

On Inexpensive Displays

Don ... With help from DCW and KEV, I've prepared a draft Request for Proposal to be sent to manufacturers of low- and medium- cost display terminals which I would like you to review and comment on. It's in my file, (KUDLICK, TERMINALS, 1:wy).

1

On Inexpensive Displays

(J10796) 20-JUN-72 10:47; Title: Author(s): Michael D. Kudlick/MDK;
Distribution: Don I. Andrews/DIA; Sub-Collections: SRI-ARC; Clerk: MDK;

report from SDIS planning team.

INTRODUCTION

1

The SDIS planning team has reviewed the most recent annual report to the Office of Naval Research, and come to several conclusions regarding future work on the ONR Project. This document describes these conclusions. It includes a discussion of the meaning of "intelligence" systems as we view them, and a proposed set of subprojects on which work can and should begin now. It also includes an estimate of manpower requirements and recommended personnel assignments. We consider it a working document, to be revised, upgraded, etc., as our ideas grow.

1a

The basic aspects of an "intelligence system" which we consider to be of paramount importance are:

1b

(1) the classes of data that are placed in the system, along with the processes of acquisition, classification, and cross-referencing;

1b1

(2) the methods of making vast amounts of information available to a searcher using the system;

1b2

(3) the data structures, search techniques, and query languages available to the user, and the relevance of these techniques to his personal work habits, vocabulary, and need for flexibility.

1b3

The short-term tasks by which we can begin to directly address these aspects and show tangible results are:

1c

(1) review state of the art information structures;

1c1

(2) experiment with microfilm data-bases and displays;

1c2

(3) experiment with prototype query languages, interfaced to NLS;

1c3

(4) accumulate "intelligence information" about existing systems that would serve the dual role of providing data for an experimental data base and providing information for ARG.

1c4

The manpower requirements for these tasks are estimated to be:

1d

MDK 0.2 MY
JBN 0.5 MY
JFV 0.5 MY.

1d1

report from SDIS planning team.

INITIAL IDEAS ON SDIS RESEARCH

2

What follows is a summary by JFV of discussions among JFV, MDK, and JBN regarding possible developments beyond the research reported to ONR in the April 1972 Annual Report, "Experimental Development of a small Computer-Augmented Information System". The main concept proposed here is to emphasize the orientation of an "Intelligence System" supporting a discipline (such as systems development) in a general sense. It is shown that this implies expanding our conceptualization and activities in three specific directions, namely the definition of a wider class of inputs, the recognition of new user interaction modes, and an increased awareness of trends in information-seeking behavior among programmers and other system builders. Specific recommendations are made in all three areas.

2a

1. CLASS OF DATA

3

Our last report states that "We have continued the building of a reference data-base. Procedures have been developed for improved citation form and citation file building, and many documents have been recorded with these procedures." It would be logical to continue this development with the definition of new classes of data that would be more oriented towards the augmentation of a software development team and the support of individual system builders. In the present state of our thinking this might include: trip reports, summaries of conversations and meetings, notes from conferences, etc. as well as the actual citations of published material. Such material is already present in the Journal system, and it would be valuable to experiment with limited cross-linking facilities between such inputs and our catalogued documents.

3a

On the basis of these limited experiments, classes of relevant documents would then be defined, together with acquisition, classification and possibly encoding procedures.

3b

2. USER INTERACTION

4

It is not clear that the user of an 'intelligence' system meets the criteria of the typical information system user, who is generally viewed as a passive recipient of textual data. Here the user must be able to make full use of the augmentation techniques at his disposal, not only to browse through files (as he could do under any number of already existing, classical systems) but to refine his search, obtain different portrayals of data relationships, and reach decisions through use of the machine.

4a

report from SDIS planning team.

One element of such a process is obviously a hardware component capable of storing vast amounts of information at a fraction of the cost of on-line storage: In this respect we recommend that an aggressive program of development and utilization of a microfilm data-base be undertaken. A first practical objective of the intelligence system might be to support an on-going program of monitoring the hardware components available on the market for coupling software search with microfilm display. This might be the occasion of creating a separate file within our expanding data-base.

4b

3. SUPPORTING SYSTEM BUILDERS

5

System builders are highly specialized, and in that respect they are not typical of the general user population. On the other hand, they are closely related to a category of individuals who need the best support in terms of timely, accurate data.

5a

Similar situations have been recognized in the creation and development of data-bases intended for use by foreign policy specialists or in the case of small communities that have developed a great deal of sophistication in the vocabulary they use to describe their own data. Support of such a community through a generalized set of keywords and indices is impossible. Current thinking seems to indicate, on the contrary, that it is important to isolate the major subsets of the problem area and automate the data base with a view to supporting very well key individuals in that field. If this is accomplished under a high-level structure with sufficient commonality, the end result is a collection of documents that remains versatile and uncluttered with noisy data. Such a collection, naturally, can become quite large while escaping the problems of duplication and redundancy that are encountered in our own catalogue, intended to serve a wide, unrestricted public.

5b

On the basis of these remarks, it is suggested that our team look specifically into the problem of augmenting selected, highly-specialized individuals, while developing techniques for monitoring their activity in terms of performance and retrieval effectiveness. At the same time, relationships between this "personalized" intelligence system and the more general tools we are building should be identified and supported.

5c

A technical consequence of the above will appear in terms of file structures available for the building of such data-bases. Our previous report to ONR has mentioned the "Design of data

report from SDIS planning team.

elements" as an important area of research: "Data elements should be adaptable to economical use by programs...", "Data elements should be adequate to describe all species of information items which are anticipated to be added to the RINS collection." When viewed in connection with our remarks about "Classes of data", this means that it will be necessary for the file system to handle not only text items but relationships, graphs, tables, etc. in the context of non-classical information supports. It will be necessary, then, for our team to closely follow and participate in the development of the new file system, which is intended to include such facilities. We might volunteer to test it out with complex data representations from real-life cases.

5d

4. SHORT-TERM OBJECTIVES

6

In the near future it is necessary for us to initiate several well-defined projects and bring them to completion. As a first objective, we should plan a review of existing information structures (not limited to NLS) to reach a clear definition of what we might call our future "data elements". It is necessary to do this rapidly to keep pace with and take best advantage of the file system implementation progress.

6a

A second thrust would be devoted to experimentation with microfilm and coupling (manually) between the NLS search and microform display. The result of this project might be recommendation for specific hardware to be acquired or built in future phases of development.

6b

A third project that we can see beginning immediately would be aimed at experimenting with prototype query languages, not immediately intended for release to the ARC community or to a wider audience through the NIC, but aimed at gaining experience with an NLS/query language interface. Here we could work closely with the NIC team, the ASIS conference being a target for implementation of a demonstration package.

6c

A fourth, medium-range project, would correspond with our team's taking an information role within ARC and accumulating data about existing systems that might have useful features or interesting parameters that NLS system builders would like to know about. Typically such information must be published within a sufficiently small interval to have an impact on our own work (if it is published at all). In view of our limited resources it is suggested to limit such visits to the geographic neighborhood of ARC, where a list of potential sites would include the IBM research center, various service bureaus, local administrations (Santa Clara has a pilot county

report from SDIS planning team.

automation system), universities ,and so on. A small number of site visits are proposed to experiment with various ways in which we could record the information thus gained 'in the field' in the most flexible fashion.

6d

5. LONG-TERM OBJECTIVES

7

Our long-term objectives will remain consistent with i) the development of ARC as an information center, through the expansion of the catalogue system and its improvement and ii) the exportation of the NLS service through the NIC.

7a

Hopefully we will reach a point where NLS will contain both a flexible file system capable of handling "intelligence" data and a generalized query language for accessing it. In our next report, however, we foresee points one, two and three above to be our main contributions in terms of research results.

7b

<JJOURNAL>10806.NLS;1, 21-JUN-72 12:53 XXX ; Title: Author(s):
Michael D. Kudlick/MDK; Sub-Collections: PERC; Clerk: JFV;
Origin: <KUDLICK>SDIS.NLS;3, 20-JUN-72 14:39 MDK ;

(J10808) 21-JUN-72 14:51; Title: Author(s): Thomas F. Knight/TFK;
Distribution: James E. White/JEW; Sub-Collections: NIC; Clerk: TFK;

JIM:

I WOULD LIKE TO IMPLEMENT A MORE REASONABLE HUMAN INTERFACE TO THE NLS SYSTEM HERE AT THE AI LAB. I UNDERSTAND THAT PETER DEUTSCH AT XEROX HAS SOME SORT OF INLAC PROGRAM THAT DOES A REASONABLE JOB. PERHAPS SOME MODIFICATION OF THAT SYSTEM COULD WORK, OR PERHAPS SOME OTHER WAY OF GETTING GRAPHIC STYLE OUTPUT FROM NLS COULD BE USED. I AM RATHER UNFAMILIAR WITH THE GUTS OF NLS BUT WOULD BE WILLING TO LEARN. IT SEEMS THAT ALL THAT WOULD BE NEEDED IS SOME SORT OF LOW LEVEL CHARACTER GRAPHICS LANGUAGE, WHICH PROBABLY ALREADY EXISTS INTERNALLY TO NLS. DO YOU RUN TERMINALS LIKE ARDS ETC? PERHAPS WE COULD INTERPRET THEIR CODE.

1

MEMO TO DHC

MEMO TO DHC

1

THE NBS CCST-TIP WILL BE DOWN FROM 8:30 A.M. TO 12:00 NOON
(E.S.T.) ON THE FIRST TUESDAY OF EACH MONTH.

1a

SWW 22-JUN-72 6:50 10809

MEMO TO DHC

(J10809) 22-JUN-72 6:50; Title: Author(s): Shirley W. Watkins/SWW;
Distribution: David H. Crocker/DHC; Sub-Collections: NIC; Clerk: SWW;
Origin: <NBS-TIP>SWW.NLS;9, 22-JUN-72 6:47 SWW ;

sample journal entry 22 june 1972

(author) Journal documents authored

1

GG 22-JUN-72 6:56 10810

sample journal entry 22 june 1972

(J10810) 22-JUN-72 6:56; Title: Author(s): Glen Grazier/GG;
Distribution: James H Shiffirin/JHS2; Sub-Collections: NIC; Clerk: GG;

graphics information for graphics notebook

reference nwgRfc 351 nic 10593, usaf-etac has no graphic capabilities. we are a tip oriented user only.

1

JHS2 22-JUN-72 7:11 10811

graphics information for graphics notebook

(J10811) 22-JUN-72 7:11; Title: Author(s): James H Shiffrin/JHS2;
Distribution: David H. Crocker/DHC; Sub-Collections: NIC; Clerk: JHS2;

Verify Arnie's ID

Please verify the ident ANO. Thanks. Stan Cohen

SC 22-JUN-72 11:02 10812

Verify Arnie's ID

(J10812) 22-JUN-72 11:02; Title: Author(s): Stanley Cohen/SC;
Distribution: Barbara E. Row/BER; Sub-Collections: NIC; Clerk: SC;

Conversation with Bob Kahn about ICCC

A number of important questions about the ICCC meeting in Oct. were raised at this mornings PERC meeting. I called Bob Kahn and had a good chat with him about these. The points discussed and his answers were the following.

1

What is the physical environment going to be like?

1a

Bob has arranged for the Georgetown Ballroom of the Hilton Hotel; it has 3000 square feet probably about 75x40 (exact dimensions are not known to Bob at this time).

1a1

A carpeted false floor is to be installed and there will be a partition for an area to show a movie on resource sharing which will run more or less continuously.

1a2

There will not be booths in the normal sense but details are not yet worked out. Bob expects to have tables arranged around the room holding 1 or 2 terminals and low 4 ft. partitions to break up the space.

1a3

Who will man the booths?

1b

Bob presently expects many terminals to be unmanned, although maufacturers could have a sales rep.

1b1

There will probably be people from net sites at some of them from time to time to demonstrate their programs.

1b2

We discussed the need for lots of experts from the network around who can answer questions and help people. Bob is in favor of such a thing but has not had time yet to pull such a group together and there is some uncertainty if enough people will volunteer, his initial approach to people yielded lots of people interested but with other things to do.

1b3

When will the equipment be on the floor and available to try there?

1c

The conference is tues-thurs oct 24-26 and at the moment the room is only going to be available from the 22nd. This seems to be a short time to install floor, TIP, terminals, phone lines, etc and Bob is going to try to get the room the previous Wed or Fri, but doesn't know yet.

1c1

Bob would like to have a working demo with terminals, resources at BEN in Aug.

1d

What resources are presently committed?

1e

Conversation with Bob Kahn about ICCG

UCLA Klinerock will have some network measurement stuff to demonstrate. 1e1

Stanford AI has indicated they will have some programs to demone. 1e2

Utah Tony Hearn will have his Reduce program to demo, Berry Wessler plans to domo something, Harvey Greenfield has some sort of mdical program. 1e3

RAND before recent sakeup indicated iterest, satus now not known. 1e4

SDC will have some natural language programs and their data management systems. 1e5

MIT Vezza and Kahn are looking into having some hardcopy device such as Photon attached to a GE IS system. Al has some other things to demo. 1e6

Bob has contacted a number of non net sources such as the NY Times (data base), Eastern Airlines etc., but has no definite commitmets yet. 1e7

This ARPANET demo is the only show at the conference, Larry Roberts has invited the ASIS people over for Thurs afternoon. 1f

The TIP has at this time never been tested with 64 terminals, but they are going to be puttin as many terminal on as possible during July and testing the terminal to be supplied ford the conference. 1g

Starting about AUg 1 he expects to start testing to see what terminals work with what resources and what glitches exist. 1h

I asked Bob if we could have enough space for a ARC NIC booth with an IMLAC TI and room for people to watch our demo'S. Bob saw no problem there; IMLAC has committed 2 terminals and we can have one. TI has not indicated they plan to loan terminal for the show. Hiram French is the man from IMLAC who is supposed to call us. If there are more thn 64 terminal or the TIP can't support all, then they will have to schedule terminal connections to the TIP. 1i

Bob is not ssre who will be at the conference, its theme is networks and their impact on socety, implications for regulation etc so ther should be a broad spectrum of people; he will send me a program. 1j

Conversation with Bob Kahn about ICCC

We talked about what kind of documentation help, signs etc will be around to help people connect and use resources and generally understand wha the deomo is allabout.

1k

Bob's present picture is some Large 2x3 signss in the lobby outside the main lecture room directing people to the demo and then some signs inside explaining whats happening, a card with some TIP commands,resources, a brochure with actual scenerios of resource usage, the movie and some people as the main on te floor sources of help.

1k1

We talked about the need to coordinate all this with NIC documentation and our desire to help design all this. Bob and I are tentatively planning to get together the weekend of July 7 here to plan this. We need to have thoughtthrough ahead of time what we think should be done.

1k2

ARPA except for Bob does not seem to have set up any budget for this show except what their contractors are essentially willing to give as part of their ongoing research, therefore he has a very difficult job of pulling off such a large show without more help than he seems to have gotten so far (these are my conclusions not satements of his).

