ARC 26425

Help Services Sample Session

SRI-ARC

9 SEP 75

Augmentation Research Center

STANFORD RESEARCH INSTITUTE
MENLO PARK, CALIFORNIA 94025

HELP SERVICES SAMPLE SESSION

INTRODUCTION

The online computer system you will be using provides direct response to what you have just typed into your terminal. It also provides help services, various types of feedback which you can request for immediate, online information about all aspects of the system. Three help services are:

1. Lists of command alternatives you may use at any point (reached by typing a questionmark).

2. Explanations of these alternatives (reached by typing a <CTRL-Q>).

3. Descriptive information, such as definitions, explanations, and instructions (reached by giving the Help command).

This sample session demonstrates these help services. The process is explained for a typewriter terminal. It is essential that you be at a terminal, typing in commands and text as the scenario describes them.

Throughout this sample session we spell out the sequence of keys you will strike to make something happen and explain to you what the results should be. Keys that do not print, such as carriage return and escape (also called altmode), are named inside angle brackets, e.g. <CR>, <ESC>. <SP> represents a space. The control key <CTRL> is used like the shift key. You hold it down while you type the letter that is after the hyphen. The notation for control key is <CTRL-(some character)>.

Some control keys to remember...

```
<CTRL-X> aborts commands before you have typed a <CR>.
<CTRL-O> stops printing.
<CTRL-A> deletes the character you have just typed.
<CTRL-W> deletes the word you have just typed.
<CTRL-Q> gives an explanation of your current location
         in command space.
```

When you see <CR>, use the return or carriage return on your keyboard.

INSTRUCTION

1. You have logged in and are ready to use the online system. Let's first use the Help command to see how it describes itself.

To execute the Help command you type "h" for Help, and then the term you want to have explained. In this case, you want to learn more about the command "Help."

```
............................................................
You type:
            hhelp<CR>
You see:
            BASE C: Help  OK/T:  T: help
............................................................
```

This command will produce an explanation of Help at your terminal, followed by a list of items called a "menu" which you can use to obtain further information. Below the menu you will see the following line:

```
                    (HELP)</T:  T:
```

As explained in your Help typeout, this line prompts you to ask Help for more information.

2. Let's use the menu to learn more about help services. Menu item 4 looks like it will supply us with this information.

```
............................................................
You type:
            4<CR>
You see:
            Help </T: T: 4
............................................................
```

This time an explanation in the form of instructions appears, followed by more menu items. You are informed about two other help services: <CTRL-O> and questionmark. This sample session will also demonstrate these services.

3. Now let's try a practical exercise with the Help command to aid us in working online. After having read the "Preface to NLS Tools" you know that you need a workspace called a file. You can use Help to discover how to set up your own files.

```
    ............................................................
    You type:
              files<CR>
    You see:
              (Help) </T: T: files
    ............................................................
```

This is followed by an explanation about files, called "File STRUCTURE." Notice that the title of the explanation is not exactly what you typed in. "File STRUCTURE" is the most general explanation of files in Help. For more specific details refer to the menu items, or type in a term discussed in the explanation.

4. Since you want to know how to go about creating a file, and none of the menu items directly refers to this kind of information, you need to look for a new term to type in. The description of File STRUCTURE mentions a command that creates a file. You might choose to type in this command name.

```
    ............................................................
    You type:
               create<SP>file<CR>
    You see:
               (Help) </T: T: create file
    ............................................................
```

This produces the syntax of the command, followed by an explanation of how the command works, references, and a menu. If you type a 1<CR>, you can see an example of how the command looks when implemented at your terminal. Note: You could have left off the word "file" in this case, but sometimes more than one word is necessary.

5. You now have enough information to create a file. To leave Help, type a <CTRL-X>.

```
..............................................................
You type:
                <CTRL-X>
You see:
                (Help) </T:

                BASE C:
..............................................................
```

The control-X key <CTRL-X> takes you back to Base, as indicated by the prompt "BASE C:".

6. You create a file by use of the information you have learned, and name the file FIRST-AID.

```
..............................................................
You type:
                <SP>crffirst-aid<CR>
You see:
                BASE C: Create  C: File T: first-aid

                <DIRECTORYNAME, FIRST-AID.NLS;1,>
..............................................................
```

7. Now you have a workspace, but what can you do next? You can see a list of your current possible alternatives if you type a questionmark.

```
..............................................................
You type:
                ?
You see:
                BASE C:
                Current Alternatives are:
..............................................................
```

Below the heading appears a long list of commandwords (the "verb" type) that you may use at this time.

8. Scanning the commandwords, Insert looks like the most promising choice. Notice that a broken line has ended the list. At this point you can type the first letter or a space and the first letter of one of the commands and it will become part of your command. If you type <CR>, you return to where you were before you typed a questionmark.

```
............................................................
You type:
          i
You see:
          ---Insert C:
............................................................
```

You now have another prompt (C:) telling you that another commandword is necessary. Again, you can type a questionmark to see the commands available to you.

```
............................................................
You type:
          ?
You see:
          ---Insert C:
          Current Alternatives are:
............................................................
```

This heading is followed by a list of commandwords, this time the "noun" type.

9. You choose the word Statement since you know that a statement is the basic component of the online system files.

```
............................................................
You type:
          s
You see:
          ---Statement (to Follow) A:
............................................................
```

You are again prompted to continue your command, this time with an A:, which you recognize is the prompt for ADDRESS. You type in a 0, since you want your statement to follow the header (name of the file) which is always statement 0.

```
............................................................
You type:
          0<CR>
You see:
          ---Statement (to follow) A: 0
          L:
............................................................
```

10. After you type in your address, another prompt appears (L:) which does not look familiar to you. Having had so much luck with using the ?, you try it once again.

```
..............................................................
You type:
          ?
You see:
          L:
          Please type a LEVEL-ADJUST String:
..............................................................
```

11. Alternatives follow the line requesting a LEVEL-ADJUST, but this time you probably do not know enough about the alternatives for them to be helpful to you. There is a help service that provides a way to get detailed information about the command you are using at the moment. You can obtain this information by typing a <CTRL-Q>.

```
..............................................................
You type:
          <CTRL-Q>
You see:
          BASE
          ()
          LEVEL-ADJUST:
..............................................................
```

This is followed by a Help description which both explains the term and tells you what to do when presented with the prompt L:. You can now complete your command.

12. When you use <CTRL-Q> you are taken into a Help description and placed in the Help command. To return to Base from the Help command, you must now type the same <CTRL-X> that you used in step 5.

```
..............................................................
You type:
          <CTRL-X>
You see:
          (Help) </T:
          BASE C:
..............................................................
```

13. Typing <CTRL-Q> aborts any command you were typing in when you hit <CTRL-Q>. Therefore you are now back at BASE C:, ready to insert a statement.

```
..............................................................
You type:
           is0<CR>You can get Help by typing an h at
           the herald or at BASE C:, followed by a
           term.<CR>
You see:
           BASE C: Insert C: Statement (to follow) A: 0
           L:
           T: You can get Help by typing an h at the
           herald or at BASE C:, followed by a term.
..............................................................
```

14. After considering your first statement, you decide to add more to it. You know that Insert is the correct first term in your command, but you do not want to begin a new statement. To check other alternatives:

```
..............................................................
You type:
           i?
You see:
           BASE C: Insert C:
           Current Alternatives are:
..............................................................
```

15. Scanning the alternatives, Text looks like the most likely choice, but you are not sure what its effect will be. Two ways to obtain this information are available. The first, illustrated here, is to type a "t" for Text, then hit the <CTRL-Q>. An explanation of the command will follow.

```
..............................................................
You type:
           t<CTRL-Q>
You see:
           ---Text (to follow) A:
           BASE
           ()
           TEXT: Insert Text (to follow) DESTINATION CONTENT
           OK:
..............................................................
```

The second way to obtain this information is to type a <CTRL-X>, then use the Help command, typing in the words "insert text". Try this if you want more practice.

16. You return to Base and implement your command. You add text in the form of a sentence to statement 1 that will tell more about using help services. Since you want your new text to come at the end of the statement, your ADDRESS will be the statement number and "+e". (See the "Editing Sample Session I" for more information on adding text.)

```
............................................................
You type:
           <CTRL-X>it1+e<SP>Using CTRL-Q also takes you to
           Help, whereas typing a ? shows you a list of
           Current Alternatives.
You see:
           BASE C: Insert C: Text (to follow) A: 1+e
           T: Using CTRL-Q also takes you to Help, whereas
           typing a ? shows you a list of Current
           Alternatives.
............................................................
```

To see how the completed statement now looks, type a \.

```
............................................................
You type:
           \
You see:
           BASE C: \
           1 You can get Help by typing an h at the herald
           or at Base C:, followed by a term. Using
           CTRL-Q also takes you to Help, whereas typing
           a ? shows you a list of Current Alternatives.
............................................................
```

17. You have completed the sample session and are familiar with help services. You can now delete your First-Aid file.

```
..........................................................
You type:
           dffirst-aid<CR><CR>
You see:
           BASE C: Delete  C: File T: first-aid
           OK:

           Deleted Files Are:
              <DIRECTORYNAME,FIRST-AIDE,NLS;1,> and its
              partial copy
..........................................................
```

SAMPLE SESSION SUMMARY

To be placed in the Help command:*

Type h for Help followed by the term to be defined.

Type <CTRL-Q> at any point to receive an explanation of what you are currently doing.

To return from Help to your current subsystem (e.g. Base):

Type <CTRL-X>.

To see a list of Current Alternatives:

Type ? after the prompt of the subsystem you are in (e.g. at BASE C:), or at any point during a command.

To find out the effects of any command:

Type h (for Help), followed by the commandwords, or type a <CTRL-Q> at some point in the command.

*NOTE: A printed version of the responses in the Help command is available, entitled "The NLS-8 Glossary." The Glossary differs somewhat from what you see at your terminal because it has been reformatted for printing. It may also differ because the online file has been updated since the last printing of the Glossary.

HELP SERVICES SAMPLE SESSION

```
(J26425)  9-SEP-75 20:18;;;;   Title:  Author(s): Beverly Boli/BEV;
Distribution: /SRI-ARC( [ INFO-ONLY ] ) DIRT( [ INFO-ONLY ] ) ;
Sub-Collections:  SRI-ARC DIRT; Clerk: BEV;          Origin: < BOLI,
HELP/SCENARIO.NLS;18, >, 9-SEP-75 12:47 BEV ;;;;
 ####;
```

26425 Distribution

Douglas C. Engelbart, Martin E. Hardy, J. D. Hopper, Charles H. Irby, Harvey G. Lehtman, James C. Norton, Jeffrey C. Peters, Dirk H. Van Nouhuys, Kenneth E. (Ken) Victor, Richard W. Watson, Don I. Andrews, Jonathan B. Postel, Priscilla A. Wold, Rita Hysmith, Pamela K. Allen, Delorse M. Brooks, Elizabeth F. Finney, Beverly Boli, Lawrence A. Crain, Kirk Sattley, Susan Gail Roetter, Robert N. Lieberman, Ann Weinberg, Kenneth E. (Ken) Victor, Douglas C. Engelbart, James H. Bair, Elizabeth K. Michael, Richard W. Watson, Elizabeth J. Feinler, Harvey G. Lehtman, Kirk E. Kelley, Laura E. Gould, Jeanne M. Beck, Dirk H. Van Nouhuys, James C. Norton,

David C. Smith, Mary Ann Kellan, Buddie J. Pine, Andy Poggio, David L. Retz, Laura J. Metzger, Karolyn J. Martin, Jan A. Cornish, Larry L. Garlick, Priscilla A. Wold, Pamela K. Allen, Delorse M. Brooks, Beverly Boli, Rita Hysmith, Log Augmentation, Joseph L. Ehardt, Raymond R. Panko, Susan Gail Roetter, Robert Louis Belleville, Rene C. Ochoa, Ann Weinberg, Adrian C. McGinnis, Robert S. Ratner, David S. Maynard, Robert N. Lieberman, Sandy L. Johnson, James H. Bair, Jeanne M. Leavitt, Rodney A. Bondurant, Jeanne M. Beck, Marcia L. Keeney, Elizabeth K. Michael, Jonathan B. Postel, Elizabeth J. Feinler, Kirk E. Kelley, N. Dean Meyer, James E. (Jim) White

ARC 26426

Sendmail Sample Session I

SRI-ARC

19 SEP 75

Augmentation Research Center

STANFORD RESEARCH INSTITUTE
MENLO PARK, CALIFORNIA 94025

SENDMAIL SAMPLE SESSION I

A piece of the Secretary Functions Guide that didn't get journalized with the rest of it on Sept. 9.

SENDMAIL SAMPLE SESSION I

INTRODUCTION

The online computer system you will be using responds immediately to what you type at your terminal. This "Sendmail Sample Session" shows you how to use a subsystem called Sendmail to send messages and documents to people known to the system, and have these messages cataloged and stored in the Journal. The process is explained for a typewriter terminal. You will find it useful to be at a terminal, typing in the commands and text as the sample session describes them.

Although this document shows you how to perform a specific Sendmail task, notes at each step provide general explanations about what you are learning. Using this scenario as a model, you will be able to perform any of the commands described here in your daily work, referring to Help and other documentation for more information about the system.

Throughout this sample session we spell out the sequence of keys you strike to make something happen and separately show what will appear on your terminal in response. Keys that do not print, such as carriage return and escape (also called "altmode"), are named inside angle brackets, e.g. <CR>, <ESC>. <SP> represents a space. The control key <CTRL> is used like the shift key. You hold it down while you type the letter that is after the hyphen. The notation for control keys is <CTRL-(some character)>.

Some control keys to remember...

<CTRL-X>	aborts commands before you have typed a <CR>.
<CTRL-O>	stops printing.
<CTRL-A>	deletes the character you have just typed.
<CTRL-W>	deletes the word you have just typed.

When you see <CR>, use the return or carriage return on your keyboard.

If you get stuck or confused, typing "?" will show you the next possible alternatives. You then type in one of the alternatives and continue your command.

Typing <CTRL-Q> will provide you with information and explanations about NLS. Type a <CTRL-X> to return to where you were before you typed <CTRL-Q>. For more about getting information via <CTRL-Q> see the "Preface to NLS Tools" and "Help Services Sample Session."

INSTRUCTION

1. A short Sendmail session appears here to enable you to send a message to people recognized by the system. Although Sendmail has a very extensive system for sending, distributing, cataloging, indexing, and storing documents, most of these steps are done automatically (and invisibly) for you.

You first go to the Sendmail subsystem.

```
............................................................
You type:
            gs<CR>
You see:
            BASE C: Goto (subsystem) C: Sendmail
            OK:
            SEND C:
............................................................
```

Notice that the herald has changed from BASE to SEND, indicating a new subsystem.

2. You will use the Sendmail command Interrogate to send your message. This command automatically begins six different commands for you. The responses you enter have the same effect as executing six commonly used Sendmail commands.

Six questions, each appearing one at a time, will be asked. They are:

```
     distribute for action to:
     distribute for information-only to:
     title:
     type of source:
     show status?
     send the mail now?
```

You may type in the responses given in this sample session, or use this opportunity to send a Sendmail item of your own.

The Interrogate command will always prompt in the same way. If you do not want to respond to one of the prompts, just type a

<CR> (e.g., if you did not want to distribute your message to anyone for information-only, you would type a <CR> when prompted).

You have several options when prompted by "type of source:". You will choose Message in this step, a command which allows you to type in one statement (up to 2000 characters). You may edit the message with <CTRL-A> and <CTRL-W> while typing it in. Other "types of sources" are:

Branch
File
Group
Offline
Plex
Statement

For more information about these alternatives, use the Help command.

The "show status" prompt displays to you what the system knows about the current journal item. You are shown the questions you have answered and the responses you have given.

```
..............................................................
You type:
           i<CR>
You see:
           Send C: Interrogate OK:
           (distribute for action to:) T:


You type:
           pka<CR>
You see:
           (distribute for action to:) T: pka
           (distribute for information-only to:) T:

You type:
           paw2<CR>
You see:
           (distribute for information-only to:) T: paw2
           (title:) T:

You type:
           Using the Interrogate Command<CR>
You see:
           (title:) T: Using the Interrogate Command
           (type of source:) C:


You type:
           mThis message is being sent to you so I
```

can practice using the Sendmail Interrogate command.<CR>

```
    You see:
                 (type of source:) C: Message T: This message
                 is being sent to you so I can practice using
                 the Sendmail Interrogate command.
                 (show status?) Y/N:


    You type:
                 <CR>
    You see:
                 TITLE: Using the Interrogate Command
                 AUTHOR(S):  IDENT
                 DISTRIBUTE FOR ACTION TO: pka
                 DISTRIBUTE FOR INFO-ONLY TO: paw2

                 MESSAGE: This message is being sent to you so
                 I can practice using the Sendmail Interrogate
                 command.


    You type:
                 <CR><CR>
    You see:
                 (send the mail now?) Y/N:
                 Completed
    ...........................................................
```

As you might have recognized, each of the "prompts" in the Interrogate command has as its counterpart an individual Sendmail command. You will get more practice using the individual Sendmail commands in "Sendmail Sample Session II".

3. You have completed sending your message and would like to return to BASE. To do so, use the Quit command. (You may log out after quitting by typing <SP>l<CR>.)

```
    ...........................................................
    You type:
                 q<CR>
    You see:
                 SEND C: Quit
                 OK:
                 BASE C:
    ...........................................................
```

SAMPLE SESSION SUMMARY

Goto Sendmail:

Takes you to the Sendmail subsystem. You can leave the subsystem by typing Quit.

