KEV 21-JAN-75 10:34 25156

second draft of OSI document

This revision reflects the input I have received since the first draft. It still needs work in the following areas before it can be used as a programming guide: 1) actual references must be added, 2) procedure names must be decided on, 3)arguments and results must be firmed up, 4) may be missing some procedures which will be discovered as implementation proceeds. I consider myself to be done with it however, and expect DIA, who will be implementing the file stuff, and JLE, who will be doing the terminal stuff, to firm up the document (with my assistance). For those with no vested interest, I recommend re-reading the overview and if desired, browsing the rest of it.

second draft of OSI document

Overview 1

A process is an instance of a program with (among others) the following characteristics: 1a

it is running on one physical machine 1a1

it is running under the operating system of that machine 1a2

it has its own logical address space 1a3

it has its own state information 1a4

it consists of a collection of procedures and data stores 1a5

Thus the environment that a process sees consists of the physical machine that it is running on and the operating system under which it is running. We will call this environment the System Machine (SM). 1b

The environment that a programmer sees when writing a program in machine (or assembly) language is usually just the SM. However, when a programmer writes in a higher level language, such as COBOL, PL/1, FORTRAN, or L10, the environment that the programmer then sees consists of the SM and the runtime support for the language being used. Thus, such a programmer sees a COBOL machine, or a FORTRAN machine, etc. We will call this environment the Language Machine (LM). 1c

Application programs, written for the LM, then provide a new environment, e.g. an NLS Machine, for their users. 1d

Thus, we have the following hierarchy of environments: 1e

the physical machine environment at the bottom, 1e1

next the system machine environment, consisting of the physical machine and the operating system, 1e2

next the language machine environment, consisting of the SM and the runtime support package for the higher level languages, and finally 1e3

(perhaps several levels of) application environments. 1e4

When writing applications programs, to provide an application environment, a programmer faces the following two areas of concern: 1f

implementing and manipulating data stores and functions supported by the LM, and 1f1

implementing and manipulating data stores and functions supported by the SM. 1f2

For example (using L10 as our language, and TENEX as our operating system), the statement

```
a _ a+b;
```

is a statement written for, and supported by, the L10 LM; on the other hand, the statement

```
!gtjfn( a, b);
```

is a statement written for, and supported by, the TENEX SM. 1g

Once a programmer recognizes these two areas, it becomes possible to structure his/her programs in a manner that physically groups together those procedures and data stores for the LM, and those procedures and data stores for the SM. Those procedures and data stores written for the SM, and callable from procedures written for the LM, can then be said to provide an interface to the SM. 1h

We will call this collection of procedures and data stores that are written for the SM (as opposed to those for the LM), the Operating System Interface, OSI. 1i

We now see that if we have a running applications program (e.g. NLS) written in a higher level language (e.g. L10) that supports an application environment on one SM (e.g. an NLS environment for a PDP-10 running TENEX), and we now wish to provide this same application environment on a different SM (e.g. an NLS environment for a Honeywell-6080 running MULTICS), all that is required is that we examine and possible recode the procedures and data stores that make up the OSI. All the LM procedures and data stores should work as written with only a recompilation required to produce object code for the new SM. 1j

Procedures and data stores fall into logical groupings by virtue of their functions. Each such group will be called a package. Most of the following packages will, for efficiencies sake, live within the process that uses the package's procedures. However, this need not be the case and the procedures could in fact be invoked via PCP. In fact, the only procedures that need to exist in the process, in addition to those procedures implementing the applications environment, are those procedures that implement the LM (e.g., the L10 runtime package) and the procedures that implement PCP. 1k

The following packages (in addition to those procedures

implementing the application environment) are likely to exist, in a process' address space, in an NSW environment: 11

a LM runtime package for the implementation of the LM, 111

Procedure Call Protocol Packages (see -- xxx), 112

a File System package for the manipulation of files (see -- xxx), 113

a Process Management Package for the manipulation of other processes (see -- xxx), 114

a Debugging Package for the debugging of programs, and 115

a Miscellaneous Support Package containing procedures that don't fall into any of the above packages, but are of general use to a number of different applications. 116

The rest of this document will be concerned with specifying the procedures and data stores that make up the File System and Miscellaneous support packages within the OSI. 1m

It is not expected that any one process will contain all of the following packages. A process will contain only those packages that it needs. However, if a process requires one of the functions provided for by one of the following procedures, it is expected to use this procedure and NOT to write one of its own. 1n

The procedures described below are intended to be the set of primitives that are needed by NLS in an NSW environment. No attempt is being made at this time to provide a complete and exhaustive set of primitives that can be used by any tool at any time in the future. However, it is intended that this set will be a subset of some final package. 1n1

Syntax for the rest of this document 2

Each procedure will be described using PCP syntax (see -- xxx,). In addition, the following shorthand notations and formats will apply throughout (semantics will be discussed where appropriate with individual procedures): 2a

FILE* ==> file-id / LIST(user-id, file-id) 2a1

The list form of this must be used when refering to files supported by the NSW frontend (i.e., terminal files). 2a1a

user-id = INTEGER 2a2

This is the NSW user id that is associated with an individual user. 2a2a

file-id = INTEGER / empty 2a3

The empty format is only legal in certain procedure calls. 2a3a

NODE* ==> LIST(file-id, node-id) 2a4

node-id = INTEGER 2a5

WINDOW* ==> LIST(user-id, file-id, window-id) 2a6

window-id = INTEGER / empty 2a7

The empty format is only legal in certain procedure calls. 2a7a

ATOM* ==> LIST(user-id, file-id, window-id, atom-id) 2a8

atom-id = INTEGER / empty 2a9

The empty format is only legal in certain procedure calls. 2a9a

Introduction to the OSI File System 3

Files 3a

As far as NLS is concerned, a file is just a PCP data store of a specific USE-TYPE (see -- xxx). The following are the particular USE-TYPEs that we will be concerned with here: 3a1

NLS files 3a1a

An NLS file is a tree structured data store. This tree has certain global or file attributes associted with it, e.g. the creation time of the file, who last wrote the file, etc. In addition, each node of the tree has the following information associated with it: 3a1a1

structural information which reflects this node's position in the tree, 3a1a1a

statement attributes such as the name of this statement and "information retreival keys", 3a1a1b

implementation specific overhead, and 3a1a1c

properties such as the text property of this node or the graphics property of this node. 3a1a1d

All properties have associated with them the following information: 3a1a2

the name of this property, 3a1a2a

the size of this property, 3a1a2b

implementation specific overhead, 3a1a2c

optionally, a pointer to another tree, and 3a1a2d

the property information itself (text, picture, etc.). 3a1a2e

We will provide primitives, within the OSI, for the allocation, deletion, and manipulation of these nodes and properties. When an NLS file is first initialized, a process can indicate the number and the size of the "information retreival keys" to be associated with each node. The number and size of these "keys" then become non-alterable attributes of the file. 3a1a3

There exists a well defined algorithm for converting this tree structure into a PCP list of the following format (note that the structural information is imbedded in the syntax of the list): 3a1a4

```
<%filename% STRING>
   LIST( %file-attributes% LIST( any, ... ), branch )
branch = <%statement-id% INTEGER>
   LIST( stmnt-att, property, ... [, branch, ...] )
stmnt-att = LIST( %statement attribute% any, ... )
property = LIST( prprty-att, %property% any )
prpty-att = LIST( %property attribute% any, ... )
```

3a1a4a

Sequential files 3a1b

A sequential file is simply a PCP list. The KEY for this list is the file name (pathname) that uniquely identifies the file. The first element of this list is a list that contains certain attributes of the file (e.g. the bit length of individual bytes); succesive elements of the list are the data bytes of the file. 3a1b1

```
%filename% STRING; LIST( %attributes% LIST(), %bytes%
BITSTR, ...)
```

3a1b1a

Terminal files 3a1c

A terminal file is merely a sequential file with certain

attributes and with specific semantics applied to the data bytes. 3a1c1

We (OSI) will be providing procedures for manipulating the data contained in each of these types of files. While any procedure can be called and passed a file handle of any type of file, this will not always provide meaningful results (e.g. allocating a window in an NLS file). Such mismatches of functions and file types will: 3a2

be simulated as best as possible, or 3a2a

be a slow NOP, or 3a2b

generate an error. 3a2c

It should be noted that we are talking about logical files and not the physical devices that the data may actually reside on. 3a3

For example, a terminal file (of which a typescript file is an example) could very well live on a disk. In this case, there would be semantics associated with the data in the file such that if the disk terminal file is ever copied to a physical terminal, it would appear to the user just as if it had been created originally for the terminal. 3a3a

Filenames 3b

In the primitives described below, we will be using the following terms, with the following meanings: 3b1

pathname: a character string which fully describes the location of the file in the ARPA Network, 3b1a

file-id: an integer which is local to a user (who may have several processes) and is a shorthand way of referring to a pathname, and 3b1b

file-handle: a pathname or a file-id, 3b1c

More on Terminal Files 3c

When we talk of manipulating a terminal, in fact what we do is issue a procedure call, to a procedure in the OSI File System package, requesting that I/O be performed to a terminal file. The called procedure will then decide if the terminal file concerned is currently residing on a physical terminal, and if so issue the appropriate operating system calls to actually manipulate the terminal. If the concerned terminal file is not

currently residing on a physical terminal, then the called procedure simulates the requested function as best as possible, e.g. if the terminal file is currently residing on a disk, then the appropriate data bytes will be written into the file so that the terminal actions can be reproduced later. 3c1

When thinking about terminals, we see that they serve the following two basic functions: 3c2

accepting input from the user, and 3c2a

presenting information to the user. 3c2b

On any given terminal, there is a display space available for the presentation of information to a user. This total display space can be subdivided into informationally functional "windows", and windows can be subdivided into primitive units for the presentation of atomic pieces of information (e.g. strings, pictures). (Note that atoms are made up of elementary particles such as characters or lines, etc.) 3c3

The functional windows that are likely to exist for an NSW terminal are: 3c4

a status information window, called the status window, 3c4a

an error information window, called the error window, 3c4b

a command feedback and prompting information window, called the command window, and 3c4c

a tool specific information window, called the tool window. 3c4d

In an ideal world, where all NSW users have a super-sophisticated full graphics display terminal with an infinitely large display area, we could afford to reserve unique areas of the display space for each of the above windows. Getting back to earth, however, we are faced with the problem of mapping these logical areas into physical areas on a variety of different physical terminals, including typewriter-like devices. 3c5

One approach to this problem is to make NSW processes aware of the myriad terminals and the ideosyncrocies of each, and to tailor its interaction appropriately for each terminal. This would seem to be a never ending task (and perhaps counter-productive in that there might not be a "standard" interface anymore). 3c5a

Another approach is to provide an NSW Virtual Terminal and an interface to this logical terminal. 3c5b

A third approach is to provide a small set of NSW virtual terminals and interfaces to each of these logical terminals. Each virtual terminal is then representative of a class of physical terminals with similar characteristics. 3c5c

With approaches 2 and 3 it is then the function of the OSI procedures (in conjunction with operating system calls) to map these logical terminals into physical devices. These approaches have the advantage that supporting a new terminal only requires the addition of a new "device" driver, a much simpler conceptual task compared with changing code within the NSW frontend. 3c5d

NSW will use approach 3, i.e. a small set of NSW virtual terminals will exist, and we will be specifying the primitives for manipulating these virtual terminals. The classes of NSW virtual terminals will be: 3c5e

Class 0: ascii, half-duplex, line at a time, typewriter-like terminals, 3c5e1

Class 1: ascii, half-duplex, character at a time, typewriter-like terminals, 3c5e2

Class 2: ascii, full-duplex, character at a time, typewriter-like terminals, 3c5e3

Class 3: ascii, alpha-numeric displays, with editing capabilities, 3c5e4

Class 4: terminals that support ARC's line-processor protocol, and 3c5e5

Class 5: sophisticated graphics terminals (these will not be supported in the first year of the NSW). 3c5e6

(A forthcoming document will describe the characteristics of each of these calsses in more detail.) 3c5f

As mentioned above, windows can be subdivided into atoms such as strings or pictures. When outputting to a terminal, it is these atoms that are actually manipulated (written, deleted, etc.). 3c6

Note that the actual manipulation of terminal files, windows, and atoms that exist on physical terminals is only performed by

procedures, that are PCP-called, in the NSW frontend, and that all such calls must contain as one of their arguments the NSW id of the user (user-id). Thus we will be using the shorthands FILE*, WINDOW*, and ATOM* as described above. Also note that window names (window-id) are local to a file and atom names (atom-id) are local to a window. 3c7

There are three basic types of windows: 3c8

sequential windows, 3c8a

random windows, and 3c8b

cursor windows. 3c8c

A sequential window, consists of one and only one string, and is a window which simulates an alpha-numeric display: 3c9

characters written to the window are always appended to the end of the window, 3c9a

the characters linefeed, carriage-return, formfeed (which clears the entire window), backspace-character, backspace-word, and tab are simulated appropriately, 3c9b

writing after the last character position in an individual line causes automatic line overflow, 3c9c

writing after the last character position in a window causes automatic line overflow and causes device dependent scrolling. 3c9d

A facility similar to the TENEX scope terminal will exist for sequential windows (user input sets a count to zero and succesive output to the window increments this count; when scrolling would cause information to leave the window if the count is non-zero, then a pause occurs until the user inputs a (special?) character and the count is reset to zero.) 3c9d1

A facility will also exist so that the user can scroll back and forth through n lines of information that has passed through the window. 3c9d2

A random window, on the other hand, can contain as many atoms as desired by the process. The atoms are allocated, and deleted, etc, under program control. 3c10

A cursor window consists of one and only one atom. The bounds

of this window are allowed to change dynamically under program control. The contents of the one atom are also allowed to change dynamically under program control. 3c11

All windows have a process assignable priority associated with them. If two atoms overlap each other, then the atom belonging to the lower priority window will be made invisible. If two atoms with equal priority overlap, then the results are unspecified. 3c12

Now lets take a look at the input functions served by a terminal. A terminal can have a number of discrete devices associated with it, e.g., an alphanumeric keyboard, a mouse, etc. For the purposes of this document (and for the first year of the NSW) we will be concerned with only the following devices: 3c13

an ascii alphanumeric keyboard, 3c13a

(The NLS keyset can and will be thought of as just an extension of the ascii alphanumeric keyboard and input from the keyset will consist of any of 31 ascii characters. There will be a well defined character set for each chord.) 3c13a1

a pointing device (such as a mouse, a tablet, or cursor buttons on a keyboard), and 3c13b

a binary switch device, such as the buttons on a mouse, 3c13c

A process is then interested in knowing when the state of any of these devices changes, and when such a change takes place is interested in knowing the state of one or more of these devices. Primitives are provided for specifying which state changes are of interest to the process and the "report form" desired when such a change takes place. 3c14

All input from terminal files will be 8-bits wide. The first 128 codes will be used to represent the 7-bit ascii code, the second 128 codes will be used for reporting the states of other devices. 3c14a

(A note on the pointing device: there will exist a short circuit between the input pointing device and the output function of displaying the cursor to the user. This shortcut consists of the cursor window coordinates for the input pointing device being automatically updated.) 3c15

Some Assumptions about the Initial State 3d

When a process is first created, with regard to the state of its terminal files, it will be in what we we call the initial state. When a process is first created it will have file-ids for the following two open terminal files: 3d1

a "Primary Input File" - PIF, with file-id PIFID, and 3d1a

a "Primary Output File" - POF, with file-id POFID, 3d1b

(Note that if a process is not concerned with presenting information to a user at a terminal, or in obtaining input from a user, but is only providing computational capabilities (to perhaps another process) it need not concern itself with the existance of terminal files.) 3d2

The POF will have the following two windows defined (and the handles of these windows will be known): 3d3

a cursor window, known as the default cursor window. This is the window that will be used for tracking the input pointing device. The single atom for this window will consist of a pre-determined string or picture, and 3d3a

a sequential window, known as the default TTY window (DTW). The bounds of this window will correspond to the bounds available to the process for the presentation of information to the user. 3d3b

The PIF will be set to report only changes in state of the ascii alphanumeric keyboard. 3d4

We expect that the NSW Frontend process will do the following as part of its intialization code: 3d5

find out which class of virtual terminal the PIF and POF are living on, and as appropriate to this class of terminals 3d5a

make the DTW invisible, 3d5b

allocate an error window, 3d5c

allocate a status window, 3d5d

allocate a command window, 3d5e

allocate a sequential window as the tool window, and 3d5f

set up the PIF for the type of input it desires. 3d5g

The POF that exists for an NSW tool backend process will have as its DTW the tool window that was created by the frontend. Primitives are provided for finding out the bounds of this window, and the backend can manipulate this window (subdivide it, make it invisible and allocate random windows that are within with its bounds, etc.) to its heart's content. 3d6

The NSW backend process will not be allowed to directly manipulate the error, status, and command windows (or allocate windows that overlap the command window or error windows), but there are primitives provided for displaying error and status messages. 3d6a

The OSI File Manipulation Package - OFMP 4

General File manipulation Procedures 4a

get-file-id(partial-pathname, wm, file-use-type, new-old ,parms => file-id, full-pathname) 4a1

This procedure is used to obtain a file-id for a file given a partial pathname and other information about the intended use of the file. If a file-id cannot be obtained (e.g. if the partial pathname is not sufficient to uniquely identify a file) an appropriate error will be generated. Since it is syntactically ambiguous from the pathname whether or not the file is one supported by the Works Manager or a file on the local file system, the parameter wm indicates which file is desired. 4a1a

