HGL 29-JAN-75 10:22 25240

ISI Request for Space in NLS File Structure

24-JAN-75 1940-PST TUGENDER at USC-ISIB: Space required within NLS files
Distribution: LEHTMAN AT SRI-ARC, arc/c:
Received at: 24-JAN-75 19:42:18 1

Harvey, 1a

Sorry to take so long in getting back to you, but I've been busy doing a bunch of other stuff and haven't been able to give the matter much more thought. Essentially, what we need is some breathing room within the NLS structure which gives us the capability
to add various kinds of flags and markers to NLS files and statements
which we could access more efficiently in the overhead structure than by adding special statements or data. We know what some of these will be, but I'm sure that several more will crop up as we iterate over the details of the IA files. I do not know what constraints you are dealing with in the implementation of the new file structure, so I'm not sure how much leeway we have in picking what we need. 1b

I suggest the following will probably be sufficient for our needs: 1c

2 words in the file header block
1 word in the ring element
1 word in the data block 1d

Should the above prove too much, they can probably be all cut in half and still be useful for us. 1e

At present, I have no documentation on what the expanded file structure will provide for users. I remember your talking about LISPish property lists, but a little more detail would be quite helpful, as would an expected implementation timetable. For example,
if the new file system is operational within a few months (say by May), it is possible that we could employ the new structure to store binary information within messages instead of having to maintain all message information in text form. 1f

I hope this is enough for you to proceed with your implementation
plans. Please let me know if there is any other info you would like from me. 1g

Regards,
Ron 1h

ISI Request for Space in NLS File Structure

ISI Space in files 2

Ron-- 2a

We are currently inplementing the new file system. It seems clear from your description that your requirements can be covered by putting the binary data you wish stored in special property data blocks and then have appropriate decoding and interpretive procedures. 2b

There is probably space available in the file header for you as well as an available word in the ring, though space in the data block headers is tight (again, a new property type may negate the need for some of these things.) 2c

We are implementing the file system now and expect it to be finished by 15 February, well before the time you said you need it. We will be using it for statement text and graphics entities at first. (Implementaton of a header entity has been delayed.) It would be good if other properites were also tried out, and I think your use of the system would be a good test of its flexibility. 2d

Harvey 2e

HGL 29-JAN-75 10:22 25240

ISI Request for Space in NLS File Structure

(J25240) 29-JAN-75 10:22;;;; Title: Author(s): Harvey G. Lehtman/HGL; Distribution: /RWW([INFO-ONLY]) NPG([INFO-ONLY]) ; Sub-Collections: SRI-ARC NPG; Clerk: HGL;

POOH 29-JAN-75 11:12 25241

Useroptions' Show Controlcharacters for terinals 35-tty and 37-tty

When using this command, these two terminals are listed as first level command words. There exists already 33-tty which is a first level command word. What I am trying to say is that you can't input 35-tty or 37-tty as part of the command. Thanks pooh 1

POOH 29-JAN-75 11:12 25241

Useroptions' Show Controlcharacters for terinals 35-tty and 37-tty

(J25241) 29-JAN-75 11:12;;;; Title: Author(s): Ann Weinberg/POOH;
Distribution: /FEED([ACTION]) ; Sub-Collections: SRI-ARC; Clerk:
POOH;

POOH 29-JAN-75 11:55 25242

a ranom bug in Useroptions' Show Controlcharacters command

Unfortunatey this bug does not appear all the time. This happened at ARC when I used this command and specified all for terminals, I got the error message: Nls internal error-String too long. When this happened, the terminals Execuport, Ti, Nyt, and 33-tty gave me the information I requested and then filled up the entire screen with the words Null. Other times when I have used this command, it has worked fine. I can't really say what I did differently. Thanks pooh 1

POOH 29-JAN-75 11:55 25242

a ranom bug in Useroptions' show Controlcharacters command

(J25242) 29-JAN-75 11:55;;;; Title: Author(s): Ann Weinberg/POOH;
Distribution: /FEED([ACTION]) ; Sub-Collections: SRI-ARC; Clerk:
POOH;

RLL 29-JAN-75 12:04 25243

References for KWAC agenda.

Here are a few references of interest to KWAC meeting. See (24167,) for suggested agenda. The suggestion (24632,) is too late to arrange although some informal discussion on real applications will be included during the week with invited guests from SRI attending.
Robert 1

JCN 28-JAN-75 05:23 25218 KWAC Meeting announcement
KWAC Meeting III in February at ARC
Location: (HJOURNAL, 25218, 1:w)
*****Note: [ACTION] *****
2

RLL 27-NOV-74 17:22 24632 sug: conference during KWAC meeting
conference during KWAC meeting for invited guests
Location: (GJOURNAL, 24632, 1:w)
*****Note: Author Copy*****
3

Comments: Such a conference should be well planned and guests invited well ahead of time. Also, speakers should be solicited and their talk planned some time in advanced. 3a

RLL 4-OCT-74 11:30 24167 Minutes from KWAC meeting
Minutes from KWAC meeting 12 SEPT 74
Location: (JJOURNAL, 24167, 1:w)
*****Note: Author Copy*****
4

RLL 29-JAN-75 12:04 25243

References for KWAC agenda.

(J25243) 29-JAN-75 12:04;;;; Title: Author(s): Robert N. Lieberman/RLL; Distribution: /FGB([ACTION]) ; Keywords: kwac KWAC; Sub-Collections: SRI-ARC; Clerk: RLL;

JAKE 29-JAN-75 17:14 25244

Site visit from Kjell Samuelson

Kjell Samuelson, Professor of Informatics from the University of Stokholm may or may not come by tomorrow or Feb. 4th. He is interested in some areas in which he might collaborate with NIC and ARC with major interests in indexing, data base problems and general information retrieval problems. He will also want to visit others so may not be available for long. He has expressed interest in helping with the NIC Newsletter, in teleconferencing and in user interaction with large data bases. He has recently come onto the Arpanet via a dedicated line to the LONDON host. Anyone interested in talking to him please let me know. My apologies for not reporting this earlier...it was mentioned so long ago that I almost forgot and just found a memo to myself.

JAKE 29-JAN-75 17:14 25244

Site visit from Kjell Samuelson

(J25244) 29-JAN-75 17:14;;;; Title: Author(s): Elizabeth J. (Jake) Feinler/JAKE; Distribution: /SRI-ARC([INFO-ONLY]) ; Sub-Collections: SRI-ARC; Clerk: JAKE;

DVN 29-JAN-75 17:49 25245

Passing the Buck on NVT

On the run in New YOrk,..I forwarded you request for information about the NVT to John Postel who has been working in that area. 1

DVN 29-JAN-75 17:49 25245

Passing the Buck on NVT

(J25245) 29-JAN-75 17:49;;;; Title: Author(s): Dirk H. Van Nouhuys/DVN; Distribution: /JOAN([ACTION] dpcs notebook please) LAC([INFO-ONLY]) JBP([INFO-ONLY]) ; Sub-Collections: SRI-ARC DPCS; Clerk: DVN;

DVN 29-JAN-75 17:55 25246

Editorial Processing Center Proposal Progress

Here writing to you from the Village via the Rutgars TIP

DVN 29-JAN-75 17:55 25246

Editorial Processing Center Proposal Progress

Sorry to take so long in answering, I've been a bit out of touch. 1

The comercial packet switching situation is well coverd in Bob Lieberman's item < gjournal, 25058, > You can get more scuttle but from him 2

In principal NLS can run on any TENEX with little effort. It has run on BBN's and ISI's aat vaious time. In principal NLS can be converted to run on any system...some notion have been formed of how much work it would be to convert to given other systems. I have heard figures like 6-12 person months. Talk to Dave Maynard or Charles Irby form more information. 3

About rpicing you have to talk with Jim Norton. He is the only in a position to tal about that. 4

For your information and TOm'S, Revens at ACM was very positive. He is most interested in Computing Reviews" because of the headaches thay have with keeping the files on reviewers straight and beacues the printing contract is just turning over. I left him mumbling sommthing about "being saturated with hope". He is drafting a letter for us. He had not gotten Tom's by the way. I am to go back tomorrow, demostrate NLS and talk to his boss. I will be back wesst Thursday evening and see you Fri. 5

DVN 29-JAN-75 17:55 25246

Editorial Processing Center Proposal Progress

(J25246) 29-JAN-75 17:55;;;; Title: Author(s): Dirk H. Van Nouhuys/DVN; Distribution: /JOAN([ACTION] dpcs notbook and please take a paper copy to Tom Humphry) PWO([INFO-ONLY]) TLH([INFO-ONLY]) RWW([INFO-ONLY]) DCE([INFO-ONLY]) JCN([INFO-ONLY]) ; Sub-Collections: DPCS SRI-ARC; Clerk: DVN; Origin: < VANNOUHUYS, PARTANSWER,NLS;1, >, 29-JAN-75 17:45 DVN ;;;;####;

This is available in output processed form as
<NSW-Sources>NSWV2CHANGES.PRT for SRI-ARCs Line Printer and
as <NLS>NSWV2CHANGES.TXT for access by network users.

INTRODUCTION 1

This document records the ways in which the NSW implementation is diverging from its Version 2 documentation. It is a dynamic document of primary interest to implementers of NSW and code which must run in a NSW environment. It is organized by V2 document so that each section can, if desired, be physically stored with the document to which it corresponds. 1a

Comments, corrections, and additions are welcomed. The contents of this document will be used eventually to generate Version 3 documentation. In the interim, the most recent copy will be available on-line in the following forms: 1b

[SRI-ARC]<POSTEL>NSWV2CHANGES.NLS 1b1
[SRI-ARC]<NLS>NSWV2CHANGES.TXT 1b2

The former is an NLS file, the latter an output-processed version suitable for printing on a non-SRI-ARC printer. 1c

HOST 2

CLARIFICATIONS 2a

TYPOS 2b

BUGS 2c

(HOST -- 24581,5) Retransmission Policy 2c1

It is clear that if the sender of data sends multiple messages on a connection, using the four bit sequence numbers in the message identification field to match the RFNMs to the sent messages, and a message results in an Incomplete Transmission; then there is no way for the sender to be sure that the remaining outstanding messages and the retransmission of the undelivered message will be processed in order by the reciever. 2c1a

CHANGES 2d

(HOST -- 24581,5a1) Retransmission Policy 2d1

K set to 5 is more reasonable. 2d1a

IDEAS 2e

(HOST -- 24581,5) Retransmission Policy 2e1

The use of the sequence number bits in the message identifier could be used to detect if the host at the other end of the connection knows the new stuff or not. That is if the other host sends sequence numbers then it knows the new stuff so this host can use the new stuff. 2e1a

In particular suppose we wish to use sequence numbers and multi-outstanding messages on a connection to achieve higher bandwidth. If we start out by sending messages using the sequence numbers but only one at a time (that is RFNM limited) and observe whether or not the messages sent by the other guy have sequence numbers. If they do then we can assume that he knows about sequence numbers and can reassemble the messages into the proper order if there is a loss of order by an incomplete transmission. 2e1a1

The retransmissions could be timed in a non linear maner. That is they could be transmitted as follows: 2e1b

first retry -- at once 2e1b1

second retry -- after 1 second 2e1b2

third retry -- after 30 seconds 2e1b3

fourth retry -- after 1 minute 2e1b4

fifth retry -- after 2 minutes. 2e1b5

forget it 2e1b6

BJP 6

CLARIFICATIONS 6a

(BJP -- 24583,3) Introduction 6a1

It should be realized that not only are traditional batch processing facilities to be utilized via the Batch Job Package, but that some interactive systems have capabilities whch can usefully be interfaced to a Batch Job Package. In this latter catagory are the TENEX RUNFIL and the Multics execcom capabilities. 6a1a

TYPOS 6b

BUGS 6c

CHANGES 6d

(BJP -- 24583,4a4) Add Terminal Streamnames 6d1

TTYIN: the job's primary terminal input stream 6d1a

TTYOUT: the job's primary terminal output stream 6d1b

IDEAS 6e

RJE-MODEL 9

CLARIFICATIONS 9a

(RJE-MODEL -- 24655,3) Expansion of the Model 9a1

Expansion of the Model to cover all types of Batch Jobs. 9a1a

There are two types of batch jobs: those that we typically think of as job entry or remote job entry jobs, and those that are run on interactive systems using input from a file rather than a human at a keyboard. 9a1b

TYPOS 9b

BUGS 9c

CHANGES 9d

(RJE-MODEL -- 24655,4) The Model 9d1

The text of this section is replaced by the following: 9d1a

Here is a scenario of use of a batch tool which is an elaboration of the discussion contained in the RJE-MODEL document. 9d1b

Note that there are two cases for batch jobs in the NSW: one is the traditional batch processing facility which normally expects as its primary input a control card file; the other is an interactive time sharing system which allows input to come from a file instead of interactively from a user at a terminal. 9d1b1

MODEL 9d1c

First we discuss the entities involved in the process of composing a batch job, having it run, and examining the results. 9d1c1

The principal entity is a batch processing facility. This is expected to be an existing hardware & software unit that will be only minimally changed to interface to the NSW.