Interrogate:

Asks you six questions to aid you in sending a piece of mail. The responses you enter have the same effect as executing six commonly used Sendmail commands. The questions are:

distribute for action to:
distribute for information-only to:
title:
type of source:
show status?
send the mail now?

SENDMAIL SAMPLE SESSION I

(J26426) 18-SEP-75 13:51;;;; Title: Author(s): Beverly Boli/BEV;
Distribution: /SRI-ARC([INFO-ONLY]) DIRT([INFO-ONLY]) ;
Sub-Collections: SRI-ARC DIRT; Clerk: BEV; Origin: < BOLI,
SEND1A.NLS;5, >, 9-SEP-75 12:50 BEV ;;;;
####;

26426 Distribution

James C. Norton,

N. Dean Meyer, James E. (Jim) White, Douglas C. Engelbart, Martin E. Hardy, J. D. Hopper, Charles H. Irby, Harvey G. Lehtman, James C. Norton, Jeffrey C. Peters, Dirk H. Van Nouhuys, Kenneth E. (Ken) Victor, Richard W. Watson, Don I. Andrews, Jonathan B. Postel, Priscilla A. Wold, Rita Hysmith, Pamela K. Allen, Delorse M. Brooks, Elizabeth F. Finney, Beverly Boli, Lawrence A. Crain, Kirk Sattley, Susan Gail Roetter, Robert N. Lieberman, Ann Weinberg, Kenneth E. (Ken) Victor, Douglas C. Engelbart, James H. Bair, Elizabeth K. Michael, Richard W. Watson, Elizabeth J. Feinler, Harvey G. Lehtman, Kirk E. Kelley, Laura E. Gould, Jeanne M. Beck, Dirk H. Van Nouhuys Susan K. Ocken, Raphael Rom, David C. Smith, Mary Ann Kellan, Buddie J. Pine, Andy Poggio, David L. Retz, Laura J. Metzger, Karolyn J. Martin, Jan A. Cornish, Larry L. Garlick, Priscilla A. Wold, Pamela K. Allen, Delorse M. Brooks, Beverly Boli, Rita Hysmith, Log Augmentation, Joseph L. Ehardt, Raymond R. Panko, Susan Gail Roetter, Robert Louis Belleville, Rene C. Ochoa, Ann Weinberg, Adrian C. McGinnis, Robert S. Ratner, David S. Maynard, Robert N. Lieberman, Sandy L. Johnson, James H. Bair, Jeanne M. Leavitt, Rodney A. Bondurant, Jeanne M. Beck, Marcia L. Keeney, Elizabeth K. Michael, Jonathan B. Postel, Elizabeth J. Feinler, Kirk E. Kelley

ARC 26427

Sendmail Sample Session II

SRI-ARC

9 SEP 75

Augmentation Research Center

STANFORD RESEARCH INSTITUTE
MENLO PARK, CALIFORNIA 94025

SENDMAIL SAMPLE SESSION II

INTRODUCTION

This sample session gives you more practice using the Sendmail subsystem to send online items to other people the system knows. In the "Sendmail Sample Session I" you used the command Interrogate, which automatically prompts you with several questions to aid you in using Sendmail. Now you will learn to use several new commands without the aid of Interrogate prompts.

SENDMAIL SAMPLE SESSION II

INSTRUCTION

1. This sample session first shows you how to send a file through the Sendmail system using individual commands rather than the Interrogate command. This step will use the file created in the "Editing Sample Session I (MEMO)" to distribute through the Sendmail system. However, you may send any file you choose if this is not available. The instructions below will send whatever file you have loaded when you go to the Sendmail subsystem.

First load your file. After that you can follow the scenario to go to Sendmail. Then you give your Sendmail item the title Elephant Meeting and indicate to whom you want it distributed and for what reasons. You also use the Comment command to add a note about your Sendmail item.

```
..............................................................
You type:
             gs<CR>
You see:
             BASE C: Goto (subsystem) C: Sendmail
             OK:

You type:
             f<CR>
You see:
             SEND C: File A:

You type:
             tElephant Meeting<CR>
You see:
             SEND C: Title T: Elephant Meeting



You type:
             cIMPORTANT!<CR>
You see:
             SEND C: Comment T: IMPORTANT!

You type:
             dabev<CR>
You see:
             SEND C: Distribute (for) C: Action (to) T: bev
```

```
You type:
               didvn<CR>
You see:
               SEND C: Distribute(for) C: Information (only)(to)
               T: dvn


You type:
               s<CR>
You see:
               SEND C: Send (the mail) OK:
               Completed
************************************************************
```

To name a recipient you type in his/her IDENT, which is a string of characters that identifies that person to sendmail. This distribution list may contain more than one IDENT, each separated by spaces or commas. A copy of your sendmail item is automatically delivered to the author (you).

2. You may occasionally need more information about someone to whom you wish to send a piece of mail. For example, you may know the last name of your recipient, but not his/her IDENT. The command Show Record will supply the IDENT and other pertinent information.

```
************************************************************
You type:
               <SP>shr.boli<CR><CR><CR>
You see:
               SEND C:  Show C: Record (for ident) T: .boli

               The following individuals with last name
               Boli are already defined
               Boli, Beverly, Organization:  SRI-ARC,
               Ident = BEV

               Is this the correct Boli?
               Ident BEV accepted


               Ident: BEV
               Name: Boli, Beverly
               Organization: SRI-ARC
```

```
                    Groups: ...
                    Mail Addresses: ...
                    Phone: ...
                    Delivery: ...
          ..............................................................
```

If you had known your recipient's IDENT you could have typed that in rather than the last name. Then the same information would have appeared.

You may search for the IDENT of an individual any time you would type in an IDENT in a Sendmail command (for example, in the Distribute for Action command). You follow the same procedure as shown above: type a period, followed by the last name and a <CR>. In a command other than Show Record the system will understand that this <CR> is only to mark the end of the last name and will not carry out the command.

3. This step will illustrate the commands Message, Show Status (both introduced in "Sendmail Sample Session I" as part of the Interrogate command), and Authors. When you send an item through the Sendmail subsystem it automatically shows your IDENT as the author of the item. This may not always be appropriate. You may wish to register more than one author or send an item for someone else. The Authors command allows you to do this.

In the scenario below you will see "IDENT(S),pka" when you get to the Authors command. At this point type in your own IDENT, the IDENT of one or more other persons, or both. Separate IDENTS in a list with a comma.

```
............................................................
You type:
           dabev<CR>
You see:
           SEND C: Distribute (for) C: Action (to) T: bev

You type:
           dipooh<CR>
You see:
           SEND C: Distribute (for) Information (Only) (to)
           T: pooh

You type:
           aIDENT(S),pka<CR>
You see:
           SEND C: Authors T: IDENT(S),pka
```

SENDMAIL SAMPLE SESSION II

```
You type:
                 mGood morning.  I'm practicing using some
                 Sendmail commands.  Hope you have a
                 pleasant day.
You see:
                 SEND C: Message T: Good morning.  I'm practicing
                 using some Sendmail commands.  Hope you
                 have a pleasant day.


You type:
                 <SP>Shs<CR>
You see:
                 SEND C:  Show C: Status OK:
                 AUTHOR(S): IDENT(S),pka
                 DISTRIBUTE FOR ACTION TO: bev
                 DISTRIBUTE FOR INFOR-ONLY TO: pooh

                 MESSAGE: Good morning.  I'm practicing using
                 some Sendmail commands.  Hope you have a
                 pleasant day.


You type:
                 s<CR>
You see:
                 SEND C: Send (the mail) OK:
                 Completed
                 SEND C:
```

****

You may now return to the Base subsystem by typing Quit <CR>.

SENDMAIL SAMPLE SESSION II

SAMPLE SESSION SUMMARY

Goto Sendmail:

Takes you to the Sendmail subsystem. You can leave the subsystem by typing Quit.

File:

Sends a file. If you type a <CR> after File, the file you have loaded is sent.

Title:

Lets you give your item a title.

Distribute for Action:

Asks you to specify the recipient(s) whom you wish to receive the item, and from whom you want some action.

Distribute for Information:

Asks you to specify recipients whom you wish to receive the item for information purposes.

Comment:

Allows you to add a comment on the item you are sending.

Show Record:

Takes an IDENT or last name (preceded by a period) and displays current information from the Identfile about how to contact that person. You may also use the command if you know only the first part of the last name. Use the Help command to learn more about "last-name search".

Show Status:

Displays to you what information you have entered for the journal item you are working on. You are shown the questions you have answered and the responses you have given.

Message:

Allows you to type one statement (up to 2000 characters). Use <CTRL-A> and <CTRL-W> to edit, and <CR> to terminate, the message.

Authors:

Allows you to name the author(s) of a Sendmail item. If you don't use this command, the system automatically names you as the author.

SENDMAIL SAMPLE SESSION II

(J26427) 9-SEP-75 20:23;;;; Title: Author(s): Beverly Boli/BEV;
Distribution: /SRI-ARC([INFO-ONLY]) DIRT([INFO-ONLY]) ;
Sub-Collections: SRI-ARC DIRT; Clerk: BEV; Origin: < BOLI,
SEND2A.NLS;5, >, 9-SEP-75 12:50 BEV ;;;;
####;

26427 Distribution

Douglas C. Engelbart, Martin E. Hardy, J. D. Hopper, Charles H. Irby, Harvey G. Lehtman, James C. Norton, Jeffrey C. Peters, Dirk H. Van Nouhuys, Kenneth E. (Ken) Victor, Richard W. Watson, Don I. Andrews, Jonathan B. Postel, Priscilla A. Wold, Rita Hysmith, Pamela K. Allen, Delorse M. Brooks, Elizabeth F. Finney, Beverly Boli, Lawrence A. Crain, Kirk Sattley, Susan Gail Roetter, Robert N. Lieberman, Ann Weinberg, Kenneth E. (Ken) Victor, Douglas C. Engelbart, James H. Bair, Elizabeth K. Michael, Richard W. Watson, Elizabeth J. Feinler, Harvey G. Lehtman, Kirk E. Kelley, Laura E. Gould, Jeanne M. Beck, Dirk H. Van Nouhuys, James C. Norton,

David C. Smith, Mary Ann Kellan, Buddie J. Pine, Andy Poggio, David L. Retz, Laura J. Metzger, Karolyn J. Martin, Jan A. Cornish, Larry L. Garlick, Priscilla A. Wold, Pamela K. Allen, Delorse M. Brooks, Beverly Boli, Rita Hysmith, Log Augmentation, Joseph L. Ehardt, Raymond R. Panko, Susan Gail Roetter, Robert Louis Belleville, Rene C. Ochoa, Ann Weinberg, Adrian C. McGinnis, Robert S. Ratner, David S. Maynard, Robert N. Lieberman, Sandy L. Johnson, James H. Bair, Jeanne M. Leavitt, Rodney A. Bondurant, Jeanne M. Beck, Marcia L. Keeney, Elizabeth K. Michael, Jonathan B. Postel, Elizabeth J. Feinler, Kirk E. Kelley, N. Dean Meyer, James E. (Jim) White

BEV KIRK 8-SEP-75 18:07 26431

This is our latest pas at the shopping list for the next proposal, with man-month estimates included. Dee--Would you please put this in the Dirt notebook. Thanks. Bev

DOCUMENTATION SHOPPING LIST

1

Online Documentation Data Base Maintenance
Augmenting the Help Command
Production of Hardcopy from Online Documentation
New Documentation
Documentation Maintenance 1a

Approximate minimum total man-months: 34

1b

1. Online Documentation Data Base Maintenance

The Help multi-file data base is an encyclopedic description of NLS tools and NSW procedures designed for quick online how-to reference by the user. To be effective it must be comprehensive and current. Methods to keep the Help data bases up-to-date and keep their extensive cross-referencing accurate are needed. 1c

Index generation and maintenance: 5 mo
Current experience shows that the automatic creation and maintenance of an alpabetic index in a multifile data base is essential if the online Help is to grow and evolve as planned for NSW tools. This index would be generated from all named statements and perhaps all meaningful words. This needs the automatic link maintenance facility described below. This feature is also necessary for #3, below, "Hardcopy Production from Online Documentation." 1c1

-Once automatic index maintenance is implemented the index information can be "compiled" into a file that is much more efficiently searched than structured NLS files. It will also allow the implementation of standard Boolean arguments such as "x AND y OR z AND NOT a" and eventually, natural language typed in (and spoken?) English queries. 1c1a

Back links: 6 mo
It is essential that automatic link maintenance via back links be implemented in order to reduce the tremendous overhead and inherent mistakes in the current procedure for discovering bad links and updating them manually. This should be implemented as a property of a node. (See Comment feature, below.) Back links have the added benefit of automatic "forward references", a list of back links to what has been written about a document since it was published. 1c2

Comment feature: 0.75 mo

The comment feature would allow any arbitrary amount of designated text to dissappear when viewspec capital-T is in effect. The text reappears when viewspec capital-S is turned on. This feature is important for four separate functions. 1c3

-Making user-invisible comments to other Help writers. This would do away with the current percent sign convention which requires a special sequence generator or content analyzer pattern since it is not a part of the standard Nls capabilities. 1c3a

-Making output processor directives invisible in Help. This would do away with the necessity of having two separate directories of files, one with Output Processor (OP) directives and one without. And it would do away with having to delete directives and update to the second directory whenever a modification is made. 1c3b

-Placing links invisibly next to referenced text. This is needed to place link syntax in a node to define how that node will be viewed. (See #2, below.) It would also automatically take the user to enriching references. 1c3c

-The comment feature would be one way of implementing the backlink property, where the links to a node are invisibly backlinked from that node. 1c3d

2. Augmenting the Help Command

Several features have emerged as essential in the evolution of the Help command. More effective user control of the information available and a closer integration with other capabilities are two areas that need attention now. 1d

Point with mouse to words and lines: 0.5 mo
One of the first things a Display help user asks is why s/he can't point with the mouse. With this feature the user would be able to simply point to words and lines of text, and Help would display the descriptions of the words pointed to, or the descriptions referenced by links contained in the lines. 1d1

Outline and verbose description views: 1.5 mo
Users need to be able to dynamically see a brief, outline view of any branch of information, or see the full multi-paragraph description. The display user must also have an easy way to view the next screen full via a "next" function of some kind. Currently it is impossible to learn about a topic by simply reading serially without backing up, re-reading, or computing complicated back-up algorithms to see

the next paragraph. The lack of these capabilities has forced us to write choppy, fragmented paragraphs of information that are many times cryptic and not easily converted into readable pages on paper. This would solve these problems. 1d2

Do feature:
The "do" feature is the capability of having Help execute a command for you or a task consisting of a scenario of commands. Should user specification be necessary in the process, an advanced version of the do feature will tell in English sentences what is expected at every step of the way. This is an active tutorial/example/service which would make Help much more valuable as a teaching aid and provide a new service as a task doer. 1d3

Addition to the Help command: 0.5 mo
It would be implemented by writing command statements which appear as menu items under Help descriptions. A special symbol placed after the right anglebracket of a link will process the commands. Just the implemenation of this part would be useful by itself. 1d3a

Writing the command branches: 0.5 mo/tool 1d3b

Addition to the "Process" command: 1.5 mo
To get the full power of the question and answer capabilities, the "Process" command must be beefed up. This would be generally useful to NLS users. 1d3c

Descriptions for two levels of users: 1.5 mo
A difficulty with the present Help is that descriptions are written for a single, mythical, "average user". Some descriptions are too advanced for the very new user and others are too simple for the more knowledgeable user. This task would be to rewrite portions of the data base to provide two descriptions for major concepts and commands--one for novices and the other for experts. Some minor additions to Useroptions software could be made so that the user could specify whether she wanted to see the simple or more advanced descriptions. The default would be for the novice user. This task would require about 2 hours per description, beginning with an estimated minimum of 100 descriptions. One week of programming time is also needed. 1d4

Working with other Help systems: 1 mo??
Before we do too much in the way of thinking about more sophisticated search and tutorial functions, we should get in touch with our friends working on Scholar at BBN and COTCO at ISI and see if some coordination is in order. 1d5

3. Production of Hardcopy from Online Documentation

Although online documentation is flexible and immeditely accessible for user reference, hardcopy is preferred by some users, and is helpful for some applications. This endeavor would entail a variety of software and writing tasks to make online documentation such as the Help data bases automatically translatable into readable hardcopy. 1e

Features already cited in this document:
Many of the features cited above would aid in making online documentation into user understandable hardcopy guides, especially the "Comment feature" listed under #1, and "Outline and verbose descriptions views" listed under #2. 1e1

Text that would make hardcopy more readable: 3 to 5 mo
With the comment feature, documentors would be able to include textual passages, such as transitions, headings, and references, in the online document that would only be turned on for hard copy. This would help transform the essentially "reference" online documentation into a form which could be read from front to back. Once the comment feature is implemented, the first pass at inserting this "hardcopy only" text would take two weeks each for the six major tool description files (including core and NSW Exec) and one week each for the 9 other tools in the core AKW. That is a total of 21 weeks. 1e2

4. New Documentation 1f

Documentation of proposed new tools and features: 1 mo/tool; 2 day/feature 1f1

Error messages and user feedback response:
Users of Help should be able to query the system about any system information and error messages they get such as "fst entry nonexistant", "bad file", "illegal user number in file header", "File not online; If Archived, Use EXEC's INTERROGATE" and "Illegal BCCNT in fndchr". There are several things that can be done in this area including fixing the bugs. 1f2

Explanations: 3 mo
Explanations of the meanings of each of these messages and what to do when you get one need to be written for each tool. This task includes maintenance as well as writing descriptions and classifying error messages. 1f2a

Help button for errors: 1 mo
The system should remember the last message sent to the

user and take the user to the description for that message if the Help button is hit after receiving it. 1f2b

