

Ident delivery change for AKS

Would like Ken Showalter of QNR (AKS = nic Ident) to have his delivery changed to hardcopy. Currently no delivery at all is specified. Jim N, do you agree?

1

RLL 12=MAY=74 18:51 22966

ident delivery change for AKS

(J22966) 12=MAY=74 18:51; Title: Author(s): Robert N.
Lieberman/RLL; Distribution: /NIC JCN; Sub=Collections: SRI=ARC NIC;
Clerk: RLL;

New EXEC with Ident Checking

I have brought up a new EXEC which implements better control of user
 idents. The philosophy behind these changes are outlined in Jim
 White's description of NLS file privacy (MJOURNAL, 22913, 11w),

All users will notice the the EXEC command of
 @CHANGE IDENT
 has been deleted. Once a user's ident has been set-up at login time,
 he (she) cannot change that ident,

Single User Directories

All single user directories have one ident associated with that
 directory. At login time, the EXEC will infer the user's ident based
 upon directory name. The user does not need to do anything to have
 her ident set-up,

Multiple User Directories

Multiple user directories have several idents associated with them,
 and therefore the EXEC will query the user as to her specific ident,
 with the text:
 "Specify IDENT to use NLS, CR to ignore, IDENT = "

The user may not specify an ident, and simply type a CR. However,
 if the user does wish to use NLS, she must type in her ident. The
 EXEC will then check that ident for proper access to the login
 directory. If the ident is correct, the EXEC will store it for later
 use by NLS. If the ident is illegal for that directory, the EXEC
 will requery the user,

If you encounter any problems with this EXEC, or (more likely) find
 that you have not been included in the ident list for a directory
 which you use, please SNDMSG to FERGUSON,

WRF 13-MAY-74 02:49 22967

New EXEC with Ident Checking

(J22967) 13-MAY-74 02:49; Title: Author(s): William R. (Ferg)
Ferguson/WRF; Distribution: /SRI=ARC; Sub=Collections: SRI=ARC; Clerk:
WRF;
Origin: (FERGUSON, EXEC-NEWS,NLS;2,), 13-MAY-74 02:46 WRF ;

JEW 13-MAY-74 09:54 22968

Request for Info About <USING>

Dave-- Please read (22898,) and respond for directory <USING>.
Thanks,

1

JEW 13=MAY=74 09:54 22968

Request for Info About <USING>

(J22968) 13=MAY=74 09:54; Title: Author(s): James E. (Jim)
White/JEW; Distribution: /DHC(* action *) ; Sub=Collections: SRI=ARC;
Clerk: JEW;

JEW 13-MAY-74 09:59 22969

Request for Info About Directory <HELP>

Jean-- Would you please read (22898,) and respond for directory
<HELP>. Thanks,

1

JEW 13-MAY=74 09:59 22969

Request for Info About Directory <HELP>

(J22969) 13-MAY=74 09:59; Title: Author(s): James E. (Jim)
White/JEW; Distribution: /JI(* action *) ; Sub=Collections: SRI=ARC;
Clerk: JEW;

JEW 13-MAY-74 10102 22970

Request for Info About Directory <ENERGY>

Dirk== would you please read (22898,) and respond for directory
<ENERGY>. Thanks,

1

JEW 13-MAY-74 10:02 22970

Request for Info About Directory <ENERGY>

(J22970) 13-MAY-74 10:02; Title: Author(s): James E. (Jim)
White/JEW; Distribution: /DVN(* action *) ; Sub=Collections: SRI=ARC;
Clerk: JEW;

Re (22954,)

After doing that command summary, it's very clear to me that information arranged for on-line access has to go through a time-consuming process of radical reorganization to be useful for hardcopy documentation. I did not have time to go as far as I would have liked with the command summary I did extract (for usability). So a user program would be very helpful. I could probably be of help in describing the general process for extracting a document from particular sections of the database.

1

JMB 13-MAY-74 10:03 22971

Re (22954,)

(J22971) 13-MAY-74 10:03; Title: Author(s): Jeanne M. Beck/JMB;
Distribution: /RWW(* info-only *); Sub=Collections: SRI=ARC; Clerk:
JMB;

JEW 13=MAY=74 10:08 22972

Request for Info About Directory <CATALOG>

Beau == Would you please read (22898,) and respond for directory
<CATALOG>, Thanks,

1

JEW 13-MAY-74 10:08 22972

Request for Info About Directory <CATALOG>

(J22972) 13-MAY-74 10:08; Title: Author(s): James E. (Jim)
White/JEW; Distribution: /BAH(* action *); Sub=Collections: SRI=ARC;
Clerk: JEW;

SRL 13-MAY-74 10:12 22973

Response to Feedback Received as of 5/13/74

This is directed to DCE, DVN, KIRK, RLL, and JMB,

Response to Feedback Received as of 5/13/74

Response to feedback received as of 5/13/74

Kirk =

In response to your problem with loading and modifying a busy file, (22008,) a feature of the load busy command is that you must update the file immediately before modifying it in any way,

<BW> and <BC> now work in the Jump to Return command as they should,

<,d> now works

Delete Invisible now works no matter where you bug,

Doug =

Load File was not designed to accept viewspecs, Jump to Link must be used to enter viewspecs,

You can bug a filename in the Load File command but it must be in the form (FILENAME,),

When you split the screen the contents of your screen will be in the half where you last bug,

There is a problem with viewspecs and jumping around in a file that the programmers are working on,

I'll report the suggestion about RPT in the Jump Word and Content commands and your problem with Compile Contentanalyzer with a bugged pattern,

Dirk =

*R now works in TNLS

Compile Program Li0 will now begin compiling at the address specified,

Sort Directory now works properly,

Insert word problems on the Lineprocessor have been fixed,

It seems that you can delete unattached partial copies - you must have bugged the wrong place,

The O PN, xu .. ,7' p p B @ ?_oww= ?_oww= ?_og(etc, that

Response to Feedback Received as of 5/13/74

you see at times at the top of your screen is a core dump,
Elizabeth says to enjoy it,

1c6

Robert =

1d

you should now be able to type a question mark first thing upon
logging in and get the right information,

1d1

Your problems with Help (22792,) were temporary and have now
been fixed,

1d2

The problems with show directory in (22917,) are fixed as best
as possible. Because of a bug in the CML you have to specify
For File as your last option. All of the command accepts are
necessary,

1d3

Jean Beck =

1e

The Reset Feedback command has been removed from the Editor
subsystem,

1e1

Secondary recognition modes aren't allowed except with the
primary recognition mode expert,

1e2

SRL 13=MAY=74 10:12 22973

Response to Feedback Received as of 5/13/74

(J22973) 13=MAY=74 10:12; Title: Author(s): Susan R. Lee/SRL;
Distribution: /SRI=ARC(* info-only *) ; Sub=Collections: SRI=ARC;
Clerk: SRL;

Jump to return bug

Jump to Return does not remember the viewspecs,

1

KIRK 13-MAY-74 13:12 22974

Jump to return bug

(J22974) 13-MAY-74 13:12; Title: Author(s): Kirk E. Kelley/KIRK;
Distribution: /BUGS(* action *) ; Sub=Collections: SRI=ARC BUGS;
Clerk: KIRK;

NIC Mailing Labels and Membership Lists; NEW NLS L10 User Program

```

FILE niclyst % (L10,) (KUDLICK,NICLYST,) %
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, t1stid, tpstid, new;
3a
  DECLARE automatic, aflag, dflag;
3b
  DECLARE end, outstid, prtstid;
3c
  DECLARE labflag, memflag, lnct;
3d
  DECLARE STRING nothing[5], grpident[10], id[10], coor[10],
  title[100];
3e
  DECLARE STRING err[300], entry[2000];
3f
  DECLARE STRING xlit[2000], lit2[200], auxlit[10]; %no longer
  global%
3g
  DECLARE STRING blankcard = "
  ";
3h
% main control %
4
  (niclyst) PROCEDURE;
4a
    % local declarations %
4a1
    LOCAL
4a1a
      i, % loop index %
4a1a1
      j, % pgct for individuals %
4a1a2
      orgp; % TRUE when current ident is orgzn or group %
4a1a3
    LOCAL TEXT POINTER tp1, tp2, tp3, tp4, z1, z2;
4a1b
    LOCAL STRING memf[50], labf[50];
4a1c

```

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

```

LOCAL STRING inpstr[400]; % input collection %           4a1d
LOCAL STRING nxtident[10], exp[300], noexp[300], name[300]; 4a1e
LOCAL STRING grplist[6000];                             4a1f
% initialization %                                       4a2
  automatic = FALSE;                                     4a2a
  *nothing* = NULL;                                     4a2b
  *inpstr* = NULL;                                       4a2c
% open identfile %                                       4a3
  IF NOT (idstid,stfile = open(0, jfname($"identfile"))))THEN 4a3a
    err($" Unable to open IDENT FILE, ");
  ON SIGNAL ELSE                                         4a3b
    BEGIN                                               4a3b1
      close(idstid,stfile);                             4a3b2
    RETURN;                                             4a3b3
  END;                                                 4a3b4
% get run parameters %                                   4a4
  % get idents from user %                               4a4a
    crlf();                                             4a4a1
    typeas ($" Manual Ident Entry Mode? (Y or N) "); 4a4a2
    *xlit* = NULL;                                       4a4a3
    txtlit($xlit);                                       4a4a4
    IF *xlit*[1] = 'Y' OR *xlit*[1] = 'y' THEN         4a4a5
      BEGIN % collect ident list %                     4a4a5a
        crlf();                                         4a4a5b
        typeas($"Idents = ");                          4a4a5c