FORMAT: 4a1b

partial-pathname - CHARSTR 4a1b1

this is the partial pathname to be used to try to uniquely specify a file 4a1b1a

wm - BOOLEAN [WMF=TRUE / LF=FALSE] 4a1b2

if wm is WMF, then the desired file is one supported by the Works Manager. 4a1b2a

file-use-type - INTEGER [NLS=0 / SEQUENTIAL=1 / TERMINAL=2] 4a1b3

this parameter specifies for which use type of file a file-id is desired 4a1b3a

new-old - BOOLEAN [NEW=TRUE / OLD=FALSE] 4a1b4

this parameter specifies whether we are trying to get a file-id for a pre-existing file (OLD file) or if we wish to create a new file (NEW) 4a1b4a

parms - %terminal files% class / %new NLS files% keys / empty 4a1b5

This parameter is optional, and only needs to be specified in the following situations: 4a1b5a

if file-use-type specifies a terminal file and the partial-pathname specified does not refer to an actual physical terminal. If this is the case, then this parameter specifies which terminal class the file is to simulate. 4a1b5a1

if file-use-type specifies an NLS file and new-old specifies NEW. If this is the case, then this parameter specifies the number, and size, of the "information retreival keys" to be associated with the file. 4a1b5a2

class - INTEGER [CLASS0=0 / CLASS1=1 / CLASS2=2 / CLASS3=3 / CLASS4=4 / CLASS5=5] / 4a1b6

keys - LIST(%number-of-keys% INTEGER, %size-key-1% INTEGER, ...) 4a1b7

file-id - INTEGER 4a1b8

This result is the file-id for the specified parameters. 4a1b8a

full-pathname - CHARSTR 4a1b9

This result is the full pathname for the input file. This result is useful in situations where "file name recognition" is being performed. 4a1b9a

release-file-id(FILE*) 4a2

This procedure releases a specific file-id, or all file-ids, from use by this process. 4a2a

FORMAT: 4a2b

Note that if file-id is empty, then all file-ids known by this process will be released. 4a2b1

open-file(FILE*, open-type, bytesize) 4a3

Before input or output to a file can actually be performed, it is necessary to open the file. This procedure performs that function, and opens a file for a specific type of I/O. An error will be generated if the file (to which FILE* refers) cannot be opened for the type of I/O requested. 4a3a

FORMAT: 4a3b

open-type = LIST(INTEGER [READ=0 / WRITE=1 / APPEND=2 / EXECUTE=3], ...) 4a3b1

This parameter specifies what type of access is desired to a file. 4a3b1a

bytesize = INTEGER / empty 4a3b2

This parameter must not be empty if the file-id specified refers to a sequential file, and in that case specifies the number of bits in each byte of the file. This parameter will be ignored for file types other than sequential. 4a3b2a

close-file(FILE*, release-parm) 4a4

This procedure closes the specified file(s) and depending on the value of release-parm may release the file-id(s). After a file is closed, no further I/O can be performed to the file until it is opened. 4a4a

FORMAT: 4a4b

If file-id is empty, then all files opened by this process will be closed. 4a4b1

release-parm = BOOLEAN [RELEASE=FALSE / KEEP=TRUE] / empty 4a4b2

This parameter specifies whether or not it is desired to keep the file-id after the appropriate file(s) have been closed. If this parameter is not specified, it defaults to RELEASE. 4a4b2a

delete-file(FILE*) 4a5

This procedure deletes the specified file and releases the file-id for the file. An error will be generated if the file cannot be deleted. 4a5a

copy-file(FILE*-1, FILE*-2, copy-parms) 4a6

This procedure copies the file specified by FILE*-1 to the file specified by FILE*-2. If FILE*-1 and FILE*-2 refer to different use types of files, then the necessary format conversions will be performed in the process of copying. These format conversions can be modified by the value of copy-parms. If FILE*-2 refers to a pre-existing file, then the contents of that old file will be lost forever. 4a6a

FORMAT: 4a6b

copy-parms - LIST( ) / empty 4a6b1

The value of this parameter is not specified yet. It will be specified in the near future when we learn what is necessary. 4a6b1a

move-file(FILE*-1, FILE*-2, copy-parms) 4a7

This procedure is a shorthand for the following two procedure calls (it may be deleted if we decide it has limited use): 4a7a

copy-file(FILE*-1, FILE*-2, copy-parms) 4a7a1

delete-file(FILE*-1) 4a7a2

(Note that this is effectively a rename file with the appropriate format conversions being performed in the process of the rename.) 4a7b

convert-file-id-to-string(FILE*, convert-parms => string) 4a8

This procedure will convert the passed file-id into a string that represents the full pathname for the specified file. The format for the output string and which fields of the full pathname to be included are governed by convert-parms. 4a8a

FORMAT: 4a8b

convert-parms - this paramter will be specified later. 4a8b1

string - CHARSTR 4a8b2

read-file-attribute(FILE*, attribute-name => attribute-value) 4a9

This procedure will read an attribute of a file. 4a9a

FORMAT: 4a9b

attribute-name - CHARSTR 4a9b1

The attribute names will be specified later when we get a better idea of what we want. 4a9b1a

attribute-value - any 4a9b2

The value and type of this result depend on which attribute is being read. 4a9b2a

write-file-attribute(FILE#, attribute-name, attribute-value) 4a10

This procedure changes an attribute of a file. An error will be generated if the attribute is not allowed to be changed. 4a10a

FORMAT: 4a10b

attribute-name - CHARSTR 4a10b1

The attribute names will be specified later when we get a better idea of what we want. 4a10b1a

attribute-value - any 4a10b2

The value and type of this result depend on which attribute is being written. 4a10b2a

NLS File manipulation Procedures 4b

lock-file(FILE#, lock-type) 4b1

This procedure either locks / unlocks / or relocks (a previously locked or unlocked) file. If the file is already locked, then lock and relock generate errors; if the file is locked then the user-id specified must match the user-id of the locker before the file can be unlocked; if a file is unlocked and a relock is specified then the user-id specified must match the user-id of the person who had the file previously locked before it was unlocked. 4b1a

FORMAT: 4b1b

lock-type - INTEGER [LOCK=0 / UNLOCK=1 / RELOCK=2] 4b1b1

update-file(FILE#, update-parms, new-file-id) 4b2

This procedure will update a locked file as modified by the update-parms parameter. If the file is not locked this is a nop; if the file is locked then the user-id specified must match the user-id of the person who has the file locked or an error will be generated. 4b2a

FORMAT: 4b2b

update-parms = INTEGER [NORMAL=0 / COMPACT=1] 4b2b1

new-file-id = INTEGER / empty 4b2b2

If this parameter is not empty then the file specified will be updated to the file specified by new-file-id; if this parameter is not specified, then the file will be "updated in place". If this parameter is specified and it refers to an old file, then the contents of that old file will be lost; if this parameter is not specified, then the contents of the file specified by FILE* will be lost and will be replaced by the value of the updated file. 4b2b2a

For the following NLS file manipulation primitives the following shorthands and formats apply: 4b3

property-id = INTEGER 4b3a

the value of the property-id is an integer which is the address of the property itself and not the address of file system maintained fields such as the length of the property. 4b3a1

Note that property-ids are transitory and are only valid (in the absence of freezing a property in core) until the next property manipulation primitive (allocate, delete, undelete, get, get-next) is issued. 4b3a1a

property-name = INTEGER 4b3b

key-number = INTEGER 4b3c

key-value = INTEGER 4b3d

initialize-file(FILE*, keys) 4b4

This procedure enables a process to completely initialize a file. It is equivalent to getting a file-id for a new NLS file. 4b4a

FORMAT: 4b4b

keys = LIST(%number-of-keys% INTEGER, %size-key-1% INTEGER, ...) 4b4b1

allocate-node(FILE* => node-id) 4b5

This procedure will allocate a new node. The only filled in field in the node wil be "key-0", which is a file system maintained key. 4b5a

delete-node(NODE*, subs) 4b6

This procedure will delete a node, and any properties belonging to this node, and if subs is TRUE then any nodes (and associated properties) under this node in the tree. If subs is FALSE and the passed node has substructure, then an error will be generated. If this node is linked in a tree, it will be unlinked from the tree before it is deleted. 4b6a

FORMAT: 4b6b

subs = BOOLEAN 4b6b1

undelete-node(NODE*) 4b7

This procedure will undelete a node. 4b7a

allocate-property(NODE*, property-name, property-size, freeze => property-id) 4b8

This procedure will allocate a property with the specified name and size and associate it with the passed node. The allocated property will be frozen in core if freeze is FROZEN. 4b8a

FORMAT: 4b8b

freeze = BOOLEAN [FROZEN=TRUE / THAWED=FALSE] 4b8b1

delete-property(property-name, property-id) 4b9

This procedure will delete the specified property. 4b9a

Note that the specified property must be in core at the time this primitive is issued or an error will be generated. 4b9a1

undelete-property(property-name, property-id) 4b10

This procedure will undelete the specified property. 4b10a

Note that the specified property must be in core at the time this primitive is issued or an error will be generated. 4b10a1

The following procedures are used for moving around in the tree structure. They will fail if for example get-down-id is called and the passed node has no down node, etc. 4b11

get-successor-id(NODE* => successor-id) 4b11a

get-predecessor-id(NODE* => predecessor-id) 4b11b

get-up-id(NODE* => up-id) 4b11c

get-down-id(NODE* => down-id) 4b11d

get-tail-id(NODE* => tail-id) 4b11e

get-head-id(NODE* => head-id) 4b11f

get-end-id(NODE* => end-id) 4b11g

get-origin-id(NODE* => origin-id) 4b11h

get-next-id(NODE* => next-id) 4b11i

get-back-id(NODE* => back-id) 4b11j

link-node(NODE*, relative-node, relationship) 4b12

This procedure will link the passed node into the tree specified by relative-node as either the down or successor of relative-node as specified by relationship. 4b12a

FORMAT: 4b12b

realtive-node - INTEGER 4b12b1

relationship - BOOLEAN [DOWN=TRUE / SUCCESSOR=FALSE] 4b12b2

this parameter specifies the relation that the new node (NODE*) is to bear to the old node (relative-node). 4b12b2a

unlink-node(NODE*) 4b13

This procedure will remove the passed node from the tree in

which it exists. The proper fixup will occur for the father and brothers of this node. 4b13a

read-key(NODE*, key-number => key-value) 4b14

This procedure will read the value of the indicated key. Note that key-0 is a file system maintained key and is guaranteed to be unique for each node within a file. (This is the sid of the statement in NLS parlance.) 4b14a

write-key(NODE*, key-number, key-value) 4b15

This procedure will change the value of the specified key. (Note that key-number must be greater than 0.) 4b15a

find-key(NODE*, key-number, key-value, type => node-id) 4b16

This procedure will return the node-id of the node that contains as the value of its key-number key the value key-value. The search will start after node NODE* and either proceed by following the tree structure if type is STRUCTURE, or in a file system dependent order if type is ANY. 4b16a

FORMAT: 4b16b

type = BOOLEAN [ANY=TRUE / STRUCTURE=FALSE] 4b16b1

get-property-id(NODE*, property-name, which, freeze => property-id) 4b17

This procedure will return the property-id of the which-th (first, second, etc.) property with name property-name at the passed node. If freeze is FREEZE then then the property will remain resident in core until it is explicitly thawed. 4b17a

FORMAT* 4b17b

which = INTEGER 4b17b1

freeze = BOOLEAN [FROZEN=TRUE / THAWED=FALSE] 4b17b2

get-next-property(FILE*, parm, freeze => property-id) 4b18

This procedure will get either the first property associated with a given node, or the next property given a property. If freeze is FREEZE then then the property will remain resident in core until it is explicitly thawed. 4b18a

FORMAT: 4b18b

parm - LIST(BOOLEAN[=TRUE], NODE*) /
LIST(BOOLEAN[=FALSE], property-name, property-id) 4b18b1

freeze - BOOLEAN [FROZEN=TRUE / THAWED=FALSE] 4b18b2

Thaw-property(property-name, property-id) 4b19

This procedure will thaw a property. In other words, a programmer can no longer assume that property-id for the named property is valid if any of several property manipulation procedures get called. 4b19a

get-property-length(property-name, property-id => length) 4b20

This procedure will get the length of the passed property. 4b20a

FORMAT: 4b20b

length - INTEGER 4b20b1

this result is the number of words associated with this property. 4b20b1a

get-property's-node(property-name, property-id => node-id) 4b21

This procedure will return the node-id for the passed property-id. 4b21a

get-property's-subtree-node-id(property-name, property-id => node-id) 4b22

This procedure will return the node-id of the top node in the subtree associated with this property. This procedure will fail if there is no subtree associated with this property. 4b22a

link-subtree-to-property(NODE*, property-name, property-id, link) 4b23

If link is TRUE, then this procedure will associate the passed subtree (as specified by NODE*) with the passed property. (Note that the node identified by NODE* must not be in any other trees other than the tree it spawns.) If link is FALSE, then this procedure will remove the association between the the subtree and the property. 4b23a

FORMAT: 4b23b

link = BOOLEAN 4b23b1

Sequential File manipulation Procedures 4c

read-byte(FILE*, byte-name -> byte-value) 4c1

This procedure will read an individual byte from the specified file. 4c1a

FORMAT: 4c1b

byte-name = INTEGER / CHARSTR / empty 4c1b1

This parameter, if not empty, specifies which byte is to be read from the file. If the bytes in the file have string keys, i.e. names, associated with them, then it is possible to read them by specifying the byte's name. If this parameter is not specified, then it defaults to be the current byte position in the file, which is an attribute of the file. Reading (or writing) a byte in a file advances the current character position attribute of the file by one byte for each byte read (or written). 4c1b1a

byte-value = BITSTR 4c1b2

This result is the value of the selected byte. 4c1b2a

write-byte(FILE*, byte-value, byte-name) 4c2

This procedure will write an individual byte in the specified file. 4c2a

FORMAT: 4c2b

byte-value = BITSTR 4c2b1

This parameter specifies the new value for the selected byte. 4c2b1a

byte-name = INTEGER / CHARSTR / empty 4c2b2

This parameter, if not empty, specifies which byte is to be written in the file. If the bytes in the file have string keys, i.e. names, associated with them, then it is possible to write them by specifying the byte's name. If this parameter is not specified, then it defaults to be the current byte position in the file, which is an attribute of the file. 4c2b2a

read-string(FILE*, termination-condition, starting-byte-name
-> string-value) 4c3

This procedure will read a number of successive bytes from the specified file. 4c3a

FORMAT: 4c3b

termination-condition - LIST(%count%INTEGER, %chars%STRING) 4c3b1

This parameter specifies that either count bytes are to be read if chars is the null string; or that bytes are to be read until one of the bytes matches any byte in the chars string if a zero count is specified; or read bytes until a byte is encountered that matches any of the bytes in the chars string if both count and chars are specified. 4c3b1a

starting-byte-name - INTEGER / CHARSTR / empty 4c3b2

This parameter specifies which byte is to be the first byte read from the file. If the bytes in the file have string keys, i.e. names, associated with them, then it is possible to read them by specifying the byte's name. If this parameter is empty, then it defaults to be the current byte position in the file, which is an attribute of the file. 4c3b2a

string-value - CHARSTR 4c3b3

This result is the value of the selected bytes. 4c3b3a

write-string(FILE*, termination-condition, string-value, starting-byte-name) 4c4

This procedure will write a number of successive bytes in the specified file. 4c4a

FORMAT: 4c4b

termination-condition - LIST(%count%INTEGER, %chars%STRING) 4c4b1

This parameter specifies that either count bytes are to be written if chars is the null string; or that bytes are to be written until one of the new bytes matches any byte in the chars string if a zero count is specified; or write bytes until a byte is

encountered in the new string that matches any of the bytes in the chars string if both count and chars are specified. 4c4b1a

string-value - CHARSTR 4c4b2

This is the value for the new string. 4c4b2a

starting-byte-name - INTEGER / CHARSTR / empty 4c4b3

Terminal File manipulation Procedures 4d

Pseudo Interrupts 4d1

A pseudo-interrup normally occurs when the user inputs the appropriate character at his/her terminal. However, in the case of class 0 (half-duplex, line at a time) terminals, the pseudo-interrupt cannot occur until the user has transmitted the entire line and the character is read. 4d1a

(Eventually, it might be nice to have pseudo-interrupts associaed with non-terminal files. However, in the initial NSW implementation, if a FILE* is specified which refers to a non-terminal file, an error will be generated.) 4d1b

Enable-Pseudo-Interrupts(FILE*) 4d1c

This procedure turns on the pseudo-interrupt system for the specified file. Individual characters can be activated and deactivated independently of the enabled status of the pseudo-interrupt system for the file. However, until the PSI system is enabled for a file, any activated characters will NOT generate pseudo-interrupts and in fact will be dealt with as normal input. 4d1c1

Disable-Pseudo-Interrupts(FILE*) 4d1d

This procedure disables (turns off) the PSI system for the specified file. 4d1d1

Read-file-psi-status(FILE* => file-psi-status) 4d1e

This procedure reads the current status of the PSI system for the specified file (i.e. is it on or off), and returns a list of the activated character psi-char-ids, and the procedures associated with each character. 4d1e1

FORMAT: 4d1e2

file-psi-status = LIST(%status% BOOLEAN [ON=TRUE / OFF=FALSE], LIST(%psi-char-id% INTEGER, %proc-addr% INTEGER), ...) 4d1e2a

Activate-char-as-PSI(FILE*, char, priority, proc-name => psi-char-id) 4d1f

This procedure specifies that a character should generate a pseudo-interrupt if the PSI system is enabled. It also specifies the priority of the pseudo-interrupt to be associated with the character and a procedure to be called when the PSI is generated. 4d1f1

FORMAT: 4d1f2

char = CHARSTR 4d1f2a

priority = INTEGER [P0=0 / P1=1 / P2=2 / P3=3] 4d1f2b

This is the priority to be associated with the PSI for this character. If a pseudo-interrupt of priority n is in progress, then only PSIs of higher priority m (m<n) will be allowed to occur. Any PSIs of equal or lower priority (m>=n) will be remembered and will take place when the current PSI is "debreaked". 4d1f2b1

proc-name = INTEGER 4d1f2c

This parameter specifies the address of a local procedure to be called when the PSI is generated. (Note that this assumes knowledge of the language that the called procedure is written in so that the proper type of procedure call can be made. Initially in the NSW, we will assume this language to be L10; eventually, this may have to be replaced with an address to receive control rather than a procedure to be called.) 4d1f2c1

psi-char-id = INTEGER 4d1f2d

This result is an id that is to be used for future references to this PSI. 4d1f2d1