Examples of batch job proocessing facilities are the B4700 and the IBM 360.

Another type of batch job capability is is the TENEX runfil or the Multics execom facility.

The NSW talks to the batch processing facility via a procedure package called the Batch Job Package (BJP).

The batch job package in a sense referees the flow of information between its PCP callers and the batch processing facility. For example the batch job package collects all the input files that are resident on other hosts before turning the job over to the batch processing facility, and the batch job package may distribute the result files to other hosts when the job is completed by the batch processing facility.

The Batch Job Package interacts with File Packages (FP) to effect the movement of files to and from the Batch Processing Facility.

The call on the batch job package to get a job submitted to a batch processing facility is:

CRTJOB (infiles, outfiles -> jobid)

The files referenced in infiles and outfiles are named so that the batch job package can get them from and put them into the directories owned by NSW at various hosts and manipulated by file packages. Thus these files are named by "file-package-filenames".

The user sees only NSW-filenames so there must be a language/grammar that controls the users interaction which results in the generation of a create job call on a batch job package. This processing for the user must include the mediation of the NSW-filenames the user supplies into the file-package-filenames included in the create job call.

The files themselves are created and examined using the text editors (e.g. NLS) available in the NSW.

Some files that are included in a create job call may be standard library files and from the users point of view part of the system. The user may not even be aware of their existence since their names could be supplied by the grammar internally.

The input files are probably in most cases job control files in a particular batch processing facility's specific job control language. There might be grammars/tools to aid the user in constructing such control files for specific batch processing facilities and applications programs.

A scenario for a user creating, submitting, retrieving, and examining a batch job follows: 9d1c2

The user interacts with the front end. The front end contains a command language interpreter that is driven by a grammar. The particular grammar in use for this user at any time depends on which tool the user is accessing.

The user interacts with an editing tool to create a source program and to concatenate it with a standard file of job control information particular to the Batch Processing Facility to which it will be submitted. The concatenation is accomplished using regular editing commands (not batch specific commands).

The user then interacts with the Works Manager and the Batch Job Package mediated by a grammar to submit the file he has created. The grammar and the Batch Job Package will require enough information from the user that the Batch Job Package can retrieve the input files from File Packages, and store the output files. The Batch Job Package will return an identifier for this job which can be used to request status information at a later time.

Some of the information needed to run a batch job could be in a standard file that the user always appends his file to, OR this type of information could be in a separate file that is included by the grammar in the create job call automatically, and the grammar could call on a function to edit a standard file to contain user and run specific parameters such as user-name, priority, run-time-limit.

When the job has been processed the user may use an editing tool to examine the output file. Note that the output files have been stored as specified in File Packages and are thus accessible to tools as permitted by the Works manager.

It may be necessary to construct special tools to reformat the output of other tools for presentation on the users terminal.

In particular the tools which were designed to ouput to line printers will produce output difficult to view adaquately on narrower display and teletype terminals.

This is a general problem so we should seek out a general solution.

A discussion of a batch program as a tool. 9d1c3

An applications program which lives on a batch processing facility can be made into a tool in the NSW such that the users of it as a tool do not need to know the control language of the facility where it lives. To do this the tool installer must create a control card file and a grammar which are stored in the Works Manager under the toolname assigned to this program.

When the user accesses the tool the front end gets the grammar from the works manager and follwes it to collect the prameters from the user. Once all the arguments are collected the front end (or the works manager) can call the batch job package. Note that one of the arguments is the name of the control card file. This argument may be built in to the grammar or supplied by the works manager.

A discussion of multi-host batch jobs. 9d1c4

Suppose a user wanted to run a series of batch jobs steps where each step was to be carried out on a different host. It is not difficult to envision a NSW-control-language in which one could say things like:

> "If the previous job step was successful then use its output file WALDO appended to control file DOITTOIT as card input to the batch processing facility ABC and call the printer output file GEORGE".

This requires a tool to "execute" files of this NSW-control-langauge to be written.

This NSW-control-language need not be different form the language the user normally uses in interactive work, in fact it should be identical except for the addition of conditional statements to continue or abort the processing of a multi-step job.

IDEAS 9e

TBH 10

CLARIFICATIONS 10a

(TBH -- 24656,1b2a) System to Process Interface 10a1

The TBH Supervisor has shrunk to be a very small thing. All that is required in that role is a "loader" that can accept a filename and load that file as the code of the PCP process created by the initial connection to the PCP contact socket. 10a1a

TYPOS 10b

BUGS 10c

CHANGES 10d

(TBH -- 24656,1b2a) System to Process Interface 10d1

Socket number 25 (31 octal) has been assigned as the PCP contact socket. 10d1a

IDEAS 10e

NVTP 11

CLARIFICATIONS 11a

(NVTP -- 24827,2) Introduction 11a1

The following discussion is added: 11a1a

Protocol Design Approaches 11a1a1

Several ARPANET applications (i.e. third- and fourth-level) protocols have been designed and implemented since the Host-Host Protocol was adopted in 1970. Most have been bootstrapped from lower-level applications protocols. For example, the File Transfer Protocol (FTP) was built upon TELNET, and the Remote Job Entry Protocol (RJE) upon both TELNET and FTP. The highest-level protocol shared by all such bootstrapped protocols is TELNET.

Although the bootstrapping principle seems a sound basis for Network protocol development, we believe that TELNET, providing little more of use than a character set, is NOT the most appropriate foundation for a large class of applications protocols.

Bootstrapping at a Higher Level 11a1a2

We contend that a Procedure Call Protocol (PCP) -- a Network-standard mechanism for invoking arbitrary named, argument-driven and result-producing procedures in a remote process -- is a much more appropriate and powerful foundation for many applications protocols. We believe that the adoption by the Network community of a PCP as the basis for most applications protocols would have at least the following effects:

1) expedite the specification of applications protocols by permitting their documentation to have a functional, rather than a syntactic orientation,

2) largely eliminate the need for separate, application-specific user processes,

3) reduce the cost of making large, existing software systems available as Network servers by allowing a Network interface more compatible with their internal organization,

4) provide the basis for a more natural interface between local and remote procedures, and therefore

5) encourage the sharing of software, by making procedures on remote hosts as accessible to the programmer as local ones.

This is an attempt to clarify the role of the Network Virtual Terminal Package (NVTP) in interfacing "Old Programs" to the National Software Works (NSW). The discussion here assumes that the reader is familiar with the Procedure Call Protocol (PCP) and the Telnet protocol. 11a1a3

The NSW is composed of two principal entities and a group of auxiliary entities. The principals are a Works Manager (WM) and a Front END (FE). The auxiliaries are called Tool Bearing Hosts (TBHs). 11a1a4

The WM and the FE always communicate with each other and with the TBHs using PCP. This is a simplifying principle that allows for a cleaner and quicker implementation of the WM and the FE. 11a1a5

(We note that at times the same machine that supports the FE may be used in a non-NSW context to communicate with other machines, including those that support the WM or those that are also TBHs, using other protocols. This does not alter our basic simplifying principle since those other communications protocols and programs are completely independent and outside of the NSW.)

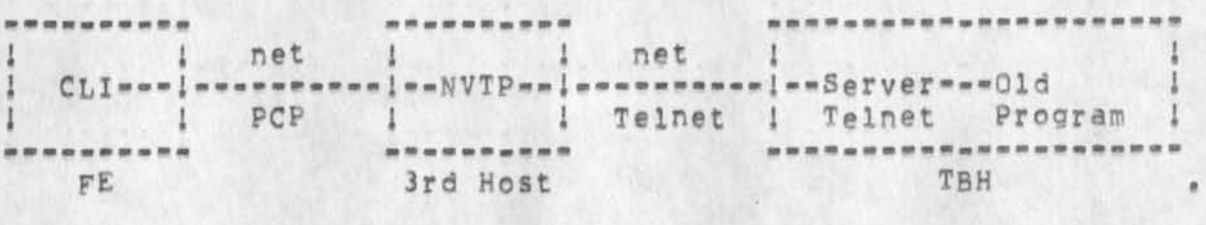
The active agent in the FE that carries out the users requests as interpreted using the Grammar and the user profile is the Command Language Interpreter (CLI). 11a1a6

A Tool is the collection of: a set of computational procedures (sometimes caled a backend or BE), a Grammar, a Help Data Base, and Tool specific parts of a User Profile. 11a1a7

The TBHs support applications programs (e.g. text editors, compilers, reformatters, ...) called tool backends. These tools are or will be constructed with the NSW in mind, and will expect to communicate via PCP. Other applications programs, here called "Old Programs", were constructed to communicate only with a controlling teletype. The Telnet protocol has been designed and implemented such that a remote user's terminal can appear to be the controlling teletype when the remote user utilizes a "user Telnet" process to communicate via the network with a "server Telnet" process that directly controls the application program. 11a1a8

To interface such Old Programs into the NSW a NVTP has been designed to act as a converter between PCP and Telnet protocol. There are two cases to be distinguished: first the case where the NVTP is in a third host, and second where the NVTP is directly controlling the Old Program. Note that in either case from the point of view of the WM and the FE the NVTP is the tool backend. 11a1a9

Case 1 11a1a10



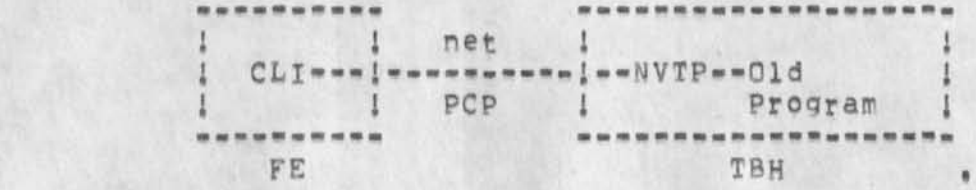
Notes:

CLI only does PCP calls.

NVTP merely copies data.

Server Telnet acts as controlling teletype to Old Program.

Case 2 11a1a11



Notes:

CLI only does PCP calls.

NVTP acts as controlling teletype to Old Program.

NVTP is a SMALL extension of Server Telnet program.

In either case there is a need for a Grammar, a Help Data Base, and a Tool specific User Profile. 11a1a12

In the NSW environment the contol features of Telnet are generally unnecessary since these functions are performed by the FE. 11a1a13

Most of Telnet's control options are for controlling aspects of the user's interaction that can be specified by a Grammar or User Profile. The difference is that in Telnet the parameters are dynamically controlled and transmitted between the user and server for each use of a program; in the NSW Procedure Call Protocol these parameters are incorporated in the Grammar and are therefore relatively static, but they are not renegotiated with each use of the tool and thus there is less network traffic. A user should be able to change aspects of the interaction by commands to the FE which do not require network traffic.

For example the most powerful Telnet Option -- Remote Controlled Transmission and Echoing (RCTE) -- is completely replaced by a Grammar tailored to the serving host and tool. And the strategy of dynamic control used in RCTE requires substantially more network traffic than is neccessary in the NSW case.

An alternative communication strategy for Old Programs has been suggested that would have the FE communicate with the Old Program using Telnet protocol. 11a1a14

The WM is the only NSW process that initiates tool processses and the WM always communicates using PCP. It would be quite awkward to have the tool process initiated using PCP and subsequently communicate using Telnet protocol. The Telnet protocol does have a reconnection option (there is only one known implementation of this feature), so that (in theory) control of a process created by a Telnet initiation by the WM could be switched to the FE, such a procedure requires both the WM and the FE to treat Old Programs differently than new tools, and requires both the WM and the FE to implement both PCP and Telnet protocol.

TYPOS 11b

BUGS 11c

CHANGES 11d

IDEAS 11e

NSW Inter-Version (2-3) Documentation
NSWV2CHANGES

29 JAN 75

Jon Postel
Augmentation Research Center

Stanford Research Institute
Menlo Park, California 94025

NSWV2CHANGES records the ways in which the implementation of NSW is diverging from its Version 2 documentation, and should be viewed by implementers of both NSW and code which must run in a NSW environment as a dynamic appendix to the Version 2 PCP-related documents. The reader is of course assumed familiar with the Procedure Call Protocol (PCP -- 24459,).

(J25247) 29-JAN-75 18:16;;;; Title: Author(s): Jonathan B. Postel/JBP; Distribution: /NPG([INFO-ONLY]) SRI-ARC([INFO-ONLY]) NSW([INFO-ONLY]) ; Sub-Collections: SRI-ARC NPG NSW; Clerk: JBP; Origin: < POSTEL, NSWV2CHANGES.NLS;10, >, 29-JAN-75 18:08 JBP ;;;; ####;

RLL 30-JAN-75 00:15 25249

Samuelson visit to ARC (Univ. of Stockholm)

yes, I would like to see/talk to Samuelson, Robert, Thanks 1

RLL 30-JAN-75 00:15 25249

Samuelson visit to ARC (Univ. of Stockholm)

(J25249) 30-JAN-75 00:15;;;; Title: Author(s): Robert N. Lieberman/RLL; Distribution: /JAKE([ACTION]) ; Sub-Collections: SRI-ARC; Clerk: RLL;

POOH 30-JAN-75 07:54 25250

This is a retest.