2

RWW 22-JUN-72 11:01 10813

Conversation with Bob Kahn about ICCC

(J10813) 22-JUN-72 11:01; Title: Author(s): Richard W. Watson/RWW;
Distribution: Douglas C. Engelbart, Walt Bass, Charles H. Irby, Michael
D. Kudlick, James C. Norton, William H. Paxton, Paul Rech, Richard W.
Watson, Douglas C. Engelbart, Robert E. Kahn, Jacques F. Vallee, James
E. White, Jeanne B. North/PERC DCE REK2 JFV JEW JBN; Sub-Collections:
SRI-ARC PERC; Clerk: RWW;
Origin: <WATSON>ICCC.NLS;3, 22-JUN-72 10:58 RWW ;

TNLS BEGINNERS GUIDE -- PRELIMINARY

1

Augmentation Research Center
Stanford Research Institute
Menlo Park, California 94025

2

NOTE

3

Corrections, updates, or questions concerning the material contained
in this document should be directed to: 3a

Marilyn Auerbach
Augmentation Research Center
Stanford Research Institute
333 Ravenswood Avenue, J2080
Menlo Park, Ca. 94025
(415) 326-6200 ext. 3722

3a1

or

3a2

Dirk Van Nouhuys
Augmentation Research Center
Stanford Research Institute
333 Ravenswood Avenue, J2078
Menlo Park, Ca. 94025
(415) 326-6200 ext. 3370

3a3

PREFACE

4

This document is a beginner's guide to TNLS. It contains a subset of the TNLS (and TENEX) command vocabulary sufficient to enable the inexperienced user to produce and edit text. Many of the features of TNLS are either simplified or entirely omitted in order to present readers of this document with a working ability in TNLS with minimal detail. The object of this document is to enable the user to access the TNLS system, create and edit TNLS text (files), save and delete files, and use the Journal system for the purposes of sending and receiving information to other users. 4a

After the user has attained expertise with the concepts and features described in this document, he is referred to the NIC TNLS User Guide (7470,) and the NIC Journal System User Guide (7635,). 4b

CONTENTS

SECTION -----	PAGE -----
	5
PREFACE	5a 4 5b
SYNTAX AND DOCUMENTATION CONVENTIONS 8	5c
HOW TO GET IN AND OUT OF THE SYSTEM	10
INTRODUCTION	5d 10
ACCESSING THE TENEX SYSTEM - LOGIN	5d1 10
FULLDUPLEX/HALFDUPLEX COMMANDS FOR NETWORK USERS	5d2 11
ACCESSING TNLS - TNLS	5d3 12
INTERRUPTING TNLS/TENEX - CONTROL C	5d4 12
SYSTEM QUERY - CONTROL T	5d5 13
RESUMING WORK IN TNLS - CONTINUE	5d6 13
LEAVING TNLS - EXECUTE QUIT	5d7 14
LEAVING THE TENEX SYSTEM - LOGOUT COMMAND	5d8 14
WHAT A FILE IS AND HOW TO WORK WITH IT	5d9 15
INTRODUCTION	5e 15
STATEMENT NUMBERS	5e1 15
ENTERING TEXT IN TNLS - INSERT COMMAND	5e2 16
EDITING WHILE ENTERING TEXT	5e3 18
FILE NAMES	5e4 19
TNLS FILE COMMANDS	5e5 20 5e6

TENEX FILE COMMANDS

22
Se7

<u>SECTION</u>	<u>PAGE</u>
HOW TO TELL WHERE YOU ARE	5f
INTRODUCTION - CONTROL MARKER	23
COMMANDS	5g
MOVING AROUND	23
INTRODUCTION	5g1
ADDRESS ELEMENTS	23
JUMPING	5g2
PRINTING ALL OR PART OF THE FILE	25
VIEWSPECS	5h
PRINT CURRENT STATEMENT	25
PRINT STATEMENT NEXT TO CM	5h1
PRINT STATEMENT BACK FROM CM	25
OUTPUT DEVICE TELETYPE - PRINTING THE FILE	5h2
HALT PRINTING	26
	5h3
	28
	5i
	28
	5i1
	29
	5i2
	30
	5i3
	30
	5i4
	30
	5i5
	30
	5i6

SECTION -----	PAGE -----
EDITING	5j 5k
INTRODUCTION	31 5k1
EDITING BY STATEMENT	31 5k2
EDITING BY CONTENT	36 5k3
EDITING ACROSS THE FILE - SUBSTITUTE BRANCH	43 5k4
USING THE SYSTEM TO SEND/RECEIVE MESSAGES TO/FROM OTHER USERS	44 5l
INTRODUCTION	44 5l1
SUBMITTING ITEMS TO THE JOURNAL	44 5l2
JOURNAL COMMANDS	45 5l3
SAMPLE JOURNAL SESSION	49 5l4
SIMPLE WAY TO READ ITEMS	51 5l5
Section 8. COMMAND SUMMARY AND HELP	53 5m
ONLINE USER HELP	53 5m1
COMMAND SUMMARY	53 5m2

SYNTAX AND DOCUMENTATION CONVENTIONS

6

The following conventions are used in the TNLS command specifications throughout this manual. 6a

NAMED CHARACTERS 6b

Special characters such as Command Accept, Command Delete, Carriage Return, space, etc. are referred to by names (CA, CD, CR, SP, etc.) in uppercase letters. 6b1

COMMANDS 6c

Commands are shown in lower case. Most TNLS commands require that only the first character of each command word be typed. TENEX commands require at least three characters for recognition. 6c1

PARAMETERS 6d

Values to be supplied by the user are shown in uppercase. The names of these parameters will not cause confusion with the uppercase named characters. 6d1

INPUT TO TNLS 6e

All but a few of the TNLS commands documented here require only a single character in order for the system to recognize each command word. When the system recognizes a command word, it echoes the remainder of the word if the user's terminal is in full-duplex mode. See below for system output. 6e1

INPUT TO TENEX 6f

Although the entire command word is shown for each command, the first few characters are usually sufficient for recognition by the system. 6f1

SYSTEM OUTPUT

6g

Text output by the system as a command is entered is shown in lower case letters enclosed in square brackets ([]). Some commands cause the system to generate a new line as the command is entered; this is shown as "[CR]". Brackets are also used to clarify the command, e.g. the command Insert Statement requires only that the user types "is". However, this is shown as "i[nsert]s[tatement]" in the syntax representation for this command. even though over the network, some sites do not receive these characters.

6g1

QUANTITY

6h

In cases where any number of entities might be supplied by the user, the entity is preceded by the dollar sign character (\$).

6h1

OPTIONS

6i

Many TNLS commands operate on a variety of entities. These choices are shown in a vertical column. The general syntax of the command applies to all choices except where specified otherwise. For example, the syntax for the delete command shown as:

6i1

```
d[elate] s[tatement] ADDR      CA CA  
w[ord]  
c[haracter]  
t[ext] ADDR CA ADDR
```

6i1a

The general syntax for all forms of the delete command is the same except that "text" requires two addresses to be specified to delimit the beginning and end of the text string.

6i2

CONTROL CHARACTERS

6j

Control characters are indicated by the appropriate letter preceded by an up-arrow" (↑).

6j1

Section 1. HOW TO GET IN AND OUT OF THE SYSTEM

7

INTRODUCTION

7a

TENEX is an interactive timesharing system produced by Bolt Beranek & Newman for the DEC PDP-10. TNLS users must use certain facilities of the TENEX system through the system's TENEX language. The TENEX facilities of primary interest to the TNLS user are those which access to the TNLS subsystem itself and the file system. 7a1

ACCESSING THE TENEX SYSTEM - LOGIN

7b

The following procedure is necessary to access TENEX which in turn provides the user with access to the TNLS system. 7b1

When the terminal is connected to the ARC PDP-10 computer via the Network, the TENEX system will print the message: 7b2

```
ARC-TENEX .... date ARC EXEC .. 7b2a
```

(If the user is not connected through the Network, he must type control C (fc) to access TENEX.) 7b3

TENEX responds that it is ready to accept information by typing the character '@'. Before the user can perform any tasks on the system he must first identify himself using the LOGIN command: 7b4

```
log CR [CR]  
[(user)] USERNAME CR [CR]  
[(password)] PASSWORD CR [CR]  
[(account #)] ACCOUNT NO. CR [CR] 7b4a
```

where: 7b5

USERNAME 7b5a

1-39 alphanumeric characters (excluding the characters ; and .) Users who wish to use the system experimentally may use the name "guest" (see Password "arc" below). 7b5a1

PASSWORD 7b5b

1-39 alphanumeric characters (excluding the characters ; and .) which are not echoed by the system. Users whose log in to the system with the name should use the password "arc". 7b5b1

ACCOUNT NO. 7b5c

1-39 characters; (#3 is currently used for all users on the
TNLS system) 7b5c1

The Backspace character Control A (fa) may be used at any time to
correct typing errors when entering characters. 7b6

NOTE: For Network users, username, password and, account number
may be obtained through the site liaison. 7b7

When the user has successfully logged into the system the
following message is printed: 7b8

JOB ... ON TTY ... date time 7b8a

If the system load is already at or beyond the prescribed capacity
for local and/or Network users, the system will refuse to log in a
user. 7b9

If a user fails to successfully login within two minutes, the
system automatically prints the message: 7b10

AUTOLOGOUT
KILLED JOB ..., TTY .., AT date time
USER time1 IN time2 7b10a

where: 7b11

time1 = total computer time used
time2 = total terminal time used 7b11a

FULLDUPLEX/HALFDUPLEX COMMANDS FOR NETWORK USERS 7c

These commands control how TENEX at ARC sees local terminals over
the Network. 7c1

fullduplex CR 7c1a

characters entered by the user are transmitted over the
Network and then echoed by TENEX 7c1a1

halfduplex CR 7c1b

TENEX assumes that characters entered by the user are to be
echoed by TENEX but by the TELNET program at the local site
or the local user's terminal. 7c1b1

The default value for Network users is local echoing - halfduplex.

ACCESSING TNLS - NLS

7c2

7d

In order for the user to enter TNLS, he must use the TENEX command NLS. 7d1

```
@nls CR [CR]
[id:] IDENT CR [CR]
[device:] DEVICE
```

7d1a

DEVICE =

7d2

"n" (n[et-tty]) if the user's system echoes locally (half duplex)

7d2a

"37" if the user's system does not have local echoing or if the user would prefer echoing and full command feedback from TENEX (full duplex) and the user's terminal has both upper and lower case characters

7d2b

"33" if the user's system does not have local echoing (full duplex) and the user's terminal has only upper characters (Note:TNLS treats uppercase characters from this device as lowercase.)

7d2c

A Network user may establish his IDENT by contacting the NIC technical liaison at his site.

7d3

An asterisk (*) in the margin is TNLS's signal that it is awaiting a command. The asterisk is printed whenever TNLS completes a command -- in other words, if the asterisk is not printed, TNLS is not yet ready to process another command.

7d4

INTERRUPTING TNLS/TENEX - CONTROL C

7e

The user may interrupt any operation in TNLS or TENEX by using the interrupt signal, Control C (↑C). When this key is pressed the user is immediately returned to the TENEX system. When work in TNLS is interrupted in this manner, the user may perform any of the operations described as TENEX functions in this document without affecting his work in TNLS. (That is except for Logging out of the system or deleting the file currently being worked on in TNLS.)

7e1

To resume work in TNLS after a ^C interrupt, the user must use the command, Continue, described below.

7e2

SYSTEM QUERY - CONTROL T

7f

At any time working in TENEX or TNLS the user may interrogate as to whether the system is running or waiting for input by using the Control character "t" (^t). This control character is frequently used when it is not apparent that the system is running. It causes the system to immediately issue one of several possible messages and then return the user to whatever his state when the control character was issued. Possible messages include:.

7g

RUNNING AT location USED time1 IN time2 7g1

IO WAIT AT location USED time1 IN time2 7g2

^C FROM RUNNING AT location USED time1 IN time2 7g3

location = 7g4

the internal location of the user's job 7g4a

time1= 7g5

CPU time 7g5a

time2 = 7g6

terminal time since the user logged in 7g6a

RESUMING WORK IN TNLS - CONTINUE

7h

If the user has used the control C (^c) character to leave the TNLS subsystem, the Continue command enables the user to reenter the TNLS subsystem.

7h1

continue CR 7h1a

When returned to the TNLS subsystem, the user will find himself at precisely the point in his work at which he left TNLS with a Control C, TNLS will not initially respond with its prompt character "*". However, it will accept commands as long as there was no output operation in progress when the user left the subsystem.

7h2

LEAVING TNLS - EXECUTE QUIT

7i

After the user has completed work in TNLS he may leave the system by using the Execute Quit command which returns him to the TENEX level. 7i1

e[xecute] q[uit] CA

7i1a

LEAVING THE TENEX SYSTEM - LOGOUT COMMAND

7j

The LOGOUT command enables the user to leave the system and causes certain accounting information to be printed at the terminal. 7j1

logout CR

7j1a

When this command is executed the system prints the message: 7j2

KILLED JOBxx,USER username, ACCT account no.,TTY yy, AT date
time
USED time1 IN time2

7j2a

where:

7j3

time1 = total computer time used
time2 = total terminal time used

7j3a

Section 2. WHAT A FILE IS AND HOW TO WORK WITH IT

8

INTRODUCTION

8a

NOTE: The following description of files and file structure was written specially for this document. As such, it is a simplified overview of TNLS files which is suitable for the purposes of this document but not in conflict with the facts of TNLS file generality.

8a1

When working in TNLS, one is at all times constructing, studying, or modifying a file.

8a2

The system automatically creates a file for each TNLS user the first time that user accesses TNLS. This file is referred to as the user's initial file, as its name consists of the user's IDENT (usually his initials).

8a3

Every TNLS file is made up of STATEMENTS, entities which may contain any sort of text (every paragraph and heading in this document is a statement).

8a4

Every TNLS file has an ORIGIN STATEMENT or "zero statement". (The origin statement has been omitted from the printout of this document). This statement may be referenced in TNLS commands as ".0".

8a4a

Statements consist of characters that are entered by the user usually with the Insert command. TNLS automatically supplies characters to segment a TNLS statement into "lines" but the user may override the TNLS convention by inserting Carriage Return/Line Feed characters himself.

8a5

STATEMENT NUMBERS

8b

Every statement has a unique "statement number". The statement number (and its following space) is NOT part of the text of the statement; it is associated with the position of the statement in the file and is subject to change when the file structure is modified by adding, deleting, or moving statements.

8b1

ENTERING TEXT IN TNLS - INSERT COMMAND

8c

Text is created in the TNLS system using the Insert Command to either enter new statements, or to add text to an existing statement. For purposes of clarity, these two applications of the Insert Command are discussed separately. First, the Insert Command as used to create a new statement:

8c1

```
i[nsert] s[tatement at] ADDR CA CA [CR]
LIT CA
      CDOT
```

8c1a

ADDR =

8c2

the location of the statement after which the new statement or series of statements is to be inserted. Valid ADDR specifications are discussed in Section 4 of this document.

8c2a

CA =

8c3

Command Accept. On most devices this is the character Control D (↑d).

8c3a

LIT =

8c4

any series of characters except CA or Centerdot (CDOT) which is the text of the statement to be inserted. The editing characters described below may be used when entering LIT to correct entries. If LIT is omitted, an empty (blank) line is generated.

8c4a

CDOT =

8c5

"center dot" character means continue insert command. This option allows the user to continue inserting statements. When this delimiter is used, the syntax for inserting subsequent statements is the same as though the user had typed the Insert command up to and including the first CA; the system expects the user to enter a CA and LIT.

8c5a

The Center dot character is a Control B (↑b).

8c5b

NOTE: Insert and several other commands require two successive CA's at points in their specification because TNLS allows options which may be specified between the CA's which are not documented here.