Deactivate-char-as-PSI(Psi-char-id) 4d1g

This procedure deactivates the PSI associated with the specified psi-char-id. 4d1g1

FORMAT: 4d1g2

psi-char-id - INTEGER 4d1g2a

Read-char-psi-status(psi-char-id -> char, priority, proc-name) 4d1h

This procedure returns the current status of the PSI associated with the specified psi-char-id. 4d1h1

FORMAT: 4d1h2

psi-char-id - INTEGER 4d1h2a

char - CHARSTR 4d1h2b

This result is the 8-bit character associated with the specified psi-char-id 4d1h2b1

priority - INTEGER 4d1h2c

This result is the priority associated with the specified psi-char-id 4d1h2c1

proc-name - INTEGER 4d1h2d

This result is the address of the local procedure that will be call if the specified PSI is generated. 4d1h2d1

Resume-from-PSI( ) 4d1i

This procedure resumes normal execution from within a PSI handling procedure. 4d1i1

Change-PSI-Return-Address(return-address) 4d1j

Normally a procedure that is called as a result of a PSI character performs a normal return and then the PSI system returns control to the instruction that was interrupted. This procedure gives a programmer the ability to "debreak" from a PSI routine (e.g. to allow lower or equal priority pseudo-interrupts to occur) and yet have control over where execution will resume. This procedure will generate an error if it is called when no pseudo-interrupts are in progress. 4d1j1

FORMAT: 4d1j2

return-address = INTEGER 4d1j2a

Echo control 4d2

Initially, the first 128 character codes will be echoed directly without any transformations applied and none of the second 128 codes will be echoed. The following two procedures are provided to modify this initial state. 4d2a

write-echo-status-file(FILE*, echo-status) 4d2b

This procedure allows a programmer to specify how characters will be echoed when they are typed at a terminal. (Note that specifying anything but no echoing for class 0 (line at a time) terminals can lead to ugly things appearing on the terminal.) 4d2b1

FORMAT: 4d2b2

echo-status = LIST(LIST(class / char, control), ...) 4d2b2a

class = INTEGER [CONTROL=0 / ALPHA=1 / NUM=2 / PUNC=3 / FIRST-128=4 / SECOND-128=5 / SPACE=6 / ALL=7] 4d2b2b

char = CHARSTR 4d2b2c

this is the 8-bit character to which the echo string applies 4d2b2c1

control = CHARSTR 4d2b2d

this is the string that should be echoed when the specified character is input. This can be a null string indicating no echoing; we will have to come up with some meta language to do things such as echoing a class of characters with a mapping of those characters, e.g. echo control-L as <^L>. 4d2b2d1

read-echo-status-file(FILE* => echo-status) 4d2c

This procedure reads the current echo status for the specified file. 4d2c1

Character translation control 4d3

Initially, all 256 character codes will be given to a process exactly as input from a terminal. The following two procedures are provided to modify this initial state. 4d3a

read-input-char-trans(FILE* => input-trans-list) 4d3b

This procedure will read the current status of input translation tables for the specified file. 4d3b1

FORMAT: 4d3b2

input-trans-list - see echo-status above 4d3b2a

write-input-char-trans(FILE*, input-trans-list) 4d3c

This procedure will write the input translation tables for the specified file. 4d3c1

Primary File Manipulation 4d4

At any time a process has associated with it one and only one primary input file and one and only one primary output file. Normally, and initially, these will correspond to the physical terminal being used by a user. These are the files that are referenced when a process uses the file-ids PIFID and POFID. The following procedures allow processes to divert input or output to or from a physical terminal to other devices, e.g. a disk file to be read later, or a command file for frequently performed functions. 4d4a

read-primary-file-handles(process-id => PIF-id, POF-id) 4d4b

This procedure reads the file-ids of the current primary input and output files. If I/O has not been diverted then these will bee PIFID and POFID. 4d4b1

FORMAT: 4d4b2

process-id - INTEGER (see -- xxx,) 4d4b2a

PIF-id - INTEGER 4d4b2b

This will be the file-id for the current primary input file. 4d4b2b1

POF-id - INTEGER 4d4b2c

This will be the file-id for the current primary output file. 4d4b2c1

set-primary-file-handles(process-id, PIF-id, POF-id) 4d4c

This procedure allows a process to change its current

primary input and/or output files. If a process does not wish to change its primary input file then it should use the file-id PIFID for the PIF-id parameter and simarily for its primary output file. 4d4c1

reset-primary-file-handles(process-id) 4d4d

This procedure resets the primary input and output files for the specified process back to what they were when the process first started up. 4d4d1

Window Manipulation 4d5

An attempt to manipulate windows on class 0, 1, or 2 (non-display) terminals or on non-terminal files will generate an error. 4d5a

allocate-window(FILE#, window-parms -> window-id) 4d5b

This procedure allocates a window with the specified characteristics in the specified file. 4d5b1

FORMAT: 4d5b2

window-parms = LIST(type, bounds, priority, visibility, hit-sensitivity, typewriter) 4d5b2a

type = INTEGER [SEQUENTIAL=0 / RANDOm=1 / CURSOR=2] 4d5b2b

bounds = LIST(%x1% INTEGER, %y1% INTEGER, %x2% INTEGER, %y2% INTEGER) 4d5b2c

this list specifies the coordinates of the lower left and upper right hand corners for the window 4d5b2c1

priority = INTEGER [FLOAT=-1 / P0=0 / P1=1 / P2=2 / P3=3] 4d5b2d

the FLOAT priority says that this window has equal priority with any windows it may be overlapping. (The default cursor window has priority FLOAT.) 4d5b2d1

visibility = BOOLEAN [VISIBLE=TRUE / INVISIBLE=FALSE] 4d5b2e

hit-sensitivity = BOOLEAN [SENSITIVE=TRUE / INSENSITIVE=FALSE] 4d5b2f

This parameter specifies whether or not the atoms

that make up this window are "hit sensitive" to the select primitive below. 4d5b2f1

typewriter - BOOLEAN / empty 4d5b2g

IF this parameter is true (only valid for sequential windows), then any characters output to the file POFID which are not part of any of the other commands specified below (e.g. part of a write-string command), will be sent to this window. 4d5b2g1

window-id - INTEGER 4d5b2h

This is the id that should be used to reference this window. 4d5b2h1

delete-window(WINDOW*) 4d5c

This procedure deletes the specified window(s), and any atoms that are belong to the window. 4d5c1

FORMAT: 4d5c2

If window-id is empty, then this procedure will delete all the windows, with the exception of the default TTY window and the default cursor window, that have been allocated by this process within the specified file. 4d5c2a

manipulate-window(WINDOW*, manipulation-parms) 4d5d

This procedure allows a process to manipulate a window. 4d5d1

FORMAT: 4d5d2

manipulation-parms - LIST(bounds, priority, visibility, hit-sensitivity, typewriter) 4d5d2a

bounds - LIST(%x1% INTEGER, %y1% INTEGER, %x2% INTEGER, %y2% INTEGER) 4d5d2b

This parameter can only be specified if the window being manipulated is a cursor window. If this is the case, then this list specifies the coordinates of the lower left and upper right hand corners for the window 4d5d2b1

priority - INTEGER [FLOAT=-1 / P0=0 / P1=1 / P2=2 / P3=3] 4d5d2c

visibility - BOOLEAN [VISIBLE=TRUE / INVISIBLE=FALSE] 4d5d2d

hit-sensitivity - BOOLEAN [SENSITIVE=TRUE / INSENSITIVE=FALSE] 4d5d2e

typewriter - BOOLEAN 4d5d2f

read-window-parms(WINDOW* => window-parms) 4d5e

This procedure allows a process to read the status of a window. 4d5e1

FORMAT: 4d5e2

window-parms - LIST() see above 4d5e2a

Atom Manipulation 4d6

allocate-atom(WINDOW*, atom-parms => atom-id) 4d6a

This procedure allocates an atom, with the specified characteristics, within the specified window. 4d6a1

FORMAT: 4d6a2

atom-parms - LIST(type, visibility, hit-sensitivity, type-parms) 4d6a2a

type - INTEGER [STRING=0 / PICTURE=1] 4d6a2b

visibility - BOOLEAN [VISIBLE=TRUE / INVISIBLE=FALSE] 4d6a2c

If the window in which this atom resides is currently INVISIBLE, then the atom will be invisible indendent of the visibility paratmeter for the atom; on the other hand, if the owning window is VISIBLE, then the visibility of the atom is governed by its visibility parameter. 4d6a2c1

hit-sensitivity - BOOLEAN [SENSITIVE=TRUE / INSENSITIVE=FALSE] 4d6a2d

If the window in which this atom resides is currently INSENSITIVE, then the atom will be insensitive indendent of the hit-sensitivity parameter for the atom; on the other hand, if the owning window is SENSITIVE, then the sensitivity of the atom is governed by its hit-sensitivity parameter. 4d6a2d1

type-parms - string-parms / picture-parms 4d6a2e

the value of this parameter is dependent on the type of atom being allocated 4d6a2e1

string-parms - LIST(coord, nlines, nchars) 4d6a2f

coord - LIST(%x% INTEGER, %y% INTEGER) 4d6a2g

this is the coordinate for where the string should start 4d6a2g1

nlines - INTEGER 4d6a2h

this parameter specifies how many lines the string should take 4d6a2h1

nchars - INTEGER 4d6a2i

this parameter specifies how many characters should appear in each line of the string 4d6a2i1

picture-parms - LIST( ) 4d6a2j

the value of this list will be specified later when we can more knowledge about pictures 4d6a2j1

atom-id - INTEGER 4d6a2k

this result is the id that should be used for all future references to the atom 4d6a2k1

deallocate-atom(ATOM*) 4d6b

This procedure will delete the specified atom(s). 4d6b1

FORMAT: 4d6b2

if atom-id is empty, then this procedure will deallocate all the atoms in the specified window. 4d6b2a

read-atom-parms(ATOM* => atom-parms) 4d6c

This procedure will read the current parameters for the specified atom. 4d6c1

FORMAT: 4d6c2

atom-parms - LIST( ) see above 4d6c2a

manipulate-atom-parms(ATOM*, new-atom-parms) 4d6d

This procedure allows a process to modify several of the parameters associated with an atom. 4d6d1

FORMAT: 4d6d2

new-atom-parms - LIST(visibility, hit-sensitivity) 4d6d2a

visibility - BOOLEAN [VISIBLE=TRUE / INVISIBLE=FALSE] 4d6d2b

hit-sensitivity - BOOLEAN [SENSITIVE=TRUE / INSENSITIVE=FALSE] 4d6d2c

String Manipulation 4d7

write-string(ATOM*, string-pos, string-parms, string) 4d7a

This procedure will write the passed string at the specifed location (specified by atom-id and string-pos). This is a destructive write, i.e., any characters ovewritten will be lost, and if this write is replacing a substring of a string with a different length substring, the resulting string will be opened or closed appropriately. The characters carriage-return line-feed will cause text to appear on succesive lines of the string. 4d7a1

FORMAT: 4d7a2

string-pos - LIST(sline, schar, eline, echar) 4d7a2a

sline - linec 4d7a2b

schar - charc 4d7a2c

eline - linec 4d7a2d

echar - charc 4d7a2e

linec - %first-line% INTEGER [=0] / %current-last-line% INTEGER [=-1] / INTEGER 4d7a2f

charc - %first-char% INTEGER [=0] / %current-last-char% INTEGER [=-1] / INTEGER 4d7a2g

Thus to insert a substring in front of an existing string, specify the string-pos of:
LIST(0, 0, 0, 0)

To append to the end of an existing string use:
LIST(-1, -1, -1, -1)
To replace an arbitrary substring use;
LIST(n, m, p, q)
To replace a substring at the front of a string use:
LIST(0, 0, m, n)
To replace a substring at the end of a string use
LIST(m, n, -1, -1)
To replace an entire string use:
LIST(0, 0, -1, -1) 4d7a2g1

string-parms - LIST(highlight) 4d7a2h

highlight - INTEGER [NORMAL=0 / HIGHLIGHT=1] 4d7a2i

This parameter specifies whether or not the newly written string should be made to "stand-out", in any terminal dependent manner, or not. 4d7a2i1

string - CHARSTR 4d7a2j

read-string(ATOM*, string-pos -> string) 4d7b

This procedure will read the specified (sub-)string. 4d7b1

FORMAT: 4d7b2

string-pos - see above 4d7b2a

string - CHARSTR 4d7b2b

move-string(ATOM*-1, ATOM*-2) 4d7c

This procedure will move the string from atom-id-1 to atom-id-2. The entire string at atom-id-1 will be replaced by a null string, and the entire string at atom-id-2 will be replaced by as much as will fit of string atom-id-1. 4d7c1

copy-string(ATOM*-1, ATOM*-2) 4d7d

This procedure will copy the string from atom-id-1 to atom-id-2. The entire string at atom-id-2 will be replaced by as much as will fit of string atom-id-1. 4d7d1

mark-characters(WINDOW*, atom-id-1, string-pos-1, atom-id-2, string-pos-2 -> mark-id) 4d7e

This procedure will cause the (sub-)string to be made to "stand-out" in any appropriate terminal dependent manner. The mark-id returned can be used for future references to these characters that are now standing out. 4d7e1

FORMAT: 4d7e2

string-pos-1 - see string-pos above 4d7e2a

string-pos-2 - see string-pos above 4d7e2b

mark-id - INTEGER 4d7e2c

remove-mark(FILE*, mark-id) 4d7f

This procedure will cause characaters that were previously made to stand-out to no longer standout. 4d7f1

FORMAT: 4d7f2

mark-id - INTEGER / empty 4d7f2a

If this parameter is empty, then all marked characters in the specified file will no longer standout. 4d7f2a1

read-marks(FILE*, mark-id -> mark-id-list) 4d7g

This procedure enables a process to determine which (sub-)strings are currently standing out. 4d7g1

FORMAT: 4d7g2

mark-id - INTEGER / empty 4d7g2a

if this parameter is empty, then it refers to all marks for the indicated file. 4d7g2a1

mark-id-list - LIST(LIST(mark-id, string-id, string-pos), ...) 4d7g2b

Picture Manipulation 4d8

This branch will be filled in later as we can more knowledge in the area of pictures. 4d8a

Input Control Manipulation 4d9

Send-Coors-with-action(FILE*, action-list) 4d9a

Don't-send-Coors-with-actions(FILE*) 4d9b

Report-Coors(FILE*, cursor-window-id, on-off, criteria-list) 4d9c

Report-mouse-button-status(FILE*, criteria-list) 4d9d

Input 4d10

read-byte(FILE* => byte-value) 4d10a

This procedure will read a character from the specified file. 4d10a1

FORMAT: 4d10a2

byte-value - CHARSTR 4d10a2a

write-byte(FILE*, byte-value) 4d10b

This procedure will write a character on the specified file. Characters written on terminal files by this procedure will appear in windows that have been designated to receive teletype output. 4d10b1

FORMAT: 4d10b2

byte-value - CHARSTR 4d10b2a

This parameter specifies the new value for the selected byte. 4d10b2a1

read-string(FILE*, termination-condition => string-value) 4d10c

This procedure will read a number of successive characters from the specified file. 4d10c1

FORMAT: 4d10c2

temination-condition - LIST(%count%INTEGER, %chars%STRING) 4d10c2a

This parameter specifies that either count bytes are to be read if chars is the null string; or that bytes are to be read until one of the bytes matches any byte in the chars string if a zero count is specified; or read bytes until a byte is encountered that matches any of the bytes in the chars string if both count and chars are specified. 4d10c2a1

string-value - CHARSTR 4d10c2b

This result is the value of the selected bytes. 4d10c2b1

write-string(FILE*, termination-condition, string-value) 4d10d

This procedure will write a number of successive characters on the specified file. Characters written on terminal files by this procedure will appear in windows that have been designated to receive teletype output. 4d10d1

FORMAT: 4d10d2

teminationcondition - LIST(%count%INTEGER, %chars%STRING) 4d10d2a

This parameter specifies that either count bytes are to be written if chars is the null string; or that bytes are to be written until one of the new bytes matches any byte in the chars string if a zero count is specified; or write bytes until a byte is encountered in the new string that matches any of the bytes in the chars string if both count and chars are specified. 4d10d2a1

string-value - CHARSTR 4d10d2b

This is the value for the new string. 4d10d2b1

Select-char(FILE*, coors => window-id, atom-id, char-pos, wcoors) 4d10e

This procedure accepts a FILE* and coordinates relative to the file and converts them to a window-id, atom-id, and line and character position within that string, and to coordinates relative to the selected window. Only strings that are hit sensitive will be considered as possible selection strings. 4d10e1

FORMAT: 4d10e2

coors - LIST(%x% INTEGER, %y% INTEGER) 4d10e2a

char-pos - LIST(%line-number% INTEGER, %character-position% INTEGER) 4d10e2b

wcoors - LIST(%relative-x-position% INTEGER, %relative-y-position% INTEGER) 4d10e2c

Select-string(FILE*, coors => window-id, atom-id, wcoors) 4d10f

This procedure accepts a FILE* and coordinates relative to the file and converts them to a window-id and a atom-id, and to coordinates relative to the selected window. Only strings that are hit sensitive will be considered as possible selection strings. 4d10f1

FORMAT: 4d10f2

coors - LIST(%x% INTEGER, %y% INTEGER) 4d10f2a

wcoors - LIST(%relative-x-position% INTEGER, %relative-y-position% INTEGER) 4d10f2b

select-window(FILE*, coors => window-id, wcoors) 4d10g

This procedure accepts a FILE* and coordinates relative to the file and converts them to a window-id and to coordinates relative to the selected window. Only windows that are hit sensitive will be considered as possible selection windows. 4d10g1

FORMAT: 4d10g2

coors - LIST(%x% INTEGER, %y% INTEGER) 4d10g2a

wcoors - LIST(%relative-x-position% INTEGER, %relative-y-position% INTEGER) 4d10g2b