Sorry to bother you again, but I received two copies of the same item yesterday--each with a different journal number. i am sending this one without sending myself one and just see what ends up in my author branch. I will be in touch...pooh 1

------- 2

POOH 30-JAN-75 07:54 25250

This is a retest.

(J25250) 30-JAN-75 07:54; Title: Author(s): Ann Weinberg/POOH;
Distribution: /KIRK; Sub-Collections: SRI-ARC; Clerk: POOH;

CHI 29-JAN-75 21:47 25251

nsw scenarios for feb review meeting

Dick, if this looks ok, we should send it to other nsw contractors.

CHI 29-JAN-75 21:47 25251

nsw scenarios for feb review meeting

The following scenarios (Journal # 25248) are in response to Larry Crain's memo announcing the Feb NSW review meeting. I have numbered the points I am addressing according to the numbers in Larry's original memo. 1

3A) LOGIN AND LOGOUT 2

This hasn't changed enough from my earlier scenario (Journal # 24534) to warrent much discussion. When a user types some character on an unused terminal, the FE collects project, username and password and calls login procedure in WM (actual call here). The WM returns user-id, user profile for fe-interaction, and list of tools available to this user. User is then talking to nsw-exec grammar with commands to manipulate whole files, perform terminal-specific operations, get acounting information, logout, etc. In addittion the user always has available (no matter which tool he is runing) the universal commands to run tools, terminate tools, get status of active tools, get semantic help with tools or the nSw as a whole. The number of commands in the universal set should be kept small to avoid undue restrictions on other tool command languages. 2a

3B) INVOKING, USING, AND LEAVING THE TELNET-ELF TOOL 3

a) using ELF outside NSW 3a

There will probably be a command in the nsw-exec that allows the user to leave the nsw FE and use the normal ELF exec. Once this is done, the user is on his own until he returns to the NSW FE. 3a1

The user will not be able to reference NSW files by their NSW names. He will not be able to talk too the WM or NSW tools. 3a1a

b) using a non-integrated tool 3b

The NSW will allow users to use tools that are not fully integrated into the NSW. These tools will be accessed either a) through a common tool grammar that knows nothing of the behavior or intended function of the tool or b) through a tool grammar that has been tailored somewhat for that tool. 3b1

In case (a) the user will type strings to the tool and it will respond, with the FE doing all echoing. This will be much like operating a line-at-a-time terminal. There will be no commands given to the tool in the normal NSW sense of command words and parameters. The user will be able to get very little help from the FE for this type of tool since it has only one command

which is just the collection of a literal string from the user, but he will have the universal commands available to him. 3b2

This could be implemented in at least the following different ways: 3b2a

1) the command consists only a request to the FE that it collect a character string from the user and send it to the tool. in this case, the user has the option of typing a universal command or typing text for the tool. There may be conflicts where what he wishes to type to the tool also specifies a universal command. In these conflicts, the universal commands would get control. 3b2a1

2) the command consists of a command word, such as "SEND" followed by the collection of a string to send to the tool. Here the conflict with universal commands is more clearly understood by the user but he must precede each string he wishes to send to the tool by the command word. Note that the command word could be as simple as a SPACE character. 3b2a2

3) The command could be as in 1 but the user must hit some escape character to specify universal commands. 3b2a3

4) the command consists of a command word, say "SEND" again, followed by infinitely many literal strings collected from the user and sent to the tool. This requires the user to say "send" followed by an arbitrary number of strings, each of which is sent to the tool. If he wishes to execute a universal command then he must hit the abort-command key first. 3b2a4

In case (b) above, the tool grammar will contain commands tailored to the function of the tool and will appear to be more like an integrated tool. 3b3

In both of these cases the NVT package will be used to drive the actual tool through telnet. The only difference is in the commands that are available to the user. In both cases the user may reference NSW files and may slue to other tools from the un-integrated one. 3b4

The use of file names (where the FE does not know it is a file name) requires that the tool's attempt to access the file be trapped and that the file be moved to the local host by the WM. 3b4a

3D) CALLING, USING, AND LEAVING NLS 4

It should be understood that NLS like NSW represents a system for accessing a number of different tools. Thus, within the NSW the various tools contained in NLS will be tools in the NSW. There will be no single NLS tool. There will be an editor, a calculator, a send-mail, a user-profile tool, and perhaps other tools. 4a

The universal command for running a tool is used to specify the desired tool, say the editor. 4b

There is a tool naming issue here. We should not, for example, use up all of the obvious good names just because we are adding the first few tools. We propose that the user or his project leader supply the simple name which he will use and that this be translated into a unique system-wide name for the tool. Thus the user may ask to run the "editor" and for him that translates into "NLS-EDITOR." For another user, "editor" might mean some other editor tool. 4b1

when the user logs into the nsw, the FE fetches from the WM a list of the tools this user is allowed to access. This list could consist of (simple name, system name) pairs. 4b2

When the user issues the run-tool command he may type ? to find out which tools he may run. When the user specifies which tool is too be run, the FE calls the WM, passing it the (system) name of the tool and gets back the tool-id for this tool. If the grammar for the tool is already in the fe, then it is not reloaded. Otherwise, the fe calls the wm with the toolid and gets back the grammar for the tool. 4c

we could implement this in such a way that the fe keeps track of tools used and does not bother to call the wm if this user has previously in this seession run this tool. 4c1

The fe then inspects the grammar to determine which pcp processes must be created to support this tool. For each such process the wm is called to create it and introduce it to the fe. The fe opens the appropriate pachages and allows the user to specify commands to the tool. 4d

While the tool is being used, various procedures in the processes are called to carry out the semantics of the commands. 4e

While the user is using the tool, he may give a universal command such as run another tool or terminate the current tool. Ih he elects to run another tool without first terminating the current tool, the fe simply switches grammars and holds any output from the old tool. the user may later terminate the new tool and thus

resume the old tool or e may give the resume command for the old tool. when this happens, the fe switches back to the original grammar. 4f

When the user terminates a tool, the wm is called to delete the processes that support this tool and the grammar's use count is decreased by one. if the use count is zero, then no user is using that tool and the core occupied by the grammar can be reclaimed if needed. 4g

3F) EXPLICIT (USER DIRECTED) FILE MOVEMENT INTO, OUT OF, AND WITHIN NSW 5

this is accomplished via the nsw-exec' move/copy/delete file commands. for copying files into and out of the nsw, the user must supply the necessary information to allow the file to be properly moved and typed. 5a

3G) HELP FEATURES 6

This is accomplished via a universal command and keys on the user's terminal. 6a

Keys: 6a1

?: The user may type ? whenever specifying a command (except in the middle of literal text of course). The FE responds with a list of current alternatives. 6a1a

SYNTAX: The user may type this to learn the full syntax of a command or part of a command. 6a1b

HELP: The user may type this at any point in specifying a command to obtain semantic, functional help with the command, the tool containing the command, or with basic concepts in the nsw as a whole. 6a1c

This is simply another way of accessing the semantic help facilities as described below. 6a1c1

Command: 6a2

The "HELP" command is in the universal commands and is thus available while using any tool. It allows the user to specify a concept or command or a tool, etc, and attempts to provide the user with useful explanations thereof. The data base for this semantic help facility will be structured nls-editor files for first-year nsw. There will be one or more such files associated (by the wm or a declaration in

the cml grammar) with each tool plus one or more containing overall nsw concepts, lists of available tools, and guidelines for installing tools and tool help data bases. We are publishing guidelines for building such data bases. 6a2a

The process that interprets the structured data base and presents help to the user will be an instance of the nls-editor process, created at login time by the wm at the fe's request. When the user first requests semantic help this process is called with the name of the data base for the current tool. It obtains this file(s) plus the nsw-help file(s) from the wm and attempts to help the user. On subsequent invokations of the help facility, no new files will have to be obtained from the wm unless the user has switched tools. 6a2b

It will be difficult for a user to find out detailed things about tools other than his current tool. Only an overview of other tools will be available to him. 6a2b1

3I) ESCAPING TO THE WM AND RETURNING TO A TOOL 7

Escaping to the WM amounts to running the NSW-EXEC. This "tool" is always immediattely available (the grammar is always in the sattelite machine and the WM process is always available). In addition, there will be an escape character which will allow the user to slue to the NSW-EXEC. Once there the user may if he wishes suspend the current tool (in the middle of execution ala control-c in tenex). We invision a "resume" command to be used to resume such a suspended tool when the user wishes this to happen. If the tool being resumed was not suspended, but rather the user merely slued (via the escape-to-nsw-exec key, a "resume", or a "run" command) to another tool and is now sluing back, any output that was waiting for the user from the tool is now presented to him. 7a

Following is a first pass at the set of universal commands and the commands in the nsw-exec: 7b

universal commands 7b1

run tool 7b1a

("GOTO" <tool-name> <confirm>) 7b1a1

terminate current tool 7b1b

("QUIT" <confirm>) 7b1b1

nsw scenarios for feb review meeting

logout 7b1c

("LOGOUT"!L2! <confirm>) 7b1c1

resume tool 7b1d

("RESUME"!L2! <tool-instance-name> <confirm>) 7b1d1

execute command in another active tool 7b1e

("EXECUTE"!L2! <tool-instance-name> <command>) 7b1e1

comment 7b1f

(";" <text> <confirm>) 7b1f1

semantic help 7b1g

("HELP" <optional-item-list> <confirm>) or 7b1g1

("<help>") 7b1g2

show current commands 7b1h

("?") 7b1h1

show syntax of commands 7b1i

("<syntax>") 7b1i1

nsw-exec commands 7b2

rename file 7b2a

copy file 7b2b

delete file 7b2c

show 7b2d

accounting info 7b2d1

status of active tool(s) 7b2d2

list of files 7b2d3

start/stop recording session (typescript) 7b2e

playback session 7b2f

connect/disconnect terminals	7b2g
simulate terminal type	7b2h
set/reset tty window position and size (display terminal only)	7b2i
scroll back tty window (display only)	7b2j

CHI 29-JAN-75 21:47 25251

nsw scenarios for feb review meeting

(J25251) 29-JAN-75 21:47;;;; Title: Author(s): Charles H. Irby/CHI;
Distribution: /RWW([INFO-ONLY]) NPG([INFO-ONLY]) ;
Sub-Collections: SRI-ARC NPG; Clerk: CHI; Origin: < IRBY,
SCENARIOS-NSW-MEETING.NLS;1, >, 29-JAN-75 21:45 CHI ;;;;####;

KIRK 29-JAN-75 22:42 25252

Items for discussion at the meeting 1:30 Thursday

Please see <kelley, scenarios,> <documentation, helpd,> and <kelley, filename,>. See you there. 1

KIRK 29-JAN-75 22:42 25252

Items for discussion at the meeting 1:30 Thursday

(J25252) 29-JAN-75 22:42;;;; Title: Author(s): Kirk E. Kelley/KIRK;
Distribution: /CHI([ACTION]) RWW([ACTION]) EKM([ACTION])
DSM([ACTION]) JBP([ACTION]) JEW([ACTION]) HGL([ACTION])
KLM([ACTION]) ; Sub-Collections: SRI-ARC; Clerk: KIRK;

JBP 30-JAN-75 09:19 25253

Nsw Scenarios for discussion today

In response to Dick Watson's request here are my coments on three items from the list of meeting topics suggested by Larry Crain in his message of 20 JAN 75. 1

Scenarios 2

3c Creating Batch Job 2a

This is covered in the NSWV2CHANGES file under the RJE-MODEL section. 2a1

3h Invoking a TBH (TENEX, Multics, ?OS360/370) 2b

It is difficult for me to write a scenario about this since it violates my model of the NSW. The thing i think is implied here is starting a tool. It might mean starting a tool that is the interactive executive. 2b1

This should be no different than starting any other tool so the scenario should be the same as 3d (Calling using and leaving NLS). 2b2

3j passing messages in NSW (not NLS Journal or Netmail) 2c

This will not happen. The only mechanism for user to exchange arbitrary text messages will be a mail tool either based on sndmsg or the journal (most likely sndmsg) with some interaction with a Works Manager maintained data base like an "Ident file". 2c1

JBP 30-JAN-75 09:19 25253

Nsw Scenarios for discussion today

(J25253) 30-JAN-75 09:19;;;; Title: Author(s): Jonathan B. Postel/JBP; Distribution: /NPG([INFO-ONLY]) JAKE([INFO-ONLY]) ; Sub-Collections: SRI-ARC NPG; Clerk: JBP;