Sharpening the tools: 2.4 to 12 mo/contract year**
The bugs which cause some of these messages should be fixed. Cryptic words and misspellings can be changed to something more meaningful to the user. At a minimum with a collection of tools as large as the core AKW, 20 percent of one user-sensitive programmer should be allocated for sharpening the existing tools. This involves doing the many little things which get overlooked and are not deemed important by experienced users of the tools. These include minor bugs, design flaws or "unfeatures" which are particularly obstructive to new users. A user-sensitive programmer would be directly responsive to feedback from users. Since no time has been allocated to this in the past, there is a huge backlog of these impediments to the use and learning of the system. Twenty percent of one person over a 12 month contract would be approximately 2.5 months, although one whole person could easily be absorbed into it. **Perhaps this item should be located elsewhere in the proposal shopping list. 1f2c

5. Documentation Maintenance

Existing documentation must be updated to accurately describe changes in an experimental system. Some of the major documents that require much effort to maintain are: 1g

Help data bases: 3 mo
Base
Core
Sendmail
Calculator
Useroptions
Message
Programs
Graphics
Publications
Letter
Helpd 1g1

User manuals: 1 mo
Secretarial Functions Guide
DEX User's Manual 1g2

Systems documentation: 1 mo
Tool Suppliers documentation
FE System Interface documentation 1g3

(J26431) 8-SEP-75 18:07;;;; Title: Author(s): Beverly Boli, Kirk E. Kelley/BEV KIRK; Distribution: /JBP([ACTION]) DIRT([INFO-ONLY]) ; Sub-Collections: SRI-ARC DIRT; Clerk: BEV;

26431 Distribution

Jonathan B, Postel, Jonathan B, Postel, Priscilla A, Wold, Rita Hysmith, Pamela K, Allen, Delorse M, Brooks, Elizabeth F, Finney, Beverly Boli, Lawrence A, Crain, Kirk Sattley, Susan Gail Roetter, Robert N, Lieberman, Ann Weinberg, Kenneth E, (Ken) Victor, Douglas C, Engelbart, James H, Bair, Elizabeth K, Michael, Richard W, Watson, Elizabeth J, Feinler, Harvey G, Lehtman, Kirk E, Kelley, Laura E, Gould, Jeanne M, Beck, Dirk H, Van Nouhuys, James C, Norton, Help at BBNB: response to JHB's Comment in <33419,>

Jim, first I am including your Comment of <33419,>: 1

Comments: Help at O-1 has been updated to reflect the current Letter program. We need someone to update Help 8.0 at BBNB -- volunteer? Currently there is no mechanism for upating other Helps. Other changes (minor) I have made to O-1 Help can be filtered out using signatures and copied to BBNB. All we need is someone to do it. 1a

My response: 2

As I understand it, the procedure for getting the Help files at O-1 & BBNB both up-to-date has historically been to make the changes in one of the files and then to FTP that file to the other site to replace the one there. At least, it seems this would be a very fast way to solve the problem of time & manpower to get the presently accumulating changes at O-1 over to BBNB. 2a

I would be interested in doing the FTP. Also, I feel I'd be able to work on Help information for future changes if that fits into JHB's manpower schedule ....jeanne 2b

Help at BBNB: response to JHB's Comment in <33419,>

(J26432) 8-SEP-75 18:54;;;; Title: Author(s): Jeanne M. Beck/JMB; Distribution: /JHB([ACTION]) FEEDBACK([INFO-ONLY]) DVN([INFO-ONLY]) BEV([INFO-ONLY]) KIRK([INFO-ONLY]) JCN([INFO-ONLY]) SGR([INFO-ONLY]) LJM([INFO-ONLY]) ; Sub-Collections: SRI-ARC FEEDBACK; Clerk: JMB;

26432 Distribution

James H. Bair, Special Jhb Feedback, Dirk H. Van Nouhuys, Beverly Boli, Kirk E. Kelley, James C. Norton, Susan Gail Roetter, Laura J. Metzger,

What 33391 Is About

I describes how to build a commands branch that will go to other sites and collect the messages all into one file so you can read them at your home site without logging in to other sites. We thought people in the ARPA office who have directories and get messages at more than one site might be interested. ...I see you have real journal problems and I sure Dave will= 1

What 33391 Is About

(J26433) 9-SEP-75 11:51;;;; Title: Author(s): Dirk H. Van Nouhuys/DVN; Distribution: /RH([INFO-ONLY]) JMB([INFO-ONLY]) ; Sub-Collections: SRI-ARC; Clerk: DVN;

26433 Distribution
Rita Hysmith, Jeanne M. Beck,

Keeping Up Help Files (inspired by JHB item <33419>)

Jim--Your Journal item prompted me to remind you of the tentative procedures we mapped out for handling Help upkeep. Maybe this would be a good time to try to finalize those procedures. I'm sending you the branch from my file 'coop' that I believe reflects accurately the ideas we came up with a while back. Since Help at BBNB serves Utility clients, and is the running system, I believe it is covered in these provisions. Bev

Keeping Up Help Files (inspired by JHB item <33419>)

Applications and Development reached tentative agreement on ff. procedures (pending approval by Jim Norton): 1

1. One person is "responsible" for Help from each group (i.e. Dev. and App.) 1a

2. Dev. person fixes xHelp. App. person fixes running Help. 1b

3. The two persons try to keep in close communication so that there is constant "dialoge" which will enable them to help one another. 1c

Keeping Up Help Files (inspired by JHB item <33419>)

(J26434) 9-SEP-75 12:02;;;; Title: Author(s): Beverly Boli/BEV; Distribution: /JHB([ACTION]) DVN([INFO-ONLY]) JCN([INFO-ONLY]) SGR([INFO-ONLY]) KIRK([INFO-ONLY]) FEEDBACK([INFO-ONLY]) ; Sub-Collections: SRI-ARC FEEDBACK; Clerk: BEV;

26434 Distribution
James H, Bair, Dirk H, Van Nouhuys, James C, Norton, Susan Gail Roetter, Kirk E, Kelley, Special Jhb Feedback,

BEV 9-SEP-75 14:12 26435

Reserve Journal numbers for sample sessions

Jim--Here are the numbers I've assigned to the sample sessions:

Editing 2	26422
Editing 3	26423
File-V.	26424
Help	26425
Sendm. 1	26426
Sendm. 2	26427
Format	25925
Intro. to Doc. Prod.	26429

Editing 1 already had a number--It's on the document you have. The same also for the Preface. --Bev 1

Reserve Journal numbers for sample sessions

(J26435) 9-SEP-75 14:12;;;; Title: Author(s): Beverly Boli/BEV; Distribution: /JHB([INFO-ONLY]) ; Sub-Collections: SRI-ARC; Clerk: BEV;

26435 Distribution
James H. Bair,

xl10 format control for STRING construct

There is a new xl10runtime package (and a corresponding new xl10 compiler) at ISIC that implements the following format control (as an optional third parameter) for the STRING construct. let me know of any problems. 1

(frmtctrl) RECORD %format control for "STRING" construct % 2

fmncols[7], % # columns (with punctuation) / 0: as many as needed % 2a

fmjstfy[1], % TRUE: right justify; FALSE: left justify % 2b

fmfill[1], % fill field with TRUE: 0s; FALSE: spaces % 2c

fmlgcl[1], % TRUE: treat as 36 bit unsigned quantity % 2d

fmsgne[1], % TRUE: ignore LH of value and extend sign of RH % 2e

fmpsgn[1], % TRUE: print + if val > 0 / val is 36 bit quantity % 2f

fmdecm[1], % TRUE: print terminating decimal point % 2g

fmovrf[3], % column overflow control 2h

0 - print nothing 2h1

1 - print as many most significant digits (MSDs) as fit 2h2

2 - print as many least significant digits (LSDs) as fit 2h3

3 - print blanks 2h4

4 - print astericks in entire field 2h5

5 - print as many MSDs as fit followed by one asterick 2h6

6 - print as many LSDs as fit preceded by one asterick % 2h7

fmfloat[20]; % reserved for eventual floating point format control % 2i

2j

x110 format control for STRING construct

(J26436) 9-SEP-75 11:05;;;; Title: Author(s): Kenneth E. (Ken) Victor/KEV; Distribution: /NPG([ACTION]) DCE([INFO-ONLY]) RWW([INFO-ONLY]) JCN([INFO-ONLY]) ; Sub-Collections: SRI-ARC NPG; Clerk: KEV;

26436 Distribution

David C. Smith, Andy Poggio, David L. Retz, Jan A. Cornish, Larry L. Garlick, Robert Louis Belleville, Elizabeth J. Feinler, Joseph L. Ehardt, Jonathan B. Postel, Kirk E. Kelley, Karolyn J. Martin, David S. Maynard, Kenneth E. (Ken) Victor, James E. (Jim) White, Elizabeth K. Michael, Don I. Andrews, J. D. Hopper, Charles H. Irby, Harvey G. Lehtman, Douglas C. Engelbart, Richard W. Watson, James C. Norton, Further on item (26410,) Journal BUG

It seems that in one case I did get a journal item as a Message WITH the ACTION/INFO field. This case was an item from DMB, which she sent from BBNB and was delivered to me at BBNB. So all the others that I mentioned yesterday probably were sent across computers. ...jeanne 1

Further on item (26410,) Journal BUG

(J26437) 9-SEP-75 17:16;;;; Title: Author(s): Jeanne M. Beck/JMB; Distribution: /FEEDBACK([ACTION]) ; Sub-Collections: SRI-ARC FEEDBACK; Clerk: JMB;

26437 Distribution
    Special Jhb Feedback,

JMB 9-SEP-75 17:36 26438

Problem sending a message to feedback via Message subsystem at BBNB

I was trying to send a message to feedback with the Send Message command in the Message subsystem in NLS at BBNB. For the TO field I input "feedback" and then went on with the subject and message fields. When it was time to verify the distribution list, NLS interpreted that as a group ident with one member: "FEED (FEEDBACK@)". when it tried to send the message it failed, saying it found no such directory. I notice now that it messed up in interpreting the directory for FEED as "FEEDBACK@", it should have been "FEEDBACK@OFFICE-1" so that delivery wouln't look for the dir FEEDBACK at BBNB. 1

I looked at the listing for the ident FEED at BBNB and it has the Online (NLS) field listed as user: feedback and host:office-1, but there is no network delivery listing. Does the message subsystem use the network delivery listing to figure out where the sndmsg goes? This is the only clue to the problem I could find. 2

Problem sending a message to feedback via Message subsystem at BBNB

(J26438) 9-SEP-75 17:36;;;; Title: Author(s): Jeanne M. Beck/JMB; Distribution: /FEEDBACK([ACTION]) ; Sub-Collections: SRI-ARC FEEDBACK; Clerk: JMB;

26438 Distribution
Special Jhb Feedback,

OUTLINE FOR FINAL REPORT

Following RWW's suggestions I have made up an outline for the final report covering the contract from July 74-July 75. Much of the material for the report already exists elsewhere. I'd appreciate your help in trying to gather it together. Will you please send me locations of the material, or send the file itself through the journal (this would help, since I have to work at BBNB)? Also, if I can help anyone write the new stuff, please let me know. Dick would like to see this out by the end of the month, so I guess first drafts and references to materials should be in my hands by the 19th. Anyway, it will give us all something to do during these idle times... Thanks, Bev

OUTLINE FOR FINAL REPORT

OUTLINE FOR FINAL REPORT COVERING PERIOD JULY,1974-JULY, 1975 1

Front End (CHI) 1a

1. Outline of implementation changes 1a1

2. Brief overview of Front End (with reference to Appendix, which will contain papers already written describing FE, etc.) 1a2

3. Motivation for FE (lifted from last proposal) 1a3

4. Changes to CML since last contract period 1a4

5. Brief description of work on L-10 1a5

Protocols (JEW, JBP) 1b

1. Outline of completed work 1b1

2. Papers written by JEW on DPS 1b2

3. Description of packages (this would only be 2-3 pages, with reference to papers already written in Appendix) 1b3

NLS (EKM) 1c

1. A 3-4 page outline of what was done in last contract, itemizing features, etc. (Note--Could we use stuff in Quarterly Report, and just flesh it out?) References to Appendix for papers going into greater detail. 1c1

Debugger (KEV) 1d

1. Brief outline of debugger, with reference to Appendix for papers going into more detail 1d1

Appendix 1e

Front End -- containing papers already prepared by CHI on FE. 1e1

Protocols -- containing papers already prepared on Protocol Packages, PCP B8 documents (does this last make sense?) 1e2

NLS -- containing papers already prepared on individual features, etc. of NLS 1e3

Debugger -- Documents already prepared on debugger 1e4

OUTLINE FOR FINAL REPORT

(J26439) 9-SEP-75 18:58;;;; Title: Author(s): Beverly Boli/BEV; Distribution: /RWW([ACTION]) CHI([ACTION]) JEW([ACTION]) JBP([ACTION]) EKM([ACTION]) KEV([ACTION]) ARC-DEV([INFO-ONLY]) ; Sub-Collections: SRI-ARC ARC-DEV; Clerk: BEV; Origin: < BOLI, FINALREP.NLS;1, >, 9-SEP-75 18:46 BEV ;;;;####;

26439 Distribution

Richard W. Watson, Charles H. Irby, James E. (Jim) White, Jonathan B. Postel, Elizabeth K. Michael, Kenneth E. (Ken) Victor, David C. Smith, Mary Ann Kellan, Andy Poggio, David L. Retz, Jan A. Cornish, Larry L. Garlick, Delorse M. Brooks, Beverly Boli, James E. (Jim) White, Ann Weinberg, Kenneth E. (Ken) Victor, Dirk H. Van Nouhuys, Jonathan B. Postel, Elizabeth K. Michael, David S. Maynard, Karolyn J. Martin, Harvey G. Lehtman, Kirk E. Kelley, Charles H. Irby, Joseph L. Ehardt, Robert Louis Belleville, Don I. Andrews, Richard W. Watson, Douglas C. Engelbart,

BEV 9-SEP-75 19:20 26441

A little help from my friend

Dirk--Has the Introduction to Doc. through Nls been journalized in its final form? If so, do you know the number? If not, would you mind journalizing it with the number 26429 and attaching a comment that it will now be printed as part of the Secretarial Functions Guide, although it is also available individually? I have lost track of where it is kept, but suspect it's at isic. If not, and I just can't find it at BBNB, let me know the directory and I'll journalize it. Thanks. Bev 1

A little help from my friend

(J26441) 9-SEP-75 19:20;;;; Title: Author(s): Beverly Boli/BEV; Distribution: /DVN([ACTION]) ; Sub-Collections: SRI-ARC; Clerk: BEV;

26441 Distribution
   Dirk H, Van Nouhuys,

SECRETARIAL FUNCTIONS GUIDE

This is the introductionto the Secretarial Functions Guide, a document produced for the last NSW contract. Designed for new users performing secretarial tasks with NLS, the Guide is a collection of individual modules which can be used alone or together as a package. Since each of the documents is separate, they will be journalized indivudually. (A couple of them have already been journalized.) The Guide will be printed next week. This 'First Edition' contains: Editing Sample Sessions I, II, and III: File-Viewing Sample Session; Help Services S. S.; Sendmail S> S> I and II; Format S. S.; Tutorial...on Letter Program; Preface to NLS Tools; Introduction to Documentation through NLS.

BEV 9-SEP-75 20:04 26442

ARC 26421

SECRETARIAL FUNCTIONS GUIDE

SRI-ARC

9 SEP 75

Augmentation Research Center

STANFORD RESEARCH INSTITUTE
MENLO PARK, CALIFORNIA 94025

INTRODUCTION

The "Secretarial Functions Guide" is designed to help you use the oNLine System (NLS) to perform secretarial tasks. The interactive computer system that you will be using enables you to compose, transcribe, and edit text; set up files; and send, receive, and print documents. This Guide introduces you to the system and shows you how to put some of its features to use.

The Guide is a collection of instructional documents placed in a folder. This design allows for maximum flexibility, so that you can choose those materials best suited to your tasks. Documents in the Guide fall into the following categories:

1. Transcribing (or composing) and modifying text.

   Writing a memo.

   Creating first and final drafts of reports.

   Transcribing an outline.

2. Viewing and printing files at the terminal.

3. Creating tables.

4. Writing (and formatting) a letter.

5. Formatting a document for printing.

6. Sending messages or documents to other people.

7. Reading messages or documents received through the system.

8. Other pertinent introductory or descriptive documents.

Most of these documents are called "Sample Sessions". Each session includes a brief introduction, an instructional section that takes you step by step through the specific commands necessary to perform certain secretarial tasks, and a summary of all of the commands taught in that session. An exact presentation of what you type in to execute the commands and what you will see at your terminal, along with general explanations, should make these sessions self-instructional.