Global and Miscellaneous Procedures 4d11

process-batch(procedure-list) 4d11a

This procedure allows a process to "batch" a group of procedure calls into one transmission. Any results from any of the procedures within the batch will be lost. 4d11a1

FORMAT: 4d11a2

procedure-list - LIST(LIST(pname, pargs), ...) 4d11a2a

pname - INTEGER 4d11a2b

the name of the procedure to be called 4d11a2b1

pargs - any 4d11a2c

these are the parameters to be passed to the procedure pname. 4d11a2c1

write-error-message(error-string) 4d11b

This procedure enables a process to place a message in the error window. 4d11b1

FORMAT: 4d11b2

error-string - CHARSTR 4d11b2a

write-status-message(status-string) 4d11c

This procedure enables a process to place a message in the status window. 4d11c1

FORMAT: 4d11c2

status-string - CHARSTR 4d11c2a

read-terminal-file-characteristics(FILE* =>
file-characteristics-list) 4d11d

This procedure enables a process to determine the logical class and the physical characteristics of the terminal to which file-id refers. 4d11d1

FORMAT: 4d11d2

file-characteristics-list - LIST( ) 4d11d2a

The value of this list will be specified later; however, it will probably include such things as the logical terminal class, whther or not the terminal has lower case, etc. 4d11d2a1

set-terminal-file-characteristics(FILE*,
new-file-characteristics-list) 4d11e

This procedure enables a process to modify some of the physical characteristics of the terminal to which file-id refers. 4d11e1

FORMAT: 4d11e2

file-characteristics-list - LIST( ) 4d11e2a

The value of this list will be specified later;

however, it will probably include such things as tab stops, etc. 4d11e2a1

reset-process-terminal-file(process-id) 4d11f

This procedure will reset the PIF and POF files for the specified process back to their initial state (see above -- xxx). 4d11f1

FORMAT: 4d11f2

process-id - INTEGER 4d11f2a

The OSI Miscellaneous Support Package - OMSP 5

Timing 5a

get-current-time (=> time) 5a1

This procedure will return the current date and time in a canonical form (probably similar to the internal form used by TENEX). 5a1a

FORMAT: 5a1a1

time - INTEGER 5a1a1a

convert-time-to-string (time, format-qualifiers => time-string) 5a2

This procedure will convert the passed date and time (in canonical form) to a string as specified by format-qualifiers. 5a2a

FORMAT: 5a2b

time - INTEGER / empty 5a2b1

If this parameter is empty, then the current date and time will be used. 5a2b1a

format-qualifiers - LIST( ) 5a2b2

this parameter will be specified later when we get more knowledge. 5a2b2a

time-string - CHARSTR 5a2b3

convert-string-to-time (time-string, format-qualifiers -> time) 5a3

This procedure will convert the passed date and time string to canonical form. 5a3a

FORMAT: 5a3b

time-string - CHARSTR 5a3b1

format-qualifiers - LIST( ) 5a3b2

This parameter will be specified later when we gain more knowledge; however, we expect it to contain elements that describe the content and format of the passed string. 5a3b2a

time - INTEGER 5a3b3

Buffers 5b

The following primitives provide a process with storage managment capabilities. It is the process' responsibility not to write outside the limits of an allocated buffer. 5b1

Initialize-buffer-pool(pool-address, pool-size, parm -> pool-id) 5b2

This procedure informs the OSI of the virtual address space within a process that is to be used for buffers. The pool-id returned should be used in future calls to allocate and delete individual buffers. 5b2a

FORMAT: 5b2b

pool-address - INTEGER 5b2b1

pool-size - INTEGER 5b2b2

This is the total size of the buffer pool. 5b2b2a

parm - LIST(type, size, gcol) 5b2b3

This parameter indicates if all the individual buffers are to be fixed size (and their size) or if the individual buffers are going to be variable in size (and the maximum size of any individual buffer). The element gcol must be specified for variable size pools. If gcol is IMPLICIT and a subsequent

allocate-buffer would fail because it could not allocate the requested buffer, a garbage collection will take place and an appropriate SIGNAL will be generated. During this garbage collection, no "fixup" of the contents of any of the buffers will take place. If gcol is EXPLICIT, then subsequent allocates will fail if the requested buffer size cannot be allocated. 5b2b3a

type = BOOLEAN [FIXED=TRUE / VARIABLE=FALSE] 5b2b4

size = INTEGER 5b2b5

gcol = BOOLEAN [IMPLICIT=TRUE / EXPLICIT=FALSE] / empty 5b2b6

pool-id = INTEGER 5b2b7

Allocate-buffer(pool-id, size, dest -> buffer-id) 5b3

This procedure will allocate a fixed size buffer, or a buffer of the requested size in variable size pools, in the specified buffer pool. The buffer-id returned is the address of cell that contains the address the first usable word in the allocated buffer. The address of any specific buffer can change due to a garbage collection. A error will be generated if no more buffers exist, or if the requested size buffer cannot be allocated. If the special pool-id of -1 is used then storage will be obtained from the monitor and placed in the process' address space at location dest. 5b3a

FORMAT: 5b3b

pool-id = INTEGER 5b3b1

size = INTEGER / empty 5b3b2

dest = INTEGER 5b3b3

buffer-id = INTEGER 5b3b4

this result is the address that the process can start writing in the buffer at. 5b3b4a

Delete-buffer(buffer-id) 5b4

This procedure returns a previously allocated buffer to the buffer pool. 5b4a

FORMAT: 5b4b

buffer-id - INTEGER 5b4b1

Garbage-collect-pool(pool-id) 5b5

This procedure will garbage collect a buffer pool. This procedure only has real meaning for varable size buffer pools. This procedure will not do any "fixup" on the contents of indiviual buffers. 5b5a

FORMAT: 5b5b

pool-id - INTEGER 5b5b1

KEV 21-JAN-75 10:34 25156

second draft of OSI document

(J25156) 21-JAN-75 10:34;;;; Title: Author(s): Kenneth E. (Ken) Victor/KEV; Distribution: /NPG([ACTION]) RWW([INFO-ONLY]) ; Sub-Collections: SRI-ARC NPG; Clerk: KEV; Origin: < VICTOR, O-S-I,NLS;3, >, 21-JAN-75 09:55 KEV ;;;;####;

KEV 21-JAN-75 11:54 25157

John Pickens and distributed debugging

this consists of 3 smdmsgs i have received from john about his thesis on distributed debugging

KEV 21-JAN-75 11:54 25157

John Pickens and distributed debugging

17-JAN-75 1524-PST PICKENS at USC-ISIB: NSW Debugging
Distribution: VICTOR AT SRI-ARC, pickens at isib
Received at: 17-JAN-75 15:33:59 1

Ken, 1a

Jim indicated to me that you were involved in developing debugging tools for NSW. I've read the description of the Low Level Debugger.
Are you envisioning any tools that are at higher levels? (i.e. Middle Level Debugger) 1b

I'm into debugging distrbuted processes, but at a more abstract level. Distributed DDT is certainly one valuable tool, but others
that I am interested in are 1) Using inter-module communications to set breakpoints (which involves defining a communications description
language), and 2) Debugging from some sort of machine readable design specification of distributed process structure. 1c

Let me know what your interests are and maybe we can talk some more. 1d

John Pickens 1e

20-JAN-75 1330-PST PICKENS at USC-ISIB: Warning! This is big!
Distribution: VICTOR AT SRI-ARC
Received at: 20-JAN-75 13:39:20 2

Distributed Debugging Techniques Page 1
John R. Pickens 1 January 1975 2a

Introduction 2b

The goal of this thesis is to bring together the areas of debugging, distributed process structuring and modular programming in order to outline useful tools for dynamic debugging and monitoring. There is no reason to expect that the final thesis will conform chapter by chapter to the material here outlined, but it is presented in this manner anyway to give the

writer a glimmer of hope that the end may someday be achieved. 2c

In the first chapter the thesis surveys current work on distributed computing, modular programming, and debugging. (The conciseness of this chapter's definition is certainly misleading.) 2d

In Chapter II a Meta Communications Description Language is defined which has among its features the following attributes: 2e

1) One or more communications streams (or channels) may be monitored, interpreted, and correlated at a level potentially very high compared to the normally available bit and byte level, 2f

2) The descriptive mechanism is essentially that of a human engineered BNF in which the user-defined grammar describes in meta-rules the syntax of single channel communications, 2g

3) Local storage in the form of partially enumerated meta-rules as well as integer and character variables is available for user manipulation, 2h

4) Multiple channel communications may be correlated through the use of State-Variables and State-Changing constructs, 2i

5) Semantics may be associated with the Meta-rules which validates user defined conditions (e.g. value out of range), 2j

6) Breakpoints and breakpoint processing are associated

with combinations of meta-rules and are used both in semantic validation and causing formatted output to the user or backup storage. 2k

7) The communications description language contains macro programming and library storage constructs to relieve the ultimate burden on the user for creating meta-rules and

 21

validation/monitor/debug programs. 2m

Chapter III applies the techniques of Chapter II to several example applications and extends the discussion to cases in which more than two channels must be monitored. Three environments which come to mind are 1) various modules programmed in DCS, 2) the communications protocols for processes within the National Software Works, and 3) defining a distributed machine whose control is specified via a Graph Model of Computation (a la Gostelow). The goal of this latter example is not so much to define an efficient GMC machine as it is to demonstrate that the macro-descriptive facilities are powerful enough to enable monitoring and debugging in units higher than individual messages, e.g. token flow and vertex initiation. 2n

Chapter IV expands the discussion of communications description to modules in general. It is shown that useful descriptive mechanisms can be defined for other than string

formatted arguments. The technique of affiliating syntax and semantics checking with module activation is shown to be a practical realization of checking programs dynamically by assertions (James King's paper in Debugging Large Scale Systems). The primary requirement is that traps may be placed at the level of communications between modules. This dynamic binding of traps and monitoring is shown to be far superior to the extant techniques of compiling in monitor instructions. Chapter IV also discusses the use of emulation to check the module's stored behavior. 2o

Chapter V raises the issues associated with distributed processes. The details are not known yet, but questions as the following might be discussed: 1) Can the monitor be distributed itself? 2) What happens to the notion of time? 3) What extra facilities may be required because of the loose coupling of the processes? 3) At what level might distributed operating systems be monitored? 2p

Chapter VI discusses how the previous proposed tools might be integrated with more conventional tools. For example, distributed DDT (and its high level language counterparts) should exist to allow evaluation of modules at the sub-module level. Issues exist as to how this function might be distributed and what kind of an interface it should have to the user (ideally the interface should be near the level of the source language the module was written in). The transition between

inter-module monitoring and sub-module DDTing should be smooth. It should also be possible,for example, to activate sub-module breakpoints as a function of inter-module communications conditions. 2q

Somewhere in the thesis cost will have to be discussed. The traditional tradeoff between monitoring/debugging and free-running execution is speed. The ideal debugging system is one which implies zero overhead on the operating programs.

Distributed Debugging Techniques Page 3
John R. Pickens 1 January 1975 2r

Naturally this issue exists for the techniques proposed in this thesis. It should be determined if there are any environments in which the overhead is negligible. In some cases it may be possible to keep up with real time (such as in a hardware meta-message composer and analyzer). In others it will be necessary to impose the constraint that the intercommunications between modules be asynchronous and amenable to slowed message rates. 2s

A Meta Communications Description Language Page 1
John R. Pickens 1 January 1975 2t

Introduction 2u

One very basic tool for debugging processes which communicate via messages is a Communications Description Language. As defined here the language is interactive, programmable, and capable of expressing most interprocess communications. It's form is that of

a BNF modified to include dimensional data declarations and state validation/changing information. The fundamental programmable elements of the language are the grammar declarations (GD) which contain dimensional, type, and meta-linguistic declarations, and the partial enumerations (PE's) which affiliate symbolic labels with partial enumerations for meta-rules defined within the GD. 2v

The language is meant to be used both to synthesize and to analyze communications streams. For synthesis a structured editor coupled with the PE's and GD allows flexible construction of both partial and complete messages. For analysis the PE's and GD allow basic syntax checking and breakpoint synchronization on incoming data. Additional semantics routines allow more sophisticated analysis of the communications as well as primitive emulation mechanisms. 2w

The Meta Communications Description Language contains several features which won't be described in detail as of yet. Summarized they are: 2x

1) Local storage is available for character strings and integer variables. 2y

2) Macro programming facilities and backup library storage exist. 2z

3) Language constructs exist to control the setting of breakpoints, breakpoint processing, and output formatting. 2a@

4) Semantics processing may be associated with both the

GD's
and PE's, 2aa

Grammar Declarations 2ab

The root of the entire communication description language is in the declarations. The Grammar Declarations portion is composed of the following elements:

 2ac

1) A set of BNF Meta-rules and Meta Symbols to define the structure of the communication strings. No particular enumerations exist within the Grammer Declarations (i.e. no terminals). 2ad

2) Dimensional declarations. Since most communications by their nature are either fixed length fields or variable length fields (derivable from another fixed field) it is necessary to declare (where appropriate) the dimensions of the non-terminal symbols. The two types of dimensional declarations are fixed (e.g. n Bytes, n Words) and variable (e.g. fixed length "length" field followed by variable length "data" fields). 2ae

3) Type declarations. In some cases it may be desirable to associate data types with non-terminals. This feature would probably be most useful for formatted output but might also be useful in writing programs to manipulate the messages. 2af

4) State Information. The entities which correlate multiple channels are state variables. A channel's state may be

required in some cases to correctly interpret the syntax of its messages. The two types of state veriable constructs in the Grammar Declarations are a) current state and b) next state. Current state is used as part of the syntactic recognition of messages. Next state is the new state achieved after all processing associated with the metarule is completed. For now all state variables are global. Eventually it may desirable to define sub-states or states local to individual channels. 2ag

5) Special recognition metasymbols (i.e. <<metasymbol>>)
This construct allows the message parser to differentiate those fields which are required for syntactic recognition from those which aren't. In a particular metarule only a few fields (if more than one) contribute to the recognition process. Generally the most that is required is an opcode field. The other fields within the metarule identify syntactic units which are not required for recognition and, therefore, need not be qualified by scanning partial enumerations (unless a particular message is being searched for). By introducing special metasymbols (symbols surrounded by an extra pair of brackets, see the example above) the recognizer which processes message streams can differentiate between invalid messages (no matching opcode) and unnumerated messages. (See the enumerations for <MSG> in example 1, Appendix C.)

6) Constructs for iteration (see example 3 in Appendix C).
In order to describe variable format fields (i.e. keyword positioning as opposed to fixed field positioning) iteration is required. With iteration it should be possible to describe any structure representable by recursion in a form well suited for human consumption. Recursion is ruled out as it tends to be incomprehensible for casual consumption, and one of the goals is that the description language be interactive and well human engineered. 2ai

The precise syntax associated with the Grammar Declarations is stated in Appendix A. 2aj

Partial Enumerations 2ak

All defined terminal symbols associated with the Grammar Declarations are contained in the Partial Enumerations. The PE structure is defined to be general enough that it may be used both for synthesis and analysis and may contain only partially defined rules. Every non-terminal symbol in the GD has potential entries within the PE. 2al

Each partial enumeration contains as minimal information the non-terminal which it enumerates, a symbolic identifier to identify this enumeration, and a value, i.e. the partially enumerated right side of the referenced meta-rule. [To enumerate is simply to replace non-terminal symbols with specific strings. In a

general meta-rule not all of the referenced non-terminals need be enumerated.] Two other items associated with a particular enumeration are: 2am

1) An indication of the enumeration's environment. Some enumerations may be applicable for synthesis, some for analysis, 2an

2) Context qualification. In some cases referral to a meta rule will be ambiguous unless higher levels of the parsed rule are delineated. For example, a meta-rule which describes a filename may be used in more than one type of communication string, and it may be desired to define non intersecting groups of enumerations based on the category of message. In such cases qualification is required to identify the higher level meta-rules which uniquely specify the desired context (See Appendix B). 2ao

In general the value of a particular metarule enumeration is a mixture of metasymbols extant in its GD definition, literial

 2ap

entries (e.g. numeric or alphameric), symbolic labels for other partial enumerations, and , significantly, descendant metarules as defined within the GD. As an illustration of the latter consider the following set of rules (see also the examples in Appendix C): 2aq

<A> ::= <B> <C>

<C> ::= <D>
::= <E> 2ar

An enumeration for <A> might be; 2as

Enumeration-A1 = <B> <D> 2at

Or, another might be; 2au

Enumeration-A2 = <B> <E> 2av

Appendix B contains more detailed specifications of partial enumerations. 2aw

Editing 2ax

Rule construction and editing is highly structured. In defining partial enumerations, for example, the user is given not a sequential string editor, but rather a structured editor. His options for selection, composition, and replacement are in terms of the defined meta rules. For example, to enumerate a given metarule the user might type something like ENUMERATE <metarule x> RETURN. In response he would see the metarule's unenumerated definition. At this point he would have several editing options: e.g. [SPACE] skips to the next meta symbol within the rule, [?] lists the current enumerations for the rule, [LF] jumps one level lower into the definition of the current metasymbol, a numeric selects one of the displayed enumerations for editing, [R] begins a literal replacement of the current metasymbol, etc. The exact definition of the editor is not important at this stage, but it is important to point out that the structured presentation of the message syntax to the user is one of the more useful features of the

communications description language.