JAKE 30-JAN-75 10:19 25254

Hardcopy Journal

Adrian has brought the filing of the hardcopy journal reasonably up to date and as of now the NIC can no longer continue to handle this chore. There are currently at least two boxes a week coming in and it is taking too much time away from other committments. My understanding is that JCN will make some arrangements to continue with the filing so that it does not get completly back-logged again. Anyway, for the moment, enjoy!. 1

JAKE 30-JAN-75 10:19 25254

Hardcopy Journal

(J25254) 30-JAN-75 10:19;;;; Title: Author(s): Elizabeth J. (Jake) Feinler/JAKE; Distribution: /SRI-ARC([INFO-ONLY]) ; Sub-Collections: SRI-ARC; Clerk: JAKE;

SLJ 30-JAN-75 11:18 25255

acoustic coupler

hi there cleptos: in the process of finding one of the two anderson-jacobsen couplers (the brown square wood ones) the other one which was in the closet disappeared. we have to return them. someone cop to it pleeze. 1

SLJ 30-JAN-75 11:18 25255

acoustic coupler

(J25255) 30-JAN-75 11:18;;;; Title: Author(s): Sandy L. Johnson/SLJ; Distribution: /SRI-ARC([ACTION]) ; Sub-Collections: SRI-ARC; Clerk: SLJ;

DCE 30-JAN-75 11:44 25256

An ARC IR&D Proposal: AKW 'Technology Transfer Techniques'

Was sent by Division Office to Henry Blanchard, Asst. to V.P. of ORD

An ARC IR&D Proposal: AKW "Technology Transfer Techniques" Program 1

Principal Investigator: James C Norton 1a

Basic/Applied: Applied 1b

Expenditures: 1975 (Proposed) 1c

Labor: $75K 1c1

Other: $80K (Computer services and terminal lease, for two ssots and work stations devoted to this practice, coaching, materials development, etc.) 1c2

Total: $155K 1c3

Key words: Technology Transfer, Online Information Systems, Augmentation 1d

Potential client relationship: 1e

X DoD 1e1

X NASA 1e2

X Other government agency (almost any) 1e3

X Other client category (almost any) 1e4

OBJECTIVE 1f

The problem is to introduce into external institutions radical innovations representative of the way in which their future knowledge work will be done. Over the past ten years of concentrated, single-purpose work, involving over $10 million of govenment R&D support, ARC has developed an extensive, coherent system of tools and techniques to support an Augmented Knowledge Workshop, incorporating computer and communication tools so advanced that their replication within an application environment would represent extreme barriers in cost, qualified implementation personnel, qualified applications coaching and training personnel, suitably knowledgeable and oriented decision makers, etc. 1f1

Without first-hand, real-work experience with such tools, potential users are unable to perceive the possibilities, problems and potential value; they could not provide adequate justification for the very high implementation costs of acquiring computer and communication support

facilities within their own organization, and they would either have to accept a plan for new worker organization, methods, procedures, etc, as designed for them by a outsider or base their plans on untutored surmise of inexperienced internal staff. 1f1a

Over the past five years ARC has planned for an experimental, relatively large-scale program aimed toward transferring this type of technology into external institutions. For the past year we have operated a pilot service that external clients can subscribe to, where the computer-communiations tools are brought into the clients' home sites via special communication channels from a central computer facility operated by a commercial time-sharing company under contract to us. We now have about $920K/yr subscription business, and this is expected to increase perhaps three-fold during the coming year. Many government agencies are perceiving direct value from the service, and the scope of exploratory applications is rapidly expanding. 1f2

In the first year of service, we have learned quite a bit about the human support component of the service required by a client in order to facilitate the introduction of these tools into their exploratory application operation, and to build awareness, perspective of applicability potential, to overcome inappropriate fears, prejudices and expectations, to evolve skills and knowledge toward increased knowledge-worker effectiveness, to learn how to provide sensible cost/payoff exploration plans for their management, to choose personnel and exploratory applications sensibly, etc. 1f3

The importance of these human support services is so high that our further stages of this technology-transfer experiment are planned to include a significant increase in the staff of transfer specialists. These specialists require a kind of training and experience that we as yet don't know how to provide quickly and effectively -- we've depended so far on using staff that have already acquired experience in developing and using these tools and techniques, and that have an aptitude for dealing with people. We have to depend in the future upon hiring and training new people for these transfer-facilitation roles. 1f4

The objective of this IR&D Program is thus threefold: 1f5

1) Develop ARC technology-transfer staff to the point where they can directly contribute to supporting the transfer process into our client institutions. 1f5a

2) Develop the materials and methods that regularize the training, so that it will be possible for instance to transfer this training capability into a client organization. 1f5b

3) Ready our capability to run courses aimed at training AKW users or AKW-user trainers. 1f5c

APPROACH 1g

Develop a training curriculum, with appropriate materials, practice exercises, testing procedures, etc. (requiring time of current, experienced staff, plus work-station access) 1g1

After screening and recruiting staff with appropriate backgrounds and aptitudes, this program would support the experienced-staff trainers and the new-staff trainees. 1g2

The opportunity seems to exist for offering one-shot, fixed-price training courses as a significant component of this technology-transfer process. If our experience in the early stages of this program indicate its feasibility, we would prepare one or more trial courses. 1g3

FUTURE PLANS 1h

To work toward the objectives and approach outlined above during 1975. 1h1

DCE 30-JAN-75 11:44 25256

An ARC IR&D Proposal: AKW 'Technology Transfer Techniques'

(J25256) 30-JAN-75 11:44;;;; Title: Author(s): Douglas C. Engelbart/DCE; Distribution: /JCN([INFO-ONLY]) DCE([INFO-ONLY]) JML([INFO-ONLY]) ; Sub-Collections: SRI-ARC; Clerk: JML; Origin: < ENGELBART, IRD,NLS;5, >, 30-JAN-75 09:44 JML ;;;;####;

JAKE 1-FEB-75 21:25 25268

PHONE LOG: Dr. Edw. Schelonka, Los Alamos Sci. Lab.

Ed Schelonka, formerly of RML, called for some documentation and mentioned that his Lab., Los Alamos Scientific Lab., P. O. 1663, Los Alamos N. Mex. 87544, will be bringing a CDC 6600 onto the net via the AFWL-TIP at Kirtland AFB (VDH). They will be using MACSYMA at MIT and programs at UTAH as well as interacting with the other AEC hosts, Argonne, NYU, Brookhaven, and LBL. They have some researchers interested in CAD (referred them to Ingvar Lundh). Ed mentioned the Milton Rose in Washington, D.C. is the AEC policy maker for networking and is apparently co-ordinating the hook-up of all the AEC hosts. Also on Kirtland will be NMRO and ASL as well as Kirtland...only TIP with 3 hosts as on a TIP (2 are VDH). 1

JAKE 1-FEB-75 21:25 25268

PHONE LOG: Dr. Edw. Schelonka, Los Alamos Sci. Lab.

(J25268) 1-FEB-75 21:25;;;; Title: Author(s): Elizabeth J. (Jake) Feinler/JAKE; Distribution: /SRI-ARC([INFO-ONLY]) ; Sub-Collections: SRI-ARC; Clerk: JAKE;

JAKE 1-FEB-75 21:32 25269

Kwac agenda

I would like to have a fairly involved session where each of us outlines what we have been doing with NLS,..that is, building data bases, preparing documents, information retrieval, office procedures, etc. It would be interesting to learn of new applications, approaches, programs, or whatever. A kind of show-and-tell-and-share session. Object would be to see if there are areas in which various users could collaborate to produce general tools useful to many users, and also just to find out what everyone is doing that is interesting. 1

JAKE 1-FEB-75 21:32 25269

Kwac agenda

(J25269) 1-FEB-75 21:32;;;; Title: Author(s): Elizabeth J. (Jake) Feinler/JAKE; Distribution: /KWAC([ACTION]) ; Sub-Collections: SRI-ARC KWAC; Clerk: JAKE;

JAKE 1-FEB-75 21:37 25270

Promotional Brochure for Office-1

I have had several people ask me for some general information on Office-1. Also I frequently get requests from new network users for Net information in general. It would be very useful if we had a sexy looking promotional brochure to send out. (A fancier version of Jim's 24031 maybe). Anyway just thought I would make the suggestion for consideration...may already be on the drawing board. 1

JAKE 1-FEB-75 21:37 25270

Promotional Brochure for Office-1

(J25270) 1-FEB-75 21:37;;;; Title: Author(s): Elizabeth J. (Jake) Feinler/JAKE; Distribution: /DCE([ACTION]) RA3Y([ACTION]) JCN([ACTION]) RLL([ACTION]) JHB([ACTION]) ; Sub-Collections: SRI-ARC; Clerk: JAKE;

JBP 1-FEB-75 22:14 25271

SDC2's address

The Ident File has the wrong address for Steve Crocker (SDC2), he now is at ISI,
--jon,

1

JBP 1-FEB-75 22:14 25271

SDC2's address

(J25271) 1-FEB-75 22:14;;;; Title: Author(s): Jonathan B. Postel/JBP; Distribution: /FEED([INFO-ONLY]) JAKE([INFO-ONLY]) SGR([INFO-ONLY]) ; Sub-Collections: SRI-ARC; Clerk: JBP;

&SRI-ARC 2-FEB-75 14:42 25275

SRI/ARC BASIC TNLS-8 COURSE

This is a revised version of the first course in NLS, designed by ARC to be minimally complex and yet contain the commands necessary to enable a user to enter, edit, and 'mail' text. It is intended to be used by ARC trainers in all beginning courses although Architects are welcome if they have to provide their own training. Completion time ranges from 1/2 to 2 days. Printed copies are available from User Development.

SRI/ARC BASIC TNLS-8 COURSE

TNLS SYLABUS

THE BASIC TNLS-8 COURSE OUTLINE

INTRODUCTION TO NLS

NLS = oN Line System

INLS = Typewriter Version

CAPABILITIES OF SYSTEM:

Composing

Editing

Studying

Structuring

Browsing - viewing

Printing

Publishing

Communicating -

sending and receiving mail, messages, documents; teleconferencing; etc.

Storing and retrieving -

record keeping, library services, data bases, searching, etc.

Calculating

SOME NOTES

**This is designed for use when terminals are available for all participants to use/view easily. It is intended to be the first course a person receives on NLS. The commands are shown as they would appear with partial prompting.

COURSE ORGANIZATION

The course is organized by concepts of what a user can do with

TNLS at this level. The seven concepts (listed below) are ordered as one would need them to use the system. Under each concept are the exact commands that instruct the computer to perform the function that goes with the concept. There is a command summary at the end of the course outline that lists the same commands alphabetically for easy reference.

The commands which are included in this first course have been selected to let a user write, edit, store, and communicate typewritten information (text). Those commands numbered with a (2) are to be covered on the second day of the course.

TNLS CONCEPTS: (Things you can do as covered in this course:)

1. FILES FOR STORAGE
2. TYPING IN TEXT
3. TYPING OUT TEXT
4. EDITING
5. COMMUNICATING
6. ADDRESSING
7. TROUBLE SHOOTING AND HELP

DEFINITIONS FOR THE COURSE OUTLINE

COMMANDS: You typing some characters to tell the computer what to do. <> means type a SPACE.

WHERE YOU ARE: Where the computer thinks you are pointing to (to some character in some file); you tell it by specifying an address; this is where your command will be done.

CTRL = hold down the control (CTRL) key while typing the specified character.

Upper case characters in a TNLS command phrase are what you type.

<esc> = the ESC or escape key on your terminal.

BASE C: = the TNLS ready signal. It means that you can type in an editing command (like home base...).

SEND C: = the Sendmail subsystem ready signal. It means that you can type in a Sendmail command.

GETTING TO NLS

THE TERMINAL AND USE (if dialing in)

If dialing in, turn on, dial TIP number, place receiver in cradle after hearing tone, and make sure terminal is online.

NETWORK (if used)

Net login [-- see "Network Login Procedure" for differences]

Brief Summary for dial in users: [Steps are numbered]

[I] Type e [to get the Network's attention]

[II] Type LOGIN then give your name and password when asked.

[III] Type QUIT CR [after logging in to Net]

[IV] Type @ L SPACE 43 CR [to make connection to Office-1, Host 43]

You now should be connected to TENEX

TENEX Executive

Login procedure: :

[V] type your USERNAME SPACE PASSWORD SPACE CR [SPACE fills in account number automatically; you're then ready to call NLS]

(2) Group allocation quota:
gro<esc>UPSTAT [to see who else is using your slot]

Calling NLS:

[VI] Type NLS CR

(2) To get back to TENEX :

type CTRL C [under most conditions you may continue in NLS by
typing CON CR]

To leave the system, logout in NLS:

type SPACE Logout CR

BASIC TNLS:

Abort Commands = CTRL x (kills the command before the final CR!)

OK: prompts you for a carriage return.