```

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

```

        identlist(sinpstr, idstid, stfile);          4a4a5d
    END                                             4a4a5e
ELSE                                             4a4a6
    BEGIN                                         4a4a6a
    crlf();                                       4a4a6b
    typeas(s"Automatic Mode ...");              4a4a6c
    % set up the input string to specific values % 4a4a6d
        *inpstr* = SP, "TIPG", SP, "USERG", SP, "SERVERG",
        SP, "ASSOCG";                             4a4a6d1
    typeas(sinpstr);                              4a4a6e
    automatic = TRUE;                             4a4a6f
    END;                                           4a4a6g

% determine whether LABELS are desired %         4a4b
    (what); crlf();                               4a4b1
    typeas(s" Do you want LABELS ? (Y or N) ");  4a4b2
    *xlit* = NULL;                                4a4b3
    txtlit(sxlit);                                4a4b4
    IF *xlit*[1] = 'Y' OR *xlit*[1] = 'y' THEN labfil($labf) 4a4b5
    ELSE labflg = FALSE;                          4a4b6

% determine whether MEMLISTS are desired %       4a4c
    crlf();                                       4a4c1
    typeas(s" Do you want MEM=LISTS ? (Y or N, "); 4a4c2
    *xlit* = NULL;                                4a4c3
    txtlit(sxlit);                                4a4c4

```



```

        IF *xlit*[1] = 'Y' OR *xlit*[1] = 'y' THEN memfil(smef)      4a4c5
        ELSE memflg = FALSE;                                         4a4c6

% should we proceed ? %                                           4a4d
        IF (NOT memflg AND NOT labflg) THEN                          4a4d1
            BEGIN                                                    4a4d1a
                crlf();                                              4a4d1b
                typeas($" Say YES to LABELS or MEM=LISTS or both !"); 4a4d1c
                GOTO what;                                           4a4d1d
            END;                                                      4a4d1e

% initialization before getting idents from identfile %           4a5
        orgp = TRUE; % for page eject when doing individ, not grp % 4a5a
        crlf();                                                       4a5b
        FIND SF(*inpstr*)"Z1;                                         4a5c
        IF (memflg AND automatic) THEN                                4a5d
            BEGIN % get ready for title page %                       4a5d1
                *title* = EOL, "          ", "NETWORK INDIVIDUALS", EOL, 4a5d2
                EOL;
                prtstid = cis(prtstid, $blankcard, succdir);        4a5d3
                hdstid = prtstid;                                     4a5d4
                lnct = 1;                                           4a5d5
            END;                                                      4a5d6

% main processing loop %                                           4a5e
        (mainloop);                                                 4a5f
        LOOP                                                         4a5g

```



```

BEGIN                                                                    4a5g1
% initialization %                                                         4a5h
  *entry* = NULL; % ident file entry %                                    4a5h1
  *xlit* = NULL;                                                           4a5h2
  *lit* = NULL;                                                            4a5h3
  *coor* = NULL; % grp coordinator ident, if any %                       4a5h4
% get next ident from input string %                                       4a5i
  IF NOT FIND z1 > s(SP/',' ) "z1 is(LD/'=) "z2 THEN EXIT
  LOOP;                                                                    4a5i1
  *nxtident* = z1 z2;                                                       4a5i2
  z1[1] = z2[1];                                                            4a5i3
  *grpident* = NULL;                                                        4a5i4
  *grplist* = NULL;                                                        4a5i5
% see if ident is valid %                                                 4a5j
  IF NOT ckident($nxtident, $entry, idstid,$file) THEN                    4a5j1
    BEGIN                                                                    4a5j1a
      *xlit* = *nxtident*, " ... skipping this ident:
      invalid";                                                              4a5j1b
      typeas($xlit);                                                         4a5j1c
      REPEAT LOOP;                                                           4a5j1d
    END;                                                                      4a5j1e
  *xlit* = *nxtident*, EOL;                                                 4a5j2
  typeas($xlit);                                                            4a5j3
  IF NOT automatic THEN                                                    4a5j4
    BEGIN                                                                    4a5j4a

```

```

IF deleted(sentry) THEN                                4a5j4b
  BEGIN                                                4a5j4b1
    typeas(s" ... SPECIAL (deleted)");                4a5j4b2
    REPEAT LOOP;                                       4a5j4b3
    END;                                                4a5j4b4
  END;
END;                                                    4a5j4c

% see if ident is organization or group or individual % 4a5k
IF orgrptst(sentry,0) THEN % TRUE if org/group %      4a5k1
  BEGIN                                                4a5k1a
    *grpidnt* = *nxtident*;                             4a5k1b
    *grplist* = *nxtident*;                             4a5k1c
    orgp = TRUE;                                        4a5k1d
    IF (memflg AND NOT automatic) THEN                 4a5k1e
      BEGIN % get ready for new page on listing %     4a5k1e1
        getinam(sentry, stitle, 0, 0);                 4a5k1e2
        astruc(stitle);                                 4a5k1e3
        *title* = EOL, "          ", *title*, Sp, "MEMBERSHIP
LIST", EOL, EOL;                                     4a5k1e4
        prtstid = cis(prtstid, sblankcard, suedir);    4a5k1e5
        ndstid = prtstid;                              4a5k1e6
        linct = 1;                                     4a5k1e7
        % find group coordinator %                     4a5k1e8
        geticord(sentry, scoor, 0,0);                  4a5k1e8a
      END;                                              4a5k1e9
    END;
  END;

```

```

% expand group or orgzn ident ...
  ... and build records for output file(s) %           4a5k1f
  iexpmdk (sgrplist, idstid, stfile);                   4a5k1f1
  crlf();                                               4a5k1g
END % of processing for group or orgzn ident %         4a5k1h

ELSE % individual ident %                               4a5k2
  BEGIN                                               4a5k2a
  % create label info %                               4a5k2b
    IF labflg THEN process($nothing);                 4a5k2b1
  % create formatted print line %                     4a5k2c
    IF memflg THEN                                    4a5k2c1
      BEGIN                                           4a5k2c1a
        IF orgp THEN % start new output page %       4a5k2c1b
          BEGIN                                       4a5k2c1b1
            orgp = FALSE;                             4a5k2c1b2
            (donew);                                  4a5k2c1b3
            *title* = ', SP, SP, EOL;                 4a5k2c1b4
            prtstid = cis(prtstid, stitle, succdir);  4a5k2c1b5
            *title* = SP, SP, " individual idents", EOL; 4a5k2c1b6
            prtstid = cis(prtstid, stitle, succdir); 4a5k2c1b7
            j = 5;                                     4a5k2c1b8
          END;                                         4a5k2c1b9
          j = j - 1;                                   4a5k2c1c
        IF j < 0 THEN GOTO donew;                     4a5k2c1d
        % format and output this entry for mem=list % 4a5k2c1e

```

```

                list(sentry, snxtident);                4a5k2c1e1
            END;                                        4a5k2c1f
        END; % of processing for individual ident %    4a5k2d
    % end of main processing loop %                    4a5l
        END; % of main loop %                          4a5l1
% finished==clean up %                                4a6
    IF (memflg AND automatic) THEN stlist();          4a6a
    crlf();                                           4a6b
    typeas($"Processing Finished Normally ");          4a6c
    IF labflg THEN                                    4a6d
        BEGIN                                        4a6d1
            close(outstid,stfile);                    4a6d2
            astruc($labf);                              4a6d3
            *labf* = EOL, " labels file is ", *labf*;  4a6d4
            typeas($labf);                              4a6d5
        END;                                          4a6d6
    IF memflg THEN                                    4a6e
        BEGIN                                        4a6e1
            close(prtstid,stfile);                      4a6e2
            astruc($memf);                              4a6e3
            *memf* = EOL, " mem=list file is ", *memf*; 4a6e4
            typeas($memf);                              4a6e5
        END;                                          4a6e6
    close(idstid,stfile);                             4a6f

```


NIC Mailing Labels and Membership Lists; NEW NLS L10 User Program

RETURN END,

4a7

(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), sptr);

5e

intids(0);

5f

% the "aflg" flag allows the automatic-mode groups == TIPG USERG
SERVERG and ASSOCCG == to be processed despite the fact that they
are marked "Comments: SPECIAL" in the identfile. Without this
maneuver, these group idents would be bypassed by GETMDKIDS since
SPECIAL means "deleted" %

5g

IF automatic THEN aflg = TRUE ELSE aflg = FALSE;

5g1

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(sptr, sentry, 0, idfnum) THEN EXIT LOOP;

5h3a

IF dflg THEN REPEAT LOOP; % that ident was deleted %

5h3b

getiid(sentry, 0, sz1, sz2);

5h3c

auxlit = " ", z1 z2;

5h3d

typeas(sauxlit);

5h3e

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

```

% 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 [OB, 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 snothing ELSE
sgrpident);                                           5h5a
% format statement for printed mem=list %           5h6
  IF memflg THEN list(sentry, sauxlit);               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 idents%                                  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(sidstr, sinfostr, 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(sinfostr) 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; % only true first time thru getmdkids % 6h1b3
            IF orgrptst(sinfostr, 0) THEN                       6h1c
                BEGIN                                          6h1c1
                    expchr = TRUE;                             6h1c2
                    IF expchr THEN                             6h1c3
                        BEGIN                                    6h1c3a
                            getmem(sinfostr, 0, sdstpnr, 0);  6h1c3b
                            IF FIND dstpnr =EOL %membership list present% THEN 6h1c3c
                                BEGIN                            6h1c3c1
                                    pushids(&ptr);              6h1c3c2
                                    dstpnr = gpstid;            6h1c3c3
                                    FIND dstpnr =ptr;           6h1c3c4
                                    RETURN(getmdkids(&ptr, &astr, infotype, idfnum)); 6h1c3c5
                                END;                               6h1c3c6
                            END;
                        END;
                    END;
                END;
            END;
        END;
    END;