SECRETARIAL FUNCTIONS GUIDE

(J26442) 9-SEP-75 20:04;;;; Title: Author(s): Beverly Boli/BEV; Distribution: /DIRT([INFO-ONLY]) SRI-ARC([INFO-ONLY]) ; Sub-Collections: SRI-ARC DIRT; Clerk: BEV; Origin: < BOLI, GUIDE.NLS;4, >, 9-SEP-75 12:39 BEV ;;;; ####;

26442 Distribution

Rita Hysmith, Log Augmentation, Joseph L. Ehardt, Raymond R. Panko, Susan Gail Roetter, Robert Louis Belleville, Rene C. Ochoa, Ann Weinberg, Adrian C. McGinnis, Robert S. Ratner, David S. Maynard, Robert N. Lieberman, Sandy L. Johnson, James H. Bair, Jeanne M. Leavitt, Rodney A. Bondurant, Jeanne M. Beck, Marcia L. Keeney, Elizabeth K. Michael, Jonathan B. Postel, Elizabeth J. Feinler, Kirk E. Kelley, N. Dean Meyer, James E. (Jim) White, Douglas C. Engelbart, Martin E. Hardy, J. D. Hopper, Charles H. Irby, Harvey G. Lehtman, James C. Norton, Jeffrey C. Peters, Dirk H. Van Nouhuys, Kenneth E. (Ken) Victor, Richard W. Watson, Don I. Andrews,

Jonathan B. Postel, Priscilla A. Wold, Rita Hysmith, Pamela K. Allen, Delorse M. Brooks, Elizabeth F. Finney, Beverly Boli, Lawrence A. Crain, Kirk Sattley, Susan Gail Roetter, Robert N. Lieberman, Ann Weinberg, Kenneth E. (Ken) Victor, Douglas C. Engelbart, James H. Bair, Elizabeth K. Michael, Richard W. Watson, Elizabeth J. Feinler, Harvey G. Lehtman, Kirk E. Kelley, Laura E. Gould, Jeanne M. Beck, Dirk H. Van Nouhuys, James C. Norton, David C. Smith, Mary Ann Kellan, Buddie J. Pine, Andy Poggio, David L. Retz, Laura J. Metzger, Karolyn J. Martin, Jan A. Cornish, Larry L. Garlick, Priscilla A. Wold, Pamela K. Allen, Delorse M. Brooks, Beverly Boli

Secretarial Functions Guide--Ready to Go to Press

Larry--As you know, we have been working on the Secretarial Functions Guide as part of our last NSW contract. Parts of it have already been used at Gunter (some of the sample sessions), butwe are now ready to havethe whole document printed in one piece. Our plan was to have itbound in such a way that people could put different pieces of it together, lift separate sample sessions out of it, etc. We need to know how many copies you would like for us toprint. I'll call you tomorrow (Thurs.) when you've returned to Gunter. Thanks. Bev Boli 1

Secretarial Functions Guide--Ready to Go to Press

(J26443) 10-SEP-75 12:49;;;; Title: Author(s): Beverly Boli/BEV; Distribution: /LAC([ACTION]) ; Sub-Collections: SRI-ARC; Clerk: BEV;

26443 Distribution
Lawrence A, Crain,

ARC-File

Up until Monday, 9/8/75, J. Leavitt maintained an ARC file in her directory at Office-1. That file has been transfered to BBNB and is now in my directory.

This file contains the name, address, home phone numbers and other work related information of all ARC personnel. Any changes i.e., residence, status, or title change, etc., should now be brought to my attention. 1

ARC-File

(J26444) 10-SEP-75 15:18;;;; Title: Author(s): Delorse M. Brooks/DMB; Distribution: /SRI-ARC([INFO-ONLY]) ; Sub-Collections: SRI-ARC; Clerk: DMB;

26444 Distribution

Douglas C. Engelbart, Martin E. Hardy, J. D. Hopper, Charles H. Irby, Harvey G. Lehtman, James C. Norton, Jeffrey C. Peters, Dirk H. Van Nouhuys, Kenneth E. (Ken) Victor, Richard W. Watson, Don I. Andrews, David C. Smith, Mary Ann Kellan, Buddie J. Pine, Andy Poggio, David L. Retz, Laura J. Metzger, Karolyn J. Martin, Jan A. Cornish, Larry L. Garlick, Priscilla A. Wold, Pamela K. Allen, Delorse M. Brooks, Beverly Boli, Rita Hysmith, Log Augmentation, Joseph L. Ehardt, Raymond R. Panko, Susan Gail Roetter, Robert Louis Belleville, Rene C. Ochoa, Ann Weinberg, Adrian C. McGinnis, Robert S. Ratner, David S. Maynard, Robert N. Lieberman, Sandy L. Johnson, James H. Bair, Jeanne M. Leavitt, Rodney A. Bondurant, Jeanne M. Beck, Marcia L. Keeney, Elizabeth K. Michael, Jonathan B. Postel, Elizabeth J. Feinler, Kirk E. Kelley, N. Dean Meyer, James E. (Jim) White

DNLS suggestions

I am a new user, and it seems to me (personally) that a couple of things wwould be more "natural" if done a little differently; to wit (1) setting viewspecs: How about <char> turning ON a viewspec and -<char> turning it OFF? Might be more symmetrical. (2) window edges: It might be nice to be able to show window edges visibly when multiple windows are on the screen. I am frequently (usually?) confused about what text is in what window. 1

DNLS suggestions

(J26445) 10-SEP-75 16:21;;;; Title: Author(s): David C. Smith/DAV; Distribution: /FEEDBACK([INFO-ONLY]) ; Sub-Collections: SRI-ARC FEEDBACK; Clerk: DAV;

26445 Distribution
    Special Jhb Feedback,

Test

testing feed ident from bbnb 1

Test

(J26446) 10-SEP-75 19:59;;;; Title: Author(s): Elizabeth J. Feinler/JAKE; Distribution: /FEED([ACTION]) ; Sub-Collections: SRI-ARC; Clerk: JAKE;

26446 Distribution
Special Jhb Feedback,

Shared screens timeout

Dave, I thought the timeout period for shared screens was a minute (by trying tlink four times) instead of 15 seconds. Did someone change it or did tenex change or am i wrong? -- Charles. 1

Shared screens timeout

(J26447) 11-SEP-75 12:21;;;; Title: Author(s): Charles H. Irby/CHI; Distribution: /JDH([ACTION]) ; Sub-Collections: SRI-ARC; Clerk: CHI;

26447 Distribution
J. D. Hopper,

ARC and Intellegence Community

Dirk, It was very heartening to see your note regarding possible ARC involvment with NSA, CIA, DIA, etc. I do involve my personal values with my paid work. That is why I work at ARC. While there may in fact be no viable alternatives, my personal values, my instincts and intuition all indicate that working for the intellegence community is heading ARC in the wrong direction. When I first heard of all this about a year ago, I told Doug how distastful it was to me. We have discussed it several times since then, with Doug always insisting that there were good CIA and bad CIA departments and that they were very sharp people. I beleive everyone must be guided by his own values, but I have had a hard time assessing the feelings of most ARCers on this point. I am not saying that the group should be run as a democracy, but only that decissions made at a management level that do not have the support of the group will probably not work out too well. 1

(J26448) 11-SEP-75 12:39;;;; Title: Author(s): Charles H. Irby/CHI;
Distribution: /DVN([ACTION]) SRI-ARC([INFO-ONLY]) ;
Sub-Collections: SRI-ARC; Clerk: CHI;

26448 Distribution

James E. (Jim) White, Douglas C. Engelbart, Martin E. Hardy, J. D. Hopper, Charles H. Irby, Harvey G. Lehtman, James C. Norton, Jeffrey C. Peters, Dirk H. Van Nouhuys, Kenneth E. (Ken) Victor, Richard W. Watson, Don I. Andrews,
Dirk H. Van Nouhuys, David C. Smith, Mary Ann Kellan, Buddie J. Pine, Andy Poggio, David L. Retz, Laura J. Metzger, Karolyn J. Martin, Jan A. Cornish, Larry L. Garlick, Priscilla A. Wold, Pamela K. Allen, Delorse M. Brooks, Beverly Boli, Rita Hysmith, Log Augmentation, Joseph L. Ehardt, Raymond R. Panko, Susan Gail Roetter, Robert Louis Belleville, Rene C. Ochoa, Ann Weinberg, Adrian C. McGinnis, Robert S. Ratner, David S. Maynard, Robert N. Lieberman, Sandy L. Johnson, James H. Bair, Jeanne M. Leavitt, Rodney A. Bondurant, Jeanne M. Beck, Marcia L. Keeney, Elizabeth K. Michael, Jonathan B. Postel, Elizabeth J. Feinler, Kirk E. Kelley, N. Dean Meyer

USER SERVICES TRIP REPORT: Course at the Pentagon for users of the ARPA-NSW slots on August 26-27 1

1. Time: 26 August - 27 August by RH (1.5 Person-days) 1a

2. Person (users of not) contacted: 1b

Major Mortenson 1b1

Betty Finney 1b2

Carol Mahoney 1b3

Major Richard O. Wells (ROW) - Attended the Courses 1b4

Sgt. Michael Oddy - Attended the Courses 1b5

3. COURSE: 1c

"Basic TNLS-8 Course" was given in its entirety, and "The TNLS Course Outline Introduction to Structure and Viewing" was given in its entirety with the exception of the "Jump to Content, etc." commands. We went into "The Intermediate TNLS Course Outline" very lightly covering only: 1c1

Append
Break
Visible
Invisible 1c1a

The Documentation that was distributed is: 1c2

Basic TNLS-8 Course
The TNLS Course Outline Introduction to Structure and Viewing
The Intermediate TNLS Course Outline
Cue Cards
Command Summary
SRI Investments in Tomorrow
The Augmented Knowledge Workshop <14724>
Coordinated Information Services for a Discipline- Or Missions-Oriented Community <12445>
Format Library 1c2a

4. DISCUSSION: 1d

I realize it must appear that we covered a lot of material in a very short time, however it was necessary due to the time constraints of when they were accessible for training and when their report (Volume 9) is due to be completed. Both men were very quick and intuitive. Plus we were able to conduct the courses while sitting at a terminal; this minimized the confusion usually associated with a pure lecture course. 1d1

After we had finished the first day of training the two men stayed at the terminal practicing after I had left. They were actually in Volume 9 doing some of the necessary editing. This was great because they were able to discover which areas really were not clear to them, where they needed additional assistance, etc. for the second day. 1d2

User Services Report - Course at Pentagon

(J26449) 11-SEP-75 13:29;;;; Title: Author(s): Rita Hysmith/RH; Distribution: /US([INFO-ONLY]) JHM([INFO-ONLY]) JCN([INFO-ONLY]) DCE([INFO-ONLY]) RLL([INFO-ONLY]) POOH([INFO-ONLY]) EKM([INFO-ONLY]) ; Sub-Collections: SRI-ARC US; Clerk: RH;

26449 Distribution

Susan Gail Roetter, Priscilla A. Wold, Jeanne M. Beck, Pamela K. Allen, Rita Hysmith, Sandy L. Johnson, John H. McAfee, James C. Norton, Douglas C. Engelbart, Robert N. Lieberman, Ann Weinberg, Elizabeth K. Michael, Response to 'The mysteries of <CTRL-E>...' (33420,)

I tried deleting a character and ending the command with <CTRL-E> and another time with the INSRT key and both times I ended up in the Insert Mode. Also, in TNLS <CTRL-E> is taught as a way to end any command (in the Third Course) such that the command will be completed and the user will be prompted for the level of a new statement that will be inserted after the statement where the last edit was made. Seems like that's the way it works in DNLS too so I guess I don't see the inconsistency. 1

Response to 'The mysteries of <CTRL-E>...' (33420,)

(J26450) 11-SEP-75 13:37;;;; Title: Author(s): Susan Gail Roetter/SGR; Distribution: /FEED([INFO-ONLY]) ARC-APP([INFO-ONLY]) DVN([INFO-ONLY]) CHI([INFO-ONLY]) ; Sub-Collections: SRI-ARC ARC-APP; Clerk: SGR;

26450 Distribution

Special Jhb Feedback, Buddie J. Pine, Laura J. Metzger, Priscilla A. Wold, Pamela K. Allen, Rene C. Ochoa, Jeffrey C. Peters, Marcia L. Keeney, Jeanne M. Beck, Geoffrey S. Goodfellow, Rodney A. Bondurant, Douglas C. Engelbart, Jeanne M. Leavitt, Susan Gail Roetter, Raymond R. Panko, Adrian C. McGinnis, James C. Norton, J. D. Hopper, Elizabeth J. Feinler, James H. Bair, Robert N. Lieberman, N. Dean Meyer, Sandy L. Johnson, Martin E. Hardy, Dirk H. Van Nouhuys, Charles H. Irby,

Info Needed for Hotel Reservations at the KWAC Meeting

Before Jim Norton he asked that I see that the necessary hotel reservations get made for the KWAC meeting. In response to my question Bob Sheppard sent (33435,) which lists hotels and prices. He also mentions that it would be good to make reservations soon. After looking at this list and doing a little calculation, I'd suggest the Tredway Motor Hotel or the Sheraton Commander Hotel. As a group we would save approx. $420 by staying there as opposed to the cheapest Sonesta rate. I'd welcome comments and if there is agreement I'll try to get the reservations started. This message was distributed to he people JCN thought would be attending when he left. 1

Info Needed for Hotel Reservations at the KWAC Meeting

(J26451) 11-SEP-75 14:04;;;; Title: Author(s): Susan Gail Roetter/SGR; Distribution: /US([ACTION]) JHB([ACTION]) RLL([ACTION]) BJP([ACTION]) JCN([ACTION]) DCE([ACTION]) JDH([ACTION]) ; Sub-Collections: SRI-ARC US; Clerk: SGR;

26451 Distribution

Susan Gail Roetter, Priscilla A. Wold, Jeanne M. Beck, Pamela K. Allen, Rita Hysmith, Sandy L. Johnson, James H. Bair, Robert N. Lieberman, Buddie J. Pine, James C. Norton, Douglas C. Engelbart, J. D. Hopper,

JMB 11-SEP-75 19:07 26452

'Intelligence' re--26448,>

Charles, thanks for somebody, finally, putting something in the journal to SRI-ARC, 'Intelligence' re--26448,>

Dirk & Charles, Could one of you forward a copy of Dirk's item referred to in <26448,> to me? I am very interested in this dialogue and I know several others who are. ....jeanne b. 1

JMB 11-SEP-75 19:07 26452

'Intelligence' re--26448,>

(J26452) 11-SEP-75 19:07;;;; Title: Author(s): Jeanne M. Beck/JMB; Distribution: /DVN([ACTION]) CHI([ACTION]) ; Sub-Collections: SRI-ARC; Clerk: JMB;

26452 Distribution
   Dirk H. Van Nouhuys, Charles H. Irby,

Reference to My Item on ARC and the Intelli/egence Community

The item is (jjournal,33442,). 1

Reference to My Item on ARC and the Intelli/egence Community

(J26453) 11-SEP-75 22:51;;;; Title: Author(s): Dirk H. Van Nouhuys/DVN; Distribution: /JHB([INFO-ONLY]) JMB([INFO-ONLY]) CHI([INFO-ONLY] so you would know I answered them) ; Sub-Collections: SRI-ARC; Clerk: DVN;

26453 Distribution
   James H. Bair, Jeanne M. Beck, Charles H. Irby,

KIRK BEV 12-SEP-75 04:00 26454

Documentation Weekly Report for 9/6/75

Bev 1

This Week 1a

Continued working on Xhelp,Base. Got a little hung up in Output Commands. 1a1

Refined Documentation Shopping List for next Proposal with Kirk. It's now in Helpd, 1a2

Discussed with Jim B. including a sample session on Letter Program written by Laura in Sec. Func. Guide. He'll let me know as soon as possible. 1a3

Discussed new hyphenation feature with other Development people, 1a4

Next Week 1b

Continue on Xhelp, Base. 1b1

Get Sec. Func. Guide to printing. Assign Journal numbers and put them on Title pages first. (I wonder if this thing ever will get printed.) 1b2

Go over Documentation Shopping List for next proposal with Jon et. al. 1b3

Kirk 2

Done 2a

Put Documentation task schedule on line. 2a1

Added some changes to AFMFormat after reviewing latest proofs. This is looking pretty good from my end. Most of the bugs seem to be fixed except for the interface to George. 2a2

Completed documenation for the new Useroptions "Space (for tabs)" command. 2a3

Refined the Help needs list. 2a4

Do 2b

Add any more bells, and whistles to AFMFormat that seem necessary. 2b1

Teach Dave Smith about Help. 2b2

Fix do list for documentation in relation to current milestones. 2b3

Transfer Userprogram and AFMFormat responsibility to Jan. 2b4

Clean up Help files for 8.5. 2b5

Documentation Weekly Report for 9/6/75

(J26454) 12-SEP-75 04:00;;;; Title: Author(s): Kirk E. Kelley, Beverly Boli/KIRK BEV; Distribution: /DPCS([INFO-ONLY]) SRI-ARC([INFO-ONLY]) DIRT([INFO-ONLY]) DMB([INFO-ONLY] dirt) ; Sub-Collections: SRI-ARC DPCS DIRT; Clerk: KIRK;

26454 Distribution

Rita Hysmith, Pamela K. Allen, Delorse M. Brooks, Elizabeth F. Finney, Beverly Boli, Lawrence A. Crain, Kirk Sattley, Susan Gail Roetter, Robert N. Lieberman, Ann Weinberg, Kenneth E. (Ken) Victor, Douglas C. Engelbart, James H. Bair, Elizabeth K. Michael, Richard W. Watson, Elizabeth J. Feinler, Harvey G. Lehtman, Kirk E. Kelley, Laura E. Gould, Jeanne M. Beck, Dirk H. Van Nouhuys, James C. Norton, Delorse M. Brooks,

Joseph L. Ehardt, Raymond R. Panko, Susan Gail Roetter, Robert Louis Belleville, Rene C. Ochoa, Ann Weinberg, Adrian C. McGinnis, Robert S. Ratner, David S. Maynard, Robert N. Lieberman, Sandy L. Johnson, James H. Bair, Jeanne M. Leavitt, Rodney A. Bondurant, Jeanne M. Beck, Marcia L. Keeney, Elizabeth K. Michael, Jonathan B. Postel, Elizabeth J. Feinler, Kirk E. Kelley, N. Dean Meyer, James E. (Jim) White, Douglas C. Engelbart, Martin E. Hardy, J. D. Hopper, Charles H. Irby, Harvey G. Lehtman, James C. Norton, Jeffrey C. Peters, Dirk H. Van Nouhuys, Kenneth E. (Ken) Victor, Richard W. Watson, Don I. Andrews, Jonathan B. Postel, Priscilla A. Wold