1. FILES

The origin statement (no number, contains the file name - don't edit)

The initials file -- automatically your first file, named after your initials (also your mail box)

New files

BASE C: <>CReate C: File T: FILENAME CR
(FILEOWNER, FILENAME.NLS;1,)

(any short "word", like a folder label)

(2) To see a list of all your files: Show Directory:

<>SHow Directory (of) OK: CR OPT/OK CR

(2) To work in another file:

BASE C: Load C: File T: FILENAME CR

2. TYPING IN "TEXT"

Insert Statement (ADDRESS = statement number)

BASE C: Insert C: Statement (to follow) A: ADDRESS CR L: CR T: TYPEIN CR

(TEXT or TYPEIN means you type in whatever you want in the file)

(2) Continue to insert = CTRL e

(Puts you in the "Enter mode" until type a CTRL x -- you type CR to end each statement until done)

backspace character = CTRL a
backspace word = CTRL w

(2)Insert Text at the end of a statement

(2)BASE C: Insert C: Text (to follow) A: +e T: TYPEIN CR

(+e means the end of the statement)

3. TYPING OUT "TEXI" *To stop printing type a CTRL o

(takes some time to get through!!)

Printing the file:

BASE C: Print C: File OK: CR

Print Statement:

BASE C: Print C: Statement at A: ADDRESS CR V: CR

Print the rest of the file:

BASE C: Print C: Rest OK: CR

(2) Easy print = \ (prints the statement where you are)

4. EDITING

To change text that has been typed in:

Substitute Text in Statement (to correct most errors):

BASE C: Substitute C: Text (in) C: Statement (at) A:
ADDRESS CR
(New TEXT) T: TYPEIN CR
(Old TEXT) T: TYPEIN CR Finished? Y/N: Y [for yes]
Substitutions made: NUMBER

(replaces the old text with the new text every time it finds it in the statement.)

Delete Statement

BASE C: Delete C: Statement at A: ADDRESS OK: CR

Delete File

BASE C: Delete C: File T: FILENAME OK: CR

(Careful, this removes the file. You can Undelete a File before Logout.)

Update: (do periodically for backup, not imperative)

BASE C: Update C: File OK/C: CR
(FILEOWNER, FILENAME,NLS;2,)

(2) Move Statement:

BASE C: Move C: Statement (from) A: ADDRESS CR (to follow) A: ADDRESS CR L: CR

(2) Copy Statement:

BASE C: Copy C: Statement (from) A: ADDRESS CR (to follow) A: ADDRESS CR L: CR

(2) formatting technique:

To insert a carriage return, type CTRL v CR

5. COMMUNICATING

(2) SENDMAIL SYSTEM:

(2) Submit message using idents (or ,receivername) and Interrogate (where the system prompts you):

BASE C: Goto subsystem C: Sendmail OK: CR

SEND C: Interrogate OK: CR

(distribute for action to:) T: CHI,FEEDBACK,SGR CR

(distribute for information-only to:) T: JCN

(title:) T: Example CR

(type of source:) C: Message T: TYPEIN CR

(show status?) Y/N: CR [the status typed by the system:]

TITLE: Example
AUTHOR(S): JHB
DISTRIBUTE FOR ACTION TO: chi feedback sgr
DISTRIBUTE FOR INFO-ONLY TO: jcn
MESSAGE: (Typein of message will be repeated.)

(send the mail now?) Y/N: Y [for yes]

Completed

SEND C: Quit OK:/C:

(2) to send a statement use the following instead of Message (See the Command Summary for example)

<>Statement

(2) to send a file use the following instead of Message (See the Command Summary for example)

File

(2) The mail box is in your initials file under a statement called "(Journal)"

(2) Print Journal

BASE C: Print C: Journal (mail) OK: CR

(2) Empty mail box: substitute (read) for (journal)...

TENEX ways: SNDMSG and LINK

To send a Message (Tenex)

First: [You cannot log out from this Tenex, must QUIT]:

Goto Tenex CR

SND CR [The system will prompt you:]
(to:) [lastnames separated by comma]
(cc:) [lastnames separated by comma]
(subject:) [TYPEIN of your message]
(message:) TYPEIN
CRTL z
CR [to terminate and send the message]

To continue where you were in TNLS:

QUit CR

Linking (Tenex) [first ask where the person is:]

WHERE SPACE USERNAME CR [do not link when user is in SNDMSG, OUTPRC, NOUTPRC, or XLIST]

LINK SPACE USERNAME CR [precede comment with ; repeat every 3 lines]

6. ADDRESSING

Where the pointer is -- type a / (see the questionmark key on some terminals). This will show an arrow pointing to the character that you are at: ==>x

Addressing within files (to move the pointer)

Jump to new address (to change where your pointer is)

BASE C: Jump (to) C: Address A: ADDRESS CR

OR you can type in an address anytime you see the prompt A: An address can be:

statement number (NOTE: TNLS automatically renumbers statements when appropriate)

.t ("tail") for the last statement in the file

(2) "TEXT" To find some word or text

BASE C: Jump (to) C: Address A: "TEXT" CR

(enclose in quotes whatever word or series of characters you want to find -- takes you to the first occurrence of it that occurs to the right and down in your file)

(2) statement number "TEXT" To find some word or text in a particular statement

BASE C: Jump (to) C: Address A: statement number "TEXT" CR

Addressing across files and directories

Jump to Link

BASE C: Jump (to) C: Link T: (FILEOWNER,FILENAME,)CR

(2) Link can be: (FILEOWNER,FILENAME,STATEMENT NUMBER) OR (FILENAME,STATEMENT NUMBER)

(You do not have to type in the parentheses)

7. TROUBLE SHOOTING AND HELP

Immediate:

Type ? for a list of all the possible command words.

Type CTRL Q for help concerning what you are doing or type H for the Help command (after Help you can type any word in NLS you wish to know about). CTRL x gets you out of Help and back to where you were.

call SRI-ARC, (415 326-6200, ext.3630)
or Link to Bair, Roetter, Beck or Feedback at Office-1

(2) FEEDBACK mechanism:

SNDMSG or Sendmail to FEEDBACK
response should be no later than 1 working day.

(2) Status commands

CTRL t [note words RUNNING or WAIT -- system should be either running or waiting for you]

(2) Remedies

CTRL c, reset, NLS

Update File Compact

PRACTICE

In addition to trying each command, there is a Primer designed to be used for practice.

Note: The next course is called, "Introduction to Structure and Viewing", and explains many of the things skipped in this course.

TNLS COMMAND SUMMARY FOR THIS COURSE: (alphabetical) You type that part of the command that appears in capitals. CR = Carriage Return.

BACKSPACE CHARACTER = CTRL a ; BACKSPACE WORD = CTRL w

CARRIAGE RETURN (formatting) = CTRL v CR

CONTINUE TO INSERT = CTRL e instead of first CR (CTRL x to stop inserting)

COPY STATEMENT

Copy C: Statement (from) A: ADDRESS CR (to follow) A: ADDRESS CR L: CR

CREATE FILE

<>CReate C: File T: FILENAME CR

DELETE STATEMENT:

Delete C: Statement at A: ADDRESS OK: CR

DELETE FILE:

Delete C: File T: FILENAME OK: CR

INSERT STATEMENT:

Insert C: Statement to follow A: T: TYPEIN CR

INSERT TEXT at the end of a statement

Insert C: Text to follow A: +e T: TYPEIN CR

JUMP TO ADDRESS:

Jump (to) C: Address A: "TEXT" CR

Jump (to) C: Address A: statement number "TEXT" CR

JUMP TO LINK:

Jump (to) C: Link T: (FILEOWNER,FILENAME,)CR

LINK:

(FILEOWNER,FILENAME,STATEMENT NUMBER) or (FILENAME,STATEMENT NUMBER)

LOAD FILE:

Load C: File T: FILENAME CR

MOVE STATEMENT:

Move C: Statement (from) A: ADDRESS CR (to follow) A: ADDRESS
CR L: CR

PRINT STATEMENT:

Print C: Statement at A: ADDRESS CR V: CR

Easy print = \

PRINT REST:

Print C: Rest OK: CR

Stop printing = CTRL o

PRINT FILE:

Print C: File OK: CR

SHOW DIRECTORY:

<>SHow Directory (of) OK: CR OPT/OK CR

SUBSTITUTE TEXT IN STATEMENT:

Substitute C: Text in C: Statement at A: ADDRESS CR
(New TEXT) T: TYPEIN CR
(Old TEXT) T: TYPEIN CR Finished? Y/N: OK: CR
Substitutions made: NUMBER

TAIL = .t for ADDRESS

(the last statement in the file -- when single level)

UPDATE A FILE:

Update C: File OK/C: CR

SENDMAIL SYSTEM:

Submit Message or Statement or File, idents (or .receivername),
and Interrogate:

Goto subsystem C: Sendmail OK: CR

SEND C: Interrogate OK: CR

(distribute for action to:) T: CHI FEED JCN CR

(distribute for information-only to:) T: RWW

(title:)T: Example CR

(type of source:) C: Message T: TYPEIN CR

OR,,type of source:) C: STRUCTURE A: ADDRESS CR

OR,,type of source:) C: File T: FILENAME CR

(show status?) Quit OK:/C: (the status typed by the system:)

TITLE: Example

AUTHOR(S): JHB

SEND FOR ACTION TO: chi feed jcn

SEND FOR INFO-ONLY TO: rww

MESSAGE: Typein of message.

(Send the mail?) C: Y/N: (CR for yes)

Completed

(2) Print Journal

Print C: Journal mail OK: CR

(2) Empty mail box: substitute (read) for (journal)...

BASIC TNLS-8 COURSE

SRI-ARC

2 FEB 75

Augmentation Research Center

STANFORD RESEARCH INSTITUTE
MENLO PARK, CALIFORNIA 94025

&SRI-ARC 2-FEB-75 14:42 25275

This is a revised version of the first course in NLS, designed by ARC to be minimally complex and yet contain the commands necessary to enable a user to enter, edit, and "mail" text. It is intended to be used by ARC trainers in all beginning courses although Architects are welcome if they have to provide their own training. Completion time ranges from 1/2 to 2 days. Printed copies are available from User Development.

SRI/ARC BASIC TNLS-8 COURSE

(J25275) 2-FEB-75 14:42;;;; Title: Author(s): Augmentation Research Center /&SRI-ARC; Distribution: /JMB([ACTION]) KWAC([INFO-ONLY]) SRI-ARC([INFO-ONLY]) UD([INFO-ONLY]) ; Sub-Collections: NIC KWAC SRI-ARC UD; Obsoletes Document(s): 24207; Clerk: JHB; Origin: < BAIR, BASICCOURSE.NLS;7, >, 2-FEB-75 14:12 JHB ;;;;(BAIR, NEWCOURSE.NLS;3,), 1-JUL-74 09:10 JHB ;####;

JHB 2-FEB-75 23:05 25278

SRI/ARC TNLS COURSE 2: INTRODUCTION TO STRUCTURE AND VIEWING

This is the first formal publication in the Journal of the second course. It is being released for JMB 's use this week in WashDC, and has been used by JHB and SGR in earlier versions. It is created from a master course file by filtering out unrelated levels. Comments welcome especially before release to KWAC et. al. Jim

JHB 2-FEB-75 23:05 25278

SRI/ARC TNLS COURSE 2: INTRODUCTION TO STRUCTURE AND VIEWING

< BAIR, COURSE.NLS;38, >, 28-JAN-75 17:29 JHB
;;;;<BAIR>COURSE.NLS;18, 3-JUL-74 09:33 JHB ; (jhb,) \

%Filters at end; 1 = basic course ***PRINT ONLY WITH NAMES OFF***;%

(1)INTRODUCTION TO TNLS

(1) AKW = Augmented Knowledge Workshop

(1) PURPOSE OF SYSTEM: Augmentation of Knowledge Work

(1) GOAL: To provide computer based tools to accomplish all aspects of knowledge work with an emphasis on collaboration.

(1) OVERVIEW of system

(1) NLS = oN Line System

(1) TNLS = Typewriter version

(1) CAPABILITIES OF SYSTEM:

(1) Composing

(1) Editing

(1) Studying

(1) Structuring

(1) Browsing = viewing

(1) Printing

(1) Publishing

(1) Communicating =

(1) sending and receiving mail, messages, documents; teleconferencing; etc.

(1) Storing and retrieving =

(1) record keeping, library services, data bases, searching, etc.

(1) Calculating

(1) Course Organization

(1) NLS is divided under headings for the purposes of this course. The commands under each heading can be used to perform the general operation denoted by the heading, eg. "printing" includes commands that cause the system to print in various ways.

(1) NLS CONCEPTS:
1. GETTING TO NLS
2. STRUCTURE
3. PRINTING
4. ADDRESSING
5. EDITING
6. COMMUNICATING
7. TROUBLE SHOOTING AND HELP

NLS COURSE LEVEL:

* NLS training is further divided into five courses for ease of learning. Each level corresponds to what can be covered at one time. The things introduced at each level are determined by difficulty, usefulness, complexity, and quantity (i.e. so thaT there is not an excessive amount to cover at any one time).