```

```

        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, xvs; 7a
    LOCAL iptstid, ihdstid, tlstid, kmax; %for brief ident lists% 7b
    LOCAL TEXT POINTER s1,sx, t1,tx;       7c
    LOCAL STRING skipout[25], ttitle[500], tpstrg[1000], t1strg[1000]; 7d
    % sort membership list %               7e
    tpstid = hdstid;                       7e1
    tlstid = prtstid;                      7e2
    sortgrp(hdstid, tlstid);               7e3
    % set up stid's for brief ident listing when in automatic mode % 7f
    IF automatic THEN                      7f1
        BEGIN                              7f1a
            iptstid = getail(tlstid);       7f1b
            iptstid = cis(iprtstid, sblankcard, suedir); 7f1c
            ihdstid = iptstid;              7f1d
            kmax = 40;                      7f1e

```



```

END;
7f1f

% initialize to insert page breaks for output quickprint usage %
7g
new = TRUE;
7g1
pgmax = 6;
7g2
xvs = 0; % needed only to delete statement in new nls %
7g3

% 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, succdir); % force new page %
7l4
    tpstid = cis(tpstid, stitle, succdir); % 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, stytle, succdir);          719e
    tpstid = cis(tpstid, sblankcard, succdir);     719d
    new = FALSE;                                   719e
    pgct = pgct + 1;                               719f
    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) "s1;                    7112d
        IF NOT FIND s1 > [EOL] "sx THEN FIND SE(tpstid) "sx; 7112e

% when in automatic mode, eliminate duplicate entries and ... 7113
output a brief list (idents and names) of each individ %
    IF automatic THEN                            7113a
        BEGIN                                    7113a1
            % set up string for comparison with subsequent entry % 7113a2
            *tpstrg* = s1 sx;                    7113a2a
            % if next entry is equal to present one, delete it % 7113a3
            LOOP % innermost loop %              7113a4
                BEGIN                              7113a4a
                    t1stid = getsuc(tpstid);     7113a4b

```

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

```

        IF t1stid = indstid THEN EXIT LOOP;                                7113a4c
        FIND SF(t1stid) "t1;                                             7113a4d
        IF NOT FIND t1 > [EOL] "tx THEN FIND SE(t1stid) "tx;          7113a4e
        *tlstrg* = t1 tx;                                               7113a4f
        IF *tpstrg* # *tlstrg* THEN EXIT LOOP;                          7113a4g
        cdelsta(t1stid, 0, sxvs);                                       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, stlstrg, succdir);                    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

```

NIC Mailing Labels and Membership Lists; NEW NLS L10 User Program

```

        ST sl 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, sskipout, suedir); 7115c3c
                    cflg = FALSE;                     7115c3d
                END;                                    7115c3e
            EXIT LOOP; % exit inner loop %            7115c4
        END;                                           7115c5
    END;
% play this till group exhausted %                   7115d
    linct = linct + pgct;                              7116a
    IF linct > 0 THEN REPEAT LOOP                    7116b
    ELSE EXIT LOOP; % exit main loop %               7116c

```

NIC Mailing Labels and Membership Lists; NEW NLS L10 User Program

```

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 = gettail(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, sskipout, succdir);        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 = iptstid;                               7o1b1
        iptstid = ihdstid;                              7o1b2
        sortgrp(ihdstid, itlstid);                      7o1b3
      % initialize page and line counts %                7o1c
        k = 0;                                          7o1c1
        m = 0;                                          7o1c2

```



```

*skipout* = ', SP, SP, EOL;                                701e3
*title* = " ", "NIC IDENTIS", EOL;                        701e4
% LOOP TO INSERT PAGE AND LINE BREAKS IN BRIEF IDENT LISTING
%                                                         701d
LOOP                                                         701e
    BEGIN                                                  701e1
        % is new page needed %                             701e2
            IF k = 0 THEN                                   701e2a
                BEGIN                                      701e2a1
                    k = kmax;                               701e2a2
                    iptstid = cis(iprtstid, sskipout, succdir); 701e2a3
                    iptstid = cis(iprtstid, stitle, succdir); 701e2a4
                END;                                        701e2a5
            % is blank line needed (every five lines) %    701e3
                IF m = 0 THEN                               701e3a
                    BEGIN                                   701e3a1
                        m = 5;                               701e3a2
                        iptstid = cis(iprtstid, s" ", succdir); 701e3a3
                    END;                                    701e3a4
                % keep track of where we are %              701e4
                    iptstid = getsuc(iprtstid);            701e4a
                    IF iptstid.stpsid = origin THEN EXIT LOOP; 701e4b
                    k = k+1;                                 701e4c
                    m = m+1;                                 701e4d
                END; % of LOOP for page/line-break insertion % 701e5

```


NIC Mailing Labels and Membership Lists; NEW NLS L10 User Program

```

prtstid, stpsid = origin;                                701f
prtstid = getail(getsub(prtstid));                       701g
END; % of processing brief ident listing %              701h

RETURN;                                                  7p

END.

(process) PROC (groupid); % output the cards for the entry in
"entry", inserting the group ident "groupid" if supplied, %
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 < SPT 1$NP "t8 > THEN FIND t7 > 24$24CH
        "t8;                                          8f3f3b
        name,L = t8[1];                               8f3f3c
        REPEAT LOOP;                                  8f3f3d
        END;                                          8f3f3e
    END;                                              8f3f4
    crlf();                                           8f3g
    typeas($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

```

NIC Mailing Labels and Membership Lists; NEW NLS L10 User Program

```

      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 1$NP "t2 > "t1 THEN EXIT LOOP;
                                                                    9e3d3b
    difference = t2[1] - current[1];
                                                                    9e3d3c
    REPEAT LOOP;
                                                                    9e3d3d
    END;
                                                                    9e3d3e
  END;
                                                                    9e3d4
  END;
                                                                    9e3e

  *xlit* [column TO column + difference] = current t2;
                                                                    9e3f
  current[1] = t1[1];
                                                                    9e3g
  column = column + 30;
                                                                    9e3h
  IF column > 31 OR current[1] >= leng OR end THEN EXIT LOOP;
  % done with a whole card %
                                                                    9e3i
  END;
                                                                    9e4

  END;
                                                                    9f
  %truncate off any extra info in last field%
                                                                    9g
  *xlit*[61 TO 80] = *blankcard*[61 TO 80];
                                                                    9h
  xlit,L = 60;
                                                                    9i
  outstid = cis(outstid, xlit, succdir);

```


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], netaddr[200], name[200],
  phonestr[100], affil[100]; 10d
  % Initialization, % 10e
    FIND "sf; 10e1
  % Get raw info for formatting, % 10f
    % Full name % 10f1
      getinam(sentr, sname, 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(sname); 10f1b4
    % Network address % 10f2
      *netaddr* = NULL; 10f2a
      getinma(sentr, snetaddr, 0, 0); 10f2b
      FIND SF(*netaddr*) "he; 10f2c
      WHILE (FIND he [""] "he "hf _hf) DO ST hf he = NULL; 10f2d

```


NIC Mailing Labels and Membership Lists; NEW NLS L10 User Program

```

IF getihost(sentr, 0, shf, she) 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(sentr, saddr, 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 [OB, 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 ";                             10f3e
    FIND SF(*addr*) *adf SE(*addr*) *ade;                           10f3d
% Network Organization %                                           10f4

```

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

```

getiorg (sentr, saffil, 0, 0);          10f4a
astruc(saffil); %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, " ";
END;
% Now next line: Beginning of address plus phone number %
FIND adf > ([EOL] "adx _adx/ [ENDCHR] "adx);
WHILE adx[1] = adf[1] >= 32 DO FIND adx < [SP] "adx;
diddle = out,L;
*out* = *out*, adf adx;
FOR i = out,L + 1 UP UNTIL > diddle + 34 DO *out*[i] = SP;
addmor = TRUE;
% If phone put out first line of phone number %
IF phoneflag = getiphone(sentr, 0, stf, ste) THEN
BEGIN
*phonestr* = tf te;
FIND SF(*phonestr*) "tf SE(*phonestr*) "te "tx;
WHILE tx[1] = tf[1] > 23 DO FIND tx < [SP] "tx;
*out* = *out*, tf tx;
END;
*out* = *out*, EOL;
% The remaining lines %
WHILE ((addmor = (FIND adx > NP)) OR (phoneflag OR orgflg))
DO
BEGIN
diddle = out,L;
IF addmor THEN

```

NIC Mailing Labels and Membership Lists; NEW NLS L10 User Program

```

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[i] = adf[i] > 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

```


NIC Mailing Labels and Membership Lists; NEW NLS L10 User Program

```

        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

```


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

```

    linct = linct + 1;                                10h2
    prtstid = cis(prtstid, sout, succdir);            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

(labf11) 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(s" Output file for Labels = ");                       12d3
            *strg* = NULL;                                                12d4
            txtlit($strg);                                                12d5
        END
    END

```

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

```

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, sstrg) THEN              12d9
BEGIN                                                12d9a
typeas(s" (old file) CONFIRM ");                    12d9b
IF input() # CA THEN REPEAT LOOP;                   12d9c
% using old file = position to tail %               12d9d
    outstid = gettail(getsub(outstid));              12d9d1
EXIT LOOP;                                          12d9e
END                                                  12d9f
ELSE                                                12d10
BEGIN                                                12d10a
(newfile);                                          12d10b
typeas(s" (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*) > '< 1s(LD/'=) '> 1s(LD/'=) *tp1
    ([',] *tp2 = tp2 / TRUE *tp2) THEN              12d10f1
        BEGIN                                        12d10f1a
            typeas(s" cant understand filename: "); 12d10f1b
            typeas(ssstrg);                          12d10f1c
            REPEAT LOOP;                             12d10f1d

```

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