Marilynne A. Sims, Delorse M. Brooks, Elizabeth F. Finney, Beverly Boli, Joseph L. Ehardt, James H. Bair, Robert N. Lieberman, Pat Whiting O'Keefe, James H. Bair, Robert Louis Belleville, Ann Weinberg, Thomas L. Humphrey, Jeanne M. Leavitt, Kirk E. Kelley, Duane L. Stone, Elizabeth J. Feinler, N. Dean Meyer, Dirk H. Van Nouhuys, Douglas C. Engelbart, James C. Norton, Richard W. Watson, Charles H. Irby, David C. Smith, Mary Ann Kellan, Buddie J. Pine, Andy Poggio, David L. Retz, Laura J. Metzger, Karolyn J. Martin, Jan A. Cornish, Larry L. Garlick, Priscilla A. Wold, Pamela K. Allen, Delorse M. Brooks, Beverly Boli, Rita Hysmith, Log Augmentation

KEV 12-SEP-75 09:18 26455

ARC and the military

references: (dce's -- 33470,) (dyn's -- 33442,) (kirk's -- 33469,) (chi's -- 26448,) KEV 12-SEP-75 09:18 26455

ARC and the military

it appears to be that time of year again where most of us are faced with the issue of rationalizing who we work for. This issue seems to raise its head once or twice every year (not that the issue ever really disappears, just that it gets more prominent once or twice a year). this has been going on for the five years that i have worked here (and for all i know for the entire lifetime of ARC). i would like to record my feelings on the matter now. 1

my thoughts on the matter may be vague, and possibly not totally consistant ("consistancy is the hobgoblin of small minds"). I personally have never completely thought through the issue and have never really been faced with a decision of working on something so abhorrent as to be forced to think it through completely. However, i do have feelings and "intuitions" on the subject. 2

I don't object strongly to taking money from DoD, or the "military", "intelligence" "complex". I would prefer that we got funded elsewhere. 3

What i mind the most is that even though our work belongs in the public domain, and none of it is secret, the end users of our system and technology belong almost exclusively to the "military complex" and this situation (while admittedly changing, albeit very slowly) would appear to be "worsening" in the near future. (NSW is clearly to support the "military complex" and we are currently courting NSA.) 4

There are many other end users that i would rather see our system being used by and developed for, e.g. NIH, EPA, the legal profession, the medical profession, "housewives", congress, to name but a few. 5

For the past 5 years, we have been told that it takes time and energy to cultivate these other areas and that in the meantime, for our own survival, we must continue our present vector (see doug's threat in 33470). 5 years is a long time. 6

Soon after i first came here we had a number of full ARC meetings to discuss the issues. Perhaps it is time to do this again, and if need be, get more people involved with doug in seeking alternative funding. 7

ARC and the military

(J26455) 12-SEP-75 09:18;;;; Title: Author(s): Kenneth E. (Ken) Victor/KEV; Distribution: /SRI-ARC([INFO-ONLY]) ; Sub-Collections: SRI-ARC; Clerk: KEV;

26455 Distribution

Douglas C. Engelbart, Martin E. Hardy, J. D. Hopper, Charles H. Irby, Harvey G. Lehtman, James C. Norton, Jeffrey C. Peters, Dirk H. Van Nouhuys, Kenneth E. (Ken) Victor, Richard W. Watson, Don I. Andrews, David C. Smith, Mary Ann Kellan, Buddie J. Pine, Andy Poggio, David L. Retz, Laura J. Metzger, Karolyn J. Martin, Jan A. Cornish, Larry L. Garlick, Priscilla A. Wold, Pamela K. Allen, Delorse M. Brooks, Beverly Boli, Rita Hysmith, Log Augmentation, Joseph L. Ehardt, Raymond R. Panko, Susan Gail Roetter, Robert Louis Belleville, Rene C. Ochoa, Ann Weinberg, Adrian C. McGinnis, Robert S. Ratner, David S. Maynard, Robert N. Lieberman, Sandy L. Johnson, James H. Bair, Jeanne M. Leavitt, Rodney A. Bondurant, Jeanne M. Beck, Marcia L. Keeney, Elizabeth K. Michael, Jonathan B. Postel, Elizabeth J. Feinler, Kirk E. Kelley, N. Dean Meyer, James E. (Jim) White

BEV 12-SEP-75 13:37 26456

more Thoughts About Working for the 'Intelligence' Commmunity (ff. DVN 33442, CHI 26448, KIRK 33469)

Dirk, Kirk, and Charles have all raised important objections to working for the CIA, NSA, etc. What is so disturbing to me is that their objections--and I assume those of other people in ARC--have been known for some time, yet steps to get funding from these sources are taaken at a steady pace. At the moment when we are to begin writing serious proposals for these agencies, and are running out of funding from traditional sources (which are not seen as the best to begin with), it places those who object in a Cassandra role. No one wants to see lay-offs at ARC because we can'T dig up the cash, but people who object to the current direction in which we are seeking funding are placed in a position of choosing that alternative. Why are there no other choices? Why can't we all pitch in to find other groups who would be interested in our work? When Elizabeth spoke about having an agency like the EPA sponsoring us rather that military or intelligence agencies, the idea made everything we are trying to do here seem much more meaningful and satisfying. Is this matter of funding sources still an open issue for dialog and action, or are events going to simply take their course to the accompaniment of a few voices crying in the wilderness? 1

more Thoughts About Working for the 'Intelligence' Commmunity (ff. DVN 33442, CHI 26448, KIRK 33469)

(J26456) 12-SEP-75 13:37;;;; Title: Author(s): Beverly Boli/BEV; Distribution: /SRI-ARC([INFO-ONLY]) ; Sub-Collections: SRI-ARC; Clerk: BEV;

26456 Distribution

Douglas C. Engelbart, Martin E. Hardy, J. D. Hopper, Charles H. Irby, Harvey G. Lehtman, James C. Norton, Jeffrey C. Peters, Dirk H. Van Nouhuys, Kenneth E. (Ken) Victor, Richard W. Watson, Don I. Andrews, David C. Smith, Mary Ann Kellan, Buddie J. Pine, Andy Poggio, David L. Retz, Laura J. Metzger, Karolyn J. Martin, Jan A. Cornish, Larry L. Garlick, Priscilla A. Wold, Pamela K. Allen, Delorse M. Brooks, Beverly Boli, Rita Hysmith, Log Augmentation, Joseph L. Ehardt, Raymond R. Panko, Susan Gail Roetter, Robert Louis Belleville, Rene C. Ochoa, Ann Weinberg, Adrian C. McGinnis, Robert S. Ratner, David S. Maynard, Robert N. Lieberman, Sandy L. Johnson, James H. Bair, Jeanne M. Leavitt, Rodney A. Bondurant, Jeanne M. Beck, Marcia L. Keeney, Elizabeth K. Michael, Jonathan B. Postel, Elizabeth J. Feinler, Kirk E. Kelley, N. Dean Meyer, James E. (Jim) White

printer spooler suggested mod

Since we are now running printer spoolers on at least 3 different systems (that i am aware of) i propose the following modification to the spooler program: 1

have the spooler program on each host note when it got control of the printer, and have the spooler relinquish its control after some arbitrary time (say 15 minutes). 1a

thus no one host would monopolize the printer. 1b printer spooler suggested mod

(J26457) 12-SEP-75 11:41;;;; Title: Author(s): Kenneth E. (Ken) Victor/KEV; Distribution: /JCP([ACTION]) SRI-ARC([INFO-ONLY]) ; Sub-Collections: SRI-ARC; Clerk: KEV;

26457 Distribution

James E. (Jim) White, Douglas C. Engelbart, Martin E. Hardy, J. D. Hopper, Charles H. Irby, Harvey G. Lehtman, James C. Norton, Jeffrey C. Peters, Dirk H. Van Nouhuys, Kenneth E. (Ken) Victor, Richard W. Watson, Don I. Andrews,

Jeffrey C. Peters, David C. Smith, Mary Ann Kellan, Buddie J. Pine, Andy Poggio, David L. Retz, Laura J. Metzger, Karolyn J. Martin, Jan A. Cornish, Larry L. Garlick, Priscilla A. Wold, Pamela K. Allen, Delorse M. Brooks, Beverly Boli, Rita Hysmith, Log Augmentation, Joseph L. Ehardt, Raymond R. Panko, Susan Gail Roetter, Robert Louis Belleville, Rene C. Ochoa, Ann Weinberg, Adrian C. McGinnis, Robert S. Ratner, David S. Maynard, Robert N. Lieberman, Sandy L. Johnson, James H. Bair, Jeanne M. Leavitt, Rodney A. Bondurant, Jeanne M. Beck, Marcia L. Keeney, Elizabeth K. Michael, Jonathan B. Postel, Elizabeth J. Feinler, Kirk E. Kelley, N. Dean Meyer

1st rough draft of the Programmers' Guide to Operating System Modules for the Debugger

this document does not contain some of the overall debugger documentation that the Programmers' Guide to Language Modules contained as I envision combing both of them, with a couple of other chapters, into one Programmers' Guide to the Debugger.
I invite all comments, and am especially interested in design oversights. (I am aware of the lack of the routines for setting and removing breakpoints, and will specify these later.)

1st rough draft of the Programmers' Guide to Operating System Modules for the Debugger

INTRODUCTION 1

An Operating System Module (OSM) is that module in the debugger that is responsible for all reading and manipulation of the address space and state information of a target process. Each OSM in the debugger is designed to: 1a

be run under a specific operating system, and to 1a1

read and/or manipulate the address space and state information of any target processes running running under a (potentially different) system; e.g. there is one OSM for running under a TENEX to manipulate target processes running under ELF. 1a2

An OSM is loaded dynamically by the debugger dispatcher (DD) in response to certain commands by a user. Each OSM is responsible for providing a number of routines (with well defined interfaces) and has available to it a number of routines and data structures in the DD and language modules (LM) (also with well defined interfaces). 1b

This document is a programmers guide to writing and implementing operating system modules (with specific detail to writing OSMs to run under a TENEX environment). 1c

GROSS STRUCTURE OF AN OPERATING SYSTEM MODULE 2

An OSM consists basically of a dispatch table, routines and data structures that will be called and referenced by other modules of the debugger (hereafter refered to as external routines and data structures), and any routines and data structures (hereafter refered to as support routines and data structures) needed for the support of the external routines and data structures. 2a

At the heart of any OSM is its dispatch table. The dispatch table contains: 2b

addresses of external routines, and 2b1

addresses of external data structures, and 2b2

in some instances, a dispatch table entry is itself an external data structure. 2b3

When the debugger dispatcher or a language module has a need to read or write the address space or state information of a target process, the DD or LM looks in the OSM dispatch table for the address of the OSM routine that supports the requested function.

The DD or LM will then call the OSM routine and expect the OSM routine to perform its function and optionally to return some results. 2c

(All functions discussed in this document (with the execption of the initialization routine) MUST be provided by an OSM.) 2c1

(In fact, a LM need only lookp up entities in the DD's dispatch table as the DD will, as part of its initialization sequence prior to loading a LM, copy the relavent entries from the OSM's dispatch table into the DD's dispatch table. Thus, a LM need only know about the DD's dispatch table.) 2c2

To perform its function, a operating system module routine may find it necessary to call routines provided by the DD and/or the language modules, or to refernce data structures in these other modules. To do so, the OSM routine will use the dispatch table for these other modules and can thus call or reference routines and/or data structures that it does not provide itself. 2d

GENERATING AN OPERATING SYSTEM MODULE 3

The following discussion is specific for generating an operating system module designed to run under TENEX. However, the principles involved are the same regardless of what operating system the OSM will be run under. 3a

Operating system modules designed to run under TENEX live in the address space of the debugger in pages 400(octal) - 757(octal). The operating system module dispatch table MUST be the first thing in each operating system module. 3b

An operating system module is a TENEX SSAVE file that will be "GET"ted at the appropriate time. 3c

To generate am operating system module, the debugger loader must be used. The debugger loader contains: 3d

debugger-wide definitions, 3d1

the L10 runtime environment (for the debugger dispatcher and any other modules written in L10), and 3d2

the debugger frontend to backend communication package. 3d3

The following are the current TENEX and debugger loader commands to generate an operating system module (comments are bracketed by percent signs; atsign is the TENEX prompt character indicating willingness to accept a TENEX command; asterick is the debugger loader prompt character indicating willingness to accept a debugger loader command; upper case refers to primitives that are discussed below): 3e

```
@get <nsw-debugger>beldr.sav       % get the debugger loader %      3e1

@reenter                           % start it properly %            3e2

*/400000o          % start loading the language module at page
240 %                                                               3e3

*FILE1                                                              3e4

...                                                                 3e5

*FILEn                                                              3e6

*<ALTMODE>         % done loading command to the loader %           3e7

@OPTIONAL                                                           3e8

@SSAVE (pages from) 400 (to) 757 (on)
<nsw-debugger>FILENAME.sav                                          3e9
```

FILE1 ... FILEn are the rel files that comprise the operating system module. NOTE THAT THE DISPATCH TABLE MUST BE THE FIRST CELLS LOADED. 3f

OPTIONAL is an oppurtunity for the operating system module to perform some pre-initialization (such as saving the symbol table pointer in the 2nd entry in the dispatch table). 3g

FILENAME consists of the name of the operating system the OSM is designed to run on, followed by a dash (-), followed by the name of the target process' operating system name. Thus an OSM desinged to run under TENEX and to interface to target process' running under ELF would have the FILENAME: TENEX-ELF 3h

THE DISPATCH TABLE 4

The symbolic offset names for the entries in the operating system module dispatch table are contained in the file <nsw-debugger>osidsp.nls. Also, the debugger loader contains these definitions. (Note that an offset of 0 refers to the first entry in the dispatch table, i.e. the word loaded at 400000(octal) on a TENEX.) 4a

decimal offset	symbolic offset name	meaning	4b
0	osini	address of initialization procedure	4b1
1	ossymp	symbol table pointer for this module	4b2
2	osbpte	address of procedure to call when a breakpoint is hit	4b3
3	osbptl	address of procedure to call prior to resuming from a breakpoint	4b4
4	osmsta	address of coroutine to perform "memstat" function	4b5
5	osrd1w	address of procedure to read 1 word from the target process' address space	4b6
6	----	RESERVED FOR FUTURE USE	4b7
7	osrdnw	address of procedure to read n words from the target process' address space	4b8
8	----	RESERVED FOR FUTURE USE	4b9
9	oswr1w	address of procedure to write 1 word in the target process' address space	4b10
10	----	RESERVED FOR FUTURE USE	4b11
11	oswrnw	address of procedure to write n words in the target process' address space	4b12
12	----	RESERVED FOR FUTURE USE	4b13
13	ossrcm	address of procedure to do content searches in the target process' address space	4b14
14	----	RESERVED FOR FUTURE USE	4b15

15	osgpfs	address of procedure to get the state vector of the target process	4b16
16	osspfs	address of procedure to set the state vector of the target process	4b17
17-26	----	RESERVED FOR FUTURE USE	4b18
27	oscfrk	internal handle of current target process	4b19
28-77	ossvec	state vector for the current target process; ossvec is the symbolic offset for the first word of the state vector	4b20

MOTIVATION FOR AN OPERATING SYSTEM MODULE 5

Any interactive debugger must provide facilities for examining and manipulationg the address space and state informaton of the process it is debugging. The function of the OSM is to isolate all such code into a single module with a well defined interface, and to force routines in the debugger that perform such examinations and manipulations, to use routines provided by the OSM. 5a

The isolation of these functional routines into a single module then makes it possible to dynamically load the module or to replace one module with another. Thus it becomes possible to debug either processes that live on the same machine as the debugger by using one OSM, or to debug processes on a remote machine merely by loading the appropriate OSM. 5b

This is possible, e.g., because a debugger routine that wishes to examine a cell in the address space of a target process always calls a routine in the OSM to return the contents of the specific cell, rather than the debugger routine reading the cell directly. The debugger routine need not know how the OSM obtained the contents of the cell, or for that matter, whether or not the target process is a process on the same machine as the debugger. 5b1

How the OSM performs its tasks is of no concern to the rest of the debugger. All that really matters is that the routines in the OSM obey the well defined interface conditions. 5c

Thus, one OSM designed to run under TENEX and the MSG system, and designed to debug processes under the same environment, may perform many of its tasks by sharing pages with the target process; 5c1

another OSM designed to run under TENEX and the MSG system, and designed to debug processes running under ELF, may perform its functions by using an "ELF" protocol for target process examination and manipulation; 5c2

and yet another OSM designed to run under TENEX and the MSG system, and designed to debug processes running on a YUK-7, may perform its functions by using a "YUK-7" protocol. 5c3

Yet, in all these cases, the higher level debugger routines responsible for interpretting the bits of the target process and presenting information to the user, does not need to know how the OSM performed its tasks. 5d

(This is an oversimplification, as indeed the higher level routines must be aware of such things as the word size, etc. of the target process, but the higher level routines do NOT need to know how the bits were obtained.) 5d1

GENERAL DISCUSSION OF TARGET PROCESS MANIPULATION 6

An OSM may perform its functions in any manner it chooses as long as it obeys the specified interface conditions. The address space allocated to the OSM (under TENEX) is fairly large, and the functions provided by an OSM are fairly simple and should not require much code. It is therefore expected that an OSM will use most of its address space to keep local copies of the address space of the target process in its own address space. Thus, when a higher level debugger routine requests the reading of a cell, the OSM may already have a copy of the cell and may not have to go to the target process to obtain the cell. The management of its own address space is entirely up to the OSM. 6a