8c6

When a new statement is inserted into a file, all statements following the place of insertion are automatically renumbered by the system as necessary. 8c7

The maximum number of characters presently allowed per statement is approximately 2000. Every statement consists of at least one character. 8c8

Examples: 8c9

Using the following the command in a new file: 8c9a

```
i[nsert] s[tatement] CA CA  
you are my sunshine, my only sunshine CA 8c9a1
```

causes the system to insert "you are my ..." into the file as statement 1. 8c9b

```
i[nsert] s[tatement] .1 CA CA  
play it again CA 8c9b1
```

causes the system to insert "play it ..." as statement 2. 8c9c

```
i[nsert] s[tatement] .1 CA CA  
you make me happy when skies are grey CA 8c9c1
```

causes the system the insert the text "you make me..." after the statement number one. 8c9d

At this point the file contains the statements: 8c9e

```
1 you are my sunshine, my only sunshine 8c9e1
```

```
2 you make me happy when skies are grey 8c9e2
```

```
3 play it again 8c9e3
```

Using the CDOT option to enter successive statements: 8c9f

```
i[nsert] s[tatement] .3 CA CA 8c9f1
```

```
you'll never know dear CDOT CA 8c9f2
```

```
how much I love you CA 8c9f3
```

At this point the file contains the statements: 8c9g

1 you are my sunshine, my only sunshine	8c9g1
2 you make me happy when skies are grey	8c9g2
3 play it again	8c9g3
4 you'll never know dear	8c9g4
5 how much I love you	8c9g5

EDITING WHILE ENTERING TEXT 8d

The following characters may be used as follows when entering text: 8d1

BC or ↑A - 8d1a

backspace previous character. One character is deleted each time this key is pressed. 8d1a1

BW or ↑W - 8d1b

backspace previous visible/word. One word and its preceding space is deleted each time this key is pressed. 8d1b1

A visible is any series of printing characters. 8d1b1a

BS or ↑Q - 8d1c

backspace statement. The entire current LIT is deleted when this key is pressed. 8d1c1

↑R - 8d1d

show current LIT. This key causes the system to print the current content of LIT as it is being entered. This is helpful when several of the above editing characters have been used during text entry and the user is unsure of the current content of LIT. 8d1d1

FILE NAMES 8e

The names of files in TENEX/TNLS are of the following form: 8e1

<DIRECTORY>FILENAME.EXTENSION;VERSION # 8e1a

DIRECTORY = 8e2

1-39 alphanumeric characters, excluding control characters, non-printing characters, period (.), and semicolon (;). This element is a TENEX user name and is required only when a user references a file belonging to a directory other than the directory specified when the user logged in to the system. 8e2a

FILENAME = 8e3

1-39 alphanumeric characters, excluding control characters, non-printing characters, period (.), and semicolon (;) 8e3a

EXTENSION = 8e4

1-39 alphanumeric characters, excluding characters control, non-printing characters, period (.), and semicolon (;) 8e4a

VERSION # = 8e5

a numeric value (1 to 131071) 8e5a

The length of the entire filename (including the delimiters . and ;) must not exceed 39 characters. Otherwise, there are no restrictions on the length of any field within the total filename. 8e6

The only required filename field in any TNLS command is the name of the file. Directory name is assumed to be the directory of the current user; extension name is assumed to be "NLS"; and, version number is assumed to be the highest version number for that file in the user's directory. 8e7

TNLS FILE COMMANDS

8f

NULL File

8f1

The Null File command enables the user to create a new file which contains only an origin statement.

8f1a

n[ull] f[ile] FILENAME CA

8f1a1

LOAD FILE

8f2

The load file command causes the file specified to be opened and made available to the user for work in the TNLS subsystem.

8f2a

l[oad] f[ile] FILENAME CA

8f2a1

FILENAME =

8f2b

the name of the file to be opened. If the user enters only the name field of FILENAME, extension NLS and the highest version number, are the default values for the remaining fields. If the file belongs to another user's directory, FILENAME must include the directory name enclosed in anglebrackets.

8f2b1

UPDATE FILE

8f3

The update file command causes the system to merge the contents of the current TNLS file with any modifications made to the file since the last time the file was updated or output.

8f3a

u[pdate] CA

8f3a1

When this command is executed the system will print the name of the file being updated.

8f3b

It is advised that the user issue this command before leaving the system.

8f3c

OUTPUT FILE

8f4

The Output File command is similar to the Update File command except that it causes the system to perform certain "housekeeping" functions on the file specified; it also allows the user to change the name of the file.

8f4a

To the user the effects of the Update and Output command are practically identical: the Update command is generally used for frequent updating and the Output command is used at less frequent intervals. 8f4b

o[utput] f[ile] EMPTY CA
FILENAME 8f4b1

EMPTY = 8f4c

the file created by the Output command will bear the same name as the current filename with a version number one higher than the current. 8f4c1

FILENAME = 8f4d

the name of the file to be created. If only the name field of FILENAME is supplied, the system creates a file having the extension name "NLS" and assigns it the next highest version number. 8f4d1

The origin statement of the destination file will contain FILENAME, the current date and time, and the identification string of the user who is creating the file. 8f4e

The contents of the currently open file and any changes made to it since its creation, last update, or output are then copied into the named file. The Output File command always leaves the user with the named file open. 8f4f

The difference between Output File and Update File is that the file being created by Output File is ordered internally to provide more efficient access and storage. 8f4f1

Example: 8f4f2

if there is a file APPLE.NLS;4 8f4f2a

o[utput] f[ile] apple CA (creates a file
APPLE.NLS;5) 8f4f2a1

This is identical to using the command: 8f4f2b

o[utput] f[ile] CA 8f4f2b1

TENEX FILE COMMANDS

8g

DIRECTORY

8g1

The directory command causes the system to print the names of all the files in the user's file directory. 8g1a

directory CR

8g1a1

DELETE

8g2

The DELETE command removes a file from the user's directory.

8g2a

delete SP FILENAME CR

8g2a1

SP =

8g2b

hitting the space key

8g2b1

Section 3. HOW TO TELL WHERE YOU ARE

9

INTRODUCTION - CONTROL MARKER

9a

TNLS maintains a marker (CM) or "control marker" or "cursor" which is always pointing to some statement and character position within that statement in the file. When a file is first loaded into TNLS, the CM is pointing to the first character position in statement 0. If and when any commands operate on any other part of the file, the CM is repositioned. 9a1

In order to perform many of the commands described in the EDITING section of this document, the user may find it necessary to interrogate the location of the CM using the commands described below. 9a2

COMMANDS

9b

PRINT CURRENT STATEMENT

9b1

This command causes the system to move the CM to the beginning of the statement at which the CM is currently positioned and to print the statement at the terminal. 9b1a

9b1a1

This command does not require a terminating Command Accept.

9b1b

PRINT CURRENT CONTEXT

9b2

There is another version of the slash command which causes the system to print a few characters on either side of the CM with anglebrackets and a line feed break showing character position: 9b2a

/

9b2a1

For example if the CM were positioned at the seventh character in a statement which contained a series of digits separated by spaces (1 2 3 4 ...), using the slash command would cause the following to be printed: 9b2b

1 2 3 <

>4 ...

9b2b1

These commands do not require a terminating Command Accept.

PRINT CURRENT LOCATION

9b2c
9b3

This command causes the system to print out the current location of the CM:

9b3a

.

9b3a1

The position of the CM is expressed as a statement number followed by a character position within that statement preceded by a plus sign (+).

9b3b

This command does not require a terminating Command Accept.

Example:

9b3c
9b3d

. =1 +27

indicates that the CM is positioned to the 27th character of statement 1

9b3d1

Section 4. MOVING AROUND - ADDRESSING

10

INTRODUCTION

10a

The elements and commands described in this section enable the user to move to any part of a file. Normally, when one is working in TNLS, this type of moving around is done as part of a specific operation; the syntax of almost every TNLS command requires that the user specify an address at which the operation is to be performed. The elements described here are to be used when the syntax of a command demands "ADDR". Multiple address elements may be used as one address specification as long as they are separated by a space. TNLS interprets complex address expressions starting from the leftmost element to the rightmost element. 10a1

In addition to address elements, some addressing commands are described here which serve only to move the Control Marker to a specific place in the file. TNLS terminology for moving around in a file is called "jumping". Hence the commands described here are "Jump to ...".

10a2

ADDRESS ELEMENTS

10b

STATEMENT NUMBERS

10b1

Whenever statement numbers (described under file structure) are used in an address specification, they must be preceded by a period (.). 10b1a

CONTENT

10b2

The user may address a statement by content as well as by statement number by specifying a character or string of characters (LIT) bounded by square brackets. 10b2a

[LIT]

10b2a1

LIT bounded by squarebrackets causes the system to search for a statement containing LIT. The search is begun from the address specification (if any) preceding the LIT specification, or, if no other address is specified, from the character to the right of the current position of the CM. When LIT is found, the CM points to the last character of the first occurrence of LIT found. For example, if the CM is positioned to the first character position in statement 1 and the file contains: 10b2b

```
1 the a is for able 10b2b1
2 the b is for baker 10b2b2
3 the c is for catastrophic 10b2b3
```

And the user specifies [a] as an address the CM will be positioned to before the 5th character in statement 1. 10b2c

```
1 the <
   >a is for able 10b2c1
```

At this point the CM is on the first character after the <. 10b2d

If the user subsequently specifies [a] as an address again, the CM would be positioned to the 14th character in statement 1. 10b2e

```
1 the a is for <
   >able 10b2e1
```

If LIT is not found following the location of the CM, the CM is not repositioned and the user is notified that the content specified does not exist in the remainder of the file. 10b2f

NEXT 10b3

The user references the succeeding statement by the element "n". 10b3a

BACK 10b4

The user references the preceding statement by the element "b". 10b4a

CURRENT CM POSITION 10b5

The user may use of the current location of the CM in a command by not specifying an address. For example, if the user does not specify an address in the Insert command, the new statement is inserted immediately following the statement at which the CM is currently positioned as done in the first example of the Insert command. 10b5a

JUMPING 10c

JUMP TO STATEMENT 10c1

This command causes the system to move the CM to a specific statement. 10c1a

SP ADDR CA 10c1a1

SP = 10c1b

hitting the space key 10c1b1

ADDR = 10c1c

any valid address specification. Note that statement number must be preceded by a period (.). 10c1c1

When this command is executed the CM is positioned to the first character of the statement specified, 10c1d

Section 5. PRINTING ALL OR PART OF THE FILE

11

VIEWSPECS

11a

The operation of the Print command is affected by a set of parameters called "viewspecs" (a name derived from a version of the system designed for use from graphics display terminals, where the same parameters affect the "view" shown on the display screen).

11a1

When the user first enters TNLS, all of the viewspecs are automatically preset to standard values. Whenever the user issues a Print command, he has the option of changing any of the viewspecs by typing special one-letter codes.

11a1a

Generally speaking, the most common and important use of viewspecs is to cause some part of the statements in the file to be ignored for various reasons.

11a2

There are two types of viewspecs. The first type includes the Line specification whose value may range from 1 to ALL. All other viewspecs (the second type) are essentially switches which activate/deactivate various TNLS features affecting the format of the file when printed.

11a3

Any viewspecs remain in effect until specifically changed by the user.

11a4

NOTE: only those viewspecs relevant to the purposes of this document are described here. They are but a small subset of all available viewspecs.

11a5

LINES VIEWSPEC

11a5a

The lines viewspec is a value from 1 to ALL which allows the user to specify how many lines of each statement are to be printed. The lines viewspec is preset to ALL; if the user changes it to 1, only the first line of any statement will be printed.

11a5a1

The codes for setting the lines viewspec are as follows:

11a5a2

t sets T to 1
s sets T to ALL

11a5a2a

PRINTOUT OF STATEMENT NUMBERS ON/OFF (Codes m/n) 11a5b

The viewspec code "m" causes TNLS to print statement numbers before statements. This option is extremely helpful when editing files. 11a5b1

m turns statement numbers ON
n turns them OFF 11a5b1a

The standard setting for this viewspec is OFF (n). 11a5b2

BLANK LINES BETWEEN STATEMENTS ON/OFF (Codes y/z) 11a5c

The viewspec code "y" causes TNLS to put blank lines between statements on output. This makes the printout more legible, especially if statement numbers have been turned OFF. 11a5c1

y turns blank lines ON
z turns them OFF. 11a5c1a

The standard setting for this viewspec is OFF (z). 11a5c2

PRINT STATEMENT 11b

The Print command causes the specified statement of the file to be printed out on the terminal, with format control by VIEWSPEC parameters (see the above discussion of viewspecs). 11b1

p[rint] s[atement] ADDR CA VIEWSPEC CA 11b1a

ADDR = 11b2

any valid combination of address elements which specify statement location. If not specified, the current value of the CM is used. 11b2a

VIEWSPEC = 11b3

a string of viewspec parameters which affect the way that the output item appears when displayed. If no viewspecs are specified in this command, the system uses any viewspecs in effect when this command is executed. 11b3a

After the user presses the final CA, the specified statement is printed at the terminal. The printout can be stopped at any time by pressing the control o key (fo) which causes printing to cease at the end of the current statement. 11b4

After the print command is executed, the CM is positioned to the first character position of the last statement printed.

11b5

PRINT STATEMENT NEXT TO CM COMMAND

11c

This command causes the system to move the CM to the statement which is next (one below) to the statement at which the CM is currently positioned and to print the statement at the terminal according to the current viewspecs.

11c1

LF

11c1a

This command does not require a terminating Command Accept.

11c2

PRINT STATEMENT BACK FROM CM COMMAND

11d

This causes the system to move the CM to the statement which immediately precedes the statement to which the CM currently points and to print the statement at the terminal.

11d1

↑

11d1a

This command does not require a terminating Command Accept.

11d2

OUTPUT DEVICE TELETYPE - PRINTING THE FILE

11e

The output device command causes the system to process the current file so that it may be listed at the teletypewriter.

11e1

o[utput] d[evice] t[eletype] CA

11e1a

The file is printed beginning with the statement to which the Control Marker (CM) is currently positioned. To print an entire file, the CM must be positioned to statement 0 of the file.

11e2

HALT PRINTING

11f

The user may terminate printing at any time by using the character Control O (↑O). After the system receives this character, it expects the user to enter a command.

11f1

Section 6. EDITING

12

INTRODUCTION

12a

The following commands are used to change file structure and content (text). One difference exists in the operation of these commands with respect to whether the operand is a structural entity (statement) or textual entity (character, word, etc.) -- the position of the Control Marker after the command is executed. Commands which operate on statements generally cause the CM to be positioned to the first character of the statement affected by the command. Commands which operate on textual entities cause the CM to remain on the character specified by the destination address.

EDITING BY STATEMENT

12a1

12b

INSERT COMMAND

12b1

This statement is discussed in Section 2 under the creation of files.

12b1a

DELETE COMMAND

12b2

The Delete command enables the user to delete a statement from a file.

12b2a

d[ele] s[tatement] ADDR CA [ok?] CA

12b2a1

ADDR =

12b2b

any valid combination of address elements which specify the statement to be deleted.

12b2b1

After this command is executed on a statement the CM is positioned to the first character of the next statement following the deleted statement, or to the preceding statement if the deleted statement is the last statement in the file.

12b2c

Example:

12b2d

If a file consists of the statements -

12b2d1

1 inka

12b2d1a

2 dinka

12b2d1b

3 doo

12b2d1c

and the user entered the command - 12b2d2
d[el]e]t]e] s[tat]e]m]e]n]t] a]t] .2 CA [ok?] CA 12b2d2a
The file would consist of - 12b2d3
1 inka 12b2d3a
2 doo
12b2d3b

COPY COMMAND 12b3

The copy command enables the user to reproduce a statement within the file at another location. 12b3a

c[opy] s[tat]e]m]e]n]t] t]o] ADDR CA [from] ADDR CA CA 12b3a1

first ADDR = 12b3b

address at which (i.e. after which) the specified statement will be inserted. 12b3b1

second ADDR = 12b3c

address of the statement to be copied. 12b3c1

After this command is executed, the CM is positioned to the first character of the new statement created by the command. 12b3d

When this command is executed, two versions of the statement exist. 12b3e

Example: 12b3f

If a file consists of the statements - 12b3f1
1 inka 12b3f1a
2 dinka 12b3f1b
3 doo 12b3f1c

and the user entered the command - 12b3f2

c[opy] s[tat]e]m]e]n]t] t]o] .2 CA [from] .2 CA 12b3f2a

The file would consist of - 12b3f3

1 inka 12b3f3a

2 dinka 12b3f3b

3 dinka 12b3f3c

4 doo 12b3f3d

MOVE COMMAND 12b4

The move command enables the user to move a statement from one location to another within a file. 12b4a

m[ove] s[tatement to] ADDR CA [from] ADDR CA CA 12b4a1
first ADDR = 12b4b
address at which (i.e. after which) the statement specified
will be inserted 12b4b1
second ADDR = 12b4c
address of the statement to be moved. 12b4c1

After this command is executed, only one copy of the statement
exists at the first (destination) address specified and the CM
is positioned to the first character of that statement. 12b4d

Example:

12b4e
If a file consists of the statements - 12b4e1
1 inka 12b4e1a
2 dinka 12b4e1b
3 doo 12b4e1c

and the user entered the command - 12b4e2
m[ove] s[tatement to] .2 CA [from] .1 CA 12b4e2a

The file would consist of - 12b4e3
1 dinka 12b4e3a
2 inka 12b4e3b
3 doo 12b4e3c

REPLACE COMMAND 12b5

The Replace command causes a statement to be replaced with
another statement. The replacement may be another statement in
the file or text entered by the user. 12b5a

r[eplace] s[tatement at] ADDR CA [by text?] y[es CR] LIT CA
n[o] ADDR 12b5a1

first ADDR = 12b5b
 any valid combination of address elements which indicates
 the location in the file of the statement to be replaced.

y[es] = 12b5b1
 12b5c
 the entity specified will be replaced by text entered from
 the terminal (LIT). 12b5c1

LIT = 12b5d
 any string of valid characters. 12b5d1

n[o] = 12b5e
 the entity specified will be replaced by a statement already
 in the file. 12b5e1

second ADDR = 12b5f
 any valid combination of address elements which specify the
 location of the replacement statement. 12b5f1

After this command is executed, the CM is positioned to the
 first character of the replacement statement. 12b5g

An attempt to replace statement 0 by any other statement will
 cause an error condition. 12b5h

Example: 12b5i
 If a file consists of the statements - 12b5i1
 1 inka 12b5i1a
 2 dinka 12b5i1b
 3 doo 12b5i1c
 and the user entered the command - 12b5i2
 r[eplace] s[tatement at] .3 CA [by text?] n[o] .1 CA
 12b5i2a
 The file would consist of - 12b5i3
 1 inka 12b5i3a
 2 dinka 12b5i3b
 3 inka 12b5i3c
 If the user then entered the command - 12b5i4
 r[eplace] s[tatement at] .1 CA [by text?] y[es] xx CA
 12b5i4a
 The file would consist of - 12b5i5
 1 xx 12b5i5a
 2 dinka 12b5i5b

ESRI-ARC 28-JUN-72 17:41 10814
SRI-ARC 6 JUL 52 10814
TNLS BEGINNERS GUIDE -- PRELIMINARY

3 inka

12b5i5c

APPEND COMMAND

12b6

The Append command enables the user to add the text of one statement to the end of another statement.