```

        END;                                12d10f1e
        *strg* = SF(*strg*) tp1, ".NLS", tp2 SE(*strg*);    12d10f2
    IF NOT outstid,stfile = opwk(0, sstrg) 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    13
mem=lists%
    LOCAL TEXT POINTER tp1, tp2;          13a
    REF strg;                              13b
    memflg = TRUE;                          13c
    LOOP                                    13d
        BEGIN                                13d1
            erlf();                            13d2
            typeas($" Output file for Mem=Lists = ");    13d3
            *strg* = NULL;                      13d4
            txtlit(ssstrg);                    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, sstrg) THEN    13d9

```

```

BEGIN                                                    13d9a
typeas($" (old file) CONFIRM ");                        13d9b
IF input() # CA THEN REPEAT LOOP;                       13d9c
% using old file = position to tail %                  13d9d
    prtstid = gettail(getsub(prtstid));                 13d9d1
EXIT LOOP;                                              13d9e
END                                                      13d9f
ELSE                                                    13d10
    BEGIN                                              13d10a
        (newfil);                                     13d10b
        typeas($" (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
    
```


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

```

        END;                                13d10f
    END;                                    13d11
RETURN;                                    13e
END.                                        13f

%conversion to new nls: The following six procedures were copied from
the old nls system into this file in order to resolve all
references. These procedures are not global under the new nls,%      14
(identlist) PROC(astr, fileno); %read a list of idents into astr%    15
    %Read and verify a list of idents into astr...terminated by CA%    15a
LOCAL idfile;                              15b
REF astr;                                   15c
%open file, if necessary%                  15d
    IF fileno = 0 THEN                      15d1
        BEGIN                               15d1a
            idfile = 0;                     15d1b
            ON SIGNAL ELSE sigclose(idfile := 0); 15d1c
            idfile = open(0, jfname($"identfile")); 15d1d
        END                                 15d1e
    ELSE idfile = fileno;                   15d2
*astr* = NULL;                             15e
DO                                          15f
    BEGIN                                   15f1
        rident(&astr, 0, idfile);          15f2
        IF curchr = '( THEN %comment%      15f3
            BEGIN                           15f3a

```


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

```

*xlit* = NULL;                                15f3b
echoit();                                     15f3c
rdlit($xlit, scmntd1); %read comment%        15f3d
echoff();                                     15f3e
*astr* = *astr*, '(', *xlit*, ") ";          15f3f
IF lookc() = CA THEN                           15f3g
    BEGIN                                       15f3g1
        echon();                               15f3g2
        input();                               15f3g3
        echoff();                             15f3g4
    END;                                       15f3g5
END                                             15f3h
ELSE                                           15f4
    BEGIN                                       15f4a
        IF *astr*[astr,L] NOT= SP THEN *astr* = *astr*, SP;
        %separate idents by space%            15f4b
        IF curchr = 'Y OR curchr = 'N THEN    15f4c
            typeas($"Please continue, Ident = "); 15f4c1
        END;                                   15f4d
    END                                         15f5
UNTIL curchr = CA;                             15g
WHILE *astr*[astr,L] = SP DO BUMP DOWN astr,L; 15h
IF fileno = 0 THEN close(idfile := 0);        15i
RETURN END,                                    15j

```

NIC Mailing Labels and Membership Lists; NEW NLS L10 User Program

```

(rident) PROCEDURE (astr, infostr, fileno); %Read an IDENT from the
user%
16

%This procedure reads and verifies an ident, and appends it to the
string astr.%
16a

%If fileno is 0 then, the ident file will be opened and closed in
lower level routines. If infostr is non-zero, then the contents
of the entry will be returned in the string pointed to by infostr%
16b

%-----%
16c

LOCAL breakchar;
16d

LOCAL STRING idstr(30);
16e

REF astr;
16f

LOOP
16g

    BEGIN
16g1

        *idstr* = NULL;
16g2

        CASE breakchar = tild(sidstr) OF
16g3
            = '?; %help the user%
16g3a

                BEGIN
16g3a1

                    typeas($"
                    Please type one of the following:
                    ");
16g3a2

                    typeas($"an Ident, CA to terminate the list, CD to abort,
                    ",LastNameQuery, ",BeginLastNameQuery",",",
                    "[NameStringQuery]",
                    ");
16g3a3

                REPEAT LOOP;
16g3a4

            END;
16g3a5

        ENDCASE
16g3b

            IF idstr,L THEN
16g3b1

                IF NOT tiptid(sidstr, infostr, fileno) THEN
16g3b1a

```

NIC Mailing Labels and Membership Lists; NEW NLS L10 User Program

```

BEGIN                                                    16g3b1a1
  crlf();                                                16g3b1a2
  IF astr,L > 0 THEN                                     16g3b1a3
    BEGIN                                                16g3b1a3a
      prompt($"
      Idents accumulated so far: ");                    16g3b1a3b
      prompt(&astr);                                     16g3b1a3c
      prompt($"
      Next Ident = ");                                  16g3b1a3d
    END                                                  16g3b1a3e
  ELSE prompt($"
  Please continue, Ident = ");                          16g3b1a4
  REPEAT LOOP;                                         16g3b1a5
  END                                                  16g3b1a6
ELSE                                                    16g3b1b
  BEGIN                                                16g3b1b1
    astruc($idstr);                                     16g3b1b2
    *astr* = *astr*, *idstr*;                          16g3b1b3
    RETURN(curchr); %tiptid may have read more text ==
    hence a new break char%                            16g3b1b4
  END                                                  16g3b1b5
  ELSE RETURN(breakchar);                              16g3b2
END;                                                    16g4
END;                                                    16h
(tiptid) PROCEDURE(idstr, infostr, fileno); %prgroup Ident Queries% 17
  LOCAL expchr, type;                                  17a

```

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

```

LOCAL TEXT POINTER tptr1, tptr2;                                17b
LOCAL STRING work[500];                                        17c
REF idstr;                                                    17d
IF idstr,L = empty THEN RETURN(TRUE);                          17e
CASE *idstr*[1] OF                                           17f
  =',: %lastname search%                                       17f1
    BEGIN                                                       17f1a
      IF curchr NOT= CA THEN                                     17f1b
        BEGIN                                                   17f1b1
          *idstr* = *idstr*, curchr;                             17f1b2
          txlit(&idstr);                                         17f1b3
        END;                                                     17f1b4
      *idstr* = *idstr*[2 TO idstr,L];                           17f1c
      CCPOS SE(*idstr*); %check for incomplete name%           17f1d
      IF READC = ', THEN                                         17f1e
        BEGIN %partially specified last name, remove trailing
          periods%                                              17f1e1
          BUMP DOWN idstr,L;                                     17f1e2
          WHILE READC = ', DO BUMP DOWN idstr,L;               17f1e3
          type = idchr;                                         17f1e4
        END                                                       17f1e5
      ELSE type = lname;                                         17f1f
      stnamcap(&idstr);                                         17f1g
      RETURN( idflsearch(type, &idstr, infostr, fileno) );    17f1h
    END;                                                         17f1i

```


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

```

= 'I; 17f2
  BEGIN 17f2a
  IF curchr NOT= CA THEN 17f2b
    BEGIN 17f2b1
    *idstr* = *idstr*, curchr; 17f2b2
    txtlit(&idstr); 17f2b3
    END; 17f2b4
  FIND SF(*idstr*) CH *tptr1 ( ['I] *tptr2 =tptr2 / 17f2e
  SE(*idstr*) *tptr2);
  *idstr* = tptr1 tptr2; 17f2d
  RETURN( idflsearch(litstr, &idstr, infostr, fileno) ); 17f2e
  END; 17f2f
= '&, = '"; 17f3
  BEGIN 17f3a
  expchr = *idstr*[1]; 17f3b
  *idstr* = *idstr*[2 TO idstr,L]; 17f3c
  IF NOT infostr THEN infostr = swork; 17f3d
  END; 17f3e
=CD; GOTO STATE; 17f4
= SP; %deblank front of string% 17f5
  BEGIN 17f5a
  *idstr* = *idstr*[2 TO idstr,L]; 17f5b
  REPEAT CASE; 17f5c
  END; 17f5d
ENDCASE expchr = FALSE; 17f6

```



```

IF NOT infostr THEN                                     17g
  BEGIN                                               17g1
    IF NOT oldid(&idstr, fileno) THEN %No such ident% 17g2
      BEGIN                                           17g2a
        tiderr();                                     17g2b
        RETURN(FALSE);                               17g2c
      END;                                           17g2d
    END                                               17g3
  ELSE                                               17h
    BEGIN                                           17h1
      IF NOT ckident(&idstr, infostr, fileno) THEN %No such ident% 17h2
        BEGIN                                           17h2a
          tiderr();                                     17h2b
          RETURN(FALSE);                               17h2c
        END;                                           17h2d
      IF expchr AND NOT orgrptst(swork,0) THEN        17h3
        BEGIN                                           17h3a
          typeas($" Not a group or organization
          ");                                           17h3b
          RETURN(FALSE);                               17h3c
        END;                                           17h3d
      END;                                           17h4
    IF expchr THEN *idstr* = expchr, *idstr*;        17i
    RETURN(TRUE);                                     17j
  END;                                               17k

```

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

```

(tiid) PROCEDURE(string); %collect an Ident literal from the user%      18
    LOCAL breakchar;                                                    18a
    IF inprompt THEN prompt("$I: ");                                     18b
    echoff();                                                            18c
    WHILE lookc() = SP DO todeo( input() );                             18d
    echolt();                                                            18e
    rdlit(string, sidntdel ; breakchar);                                18f
    echoff();                                                            18g
    RETURN(breakchar);                                                  18h
    END,                                                                  18i