A Meta Communications Description Language Page 5
John R. Pickens 1 January 1975 2ay

Appendix A -- Syntax of Grammar Declarations 2az

General Comments 2b@

Grammar declarations are BNF-like meta-rules which are composed of meta-symbols, data declarations, and state information. Each declaration is composed of a single Left Side meta-symbol (i.e. context free), followed by the assignment symbol "::=", followed by one or more meta-symbols or, in the case that the meta-rule is at the lowest syntactic level, a group of data declaration keywords. Metasymbols are denoted by surrounding valid identifiers (definition left vague for now) with brackets, three examples are: 2ba

<MESSAGE>, <OPERAND>, <LENGTH> 2bb

Multiple alternatives for metarules are indicated by writing each alternative on a new line in the same format as the original metarule, but minus the metarule head symbol. An example follows: 2bc

```
<A>  ::= <B>
     ::= <C>
```
2bd

Data Declarations 2be

Metarules which are nearest the message data fields are termed data declaration metarules. The form of a data declaration metarule is as follows: 2bf

<metasymbol> ::= "type" "number of units" "dimensional unit" 2bg

"type" is optional, but the next two fields are required. Some examples of possible data types are INTEGER, HEX, CHARACTER, and ANY which denote integer, hexidecimal, character, and "don't care" data respectively. 2bh

The next field can be either a fixed numeric or a simple expression. In case of expressions the appearance of a metasymbol implies that the symbol's value is to be computed from the current string (it's type must be integer). Addition, subtraction, multiplication and division are supported. An example of a variable length value is as follows:

A Meta Communications Description Language Page 6
John R. Pickens 1 January 1975 2bi

<FILENAME> ::= CHARACTER (<LENGTH>=2) BYTES 2bj

The final field identifies the units in which the length is measured. Some possible units are BITS, HEXDIGITS, BYTES, and WORDS. 2bk

State Information 2bl

State validation and state changing information may also be associated with a metarule. The construct ":STATE=" requires that the current state be identical to the one indicated or else recognition will not occur. Logical conditions on one or more state variable are allowed. The construct ":NEWSTATE=" changes the state to the new state indicated after processing for this metarule has completed. Both constructs are optional and would probably only be used to coordinate the actions of multiple

communications
channels. A simple example follows: 2bm

```
<INQUIRY>      ::= <QUERY> <A>      :NEWSTATE = WAIT-A
                   <QUERY> <B>      :NEWSTATE = WAIT-B          2bn

<RESPONSE-A>   ::= <ANSWER-STRING>  :STATE = WAIT-A
<RESPONSE-B>   ::= <ANSWER-STRING>  :STATE = WAIT-B             2bo
```

Iteration 2bp

Iteration may be used to describe multiple instances of a
metarule. An iteration may be defined either as having upper and
lower bounds or as occuring an exact number of times. In either
case the iteration specification immediately follows the
metasymbol. Numeric fields may be any valid numeric (including
evaluated metasymbols). The form of the bounded iteration is: 2bq

<metasymbol> (lower,upper) 2br

The form of the fixed iteration is: 2bs

<metasymbol> ( )

 2bt

See example 3, Appendix C for an illustration of how iteration
might be applied.

 2bu

Appendix B -- Syntax of Partial Enumerations 2bv

Partial enumerations depend upon metarule declarations for their
basic structure, but in fact may appear quite different from their
unenumerated definition. They are distinguished from grammar
declarations by the detail which they contain.

Partial enumerations are really partial parse trees. The head node (which has a symbolic name) and first level descendants correspond to the left and right sides respectively of the original grammar declaration. But the first level descendants themselves have descendants. The most correct way to display a partial enumeration would be to actually show the partially enumerated parse tree. In most cases, however, it is satisfactory to show only the terminal nodes of the partial enumeration (See the example given above in the original discussion of partial enumerations). 2bw

The left side of an enumeration consists first of one or more metasymbols defining the context of the enumeration and second of a symbolic identifier. Normally only one metasymbol is denoted on the left side and it identifies the metarule which is being enumerated. In cases where more detailed context qualification is required the format is simply to list from left to right with the most distantly related metasymbol on the left all the required ancestors of this metarule. The general form for the left side of a partial enumeration is therefore; 2bx

< >...<g-parent>,<parent>,<this metarule> SYmbolic Name = 2by

One other item which may appear on the left side is an indication that the eumeration is to be used for analysis (:ANALYSIS) or synthesis (:SYNTHESIS , Default is both). This indication is useful in separating enumerations which are required only for synthesis of messages from those used only in analysis of messages. 2bz

As indicated previously, the right side contains enumerations for one or more of the metasymbols in the original metarule. An enumeration for a metasymbol is either symbolic (i.e. referring to the left side symbol of another partial enumeration) or literal (i.e. either the right side from another partial enumeration or actual message characters). Since the right side is really a partially enumerated parse tree it is possible to have an arbitrary mixture of symbolics and literals for each enumerated metasymbol.

Appendix C -- Examples 2ca

Examples 1 and 2 demonstrate two ways to define the same grammar. Example 3 describes a keyword oriented message and in the process, illustrates an application for iteration. In all three examples square brackets, [ ] ,are used to denote partial enumerations. As the GD metarule which the PE applies to is implied by position (i.e. the PE is immediately below the corresponding GD metarule) a further simplification is effected by not listing the GD metasymbol on the left side. Partial enumerations are listed with the grammar declarations for convenience only. 2cb

Comment on First Two Examples 2cc

The message grammar which is described by the first two examples is quite simple. A message in this grammar consists of a 1 byte length field followed by a one byte opcode field followed by a variable length operand field. The two specific messages

defined
are OPENFILE (opcode = 1, operand = filename) and READFILE (opcode
= 2, operand = disk address). Examples 1 and 2 both describe the
same grammar, but example 1 is more consise and easier to read.
The first example conforms more closely to the fields of the
message with all higher level interpretations appearing in the
partial enumerations. The second example attempts to bring the
OPENFILE and READFILE groupings into the grammar declarations. The
first method appears to be preferable from the point of view of
simplicity, though a final judgement cannot be made at this time. 2cd

Example 1 2ce

<MSG> ::= <LEN> <<OPCODE>> <OPERAND> 2cf

[OPENFILE = <LEN> 01 <FILENAME>]
[READFILE = <LEN> 02 <DISKADDRESS>] 2cg

<LEN> ::= INTEGER 1 BYTE
<OPCODE> ::= INTEGER 1 BYTE
<OPERAND> ::= <FILENAME>
<DISKADDRESS>
<FILENAME> ::= CHARACTER (<LEN> - 2) BYTES
<DISKADDRESS> ::= INTEGER 2 BYTES

 2ch

Example 2 2ci

<MSG> ::= <OPENFILE>
::= <READFILE>
<OPENFILE> ::= <LEN> <<OPCODE>> <FILENAME>
<READFILE> ::= <LEN> <<OPCODE>> <DISKADDRESS>
<OPCODE> ::= INTEGER 1 BYTE 2cj

[<OPENFILE>.<OPCODE> OPENOP = 01]
[<READFILE>.<OPCODE> READOP = 02]
<LEN> ::= INTEGER 1 BYTE

<FILENAME> ::= CHARACTER (<LEN> - 2) BYTES
<DISKADDRESS> ::= INTEGER 2 BYTES 2ck

Example 3 2cl

The message grammar described by this example is also fairly simple. A message may consist of from 0 to 3 instances of either of two types of fields. No order is imposed and the fields differ in length and type. Each field type is distinguished by a single byte key. Iteration is required to represent this grammar. It is assumed that some mechanism exists for determining the beginning and end of a message. The definition follows: 2cm

<MSG> ::= <UNIT> (0,3)
<UNIT> ::= <KEY> <DATA> 2cn

[UNIT1 = KEY1 <FIELD-TYPE-1>]
[UNIT2 = KEY2 <FIELD-TYPE-2>] 2co

<KEY> ::= INTEGER 1 BYTE 2cp

[KEY1 = 00]
[KEY2 = 01] 2cq

<DATA> ::= <FIELD-TYPE-1>
::= <FIELD-TYPE-2>

 2cr

Appendix D -- Metadescription of DCS I/O Handler 2cs

This appendix gives a fairly detailed definition of the message interface to the UCI DCS I/O Handler (IOH). A more detailed specification for IOH may be found in "DISTRIBUTED COMPUTER OPERATING SYSTEM, Programming Guide" (Internal document to the Dept. of Information and Computer Science at UC Irvine). However, it is suspected that the following definition is more

readable as
to the IOH message structure. Due to the requirement of
conciseness only three IOH commands are defined, but they are
chosen to illustrate a wide range of message formats allowed within
the IOH. Following the definition of the IOH message grammar is an
example of a completely parsed message showing the derived parse
tree. 2ct

```
<MSG> ::= <<DPN>> <<OPN>> <LEN> <BODY>                                  2cu

[ IOH-REQUEST = IOH <OPN> <LEN> <IO-REQUEST>   ]
[ IOH-RESPONSE = <DPN> IOH <LEN> <IO-RESPONSE> ]                        2cv

<DPN> ::= <PN>
<OPN> ::= <PN>
<PN>  ::= <CLASS> <MACHINE> <SEQUENCE>                                  2cw

[ NUCLEUS  = 1    <MACHINE>   01     ]
[ IOH      = 1    <MACHINE>   02     ]
[ CP       = 1    <MACHINE>   03     ]
[ SEQUENCE = 1    <MACHINE>   04     ]
[ CHECKER  = 1    <MACHINE>   05     ]                                  2cx

<CLASS>    ::= INTEGER 1 HEXDIGIT
<MACHINE>  ::= INTEGER 1 HEXDIGIT
<SEQUENCE> ::= INTEGER 1 BYTE                                           2cy

<LEN> ::= INTEGER 2 BYTES                                               2cz

<BODY> ::= <IO-REQUEST>
       ::= <IO-RESPONSE>
       ::= ANY (<LEN>) BYTES                                            2d@
```

 2da

```
<IO-REQUEST> ::= <LFN>  <<OPCODE>>  <ARGUMENT>                          2db

[ READ-SYM  = <LFN> OPC-RS <IO-COUNT>
;NEWSTATE=READ]
[ OPEN-FILE = <LFN> OPC-OF <OPEN-FLAGS> <FREE-FORM>
;NEWSTATE=OPEN]
[ CLOSE-FILE= <LFN> OPC-CF <CLOSE-FLAGS>
```

```
:NEWSTATE=CLOSE]
        .
        .
        .
        ETC.                                                        2dc

      <LFN> ::= INTEGER 1 BYTE                                      2dd

      <OPCODE> ::= INTEGER 1 BYTE                                   2de

      [ OPC=RS = 00 ]
      [ OPC=OF = 14 ]
      [ OPC=CF = 25 ]
         .
         .
         .
         ETC.                                                       2df

      <ARGUMENT> ::= <IO=COUNT>
                 ::= <OPEN=FLAGS> <FREE=FORM>
                 ::= <CLOSE=FLAGS>                                  2dg

      <IO=COUNT> ::= INTEGER 2 BYTES                                2dh

      <OPEN=FLAGS> ::= HEX 2 BYTES                                  2di

      [ OCORD = 00 ]
      [ OCOWR = 01 ]
      [ OCOWN = 02 ]
         .
         .
         .
         ETC.                                                       2dj

      <CLOSE=FLAGS> ::= INTEGER 2                                   2dk

      [ CLOSE=EOF    = 0]
      [ CLOSE=NO=EOF = 1]                                           2dl

      <FREE=FORM> ::= CHARACTER (<LEN> = 4) BYTES

      A Meta Communications Description Language    Page 13
      John R. Pickens                          1 January 1975       2dm

      <IO=RESPONSE> ::= <LFN> <CC> <VALUE>  :NEWSTATE=IDLE          2dn

      [ :STATE=READ  READ=REPLY = <LFN> <CC> (ANY (<LEN>=2) BYTES
]
      [ :STATE=OPEN  OPEN=REPLY = <LFN> <CC> <OPEN=FLAGS>
```

John Pickens and distributed debugging

```
    <FIXED-FORM>]
         [ :STATE=CLOSE CLOSE-REPLY= <LFN> <CC> NULL
    ]
              .
              .
              .
              ETC.                                                    2do

         <CC> ::= INTEGER 1 BYTE                                      2dp

         [ ECNRM =0                                                  ]
         [ ECDNR =1   :STATE,NOT="RESERVE,OR,OPEN,OR,TRANSFER        ]
              .
              .
              .
              ETC.                                                    2dq

         <VALUE> ::= ANY (<LEN> -2) BYTES
                 ::= <OPEN-FLAGS> <FIXED-FORM>
                 ::= NULL                                             2dr

         <FIXED-FORM> ::= <DEV-NAME>
                          <FILE-NAME>
                          <FILE-EXTENSION>
                          <DIR-NAME>
                          ANY 2 BYTES
                          <WILD-MASK>
                          <RFN>
                          ANY 5 BYTES                                 2ds

         <DEV-NAME> ::= CHARACTER 6 BYTES                             2dt

         [ DCS     ::= 'DCS'                                         ]
         [ DSKAON  ::= 'DSKA0'    CHARACTER 1 BYTE                   ]
         [ LPT0    ::= 'LPT0'                                        ]
              .
              .
              .
              ETC.                                                    2du

         <FILE-NAME>       ::= CHARACTER 6 BYTES
         <FILE-EXTENSION>  ::= CHARACTER 2 BYTES
         <DIR-NAME>        ::= CHARACTER 6 BYTES
         <WILD-MASK>       ::= CHARACTER 2 BYTES
         <RFN>             ::= INTEGER   1 BYTE                      2dv

20-JAN-75 1328-PST  PICKENS at USC-ISIB:  My stuff
   Distribution:  VICTOR AT SRI-ARC, pickens
   Received at:   20-JAN-75 13:40:42                                    3
```

KEV 21-JAN-75 11:54 25157

John Pickens and distributed debugging

I'm sending you a large file with two papers in it. One is a sketch of
what my thesis may (!) contain (nothing in it is absolutely guaranteed);
the other is my first pass at a communications description language.
The last appendix in the latter paper refers to a module in the UC Irvine Distributed Computing System, which I have given no details about in the paper. So don't worry about understanding that example. 3a

I'm still trying to get in my mind a clear overall picture of my project. When it all solidifies I'll have a good top-level document to give to you. But until then... 3b

I'll read your stuff today. 3c

John 3d

KEV 21-JAN-75 11:54 25157

John Pickens and distributed debugging

(J25157) 21-JAN-75 11:54;;;; Title: Author(s): Kenneth E. (Ken) Victor/KEV; Distribution: /NPG([INFO-ONLY]) RWW([INFO-ONLY]) ; Sub-Collections: SRI-ARC NPG; Clerk: KEV;

DVN RWW POOH 21-JAN-75 12:45 25158

Maintenance of NLS-8 Documentation

Just as working on NLS software is divided into a developmental stage and an application stage, so documentation may be divided. Up to now the development group has handled both stages of NLS-8 documentation. After the beginning of February commitment to NSW will force us to hand maintenance of NLS-8 documentation over to Applications with the exception of a few files listed below of special interest to development programming. 1

In general this obligation means that Application must maintain the off and online supplies of user documents, and Help if they want to offer them to their customers. In certain cases Documentation undertakes to complete some unit of work on a document even if work should trail after February 3. This change involves problems of communication and updating documentation similar to those that Dave Hopper co-ordinates for software. For example any changes made in NLS-8 documentation will not appear in the NLS-9 documentation that will eventually replace it unless some one makes them. We need to work out appropriate procedures to keep NLS-9 documentation up to date with such changes as are appropriate. 2

The following list spells out in detail what Development will stop doing. 3

MASTER DOCUMENTATION LIST 3a

For each document listed below, the following information is provided: 3b

a brief description of the docment 3b1

Online and offline status 3b2

offline title: 3b2a

online location: 3b2b

Relevant system: 3b3

Size in pages 3b4

offline: 3b4a

online: 3b4b

Intended audience: 3b5

Caretaker: 3b6

Up to dateness: 3b7

Fate: 3b8

HELP 3c

a complete explanation of NLS commands, terms and concepts 3c1

Online and offline status: 3c2

offline title: Glossary (to be printed) 3c2a

online location: <documentation,help,> 3c2b

System: NLS-8 3c3

Size in pages: 3c4

offline: 167 3c4a

online: 250 3c4b

Caretaker: ? 3c5

HELP is kept as up to date a possible for running version of NLS. 3c6

Fate: When the revisions in HELP necessary to the Glossary are finished, we will hand over the file to Applications. 3c7

TNLS-8 Primer 3d

for TNLS-8 users, a simple scenario (with additional information) for creating a file and journalizing it. 3d1

Online and offline status 3d2

offline title: TNLS-8 Primer 3d2a

online location: <userguides,primer,> 3d2b

Relevant system: NLS-8 3d3

Size in pages 3d4

offline: 13 3d4a

online: 16 3d4b

Intended audience: naive users 3d5

Caretaker: ? 3d6

Up to dateness: revised 10-1-74 3d7

Fate: Dirk will wear his DCPS hat and produce a Com version and turn this over to Applications . 3d8

DNLS Viewspecs Card -- Mouse and Keyset Codes and Cases 3e

a wallet size table of mouse and keyset input code commands 3e1

Online and offline status 3e2

offline title: Mouse and Keyset, Codes and Cases 3e2a

Relevant system: NLS-8 3e3

Size in pages 3e4

offline: both sides of a wallet size card 3e4a

Intended audience: all users 3e5

Caretaker: Beck 3e6

Up to dateness: revised 12/1/74 3e7

Fate: We will hand over updating and maintenance of offline copies and online files to Applications. 3e8

TNLS-8 Cue Card 3f

multi-color reference list of most frequently used TNLS-8 commands 3f1

Online and offline status 3f2

offline title: TNLS-8 Quick Reference Guide 3f2a

Relevant system: NLS-8 3f3

Size in pages 3f4

offline: 2 3f4a

Intended audience: all users 3f5

Caretaker: Beck 3f6

Up to dateness: revised 10/1/74 3f7

Fate: We will hand over updating and maintenance of offline copies and online files to Applications. 3f8

Userguides Glossary 3g

alphabetical listing of items in the Help Data Base 3g1

Online and offline status 3g2

offline title: to be printed 3g2a

Relevant system: NLS-8 3g3

Size in pages 3g4

offline: 168 3g4a

online: 182 3g4b

Intended audience: all users 3g5

Caretaker: Kirk 3g6

Up to dateness: being revised 3g7

Fate: When the glossary is completed to the point of camera ready copy, we will hand over updating and maintenance of offline copies and online files to Applications. 3g8