12b6a

a[ppend to] ADDR CA [from] ADDR CA EMPTY CA
LIT

12b6a1

first ADDR =

12b6b

the destination statement

12b6b1

second ADDR =

12b6c

the statement to be added to the end of the statement at first ADDR.

12b6c1

EMPTY =

12b6d

the statement at second ADDR will be added to the end of the statement at first ADDR with no intervening text.

12b6d1

LIT =

12b6e

optional text which is inserted between the statement at first ADDR and the statement at second ADDR.

12b6e1

After this command is executed the CM is positioned to first ADDR.

12b6f

Example:

12b6g

If a file consists of the statements -

12b6g1

1 inka

12b6g1a

2 dinka

12b6g1b

3 doo

12b6g1c

and the user entered the command -

12b6g2

a[ppend to] .1 CA [from] .2 CA and CA

12b6g2a

The file would consist of -

12b6g3

1 inka and dinka

12b6g3a

2 doo

12b6g3b

12b6g3b

BREAK COMMAND

12b7

The Break Command enables the user to break any statement at a specified location causing two separate statements. 12b7a

b[reak statement at] ADDR CA CA 12b7a1

ADDR = 12b7b

location within statement where break is to occur. If the user specifies a character position that falls within a visible in the statement, the statement will be broken immediately following that visible. (Refer to the description of the Insert command in the latter part of this section for definitions of the entities character, word, and text.) 12b7b1

After this command is executed, the CM is positioned to the beginning of the new statement created by the break.

12b7c

Example:

12b7d

If a file consists of the statements -

12b7d1

1 inka and dinka

12b7d1a

2 doo

12b7d1b

and the user entered the command -

12b7d2

b[reak s[tatement at] .2 CA

12b7d2a

The file would consist of -

12b7d3

1 inka

12b7d3a

2 and dinka

12b7d3b

3 doo

12b7d3c

EDITING BY CONTENT

12c

INSERT COMMAND

12c1

In the Section 2 of this document the Insert Command is defined as the way in which the user creates information in a file by entering new statements. 12c1a

Another use of the Insert Command is to insert words, characters, and text into an existing statement: 12c1b

i[nsert] c[haracter at] ADDR CA [CR]
 w[ord at]
 t[ext at]

LIT CA 12c1b1

character = 12c1c
any letter, digit, symbol, or non-printing character 12c1c1

word = 12c1d
a contiguous string of letters and/or digits. 12c1d1

text = 12c1e
a contiguous string of characters within a statement,
delimited by two character position addresses. 12c1e1

ADDR = 12c1f
any valid combination of address elements indicating a
statement location in the current file and a character
position within that statement at which (i.e. after which)
the new entity is to be inserted. 12c1f1

LIT = 12c1g
any series of characters except CA or Centerdot (CDOT) which
is the text of the entity to be inserted. The editing
characters Backspace Character (BC), and Backspace Word (BW)
may be used when entering LIT to correct entries; ↑R may be
used to cause the system to print out the current content of
LIT. If LIT is omitted, the original statement is not
modified. 12c1g1

After this command is executed, the CM remains positioned to
the character specified by ADDR. 12c1h

The entity specified in this command is significant only in
that it affects where and how LIT will be inserted in the
statement. Characters and text are treated in exactly the same
way; LIT is inserted after the character to which the CM is
pointing. Words, are inserted after the corresponding entity
to which the CM is pointing and LIT is inserted with
appropriate spacing. 12c1i

Example: 12c1j

In the following table, the statement "i.o.u. at least \$25.00" is edited using the insert command and a variety of entities. In each case, the LIT to be inserted is the same (xxx), and each address specification assumes that the CM is positioned at the beginning of the statement. 12c1j1

entity	ADDR	new statement	
-----	-----	-----	
character	[s]	i.o.u. at leasxxxxt \$25.00.	12c1j1a
text	[s]	i.o.u. at leasxxxxt \$25.00.	
word	[s]	i.o.u. at least xxx \$25.00.	

DELETE COMMAND 12c1j1b
12c2

The Delete command enables the user to delete an entity from a statement. 12c2a

d[elate] c[haraacter] ADDR CA
 w[ord]
 t[ext] ADDR CA ADDR 12c2a1

ADDR = 12c2b

any valid combination of address elements which specify the entity to be deleted. Note that text that the beginning and ending address of the text be specified. 12c2b1

After this command is executed, the CM is positioned to the first character following the deleted entity. 12c2c

Example: 12c2d
 1 O Gilgamesh, lord of Kullab, great is thy praise. 12c2d1
 2 This was the man to whom all things were known; 12c2d2
 d w .1 CA [ok?] CA 12c2d3
 12c2d4
 1 Gilgamesh, lord of Kullab, great is thy praise. 12c2d5
 d c .2 [wn] CA [ok?] CA 12c2d6
 12c2d7
 2 This was the man to whom all things were known 12c2d8

COPY COMMAND

12c3

The copy command enables the user to reproduce an entity within the file at another location. 12c3a

c[opy] w[ord to] ADDR CA [from] ADDR CA
c[haraacter to]
t[ext to] ADDR CA ADDR 12c3a1

first ADDR = 12c3b

address at which (i.e. after which) the specified entity will be inserted. 12c3b1

second ADDR = 12c3c

address of the entity to be copied. In the case of text, the beginning and ending address of the text must both be specified. 12c3c1

After execution the CM remains on the character indicated by first ADDR. 12c3d

When this command is executed, two versions of the entity exist.

12c3e

Example:

12c3f

1 O Gilgamesh, lord of Kullab, great is thy praise 12c3f1

2 This was the man to whom all things were known; 12c3f2

3 This was the king who knew the countries of the world. 12c3f3

4 He was wise; 12c3f4

c[opy] w[ord] .3 [the] CA [from] .4 [wi] CA 12c3f5

12c3f6

3 This was the wise king who knew the countries of the world.

12c3f7

MOVE COMMAND

12c4

The move command enables the user to move an entity within a statement to another location within the same or another statement.

12c4a

m[ove] w[ord to] ADDR CA [from] ADDR CA
c[haracter to]
t[ext to] ADDR CA ADDR

12c4a1

first ADDR =

12c4b

address at which (i.e. after which) the entity specified will be inserted

12c4b1

second ADDR =

12c4c

address of the entity to be moved. In the case of text, the beginning and ending address of the text must both be specified.

12c4c1

After this command is executed, the CM remains on the character indicated by first ADDR. 12c4d

After this command is executed, only one copy of the entity exists at the first (destination) address specified.

12c4e

Example:

12c4f

1 O Gilgamesh, lord of Kullab, great is thy praise. 12c4f1

2 This was the man to whom all things were known; 12c4f2

3 This was the king who knew the countries of the world. 12c4f3

4 He was wise; 12c4f4

5 He saw mysteries and knew secret things; 12c4f5

6 He brought us a tale of before the days the flood. 12c4f6

m[ove] w[ord to] .6 [y] CA [from] [of] CA 12c4f7

6 He brought us a tale of before the days of the flood. 12c4f8

6 He brought us a tale of before the days of the flood. 12c4f9

12c4f9

REPLACE COMMAND

12c5

The Replace command causes an entity to be replaced with an equivalent entity. 12c5a

r[ep]l[a]c[e] w[ord at] ADDR CA [by text?] y[es CR] LIT CA
n[o] ADDR

c[h]a[r]a[ct]e[r at]

t[ext at] ADDR CA ADDR

ADDR CA ADDR 12c5b

first ADDR =

12c5c

any valid combination of address elements which indicates the location in the file of the entity to be replaced. Note that text requires that both the beginning and ending address of the text be specified. 12c5c1

y[es] =

12c5d

the entity specified will be replaced by text entered from the terminal (LIT). 12c5d1

LIT =

12c5e

any string of valid characters. 12c5e1

n[o] =

12c5f

the entity specified will be replaced by an equivalent entity already in the file. (That is, words are replaced by words, characters by characters, etc.) 12c5f1

second ADDR =

12c5g

any valid combination of address elements which specify the location of the replacement entity. Note that text requires that both the beginning and ending address of the replacement text be specified. 12c5g1

After this command is executed, the CM remains on the character indicated by first ADDR. 12c5h

An attempt to replace statement 0 by a second ADDR will cause an error condition. 12c5i

Example:

12c5j

1 0 Gilgamesh, lord of Kullab, great is thy praise. 12c5j1

2 This was the man to whom all things were
known; 12c5j2
3 This was the king who knew the countries
of the world. 12c5j3
4 He was wise; 12c5j4
r w [at] .2 CA [by text?] n[o] .4 CA 12c5j5
p[rint] s[tatement] .2 CA CA 12c5j6
2 He was the man to whom all things were
known; 12c5j7

SUBSTITUTE COMMAND

12c6

The Substitute Statement command is used to automatically substitute a specified text string for all occurrences of another specified string throughout a specified statement.

12c6a

s[ubstitute] s[tatement at] ADDR CA [CR]
[text] LIT CA [for] LIT CA [Go?] y[es]
CA
n[o]...

12c6a1

ADDR =

12c6b

any valid combination of address elements which specify statement location.

12c6b1

first LIT =

12c6c

any string of valid characters which will replace the characters specified by the second LIT.

12c6c1

second LIT =

12c6d

any string of valid characters in the statements specified by ADDR which will be replaced by the first LIT. (first LIT and second LIT do not have to correspond in length.)

12c6d1

y[es] = 12c6e

(in response to "Go?") only the substitution(s) indicated by LIT are to be executed. CA is equivalent to typing y[es].

n[o] = 12c6f

other substitutions are to be made in addition to those already specified. In essence this continues the substitute mode and the subsequent syntax begins with the system request "text". 12c6f1

When multiple pairs are specified in the substitute command, the system looks for matches at each character position in the structure specified for occurrences of all pairs. This mode of operation allows the user to for example substitute all occurrences of the character "A" for "B" and at the same time substitute "B" for "A". 12c6g

After this command is executed, the CM is positioned to the character indicated by ADDR. 12c6h

Example: 12c6i

l and on ...and on ... and on... 12c6i1

s s .1 CA

[text] * CA [for text] SP CA [Go?] n[o CR] 12c6i2

[text] / CA [for text] ... CA [Go?] CA 12c6i3

p s CA CA

l and*on*/and*on*/*and*on/ 12c6i4

EDITING ACROSS THE FILE - SUBSTITUTE BRANCH

12d

The following command is documented here because although some of its terminology has not been explained in this beginner(s) guide, it is of great value to the TNLS user. It is the Substitute Branch command which can be used to effect a substitution of one text string for another throughout a file according to the viewspecs in effect when the substitute is executed.

12d1

```
s[ubstitute] b[ranch at] .0 CA [CR]
[ text ] LIT CA [for] LIT CA [Go?] CA
```

12d1a

first LIT =

12d2

any string of valid characters which will replace the characters specified by the second LIT.

12d2a

second LIT =

12d3

any string of valid characters in the statements specified by ADDR which will be replaced by the first LIT. (first LIT and second LIT do not have to correspond in length.)

12d3a

Viewspecs affect the operation of this command. For example, if the viewspec t (one line) is in effect when a substitute is performed, only the first line of statements will be affected.

12d4

After this command is executed, the CM is positioned to the character indicated by ADDR.

12d5

Example:

12d6

If a file contained the statements:

12d6a

```
<SOMEONE>XFILE.NLS;2, 22-JUN-72 9:27 SSS ;
```

12d6a1

```
1 This is the NLS beginner's guide
```

12d6a2

```
2 for new NLS users
```

12d6a3

```
3 containing information about the
```

12d6a4

```
4 nls system and
```

12d6a5

```
5 NLS commands.
```

12d6a6

The command:

12d6b

```
s[ubstitute] b[ranch] .0 CA
```

```
[ text ] TNLS [for text] NLS CA [go?] CA
```

12d6b1

would change the file to be:

12d6c

```
<SOMEONE>XFILE.TNLS;2, 22-JUN-72 9:27 SSS ;
```

12d6c1

```
1 This is the TNLS beginner's guide
```

12d6c2

```
2 for new TNLS users
```

12d6c3

```
3 containing information about the
```

12d6c4

```
4 nls system and
```

12d6c5

```
5 TNLS commands.
```

12d6c6

Section 7. USING THE SYSTEM TO SEND/RECEIVE MESSAGES TO/FROM OTHER
USERS

13

INTRODUCTION

13a

The Journal is an open-ended information storage and retrieval system intended to facilitate communication and the availability of documentation within the TNLS user community. The Journal System consists of a set of procedures which enable the automatic cataloging and distribution of information items (messages and documents) within the community. 13a1

The user may send messages, files, or statements from files to another user by using the Journal system. In order for the user to submit a file or a statement from a file to the Journal, that file must be currently loaded (open) in TNLS. 13a2

SUBMITTING ITEMS TO THE JOURNAL

13b

The user enters the Journal submode from TNLS with the Execute Journal command: When the user types the characters "e j" the system prints the Journal herald character "E". 13b1

e[xcute] j[ournal] CA
Es[ubmit] s[atement] ADDR CA [CR]
f[ile]
m[essage CR] LIT 13b1a

[number:] CA [CATNUM] 13b1b

ADDR = 13b1c

any combination of address elements specifying a statement
in a file. 13b1c1

LIT = 13b1d

any string of characters (message content). When entering
LIT, the control characters for editing entries may be used. 13b1d1

CATNUM = 13b1e

the system-assigned catalog number which will be associated with the current Journal entry. 13b1e1

"Submit statement" is identical to "submit message" in terms of Journal processing; however, messages are recommended for short items. They differ in that "submit statement" causes the system to go to the address specified to obtain the text of the message whereas "submit message" requires the user to enter the text from the terminal at the time of submission. 13b2

When a message or statement is delivered it appears in a file belonging to the distributee(s). When a file is delivered, the distributee receives only an entry giving information about the file sent to him. He must use the command described for loading files in order to access the Journal item. 13b3

Note that depending on the load on the system, the Journal process may take more than a few seconds. The user may ascertain that the Journal processing is running and that the system is active by typing ^T. If there is no response to this control character, the system is down. 13b4

NOTE: Do not attempt to issue a Control C (^c) during this process as the results are unpredictable. 13b4a

Once this procedure is completed, the user is at the Journal submode command level indicated by the system typing the herald character "E8".