(tiderr) PROCEDURE; %error routine%                                     19
    todeo(" ");                                                         19a
    todeo("?");                                                         19b
    crlf();                                                             19c
    RETURN;                                                             19d
    END,                                                                  19e

FINISH                                                                    20
                                                                    21

```

NIC Mailing Labels and Membership Lists; NEW NLS L10 User Program

(J22975) 13-MAY-74 14:33; Title: Author(s): Michael D. Kudlick/MDK;
Distribution: /MDK; Sub=Collections: SRI=ARC; Clerk: MDK;
Origin: (KUDLICK, NEWADDR,NLS;12,), 10-MAY-74 10:48 MDK ;

(MDK) 22975 (MDK) 22975 (MDK) 22975 (MDK) 22975 (MDK) 22975 (MDK) 22975
(MDK) 22975 (MDK) 22975 (MDK) 22975 (MDK) 22975 (MDK) 22975 (MDK) 22975
(MDK) 22975 (MDK) 22975 (MDK) 22975 (MDK) 22975 (MDK) 22975 (MDK) 22975

LABELS / MEM-LIST PROGRAM LISTING

runs in new NLS

```
FILE niclyst % (L10,) (KUDLICK,NICLYST,) %
```

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

```
% global declarations %
```

```
DECLARE idstid, hdstid, tlstid, tpstid, new;
DECLARE automatic, aflag, dflag;
DECLARE end, outstid, prtstid;
DECLARE labflag, memflag, lnct;
DECLARE STRING nothing[5], grpident[10], id[10], coor[10],
title[100];
DECLARE STRING err[300], entry[2000];
DECLARE STRING xlit[2000], lit2[200], auxlit[10]; %no longer global%
DECLARE STRING blankcard = "
";
```

```
% main control %
```

```
(niclyst) PROCEDURE;
```

```
  % local declarations %
```

```
    LOCAL
```

```
      i, % loop index %
```

```
      j, % pgct for individuals %
```

```
      orgp; % TRUE when current ident is orgzn or group %
```

```
    LOCAL TEXT POINTER tp1, tp2, tp3, tp4, z1, z2;
```

```
    LOCAL STRING memf[50], labf[50];
```

```
    LOCAL STRING inpstr[400]; % input collection %
```

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

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

```
  % initialization %
```

```
    automatic ← FALSE;
```

```
    *nothing* ← NULL;
```

```
    *inpstr* ← NULL;
```

```
  % open identfile %
```

```
    IF NOT (idstid.stfile ← open(0, jfname($"identfile"))) THEN
err($" Unable to open IDENT FILE. ");
```

```
    ON SIGNAL ELSE
```

```
      BEGIN
```

```
        close(idstid.stfile);
```

```
        RETURN;
```

```
      END;
```

```
  % get run parameters %
```

```
    % get idents from user %
```

```
      crlf();
```

```
      typeas ($" Manual Ident Entry Mode? (Y or N) ");
```

```
      *xlit* ← NULL;
```

```
      txtlit($xlit);
```

```
      IF *xlit*[1] = 'Y' OR *xlit*[1] = 'y' THEN
```

```
        BEGIN % collect ident list %
```

```
          crlf();
```

```
          typeas($" Idents = ");
```

```
          identlist($inpstr, idstid.stfile);
```

```
        END
```

```
      ELSE
```

```
        BEGIN
```

```

    crlf();
    typeas($"Automatic Mode ...");
    % set up the input string to specific values %
    *inpstr* ← SP, "TIPG", SP, "USERG", SP, "SERVERG",
    SP, "ASSOCG";
    typeas($inpstr);
    automatic ← TRUE;
    END;

% determine whether LABELS are desired %
(what): crlf();
typeas($" Do you want LABELS ? (Y or N) ");
*xlit* ← NULL;
txtlit($xlit);
IF *xlit*/1/ = 'Y OR *xlit*/1/ = 'y THEN labfil($labf)
ELSE labflg ← FALSE;

% determine whether MEMLISTS are desired %
crlf();
typeas($" Do you want MEM-LISTS ? (Y or N) ");
*xlit* ← NULL;
txtlit($xlit);
IF *xlit*/1/ = 'Y OR *xlit*/1/ = 'y THEN memfil($memf)
ELSE memflg ← FALSE;

% should we proceed ? %
IF (NOT memflg AND NOT labflg) THEN
    BEGIN
    crlf();
    typeas($" Say YES to LABELS or MEM-LISTS or both !");
    GOTO what;
    END;

% initialization before getting idents from identfile %
orgp ← TRUE; % for page eject when doing individ, not grp %
crlf();
FIND SF(*inpstr*)↑z1;
IF (memflg AND automatic) THEN
    BEGIN % get ready for title page %
    *title* ← EOL, " ", "NETWORK INDIVIDUALS", EOL, EOL;
    prtstid ← cis(prtstid, $blankcard, succdir);
    hdstid ← prtstid;
    lnct ← 1;
    END;

% main processing loop %
(mainloop):
LOOP
    BEGIN
% initialization %
    *entry* ← NULL; % ident file entry %
    *xlit* ← NULL;
    *lit* ← NULL;
    *ccor* ← NULL; % grp coordinator ident, if any %

% get next ident from input string %

```

```

IF NOT FIND z1 > $(SP/',' ) ↑z1 l$(LD/'- ) ↑z2 THEN EXIT
LOOP;
*nxtident* ← z1 z2;
z1/l/ ← z2/l/;
*grpident* ← NULL;
*grplist* ← NULL;

% see if ident is valid %
IF NOT ckident($nxtident, $entry, idstid.stfile) THEN
BEGIN
*xlit* ← *nxtident*, " ... skipping this ident:
invalid";
typeas($xlit);
REPEAT LOOP;
END;
*xlit* ← *nxtident*, EOL;
typeas($xlit);
IF NOT automatic THEN
BEGIN
IF deleted($entry) THEN
BEGIN
typeas("$" ... SPECIAL (deleted)");
REPEAT LOOP;
END;
END;

% see if ident is organization or group or individual %
IF orgrptst($entry,0) THEN % TRUE if org/group %
BEGIN
*grpident* ← *nxtident*;
*grplist* ← *nxtident*;
orgp ← TRUE;
IF (memflg AND NOT automatic) THEN
BEGIN % get ready for new page on listing %
getinam($entry, $title, 0, 0);
astruc($title);
*title* ← EOL, "          ", *title*, SP, "MEMBERSHIP
LIST", EOL, EOL;
prtstid ← cis(prtstid, $blankcard, SUCDIR);
hdstid ← prtstid;
lnct ← 1;
% find group coordinator %
geticord($entry, $coor, 0,0);
END;
% expand group or orgzn ident ...
... and build records for output file(s) %
iexpmdk ($grplist, idstid.stfile);
crlf();
END % of processing for group or orgzn ident %

ELSE % individual ident %
BEGIN
% create label info %
IF labflg THEN process($nothing);
% create formatted print line %
IF memflg THEN

```

```
BEGIN
IF orgp THEN % start new output page %
BEGIN
  orgp ← FALSE;
  (donew):
  *title* ← ', SP, SP, EOL;
```



```

        prtstid ← cis(prtstid, $title, succdir);
        *title* ← SP, SP, " individual idents", EOL;
        prtstid ← cis(prtstid, $title, succdir);
        j ← 5;
        END;
    j ← j - 1;
    IF j < 0 THEN GOTO donew;
    % format and output this entry for mem-list %
    list($entry, $nxtident);
    END;
END; % of processing for individual ident %

% end of main processing loop %
END; % of main loop %

% finished--clean up %
IF (memflg AND automatic) THEN stlist();
crlf();
typeas($"Processing Finished Normally ");
IF labflg THEN
    BEGIN
    close(outstid.stfile);
    astruc($labf);
    *labf* ← EOL, " labels file is ", *labf*;
    typeas($labf);
    END;
IF memflg THEN
    BEGIN
    close(prtstid.stfile);
    astruc($memf);
    *memf* ← EOL, " mem-list file is ", *memf*;
    typeas($memf);
    END;
    close(idstid.stfile);
RETURN END.

(iexpmdk)PROC(delstr,idfnum); % get next individ ident for current grp%
LOCAL count;
LOCAL TEXT POINTER ptr, z1, z2, s0, s1, s2;
REF delstr;
*delstr* ← *delstr*, ';';
makeptr(asrref(&delstr), $ptr);
intiás(0);

% the "aflg" flag allows the automatic-mode groups --- TIPG USERG
SERVERG and ASSO CG --- to be processed despite the fact that they
are marked "Comments: SPECIAL" in the identfile. Without this
maneuver, these group idents would be bypassed by GETMDKIDS since
SPECIAL means "deleted" %

    IF automatic THEN aflg ← TRUE ELSE aflg ← FALSE;

LOOP
    BEGIN
    % get next individual ident within current group/orgzn %
    *entry* ← NULL;

```

```

% get full identfile info for this ident %
IF NOT getmdkids($ptr, $entry, 0, idfnum) THEN EXIT LOOP;
IF dflg THEN REPEAT LOOP; % that ident was deleted %
getiid($entry, 0, $z1, $z2);
*auxlit* ← " ", z1 z2;
typeas($auxlit);
% delete all control characters except EOL's %
FIND SF(*entry*) ↑s0;
LOOP
  BEGIN
    IF FIND s0 > ENDCHR THEN EXIT LOOP;
    IF FIND s0 > [NLD] ↑s2 ↑s1 ←s1 THEN
      BEGIN
        count ← s1[1];
        IF ((*entry*[count] IN [OB, 37B]) AND (*entry*[count] #
          EOL)) THEN ST s1 s2 ← NULL;
        s0[1] ← s2[1];
      END
    ELSE EXIT LOOP;
  END;
% format statements for CDC 6600 mailing labels program %
IF labflg THEN process(IF automatic THEN $nothing ELSE
  $grpident);
% format statement for printed mem-list %
IF memflg THEN list($entry, $auxlit);
END;
% sort membership list, but not now if in automatic mode %
IF (memflg AND NOT automatic) THEN stlist();
RETURN;
END.

```