(The OSM designed to run under TENEX and the MSG system and designed to debug processes on the same TENEX, also under MSG, will share pages with the target process. This OSM keeps a local map of what pages it currently has locally.) 6b

An OSM that is designed to debug processes running on a separate machine from the one on which the debugger is running must, of course, get a fresh copy of the target process memory and state information each time a breakpoint or tracepoint is encountered (or when a new target process is specified), since there is no guarantee that its local copy will be valid after the target process has been allowed to execute for any length of time. 6c

The state vector is the mechanism by which the higher level debugger routines examine and modify the state of the target process. There will be one fixed format state vector for each machine that a target process can run on, and this format will be published and known by the LMs and the DD. 6d

Thus there will be one state vector format for any process running on a TENEX; 6d1

one state vector format for any process running on an ELF; 6d2

etc. 6d3

(The first 16 words of the TENEX state vector consist of the target process' registers. However, since the PDP-10 considers the registers to be part of the address space of an active process, the registers may be read and/or manipulated either by state vector manipualtion or by normal address space manipulation and the specification of the appropriate addresses.) 6e

(More detail on state vectors to be specified later.) 6f

DETAILED DISCUSSION OF EACH ENTRY IN THE OPERATING SYSTEM MODULE DISPATCH TABLE 7

This section will discuss in detail each entry in an operating system module's dispatch table. Each entry will be discussed under its symbolic offset name. 7a

osini 7b

entry type - procedure address 7b1

procedure function (brief) - 7b2

perform module initialization 7b2a

when called - 7b3

This procedure is called (once and once only) after the operating system module has been loaded by the debugger. 7b3a

arguments - 7b4

1st argument: 7b4a

The address of the debugger dispatch table. 7b4a1

2nd argument: 7b4b

The address of an output string to be used for potential error conditions or for presenting an initialization message to the user. 7b4b1

results - 7b5

1st result: 7b5a

If initialization is successful then this procedure should return TRUE as its first result; if initialization is not successful, then this procedure should return FALSE as its first result. In either case, this procedure may write the output string (whose address is passed as the 2nd argument) with a message to be presented to the user. 7b5a1

error conditions - 7b6

Any error conditions detected by this procedure should either be dealt with by this procedure or translated into a FALSE return with an appropriate error message written in the output string. 7b6a

discussion - 7b7

The function of this procedure is to perform any module initialization required by the operating system module. 7b7a

This procedure may wish to copy entries from the DD's dispatch table into local variables to speed up future references. This is not necessary, but merely an efficiency consideration, as the address of the DD dispatch table, and its entries (with the exception of those entries that are simple data data structures), is guaranteed not to change for the lifetime of an instance of an OSM. 7b7b

ossymp 7c

entry type - symbol table pointer 7c1

discussion - 7c2

This entry is a symbol table pointer for the symbol table for the OSM. (For most languages running on a TENEX this consists of the lefthalf of the word being a negative count of the number of words in the symbol table and the righthalf of the word being the address of the first word of the symbol table.) This entry is not used by the debugger, but is merely a convience to aid in the debugging of the OSM itself. 7c2a

(Note that the OSM symbol table must reside in the same part of the debugger address space allocated to the OSM.) 7c2a1

osbpte 7d

entry type - procedure address 7d1

procedure function (brief) - 7d2

Perform any operating system and/or module specific action required at breakpoint hit (and other, see below) time(s). 7d2a

when called - 7d3

This procedure will be called: 7d3a

when a target process is specified, and 7d3a1

when a breakpoint is encountered in the target process, and 7d3a2

when a tracepoint is encountered in the target process. 7d3a3

arguments - 7d4

1st argument: 7d4a

a value (n) that indicates why the procedure is being called this time as follows: 7d4a1

n = 0: user has just specified a target process 7d4a1a

n > 0: n is the number of the breakpoint just encountered 7d4a1b

n < 0: -n is the number of the tracepoint just encountered 7d4a1c

results - 7d5

NONE 7d5a

error conditions - 7d6

NONE 7d6a

discussion - 7d7

The function of this procedure is to obtain the state vector of the target process and to build any support data structures that may be required to reflect this state, 7d7a

This procedure may wish to remove from the address space any code that was inserted to implement breakpoints (or tracepoints) and replace such code with the original code, 7d7b

This procedure will be called after a breakpoint or tracepoint is encountered in the target process or when a new (or the first) target process is specified by the user. 7d7c

osbptl 7e

entry type - procedure address 7e1

procedure function (brief) - 7e2

address of procedure to call prior to resuming from a breakpoint or tracepoint 7e2a

when called - 7e3

to be specified later 7e3a

arguments - 7e4

to be specified later 7e4a

results - 7e5

to be specified later 7e5a

error conditions - 7e6

to be specified later 7e6a

discussion - 7e7

This procedure is responsible for restoring the state of the target process to its pre- breakpoint/tracepoint state (as modified by any commands given while in the breakpoint/tracepoint state). This procedure is also responsible for inserting into the address space of the target process any code needed to implement breakpoints and/or tracepoints that are currently set. 7e7a

more detail to be specified later 7e7b

osmsta 7f

entry type - coroutine address 7f1

coroutine function (brief) - 7f2

The function of this coroutine is to build strings (using the current output mode) for the display of the address space of the target process. (This coroutine implements the TENEX "MEMSTAT" command.) 7f2a

when called - 7f3

This coroutine will be called by a LM in response to a user's request to display the gross address range of type dadr (see -- xxx,). 7f3a

arguments - 7f4

at openport time - 7f4a

NONE 7f4a1

at cycle start time - 7f4b

1st argument: 7f4b1

the address of the output string that should be written by this coroutine 7f4b1a

2nd argument: 7f4b2

the address of the current output mode record to be used in the building of the output string 7f4b2a

at pulse after a positive return - 7f4c

NONE 7f4c1

at pulse after a negative return - 7f4d

NONE 7f4d1

(This coroutine should not generate any negative returns.) 7f4d1a

results - 7f5

at openport time - 7f5a

NONE 7f5a1

all other times - 7f5b

returns > 0 means it has written the output string and that string should be presented to the user. it is not finished and expects to be called again with no arguments. 7f5b1

returns = 0 means it is done, i.e. it has completed the current cycle; the output string may be NULL or if non null, then it has detected an error and the output string is the error condition to be presented to the user. in either case it may be called again, but it must be presented fresh arguments. 7f5b2

(Note that if this error return occurs on a "first" pulse, it more than likely means that this coroutine got invalid or unsupported address range elements.) 7f5b2a

error conditions - 7f6

any error conditions detected by this coroutine should be translated into a 0 return with an appropriate error message written in the passed output string. The coroutine should then be ready to accept new arguments to start a new cycle. 7f6a

discussion - 7f7

This coroutine is used to build strings (for eventual display to a user) that describe the gross utilization of the address space of the target process. 7f7a

This coroutine should build a line of information for each pulse that corresponds to the equivalent lines of a TENEX "MEMSTAT" command. 7f7b

(To be expanded later.) 7f7b1

osrd1w 7g

entry type - procedure address 7g1

procedure function (brief) - 7g2

This procedure is used to read one word in the address space of the target process. 7g2a

when called - 7g3

This procedure will be called whenever a LM or a DD routine wishes to examine (for whatever reason) a word in the address space of the target process. 7g3a

arguments - 7g4

1st argument: the address of the word the LM or DD wishes to read 7g4a

results - 7g5

1st result: 7g5a

a value (n) indicating the success or failure of this routine as follows: 7g5a1

n = 0: word read successfully 7g5a1a

n < 0: invalid address passed to this routine 7g5a1b

n > 0: address passed to this routine represents a non-existant page in the target process 7g5a1c

2nd result: 7g5b

the contents of the addressed word, or 0 on error conditions 7g5b1

error conditions - 7g6

Any error conditions detected by this procedure should either be dealt with by this procedure or translated into the appropriate error return. 7g6a

discussion - 7g7

The function of this procedure is to read, and return, the contents of the passed address in the target process' address space. 7g7a

osrdnw 7h

entry type - procedure address 7h1

procedure function (brief) - 7h2

This procedure is used to read one or more words in the address space of the target process. 7h2a

when called - 7h3

This procedure will be called whenever a LM or a DD routine wishes to examine (for whatever reason) one or more words in the address space of the target process. 7h3a

arguments - 7h4

1st argument: the address of the first word to read 7h4a

2nd argument: the number of words the to read 7h4b

3rd argument: an address at which to store the read words 7h4c

results - 7h5

1st result: 7h5a

the number of words read and returned to this routine's caller 7h5a1

2nd result: 7h5b

a value (n) indicating the success or failure of this routine as follows: 7h5b1

n = 0: words read successfully 7h5b1a

n < 0: invalid address passed to this routine 7h5b1b

n > 0: address passed to this routine represents a non-existant page in the target process 7h5b1c

error conditions - 7h6

Any error conditions detected by this procedure should either be dealt with by this procedure or translated into the appropriate error return. 7h6a

discussion - 7h7

The function of this procedure is to read, and return, the requested words in the target process' address space. 7h7a

oswr1w 7i

entry type - procedure address 7i1

procedure function (brief) - 7i2

This procedure is used to write one word in the address space of the target process. 7i2a

when called - 7i3

This procedure will be called whenever a LM or a DD routine wishes to write (for whatever reason) a word in the address space of the target process. 7i3a

arguments - 7i4

1st argument: the address of the word the LM wishes to write 7i4a

2nd argument: the value to be written in the addressed word 7i4b

results - 7i5

1st result: 7i5a

TRUE if the word was written successfully; FALSE otherwise. 7i5a1

error conditions - 7i6

Any error conditions detected by this procedure should either be dealt with by this procedure or translated into the appropriate error return. 7i6a

discussion - 7i7

The function of this procedure is to write the passed value at the passed address in the target process' address space. 7i7a

oswrnw 7j

entry type - procedure address 7j1

procedure function (brief) - 7j2

This procedure is used to write one or more words in the address space of the target process. 7j2a

when called - 7j3

This procedure will be called whenever a LM or a DD routine wishes to write (for whatever reason) one or more words in the address space of the target process. 7j3a

arguments - 7j4

1st argument: 7j4a

the address, in the target process' address space, of the first word to write 7j4a1

2nd argument: the number of words to write 7j4b

3rd argument: 7j4c

an address, in the debugger's addres space, from which to get successive words to write in the target process' address space 7j4c1

results - 7j5

1st result: 7j5a

TRUE if the words were written successfully; FALSE otherwise. 7j5a1

2nd result: 7j5b

the number of words actually written if the first result is FALSE 7j5b1

error conditions - 7j6

Any error conditions detected by this procedure should either be dealt with by this procedure or translated into the appropriate error return. 7j6a

discussion - 7j7

The function of this procedure is write the requested words in the target process' address space. 7j7a

ossrcm 7k

entry type - procedure address 7k1

procedure function (brief) - 7k2

This procedure will search the target process' address space between 2 passed addresses (inclusively) for cells that contain the passed value (after both have been masked appropriately). 7k2a

when called - 7k3

This procedure will be called by the LM infmem coroutine to perform content searches in the address space of the target process. 7k3a

arguments - 7k4

1st argument: a starting address, in the target process' address space 7k4a

2nd argument: an ending address, in the target process' address space 7k4b

3rd argument: value to search for 7k4c

4th argument: mask to apply to search value and words in target process 7k4d

5th argument: TRUE for a content search; FALSE for a not content search 7k4e

results - 7k5

1st result: TRUE if this procedure found a word that met the passed requirements; FALSE if not. 7k5a

2nd result: the address of the found word on success; FALSE otherwise. 7k5b

3rd result: the contents of the found word on success; indeterminate otherwise 7k5c

error conditions - 7k6

Any error conditions detected by this procedure should either be dealt with by this procedure or translated into the appropriate error return. 7k6a

discussion - 7k7

The function of this procedure is to search the inclusive address bounds, in the target process' address space, for the passed value. Both the passed value, and cells in the target process' address space will be masked (logically ANDed) with the the passed mask, before any compares are preformed. This procedure can succeed if the resulting compares are equal if this was a content search; and it can also succeed if the resulting compares are not equal and a not content search was specified. 7k7a

osgpfs 7l

entry type - procedure address 7l1

procedure function (brief) - 7l2

This procedure is used to obtain the state vector of the target process and to write the obtained state vector in the cells allocated for the state vector in the OSM's dispatch table. 7l2a

when called - 7l3

This procedure will normally only be called by the OSM's osbpte routine. 7l3a

arguments - 7l4

NONE 7l4a

results - 7l5

NONE 7l5a

(This procedure must write the appropriate cells in the OSM's dispatch table with the appropriate state vector.) 7l5a1

discussion - 7l6

This function of this procedure is to obtain the state vector of the target process and to write the obtained state vector in the cells allocated for the state vector in the OSM's dispatch table (see discussion below). 7l6a

osspfs 7m

entry type - procedure address 7m1

procedure function (brief) - 7m2

This procedure is used by LM or DD routines to modify the state vector of a target process. 7m2a

when called - 7m3

This procedure will be called when a LM or DD routine wishes to modify the state vector of a target process. Neither LM nor DD routines should modify the state vector directly, but they must use this procedure.. 7m3a

arguments - 7m4

To be specified later. 7m4a

results - 7m5

To be specified later. 7m5a

discussion - 7m6

To be specified later. 7m6a

oscfrk 7n

entry type - simple data strucure 7n1

data structure name - CURFORK 7n2

data structure meaning - 7n3

this is a debugger internal name for the target process that is currently being debugged. 7n3a

data structure type - 7n4

this data structure is a single word in the OSM's dispatch table 7n4a

discussion - 7n5

this data structure will be set and maintained by the DD. It consists of an internal debugger handle for the current target process. 7n5a

(More detatiled to be specified later.) 7n5b

ossvec 7o

entry type - 50 word data structure 7o1

data structure name - PFSTATE 7o2

data structure meaning - 7o3

this data structure contains the state vector of the current target process. 7o3a

data structure type - 7o4

this data structure is composed of 50 words (under TENEX, the first 16 of these words represent the registers of the target process; the meaning of the rest of the words will be specified later.) 7o4a

1st rough draft of the Programmers' Guide to Operating System Modules for the Debugger

(J26458) 12-SEP-75 12:49;;;; Title: Author(s): Kenneth E. (Ken) Victor/KEV; Distribution: /NPG([INFO-ONLY]) RWW([INFO-ONLY]) ; Sub-Collections: SRI-ARC NPG; Clerk: KEV; Origin: < NSW-DEBUGGER, !DOC/OSI-MODULES.NLS;1, >, 12-SEP-75 12:41 KEV ;;;;####;

26458 Distribution

David C. Smith, Andy Poggio, David L. Retz, Jan A. Cornish, Larry L. Garlick, Robert Louis Belleville, Elizabeth J. Feinler, Joseph L. Ehardt, Jonathan B. Postel, Kirk E. Kelley, Karolyn J. Martin, David S. Maynard, Kenneth E. (Ken) Victor, James E. (Jim) White, Elizabeth K. Michael, Don I. Andrews, J. D. Hopper, Charles H. Irby, Harvey G. Lehtman, Richard W. Watson, Gunter Report for week ending 9/14/75

The Week Ending 9/14/75 1

66-1 1a

Pete and Lynne ran the rewrite of 66-1, volume 6. It went quite smoothly in that they were able to get fresh drafts when the writers wanted them. Joanne and Grace did the editing and that went well. We produced the first formatted copies on Thursday and these also had SID's on them. The writers were very pleased and commented that this rewrite went much more smoothly than any of the previous ones they had worked on. 1a1

PR 1b

Cindy and Jo have just about finished the first input for both the PR documents. We had inserted the text with the understanding that we would use the same format as the 66-1 series. We discovered late this week, that that format is not acceptable and that a new format program will have to be developed. I met with Capt. Grandy and Mr. Brown who work in Standards. Apparently, I was previously directed to talk to the wrong person. Standards are the ones who decide how the format will look. Quality control (Mr. Fisher) makes sure the standards are followed. I met with Standards twice and they will issue a letter to Larry Crain and other appropriate people stating that the 66-1 format will not go and stating the specifications of their format. They will put in writing that this format will be acceptable for all future Gunter Documents except those that need the 66-1 format. 1b1

A meeting has been set for Tuesday of next week for Standards, Quality Control and me to get the specifications in detail. 1b2

Base Tops--the PR annexes 1c

The PR annexes were one of the few that were turned in on time and the format was entirely accepted. They were very pleased with the results as were the BAse Top people who were coordinating the entire effort. 1c1

Demonstation for Monday 1d

A Demonstartion will be given monday morning at 9:oo to Col Bruner who is the commander of the whole base. By the time you all in California are at work, it will be all over.... 1d1

General 1e

There was some problem with ELF at the start of the week, but by the end, it was running well. There is still only one DNLS terminal as we are waiting for some parts from Datamedia. Two of the Ti's are broken and the TI man said he would be here today..it has now been posponed until Monday morning.The wires for the new DNLS station exist and we have been running a TI on them. The lines are, however, not the best and there is alot of noise coming in on them. The morale is quite high. All of the people who were in California are doing very well and are quite excited to keep learning more. The PR document is our weakest point at the moment and now that a new format has to be developed, we will be working extra hard in that area. Monday morning, we start a rewrite of volume 12 of 66-1 and that is a large document....That is all from the deep south.. 1e1

Gunter Report for week ending 9/14/75

(J26460) 12-SEP-75 15:24;;;; Title: Author(s): Ann Weinberg/POOH;
Distribution: /SRI-ARC([INFO-ONLY]) LAC([INFO-ONLY]) ;
Sub-Collections: SRI-ARC; Clerk: POOH;

26460 Distribution

Douglas C. Engelbart, Martin E. Hardy, J. D. Hopper, Charles H. Irby, Harvey G. Lehtman, James C. Norton, Jeffrey C. Peters, Dirk H. Van Nouhuys, Kenneth E. (Ken) Victor, Richard W. Watson, Don I. Andrews, Lawrence A. Crain,

David C. Smith, Mary Ann Kellan, Buddie J. Pine, Andy Poggio, David L. Retz, Laura J. Metzger, Karolyn J. Martin, Jan A. Cornish, Larry L. Garlick, Priscilla A. Wold, Pamela K. Allen, Delorse M. Brooks, Beverly Boli, Rita Hysmith, Log Augmentation, Joseph L. Ehardt, Raymond R. Panko, Susan Gail Roetter, Robert Louis Belleville, Rene C. Ochoa, Ann Weinberg, Adrian C. McGinnis, Robert S. Ratner, David S. Maynard, Robert N. Lieberman, Sandy L. Johnson, James H. Bair, Jeanne M. Leavitt, Rodney A. Bondurant, Jeanne M. Beck, Marcia L. Keeney, Elizabeth K. Michael, Jonathan B. Postel, Elizabeth J. Feinler, Kirk E. Kelley, N. Dean Meyer, James E. (Jim) White

Issues in the Design of the NLS User Interface

By Richard W Watson

INTRODUCTION

The user interface has two sides: the input side by which the user inputs information, indicating by various conventions and controls what he wishes accomplished; and the output side by which the machine provides feedback and other assistance to the user in command specification, and provides various forms of information portrayal. Man has many motor and other capabilities that could be the basis for input and command specifications; similarly he has his full range of senses that could be targets for system output.