Userguides Commands 3h

complete list of NLS commands in formal syntax 3h1

Online and offline status 3h2

offline title: NLS-8 Command Summary 3h2a

online location: <USERGUIDES,COMMANDS,> 3h2b

Relevant system: NLS-8 3h3

Size in pages 3h4

offline: 29 3h4a

online: 26 3h4b

Maintenance of NLS-8 Documentation

Intended audience: all users 3h5

Caretaker: Beck 3h6

Up to dateness: revised 12/6/74 3h7

Fate: Dirk will wear his DCPS hat and produce a Com version and turn this over to Applications . 3h8

New and Changed Features in TNLS-8 (Advanced Copy 31039) 3i

prose explanation of command language changes from NLS-7 to NLS-8 3i1

Online and offline status 3i2

online location: <Journal,31039,> 3i2a

Relevant system: NLS-8 3i3

Size in pages 3i4

online: 3i4a

Intended audience: experienced TNLS-7 users 3i5

Caretaker: Jim Bair 3i6

Up to dateness: revised 10/4/74 3i7

userguides, Dex-Primer 3j

primer for Delayed Execution offline input system (Techtran version) 3j1

Online and offline status 3j2

offline title: DEX-Primer 3j2a

online location: <USERGUIDES,DEX-PRIMER,> 3j2b

Relevant system: NLS-7 or NLS-8 3j3

Size in pages 3j4

offline: 9 3j4a

online: 13 3j4b

Intended audience: naive DEX users 3j5

Caretaker: Leavitt 3j6

Up to dateness: revised 1/8/74 revision to be made for ICP Termicette 3j7

Fate: We will hand over updating and maintenance of offline copies and online files to Applications. 3j8

Userguides DEX-Transcription 3k

complete account of the DEX system for naive users. 3k1

Online and offline status 3k2

offline title: Deferred Execution (DEX) User Guide 3k2a

online location: <USERGUIDES,DEX-TRANSCRIPTION,> 3k2b

Relevant system: NLS-7 or NLS-8 3k3

Size in pages 3k4

offline: 40 3k4a

online: 41 3k4b

Intended audience: naive DEX users 3k5

Caretaker: ? 3k6

Up to dateness: revisions to be made to reflect other terminal 3k7

Fate: We will hand over updating and maintenance of offline copies and online files to Applications. 3k8

Userguide DEX-Two 31

explanation of Delayed Execution System for sophisticated offline editing of online files 311

Online and offline status 312

offline title: 312a

online location: <USERGUIDES,DEX-TWO,> 312b

Relevant system: NLS-7 or NLS-8 313

Size in pages 314

offline: 314a

online: 20 314b

Intended audience: people who want to get a general notion of DEX-TWO 315

Caretaker: Harvey 316

Up to dateness: to be brought up to date when running version becomes operational 317

Fate: We will hand over updating and maintenance of offline copies and online files to Applications. 318

Userguides Journal-Netsub 3m

guide on how to enter documentation into journal system using the network mail protocol. 3m1

Online and offline status 3m2

offline title: Scenario For Using The Network Journal 3m2a

online location: <USERGUIDES, JOURNAL-NETSUB> 3m2b

Relevant system: NLS-7 or 8 3m3

Size in pages 3m4

offline: 3m4a

online: 5 3m4b

Intended audience: all network users 3m5

Caretaker: ? 3m6

Up to dateness: revised 3/28/74 3m7

Fate: We will hand over updating and maintenance of offline copies and online files to Applications. 3m8

L10 Userguides 3n

userguide to L-10 programming language in NLS environment 3n1

Online and offline status 3n2

offline title: L10 Users' Guide 3n2a

online location: <USERGUIDES,L1o-GUIDE> 3n2b

Relevant system: NLS-8 3n3

Size in pages 3n4

offline: 29 3n4a

online: 179 3n4b

Intended audience: users who wish to write user programs and use content filter, and systems programmers 3n5

Caretaker: Dean 3n6

Up to dateness: Parts One and Two revised 10/20/74, Parts Three and Four to be revised, 3n7

Fate: We will hand over updating and maintenance of offline copies and online files to Applications, 3n8

Userguide Locator 3o

a file of links that points people to other online documents, 3o1

Online and offline status 3o2

online location: <USERGUIDES, LOCATOR,> 3o2a

Relevant system: NLS-8 3o3

Size in pages 3o4

online: 9 3o4a

Intended audience: all NLS users 3o5

Caretaker: ? 3o6

Up to dateness: revised 11/10/74 3o7

Fate: We will hand over updating and maintenance of offline copies and online files to Applications, 3o8

NDDT Symbolic Debugger Guide 3p

a description of the features of a symbolic debugging routine for L10 programs written for the NLS system. 3p1

Online and offline status 3p2

offline title: NDDT Symbolic Debugger Users' Guide 3p2a

online location: <USERGUIDES, NDDT-SYMBOLIC-DEBUGGER,> 3p2b

Relevant system: NLS-7 or NLS-8 3p3

Size in pages 3p4

offline: 13 3p4a

online: 14 3p4b

Intended audience: sophisticated L10 programmers 3p5

Caretaker: Ken Victor 3p6

Up to dateness: revised 11/6/74 3p7

Fate: Development will maintain document. 3p8

Output Processor Users' Guide 3q

description of how the output processor directives operate from the viewpoint of the user and a list of directives and explanations 3q1

Online and offline status 3q2

offline title: Output Processor Users' Guide 3q2a

online location: The online version consists of the following files: 3q2b

<USERGUIDES,OP-COM,> 3q2b1

<USERGUIDES,OP-DIREXP,> 3q2b2

<USERGUIDES,OP-DIRLIST,> 3q2b3

<USERGUIDES,OP-INDEX,> 3q2b4

<USERGUIDES,OP-INTRO,> 3q2b5

<USERGUIDES,OP-PARAMS,> 3q2b6

Maintenance of NLS-8 Documentation

<USERGUIDES,OP-VALOP,> 3q2b7

Relevant system: NLS-8 3q3

Size in pages 3q4

offline: 100 3q4a

online: 132 3q4b

Intended audience: NLS users interested in formatting documents for printing 3q5

Caretaker: Dean 3q6

Up to dateness: presently being revised 3q7

Fate: We will hand over updating and maintenance of offline copies and online files to Applications. Dirk may continue to be involved while wearing his DCPS hat. 3q8

Userguide RSEXEC 3r

NLS version of a sequential help file from BBN 3r1

Online and offline status 3r2

offline location: file drawer in room j2082 (pooh's desk) 3r2a

online location: archived in directory Weinberg. 3r2b

Relevant system TENEX 3r3

Size in pages 3r4

offline: 12 3r4a

online: 12 3r4b

Intended audience: any network user 3r5

Caretaker: ? 3r6

Up to dateness: updated information was obtained from BBN on 1/1/75, and this needs to be edited to make it a pretty NLS file. 3r7

Fate: We will hand over updating and maintenance of offline copies and online files to Applications. 3r8

TENEX Userguide 3s

to introduce the novice user to the more commonly used features of ARC's version of the TENEX EXECUTIVE language 3s1

Online and offline status 3s2

offline title: SRI-ARC / Office-1 TENEX USERS' GUIDE 3s2a

online location: <USERGUIDES, TENEX,> 3s2b

Relevant system: TENEX 3s3

Size in pages 3s4

offline: 40 3s4a

online: 33 3s4b

Intended audience: NLS users interested in TENEX 3s5

Caretaker: Dean 3s6

Up to dateness: revised 6/26/74 3s7

Fate: We will hand over updating and maintenance of offline copies and online files to Applications. 3s8

New to Old Equivalents List 3t

alphabetical list of NLS-7 commands with the equivalent NLS-8 commands 3t1

Online and offline status 3t2

offline title: NLS-8 Equivalents of NLS-7 Commands 3t2a

Relevant system: NLS-8 3t3

Size in pages 3t4

offline: 13 3t4a

Intended audience: NLS-7 users who want to learn NLS-8 3t5

Caretaker: ? 3t6

Up to dateness: revised 10/6/74 3t7

Fate: We will hand over updating and maintenance of offline copies and online files to Applications. 3t8

Updated CML Documentation 3u

A brief description. 3u1

Online and offline status 3u2

offline title: Updated CML Documentation 3u2a

online location: <Journal,20438,> 3u2b

Relevant system:NLS-8 3u3

Size in pages 3u4

offline: 20 3u4a

online: 20 3u4b

Intended audience: people interested in NLS programming 3u5

Caretaker: Ken Victor 3u6

Up to dateness: revised 11/21/73 3u7

Fate: Development will maintain this document. 3u8

NLS File Structure 3v

a discussion of the structure of NLS Files 3v1

Online and offline status 3v2

offline title: NLS File Structure* Documentation 3v2a

online location: <Journal,17069,> 3v2b

Relevant system: NLS-7 and NLS-8 3v3

Size in pages 3v4

offline: 16 3v4a

online: 16 3v4b

Intended audience: NLS system programmers 3v5

Caretaker: ? 3v6

Up to dateness: 6/6/73 3v7

Fate: We will hand over updating and maintenance of offline copies and online files to Applications. 3v8

Links in xnls 3w

description of some of the new precedures available in xnls for dealing with links and address expressions, and a formal description of the syntax for a link and an address expression 3w1

Online and offline status 3w2

offline title: links in xnls 3w2a

online location: <Journal,22174,> 3w2b

Relevant system: 3w3

Size in pages 3w4

offline: 5 3w4a

online: 5 3w4b

Intended audience: people writing code that deals with links 3w5

Caretaker: ? 3w6

Up to dateness: revised 2/28/74 3w7

Fate: We will hand over updating and maintenance of offline copies and online files to Applications. 3w8

Proposed NLS Code Format and Documentation Standards 3x

suggested formats for various L10 language constructs and standardsuggested standards for srror handling and source code file arrangement 3x1

Online and offline status 3x2

offline title: Proposed NLS Code Format and Documentation Standards 3x2a

online location: <Journal,15934,> 3x2b

Maintenance of NLS-8 Documentation

Relevant system: NLS-7 and NLS-8 3x3

Size in pages 3x4

offline: 21 3x4a

online: 21 3x4b

Intended audience: 3x5

Caretaker: Ken Victor 3x6

Up to dateness: 4/13/73 3x7

Fate: Development will maintain this document. 3x8

L10 Documentation 3y

description of a programming language, L10, for use on the PDP10, and examples to illustrate the L10 language. 3y1

Online and offline status 3y2

offline title: L10 Documentation 3y2a

online location: <Journal,7052,> 3y2b

Relevant system: NLS-7 and NLS-8 3y3

Size in pages 3y4

offline: 56 3y4a

online: 56 3y4b

Intended audience: L10 programmers 3y5

Caretaker: Charles Irby 3y6

Up to dateness: revised 5/29/71 3y7

Fate: Development will maintain this document. 3y8

Userguides Identification 3z

instructions on using the new ident subsystem. 3z1

Online and offline status 3z2

offline title: none 3z2a

online location: <userguides,Identification,> 3z2b

Relevant system: NLS-8 3z3

Size in pages 3z4

online: 6 3z4a

Intended audience: 3z5

Caretaker: ? 3z6

Up to dateness: revised 12/14/74 3z7

Fate: We will hand over updating and maintenance of offline copies and online files to Applications. 3z8

Workstation Equipment Reference Manual 3a@

defines the three workstations called Lineprocessor, Teleprinter, and DEX, the specifications for the equipment involved, compatible equipment, and reference documents 3a@1

Online and offline status 3a@2

offline title: Workstation Equipment Reference Manual 3a@2a

Relevant system: NLS-8 3a@3

Size in pages 3a@4

offline: 53 3a@4a

Intended audience: hardware people supporting the lineprocessor 3a@5

Caretaker: Martin Hardy 3a@6

Up to dateness: revised 12/10/74 3a@7

Fate: We will hand over updating and maintenance of offline copies and online files to Applications. 3a@8

Lineprocessor Users' Guide 3aa

instructions for startup and use of the line processor and trouble shooting 3aa1

Online and offline status 3aa2

offline title: Line Processor Users' Guide 3aa2a

online location: <USERGUIDES, LINEPROCESSOR,> 3aa2b

Relevant system: Lineprocessor 3aa3

Size in pages 3aa4

offline: 10 3aa4a

online: 12 3aa4b

Intended audience: people using the lineprocessor through a TIP. 3aa5

Caretaker: ? 3aa6

Up to dateness: revised 12/31/74 3aa7

Fate: We will hand over updating and maintenance of offline copies and online files to Applications. 3aa8

Master list of AVailable Documentation 3ab

master list with descriptions of available documentation 3ab1

Online and offline status 3ab2

online location: <weinberg,docufate,> 3ab2a

Relevant system: all 3ab3

Size in pages 3ab4

online: 7 3ab4a

Intended audience: everyone 3ab5

Caretaker:? 3ab6

Up to dateness: revised 1/16/75 3ab7

Fate: We will hand over updating and maintenance of offline copies and online files to Applications. 3ab8

DVN RWW POOH 21-JAN-75 12:45 25158

Maintenance of NLS-8 Documentation

(J25158) 21-JAN-75 12:45;;;; Title: Author(s): Dirk H. Van Nouhuys, Richard W. Watson, Ann Weinberg/DVN RWW POOH; Distribution: /JOAN([ACTION] dirt notebook please) UD([ACTION]) SRI-ARC([INFO-ONLY]) DIRT([INFO-ONLY]) ; Sub-Collections: SRI-ARC DIRT UD; Clerk: DVN; Origin: < HAMILTON, DVNCOLDTURKEY.NLS;4, >, 21-JAN-75 12:40 DVN
;;;;####;

RLL 21-JAN-75 12:46 25159

bug: misspelled name in show record comman

the word 'beginning' is misspelled ('begining') when show record command (in sendmail) is request for name beginning with given letters (e.g. show record(for ident) .xxx...). 1

RLL 21-JAN-75 12:46 25159

bug: misspelled name in show record comman

(J25159) 21-JAN-75 12:46;;;; Title: Author(s): Robert N. Lieberman/RLL; Distribution: /FEED([ACTION]) KIRK([INFO-ONLY]) ; Sub-Collections: SRI-ARC; Clerk: RLL;

POOH 21-JAN-75 14:43 25160

test mail

This is a sample statement to find out

POOH 21-JAN-75 14:43 25160

test mail

TITLE:test mail
COMMENT: This is a sample statement to find out
if you can inser
carriage
retur
ns
or whe
it will
be rejected.
AUTHOR(S):POOH
NUMBER:25160

DISTRIBUTE FOR ACTION TO:pooh
DISTRIBUTE FOR INFO-ONLY TO:pooh
SUBCOLLECTION(S):
KEYWORD(S):
HANDLING INSTRUCTION:
RECORDING INSTRUCTION:
OFFLINE ITEM -- LOCATED AT:
RFC NUMBER:
OBSOLETES ITEM NUMBER(S):
ACCESS STATUS:
UPDATE TO ITEM NUMBER(S):
INSERT LINK TO FOLLOW:
FORWARD ITEM NUMBER:
MESSAGE:will this work
BRANCH AT:
PLEX AT:
GROUP AT:
FILE:
SEND THE MAIL. 1

> The " ( ) 2

LINKS 3

(GJOURNAL, 25034,) business card file 3a

<documentation, xhelp,> 3b

< WEINBERG, PAGR,NLS;, > 3c

(Journal) Journal documents (most recent first) 4

FEED 21-JAN-75 09:50 31678
Status of HELP Suggestions - Criticisms
Message: Your two messages concerning OFFICE-1 He6p have been
given to POOH. I'll let you know what the outcome is.

POOH 21-JAN-75 14:43 25160

test mail

*****Note: [ACTION] *****

4a

JBP 20-JAN-75 12:29 25153
Batch Job Model in NSW
Location: (HJOURNAL, 25153, 1:w)
*****Note: [INFO-ONLY] *****

4b

JHB 20-JAN-75 10:16 25151
USER DEVELOPMENT REPORT: TRIP TO THE EAST COAST, DEC, 74
Location: (HJOURNAL, 25151, 1:w)
*****Note: [INFO-ONLY] *****

4c

Comments: The following trip report lists users contacted and trained at respective user sites plus brief comments on the NLS course material covered, assistance given, issues raised, client applications, and workshop development.

4c1

NDM 18-JAN-75 16:18 31665
DDSI Run (including final version DCA Internet Study)
Location: (HJOURNAL, 31665, 1:w)
*****Note: [INFO-ONLY] *****

4d

JHB 17-JAN-75 18:39 25145
Responsibility for Transfering Online Documentation to Office-1
Location: (HJOURNAL, 25145, 1:w)
*****Note: [ACTION] *****

4e

KIRK 17-JAN-75 16:39 25142
Response to HGL's 25132
Message: When the back arrow command does what it does if you type one too many "no"s in the Help command as I described in 25131, sloppy is the best adjective I can think of to describe it's implementation. It is especially cruel in TNLS. Since Harvey brought up the point of consistency, I would like to point out that the "back" function is confusing as it does not work like a straight stack forgetting places you back out of and it is also inconsistant with the NLS return ring which remembers where you last backed out of. The problem I addressed in 25131 is not mentioned in HGL's 25132. What I described occurs when the program thinks the user is trying to back off the end of the return stack. It is a very bad bug and should be fixed asap.
*****Note: [INFO-ONLY] *****

4f

POOH 21-JAN-75 14:43 25160

test mail

JEW 17-JAN-75 10:23 25133
Proposed List Data Type for L10
Location: (HJOURNAL, 25133, 1:w)
*****Note: [ACTION] *****

4g

HGL 17-JAN-75 09:48 25132
Back Arrow in Help -- re Kirk's recent message
Message: Granted the sloppiness of the current Help implementation, I don't think the adjective should apply to the instance Kirk describes. A user has the option of accepting the place presented to him in the feedback window after a backarrow has been hit by typing a yes or a no (any character other than a y or CA). This is exactly the same converntion used in Jump to Return and Jump to File Return. In the name of holy consistency, I feel the criticized aspect of the current help back implementation does not warrant modification unless and until the design for those other commands ha been changed. (There are aspects of the backk command which should be changed first, if anyone is interested: e.g., the fact that in multi-displays of long menus you don't go back into inter-menu frames. The mods are somewhat expensive inthese resource inflationary times.)
*****Note: [INFO-ONLY] *****

4h

Comments: cf (25131,)

4h1

KIRK 17-JAN-75 01:59 25131
utter ridiculousness
Message: Why is it that when you type the back arrow key in the Help command to go back to the beginning, it says "no others have been shownY/N OK:" and then re-prints the entire view of your current location? This is utter ridiculousness. So what if the user typed one too many backarrows? I swear, the help accessing system which should handle the user with kid gloves has the sloppiest implementation in all of NLS.
*****Note: [INFO-ONLY] *****

4i

CHI 16-JAN-75 10:03 25125
Lastname querry in SENDMAIL distribution lists
Message: Kirk's 25114 documents a misunderstanding about how .lastname querries work in SENDMAIL and IDENTIFICATION subsystems. When you type.LASTNAME you must terminate it with ,, or CA or , (comma). SPACE CANNOT be used to terminate a lastname since there are obviously lastnames that include the character SPACE (van de Riet). Thus, I suggest that as a methodology that you always use comma to separate idents and querries in identlists. -- Charles.