```

(getmdkids) PROCEDURE (ptr, astr, infotype, idfnum); % expand grp %
LOCAL expchr, gpstid;
LOCAL TEXT POINTER idf, ide, tmpptr, srcptr, dstptr;
LOCAL STRING idstr[20], infostr[500];
REF ptr, astr;
%reads an ident from pointer ptr into astring idstr, and then calls
ckident to get info on it.
IF infotype = 0, then it returns all of the info in astr, if
infotype = 1, then the name only is returned.
Uses infostr as a work area%
expchr ← 0;
%first, read ident%
LOOP
  BEGIN
    CCPOS ptr;
    IF FIND $NP ↑idf ' ; THEN
      BEGIN
        IF NOT popids(&ptr) THEN RETURN(FALSE); %no more idents%
      END
    ELSE EXIT LOOP;
  END;
IF NOT FIND idf [ NP / ' ; / '( ]
  ↑ptr ←ptr ↑ide ←ide THEN
  err($"Ident List Format Error");
FIND ptr ($NP '( [']) ↑ptr);

```

```

IF FIND idf ('&/!') ↑idf < CH > THEN expchr ← READC;
*idstr* ← idf ide; %ident%
%Now get info, and check ident%
IF ckident($idstr, $infostr, idfnum : gpstid) THEN %return
something%
  BEGIN
  % set flag to show if ident was SPECIAL (i.e., deleted)
  ... but don't test TIPG SERVERG USERG or ASSOCG %
  IF (deleted($infostr) AND NOT aflg) THEN
  BEGIN
  dflg ← TRUE; % true if ident is deleted %
  GOTO ea;
  END
  ELSE dflg ← FALSE;
  aflg ← FALSE; % only true first time thru getmdkids %
IF orgrptst($infostr, 0) THEN
  BEGIN
  expchr ← TRUE;
  IF expchr THEN
  BEGIN
  getmem($infostr, 0, $dstptr, 0);
  IF FIND dstptr -EOL %membership list present% THEN
  BEGIN
  pushids(&ptr);
  dstptr ← gpstid;
  FIND dstptr ↑ptr;
  RETURN(getmdkids(&ptr, &astr, infotype, idfnum));
  END;
  END;
  END;
  END;
  END;
%Now edit and append to astr %
(ea): IF infotype = 1 THEN getifnf($infostr, $infostr);
*astr* ← *astr*, *infostr*;
RETURN(TRUE) END.

```

```

(stlist) PROCEDURE; % sort group membership list %
LOCAL cflg, i, k, m, pgct, pgmax, xvs;
LOCAL iptstid, indstid, itlstid, kmax; %for brief ident list%
LOCAL TEXT POINTER sl, sx, tl, tx;
LOCAL STRING skipout[25], ttitle[500], tpstrg[1000], tlstrg[1000];

% sort membership list %
tpstid ← hdstid;
tlstid ← prtstid;
sortgrp(hdstid, tlstid);

% set up stid's for brief ident listing when in automatic mode %
IF automatic THEN
  BEGIN
  iptstid ← gettail(tlstid);
  iptstid ← cis(iptstid, $blankcard, succdir);
  indstid ← iptstid;
  kmax ← 40;
  END;

```

```
% initialize to insert page breaks for output quickprint usage %  
new ← TRUE;  
pgmax ← 6;  
xvs ← 0; % needed only to delete statement in new nls %  
  
% count output entries PER AGE, and insert page breaks %  
% also, eliminate duplicate entries if in AUTOMATIC mode %  
% produce the brief ident listing in automatic mode, %  
% and, identify the group coord'tr when NOT in automatic mode %  
  
LOOP % mainloop %  
  BEGIN  
    % set up % <14>  
    *skipout* ← ', " " ;
```

```
{ IF NOT automatic THEN  
  BEGIN  
    tpstid ← cis (tpstid, # skipout, suedir);  
    tpstid ← cis (tpstid, # title, suedir);  
  END;
```



```
{ .tpstid ← cis(tpstid, $skipout, succdir); % force new page %
. .tpstid ← cis(tpstid, $title, succdir); % title is group name %
pgct ← pgmax;
cflg ← FALSE;
```

```
% print explanation at start of group %
```

```
IF new THEN
```

```
BEGIN
```

```
*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;
tpstid ← cis(tpstid, $style, succdir);
tpstid ← cis(tpstid, $blankcard, succdir);
new ← FALSE;
pgct ← pgct - 1;
END;
```

```
% allow several entries ("pgmax" value) per page %
```

```
i ← pgct;
```

```
LOOP % inner loop %
```

```
% initialization %
```

```
BEGIN
```

```
tpstid ← getsuc(tpstid);
IF tpstid = ihdstid THEN EXIT LOOP; % exit inner loop %
FIND SF(tpstid) ↑sl;
IF NOT FIND sl > [EOL] ↑sx THEN FIND SE(tpstid) ↑sx;
```

```
% when in automatic mode, eliminate duplicate entries and ...
output a brief list (idents and names) of each individ %
```

```
IF automatic THEN
```

```
BEGIN
```

```
% set up string for comparison with subsequent entry %
```

```
*tpstrg* ← sl sx;
```

```
% if next entry is equal to present one, delete it %
```

```
LOOP % innermost loop %
```

```
BEGIN
```

```
tlstid ← getsuc(tpstid);
IF tlstid = ihdstid THEN EXIT LOOP;
FIND SF(tlstid) ↑tl;
IF NOT FIND tl > [EOL] ↑tx THEN FIND SE(tlstid) ↑tx;
*tlstrg* ← tl tx;
IF *tpstrg* # *tlstrg* THEN EXIT LOOP;
cdelsta(tlstid, 0, $xvs);
lnct ← lnct - 1;
END;
```

```
% output ident and name using already-formatted data %
```

```
FIND sl > 37$37CH ↑tx; % end of ident field %
FIND tx < $NP $PT ↑tl >; % start of ident field %
*tlstrg* ← " ", tl tx, " ";
*tlstrg* ← *tlstrg*/(1 TO 8), sl tl;
iprtstid ← cis(iprtstid, $tlstrg, succdir);
```

```
END;
```

```

% when NOT in automatic mode, identify group coordinator %
IF NOT automatic THEN
  BEGIN
  IF FIND sx < [2SP] ↑tx $PT ↑tl > THEN *tpstrg* ← tl tx
  ELSE *tpstrg* ← NULL;
  FIND sl > CH ↑sx;
  IF *coor* = *tpstrg* THEN
    BEGIN
    ST sl sx ← '*';
    cflg ← TRUE;
    END;
  END;

% keep count for insertion of page breaks %
i ← i - 1;
IF i > 0 THEN REPEAT LOOP
ELSE
  % if a grp coord was found on this page, footnote it %
  BEGIN
  IF (cflg AND NOT automatic) THEN
    BEGIN
    *skipout* ← "* Group Coordinator";
    tpstid ← cis(tpstid, $skipout, succdir);
    cflg ← FALSE;
    END;
  EXIT LOOP; % exit inner loop %
  END;
END;

% play this till group exhausted %
lnct ← lnct - pgct;
IF lnct > 0 THEN REPEAT LOOP
ELSE EXIT LOOP; % exit main loop %
END; % of main loop %

% update print stid to tail of output, for start of next group %
IF NOT automatic THEN
  BEGIN
  prtstid, stpsid ← origin;
  prtstid ← getail(getsub(prtstid));
  END;

% footnote if a grp coord was found on last pg for this group %
IF (cflg AND NOT automatic) THEN
  BEGIN
  *skipout* ← "* Group Coordinator";
  prtstid ← cis(prtstid, $skipout, succdir);
  END;

% in automatic mode sort brief ident list & insert page breaks %
IF automatic THEN
  BEGIN
  % sort ident list %
  itlstid ← iptstid;
  iptstid ← indstid;

```

```
sortgrp(indstid, itlstid);  
% initialize page and line counts %  
k ← 0;  
m ← 0;  
*skipout* ← ', SP, SP, EOL;
```

```
... delete  
... keep  
... delete
```



```

        *title* ← "      ", "NIC IDENTs", EOL;
% LOOP TO INSERT PAGE AND LINE BREAKS IN BRIEF IDENT LISTING %
LOOP
  BEGIN
    % is new page needed %
    IF k = 0 THEN
      BEGIN
        k ← kmax;
        iptstid ← cis(iprtstid, $skipout, succdir);
        iptstid ← cis(iprtstid, $title, succdir);
      END;
    % is blank line needed (every five lines) %
    IF m = 0 THEN
      BEGIN
        m ← 5;
        iptstid ← cis(iprtstid, $"      ", succdir);
      END;
    % keep track of where we are %
    iptstid ← getsuc(iprtstid);
    IF iptstid.stpsid = origin THEN EXIT LOOP;
    k ← k-1;
    m ← m-1;
  END; % of LOOP for page/line-break insertion %
  prtstid.stpsid ← origin;
  prtstid ← gettail(getsub(prtstid));
END; % of processing brief ident listing %

```