13b5

JOURNAL COMMANDS

13c

In the Journal submode, the following commands are allowed: 13c1

INTERROGATE

13c2

This command causes the Journal system to enter a mode whereby the user is interrogated for information. 13c2a

i[nterrogate] CA

13c2a1

The information which will be requested by the system will be:

Title:

Distribution:

Status:

Go?

13c2b

NOTE: Much additional information is maintained by the system about each Journal item. This information generally consists of default values supplied by the system. This information may be modified by the user only by deliberate specification. The commands which enable this are not documented here but may be found in the NIC Journal User Guide (see -- 7635,). 13c2c

The user should respond to the requests as follows: 13c2d

Title: LIT CA 13c2e

Where LIT = any series of characters which constitute the title. 13c2e1

This allows the user to enter a title which will be used in the hard copy printouts of the document or catalog listings and indices. 13c2e2

Distribution: IDENTLIST CA 13c2f

Distribution allows the user to specify a list of persons who will receive copies of the document being entered. 13c2f1

IDENTLIST consists of a series of IDENT's separated by spaces. It may consist of IDENTs of groups as well as individuals. 13c2f1a

The list of IDENTs is maintained in the ARPA Network Current Directory of Network Participants , NIC 5150. 13c2f1a1

In a case where the user does not know the official IDENT of an individual, he may cause the system to search for the individual's last name in the master list of persons known to the system called the IDENTFILE. By reponding to any request for IDENT in any Journal command by typing a period followed by the last name of the individual (.LASTNAME) the system will search the IDENTFILE for all occurences of that name. The system will print out all such names found along with their IDENTs. It will then ask if the user wishes to enter a new IDENT (i.e., if the individual is not found in the list). If the individual is found the response to this question should be "n" (no). The user may then continue responding to the original request by entering the appropriate IDENT. 13c2f1a2

If the name does not appear in the list generated by the system, the individual is not known to the system and as such may not receive items through the Journal.

13c2f1a3

The procedure for obtaining an IDENT for an individual not known to the system is not covered in this document but may be found in the Journal User Guide.

13c2f1a3a

If the user attempts to enter an IDENT that the system does not recognize, the system will print a question mark (?) at the terminal.

13c2f1b

The user may insert notes for particular individuals in the list by enclosing a text string in parentheses immediately after (with no intervening blank spaces) the IDENT of the person to whom the comment is directed.

13c2f2

Status: CA

13c2g

This causes the system to print the values for the items supplied by the user in response to the above.

13c2g1

Go? y[es]
CA
n[0]
SP

13c2h

Responding to this prompt with "yes" or CA causes the Journal entry process to begin.

13c2h1

Responding by "no" or SP returns the user to the Journal system command level wherein he may modify and/or add to information entered during the Interrogate session.

13c2h2

If the user does not wish to respond to a certain request during the Interrogate session he may use a Command Delete (CD) and proceed to the next request.

13c3

Instead of using the Interrogate command to cause the system to request information from the user for each entry item, the following commands may be used to enter each item independently, to change a field previously entered and/or to add additional information not requested by the Interrogate procedure.

13c4

AUTHOR 13c5

This allows the user to specify the contents of the author field in the Journal header statement and in catalog entries. 13c5a

a[uthor(s):] CA
IDENTLIST CA 13c5a1

The normal response is CA indicating the IDENT of the username used by the current user when he entered TNLS. 13c5b

TITLE 13c6

This command is identical to responding to the Title request under the interrogate command. 13c6a

t[itle:] LIT CA 13c6a1

DISTRIBUTION 13c7

This command is identical to responding to the Distribution request under the interrogate command. 13c7a

d[istribution:] IDENTLIST CA 13c7a1

STATUS 13c8

As or after the user enters header information, he may review it by causing the system to type out the values currently in the header statement of the Journal item. 13c8a

st[atus] CA 13c8a1

At this time, the user may replace the items in this list by re-entering any command. 13c8b

QUIT 13c9

This command returns the user to the TNLS command mode. It should not be necessary except in cases where there has been a system error. 13c9a

q[uit] CA 13c9a1

GO 13c10

This causes the Journal entry process to begin after the user has specified any header information he wishes. 13c11

```
g[o?] y[es]
      CA
      n[0]
      SP 13c11a
```

This command is identical to responding to the Go? request under the Interrogate command. 13c12

The actual Journal process may take from 30 seconds to three minutes depending on the current system load. When the Journal process is completed, the system prints a reference to the new Journal entry created by the process and issues the TNLS herald character "*". 13c13

USER STATUS ON LEAVING JOURNAL SYSTEM 13c14

After the user has successfully entered an item into the Journal the user is returned to TNLS. The current file open is the file being submitted. If the item was a document (i.e. not a message) the Journal entry process will have created a permanent file named CATNUM which may not be further changed; however, it may be viewed, referenced, etc. If the item was a message, it will be added to the Journal message file.

13c14a

SAMPLE JOURNAL SESSION 13d

The following is a demonstration of how a message is submitted to the Journal by a Network user (including login, TNLS access, and logout procedure). 13d1

```
tc
[ARC-TENEX 1.28.01.11 3-AUG-71 ARC EXEC 1.32.06] 13d2
```

```
log CR [CR]
[(user)] DOE CR [CR]
[(password)] DDD CR [CR]
[(account #)] 1 CR [CR] (a user named DOE logs into
                        the system; his IDENT is DDD) 13d3
```

```
[JOB11 ON TTY14 3-AUG-71 17:11] 13d4
```

@nls CR (accesses TNLS system)
[ID:] DDD CR
[device:] N[et-tty] 13d5

*e[xecute Journal] (access Journal system) 13d6

S[ubmit] m[essage] 13d7

THIS IS A SAMPLE MESSAGE CA (contents of message) 13d8

[number] CA [3333] (typing CA after request for
number causes system to assign
DDD a unique catalog number [3333]
for the message) 13d9

EEi[nterrogate] CA (causes the system to prompt the
user for the correct information
required to send the item through
the Journal) 13d10

EE[title:] A SAMPLE JOURNAL SESSION CA 13d11

EE[distribution] XXX YYY CA
(XXX and YYY are IDENTs of other
persons known to the system) 13d12

EE[status] CA 13d13

.
. (the system reiterates all information entered
. by the user)
. 13d14

EE[go:] CD (the user does not want to "go" now but
change information requested by the
Interrogate procedure) 13d15

EE[distribution] ZZZ YYY CA (user changes the distribution list) 13d16

EEg[o:] CA (tells the system to begin Journal process.
Note that as the author command was not
used, DDD is assumed) 13d17

[JOURNAL SYSTEM IN PROGRESS] (the system is processing
the new Journal entry) 13d18

NOTE: The time required to process a Journal entry ranges from a minimum of about 30 seconds to three minutes depending on the current system load. When the Journal process is finished and the system prints a reference to the new Journal entry created by the process and issues the TNLS herald character "*". 13d18a

[Journal,JRN1,J333:gw] 13d19

e[xecute] q[uit] (the user has been returned to the TNLS command level as signified by the system typing the herald character ""; the user then types the Execute Quit command to return to TENEX) 13d20

@logout CR (the user logs out) 13d21

SIMPLE WAY TO READ ITEMS 13e

Journal messages and documents are currently sent in hardcopy form and, for TNLS users, by direct online delivery through the addition of a reference to the Journal entry in the "Journal" section in the addressee's TNLS initial file. (This section is created in a user's initial file automatically when any Journal item is sent to him.) 13e1

Since a user's initial file is automatically ploaded for him when he enters TNLS, Journal items are readily accessible. 13e2

Messages are presented in their entirety in the user's initial file. When the user is through reading the message, he may remove it from the file by using the delete statement command and specifying the statement number of the journal item in his file. 13e2a

A Journal item (document) that is a file may be read by accessing the file and printing it at the terminal. 13e2b

Journal files may be accessed by using the Load File command (see Section 2) and specifying the catalog number of the Journal item as the filename. Once the user has accessed the Journal item, he may view it by printing each statement in the Journal file, or by using the Output Device Teletype command (see Section 6) to print the entire file at the terminal. 13e2b1

Example: 13e2b2

The following is a sample user file containing two items delivered to the user through the Journal. The first is a message and the second is a file. 13e2b2a

Meeting Notice 13e2b2a1

HGL 20-JUN-72 13:16 2797

Meeting Notice

Message: There will be a meeting of all persons interested in the recent changes to TNLS, on Monday June 26, in the conference room.

13e2b2a2

DSK 20-JUN-72 14:06 2798

Software Interface

Location: (JJOURNAL, 2798, 1:w)

13e2b2a3

The format of the location specification for the Journal file is: the directory (JJOURNAL), the name of the file (2798), the first statement in the file (1), and a viewspec (w). Thus to access the file use the command:

13e2b2b

l[oad file] 2798 CA

13e2b2b1

Note: it is not necessary to specify the directory name of the file since the system assumes any filename beginning with a digit to be a journal file. 13e2b2b1a

The file may then be listed at the terminal by using the output device teletype command. The entire file will be printed as the CM is positioned at statement 0 when any file is loaded. 13e2b2c

Section 8. COMMAND SUMMARY AND HELP

14

ONLINE USER HELP

14a

The syntax of all TENEX and TNLS commands available to the user are also documented online. The user may obtain a complete list of these commands by using the Help command by typing a question mark (?) in either TENEX or TNLS. Similarly, if the user is unsure of the required elements of any command he may type "?" to cause the system to print all arguments permitted for the current command. 14a1

NOTE: the commands and required elements printed out for any command when the user types "?" are a superset of the commands documented in this manual. 14a1a

COMMAND SUMMARY

14b

HOW TO GET IN AND OUT OF THE SYSTEM

14b1

LOGIN

14b1a

```
log CR [CR]
[(user)] USERNAME CR [CR]
[(password)] PASSWORD CR [CR]
[(account #)] ACCOUNT NO. CR [CR]
```

14b1a1

ACCESSING TNLS - TNLS

14b1b

```
@nls CR [CR]
[id:] USER IDENTIFICATION CR [CR]
[device:] DEVICE
```

14b1b1

RESUMING WORK IN TNLS - CONTINUE

14b1c

```
continue CR
```

14b1c1

LEAVING TNLS - EXECUTE QUIT

14b1d

```
e[execute] q[uit] CA
```

14b1d1

LEAVING THE TENEX SYSTEM - LOGOUT COMMAND

14b1e

```
logout CR
```

14b1e1

WHAT A FILE IS AND HOW TO WORK WITH IT	14b2
ENTERING TEXT IN TNLS - INSERT COMMAND	14b2a
i[nsert] s[tatement at] ADDR CA CA [CR] LIT CA CDOT	14b2a1
EDITING WHILE ENTERING TEXT	14b2b
The following characters may be used as follows when entering text:	14b2b1
BC or ↑A -	14b2b1a
backspace previous character. One character is deleted each time this key is pressed.	14b2b1a1
BW or ↑W -	14b2b1b
backspace previous visible/word. One word and its preceding space is deleted each time this key is pressed.	14b2b1b1
BS or ↑Q -	14b2b1c
backspace statement. The entire current LIT is deleted when this key is pressed.	14b2b1c1
↑R -	14b2b1d
show current LIT. This key causes the system to print the current content of LIT as it is being entered. This is helpful when several of the above characters have been used during text entry and the user is unsure of the actual content of LIT.	14b2b1d1
TNLS FILE COMMANDS	14b2c
NULL File	14b2c1
n[ull] f[ile] FILENAME CA	14b2c1a
LOAD FILE	14b2c2
l[oad] f[ile] FILENAME CA	14b2c2a

UPDATE FILE	14b2c3
u[pdate] CA	14b2c3a
TENEX FILE COMMANDS	14b2d
DIRECTORY	14b2d1
directory CR	14b2d1a
DELETE	14b2d2
delete SP FILENAME CR	14b2d2a
HOW TO TELL WHERE YOU ARE	14b3
PRINT CURRENT STATEMENT	14b3a
	14b3a1
PRINT CURRENT CONTEXT	14b3b
/	14b3b1
PRINT CURRENT LOCATION	14b3c
.	14b3c1
MOVING AROUND	14b4
JUMP TO STATEMENT	14b4a
SP .STATEMENT NUMBER CA	14b4a1
PRINTING ALL OR PART OF THE FILE	14b5
PRINT STATEMENT	14b5a
p[rint] s[tatement] ADDR CA VIEWSPEC CA	14b5a1
PRINT STATEMENT NEXT TO CM COMMAND	14b5b
LF	14b5b1

PRINT STATEMENT BACK FROM CM COMMAND	14b5c
↑	14b5c1
HALT PRINTING	14b5d
↑o	14b5d1
OUTPUT DEVICE TELETYPE - PRINTING THE ENTIRE FILE	14b5e
o[utput] d[evice] t[eletype] CA	14b5e1
EDITING	14b6
EDITING BY STATEMENT	14b6a
DELETE COMMAND	14b6a1
d[elate] s[tatement] ADDR CA [OK?] CA	14b6a1a
COPY COMMAND	14b6a2
c[opy] s[tatement to] ADDR CA [from] ADDR CA CA	14b6a2a
MOVE COMMAND	14b6a3
m[ove] s[tatement to] ADDR CA [from] ADDR CA CA	14b6a3a
REPLACE COMMAND	14b6a4
r[eplace] s[tatement at] ADDR CA [by text?] y[es CR] LIT CA n[o] ADDR	14b6a5
APPEND COMMAND	14b6a6
a[ppend to] ADDR CA [from] ADDR CA EMPTY CA LIT	14b6a6a
BREAK COMMAND	14b6a7
b[reak statement at] ADDR CA CA	14b6a7a

EDITING BY CONTENT	14b6b
INSERT COMMAND	14b6b1
i[nsert] c[haracter at] ADDR CA [CR] w[ord at] t[ext at]	
LIT CA	14b6b1a
DELETE COMMAND	14b6b2
d[elate] c[haracter] ADDR CA [OK?] CA w[ord] t[ext] ADDR CA ADDR	14b6b2a
COPY COMMAND	14b6b3
c[opy] w[ord to] ADDR CA [from] ADDR CA c[haracter to] t[ext to] ADDR CA ADDR	14b6b3a
MOVE COMMAND	14b6b4
m[ove] w[ord to] ADDR CA [from] ADDR CA c[haracter to] t[ext to] ADDR CA ADDR	14b6b4a
REPLACE COMMAND	14b6b5
r[eplace] w[ord at] ADDR CA [by text?] y[es CR] LIT CA n[o] ADDR c[haracter at] t[ext at] ADDR CA ADDR	14b6b5a
SUBSTITUTE COMMAND	14b6b6
s[ubstitute] s[tatement at] ADDR CA [CR] [text] LIT CA [for] LIT CA [Go?] CA	14b6b6a
EDITING ACROSS THE FILE	14b6c
s[ubstitute] b[ranch at] .0 CA [CR] [text] LIT CA [for] LIT CA [Go?] CA	14b6c1

SENDING/RECEIVING MESSAGES TO/FROM OTHER USERS

14b7

SUBMITTING ITEMS TO THE JOURNAL

14b7a

e[xecute] j[ournal] CA
 s[ubmit] s[tatement] ADDR CA [CR]
 f[ile]
 m[essage CR] LIT

14b7a1

[number:] CA [CATNUM]

14b7a2

JOURNAL COMMANDS

14b7b

INTERROGATE

14b7b1

i[nterrogate] CA

14b7b1a

AUTHOR

14b7b2

a[uthor(s):] CA
 IDENTLIST CA

14b7b2a

TITLE

14b7b3

t[itle:] LIT CA

14b7b3a

DISTRIBUTION

14b7b4

d[istribution:] IDENTLIST CA

14b7b4a

STATUS

14b7b5

st[atus] CA

14b7b5a

QUIT

14b7b6

q[uit] CA

14b7b6a

GO

14b7b7

g[o?] y[es]
 CA
 n[0]
 SP

14b7b7a

(J10814) 28-JUN-72 17:41; Title: Author(s): S.R.I. - Augmentation Research Center/SSRI-ARC; Distribution: N. Dean Meyer, Joy A. Glenn, Kay F. Byrd, Ralph Prather, James E. White, Augmentation Research Handbook, Jacques F. Vallee, Diane S. Kaye, Paul Rech, Michael D. Kudlick, Don Limuti, William R. Ferguson, Linda L. Lane, Marilyn F. Auerbach, Walt Bass, Douglas C. Engelbart, Beauregard A. Hardeman, Martin E. Hardy, J. D. Hopper, Charles H. Irby, Mil E. Jernigan, Harvey G. Lehtman, Jeanne B. North, James C. Norton, Cindy Page, William H. Paxton, Jeffrey C. Peters, Jake Ratliff, Barbara E. Row, Ed K. Van De Riet, Dirk H. van Nouhuys, Kenneth E. Victor, Donald C. Wallace, Richard W. Watson, Don I. Andrews, Arnold M. Ostebee/SRI-ARC AMO; Sub-Collections: SRI-ARC; Clerk: MFA;
Origin: <AUERBACH>TNLS.NLS;1, 28-JUN-72 17:38 MFA ;