POOH 21-JAN-75 14:43 25160

test mail

*****Note: [INFO-ONLY] *****

4j

DVN 16-JAN-75 10:00 25124
Please make some Ident file changes.
Message: Please add to the ident HELP Pooh, DVN and &dirt. Please add to dpcs r11, please remove NDM from DIRT. Thanks. D.
*****Note: [INFO-ONLY] *****

4k

HELP 14-JAN-75 19:27 25112
Development-Applications interface at the Help-Feedback level
Location: (HJOURNAL, 25112, 1:w)
*****Note: [INFO-ONLY] *****

41

Comments: In response to FEED's 31636

411

KIRK 15-JAN-75 00:15 25113
How about the OFFICE-1 operator?
Message: If for various reasons (mostly having to do with the hassle that's currently necessary) no one in applications or development wishes to be responsible for moving userguide and documentation files to office-1, and since our own operators will be moving on to other things at the end of February, perhaps the appropriate person for "retrieving" files for OFFICE-1 from SRI-ARC is the same person responsible for "retrieving" files for OFFICE-1 from the archive: the OFFICE-1 operator. Anyone should be capable of requesting a file to be at OFFICE-1 by sending a message to the operator there. The tymshare operators certainly should know how to tranfer files.
*****Note: [INFO-ONLY] *****

4m

UD 13-JAN-75 08:40 31622
The Network Login Procedure
Location: (GJOURNAL, 31622, 1:w)
*****Note: [INFO-ONLY] *****

4n

Comments: This is a complete description of the recently introduced Network login procedure. Architects may wish to give their users a simplified tutorial. Since we do not control the Network, and it is experimental, this is subject to change....

4n1

DVN PWO 12-JAN-75 21:01 25065
Commerce Business Daily Request for Qualifications to Supply Computer Publication System to Bonneville Power Authority and

POOH 21-JAN-75 14:43 25160

test mail

Response,
Location: (GJOURNAL, 25065, 1:w)
*****Note: [INFO-ONLY] *****

4o

Comments: Mailed Friday the 10th

4o1

JEW 10-JAN-75 17:28 25009
NSWSTRUC 2 / NSW Process Structure
Location: (GJOURNAL, 25009, 1:w)
*****Note: [INFO-ONLY] *****

4p

JEW 10-JAN-75 17:25 25008
NTP 2 / The NSW Tool Package
Location: (GJOURNAL, 25008, 1:w)
*****Note: [INFO-ONLY] *****

4q

JEW 10-JAN-75 17:30 25062
PCPV2CHANGES / PCP Inter-Version (2-3) Documentation
Location: (GJOURNAL, 25062, 1:w)
*****Note: [INFO-ONLY] *****

4r

DVN 9-JAN-75 21:22 25042
A Simple Commands Branch to Reject Journal Mail on Specific Subjects
Location: (GJOURNAL, 25042, 1:w)
*****Note: [INFO-ONLY] *****

4s

Comments: This journal item is sent with tongue in cheek, but only partly.

4s1

FEED 9-JAN-75 15:24 31612
New Status of Viewspec Q
Message: The last and final word on Viewspec Q is that a close approximation is the combination of viewspecs B and g. Viewspec was intended to be implemented but didn't make it over the line so I'm now moving (25015,) to be in the design bugs branch of the feedback file. A note to documentors - the viewspec card does not indicate that viewspec Q is not implemented. Susan/FEED
*****Note: [INFO-ONLY] *****

4t

RWW 8-JAN-75 15:58 25028
NOTES ON NLS BACKEND ON OTHER MACHINES
Location: (GJOURNAL, 25028, 1:w)

POOH 21-JAN-75 14:43 25160

test mail

*****Note: [INFO-ONLY] *****

4u

Comments: Notes compiled at request of Bill Carlson, 4u1

RWW 2-JAN-75 17:09 24986
Quick Response to Crocker NSW Plan Input Request
Location: (GJOURNAL, 24986, 1:w)
*****Note: [INFO-ONLY] *****

4v

KIRK 30-DEC-74 15:35 24857
Response to > ROUGH DRAFT NSW Documentation Work Breakdown and
Time Allocation <24848,>
Location: (GJOURNAL, 24857, 1:w)
*****Note: [INFO-ONLY] *****

4w

DVN 28-DEC-74 17:34 24848
ROUGH DRAFT NSW Documentation Work Breakdown and Time Allocation
Location: (GJOURNAL, 24848, 1:w)
*****Note: [INFO-ONLY] *****

4x

DCE 26-DEC-74 18:36 24837
SRI-ARC CY74 Accomplishment List
Location: (GJOURNAL, 24837, 1:w)
*****Note: [INFO-ONLY] *****

4y

Comments: A list requested by Licklider of all PI's --
contributed to by Watson, Norton and Feinler, and
assembled/edited by Engelbart. The original was Output to
Sequential File with VSPECS 'wynhA', and the resulting
sequential file SNDMSG'd to Licklider at ISI, with alternate
copies at OFFICE-1 and BBN-TENEX. Considerable trouble getting
through to ISI, don't know if succeeded. 4y1

KIRK 24-DEC-74 01:17 24806
A Multi-Level Integrated User Help System
Location: (GJOURNAL, 24806, 1:w)
*****Note: [INFO-ONLY] *****

4z

JBP 19-DEC-74 14:25 31524 [NWG/RFC# 678]
Standard File Formats
Location: (GJOURNAL,31524,1:w)
*****Note: [INFO-ONLY]

POOH 21-JAN-75 14:43 25160

test mail

(Secondary Distribution Copy from JBP)*****

4a@

10-JAN-75 0838-PST WEINBERG: Tenex Doc.
Distribution: KIDSTON AT BBN, weinberg
Received at: 10-JAN-75 08:38:15

5

Carol: Thank for your quick response. The P>O> number is A96874. We ordered 7 Tenex Exec. Manuals, 8 Tenex User's Guides, 5 Tenex Memos (1970), 5 Tenex Monitor Manuals, and 4 Tenex Tecos. The date on the purchase order from the institute is 12/02/74. If you need any more information, just let me know. ann

5a

10-JAN-75 1013-EST KIDSTON at BBN-TENEXA: documentation
Distribution: WEINBERG AT SRI-ARC
Received at: 10-JAN-75 07:15:23

6

Ann,
could you let me know the manuals you ordered and the P.O. number if you have one and I will look into it.
Carol

6a

24-DEC-74 1844-PST KELLEY: identification userguide
Distribution: WEINBERG
Received at: 24-DEC-74 18:44:40

7

do you have it listed in doculist? it's a file journalized by KJM and modified by me. It is in <userguides,identifcations,>

7a

author Journal documents authored

8

DVN RWW POOH 21-JAN-75 12:45 25158
Maintenance of NLS-8 Documentation
Location: (HJOURNAL, 25158, 1:w)
*****Note: Author Copy*****

8a

DVN POOH KIRK 17-JAN-75 17:21 25144
Informal Documentation Report for Week Ending Jan 17
Location: (HJOURNAL, 25144, 1:w)
*****Note: Author Copy*****

8b

POOH 15-JAN-75 12:19 25115
Set Tenex (protection for file named) CONTENT CONTROLS OK:
Message: This command does not work at all. you receive the error message "Illegal Protection Specified" When it is fixed, I suggest that a Reset Tenex command also be put in. This way it

POOH 21-JAN-75 14:43 25160

test mail

will be like all the other Set commands that have Reset commands also.
*****Note: Author Copy*****

8c

POOH 23-DEC-74 11:01 24799
Business Cards
Message: We are going to print new business cards very soon. I have samples of ten different formats. Anyone who wishes to order a card should come see me to choose a style and content. Orders will be taken through the first week in January.
*****Note: Author Copy*****

8d

POOH 8-NOV-74 10:08 24444
Tenex Documentation
Message: The following documentation can be ordered from BBN. Please let me know if you would like any of these manuals. I plan to place an order in the next week.
TENEX EXECUTIVE Manual:Belleville,Postel, Jake, Beck
TENEX JSYS Manual,Postel
TENEX USER7s Guide:Belleville,Postel, Jake, Beck
TENEX memos (1970):Belleville,Postel, Jake
TENEX Monitor Manual:Belleville,PostelJake
PAL11X,Postel, Jake
BCPL
TENEX TECO, Jake, Beck
*****Note: Author Copy*****

8e

Action 9

Info 10

READMAIL 11

EKM 2-DEC-74 16:29 24645
Network Journal Delivery
Location: (GJOURNAL, 24645, 1:w)
*****Note: [ACTION] *****

11a

JHB 30-NOV-74 22:50 24636
Conolidation of the Feedback Mechanisms for NLS
Location: (GJOURNAL, 24636, 1:w)
*****Note: [INFO-ONLY] *****

11b

EKM 21-NOV-74 15:51 24570

POOH 21-JAN-75 14:43 25160

test mail

NSW / NLS Plans
Location: (GJOURNAL, 24570, 1:w)
*****Note: [ACTION] *****

11c

MESSAGES (:Xb)

12

13

ME EXE PRO LOA PRO MESSAGE

14

EXE MES MOV MES JOURNALD

14a

EXE PRO DEL LAS

14b

< USERGUIDES, ARCLOCATOR,NLS;48, >

14c

Last Writer: KELLEY
Last Written: 7-AUG-74 15:50:08

14c1

< USERGUIDES, COMMANDS,NLS;144, >

14d

Last Writer: BECK
Last Written: 2-JAN-75 18:35:16

14d1

< USERGUIDES, COMMANDS-COM,NLS;2, >

14e

Last Writer: BECK
Last Written: 2-JAN-75 18:09:50

14e1

< USERGUIDES, DEX-PRIMER,NLS;11, >

14f

Last Writer: SYSTEM
Last Written: 8-JAN-74 15:50:51

14f1

< USERGUIDES, DEX-TRANSCRIPTION,NLS;3, >

14g

Last Writer: BECK
Last Written: 6-AUG-74 21:38:44

14g1

< USERGUIDES, DEX-TWO,NLS;1, >

14h

Last Writer: SYSTEM

POOH 21-JAN-75 14:43 25160

test mail

Last Written: 21-JUN-73 10:16:00 14h1

< USERGUIDES, GLOSSARY.NLS;35, > 14i

Last Writer: KELLEY
Last Written: 9-OCT-74 20:32:31 14i1

< USERGUIDES, GLOSSARY.PRINT;3, > 14j

Last Writer: KELLEY
Last Written: 2-OCT-74 17:11:23 14j1

< USERGUIDES, HELP.NLS;2, > 14k

Last Writer: WEINBERG
Last Written: 13-NOV-74 15:21:45 14k1

< USERGUIDES, JOURNAL-NETSUB.NLS;15, > 14l

Last Writer: KELLEY
Last Written: 17-APR-74 12:15:03 14l1

< USERGUIDES, JOURNAL-QUERY.NLS;61, > 14m

Last Writer: KELLEY
Last Written: 10-DEC-74 23:32:50 14m1

< USERGUIDES, L10-GUIDE.NLS;323, > 14n

Last Writer: WEINBERG
Last Written: 14-NOV-74 09:14:32 14n1

< USERGUIDES, L10-GUIDE.PRINT;7, > 14o

Last Writer: WEINBERG
Last Written: 8-NOV-74 09:24:27 14o1

< USERGUIDES, L10USERGUIDES.PRINT;1, > 14p

Last Writer: SYSTEM

POOH 21-JAN-75 14:43 25160

test mail

Last Written: 30-OCT-74 06:56:24

14p1

< USERGUIDES, LINEPROCESSOR.NLS;1, > 14q

Last Writer: VANNOUHUYS
Last Written: 7-JAN-75 19:24:03

14q1

< USERGUIDES, LINEPROCESSOR.PRINT;1, > 14r

Last Writer: SYSTEM
Last Written: 7-JAN-75 19:24:30

14r1

< USERGUIDES, LOCATOR.NLS;14, > 14s

Last Writer: PETERS
Last Written: 17-JAN-75 09:30:45

14s1

< USERGUIDES, MANUAL.NLS;29, > 14t

Last Writer: BECK
Last Written: 17-NOV-74 15:14:20

14t1

< USERGUIDES, NDDT-SYMBOLIC-DEBUGGER.NLS;4, > 14u

Last Writer: WEINBERG
Last Written: 6-NOV-74 16:24:55

14u1

< USERGUIDES, NDDT-SYMBOLIC-DEBUGGER.PRINT;4, > 14v

Last Writer: WEINBERG
Last Written: 6-NOV-74 16:25:45

14v1

< USERGUIDES, OP-GUIDE.NLS;52, > 14w

Last Writer: MEYER
Last Written: 9-JAN-75 10:03:07

14w1

< USERGUIDES, OP-INTRO.NLS;11, > 14x

Last Writer: MEYER

POOH 21-JAN-75 14:43 25160

test mail

Last Written: 23-AUG-73 11:49:31

14x1

< USERGUIDES, PRIMER.NLS;19, > 14y

Last Writer: WEINBERG
Last Written: 17-JAN-75 15:51:16

14y1

< USERGUIDES, RSEXEC.NLS;1, > 14z

Last Writer: MEYER
Last Written: 12-JAN-74 11:51:58

14z1

< USERGUIDES, TENEX.NLS;115, > 14a@

Last Writer: MEYER
Last Written: 26-JUL-74 16:27:42

14a@1

< USERGUIDES, TUG.NLS;1, > 14aa

Last Writer: SYSTEM
Last Written: 24-MAY-73 21:27:55

14aa1

< USERGUIDES, VIEWSPECCARD.NLS;10, > 14ab

Last Writer: BECK
Last Written: 2-NOV-74 10:20:50

14ab1

< USERGUIDES, [ARCHIVE-DIRECTORY].;1, >
[<BECK>(USERGUIDES)[ARCHIVE-DIRECTORY].PC;1: File Not On-line;
If Archived, Use EXEC's INTERROGATE] 14ac

Last Writer: BEACH
Last Written: 10-JAN-75 03:57:21

14ac1

POOH 21-JAN-75 14:43 25160

test mail

(J25160
) 21-JAN-75 14:43;;;; Title: Author(s): Ann Weinberg/POOH;
Distribution: /POOH([ACTION]) POOH([INFO-ONLY]) ;
Sub-Collections: SRI-ARC; Clerk: POOH;

DCE 21-JAN-75 15:04 25161

Phone Log, 21 Jan 75: A. E. Tyler of NSWSES

An explicit DPCS interest; asking Dirk to follow up, keep me and RLL tied in

DCE 21-JAN-75 15:04 25161

Phone Log, 21 Jan 75: A. E. Tyler of NSWSES

I received a call today from: 1

Mail Address: 1a

A. E. Tyler, Code 5603C
NSWSES
Port Hueneme, Calif 93043 1a1

(805) 982-5812 1b

NSWSES is for "Navy Ship Weapons System Engineering Station" 1c

Tyler said that a person named Berkowitz referred us to him; RLL and I assume that this is Dr. Sid Berkowitz, of NSRDC (see RLL's contact report dated 14-JAN-75 from a meeting on 13 Dec 74 -- 25068, 8j); Sid is involved with the Navy Technical Document project, and could have been communicting with Tyler in this context. 2

Tyler is in the publications business, concerned with producing Navy instruction manuals. I don't know how extensive is his particular application domain, or his range of influence, 3

I told him briefly that we had an applications group capable of providing exploratory-application support, and that Dirk VanNouhuys was our man specially involved with encouraging DPCS activity. I promised that we'd send him descriptive literature, and follow that at an appropriate later time with a phone call. I'm asking Dirk to see that this is followed up (literature mailed, followup phone contact, etc.). 4

DCE 21-JAN-75 15:04 25161

Phone Log, 21 Jan 75: A. E. Tyler of NSWSES

(J25161) 21-JAN-75 15:04;;;; Title: Author(s): Douglas C. Engelbart/DCE; Distribution: /DVN([ACTION]) JCN([INFO-ONLY]) RLL([INFO-ONLY]) JML([INFO-ONLY]) ; Sub-Collections: SRI-ARC; Clerk: DCE;

POOH 21-JAN-75 15:10 25162

test

> The " ( ) 1

LINKS 2

(GJOURNAL, 25034,) business card file 2a

<documentation, xhelp,> 2b

< WEINBERG, PAGR,NLS;, > 2c

(Journal) Journal documents (most recent first) 3

FEED 21-JAN-75 09:50 31678
Status of HELP Suggestions - Criticisms
Message: Your two messages concerning OFFICE-1 He6p have been given to POOH. I'll let you know what the outcome is.
*****Note: [ACTION] *****
3a

JBP 20-JAN-75 12:29 25153
Batch Job Model in NSW
Location: (HJOURNAL, 25153, 1:w)
*****Note: [INFO-ONLY] *****
3b