Each level contains the commands from the previous level for review in addition to the commands to be introduced (which are marked by an *).

(1)BASIC TNLS
This is the first course level (basic) which covers those commands necessary to enter, edit, and "mail" typewritten information. It has a special structure and is published in the Journal (see -- Journal, 24207,).

* INTRODUCTION TO TNLS STRUCTURE AND VIEWING
This is the outline for the second course which introduces NLS structure (hierarchical) and special tools for viewing structured information ("view specs").

DEFINITIONS FOR THE COURSE OUTLINE

COMMANDS: you type some characters to tell the computer what to do. The characters you type are represented by the uppercase letters in each "command word".

CONTROL MARKER = WHERE YOU ARE: Where the computer thinks you are pointing to (to some character in some file); you may move it by specifying an ADDRESS; this is where your command will be done.

CTRL = hold down the control (CTRL) key WHILE typing the specified character.

Upper case characters in a TNLS command phrase are what you type.

OK or CR means you type a Carriage Return.

<esc> = the ESC or escape key on your terminal (sometimes labeled "alt mode").

(1) BASE C: = the TNLS ready signal. It means that you can type in an editing command (like home base...).

(1) SEND C: = the Sendmail subsystem ready signal. It means that you can type in a Sendmail command.

* DESTINATION:
In TNLS: DESTINATION = ADDRESS.
When referring to Group or Text, two ADDRESSES are needed.

* SOURCE:
In TNLS: SOURCE = ADDRESS OR TYPEIN
When referring to Group or Text, two ADDRESSES are needed.

* TYPEIN = a string of characters from the keyboard, ending with an OK. [TYPEIN has a special form when a FILE ADDRESS or Link or Ident is called for (You can tell from the noise words)].

*INTRODUCTION TO TNLS STRUCTURE AND VIEWING

1. GETTING TO NLS

(1) THE TERMINAL AND USE
See the "Basic TNLS-8 Course" [You usually have to dial a telephone number and place the receiver in your terminal's cradle]

NETWORK (ARPA)

(1) Network Login [for a new connection where you dial in]

[I] Type e [to get the Network's attention]

[II] Type LOGIN then give your name and password when asked.

[III] Type QUIT CR [after logging in to Net]

[IV] Type @ L SPACE 43 CR [to make connection to Office-1, Host 43]

You now should be connected to TENEX

TENEX "Executive"

(1) Login procedure,:

(1) [V] type your USERNAME SPACE PASSWORD SPACE CR [SPACE fills in account number automatically; you're then ready to call NLS]

* express Login, type:
ELOG USERNAME PASSWORD ACCOUNT CR

(1) Calling NLS
(V) Type NLS CR [it not necessary to call NLS more than once during one login session]

(1) CTRL c and continue

* Quit NLS and continue

2. ORGANIZATION OF THE SYSTEM

(1) FILES & DIRECTORIES

(1) Information in the origin ("parent") statement of a file [not numbered, contains file name, etc., do not edit]

(1) File names

* Types of files (indicated by filename extensions)

* TXT = sequential file which can be copied into NLS
COPY = a temporary sequential file, usually a message

(1) Load File:
Load File FILENAME OK [FILENAME WILL BE ECHOED]

(1) User creation of files

(1) <>CReate File FILENAME OK

To see a list of all your files:

<>SHow Directory (of) OK:OK OK [this defaults to your directory]

(1) FILE STRUCTURE

(1) STATEMENTS and statement numbers [The basic element of structure in a file]

* Primary relationships between statements
substatement & source
SUCCESSOR: statement immediately succeeding current statement at same level with same source
PREDECESSOR: statement immediately preceding current statement at same level with same source

* STRUCTURES made up of statements:
BRANCH: statement plus substructure
GROUP: set of branches with same source

3. PRINTING: to see specified view of stored information
[To see anything in TNLS you must print it]

Printing on a terminal:

(1) Print STRUCTURE (at) DESTINATION VIEWSPECS

(1) Print File OK

(1) Print Journal (mail) OK

(1) Print Rest OK

Easy print, type a \ [prints the statement where you are]

(1) VIEWSPECS: to specify what you see

(1) Default is show all of the text; m/n = numbers on or off; and y/z = blank lines on or off

* To clip levels and lines, use lower case viewspecs including:
a/b - show one level less/more
c/d - show all levels/show first level
e - show referenced statement level
g/h - show branch only/show all branches
q/r - show one line less/more
s/t - show all lines/show first lines only
w/x - show all lines, all levels/show one line, one level

* SIDS (Statement Identification Numbers)
I/j - SIDS on instead of statement numbers/statement numbers instead of SIDS
[can be used in place of statement numbers in NLS]

4. ADDRESSING

(1) Control Marker concept (searches left to right) = where you are

(1) Jump to a new address:
(1) Jump (to) Address ADDRESS VIEWSPECS OK

* To tell where marker is

* period [.] command

* slash [/] command shows marker context

* Addressing combined with editing

(1) ADDRESSING WITHIN A FILE, use a:

(1) STATEMENT NUMBER:
A statement number is assigned to a statement, but not included in it.

* CONTENT ADDRESS: "TYPEIN" (surrounded by quotes)
where TYPEIN = the text to be searched for.

* ADDRESSES within one statement(preceded by plus or minus)
Letters PRECEDED IMMEDIATELY BY A PLUS (+) mean SKIP FORWARD, BY A MINUS (-) mean SKIP BACKWARD. A number between the plus or minus and the letter indicates the number of skips.

* e skip to end (last character) of statement

* f skip to front (first character) of statement

* ADDRESSING BY JUMPING

TO FIND A WORD OR STRING OF CHARACTERS ("CONTENT"):
[type a CTRL b for CONTENT to continue search for same thing]

* Jump (to) Word Next CONTENT VIEWSPECS OK

* Jump (to) Word First CONTENT VIEWSPECS OK

* Jump (to) Content Next CONTENT VIEWSPECS OK

* Jump (to) Content First CONTENT VIEWSPECS OK

* TO GO BACK TO PREVIOUS OR OTHER FILES

  * Jump (to) File Return OK ANSWER OK
    [type an N instead of second OK - next filename
    in stack will be echoed]

* TO GO BACK TO PREVIOUS VIEWS:

  * Jump (to) Return OK ANSWER OK

* TO JUMP BY STRUCTURE:

  * Jump (to) Origin DESTINATION VIEWSPECS OK

  * Jump (to) End (of Branch) DESTINATION VIEWSPECS OK

* TO JUMP TO AN ADDRESS OR LINK:

  * Jump (to) Link CONTENT OK

(1) ADDRESSING BETWEEN FILES AND DIRECTORIES:

* ADDRESSES in and among files (preceded by a period): [A number before any of these letters indicates the number of moves (default for number is 1].

  * .l link [find and jump on the next link]

(1) LINKS for addressing in all the above cases:
[see also Jump to Link]
  (FILENAME,ADDRESS) OR
  (DIRECTORY,FILENAME,ADDRESS) OR
  (DIRECTORY,FILENAME,ADDRESS:VIEWSPECS)

5. EDITING -- may be combined with addressing.
Syntax: VERB NOUN A: ADDRESS (L: LEVEL) CR (OK? CR)

* STRINGS and STRUCTURE = "nouns":

* STRINGS:

* Character
Word [note that the system readjusts spaces]
Text

* STRUCTURE:

Statement (statements renumbered automatically)

* Branch

* Group

(1) EDITING COMMANDS:

* LEVEL-ADJUST determines the level of a statement at a new location

(1)Just a CR = same level

*u (adjust up a level from preceding statement)
d (adjust down a level from preceding statement)

INSERT

(1) Insert STRUCTURE (to follow) DESTINATION LEVEL-ADJUST CONTENT OK

* Insert STRING (to follow) DESTINATION CONTENT OK

(1) continue to insert: CTRL e instead of OK puts you in the Enter mode, CTRL x to get out.

DELETE

(1) Delete File CONTENT OK

(1) Delete STRUCTURE (at) DESTINATION OK

* Delete STRING (at) DESTINATION OK

(1) SUBSTITUTE

(1) Substitute STRING in STRUCTURE (at) ADDRESS CR
(New STRING) T: TYPEIN CR
(Old STRING) T: TYPEIN CR Finished? Y/N: Y [for yes]
Substitutions made: NUMBER

[will replace the old STRING with new STRING every time it finds it in the STRUCTURE.]

(1) MOVE

(1) Move STRUCTURE (from) SOURCE (to follow) DESTINATION LEVEL-ADJUST OK

* Move STRING (from) SOURCE (to follow) DESTINATION OK

(1) COPY (see also Directory)

(1) Copy STRUCTURE (from) SOURCE (to follow) DESTINATION LEVEL-ADJUST OK

* Copy STRING (from) SOURCE (to follow) DESTINATION OK

* REPLACE

* Replace STRUCTURE (at) DESTINATION (by) CONTENT OK

* TRANSPOSE

* Transpose STRUCTURE (at) DESTINATION (and) DESTINATION OK

* Transpose STRING (at) DESTINATION (and) DESTINATION OK

(join with) CONTENT OK

(1) UPDATE FILE [not imperative, but good practice]

(1) Update File OK

6. COMMUNICATING with other users

(1) SENDMAIL and the Journal

(1) Interrogate

(1) Interrogate OK (distribute for action to:)
IDENT/,Lastmane
(distribute for information-only to:) IDENT/,Lastmane
(title:) CONTENT
(type of source:) Message, STRUCTURE, or FILE (at)
SOURCE
(show status?) ANSWER
(distribute the mail now?) ANSWER

* Individual commands: instead of Interrogate, specify by using the following:

* Title CONTENT OK

* Distribute Information (Only) (copies to) CONTENT OK

* Distribute Action (copies to) CONTENT OK

* Comments CONTENT OK

* Send a message or statement, (or ,name),

* Message CONTENT OK

* Statement (at) SOURCE OK

* Send STRUCTURE and files

* Group (at) SOURCE OK

* Plex (at) SOURCE OK

* Branch (at) SOURCE OK

* File DESTINATION OK

* Show (status) OK

(1) Initial file = mailbox

* To identify a user by lastname or ident:

* Show Record (for ident) CONTENT OK

(1) Send Message (Tenex) [Goto Tenex first and type:]

(1) SNDmsg CR [The system will ask you for:]
(to:) [lastnames separated by comma]
(cc:) [lastnames separated by comma]
(subject:) [TYPEIN of your message]
(message:) TYPEIN
CRTL z
CR [to terminate and send the message]

(1) Linking (Tenex)
WHEre (is) USERNAME CR [do not link when user is in SNDMSG, OUTPRC, NOUTPRC, or XLIST]

LINk (to) USERNAME CR [precede comment with ; repeat every 3 lines]

7. TROUBLE SHOOTING AND HELP

Type ? for commands or needed info.; available after any prompt.

* HELP:

Type CTRL q for help concerning what you are doing or type H for Help command (after Help you can type any word in NLS you wish to know about). CTRL x gets you out of Help and back to where you were.

* Help TYPEIN OK

* Help OK

(1) Send a message or Journal item to: FEEDBACK

(1) call SRI/ARC, (415 326-6200, ext.3630)

(1) link to FEEDBACK

(1) Status commands

(1) ^t

* Show Disk Status (see allocation)

(1) Remedies

(1) ^c, reset, NLS

* dsk pages and over allocation,..expunge

(1) Update File Compact
Delete Modifications (destroys all changes since last update)

PRACTICE

(1) Primer

* Exercise file

* Use Strategies
How to use the system to accomplish specific tasks, from daily routine tasks to creative intellectual enterprise.

NEXT AVAILABLE COURSES:

* 3. INTERMEDIATE TNLS
This is the third formal course or level of expertise, and represents significant experience with the system. The Programs and Useroptions subsystems are introduced as well as Output Processing for printer formating.

* 4. ADVANCED TNLS
This is the fourth course or level, which introduces new commands that emphasize alternatives, short cuts and special applications.

* 5. EXPERT ADVANCED TNLS
This is the fifth, most advanced level which covers any NLS command phrases or application techniques remaining in the running system.