To date, computer information systems make use of only a few motor and sensory capabilities in their man-machine dialog. An important area of research involves exploring the advantages to be gained and the techniques to be used to extend this range. There is interesting research going on in areas of speech, eye movement, brain wave control, hand written script, and video graphics that will undoubtedly be integrated into the truly multimedia systems to be built in the near future.

We call the user's collection of input-output equipment and arrangement of work tables and work space, the workstation. At the present time, input centers around various types of keyboard devices: standard typewriter-type, function button, keyset (chord), and graphical pointing devices: mouse, electronic pen-tablet, light pen, joystick. The dominant output means are printers and displays of varying capabilities.

The present NLS user interface has been developed around this equipment, although many of the principles used in its design can be easily extended for use with other media [3]. The prime motivation for the use of the mouse for pointing and two keyboards, (standard typewriter-like and keyset), as the input devices for the display version NLS 7 (DNLS), are described in references [2][3]. NLS can also be used from typewriter terminals (TNLS). In this chapter, we concentrate on describing some of the motivations behind the design of the NLS command language and the forms of information portrayed to assist the user in command specifications. Forms of general NLS information portrayal are described in reference [1].

The NLS is a prototype collection of tools in a growing workshop of tools and services to aid knowledge work [1][4], and we expect the number of tools and vocabulary that controls their use to grow. We further expect that the use of such a workshop will spread throughout those occupations involved with information in various forms and that there will be infrequent and casual users of such systems, along with many people who will spend large fractions of their day using such workshops. Another goal is to match the speed of system responsiveness to the natural speed and flow of man's thought processes. It is from these basic expectations that our user interface work has developed. The sections below enumerate several assumptions and areas of concern around which the NLS user interface has developed to date. A key point to mention is that we do not consider the NLS user interface a static, finished product. It will change, based on analysis of usage experience, and the technology and media available.

HIGH LEVEL ASSUMPTIONS UNDERLYING THE DESIGN OF THE NLS USER INTERFACE

First we describe a few high-level assumptions that affect the user interface design and then discuss some of the lower level issues and the specific techniques used to deal with them.

1) Coordinated Set Of User Interface Principles

There will be a common command interaction discipline, over the many application areas in the workshop, that shapes user interface features, such as the language, control conventions, methods for obtaining help, and computer-aided training.

This commonality has two main implications. One, it means that while each domain within the core workshop area or within a specialized application system may have a vocabulary unique to its area, this vocabulary will be used within language and control structures common throughout the workshop system. A user will learn to use additional functions by increasing vocabulary, not by having to learn separate "foreign" languages. Two, when in trouble, he will invoke help or tutorial functions in a standard way.

2) Grades Of User Proficiency

A once-in-a-while user with a minimum of learning will want to be able to get at least a few straightforward things done. In fact, even an expert user in one domain will be a novice in others. Users will be clerical workers, information specialists, executives, engineers, and others. Attention to novice-oriented, and tutorial help features is required.

Users also want and deserve the reward of increased proficiency and capability from improvements in their skills and knowledge, and in their conceptual orientation to the problem domain and to their workshop's system of tools, methods, conventions, etc. "Advanced vocabularies", short concise control notation and conventions in every special domain will be important and unavoidable.

A corollary feature is that workers in the rapidly evolving augmented workshops should be involved continuously with testing and training in order that their skills and knowledge may most effectively harness available tools and methodology.

3) Ease Of Communication Between Subsets And Addition Of Workshop Domains

One cannot predict which domains or application systems within the workshop will want to communicate in various sequences with which others, or what operations will be needed in the future. Thus, results must be easily communicated from one set of operations to another, and it should be easy to add or interface new domains to the workshop. A corollary is that the total workshop may contain a very large number of tools and services. Some users may have access to only a subset of its capabilities while others will have access to many or all capabilities.

As described below, we expect the workshop to be embedded in a computer network and thus communication between tools and between users must take place across both process and host boundaries according to well specified conventions and protocols [5][6].

4) User Programming Capability Or User Interface Extensibility

There will never be enough professional programmers and system developers to build or interface all the tools that users may need for their work. Therefore, it must be possible, with various levels of ease, for users to add or interface new tools, and extend the language to meet their needs. They should be able to do this in either a variety of programming languages with which they may have training, or in the basic user-level language of the workshop itself.

5) Range Of Workstations And Symbol Representations

The range of work stations available to the user will increase in scope and capability. These work stations will support text with large, open-ended character sets, pictures, voice, mathematical notation, tables, numbers, and other forms of knowledge. Even small portable hand-held consoles will be available. The multiplicity of possible terminals indeed raises the question of whether a consistent set of control and portrayal conventions is possible.

As hardware decreases in cost, more and more capabilities will be placed in the work station both in the form of user interface aids and facilities, and in the form of frequently used tools.

6) Distributed Nature Of The User Interface Processes

The collection of facilities to support interfaces with the system of tools can be conceived of as a single service as seen by the user. These facilities may all reside in a processor in the work station or be distributed in two or more processors, depending on the level of their sophistication and state of the art with respect to cost, hardware capability, and so forth.

7) Embedded In a Computer Network

The computer-based tools of a knowledge workshop will be provided in the environment of a computer network, such as the ARPANET [7]. For instance, the core functions will consist of a network of cooperating processors performing special functions, such as editing, publishing, exchanging documents and messages, data management, and so forth. Less commonly used, but important functions, might exist on a single machine. The total computer-assisted workshop will be based on many geographically separate systems.

Once there is a "digital-packet transportation system," it becomes possible for the individual user to reach out through his processor to other people and other services scattered throughout a "community". The "labor marketplace" where he transacts his knowledge work will be literally independent of geographical location.

Specialty application systems will exist in the way that specialty shops and services now do--and for the same reasons. When it is easy to transport the material and negotiate the service transactions, one group of people will find that specialization can improve their cost/effectiveness, and that there is a large enough market within reach to support them. And, in the network-coupled computer-resource marketplace, there will be a growth of specialty shops, such as application systems specially tailored for particular types of analyses, or for checking through text for spelling errors, or for doing the text-graphic document typography in a special area of technical portrayal, and so on. There will be brokers, wholesalers, middle men, and retailers.

The key point to emphasize is that even when hardware costs decrease to the point where a user can perform 90% of his work using tools and information that operate in the processor in his work station, he will want to have access to a computer network to:

a) Communicate in various forms with others
b) Access very large or special data bases
c) Access special tools that run elsewhere

8) Problem Orientation Of The Command Language And Tolerance For Ambiguity

The user has a task that he wishes performed by the system. Depending on the nature of the task and operations available to him on the system, he may be able to express what he wants accomplished in a single "statement" or command to the machine, or it may require a series of commands.

One of the goals of the designers of the command language and system is to understand the nature of the user's application domain so that the user can express his needs with words that are similar to his natural problem solving vocabulary and language forms. The machine should then break down the request into smaller steps as required.

If there is ambiguity in the user's command, the machine should recognize it, if possible, and prompt appropriately for clarification. There is still much research and development required to fully meet this goal.

Many people hope to allow novice users or users in certain applications to use natural language in making statements to the machine. This capability will require models of the user and task domains for understanding.

Even when systems are able to interpret commands given in natural language, the precision and usage efficiency of appropriate artificial languages will make the latter's continued use preferable, especially for skilled users.

Given the above general considerations as background, we can move on to examine features of the NLS user interface in more detail.

MORE DETAILED DISCUSSION OF THE NLS USER INTERFACE

A command language must allow unambiguous specification of what the user wishes accomplished. The operation to be performed, and the entities or information items (arguments) to be acted upon, or used to determine what is to be acted upon, must be specified. These can be specified in a variety of ways: by typing them in in full or in some form of abbreviation, by pointing at them on a screen, by pronominal reference, or by use of default values where appropriate. The order of their specification, the syntax or grammar of the language, can have various forms. For example, operational keywords can be specified, followed by the arguments, or vice versa. Arguments can be in fixed positions or explicitly named and occur in any order. Some arguments or keywords can be optional and require special characters to indicate their presence. Arguments or keywords can have defaulted values under certain conditions. Pronominal references can be allowed to refer to previous occurences. Arguments may be given types by the system and language designer for more extensive error checking and feedback.

Arguments and keywords can be specified by complete or partial typein (there are a variety of forms of command recognition that are discussed later) or designated by pointing to representations on a display or by use of specially coded function keys. Or, the machine may ask questions and the user just fill in the blanks.

Depending on the characteristics of the computer and communications system, it may or may not be possible to provide command word or keyword completion, prompts or other feedback, argument checking, default value fill in, and so forth, during the command specifications.

For example, in line-at-a-time, half-duplex systems, the user usually must complete the entire specification of the command before transmission to the system, while in character-at-a-time, full-duplex systems, the system can react to each character received and provide more extensive aids to the user during command specification.

The above discussion outlines just a few of the many choices available to the language designer. As the purpose of this section is not to be a complete tutorial on all possible choices available and their advantages and disadvantages, the following discussion only gives main NLS command language features and the motivation for their adoption.

THE NLS COMMAND LANGUAGE

The NLS command language generally has the following form, where angle brackets group meta symbols:

<operation specification> <operand specification> <command completion>

The fields in a command are of a fixed order, although some commands have optional fields that can be specifically requested. Other fields can have a system-supplied default value. Because NLS operates from a character-at-a-time, full-duplex system, several levels of help are available, as described later, for giving cues and prompts, explicitly listing options or syntax, and giving full documentation on what the system expects next during command specification. It was not felt that much would be gained for novice users by allowing fields to be specified in any order by using explicit field names. Novice users do not need to be aware of optional fields.

As much as possible NLS makes the operational specification of the form verb-noun followed by arguments and possibly other keywords. We have also tried to maximize the fullness of the verb-noun matrix.

This approach seemed to be natural, and follows normal English imperative forms to aid learning. The choice of verb-noun form seemed to fall out naturally when considering such important areas as editing. A given verb, such as DELETE, can naturally be applied to many entities, such as statement (a paragraph, title, equation), character, number, text, file etc. Learning is easier if the user can form a model of how the system works that can be consistently applied. In this case, a user can learn n verbs and m nouns and understand that generally, if it is meaningful, they can be used in pairs. Having learned n+m vocabulary terms, he can apply them in the form of n x m commands.

We have tried to pick command keywords that have normal usage related to the operation described. A synonym capability would be easy to implement.

Four forms of command keyword recognition are provided to enable the user to choose the one most appropriate to his terminal type, system response, previous system experience, and present NLS experience level. We have worked to pick an operational vocabulary for the present system that guarantees keywords to be unique in a maximum of three characters:

1) A single-character mode allowing high-speed single-character recognition of the most commonly used commands; less commonly used commands require an escape character followed by enough characters for unique recognition: with large and expanding command sets one cannot choose keywords with mnemonic value and guarantee uniqueness with the first character. This mode is generally preferred by experienced users because of the simplicity and speed with which frequently used operations can be expressed. We find that experienced users are very concerned that commands be formed with the minimum number of input operations, and that commands have the richness needed to specify adjective or adverb type operations as needed. There is thus some conflict in certain commands between these goals for the experienced user and the need for command simplicity for the novice.

2) A demand mode requiring a right delimiter to initiate recognition: This has proved to be popular for new users of typewriter terminals, particularly those with experience using the TENEX operating system. Modes C and D have not turned out to be heavily used.

3) An anticipatory mode requiring the user to type enough characters until the command is uniquely specified; the system then automatically fills in the remainder.

4) A fixed mode that guarantees recognition on entry of three characters.

Given the implementation approach outlined later, it is quite easy to add other recognition modes, such as allowing the user to choose keywords from a menu displayed on the screen. However, experiments have shown that the time it takes to point at some item on the screen is equivalent to several keystrokes and thus would be disadvantageous to skilled users, although possibly of value to novices [2][3].

Operand argument specification is contained in a number of fields that are variable with the type of command. All commands of a similar type have had the order of the operands made as consistent and as natural (relative to normal English usage) as possible. Infrequently used operand fields are optional and novice users need not be aware of their existence.

Related to argument specification is the problem of choosing argument delimiters. One can recognize the following delimiting functions.

1) Delimiting command words
2) Delimiting arguments
3) Delimiting optional arguments, selection type, or command word fields
4) Delimiting commands
5) Selecting arguments off a display screen, and confirming the selections

One could choose separate characters (codes) to represent each of these functions. To do so seemed to us to add an unnecessary complication for the user and so, except for using a special character to indicate an optional argument, selection type, or command word, a single code is used for the other function in NLS. We call this code "Command Accept" (CA) even though it is used for other purposes as well. The system allows the user to define which keyboard character is to serve this function if he finds the system default to be inconvenient. One of the buttons on the mouse also serves this function.

Arguments can be typed in, defaulted where appropriate, or specified by pointing to appropriate entities on the display screen.

There are three flavors of command completion.

1) Completion of the command indicating execute the command and return to the base state to await input of the next command: The default indication for this form is one of the buttons on the mouse in DNLS, which is translated into a control character, or CR in TNLS. The use of CR in TNLS is quite natural and generally does not conflict with textual input as most text in NLS is typed in without explicit CRs and is appropriately formatted by the system for various output devices. If the TNLS user wishes to input an explicit CR in his text file, he must precede it with an escape character. If he has need to enter many CRs in his text string, he can

redefine the completion character, Command Accept, to be some other character.

2) Completion of the command and return to an appropriate point for quick repetition of the command. Repetition mode continues until explicitly commanded to delete out of it. This mode is very useful when a delete or other operation is repeated several times.

3) Completion of the command and entry to insert-statement mode for addition of new paragraphs or other text statements: This mode is like command repeat above except that it always takes you to the insert command. It is used frequently when one adds, replaces, or moves text, and then wants to follow it with new statements. It speeds text input when inserting sequences of paragraphs.

The system is to be used from a variety of terminal types, including both typewriter-type terminals and displays. The two-dimensional displays are to be the preferred work station types whenever a design decision must be made between language forms possibly favoring one type or the other.

It was decided to make the command language syntax for the typewriter (TNLS) version and the display (DNLS) version as close as possible, except where the difference between the one-dimensional and two-dimensional media clearly prohibits this or would seriously limit one or the other version. This decision was made to allow people working in environments consisting of both typewriter and display terminals to be able to move back and forth with ease.

The system has been organized into clearly defined subsystems with uniform rules for their entry and exit. Any subsystem can be entered from any other, either to "execute" a single command with automatic return or to perform a chain of commands. The user can return, either to a specifically named subsystem in the path of subsystems traversed or enter a new subsystem. The issue of how to group commands into subsystems has to do with training and patterns of use rather than system constraints. It relates to learnability and, to some extent ease of command specification using single characters, and to "knowing where you are" in a command or operational space.

One could construct a system where all commands were in a single subsystem. Study of the command set of a large system particularly conceived of as a set of tools shows that operations tend to group together such that to perform a given task, such as sending a message or calculating a budget, generally require several related suboperations. Certain operations, such as moving in information space or seeking help, tend to be used as suboperations of many or all tasks. This latter observation has led to "universal" commands available from within any subsystems. One can also imagine certain commands to be needed frequently in just two or more subsystems and thus implemented in each subsystem having the need. There are now no instances of this case in NLS. The ability to execute a single command in another subsystem with automatic return has been very useful.

Provision has been made for user-controllable options on prompting, feedback, and other parameters whenever it seemed a single option, might not be appropriate to some significant class of users.

A mechanism is implemented that enables the user, or someone acting in his behalf, to create a file stating what options he wants to run. The system automatically sets his options when he enters. This facility can also be used with small extensions to subset commands. This user option capability, when coupled with the ease by which the user interface can be redefined using the Control Meta Language described below, makes possible tailoring the user interface to specific users or groups of users.

All operations that have a natural inverse command have been given one. (NLS still does not have an "undo" facility.) A general undo/redo facility has a number of technical difficulties and its value can be questioned. However, the ability to undo or redo the last one, two, or three commands would clearly be useful.

User Programming: As indicated earlier the ability of the user to extend the system himself is important. There is a tradeoff between ease of extension specification and operational efficiency. In providing such a facility one does not have to be deeply concerned with efficiency if the task handled by the extension is performed infrequently.