```

RETURN;
END.

```

```

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

```

```

LOCAL i;
LOCAL TEXT POINTER current, tp1, tp2, t7, t8;
LOCAL STRING name(300);
REF groupid;

getinam($entry, $name, 0, 0); % put name in "name" %
% DO LINE 1 (really, "card image" #1): NAME and GROUPID %
IF FIND > (SF(*name*) ['']) ↑tp1 ↑tp2 +tp2) THEN
  *name* ← SF(*name*) tp2, SP, tp1 SE(*name*);
  *xlit* ← *blankcard*;
IF name.L > 24 THEN
  BEGIN
    *lit2* ← EOL, *name*, "      name too long, truncated to:";
    typeas($lit2);
    FIND SF(*name*) ↑t7;
    FIND SE(*name*) ↑t8;
  LOOP
    BEGIN
      IF name.L < 25 THEN EXIT LOOP
    ELSE
      BEGIN
        IF NOT FIND t8 < $PT 1$NP ↑t8 > THEN FIND t7 > 24$24CH
          ↑t8;

```

delete this
branch to
avoid title k
" < PL > " on
idents list

(X)


```

        name.L ← t8[1];
        REPEAT LOOP;
        END;
    END;
    crlf();
    typeas($name);
    crlf();
    END;
    *xlit*[1 TO 1+name.L] ← *name*;
    IF NOT automatic THEN IF groupid.L THEN *xlit*[30-groupid.L TO
    30] ← SP, *groupid*;
% DO LINES 2 THRU 8 %
    end ← FALSE;
    *lit2* ← NULL;
    laddress($entry, $lit2 %dest%, 0,0, idstid.stfile);
    *lit2* ← *xlit*[1 TO 30], EOL, *lit2*;
    astruc($lit2); % capitalize %
    FIND SF(*lit2*) ↑current;
    FOR i ← 1 UP 1 UNTIL > 4 DO
        card($current, lit2.L);
    IF NOT end THEN
        BEGIN
            crlf(-);
            *xlit* ← *name*, " address label truncated ", EOL;
            typeas($xlit);
            END;
    RETURN END.

```

(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. %

```

    LOCAL column, difference;
    LOCAL TEXT POINTER t1, t2, t7, t8;
    REF current;

```

```

*xlit* ← *blankcard*;
IF NOT end THEN % more input %

```

```

    BEGIN
        column ← 1;
        LOOP
            BEGIN
                IF NOT FIND current > ([EOL] ↑t1 ↑t2 ←t2) THEN
                    BEGIN
                        t2[1] ← leng + 1;
                        t1[1] ← leng;
                        end ← TRUE;
                    END;
                difference ← t2[1] - current[1];
            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 %
            BEGIN

```

```

                BEGIN

```

```

        IF difference < 31 THEN EXIT LOOP
      ELSE
        BEGIN
          IF NOT FIND t2 < $PT l$NP ↑t2 > ↑t1 THEN EXIT LOOP;
          difference ← t2[l] - current[l];
          REPEAT LOOP;
        END;
      END;

      *xlit* [column TO column + difference] ← current t2;
      current[l] ← t1[l];
      column ← column + 30;
      IF column > 31 OR current[l] >= leng OR end THEN EXIT LOOP; %
      done with a whole card %
    END;
  END;

%truncate off any extra info in last field%
*xlit*[61 TO 80] ← *blankcard*[61 TO 80];
xlit.L ← 60;
outstid ← cis(outstid, $xlit, succdir);
RETURN END.

(list) PROCEDURE (entr, idnt); % compose individ print line %
  REF entr, idnt;
  LOCAL i, j, diddle, phoneflag, addmor, orgflg;
  LOCAL TEXT POINTER sf, adf, adx, ade, tf, tx, te, tl, hf, he, nf,
  nx, ne;
  LOCAL STRING out[2000], addr[400], netaddr[200], name[200],
  phonestr[100], affil[100];

  % Initialization. %
  FIND ↑sf;

  % Get raw info for formatting. %
  % Full name %
  getinam($entr, $name, 0, 0);
  % Capitalize individual's last name %
  FIND SF(*name*) ↑nf;
  FIND SE(*name*) ↑ne;
  IF FIND nf > [","] ↑nx THEN
    *name* ← +nf nx, nx ne
  ELSE astruc($name);

  % Network address %
  *netaddr* ← NULL;
  getinma($entr, $netaddr, 0, 0);
  FIND SF(*netaddr*) ↑he;
  WHILE (FIND he ["<↑V>"] ↑he ↑hf ←hf) DO ST hf he ← NULL;
  IF getihost($entr, 0, $hf, $he) THEN
    *netaddr* ← *netaddr*, "@", hf he;
  IF NOT netaddr.L THEN *netaddr* ← "Network Address Not Known";

  % Full address. %
  addr.L ← 0;
  IF laddress($entr, $addr, 0, 0, idstid.stfile) THEN
    BEGIN % delete all control characters except EOL's %

```

```

FIND SF(*addr*) ↑adf;
LOOP
  BEGIN
  IF FIND adf > ENDCHR THEN EXIT LOOP;
  IF FIND adf > [NLD] ↑ade ↑adx ←adx THEN
  BEGIN
    J ← adx[1];
    IF ((*addr*[j] IN [OB, 37B]) AND (*addr*[j] # EOL))
    THEN ST adx ade ← NULL;
    adf[1] ← ade[1];
  END
  ELSE EXIT LOOP;
  END;
END
ELSE *addr* ← " Address Not Known ";
FIND SF(*addr*) ↑adf SE(*addr*) ↑ade;

% Network Organization %
getiorg ($entr, $affil, 0, 0);
astruc($affil); %capitalize %
orgflg ← TRUE;

% Build output string. %

% Initialization %
diddle ← 0; % USED TO GET CORRECT SPACING %
*idnt* ← *idnt*, " ";

% Full name and ident in 36 spaces. %
FIND ne ↑nx;
WHILE nx[1] - nf[1] >= 36 - idnt.L DO FIND nx < [SP] ↑nx;
  % Will not work for names with elements longer than
  48-idnt.L characters not separated by spaces! %
*out* ← *out*, nf nx, SP;
FOR i ← out.L + 1 UP UNTIL > diddle + 36 DO *out*[i] ← SP;
*out*[out.L-idnt.L TO out.L] ← *idnt*;

% Rest of line is network address-- indent three characters on
next line. %
*out* ← " ", *out*, *netaddr*, EOL, " "; % room for * %

% Put out rest of name if necessary %
WHILE (FIND > BETWEEN nx ne (SP [PT])) DO
  BEGIN
  FIND nx > SP ↑nf ne ↑nx;
  WHILE nx[1] - nf[1] > 28 DO FIND nx < [SP] ↑nx;
  *out* ← *out*, " ", nf nx, EOL, " ";
  END;

% Now next line: Beginning of address plus phone number %
FIND adf > ([EOL] ↑adx ←adx/ [ENDCHR] ↑adx);
WHILE adx[1] - adf[1] >= 32 DO FIND adx < [SP] ↑adx;
diddle ← out.L;
*out* ← *out*, adf adx;
FOR i ← out.L + 1 UP UNTIL > diddle + 34 DO *out*[i] ← SP;
addmcr ← TRUE;

```



```

% If phone put out first line of phone number %
  IF phoneflag ← getiphone($entr, 0, $tf, $te) THEN
    BEGIN
      *phonestr* ← tf te;
      FIND SF(*phonestr*) ↑tf SE(*phonestr*) ↑te ↑tx;
      WHILE tx[1] - tf[1] > 23 DO FIND tx < [SP] ↑tx;
      *out* ← *out*, tf tx;
      END;
    *out* ← *out*, EOL;

```

```

% The remaining lines %
  WHILE ((addmor ← (FIND adx > NP)) OR (phoneflag OR orgflg)) DO
    BEGIN
      diddle ← out.L;
      IF addmor THEN
        BEGIN
          IF FIND adx > EOL ↑adx THEN
            BEGIN
              *out* ← *out*, " "; %don't indent any more%
            END
          ELSE
            BEGIN
              *out* ← *out*, " ";
              FIND adx > NP ↑adx;
            END;
          IF (FIND adx ↑adf > ( [EOL] ↑adx ←adx / ade ↑adx ))
            THEN
              BEGIN
                WHILE adx[1] - adf[1] > 33-(out.L-diddle) DO FIND adx
                  < [SP] ↑adx;
                *out* ← *out*, adf adx;
                FOR i ← out.L + 1 UP UNTIL > diddle + 37 DO *out*[i]
                  ← SP;
                END;
                j ← 35;
              END
            ELSE j ← 37; % j controls spacing when no more addr data
              but more phone or org data %
          IF phoneflag THEN
            BEGIN
              IF FIND > BETWEEN tx te ( SP [PT] ) THEN
                BEGIN
                  FIND tx > SP ↑tf te ↑tx;
                  WHILE tx[1] - tf[1] > 17 DO FIND tx < [SP] ↑tx;
                  FOR i ← out.L + 1 UP UNTIL > diddle + 41 DO *out*[i]
                    ← SP;
                  *out* ← *out*, tf tx;
                END
              ELSE
                BEGIN
                  phoneflag ← FALSE;
                  orgflg ← FALSE;
                  FOR i ← out.L + 1 UP UNTIL > diddle + j DO *out*[i] ←
                    SP;
                  *out* ← *out*, *affil*;
                END;
            END;

```



```

        END;
    IF (orgflg AND NOT phoneflag) THEN
        BEGIN
            orgflg ← FALSE;
            FOR i ← out.L + 1 UP UNTIL > diddle + j DO *out*/i/ ←
                SP;
            *out* ← *out*, *affil*;
        END;
    *out* ← *out*, EOL;
END;

```

```

% output the text for current member of group being processed %
    *out* ← *out*, EOL;
    lnct ← lnct + 1;
    prtstid ← cis(prtstid, $out, succdir);

```