Outline and Teaching Load for June 29-30th TNLS Course

This outline shows the order of lecture subjects in the June 29th-30th TNLS Course and offers rough guesses about the timing.	1
Because the class is large it is more likely to progress through the course slowly than quickly.	1a
The initials in angle brackets name who should talk about each subject.	1b
Practice periods break the lecturing frequently.	1c
I will be present all of both days to answer questions and help with practice; I would like Marilyn to be there too on the morning of the first day and the afternoon of the second day and Beau on the afternoon of the first day and the morning of the second day.	1d
Block I	2
Introduction<DVN>	2a
Of class members to one another	2a1
To ARC	2a2
To NIC	2a3
To Concept of NLS	2a4
Sending a Journal Message<MFA>	2b
TENEX, Log in, Log out, CR,	2b1
NLS login, quit, ↑A, ↑Q, ↑W, CA, ↑X	2b2
Sending the message	2b3
Basic Text Handling<DVN>	2c
NLS vs TENEX command grammar	2c1
Insert, Delete, Print	2c2
Substitute, Cdot	2c3
Load, output	2c4
Systat, links<dvN>	2d

Outline and Teaching Load for June 29-30th TNLS Course

Basic Editing<RWW>	2e
copy,move,replace:statement, word, text	2e1
<Lunch?>	2f
Basic File Handling<DvN>	2g
Simple Directory Commands	2g1
Altmode, space	2g2
Filenames	2g3
Output, Load, SP (link)	2g4
NLS File Structure<BAH>	2h
Block I address<DVN>	2i
Address matrix	2i1
Links	2i2
Address by character count	2i3
Address by content	2i4
Where am I?	2i5
Editing Matrix<DvN>	2j
Matrix	2j1
Command by command	2j2
<1st day end?>	3
Block II	4
Full Journal Entry<DVN>	4a
ALL @ and @@ commands	4a1
., ?, ?[LIT]	4a2
Partial Copies<BAH>	4b
Idea	4b1

Outline and Teaching Load for June 29-30th TNLS Course

Execute status file	4b2
Viewspecs<dvN>	4c
including assimilate	4c1
Locator<bah><MFA>	4d
<2nd day has often ended here>	5
View change<dvN>	5a
Block II address - Strucrels<BAH>	5b
Print directives<MFA>	5c
Block III	6
Complete journal, ident, number <MFA>	6a
Content <DVN>	6b
Execute Edit	6c
Go to Sort, etc.	6d

Outline and Teaching Load for June 29-30th TNL Course

(J10815) 22-JUN-72 17:20; Title: Author(s): Dirk H. van Nouhuys/DVN;
Distribution: Richard W. Watson, Marilyn F. Auerbach, Beauregard A.
Hardeman/RWW MFA BAH; Sub-Collections: SRI-ARC; Clerk: DVN;
Origin: <VANNOUHUYS>JUNOUT.NLS;3, 22-JUN-72 17:16 DVN ;

Dave: Cordell just got back from Europe and called me because he couldn't reach you. It's fine to stay at his place. Come absolutely anytime. Use second bedroom on right upstairs.
-Steve.

(J10816) 23-JUN-72 13:07; Title: Author(s): Steve D. Crocker/SDC2;
Distribution: David H. Crocker/DHC; Sub-Collections: NIC; Clerk: SDC2;

ISI ORGANIZATION CHART

Jim--this is the current organization of the branch at RADG in which I work and which will be the test subjects in our evaluation.

ISI ORGANIZATION CHART

The ISI branch contains the following people as of July 72:	1
INFORMATION PROCESSING BRANCH (ISI)	1a
Frank Tomaini-chief	1a1
Lee Kortz-deputy cheif	1a2
Thomas Bucciero-admin	1a3
Louis Cassetta-admin	1a4
Rebecca Levine-sec	1a5
Marcelle Petell-sec	1a6
INFORMATION MANAGEMENT SCIENCES SECTION (ISIM)	1b
John McNamara-chief	1b1
Josephine Stellato-sec	1b2
James Bair	1b2a
Dean Bergstrom	1b2b
Richard Caliccia	1b2c
Joe Cavano	1b2d
Dave Daughtry	1b2e
Roc Iuorno	1b2f
Frank LaMonica	1b2g
Thomas Lawrence	1b2h
Ray Liuzzi	1b2i
Roger Panara	1b2j
William Rzebka	1b2k
Frank Slwia	1b2l
Duane Stone	1b2m
Don VanAlstine	1b2n

ISI ORGANIZATION CHART

SYSTEM SOFTWARE SCIENCES SECTION (ISIS)

	1c
Richard Nelson-chief	1c1
Carmella Marcoccia-sec	1c2
Lyal Bauer	1c2a
James Cellini	1c2b
Casper DeFiore	1c2c
Sam DiNitto	1c2d
Don Elefante	1c2e
Jeff Ives	1c2f
Alan Klayton	1c2g
Mike Landes	1c2h
John McLean	1c2i
Fred Norman	1c2j
William Patterson	1c2k
James Previte	1c2l
Oscar Reiman	1c2m
Richard Robinson	1c2n
Rona Stillman	1c2o
Dave Trad	1c2p
Armond Vito	1c2q
Dennis Williams	1c2r
Doug White	1c2s

ISI ORGANIZATION CHART

(J10817) 23-JUN-72 13:57; Title: Author(s): Duane L. Stone/DLS;
Distribution: James C. Norton, Thomas J. Barcalow, John L. McNamara/JCN
TJB JLM; Sub-Collections: RADC; Clerk: DLS;
Origin: <STONE>ORG.NLS;1, 23-JUN-72 13:52 DLS ;

AHI GROUP INPUT TO TPO-14

This is the AHI group's input to the annual RADC planing exercise for the year FY-74---FY-78.

AHI GROUP INPUT TO TPO-14

Job Performance Aids- Efforts in this area will result in a general purpose integrated system of hardware, software, method, procedures and training which can significantly improve the performance of individuals and teams engaged in complex problem solving and decision making tasks.

1

All the efforts in this area are based on the Augmented Human Intellect (AHI) technology developed at Stanford Research Institute (SRI) under ARPA sponsorship over the past 10 years. RADC personnel have monitored this program for the past 5 years and are convinced that the feasibility of significant job performance improvement in an intellectual environment is indeed possible. SRI has developed the system through a "bootstrapping" process where the system's capabilities at any particular instance in time are used to further develop the system. The augmentation capabilities of the system have evolved through the individual, to the team, to the organization stage, and are now approaching the community stage via use by individuals and teams around the ARPANET.

1a

The development activity during FY-72 has been concentrated on acquiring the necessary physical plant for use and evaluation of the SRI AHI system. RADC is connected to the ARPANET via a standard H316 Terminal Interface Processor (TIP). Internal lines have been acquired to connect the terminals to the TIP. Six Execuport 300 teleprinters, three IMLAC PDS-ID crt display terminals, and four Termicette cassette recorders have been purchased. Five engineers, a section chief, a secretary and an administrator have recieved varying amounts of training in the use of the system. The initial specification of a baseline management system complete with SOPs, file design, and personnel role assignment has been completed. The research group at RADC has progressed to the point where they are doing the bulk of their daily work using the system; in fact, this TPO was prepared, coordinated and edited a number of times using the system. The FY-73 activity will be concentrated on training the rest of the ISI branch (approx. 40 people), procuring additional terminals and a medium speed line printer, and refining the baseline management system--to include interfacing it with MASIS. The activity in FY-74 and out years will be concerned with developing and incorporating advanced management techniques such as DELPHI, resource allocation strategies, and advanced graphics capabilities. This activity will be significantly enhanced by the use of others' research work via the ARPANET.

1b

AHI GROUP INPUT TO TPO-14

The evaluation of a system as complex as AHI must be conducted over an extended period of time, an extended set of jobs, and across a representative sample of Air Force people to allow the results to be generalized to other environments. The evaluation at RADC will cover a two year period, and include engineers, secretaries, administrators, and managers at three levels in the chain of command. The evaluation will evolve slightly behind the actual implementation of the system within the ISI branch. Results on the effectiveness of the system in augmenting individuals and teams will be available near the middle of FY-73, and organizational results at the end of FY-74. A parallel implementation and evaluation will be conducted within the office off the Assistant for Automation DCS/PEO to the extent that funds will allow. The measures currently used in the evaluation include:

lc

Psychometric--questionnaires, surveys, and interviews given in a controlled experimental environment.

lc1

Performance--whenever a working group, section, or branch is charged with a specific job their performance will be compared with that of an equivalent organizational unit. Data will be collected on elapsed time, manhour time, and the subjective judgements of the quality of the work made by the two units' common manager. Comparisons will be made between the AHI system and other available systems with comparable capability in specific areas; eg. text editing.

lc2

Cost/Benefit--A running systems analysis will be conducted over the two year period to enable detailed specification of the cost/benefit tradeoffs which can be made in implementing portions of the general system in other environments.

lc3

During FY-73 & 74 it will be possible (based on the evaluation activity under 6.3) to specify and refine the design of a prototype system for a portion of the DCS/PEO's problems. An engineering development project will be created to allow for equipment procurement and installation. The current estimate of a minimal configuration for useful test would consist of a TIP to connect to the ARPANET, an IMP for connection to the Command Post computer or the Data Services computer, a secure line or krypto device connecting the two IMPs and a mix of 10-15 teleprinter and crt terminals. The test population within DSC/PEO should consist of a working group, its secretaries, and at least three levels of administrative and command personnel in the management chain.

ld

AHI GROUP INPUT TO TPO-14

DCS/P&O has been given engineering support by RADC at a low level since FY-68 under ESP project 921A-9339, "Data Handling Support for Air Staff". During FY-72 an Execuport-300 was loaned to the office of Assistant for Automation (AF/XOA) and one action officer given preliminary training in using the AHI system via the NBS TIP and the ARPANET. He in turn has given instruction to another officer in Data Services. The purpose of this activity is to familiarize a limited number of people within Air Staff with the capabilities of the AHI system. In addition to the AHI activity, RADC has leased a BR-700 Information System for a 3 month operational test within DCS/P&O. The 700 will be used by administrators to track the progress/status of some 800-1000 actions which come into the organization each month. This system is a stand-alone message composition, sending and retrieving device; which is representative of an off-the-shelf intermediate operational capability for use prior to the installation of any portion of the AHI technology. 1e

During FY-73 & 74 a continuing operations research activity will be carried on within DCS/P&O. This will be conducted by a combined team from AF/XOA, RADC and SRI. The purpose of this activity will be to explicitly identify the functional and organizational units and their environmental constraints which will allow a meaningful test of the AHI technology. The team's productivity should be significantly enhanced by using the system to plan, implement, and document this activity; and will constitute an informal evaluation of the effect the technology can have on team performance, where the team members are geographically separated. 1f

The equipment purchased, procedures established, and training acquired under the 6.4 EDP should allow an early conversion to an initial operating capability. It will be necessary to acquire additional NLS service either by procuring it from a commercial organization, transferring software to an internal Pentagon computer or acquiring an additional computer. Also, an additional number of terminals will be required and another IMP may be desired to link an additional Pentagon computer into the total system available to the action officer. A COC is anticipated by FY-80.

AHI GROUP INPUT TO TPO-14

JOB PERFORMANCE AIDS

EFFORT	P.E.	72	73	74	75	76	77	78	79	I/C
		1200	1300	1300	1400	1400	1500	1500	1600	
NET INFO CENTER & AUG SYS DEV	6.1 0967	.5	.5	.5	.5	.5	.5	.5	.5	C
DEV & IMP AHI TECH	6.2 5581	76 1	150 2	100 2	50 2	50 2	50 3	50 3	50 3	I/C
EVALUATION	6.3 5550	84/75 2	0/75 3	100 3						I
PROTOTYPE SYS	6.4				300 2	100 3	100 3			
OPS RES SUPPORT	oth 9339	.5	.5	.5						I
AUTO AIR STAFF PLAN & ADMIN SYS	6.5							500 I/	100	C
TOTAL 6.2		76	150	100	50	50	50	50	50	
TOTAL 6.3		84	0	100						
TOTAL OTHER		1200	1300	1300	1700	1500	1600	2000	1700	

AHI GROUP INPUT TO TPO-14

(J10818) 23-JUN-72 14:03; Title: Author(s): Duane L. Stone/DLS;
Distribution: Duane L. Stone, James H. Bair, Thomas F. Lawrence, James
C. Norton, Paul Rech, Dirk H. van Nouhuys, John L. McNamara, Marcelle D.
Petell, Robert Logan/RBMS JLM MDP RL; Sub-Collections: RADC RBMS; Clerk:
DLS;
Origin: <STONE>AHITPO.NLS;1, 13-JUN-72 6:57 DLS ;

RADG TPO-14 DATA PROCESSING SOFTWARE

This is the narrative of the annual planning document submitted by the branch, which is combined with some 20 other similiar documents from around the Center, and submitted to higher Headquarters for review etc. It should give you some feel for the technical areas covered by the branch.

RADC TPO-14 DATA PROCESSING SOFTWARE

3.14 Technology Planning Objective No. 14 -DATA PROCESSING SOFTWARE 1

3.14.1 GENERAL OBJECTIVE 1a

(U) The general objective of this TPO is to develop techniques to improve the specification, testing, evaluation and reliability of software and to evaluate associative processing for AF applications. 1a1

3.14.2 SPECIFIC GOALS AND TECHNICAL APPROACHES 1b

3.14.2.1 The Overview Chart (Fig. 3.14-1) illustrates the plan for meeting the general objective. This objective will be systems shown on the right side of the chart. In general, they are of the command and control or management information nature. These system are essential to any application of Air Force power in response to a directive from the President of the United States. In addition, data processing software supports data reduction activities in practically all other systems. To support these conceptual systems, the efforts in this TPO are oriented toward the eleven major goals or "products" shown on the overview chart. Although details of these cannot be defined fully at this time, available estimates of the performance requirements, design configuration and operational utilization are provided in the following paragraphs. 1b1

3.14.2.1.1 (U) Standard Specification for HOL--This product will provide a complete, standard and unambiguous specification for AF HOLs by eliminating varying interpretations of HCL specifications resulting from unclear or incomplete English language definition. It will aid in the "standardization" of HOLs for AF use. The Semantics Oriented Language (SEMANOL) development will be employed to accomplish this goal; as it has been mathematically proven that SEMANOL is an unambiguous meta-language and provides a method of testing for language completeness. 1b1a

3.14.2.1.2 (U) Compiler Validation Systems--This product will enable AF installations to procure HOL compilers with a greater confidence by providing extensive testing capabilities which will determine the compilers' degree of adherence to the language specification and compiler performance requirements. This capability will be provided by capitalizing on previous efforts by the Air Force, Navy and the National Bureau of Standards in the area of compiler validation, and will incorporate the use of modern thorem proving techniques. 1b1b

RADAC TPO-14 DATA PROCESSING SOFTWARE

3.14.2.1.3 (U) Standard HOL Implementation Tool--This product will enable the AF to generate standard high quality HOL compilers so that the use of Higher Order Languages will be promoted with attendant increases in programming efficiency and software transferability. Modern technology in meta-compilers will be exploited to accomplish this goal. 1b1c

3.14.2.1.4 (U) Reliability Analysis Central (RAC)--The generalized data management system, DM-1, will be used to provide the RAC a computerized information capability in the first part of FY-74. 1b1d