If the operation is performed frequently, then it should probably be inserted as a system feature and implemented efficiently by professionals. This area is ripe for much additional development. The extensions must be specified in some language to indicate what sequence of events is to take place, what arguments to collect, and so forth, when a given user action is performed.

NLS now offers two forms of extensibility. The first allows users with some basic programming knowledge to write programs in the Algol like L10 language in which the system is implemented, calling on NLS system primitives as needed. They can use the Control Meta Language to specify a user interface if desired. These programs can be installed by the user as part of his default subsystems, loaded as subsystems as needed, or used as content analyzer patterns [8].

The user can also write sequences of NLS commands and have these sequences executed at will. A specific sequence of commands can be automatically invoked when the user first enters NLS.

HELP, STATUS, AND PORTRAYAL FACILITIES

ORGANIZATION of the TERMINAL DISPLAY AREA

The NLS display screen is organized into windows as described in some detail in [9] These windows are arbitrary rectangles. Windows can be displayed essentially all the time or overlayed with others. Windows can grow dynamically. Some windows are allocated and displayed or not displayed under system control for status and feedback information. Others can be created and manipulated by the user for display in his information space. With typewriter terminals, one does not have this two-dimensional random display capability and while the same information can be given to him, less can be given automatically or must be given in an altered form. Let us now consider each of the information spaces and the type of feedback, help, and other status information available to the user.

1) Information space

The present NLS information space is hierarchically organized. A user has a directory or directories within which there are files. A file can contain notes on many subjects stored under various headings, his mail, or single documents. Files in turn are hierarchically organized as a tree of information nodes (now text strings but soon to be generalized to include illustrations and other entities).

Files can contain cross citations to specific points within other files or the same files, thus creating networks. NLS has appropriate commands for moving within and between files and for obtaining a display of the path over which one has traveled and commands for backtracking along this path [3].

Display screens have a limited number of lines within which to display information, and typewriters, even at 30 chars/sec or higher, cannot quickly and easily print out large documents. Also, the user often wants to see a summary or overview of a document or have it formatted in special ways to aid his understanding. To meet this need for easy control of information portrayal, NLS has a concept called "view specification". The user can change his "view" within the commands for moving in information space or by separate command. So that he can be reminded of his current view, the most commonly used view parameters are fed back to him in a small window in the upper right hand corner of the screen. When he is at a point in a command where it is permissible to change views, this fact is fedback both by prompt (if prompts are turned on) and by enlarging the characters in the view-feedback window. For more discussion on moving, viewing, and portrayal in NLS see [3][6].

2) Subsystem or tool space
NLS is viewed as a collection of tools (subsystems) that can be used cooperatively or stand alone. Each subsystem contains a number of logically related commands and has a name, such as Base (the collection of editing and file manipulating commands), Calculator, etc. All the tools work on information in the same file structure and the user can move from one tool to another, or execute commands on a single command basis in any tool from any other tool, as mentioned earlier. The user can receive a display of subsystems available to him or an ordered list of the subsystems in which he has previously been.

The current subsystem within which he is operating is fed back in a small window in the upper left-hand corner of the screen in DNLS and as a four-character prompt in TNLS.

3) Command syntax space
There are several levels of feedback and Help available to the user in formulating a command to the system (14,). Each is described below. The Help data base clearly is also generally useful for understanding the system as a whole.

a) Command keyword recognition:
The options here were described earlier and this mode is primarily useful in minimizing keystrokes and in triggering additional feedback.

b) Noise words:
When the system recognizes a keyword or field it generates what we call "noise words" set off in parentheses so the user can distinguish between what he has input and what the system has added. The noise words aid the user in remembering what to do next. Novice users report that noise words are one of the most useful initial aids. As more experience is gained, the other aids take on more importance. This is an important point to note: users at different levels of experience value different forms of feedback. Usefulness is not only determined by the inherent characteristics of the aids, but also, by how they are implemented.

c) Prompts:
When the user completes the specification of a field in a command, he is prompted with some terse characters indicating the type of thing expected next and the alternatives available to him for how he can specify, select, or address the needed argument. Users can turn prompts off, which some users of TNLS do when they reach a certain level of proficiency, although many highly skilled users always operate with them on. DNLS users tend to always operate with them on because the high speed of the display does not slow down work while providing useful information. Users can also specify terse prompting in which case optional fields are not prompted for.
Beginning users have indicated that prompting is useful,

but would like them to be more mnemonic and of word length.

d) Next Options and Syntax:
If the noise words and prompts are not sufficient to jog a user's memory about what options are available to him next, he can strike a ? or a <Control-S>. If he strikes a ?, the system displays, in alphabetical order, all the command keywords that are legitimate for the next field or more extensive information than is available in the prompts for other fields. If he strikes <Control-S>, the system prints out the syntax of the command from his present position to the end of the command. The ? facility is extensively used and is very useful in refreshing one's memory about infrequently used commands or new commands for a user with only a basic knowledge of command system concepts and vocabulary. The <Control-S> feature does not seem to be extensively used at present and may indicate that the ? facility is sufficient.

e) Help Data Base:
If the above facilities are not sufficient because of uncertainty about a basic concept or vocabulary word or the user wishes more information about the effects or use of a command, he can enter the the Help tool. Entry can be from the basic command level or from any point during command specification. In the latter case, the system utilizes the information input at this point to take the user to an initial point that describes the command and field where he is at. (15)

Once in the Help Data Base, a simple set of command conventions and the organization of the data base allow the user to easily move to reference related subjects or move to new subjects or back up to higher level descriptions (15).

f) Active Tutorial Help:
The next level of Help facility would be an active tutorial facility. We have not yet implemented such a facility but can see its value. An example of such a facility is the work going on at BBN on the NLS-Scholar system [10].

ERROR MESSAGES AND RECOVERY

Error messages indicating an incorrectly spelled file name or improperly specified entity are fed back to the user in a window at the top of the screen. The user is left at an appropriate point within the command specification or where necessary he must start over again to respecify the command. The text of error messages is important and should be as specific to the problem as possible. This has implications within the system design for trapping error conditions as early as possible and determining the appropriate message for the specific error and total context of the user. While we have made progress in this area, there is much more that could be done to meet the need stated above.

There are now no automatic error correction mechanisms built into the system, such as spelling correction or "Do What I Mean" type facilities. These would probably be useful to add when resources permit.

EDITING AND BACKUP DURING COMMAND SPECIFICATION

The user can perform certain simple editing and backup operations during command specification. At any point during command specification he can command delete, which will take him back to the basic command level. This is useful if he gets confused and wants to return to a known state or changes his mind about which command to perform next.

The user can delete the last character input or last selection made on the screen with another character or button push on the mouse. He can repeat this process and continue the incremental backup process to the basic command state.

The user can delete the last word input, or the field specified to date, or the field specified with another character or button push on the mouse. He can also repeat this process backwards to the basic command state as well.

IMPLEMENTATION

The mechanisms and data bases needed to implement the user interface have been modularized and isolated. This "Frontend" can run on a separate computer, such as a mini-computer close to the user, and communicate with the basic tool information processing routines ("Backend") over a communication network. The Frontend consists of terminal handling capabilities [9], a command language interpreter (2a1), and two data bases, a Grammar representing the language syntax and noise words; and a User Profile indicating how the user wants various parameters set for him, such as his prompt and command recognition modes, Keyboard key translations, etc. The Grammar is generated from a high-level description of the user interface written in a language special for this purpose we call Control Meta Language (2a1,).

Given this particular system organization it is very easy to tailor, subset, or modify the user interface for individuals or groups, or to create interfaces for new tools.

Further all the levels of help information, except the Help Data Base, are derived from the Grammar, which guarantees correctness of these levels of documentation as the system changes and is debugged. Various forms of hard copy documentation, such as command summaries, are also derived from the Grammar representation.

The user interface must implement a man/machine dialog. In this section, we discuss issues from machine to man. The discussion centers around the use of displays, with comments on how the problem is dealt with for typewriters. Let us examine some of the types of information that the user needs in order to keep his bearings.

There are four main areas or dimensions along which the user needs information to help him a) to know where he has been, b) to know where he is, and c) to know where he can go from here. Clearly the command language and user interface must offer provisions to move in these spaces as well as obtain status.

1) Information Space
The user needs to know where he is in his information space, and what view or portrayal of the many possible is being displayed to him. Generally he arrived at his present position from previous points and he may want to be able to backtrack to previous points or views as well as to move on.

2) Subsystem or Tool Space

In workshops containing many tools and commands, the user needs to know which tool is active and possibly needs to know which ones he was in previously and their order, and which ones he can enter from here.

3) Command Syntax Space
During the specifications of a command, the user may need to know what he can or is expected to do next and how to back up to a previous point.

4) Information Input Space
During input of information, drawings, tex, etc., the user needs to have ways to see and possibly modify, in simple ways, information that he is entering.

REFERENCES

[1] (8A4) (8A5) (8E1B) (8E1C) Douglas C. Engelbart. A Research Center for Augmenting Human Intellect. Augmentation Research Center, Stanford Research Institute, Menlo Park, California 94025. 68. (3954,)

[2] (8A4) (8D3E) William E. English. Display-Selection Technique for Text Manipulation. Augmentation Research Center, Stanford Research Institute, Menlo Park, California 94025. MAR-67. (9694,)

[3] (8A4) (8D3E) Douglas C. Engelbart. Design Considerations for Knowledge Workshop Terminals. Augmentation Research Center, Stanford Research Institute, Menlo Park, California 94025. 14-MAR-73. (14851,)

[4] (8A5) (8E1C) Douglas C. Engelbart, Richard W. Watson, James C. Norton. The Augmented Knowledge Workshop. Augmentation Research Center, Stanford Research Institute, Menlo Park, California 94025. 1-MAR-73. (14724,)

[5] (8B1C2) James E. (Jim) White. Version 2 of the Procedure Call Protocol (PCP). Augmentation Research Center, Stanford Research Institute, Menlo Park, California 94025. PCP-COVER.NLS;5,. (24590,)

[6] (8B1C2) Jonathan B. Postel and James E. (Jim) White. Notes on a Distributed Programming System. Augmentation Research Center, Stanford Research Institute, Menlo Park, California 94025. 21-MAR-75. (25613,)

[7] (8B1G1) Lawrence G. Roberts and Barry D. Wessler. (University of Utah, Computer Science Department). The ARPA Network. Advanced Research Projects Agency, Information Processing Techniques, Washington, D.C. MAY-71. (7750,)

[8] (8D10B) No Author. L10 Users' Guide: Content Analyzer. Augmentation Research Center, Stanford Research Institute, Menlo Park, California 94025. L10.NLS;7,. (24426,)

[9] (8E1) (8E1H1) Charles H. Irby. Display Techniques for Interactive Text Manipulation. Augmentation Research Center, Stanford Research Institute, Menlo Park, California 94025. 15-NOV-73. (20183,)

[10] (8E1E7) M. C. Gringnetti et. al. An Intelligent Online Assistant and Tutor--NLS Scholar. AFIPS Conference Proceedings, Vol. 44. Anaheim, California. MAY-75. (25054,)

(J26461) 12-SEP-75 19:32;;;; Title: Author(s): Susan Gail Roetter/SGR; Sub-Collections: SRI-ARC; Clerk: SGR; Origin: < ROETTER, EDITSAMPLE.NLS;2, >, 22-AUG-75 17:35 SGR ;;;; .d=on; .H2font=12p,6,Light .Hjp=Odr;; ####;

IRBY MESSAGES

9-SEP-75 22:27:04-PDT,960;000000000001
Date: 9 SEP 1975 2227-PDT
From: IRBY
Subject: MCA notes
To: watson at BBNB, norton at OFFICE-1, engelbart at OFFICE-1, To:
postel, retz 1

Some notes on what has happened so far: 2

1) Bob Millstein now president of MCA 2a

2) Bob taking devil's advocate position on CLI running in 11. Does not buy uniformity, economicsrguements says uniformitonly acheiveif all tools use cml. Thinks several tenex's and other machines will be on-site at gunter. Thinks thereshould be two flavors of fe: big 11 running cli with disk for 40 users witfancy terminals and 11/20 with 64k corrunning modified telnet with fe on another host. 2b

3) Carlson wants nsw complete mid 77 (maybe end 77) with first system delivered with some hacks and MSG protocol by next may. 2c

4) CArlson suggesting wbm be implemenped on plurabus! Bob favors FarBer's DCS system. 55) Bob said Carlson nearly tebmited ARC NSW involvemeNt but Bob talked him out of itbecauce ofCLI work.
------- 2d

11-SEP-75 13:26:30-PDT,1411;000000000001
Date: 11 SEP 1975 1326-PDT
From: IRBY
Subject: NLS with NSW file system
To: postel, michael, watson
cc: maynard, garlick, BELLEVILLE 3

Regina Bolduc expects to have the WM procedures OPEN, DELIVER, SETSEMAPHORE, UNSETSEMAPHORE, and READSEMAPHORE callable by the end of next week. Rick Schantz will be interfacing the encapsulator to it for two weeks after that. It is not until he is finished with that that he will start work on FORMAN. Thus, we agreed to first interface NLS to WM directly and convert to FORMAN sometime later. NLS will, in this model, always deliver and open and will not maintain an LND. This will cause no noticeable delays since we are all on one machine and is not anymore insecure than using the FORMAN since a system crash will take everything down. 4

Thus, work should begin on this as soon as people can get free from other tasks. I understand that debugging of nls commands is nearing completion for the obvious cases. I told Bob Millstein that we would try to get this up in the next 2 to 4 weeks and that we might even beat Schantz to the punch. It seems politically very important to get the NLS editor fully integrated ASAP. I will help to whatever extent is necessary. 5

Also, it looks like we may have won the CLI-on-an-11-FE argument and it will also be very important to produce a working FE-11 ASAP. 6

More later. -- Charles.

------- 7

11-SEP-75 14:07:20-PDT,1434;000000000001
Date: 11 SEP 1975 1407-PDT
From: IRBY
Subject: Millstein advice re carlson visit
To: watson at BBNB, postel 8

I spoke with bob to see if he could shed some light on where carlson is at and whether bob new if bill was upset with us and why. bob could shed little light. he felt bill was bright but a little strange. He also felt that since bill was very results oriented it would be best to concentrate on all the things we are/will get done for NSW than to offer excuses to bill re past events. He also felt that Dick had a tendency to be defensive and that this would not be good with bill. 9

Bob felt FE-11 would go forward although operating system issue is still very open. He feels we should think hard about no OS at all and see what problems we encounter. He half expects us to stick with ELF but assures me that SRI will have to maintain it until a replacement for ADR is found (ADR has been told to stop ELF maintenance). Is our applications group interested in ELF maintenance and setting up an NCC for FE's?? ADR's bid was 180K/yr. Bob thinks too high, carlson thinks too low. Seems high to me but I don't know what it included. 10

Everyone here seems to be all for NLS. The sooner we can get it into NSW the better. 11

Millstein's big fear is that system to be delivered in march-may may be too good and cause ARPA to consider it done and give no more funds. 12

-- Charles.

------- 13

IRBY MESSAGES

(J26462) 12-SEP-75 17:33;;;; Title: Author(s): Jonathan B. Postel/JBP; Distribution: /JBP([INFO-ONLY]) ; Sub-Collections: SRI-ARC; Clerk: JBP;

26462 Distribution
    Jonathan B, Postel,

Dave Smith (DAV) - new DEV/NPG/NSW warm body

I am new here at ARC and am trying to learn my way around the place. I have met many of you ARC people (ARCians?) but not all. Please feel free to drop by and chat (Joe Ehardt's old office) at any time, especially if I don't run into you in the next week or so. 1

I will be working in the immediate future on the HELP system, taking over from Kirk Kelley (1/2 of whom is going away). There is pretty general agreement that HELP needs help. So while my first task will be to bring it up under NLS-9 and debug all the existing versions, the long-range goal is to design and implement what we really want. All suggestions, thoughts, criticisms, comments on the current system, bugs, etc. are most welcome from everyone -- particularly if there is some capability you have long wanted to have. (Except that suggestions to name the new system BEYONDHELP will be filed in the wastebasket.) 2

P.S. There is, believe it or not, a HELP group. Send me a note if you would like to be added to it. 3

Dave Smith (DAV) - new DEV/NPG/NSW warm body

(J26463) 12-SEP-75 21:13;;;; Title: Author(s): David C, Smith/DAV; Distribution: /SRI-ARC([ACTION]) ; Sub-Collections: SRI-ARC; Clerk: DAV;

26463 Distribution

Douglas C. Engelbart, Martin E. Hardy, J. D. Hopper, Charles H. Irby, Harvey G. Lehtman, James C. Norton, Jeffrey C. Peters, Dirk H. Van Nouhuys, Kenneth E. (Ken) Victor, Richard W. Watson, Don I. Andrews, David C. Smith, Mary Ann Kellan, Buddie J. Pine, Andy Poggio, David L. Retz, Laura J. Metzger, Karolyn J. Martin, Jan A. Cornish, Larry L. Garlick, Priscilla A. Wold, Pamela K. Allen, Delorse M. Brooks, Beverly Boli, Rita Hysmith, Log Augmentation, Joseph L. Ehardt, Raymond R. Panko, Susan Gail Roetter, Robert Louis Belleville, Rene C. Ochoa, Ann Weinberg, Adrian C. McGinnis, Robert S. Ratner, David S. Maynard, Robert N. Lieberman, Sandy L. Johnson, James H. Bair, Jeanne M. Leavitt, Rodney A. Bondurant, Jeanne M. Beck, Marcia L. Keeney, Elizabeth K. Michael, Jonathan B. Postel, Elizabeth J. Feinler, Kirk E. Kelley, N. Dean Meyer, James E. (Jim) White