* Examples

STRUCTURE:

< DIRECTORY, MENU.NLS;1, >, 28-JAN-75 17:29 JHB ;;;;

1 SOUP

1A VEGETABLE

1B CREAM OF MUSHROOM

2 ENTREE

2A FRIED CHICKEN

2B PRIME RIBS

2C SALMON

2C1 WITH CREAM SAUCE

3 DESSERT

3A PIE

3A1 APPLE

3A2 BLUEBERRY

3B ICE CREAM

3B1 VANILLA

3B2 PEPPERMINT

3B3 MAPLENUT

3B4 CHOCOLATE

4 BEVERAGE

4A TEA

4B COFFEE

TITLE PAGE 25276
25277

JHB 2-FEB-75 23:05 25278

INTRODUCTION TO TNLS STRUCTURE AND VIEWING

SRI-ARC

3 FEB 75

Augmentation Research Center

STANFORD RESEARCH INSTITUTE
MENLO PARK, CALIFORNIA 94025

JHB 2-FEB-75 23:05 25278

SRI/ARC INLS COURSE 2: INTRODUCTION TO STRUCTURE AND VIEWING

(J25278) 2-FEB-75 23:05;;;; Title: Author(s): James H. Bair/JHB;
Distribution: /JCN([ACTION]) JMB([ACTION]) UD([INFO-ONLY])
RLL([INFO-ONLY]) RA3Y([INFO-ONLY]) ; Sub-Collections: SRI-ARC
UD; Clerk: JHB; Origin: < BAIR, SECONDCOURSE.NLS;3, >, 2-FEB-75
22:53 JHB ;;;;####;

KEV 3-FEB-75 08:49 25279

bursting machine

EKM informs that bursting machines can be obtained fairly cheaply.
If this is indeed the case, it would be very nice if we could get one
and keep it in the lineprinter room. 1

bursting machine

(J25279) 3-FEB-75 08:49;;;; Title: Author(s): Kenneth E. (Ken) Victor/KEV; Distribution: /DCE([ACTION]) JCN([ACTION]) RWW([ACTION]) SRI-ARC([INFO-ONLY]) ; Sub-Collections: SRI-ARC; Clerk: KEV;

JEW 3-FEB-75 11:16 25280

Answers to ISI PCP Questions re PCPTNXINT

Dick--

1) The parameter list does indeed contain the address of the result list in (PCPTNXINT,6a3b3). Sorry.

2) The parameter list lifetime you suggest is exactly what I had in mind. Just one of the myriad details missing from the documentation. I appreciate all the feedback; Version 3 may even turn out to be self-consistent.

3) [SRI-ARC]<NLS>PCPV2CHANGES.TXT has been updated. You may want to suck over a fresh copy.

--Jim

JEW 3-FEB-75 11:16 25280

Answers to ISI PCP Questions re PCPTNXINT

(J25280) 3-FEB-75 11:16;;;; Title: Author(s): James E. (Jim) White/JEW; Distribution: /JBP([INFO-ONLY]) ; Sub-Collections: SRI-ARC; Clerk: JEW; Origin: < WHITE, MANMSG,NLS;2, >, 3-FEB-75 11:12 JEW ;;;;####;

DCE 3-FEB-75 14:41 25281

To JML re arranging office interchange with CHI

Jeanne: I have spoken with Charles about changing offices with you; no problem for him. Will you now please look into the red tape involved, check timing and arrangements with Charles, and produce a plan for approval by me, RWW and JCN? Then I'd like for you to administer as much of the move as possible. Thanks, Doug 1

DCE 3-FEB-75 14:41 25281

To JML re arranging office interchange with CHI

(J25281) 3-FEB-75 14:41;;;; Title: Author(s): Douglas C. Engelbart/DCE; Distribution: /JML([ACTION]) CHI([INFO-ONLY]) RWW([INFO-ONLY]) JCN([INFO-ONLY]) ; Sub-Collections: SRI-ARC; Clerk: DCE;

JAKE 3-FEB-75 14:49 25282

Bursting Machine

Although there is a bursting machine downstairs, I too think it would be useful and timesaving to have one of our own. Bursting the sheets containing the journal alone has been fairly time consuming. And as distribution of printed material for KWAC picks up, even more time will be required for bursting. So if the expense is not prohibitive it would be worth considering. 1

JAKE 3-FEB-75 14:49 25282

Bursting Machine

(J25282) 3-FEB-75 14:49;;;; Title: Author(s): Elizabeth J. (Jake) Feinler/JAKE; Distribution: /SRI-ARC([INFO-ONLY]) ; Sub-Collections: SRI-ARC; Clerk: JAKE;

SLJ 3-FEB-75 15:13 25283

Instructions for TIP Port Parameter Set-Up

Following are instructions for setting TIP port parameters temporarily from a terminal directly connected to a TIP. (Changes will stay until TIP is reloaded or the port that is changed is given a reset command or "hung up".) 1

NOTE: Keys that do not print such as carriage return, altmode (called "escape" on some terminals) and control characters, are named inside angle brackets, e.g., <cr>, <alt>, and <ctrl-c>. <> represents a space. 2

1) @<>x<>r<cr> 2a

2) @<>x<>d<>r<>(a number which represents device rate code e.g., 1023 for 208A modem)<cr> 2b

3) @<>x<>f<cr> 2c

4) @<>x<>d<>c<>a<cr> 2d

5) @<>x<>e<>a<cr> 2e

6) @<>x<>g<>b<cr> 2f

x=octal port number 2f1

7) terminal at port x should now echo thru TIP, Before attempting to log in type: @<>e<>n<cr> 2g

TIP parameters 3

for 208 (ext. clock): dev rate 1023 base 8 3a

TIP status lites: 3b

inrate, outrate=17 base 8 3b1

size=3 3b2

mstat=10 base 8 (or greater) 3b3

sgab,snab=1 3b4

for vadic va3405 3c

1200 baud 3c1

dev rate 503 base 8 3c2

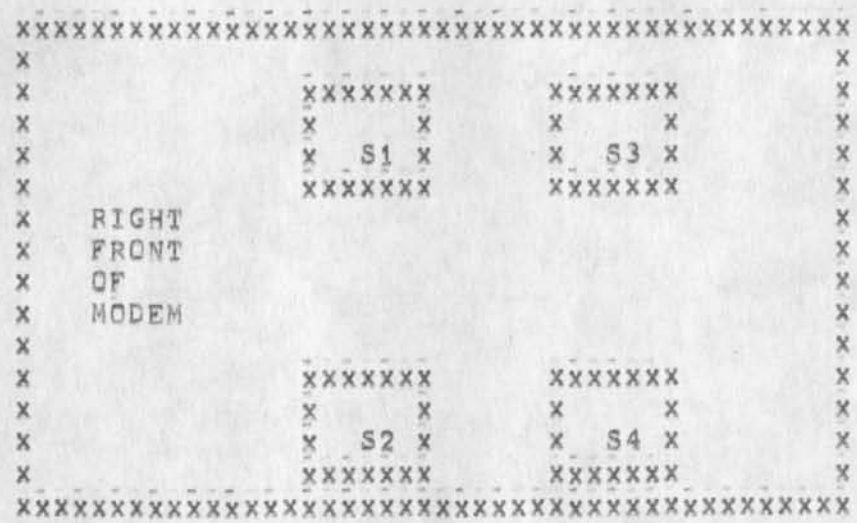
lites: inrate, outrate=7 3c3

Instructions for TIP Port Parameter Set-Up

size=3 3c4

mstat=10 base 8 (or greater) 3c5

208A MODEMS
Internal Switch Settings: 3c5a



3c6

IN RELATIONSHIP TO THE ABOVE DIAGRAM: 3c6a

SWITCH SETTINGS AT TIP END 3c6b

UP	U*
Down	U
UP	D
U	D
U	U
U	D

*This switch controls clock type
UP=EXT SYNC

3c6b1

SWITCH SETTINGS AT LINE PROCESSOR END 3c6c

U	D

Instructions for TIP Port Parameter Set-Up

U	U
U	D
U	D
U	U
U	D

3c6c1

SLJ 3-FEB-75 15:13 25283

Instructions for TIP Port Parameter Set-Up

(J25283) 3-FEB-75 15:13;;;; Title: Author(s): Sandy L. Johnson/SLJ;
Distribution: /RMS2([ACTION]) SLJ([INFO-ONLY]) ;
Sub-Collections: SRI-ARC; Clerk: SLJ; Origin: < JOHNSON,
MARTY,NLS;7, >, 31-JAN-75 16:50 SLJ ;;;;####;

DCE 3-FEB-75 15:16 25284

Latching on to some TTY33-ASRs for standby input service

Special attention for Norton (Applicactions), Watson (Development), VanNouhuys (DPCS Nucleator) and Placko (SRI Architect). 1

Dave Brown is purging some Teletype Model 33s, and I find that we have a few ourselves. It occurs to me that over the coming few months ARC may well get into tough binds for online terminal support, and that it might come in handy to have some extra spooling-typewriter input resources. 2

If there is likely to be some experimenting with providing heavier keyboarding services within the Institute, as via the SRI Slot, the SRI Architect might see some value in having some of these TTYs around. And or perhaps Dirk and the DPCS kickoff might likewise see value in extra inputting capacity. (Tom H. or Pat W-O'K?) 2a

Martin Hardy says that there 5 or 6 TTY3 ASRs (AUtomatic Send-Receive -- having paper tape I/O) available now. ARC has 3 or 4, and DRB is discarding 2. In case they would have any value, there are also 4 or 5 KSRs (Keyboard Send-Receive -- no paper tape). These could be useful for standby online input-only (or input-mainly) use. He thinks that TTY33 maintenance costs for occassional-use standby would be quite low. 2b

About the TTY33 ASRs: I realize that they are noisy, and only have one case. But I also realize that we have an adequate system for upper/lower directives in the inputting strings to produce any reasonable results for the resulting online file. Also, for temporary help, new people learn how to run paper tape much more readily and reliably than cassettes (not typing when spooler not on, or erasing, or etc.). Also, if there i a push, they are cheap-to-keep-around backup. It works to put a number of them in one room, so they only bother each other. 3

If any of you sees value, please feed back to me. Or also if you see explicit reason not to keep them around? 4

DCE 3-FEB-75 15:16 25284

Latching on to some TTY33-ASRs for standby input service

(J25284) 3-FEB-75 15:16;;;; Title: Author(s): Douglas C. Engelbart/DCE; Distribution: /RWW([ACTION]) JCN([ACTION]) DVN([ACTION]) MAP2([ACTION]) MEH([INFO-ONLY]) ; Sub-Collections: SRI-ARC; Clerk: DCE;

DCE 3-FEB-75 16:59 25285

To Farber re ARC interest in mini-net for NLS service system

For SNDMSG to FARBER@ISI

DCE 3-FEB-75 16:59 25285

To Farber re ARC interest in mini-net for NLS service system

Dave: Hope you had a good trip to Europe, and also that the NSF proposition will move along now. You had a question about our interest in your mini-net utilization for either or both the NLS Frontend and Backend. 1

This has to be a longer-term matter for us -- we have other, shorter-term problems and possibilities that I think must take precedence for that size commitment of our available manpower. Such priority decisions have to be hashed out within Dick Watson's group, and within the whole of ARC. We are going to be roughing out our next year's development plan during the next five or six weeks, and we should be able to generate a logical place for the mini-net in our longer-term picture. That "place" in our plan may well just be fixing a time for serious review, which could be followed then perhaps by submitting a proposal for support of the venture, or then dropping the idea, or etc. 2

I'm asking Dick Watson, who heads our Development group, to put this mini-net possibility study in his planning hopper. We could have some tentative notions in two weeks, and a relatively definite statement in six weeks, about how we plan to deal with this possibility.. 3

I think that we all found the concepts and possibilites quite appealing. But we also find ourselves very much overwhelmed these days with things that have to be given higher priority. For instance, given the Exclusive-Or decision as to whether to use the same manpower resources to build a mini-net system or to work out a transfer to a Systems 370 Backend, we'd probably be forced to choose the latter. However, we most likely will plan to explore and pursue the Backend transfer, and we could quite feasibly propose therein to include a serious study of the mini-net as one of the candidate, Backend-transfer target systems. 4

DCE 3-FEB-75 16:59 25285

To Farber re ARC interest in mini-net for NLS service system

(J25285) 3-FEB-75 16:59;;;; Title: Author(s): Douglas C. Engelbart/DCE; Distribution: /RWW([ACTION]) NPG([INFO-ONLY]) JCN([INFO-ONLY]) RLL([INFO-ONLY]) ; Sub-Collections: SRI-ARC NPG; Clerk: DCE;

KEV 3-FEB-75 17:45 25286

more on nsw debugging

not distributed outside arc yet

KEV 3-FEB-75 17:45 25286

more on nsw debugging

Preface 1

My previous document (25117,) on debugging in the NSW world was written from the point of view of implementing the debugging aids. This document, also intended as a working document, will approach the task of debugging from the point of view of a programmer who wishes to debug a new tool for NSW, or to fix a bug(!) in an exisitng tool or in the NSW software itself. 1a

Overview - Some Debugging Approaches 2

Over the years there have been a number of debugging approaches used to aid the development and maintainance of programs. Each approach has it relative merits and drawbacks. 2a

Console Debugging 2b

In this approach a programmer uses the operator's console to examine and modify physical core locations. 2b1

The nice features of this approach are: 2b2

This method usually does not require that there be separate versions of a program for debugging and for production use of the program. 2b2a

It is frequently possible to select a specific memory location and to cause the CPU to halt if the selected location is read and/or written and/or contains an instruction that is to be executed. 2b2b