3.14.2.1.5 (U) Generalized Data Management Specifications--This product will provide any potential Air Force user the ability to specify only those parts of a data management system that he needs to satisfy his requirements. 1b1e

3.14.2.1.6 (U) Job Performance Aids--This goal will provide the means for many staff officers to interact with each other and with a large data base on a computer to produce plans and position papers in hours instead of in weeks and months. It will be developed by using, evaluating, and modifying where necessary the Augmented Human Intellect (AHI) system developed by Stanford Research Institute under ARPA sponsorship. 1b1f

3.14.2.1.7 (U) Software First Facility--This product is a facility consisting of a very flexible computer system made up of a microprogrammable processor, assembler with a very powerful macro generation capability, micro-assembler and operating system which provides a base for computer architecture development and allows the development of software prior to hardware availability. This facility would be used to develop and evaluate techniques for the design of special purpose machines by first programming/micro-programming a given application on a flexible (reconfigurable) processing structure and using the reconfigured system as the basis for the required special purpose computer design. 1b1g

3.14.2.1.8 (U) Micro-programmable Associative Processor--This product will have all the additional features in it that have been deemed necessary by the through testing and evaluation of the Associative Processor test bed. With these features, the machine will truly bring about the long promised gains of associative processing. 1b1h

3.14.2.1.9 (U) Modular Spaceborne Computer--This product is a long life and hardened highly reliable, high-data-rate processor that functions in space. SAMSO and AFAL are primarily responsible for this product under AFAL TPO S.3. 1b1i

RADC TPO-14 DATA PROCESSING SOFTWARE

3.14.2.1.9 (U) Error Analysis Facility--This product consists of an Automated Verification System (AVS), in-house research in theorem proving, and a qualitative investigation of errors in large software systems being conducted at Syracuse University. The objective of the facility is to develop and evaluate various techniques for improving and quantifying the reliability of software. Ultimately, the goal includes developing techniques for complete certification (i.e. proof) of the correctness of programs or critical parts thereof. The AVS product package will provide the Air Force with a tool for measuring and improving the reliability of JOVIAL (J3 with embedded direct code) software on the HIS-6070. The in-house research and development in theorem proving seeks long term solutions to the problem of software reliability, while intermediate range solutions are sought by the SU analysis of errors effort. The benefits to the Air Force include increased software reliability, increased confidence in critical software items, and a reduction in manpower and computer resources used for software testing. 1b1j

3.14.2.1.11 (U) Performance Test and Evaluation Procedures--This product will improve the effectiveness of Air Force computerized data management systems by developing (1) a formalized methodology for test and evaluation of DMS software for selection of the best DMS to meet user requirements and (2) measures of performance to "fine tune" installed DMSs. 1b1k

3.14.3 RELATIONSHIP TO DESIRED CAPABILITIES or CONCEPTUAL SYSTEMS 1c

(U) This TPO will result in the "Products" or goals shown in Fig 3.14-1. The relationship between the goals and the AFSC conceptual systems will also be shown. 1c1

The Higher Order Language program is essential to the attainment of the B-1 System, (AFSC conceptual system 73), because all programming for the system will be done in the JOVIAL language. As the "standardization" technology of the HCL program will initially be developed for JOVIAL, the B-1 program will utilize the new JOVIAL specification and the compiler testing methods, and has expressed interest in the JOCIT system as this would shorten the lead time for developing a new JOVIAL compiler. 1c2

RADDC TPO-14 DATA PROCESSING SOFTWARE

It is a fact that the sophistication with which warfare can now be conducted has brought with it a digital processing load which saturates sequential computers. This problem is compounded in tactical airborne and spaceborne environments where weight, size, power and reliability are vital factors. The processing requirements mentioned above can only be met by some form of parallel processing structure. This program in computer architecture finds as its base a sequential capability and a parallel processing capability in the form of an associative processor. The associative processor has been selected as the parallel component test vehicle since of all the parallel organizations known today it provides the greatest computing power per weight, power and size and conceptually at least appears to provide the structure for obtaining a highly reliable performance. The associative processor technology is needed to attain the goals of these AFSC conceptual systems: Remotly Piloted Air to Ground Strike System (#12), Advanced Airborne Command Post (AABNCP) Growth (#35), Tactical Command and Control (Seek Digit) (#36), Airborne Warning and Control Systems (AWACS) (#38), and Air Force Ground Based Sensor System (#44). 1c3

Reliable software is essential to the successful performance of the Remotly Piloted Air to Ground System (#12). With no human intelligence aboard, critical decisions must be made by the software. The consequences of incorrect, erratic, or otherwise unreliable software, are potentially catastrophic. It is, therefore, critical that techniques be developed to measure and improve the reliability of software. 1c4

3.14.4 (U) Milestone Charts--Individual efforts within each area are summarized by a milestone chart. 1d

3.14.5 (U) TECHNICAL DISCUSSION of MILESTONE CHARTS 1e

3.14.5.1 (U) Higher Order Languages A. Efforts in this area will provide the technology in the Air Force to effectively standardize on HOLs which will meet its needs better. The updated JOVIAL language, the required HOL for all Command and Control programming, will be primarily used to develop and test the technology. This same technology will later be applied to all Air Force HOLs. This effort contains the following subtasks: 1e1

The JOVIAL Compiler Validation System (JCVS) will be delivered in FY-73 for testing JOVIAL/J3 compilers and will be extended and refined in FY-74 to validate the new Standard JOVIAL Specification completed in early FY-73. 1e1a

RADC TPO-14 DATA PROCESSING SOFTWARE

In FY-74, the new JOVIAL language will be specified in a complete and unambiguous manner through the use of the SEMANOL system completed in FY-73. In addition, the JOVIAL Compiler Implementation Tool (JOCIT) which will provide standard high quality compilers for the new JOVIAL language, will be completed. 1e1b

In addition, HOL studies, using the results of JOVIAL statistics gathering packages developed in FY-73, will be performed to analyze HOL use to provide changes to all Air Force HOLs or possibly develop a new HOL to better meet Air Force requirements, if the existing languages cannot be suitably modified. 1e1c

All efforts after FY-74 will be involved with further development of these techniques and their extension to other HOLs. 1e1d

3.14.5.2 (U) Management Information Systems B. This area has essentially two thrusts: (1) The development of nonfunctional software to provide users the ability to manipulate large data bases for Command and Control and Air Force Management systems with particular emphasis on multi-level security and data management testing procedures. (2) The development of tools, which directly aid the commander of manager, utilizing ARPA sponsored technology being developed at Stanford Research Institute (SRI) called On Line Team Interaction. RADC will be evaluating this technology extensively during FY-73 and plans to initiate prototype experimentation with DCS/PSO late in FY-74. 1e2

Generalized Data Management Specifications- RADC is completing the first implementation of a data management system DM-1 (Data Manager). This system is extremely flexible in design, is programmed in JOVIAL and its overall design contains the best features of many other systems incorporated into one system design. Two additional components, a query language and a report production system, will be implemented during FY-73. The query and report production systems are being designed for the Reliability Analysis Central (RAC) and an interim capability is expected by early FY-74. From DM-1 will come many of the items that will be in the Generalized Data Management Specifications. 1e2a

In FY-72 a mathematical model was developed for optimizing data structures in a data management environment. A continuing effort will be directed to applying this model to specific military applications. In addition a mathematical model will be developed which will identify the basic software issues of data management security. 1e2a1

RADC TPO-14 DATA PROCESSING SOFTWARE

An in-house investigation, exploring the possibilities of using the ARPA sponsored MULTICS operating system to advance the state-of-the-art in Data Management, has been completed. One aspect of this study has produced a basic, interactive data management capability in-house. The favorable results from this documented study are the basis for initiating an FY-73 MULTICS Data Management development program. 1e2a2

An in-house effort will concentrate on the development of a kernel of data management functions. These functions will include system control, storage management with special emphasis placed on the ability to provide data protection within the MULTICS environment. The objective in developing this kernel is the maximum application of operating system capabilities to the central issues of data management design in order to achieve efficient and secure data manipulation. 1e2a3

In addition, as part of Project 9339, a systems analysis of the OSD data management functions will be made. The resulting specifications will be used to develop a data management design which will provide an interim operating capability for OSD by October 1973 and a final operational capability at the end of FY-75. These capabilities will be developed by both in-house and contractual efforts. 1e2a4

On Line Team Solving Problems - Efforts in this area will result in a general purpose integrated system of hardware, software, method, procedures and training which can significantly improve the performance of individuals and teams engaged in complex problem solving and decision making tasks. 1e2b

All the efforts in this area are based on the Augmented Human Intellect (AHI) technology developed at Stanford Research Institute (SRI) under ARPA sponsorship over the past 10 years. RADC personnel have monitored this program for the past 5 years and are convinced that the feasibility of significant job performance improvement in an intellectual environment is indeed possible. SRI has developed the system through a "bootstrapping" process where the system's capabilities at any particular instance in time are used to further develop the system. The augmentation capabilities of the system have evolved through the individual, to the team, to the organization stage, and are now approaching the community stage via use by individuals and teams around the ARPANET. 1e2b1

RADC TPO-14 DATA PROCESSING SOFTWARE

The development activity during FY-72 has been concentrated on acquiring the necessary physical plant for use and evaluation of the SRI AHI system. RADC is connected to the ARPANET via a standard H316 Terminal Interface Processor (TIP). Internal lines have been acquired to connect the terminals to the TIP. Six Execuport 300 teleprinters, three IMLAC PDS-ID crt display terminals, and four Termicette cassette recorders have been purchased. Five engineers, a section chief, a secretary and an administrator have received varying amounts of training in the use of the system. The initial specification of a baseline management system complete with SOPs, file design, and personnel role assignment has been completed. The research group at RADC has progressed to the point where they are doing the bulk of their daily work using the system; in fact, this TPO was prepared, coordinated and edited a number of times using the system. The FY-73 activity will be concentrated on training the rest of the ISI branch (approx. 40 people), procuring additional terminals and a medium speed line printer, and refining the baseline management system--to include interfacing it with MASIS. The activity in FY-74 and out years will be concerned with developing and incorporating advanced management techniques such as DELPHI, resource allocation strategies, and advanced graphics capabilities. This activity will be significantly enhanced by the use of others' research work via the ARPANET. 1e2b2

The evaluation of a system as complex as AHI must be conducted over an extended period of time, an extended set of jobs, and across a representative sample of Air Force people to allow the results to be generalized to other environments. The evaluation at RADC will cover a two year period, and include engineers, secretaries, administrators, and managers at three levels in the chain of command. The evaluation will evolve slightly behind the actual implementation of the system within the ISI branch. Results on the effectiveness of the system in augmenting individuals and teams will be available near the middle of FY-73, and organizational results at the end of FY-74. A parallel implementation and evaluation will be conducted within the office of the Assistant for Automation DCS/PSO to the extent that funds will allow. The measures currently used in the evaluation include: 1e2b3

Psychometric--questionnaires, surveys, and interviews given in a controlled experimental environment. 1e2b3a

RADC TPO-14 DATA PROCESSING SOFTWARE

Performance—whenever a working group, section, or branch is charged with a specific job their performance will be compared with that of an equivalent organizational unit. Data will be collected on elapsed time, manhour time, and the subjective judgements of the quality of the work made by the two units' common manager. Comparisons will be made between the AHI system and other available systems with comparable capability in specific areas; eg. text editing. 1e2b3b

Cost/Benefit—A running systems analysis will be conducted over the two year period to enable detailed specification of the cost/benefit tradeoffs which can be made in implementing portions of the general system in other environments. 1e2b3c

During FY-73 & 74 it will be possible (based on the evaluation activity under 6.3) to specify and refine the design of a prototype system for a portion of the DCS/PEO's problems. An engineering development project will be created to allow for equipment procurement and installation. The current estimate of a minimal configuration for useful test would consist of a TIP to connect to the ARPANET, an IMP for connection to the Command Post computer or the Data Services computer, a secure line or krypto device connecting the two IMPs and a mix of 10-15 teleprinter and crt terminals. The test population within DSC/PEO should consist of a working group, its secretaries, and at least three levels of administrative and command personnel in the management chain. 1e2b4

RADC TPO-14 DATA PROCESSING SOFTWARE

DCS/PEO has been given engineering support by RADC at a low level since FY-68 under ESP project 921A-9339, "Data Handling Support for Air Staff". During FY-72 an Execuport-300 was loaned to the office of Assistant for Automation (AF/XOA) and one action officer given preliminary training in using the AHI system via the NBS TIP and the ARPANET. He in turn has given instruction to another officer in Data Services. The purpose of this activity is to familiarize a limited number of people within Air Staff with the capabilities of the AHI system. In addition to the AHI activity, RADC has leased a BR-700 Information System for a 3 month operational test within DCS/PEO. The 700 will be used by administrators to track the progress/status of some 800-1000 actions which come into the organization each month. This system is a stand-alone message composition, sending and retrieving device; which is representative of an off-the-shelf intermediate operational capability for use prior to the installation of any portion of the AHI technology.

1e2b5

3.14.5.3 (U) Architecture C. The computer architecture research and development is concentrating on evaluating associative processing and parallel processing technology on critical Air Force applications (eg. radar tracking, track-while-scan, electronic warfare). Investigation to develop a Software First Facility is also being pursued so that a special purpose machine could be economically developed based upon software requirements. In addition, this would allow the development of the software prior to actual hardware availability. A related effort is being conducted by AFAL and SAMAC to develop a modular spaceborne computer.

1e3

To accomplish these objectives, RADC is acquiring a commercially available associative processor to establish a test bed system so that hard data on the performance of this technology can be acquired. The test bed will be instrumented to make performance measurements. The test bed will be operational in the first quarter of FY-73. This facility will be used to develop the software and hardware design of an advanced Associative Processor for use in the military environment. The products that will be developed are an Associative Processor and a Software First Facility. The software first facility will be a tool for the development of the software and also the hardware design for unique, one of a kind computer systems for the Air Force.