```

RETURN;
END.

```

```

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

```

```

(labfil) PROCEDURE (strg); % get output file for labels %
    LOCAL TEXT POINTER tpl, tp2;
    REF strg;
    labflg ← TRUE;
    LOOP
        BEGIN
            crlf();
            typeas($" Output file for Labels = ");
            *strg* ← NULL;
            txtlit($strg);
            outstid ← origin;
            IF NOT FIND SF(*strg*) > ['.'] THEN *strg* ← *strg*, ".NLS";
            % Open calls err if it doesn't find the file. %
            ON SIGNAL ELSE GO TO newfile;
            IF outstid.stfile ← open(0, $strg) THEN
                BEGIN
                    typeas($" (old file) CONFIRM ");
                    IF input() # CA THEN REPEAT LOOP;
                    % using old file - position to tail %
                    outstid ← gettail(getsub(outstid));
                END LOOP;
            END
        ELSE
            BEGIN
                (newfile):
                typeas($" (new file) CONFIRM ");
                IF input() # CA THEN REPEAT LOOP;
            END
        END
    END

```

```

ON SIGNAL ELSE NULL; %disarm all of them%
% force NLS extension %
  IF NOT FIND SF(*strg*) > '< l$(LD/'-)' > l$(LD/'-)' ttp1
  ([';] ttp2 ← tp2 / TRUE ttp2) THEN
    BEGIN
      typeas($" cant understand filename: ");
      typeas($strg);
      REPEAT LOOP;
    END;
  *strg* ← SF(*strg*) ttp1, ".NLS", tp2 SE(*strg*);
  IF NOT outstid.stfile ← opwk(O, $strg) THEN
    typeas($" Bad File Name ") % (that's all it could be) %
  ELSE EXIT LOOP;
  END;
END;
RETURN;
END.

```

```

(memfil) PROCEDURE (strg); % get output file for mem-lists%
  LOCAL TEXT POINTER ttp1, ttp2;
  REF strg;
  memflg ← TRUE;
  LOOP
    BEGIN
      crlf();
      typeas($" Output file for Mem-Lists = ");
      *strg* ← NULL;
      txtlit($strg);
      prtstid ← origin;
      IF NOT FIND SF(*strg*) > ['.] THEN *strg* ← *strg*, ".NLS";
      % Open calls err if it doesn't find the file. %
      ON SIGNAL ELSE GO TO newfil;
      IF prtstid.stfile ← open(O, $strg) THEN
        BEGIN
          typeas($" (old file) CONFIRM ");
          IF input() # CA THEN REPEAT LOOP;
          % using old file - position to tail %
          prtstid ← gettail(getsub(prtstid));
          EXIT LOOP;
        END
      ELSE
        BEGIN
          (newfil):
          typeas($" (new file) CONFIRM ");
          IF input() # CA THEN REPEAT LOOP;
          ON SIGNAL ELSE NULL; %disarm all of them%
          % force NLS extension %
          IF NOT FIND SF(*strg*) > '< l$(LD/'-)' > l$(LD/'-)' ttp1
          ([';] ttp2 ← tp2 / TRUE ttp2) THEN
            BEGIN
              typeas($" cant understand filename: ");
              typeas($strg);
              REPEAT LOOP;
            END;
          *strg* ← SF(*strg*) ttp1, ".NLS", tp2 SE(*strg*);
          IF NOT prtstid.stfile ← opwk(O, $strg) THEN

```

```

        typeas($" Bad File Name ") % (that's all it could be) %
    ELSE EXIT LOOP;
    END;
END;
RETURN;
END.

```

%conversion to new nls: The following six procedures were copied from the old nls system into this file in order to resolve all references. These procedures are not global under the new nls.%

```

(identlist) PROC(astr, fileno); %read a list of idents into astr%
    %Read and verify a list of idents into astr...terminated by CA%
    LOCAL idfile;
    REF astr;
    %open file, if necessary%
    IF fileno = 0 THEN
        BEGIN
            idfile ← 0;
            ON SIGNAL ELSE sigclose(idfile := 0);
            idfile ← open(0, jfname($"identfile"));
        END
    ELSE idfile ← fileno;
    *astr* ← NULL;
    DO
        BEGIN
            rdident(&astr, 0, idfile);
            IF curchr = '(' THEN %comment%
                BEGIN
                    *xlit* ← NULL;
                    echolt();
                    rdlit($xlit, $cmntdl); %read comment%
                    echoff();
                    *astr* ← *astr*, '(', *xlit*, ") ";
                    IF lookc() = CA THEN
                        BEGIN
                            echon();
                            input();
                            echoff();
                        END;
                END
            ELSE
                BEGIN
                    IF *astr*[astr.L] NOT= SP THEN *astr* ← *astr*, SP; %separate
                    idents by space%
                    IF curchr = 'Y OR curchr = 'N THEN
                        typeas($"Please continue, Ident = ");
                END;
            END
        UNTIL curchr = CA;
        WHILE *astr*[astr.L] = SP DO BUMP DOWN astr.L;
        IF fileno = 0 THEN close(idfile := 0);
        RETURN END.

```

```

(rdident) PROCEDURE (astr, infostr, fileno); %Read an IDENT from the
user%
    %This procedure reads and verifies an ident, and appends it to the

```

```
string astr.%
%If fileno is 0 then, the ident file will be opened and closed in
lower level routines. If infostr is non-zero, then the contents of
the entry will be returned in the string pointed to by infostr%
%-----%
```

```
LOCAL breakchar;
LOCAL STRING idstr[30];
REF astr;
LOOP
  BEGIN
    *idstr* ← NULL;
    CASE breakchar ← tiid($idstr) OF
      = '?': %help the user%
        BEGIN
          typeas($"
          Please type one of the following:
          ");
          typeas($"an Ident, CA to terminate the list, CD to abort,
          '.LastNameQuery, '.BeginLastNameQuery','.',
          '[NameStringQuery]'.
          ");
          REPEAT LOOP;
        END;
      ENDCASE
      IF idstr.L THEN
        IF NOT tiptid($idstr, infostr, fileno) THEN
          BEGIN
            crlf();
            IF astr.L > 0 THEN
              BEGIN
                prompt($"
                Idents accumulated so far: ");
                prompt(&astr);
                prompt($"
                Next Ident = ");
              END
            ELSE prompt($"
            Please continue, Ident = ");
            REPEAT LOOP;
          END
        ELSE
          BEGIN
            astruc($idstr);
            *astr* ← *astr*, *idstr*;
            RETURN(curchr); %tiptid may have read more text --
            hence a new break char%
          END
        ELSE RETURN(breakchar);
      END;
    END.

(tiptid) PROCEDURE(idstr, infostr, fileno); %prgroup Ident Queries%
  LOCAL expchr, type;
  LOCAL TEXT POINTER tptr1, tptr2;
  LOCAL STRING work[500];
  REF idstr;
```



```

IF idstr.L = empty THEN RETURN(TRUE);
CASE *idstr*[1] OF
  ='.: %lastname search%
    BEGIN
      IF curchr NOT= CA THEN
        BEGIN
          *idstr* ← *idstr*, curchr;
          txtlit(&idstr);
          END;
        *idstr* ← *idstr*[2 TO idstr.L];
        CCPOS SE(*idstr*); %check for incomplete name%
        IF READC = '. THEN
          BEGIN %partially specified last name, remove trailing
            periods%
            BUMP DOWN idstr.L;
            WHILE READC = '. DO BUMP DOWN idstr.L;
            type ← idchr;
            END
          ELSE type ← lname;
          stnamcap(&idstr);
          RETURN( idflsearch(type, &idstr, infostr, fileno) );
          END;
        ='/:
          BEGIN
            IF curchr NOT= CA THEN
              BEGIN
                *idstr* ← *idstr*, curchr;
                txtlit(&idstr);
                END;
              FIND SF(*idstr*) CH ↑tptr1 ( [''] ↑tptr2 ← tptr2 / SE(*idstr*)
                ↑tptr2);
              *idstr* ← tptr1 tptr2;
              RETURN( idflsearch(litstr, &idstr, infostr, fileno) );
              END;
            = '&, = '↑:
              BEGIN
                expchr ← *idstr*[1];
                *idstr* ← *idstr*[2 TO idstr.L];
                IF NOT infostr THEN infostr ← $work;
                END;
            =CD: GOTO STATE;
            = SP: %deblank front of string%
              BEGIN
                *idstr* ← *idstr*[2 TO idstr.L];
                REPEAT CASE;
                END;
            ENDCASE expchr ← FALSE;
          IF NOT infostr THEN
            BEGIN
              IF NOT oldid(&idstr, fileno) THEN %No such ident%
                BEGIN
                  tiderr();
                  RETURN(FALSE);
                END;
            END
          END
        ELSE

```

```
BEGIN
IF NOT ckident(&idstr, infostr, fileno) THEN %No such ident%
  BEGIN
  tiderr();
  RETURN(FALSE);
  END;
IF expchr AND NOT orgrptst($work,0) THEN
  BEGIN
  typeas("$" Not a group or organization
  ");
  RETURN(FALSE);
  END;
  END;
IF expchr THEN *idstr* ← expchr, *idstr*;
RETURN(TRUE);
END.
```

```
(tiid) PROCEDURE(string); %collect an Ident literal from the user%
LOCAL breakchar;
IF inprompt THEN prompt($"I: ");
echoff();
WHILE lookc() = SP DO todco( input() );
echolt();
rdlit(string, $idntdel : breakchar);
echoff();
RETURN(breakchar);
END.
```

```
(tiderr) PROCEDURE; %error routine%
todco(' ');
todco('?');
crlf();
RETURN;
END.
```

FINISH