It is usually possible to execute a program one instruction at a time, thus making it possible to examine the state of the program any time it is desired. 2b2c

The are many drawbacks to this approach: 2b3

This approach is totally unsuited for debugging programs, other than the monitor itself, that are running in a time-sharing environment. 2b3a

This approach is difficult to use on a system that supports either a virtual or paged memory system. 2b3b

Only one person can be debugging at a time. 2b3c

The medium of information exchange is restricted to console lights and switches. 2b3d

This is often a very tedious and unforgiving (of operator mistakes in keying in data for example) approach. 2b3e

Core Dumps 2c

In this approach, a programmer receives (usually after some catastophic condition has occured, but sometimes in response to a programmed request) a core dump. The core dumps run the gamut from an octal dump of the entire contents of physical memory to highly formatted symbolic dumps of selected areas of a virtual address space. 2c1

The advantages of this method of debugging are: 2c2

This method usually does not require that there be separate versions of a program for debugging and for production use of the program. 2c2a

The entire state, at the time the dump is made, is available. 2c2b

The disadvantages of this approach are: 2c3

It usually provides after the fact information about a program that has gone bad. Frequently the cause of the error occured before the state got bad enough to cause a dump to be made. 2c3a

When dumps are only available after some catastophic condition has been reached, a programmer gets no feel of the flow that the program took to get to this state. 2c3a1

Frequently, much more information is provided than is desired. 2c3b

In the case of numeric memory dumps, it is difficult to glean the deisred information. 2c3c

In the case of requested (selective) dumps, this method requires separate versions of the program for debugging and production use. 2c3d

This method is infameous for wasting huge amounts of line printer paper. 2c3e

Program Traces 2d

In this approach the programmer puts together a version of a program with debugging checks as part of the program. These

checks take the form of decision point traces, procedure call traces, etc. The resulting traces can either be presented to the user either on some hardcopy device, or a history file can be produced that can then be analyzed by perhaps some other debugging aid. 2d1

The advantage of this approach is: 2d2

it can provide the programmer with a good picture of the flow control of the target program. 2d2a

The drawbacks associated with this method are: 2d3

The programmer, while he may be able to tell when a program has gone awry, is usually not able to modify the program and to try differnet control paths. 2d3a

Occassionally a program would function properly with the debugging checks in place, and would function differently (improperly) withouth the debugging checks. 2d3b

This approach may provide much more information than is desired. (It may be quite a while before a program goes bad and much useless information may have been collected in the mena time.) 2d3c

Interactive Debugging 2e

In this approach a programmer monitors the program as it is running and has the ability to set (conditional) breakpoints, examine, and change his program to try different tacks. This approach, depending on the complexity of the debugger, can provide anywhere from octal debugging to high level language debugging. 2e1

The advantages of this approach are: 2e2

It does not require production and debugging versions of a program. 2e2a

A programmer can set a breakpoint at the beginning of suspect code and then step through the program (at any desired rate) to see where it goes bad. 2e2b

Suspect code and/or state information can be modified and then retried. 2e2c

A sophisticated debugger can provide the programmer the

ability to examine and modify his program using the source language that the program is written in. 2e2d

The drawbacks of this approach are: 2e3

This approach may require a good deal of intuition on the programmers part to get to the appropriate area for breakpoint setting. 2e3a

Goals 3

Since the envrionment to be provided by NSW is predominently an on-line interactive one, and since interactive debuggers are generally accepted to be the most powerful, we will be providing interactive debugging tools for the development and maintainance of the NSW and the NSW tools. 3a

As mentioned briefly above, interactive debuggers can provide a wide spectrum of capabilities. Thus it is necessary for us to decide which capabilities are desirable, and which capabilities are feasible to implement, both for the fist year NSW and in succeeding years. One of the more important points to keep in mind is the goal of providing a "consistant" interface to the user, regardless of the program that is being debugged and regardless of the sophistication of the debugger (this means providing upwards compatability as the debugger grows in sophistication). 3b

Desirable Capabilities 3c

The interaction techniques for communication between the user and the debugger should be concise, easy to remember, and invariant (as much as possible) across different target languages so that the programmer can concentrate his/her effort on debugging the target progam and not on the mechanics of debugging. 3c1

A good debugger should have the ability to examine any single or group of locations in the logical address space of the program being debugged. 3c2

The debugger should provide several different forms for presenting the examined locations to the user: 3c2a

numerically - in any base requested by the user 3c2a1

low-level symbollically - in terms of the assembly language of the physical machine that the program is running on 3c2a2

full symbollically - in terms of the high-level language that the program was written in; this includes a knowledge of the data types supported by a language so that, e.g., data could be presented as L10 records, etc. 3c2a3

textually - as a text string 3c2a4

others - some other presentation forms may include presenting a text string as a sequence of numeric bytes, etc. 3c2a5

A good debugger should have the ability to modify any single or group of locations in the logical address space of the program being debugged. This should include the ability to "insert" a sequence of instructions, i.e. a patch, where previously there was only a single instruction. 3c3

The debugger should provide several different forms for the user to specify the new contents of the target locations: 3c3a

numerically - in any base specified by the user 3c3a1

low-level symbollically - in terms of the assembly language of the physical machine that the program is running on 3c3a2

full symbollically - in terms of the high-level language that the program was written in (Ultimately and ideally this implies the implementation of incremental compilers or a marriage between the debugger and some language interpretters or some compromise similar to the Compile Procedure capability of NLS-8.) 3c3a3

others - some other specification forms may include specifying a text string as a sequence of numeric bytes, etc. 3c3a4

A good debugger should have a powerful breakpoint facility. 3c4

This facility should include the ability to specify conditional breakpoints of perhaps the following two types: 3c4a

a set of canned frequently used conditions that the debugger knows about, and 3c4a1

an ability for the programmmer to specify some code that gets executed to determine whether or not the breakpoint should be taken. 3c4a2

Once a breakpoint has been taken, it should be possible to have some prespecified actions take place. This might take on any of the following forms: 3c4b

a programmer specifies that he wishes to see the value of one or more variables (program or state) upon encountering the breakpoint 3c4b1

a programmer specifies a sequence of debugger commands to be executed when the breakpoint is taken 3c4b2

a programmer specifies a sequence of code to be executed when a breakpoint is taken. 3c4b3

It should be possible to set a breakpoint anyplace in the program. This includes setting a breakpoint to be taken when a data (as opposed to instruction) location is referenced. 3c4c

In addition to a powerful breakpoint facility, a good debugger should have the following controls over the execution of the program being debugged: 3c5

The ability to start the execution of the target program at any arbitrary location. 3c5a

The ability to single step through a program. 3c5b

The ability to execute a program in a (slowed down) trace mode. 3c5c

The ability to interrupt a running and/or looping program and enter the debugger. 3c5d

A good debugger should be able to execute "out-of-line" any arbitrary instruction that is specified either in machine language or in some higher level language. This includes the ability to do procedure calls from the debugger. 3c6

A good debugger should be able to examine and modify the state of the target program, e.g. examining and modifying the program call stack, etc. 3c7

When a programmer examines and/or modifies the program state, ideally he wants to do this in terms of the language that the target program is written in and not in terms of the machine language of the machine that the program is running on. 3c7a

A good debugger should provide sufficient capabilites so that a programmer can tell where he is: 3c8

did I get into the debugger via a breakpoint? if so, which one? if not, which instruction was interrupted to get me into the debugger? etc. 3c8a

A good debugger should provide the ability for a user to get a typescript of his debugging session so that it can be examined at a later time or perhaps used as a training device for other programmers. In addition to being able to get a typescript, it should be possible to temporarily divert output to files other than the terminal the user is at. 3c9

An ideal debugger would provide facilities for automaticcaly updating source code files to reflect the patches made during a debugging session. 3c10

A good debugger should be able to take advantage of the full power of the terminal that a user is currently using. 3c11

For example, if a user is at a display screen, it might be nice to display several instructions on either side of the currently executing instruction and to have some indication of program flow presented graphically to the user. 3c11a

A good debugger must be able to deal with any fork or process structure that is supported by the combination of the implementation language of the target program and the operating system under which the target program is running. 3c12

Some Observations 3d

Upon examining the set of desirable capabilites we see that a debugger can be roughly divided into the following two areas: 3d1

those areas of the debugger that are language independent, and 3d1a

those areas of the debugger that are language dependent. 3d1b

The borderline between these areas is in fact quite fuzzy, e.g. the debugger command to examine a specific location is a facility that is desirable regardlees of the implementation language of the target program; however, the format used to display the location contents to the programmer should be a function of the implentation language. Similarly, the debugger commands to show the top frame of the call stack, or to call a procedure are valuable regardless of the target programs

implementation language, but the format used to display the top frame and the code needed to implement the procedure call are very dependent on the implementation language. 3d2

On the other hand, a debugger command such as "Show Data Record" while meaningful for languages such as PL/1 or L10 do not make any sense for a language like FORTRAN which doesn't support records. Similarly, data specifications are most likely implementation language specific. 3d3

Some Conclusions 3e

Thus in designing a debugger, we have the following goals: 3e1

1) Define a basic set of core commands which will be useful for a broad set of implementation languages. 3e1a

2) Define implementation language specific commands, and if two or more languages have similar constructs then use the same commands for each of these languages. 3e1b

3) Choose an implementation design for the debugger that allows growth in both additional core commands and the support of new languages. This design goal dictates that the internal structure of the debugger be as follows: 3e1c

There will be a debugger kernel that implements the language independent features of the debugger and makes calls on language dependent packages (or modules). 3e1c1

There will be a number of language dependent packages to implement the language dependent features. The interface to these packages must be well defined and invarient over time. These packages must be dynamically loadable so that in any one debugging session, a programmer can move about between implementation languages. 3e1c2

The first year NSW Debugger 4

For the first year of the NSW we (SRI-ARC) will implement the following: 4a

We will design and implement the kernel to support the core commands. 4a1

We will design the interface to the language dependent packages. 4a2

We will design and implement most of the L10 package.

(Features such as incremental compilation and the display of core loacations in source code may be missing. However, we expect to have a facility similar to the Compile Procedure command of NLS-8.) 4a3

Debugger growth 5

We expect the growth of the debugger beyond the first year to proceed along the following lines: 5a

Perhaps the addition of more sophisticated core commands, and 5a1

The implementation of additional language packages. 5a2

KEV 3-FEB-75 17:45 25286

more on nsw debugging

(J25286) 3-FEB-75 17:45;;;; Title: Author(s): Kenneth E. (Ken) Victor/KEV; Distribution: /NPG([INFO-ONLY]) RWW([INFO-ONLY]) ; Sub-Collections: SRI-ARC NPG; Clerk: KEV; Origin: < VICTOR, USER-DEBUGGING,NLS;2, >, 3-FEB-75 10:19 KEV ;;;;####;

DVN 3-FEB-75 20:31 25287

No Network Virtual COM, but Output Processor Specifications

Reply to 31732

DVN 3-FEB-75 20:31 25287

No Network Virtual COM, but Output Processor Specifications

To my knowledge no network-wide standards of any kind exist for COM, let alone a Network Virtual COM. I suspect that the origin of your question is statements that ARC has a virtual COM device. That is true in the sense that our Output Procesor does the line and page formatting for the COM devices that serve it rather than merely producing a stream of characters which the vendor formats. 1

More to the point may be the parameters that our Output Processor expects a COM device to understand. They appear in (ijournal,14093,). I hope that is the information you need. If it is not, please ask again. 2

No Network Virtual COM, but Output Processor Specifications

(J25287) 3-FEB-75 20:31;;;; Title: Author(s): Dirk H. Van Nouhuys/DVN; Distribution: /JOAN([ACTION] dpcs notebook please) LAC([INFO-ONLY]) DPCS([INFO-ONLY]) JPB([INFO-ONLY]) VGK([INFO-ONLY]) WEC([INFO-ONLY]) ; Sub-Collections: SRI-ARC NSW DPCS; Clerk: DVN;

DVN 3-FEB-75 21:00 25288

rogramming for Electrostatic Printers

I returned to ARC Friday and Dick Watson left this morning for Boston
so I have only had a chance to get in a snatch of conversation about
rogramming for the XGP and other electrostatic printers. The snatch
as that on recounting programmers Dick felt he could not get in the
-8 weeks work we spoke of before July because all the appropriate
eople are committed to NSW. He left an opening to plead the case
hen he gets back and I will try to do that.

1

DVN 3-FEB-75 21:00 25288

Programming for Electrostatic Printers

(J25288) 3-FEB-75 21:00;;;; Title: Author(s): Dirk H. Van Nouhuys/DVN; Distribution: /JOAN([ACTION] dpcs notebook please) CKM([INFO-ONLY]) JMB([INFO-ONLY] this updates my message of earlier today) RWW([INFO-ONLY]) JCN([INFO-ONLY]) RLB2([INFO-ONLY]) ; Sub-Collections: DPCS SRI-ARC; Clerk: DVN;