1e4

RADDC TPO-14 DATA PROCESSING SOFTWARE

3.14.5.4 (U) Software Validation and Reliability D. During FY-72, with contractor support, a DMS Test Methodology was developed and partially validated. The primary emphasis was on characterizing a DMS, identifying existing test techniques and developing a systematic method of matching DMS features with test methods. A subset of the test methods were validated against the MADAPS DMS on the CDC 1604 computer and on the AIDS DMS on the G635. 1e5

An in-house team began testing the developmental versions of DM-1 and will continue into FY-73 to evaluate its relative effectiveness against various data bases and application scenarios. Also evaluation by developing a methodology for matching user needs with DMS assess the capability of a DMS to meet stated user requirements. Further effort will also be made in the methods for collecting and analyzing test data. 1e6

Research to date has identified simulation and modeling as a high payoff area for system evaluation. Initial studies will begin in FY-73 to define and specify a DMS simulation model for evaluation and performance measurement/prediction. This work will be done in-house and with Syracuse University to investigate theoretical aspects and new techniques. In FY-74/75, the model will be developed and exercised against a particular user problem (Joint Tech Support Activity - World Wide Command Control System). During FY-75, the overall formal DMS Performance Test Methodology will be produced including all known test techniques and including simulation and modeling. 1e7

Automated Verification System (AVS) is a set of automatic software verification aids designed to partition the program in question into testable segments and logical paths, and to collect execution time data, such as a measure of the fraction of segments that are exercised by a test case, a cumulative measure for a series of test cases, a list of segments not exercised, and data to aid in the development of test cases that will exercise specific segments. This system promises to be of great value in constructing a truly representative and logically complete set of test cases, and in providing a measure of the program reliability, i.e., (the number of segments exercised) / (the total number of program segments). Furthermore, data provided on the sequence of segment execution during a test run can aid in the identification of errors. AVS provides a technology that complements current Air Force work in structured programming, theorem-proving, higher-order languages, and automated exercises. AVS is the subject of a currently pending DEF review requiring the expenditure of approximately \$150K in FY-73, and \$200K in FY-74. 1e8

RADAC TPO-14 DATA PROCESSING SOFTWARE

In-house research and development in theorem proving (a vital part of any system that automatically proves programs correct) is aimed at generating proofs more effectively and efficiently. Substantial improvements in the efficiency of the subsumption and unification computations (important subcalculations in theorem proving) have already been obtained. As part of this effort, new associative processing techniques in theorem proving have been developed and will be implemented in an advanced system, FY-77, and evaluated. 1e9

The Error Analysis Investigation is a qualitative study of errors discovered in debugging the DM-1 system, intended to provide an evaluation of current state-of-the-art techniques for improving program reliability, as well as useful information for the long-range development of better techniques. Methods of detecting and/or avoiding these errors will be evaluated, including: 1e10

- (1) new features in the design of the programming language, 1e10a
- (2) use of structured programming, 1e10b
- (3) development of more powerful proof techniques for program correctness, 1e10c
- (4) development of techniques for avoiding non-termination. 1e10d

This work will be accomplished under the existing Syracuse University contract, and useful results incorporated in an advanced (FY-77) verification system. 1e11

3.14.6 RELATED EFFORTS

1f

Under ARPA sponsorship, Stanford Research Institute has developed a general purpose integrated on-line system of software, hardware, procedures and training which supports individuals, teams and organizations engaged in complex problem solving and decision making tasks. 1f1

The Space and Missile Systems Organization is continuing the development of the SPL Implementation Tool (SPLIT) to implement compilers for airborne systems. 1f2

The National Bureau of Standards is developing a compiler validation system for the FORTRAN language. 1f3

Mr. Warren Loper of the Naval Electronics Lab is project head to develop a HOL for utilizing parallel and associative processing for Naval applications. 1f4

RADDC TPO-14 DATA PROCESSING SOFTWARE

NASA/MSC has implemented an AVS for FORTRAN programs written for the Univac 1108 as part of the Mission Trajectory Control Program and the SKYLAB activities; the Air Force is currently testing FORTRAN programs on the IBM 360 and assembly code programs on the IBM 7090 written as part of the Minuteman program; the Army is investigating the problem of verifying its Safeguard software at Huntsville, Alabama. 1f5

Mr. Richard Bacca, AVWW, Wright-Patterson AFB had a 128 word bit/word associative device built to demonstrate the applicability of associative techniques to the signal sort and classification requirements of an advanced airborne system. 1f6

Mr. John Shore, Naval Research Laboratory, is having fabricated and tested an associative processor cell design which is intended for a number of EW applications and to be used as part of the Matrix Parallel Processor of the Advanced Avionics Digital Computer. 1f7

The FAA has been conducting tests using an experimental associative processor for air traffic control and is presently funding a benchmark program leading to the selection of an associative system for Air Traffic Control application. 1f8

The Navy Air Development Center (Robert Beesburg) is considering associative processing as part of their E2C program. They are currently investigating means of obtaining an in-house associative processing capability for test bed evaluation. 1f9

The Air Force Materials Lab (Mr. Jack Garrett) has funded Goodyear Aerospace Corp to develop manufacturing methods for a 256 X 256 Plated wire associative processor plane. 1f10

ABMDA (Advanced Ballistic Missile Defense Agency) has funded the first of a two phase program to investigate the use of an associative processor for bulk filtering of phased array radar data. 1f11

3.14.7 RELATION to TNS and RADs/PMDs 1g

This TPO is responsive to the following requirements: 1g1

ESD TN-24-69-02	Associative Processor Techniques & Applications	1g2
-----------------	---	-----

ESD TN-24-69-06	Structuring large Data Bases For Data Management	1g3
-----------------	--	-----

RADC TPO-14 DATA PROCESSING SOFTWARE

ESD TN-24-71-15	Man-Computer Communication in Interactive Systems	1g4
ESD TN-24-68-12	Auto/Machine Aided Content Analysis of Textual Data for Reference and Fact Retrieval	1g5
ESD TN-24-72-17	Security Controls in Multi-Users Computer Systems	1g6
SAMSO TN-24-71-103	Ballistic Missile Computer Software	1g7
SAMSO TN-24-71-104 Techniques	Software Verification-Validation	1g8
SAMSO TN-24-71-107	Modular Operating Systems for Ground Computers	1g9
SAMSO TN-24-71-108	Computer Selection Simulator	1g10
SAMSO TN-24-71-110	Software Test Estimation and Control	1g11
SAMSO TN-24-71-111	Control of Software Development by Networking Techniques	1g12
PMD P-2P-031(1)	Advanced Computer Techniques	1g13
3.14.8 RESEARCH NEEDS		1h

Basic research is needed in the area of formalization of code generation for HOL compilers. This is the only area of a compiler which has not been effectively formalized in a "cook-book" like fashion. That is, while formal techniques exist for building code generators, the code they produce does not compare in efficiency to "hand-coded" code generators. 1h1

For large file applications the following needs exist: 1h2

- a. Automatic multi-level systems to control the movement of files within the storage hierarchy in order to prevent a high priority storage device from being overcrowded with inactive files. 1h2a

RADDC TPO-14 DATA PROCESSING SOFTWARE

b. A file processing system which permits the integration of all user dates into a data base and controls the accessing of the data base by multiple user programs. In order to achieve this type of processing, it is required that a job in multiprogramming must be able to communicate with another job to effect a synchronous processing action. Studies should be pursued to determine the modifications to conventional operating systems required to obtain this type of processing.

1h2b

c. Sophisticated hardware and software tools are required to measure the performance of complex computer software handling large file applications. The high cost of overhead software in current systems dictates a need for formal evaluation techniques.

1h2c

The advent of computers capable of supporting several users in a time shared or multiprogrammed mode has greatly increased the difficulty of safeguarding classified information processed by such systems, particularly when classified and unclassified information must be processed simultaneously. Technology is not currently available to meet Air Force requirements for a multi-level, open computer system.

1h3

Basic research is required to develop the mathematics and processing techniques necessary to produce formal proofs of correctness of software. Investigations should be pursued in the areas of theorem proving, higher order logics, developing better languages for defining correctness and generating proofs and developing more effective deductive systems.

1h4

Almost no statistics are currently available on the frequency, context and type of errors which often disable large software systems. There is a fundamental need for such measures if we are ever to quantify software reliability and improve its uncertain performance.

1h5

Research is required in the associative processing area as follows:

1h6

a. Algorithms and strategies which take advantage of the content addressable and parallel processing capabilities of an associative processor as applies to signal processing, data management, display processing, and tracking.

1h6a

b. Languages suitable for the effective application of a sequential computer-associative processor complex to the above problem types.

1h6b

RADC TPO-14 DATA PROCESSING SOFTWARE

c. Operating system strategies which effect high utilization of a sequential computer-associative processor complex.	1h6c
3.14.9 TEST REQUIREMENTS	1i
None	1i1
3.14.10 TPO FOCAL POINT	1j
Frank J. Tomaini Rome Air Development Center Griffiss AFB NY 13440 Autovon 587-7507	1j1

RESOURCE SUMMARY--MANPOWER

PROG ELEMENT/PROJECT	FY72	FY73	FY74	FY75	FY76	FY77	FY78
62702F/5581	37.1	42.6	30.8	30.2	28.2	29.2	28.0
62702F/2801	(3.0)	(2.0)	(2.0)	(2.0)	(2.0)	(1.0)	(1.0)
62702F/3176	(1.3)	(3.3)	(3.3)	(6.3)	(1.3)	(1.3)	(1.3)
TOTAL 6.2	37.1	42.6	30.8	30.2	28.2	29.2	28.0
63728F/5550	6.6	11.3	24.6	24.3	20.5	17.5	16.2
63706F/683J	0.1	0.2	0.2	0.2			
63710F/481B	0.1	0.5	0.5	0.5	0.5	0.5	
TOTAL 6.3	6.8	12.0	25.3	25.0	21.0	18.0	16.2
64718F/441A		0.5	0.5	0.1			
64708F/1266	0.5	0.5	0.5	0.3			
64701F/428A	0.2	0.2	0.2	0.4			
333125F/1144	0.3	0.3	0.3				
61101D/0967	0.5	0.5	0.5	0.5	0.5	0.5	0.5
921A/9339	0.5	0.5	0.5	0.5	0.5	0.5	
DCA/JTSA			0.5	0.5	0.5		
TOTAL OTHER	2.0	3.0	3.0	2.3	1.0	1.0	0.5
TPG TOTALS	45.9	57.6	59.1	57.5	50.2	48.2	45.2

RESOURCE SUMMARY--DOLLARS

P.E./PROJ	FY72	FY73	FY74	FY75	FY76	FY77	FY78	CUSTOMER
62702F/5581	2143	3165	2915	2465	2090	2000	2000	AFSC
62702F/2801	268	150	240	240	250	100	100	AFSC
62702F/3176	270	250	250	250	250	250	250	AFSC
TOTAL 6.2	2681	3565	3405	2955	2590	2350	2350	
63728F/5550	1900	1000	1475	1400	2830	2540	1850	AFSC
			(1300)	(1100)				
61101D/0967	1330	1330	1330	1400				ARPA
921A/9339		150	250	400				HQ USAF
64XXXX B-1	20			9000	8000	500	1000	B-1 SPO
DCA/JTSA		165	90	90				DCA
635706F/683J	25							ESD
TOTAL OTHER	3275	2615	3115	12290	10830	2950	2850	

RADC TPO-14 DATA PROCESSING SOFTWARE

(J10819) 23-JUN-72 14:10; Author(s): Duane L. Stone/DLS;
Distribution: Duane L. Stone, James H. Bair, Thomas F. Lawrence, James
C. Norton, Paul Rech, Dirk H. van Nouhuys, John L. McNamara, Thomas J.
Barcalow/RBMS JLM TJB; Sub-Collections: RADC RBMS; Clerk: DLS;
Origin: <STONE>TPO.NLS;1, 13-JUN-72 6:00 DLS ;

pass 0

There are several interesting ideas for additional features to be added to the Telnet protocol which could be very useful. These features are generally independent of each other and could be treated as options to be implemented as desired.

1

Telnet Escape

2

This is a Telnet control signal which indicates that the receiving Telnet should parse the immediately following bytes as site specific control information. It is typical that user-Telnet programs have many commands available for the human user to control the nature of the local terminal action. It could be useful for the server-Telnet or the serving program to also have access to these commands. This could be accomplished by permitting the server site to send the Telnet Escape signal followed by a command in the user-Telnet's command syntax. In the case that the data following the Telnet Escape signal is unparseable it should be printed as if the Telnet Escape did not occur.

2a

Telnet Timing Marks

3

Two Telnet control signals are used, Time-Query and Time-Reply. Time-Query is inserted into the network data stream by either the user-Telnet or the server-Telnet. Upon receiving a Time-Query signal a Telnet responds by placing a Time-Reply signal in its outgoing network data stream. This would enable a Telnet to determine that the output buffers at the other end of the connection are empty. It would also permit the Telnets to estimate the round trip time if that were of use.

3a

Clear Buffers Orders

4

It could be useful for one Telnet to tell the other to clear its buffers. This would be useful when a change of state is taking place, for example a runaway process has produced a great deal of output and been killed, rather than force the user to see all the garbage it would be nice if the buffer could be cleared.

4a

Transmission Controls

5

One Telnet might like to tell the other its preferred transmission strategy. This could be one of the following:

- Transmit on "break" characters
- Transmit every "n"th character
- Transmit every "n" milliseconds

pass 0

Transmit now.

5a

Format Effects

6

The issue of Tab settings, page fold skipping, line folding, are of interest.

6a

End of Line signal

7

The suggestion that a single specis signal be used to indicate end-of-line has been made many times and perhaps could be implemented.

7a

Format Interpreters

8

Pre-Processors

9

pass 0

(J10820) 23-JUN-72 15:10; Title: Author(s): Jonathan B. Postel/JBP;
Distribution: John W. McConnell/JWM; Sub-Collections: NIC; Clerk: JBP;
Origin: <UCLA-NMC>ORTHOGNAL.NLS;2, 23-JUN-72 12:31 JBP ;

Please change RAG's affiliation to MITRE. His address is the same as that of PMK et al. His extension there is 2380, and he is now a civillian, rather than a captain.

1

Please change VRB's and SAAC's addresses to be the same as mine.

2

thanx, Buz.

3

ADO 23-JUN-72 20:16 10821

(J10821) 23-JUN-72 20:16; Title: Author(s): A. D. (Buz) Owen/ADO;
Distribution: Cindy Page/CXP; Sub-Collections: NIC; Clerk: ADO;

Information about the SAAC TIP

(tips)Information about SAAC-TIP:

1

Reply to NIC/RFC #352

1a

To:

1b

David Crocker
c/o Jon Postel
Computer Science Department
School of Engineering and Applied Science
3804 Boelter Hall
University of California
Los Angeles, California 90024

1b1

Information about the SAAC TIP

Name of organization & pseudo-host number: 1c

Name: The Seismic Array Analysis Center 1c1

Site name: SAAC 1c2

Network address of TIP pseudo host: 154 decimal. 1c3

Preferred host name: SAAC-TIP 1c4

Regular Host computer(s) associated with the same organization: 1d

None 1d1

Mailing address of organization: 1e

The Seismic Array Analysis Center
 c/o Teledyne-Geotech
 Box 334 (813 North Royal Street)
 Alexandria, Virginia 22314 1e1

Local person responsible for TIP operations: 1f

Name: 1f1

A. D. (Buz) Owen 1f1a

Telephone: 1f2

703/836-3882 extension 278 1f2a

Alternate person(s): 1f3

Information about the SAAC TIP

Vernon R. Bruffey (extension 276)	1f3a
Operations staff of computer center	1g
Name(s):	1g1
Mark S. Goldman, operations supervisor	1g1a
Dan E. Kuykendall, 1st shift supervisor	1g1b
Michael S. Telman, 2nd shift supervisor	1g1c
Steve Bronson, 3rd shift supervisor	1g1d
Hours of operator coverage:	1g2
Full time.	1g2a
Telephone:	1g3
8am-5pm e.t.: 703/836-3882 extension 231	1g3a
5pm-8am e.t.: 703/836-7646	1g3b
Thru the NET:	1g4
None, except for TIP's tty, which is normally turned off.	1g4a
TIP options & special features:	1h
None.	1h1

Information about the SAAC TIP

Scheduled down-time: 1i

Preventive Maintenance: 8 am - 11 am E.T., first Thursday
of every month. 1ii

Additional comments: 1j

Ports: 1k

TIP address Terminal-type or dial-up phone number:
(port num.)

1k1

2 ASR 33 Teletype. 1k2

1l

ADO 23-JUN-72 20:21 10822

Information about the SAAC TIP

(J10822) 23-JUN-72 20:21; Title: Author(s): A. D. (Buz) Owen/ADO;
Distribution: David H. Crocker/DHC; Sub-Collections: NIC; Clerk: ADO;
Origin: <MITRE-TIP>ADO2.NLS;2, 22-JUN-72 16:54 ADO ;

Request for documents

Memo:

To: Cindy Page

Date: 23 June '72

Subject: Document Request

From: Buz Owen.

By mistake, in sending NIC 10698, I managed to send you the title of my file instead of it's contents. Here is another try.

Please send me copies of the following NIC documents:
8208 8142 7979 5772 8748 8719 8382 8353 8194 5775(NWG/RFC 105)
and all of 8123 through 8158

Thanks.

1

1a

1b

1c

1d

2

3

4

Request for documents

(J10823) 23-JUN-72 21:48; Title: Author(s): A. D. (Buz) Owen/ADO;
Distribution: Cindy Page/CXP; Sub-Collections: NIC; Obsoletes
Document(s): 10698; Clerk: ADO;
Origin: <MITRE-TIP>AD01.NLS;3, 23-JUN-72 21:43 ADO ;

ITS UP

SPEAKEASY is up under TSO at UCLA contact me or BRADEN about it
in fact why dont you send me a message thru NIC when you get
this.

1

ITS UP

(J10824) 26-JUN-72 15:49; Title: Author(s): Stanley Cohen/SC;
Distribution: James E. White/JEW; Sub-Collections: NIC; Clerk: SC;

This is a message to you to see if I know how to do it without help.

IT is fun but it is very hard to learn....
dont you think so?

1

(J10825) 26-JUN-72 15:53; Title: Author(s): Stanley Cohen/SC;
Distribution: Arnold M. Ostebee/AMO; Sub-Collections: NIC; Clerk: SC;