JHB 20-JAN-75 10:16 25151
USER DEVELOPMENT REPORT: TRIP TO THE EAST COAST, DEC, 74
Location: (HJOURNAL, 25151, 1:w)
*****Note: [INFO-ONLY] *****
3c

Comments: The following trip report lists users contacted and trained at respective user sites plus brief comments on the NLS course material covered, assistance given, issues raised, client applications, and workshop development. 3c1

NDM 18-JAN-75 16:18 31665
DDSI Run (including final version DCA Internet Study)
Location: (HJOURNAL, 31665, 1:w)
*****Note: [INFO-ONLY] *****
3d

JHB 17-JAN-75 18:39 25145
Responsibility for Transfering Online Documentation to Office-1
Location: (HJOURNAL, 25145, 1:w)
*****Note: [ACTION] *****
3e

KIRK 17-JAN-75 16:39 25142

POOH 21-JAN-75 15:10 25162

test

Response to HGL's 25132
Message: When the back arrow command does what it does if you type one too many "no"s in the Help command as I described in 25131, sloppy is the best adjective I can think of to describe it's implementation. It is especially cruel in TNLS. Since Harvey brought up the point of consistency, I would like to point out that the "back" function is confusing as it does not work like a straight stack forgetting places you back out of and it is also inconsistant with the NLS return ring which remembers where you last backed out of. The problem I addressed in 25131 is not mentioned in HGL's 25132. What I described occurs when the program thinks the user is trying to back off the end of the return stack. It is a very bad bug and should be fixed asap.
*****Note: [INFO-ONLY] *****

3f

JEW 17-JAN-75 10:23 25133
Proposed List Data Type for L10
Location: (HJOURNAL, 25133, 1:w)
*****Note: [ACTION] *****

3g

HGL 17-JAN-75 09:48 25132
Back Arrow in Help -- re Kirk's recent message
Message: Granted the sloppiness of the current Help implementation, I don't think the adjective should apply to the instance Kirk describes. A user has the option of accepting the place presented to him in the feedback window after a backarrow has been hit by typing a yes or a no (any character other than a y or CA). This is exactly the same converntion used in Jump to Return and Jump to File Return. In the name of holy consistency, I feel the criticized aspect of the current help back implementation does not warrant modification unless and until the design for those other commands ha been changed. (There are aspects of the backk command which should be changed first, if anyone is interested: e.g., the fact that in multi-displays of long menus you don't go back into inter-menu frames. The mods are somewhat expensive inthese resource inflationary times.)
*****Note: [INFO-ONLY] *****

3h

Comments: cf (25131,)

3h1

KIRK 17-JAN-75 01:59 25131
utter ridiculousness
Message: Why is it that when you type the back arrow key in the Help command to go back to the beginning, it says "no others have been shownY/N OK:" and then re-prints the entire view of your current location? This is utter ridiculousness. So what if the

POOH 21-JAN-75 15:10 25162

test

user typed one too many backarrows? I swear, the help accessing system which should handle the user with kid gloves has the sloppiest implementation in all of NLS.
*****Note: [INFO-ONLY] *****

3i

CHI 16-JAN-75 10:03 25125
Lastname querry in SENDMAIL distribution lists
Message: Kirk's 25114 documents a misunderstanding about how ,lastname querries work in SENDMAIL and IDENTIFICATION subsystems. When you type,LASTNAME you must terminate it with ,,, or CA or , (comma). SPACE CANNOT be used to terminate a lastname since there are obviously lastnames that include the character SPACE (van de Riet). Thus, I suggest that as a methodology that you always use comma to separate idents and querries in identlists. -- Charles.
*****Note: [INFO-ONLY] *****

3j

DVN 16-JAN-75 10:00 25124
Please make some Ident file changes.
Message: Please add to the ident HELP Pooh, DVN and &dirt. Please add to dpcs rll, please remove NDM from DIRT. Thanks. D.
*****Note: [INFO-ONLY] *****

3k

HELP 14-JAN-75 19:27 25112
Development-Applications interface at the Help-Feedback level
Location: (HJOURNAL, 25112, 1:w)
*****Note: [INFO-ONLY] *****

31

Comments: In response to FEED's 31636

311

KIRK 15-JAN-75 00:15 25113
How about the OFFICE-1 operator?
Message: If for various reasons (mostly having to do with the hassle that's currently necessary) no one in applications or development wishes to be responsible for moving userguide and documentation files to office-1, and since our own operators will be moving on to other things at the end of February, perhaps the appropriate person for "retrieving" files for OFFICE-1 from SRI-ARC is the same person responsible for "retrieving" files for OFFICE-1 from the archive: the OFFICE-1 operator. Anyone should be capable of requesting a file to be at OFFICE-1 by sending a message to the operator there. The tymshare operators certainly should know how to tranfer files.
*****Note: [INFO-ONLY] *****

3m

test

UD 13-JAN-75 08:40 31622
The Network Login Procedure
Location: (GJOURNAL, 31622, 1:w)
*****Note: [INFO-ONLY] *****

3n

Comments: This is a complete description of the recently introduced Network login procedure. Architects may wish to give their users a simplified tutorial. Since we do not control the Network, and it is experimental, this is subject to change....

3n1

DVN PWO 12-JAN-75 21:01 25065
Commerce Business Daily Request for Qualifications to Supply Computer Publication System to Bonneville Power Authority and Response,
Location: (GJOURNAL, 25065, 1:w)
*****Note: [INFO-ONLY] *****

3o

Comments: Mailed Friday the 10th

3o1

JEW 10-JAN-75 17:28 25009
NSWSTRUC 2 / NSW Process Structure
Location: (GJOURNAL, 25009, 1:w)
*****Note: [INFO-ONLY] *****

3p

JEW 10-JAN-75 17:25 25008
NTP 2 / The NSW Tool Package
Location: (GJOURNAL, 25008, 1:w)
*****Note: [INFO-ONLY] *****

3q

JEW 10-JAN-75 17:30 25062
PCPV2CHANGES / PCP Inter-Version (2-3) Documentation
Location: (GJOURNAL, 25062, 1:w)
*****Note: [INFO-ONLY] *****

3r

DVN 9-JAN-75 21:22 25042
A Simple Commands Branch to Reject Journal Mail on Specific Subjects
Location: (GJOURNAL, 25042, 1:w)
*****Note: [INFO-ONLY] *****

3s

Comments: This journal item is sent with tongue in cheek, but only partly.

3s1

test

FEED 9-JAN-75 15:24 31612
New Status of Viewspec Q
Message: The last and final word on Viewspec Q is that a close approximation is the combination of viewspecs B and g. Viewspec was intended to be implemented but didn't make it over the line so I'm now moving (25015,) to be in the design bugs branch of the feedback file. A note to documentors - the viewspec card does not indicate that viewspec Q is not implemented. Susan/FEED
*****Note: [INFO-ONLY] *****

3t

RWW 8-JAN-75 15:58 25028
NOTES ON NLS BACKEND ON OTHER MACHINES
Location: (GJOURNAL, 25028, 1:w)
*****Note: [INFO-ONLY] *****

3u

Comments: Notes compiled at request of Bill Carlson. 3u1

RWW 2-JAN-75 17:09 24986
Quick Response to Crocker NSW Plan Input Request
Location: (GJOURNAL, 24986, 1:w)
*****Note: [INFO-ONLY] *****

3v

KIRK 30-DEC-74 15:35 24857
Response to > ROUGH DRAFT NSW Documentation Work Breakdown and Time Allocation <24848,>
Location: (GJOURNAL, 24857, 1:w)
*****Note: [INFO-ONLY] *****

3w

DVN 28-DEC-74 17:34 24848
ROUGH DRAFT NSW Documentation Work Breakdown and Time Allocation
Location: (GJOURNAL, 24848, 1:w)
*****Note: [INFO-ONLY] *****

3x

DCE 26-DEC-74 18:36 24837
SRI-ARC CY74 Accomplishment List
Location: (GJOURNAL, 24837, 1:w)
*****Note: [INFO-ONLY] *****

3y

Comments: A list requested by Licklider of all PI's -- contributed to by Watson, Norton and Feinler, and assembled/edited by Engelbart. The original was Output to Sequential File with VSPECS 'wynhA', and the resulting sequential file SNDMSG'd to Licklider at ISI, with alternate

test

copies at OFFICE-1 and BBN-TENEX. Considerable trouble getting through to ISI, don't know if succeeded. 3y1

KIRK 24-DEC-74 01:17 24806
A Multi-Level Integrated User Help System
Location: (GJOURNAL, 24806, 1:w)
*****Note: [INFO-ONLY] *****
3z

JBP 19-DEC-74 14:25 31524 [NWG/RFC# 678]
Standard File Formats
Location: (GJOURNAL,31524,1:w)
*****Note: [INFO-ONLY]
(Secondary Distribution Copy from JBP)*****
3a@

10-JAN-75 0838-PST WEINBERG: Tenex Doc.
Distribution: KIDSTON AT BBN, weinberg
Received at: 10-JAN-75 08:38:15 4

Carol: Thank for your quick response. The P>O> number is A96874. We ordered 7 Tenex Exec. Manuals, 8 Tenex User's Guides, 5 Tenex Memos (1970), 5 Tenex Monitor Manuals, and 4 Tenex Tecos. The date on the purchase order from the institute is 12/02/74. If you need any more information, just let me know. ann 4a

10-JAN-75 1013-EST KIDSTON at BBN-TENEXA: documentation
Distribution: WEINBERG AT SRI-ARC
Received at: 10-JAN-75 07:15:23 5

Ann,
could you let me know the manuals you ordered and the P.O. number if you have one and I will look into it.
Carol 5a

24-DEC-74 1844-PST KELLEY: identification userguide
Distribution: WEINBERG
Received at: 24-DEC-74 18:44:40 6

do you have it listed in doculist? it's a file journalized by KJM and modified by me. It is in <userguides,identifcations,> 6a

author Journal documents authored 7

DVN RWW POOH 21-JAN-75 12:45 25158
Maintenance of NLS-8 Documentation
Location: (HJOURNAL, 25158, 1:w)

POOH 21-JAN-75 15:10 25162

test

*****Note: Author Copy*****

7a

DVN POOH KIRK 17-JAN-75 17:21 25144
Informal Documentation Report for Week Ending Jan 17
Location: (HJOURNAL, 25144, 1:w)
*****Note: Author Copy*****

7b

POOH 15-JAN-75 12:19 25115
Set Tenex (protection for file named) CONTENT CONTROLS OK:
Message: This command does not work at all. You receive the error message "Illegal Protection Specified" When it is fixed, I suggest that a Reset Tenex command also be put in. This way it will be like all the other Set commands that have Reset commands also.
*****Note: Author Copy*****

7c

POOH 23-DEC-74 11:01 24799
Business Cards
Message: We are going to print new business cards very soon. I have samples of ten different formats. Anyone who wishes to order a card should come see me to choose a style and content. Orders will be taken through the first week in January.
*****Note: Author Copy*****

7d

POOH 8-NOV-74 10:08 24444
Tenex Documentation
Message: The following documentation can be ordered from BBN. Please let me know if you would like any of these manuals. I plan to place an order in the next week.
TENEX EXECUTIVE Manual:Belleville,Postel, Jake, Beck
TENEX JSYS Manual,Postel
TENEX USER7s Guide:Belleville,Postel, Jake, Beck
TENEX memos (1970):Belleville,Postel, Jake
TENEX Monitor Manual:Belleville,PostelJake
PAL11X,Postel, Jake
BCPL
TENEX TECO, Jake, Beck
*****Note: Author Copy*****

7e

Action 8

Info 9

READMAIL 10

POOH 21-JAN-75 15:10 25162

test

EKM 2-DEC-74 16:29 24645
Network Journal Delivery
Location: (GJOURNAL, 24645, 1:w)
*****Note: [ACTION] *****
10a

JHB 30-NOV-74 22:50 24636
Conolidation of the Feedback Mechanisms for NLS
Location: (GJOURNAL, 24636, 1:w)
*****Note: [INFO-ONLY] *****
10b

EKM 21-NOV-74 15:51 24570
NSW / NLS Plans
Location: (GJOURNAL, 24570, 1:w)
*****Note: [ACTION] *****
10c

MESSAGES (:Xb) 11

12

ME EXE PRO LOA PRO MESSAGE 13

EXE MES MOV MES JOURNALD 13a

EXE PRO DEL LAS 13b

< USERGUIDES, ARCLOCATOR,NLS;48, > 13c

Last Writer: KELLEY
Last Written: 7-AUG-74 15:50:08
13c1

< USERGUIDES, COMMANDS,NLS;144, > 13d

Last Writer: BECK
Last Written: 2-JAN-75 18:35:16
13d1

< USERGUIDES, COMMANDS-COM,NLS;2, > 13e

Last Writer: BECK
Last Written: 2-JAN-75 18:09:50
13e1

< USERGUIDES, DEX-PRIMER,NLS;11, > 13f

Last Writer: SYSTEM

test

Last Written: 8-JAN-74 15:50:51

13f1

< USERGUIDES, DEX-TRANSCRIPTION.NLS;3, > 13g

Last Writer: BECK
Last Written: 6-AUG-74 21:38:44

13g1

< USERGUIDES, DEX-TWO.NLS;1, > 13h

Last Writer: SYSTEM
Last Written: 21-JUN-73 10:16:00

13h1

< USERGUIDES, GLOSSARY.NLS;35, > 13i

Last Writer: KELLEY
Last Written: 9-OCT-74 20:32:31

13i1

< USERGUIDES, GLOSSARY.PRINT;3, > 13j

Last Writer: KELLEY
Last Written: 2-OCT-74 17:11:23

13j1

< USERGUIDES, HELP.NLS;2, > 13k

Last Writer: WEINBERG
Last Written: 13-NOV-74 15:21:45

13k1

< USERGUIDES, JOURNAL-NETSUB.NLS;15, > 13l

Last Writer: KELLEY
Last Written: 17-APR-74 12:15:03

13l1

< USERGUIDES, JOURNAL-QUERY.NLS;61, > 13m

Last Writer: KELLEY
Last Written: 10-DEC-74 23:32:50

13m1

< USERGUIDES, L10-GUIDE.NLS;323, > 13n

Last Writer: WEINBERG

test

Last Written: 14-NOV-74 09:14:32 13n1

< USERGUIDES, L10-GUIDE.PRINT;7, > 13o

Last Writer: WEINBERG
Last Written: 8-NOV-74 09:24:27 13o1

< USERGUIDES, L10USERGUIDES.PRINT;1, > 13p

Last Writer: SYSTEM
Last Written: 30-OCT-74 06:56:24 13p1

< USERGUIDES, LINEPROCESSOR.NLS;1, > 13q

Last Writer: VANNOUHUYS
Last Written: 7-JAN-75 19:24:03 13q1

< USERGUIDES, LINEPROCESSOR.PRINT;1, > 13r

Last Writer: SYSTEM
Last Written: 7-JAN-75 19:24:30 13r1

< USERGUIDES, LOCATOR.NLS;14, > 13s

Last Writer: PETERS
Last Written: 17-JAN-75 09:30:45 13s1

< USERGUIDES, MANUAL.NLS;29, > 13t

Last Writer: BECK
Last Written: 17-NOV-74 15:14:20 13t1

< USERGUIDES, NDDT-SYMBOLIC-DEBUGGER.NLS;4, > 13u

Last Writer: WEINBERG
Last Written: 6-NOV-74 16:24:55 13u1

< USERGUIDES, NDDT-SYMBOLIC-DEBUGGER.PRINT;4, > 13v

Last Writer: WEINBERG

test

Last Written: 6-NOV-74 16:25:45
13v1

< USERGUIDES, OP-GUIDE.NLS;52, > 13w

Last Writer: MEYER
Last Written: 9-JAN-75 10:03:07
13w1

< USERGUIDES, OP-INTRO.NLS;11, > 13x

Last Writer: MEYER
Last Written: 23-AUG-73 11:49:31
13x1

< USERGUIDES, PRIMER.NLS;19, > 13y

Last Writer: WEINBERG
Last Written: 17-JAN-75 15:51:16
13y1

< USERGUIDES, RSEXEC.NLS;1, > 13z

Last Writer: MEYER
Last Written: 12-JAN-74 11:51:58
13z1

< USERGUIDES, TENEX.NLS;115, > 13a@

Last Writer: MEYER
Last Written: 26-JUL-74 16:27:42
13a@1

< USERGUIDES, TUG.NLS;1, > 13aa

Last Writer: SYSTEM
Last Written: 24-MAY-73 21:27:55
13aa1

< USERGUIDES, VIEWSPECCARD.NLS;10, > 13ab

Last Writer: BECK
Last Written: 2-NOV-74 10:20:50
13ab1

< USERGUIDES, [ARCHIVE-DIRECTORY].;1, >
[<BECK>(USERGUIDES)[ARCHIVE-DIRECTORY].PC;1: File Not On-line;
If Archived, Use EXEC's INTERROGATE] 13ac

test

POOH 21-JAN-75 15:10 25162

Last Writer: BEACH
Last Written: 10-JAN-75 03:57:21

13ac1

POOH 21-JAN-75 15:10 25162

test

(J25162) 21-JAN-75 15:10;;;; Title: Author(s): Ann Weinberg/POOH;
Distribution: /POOH([ACTION]) ; Sub-Collections: SRI-ARC; Clerk:
POOH; Origin: < WEINBERG, POOH.NLS;50, >, 14-JAN-75 14:25 POOH
;;;;####;

test

this is the
comment to
togo with
this test item.

test

Robert L. Belleville
Research Engineer
Augmentation Research Center
Stanford Research Institute (415) 326-6200
Menlo Park, California 94025

Robert N. Lieberman
COMPUTER SCIENTIST
AUGMENTATION RESEARCH CENTER
Stanford Research Institute (415) 326-6200
Menlo Park, California 94025 Computer Ident: RLL

Robert N. Lieberman
COMPUTER SCIENTIST
AUGMENTATION RESEARCH CENTER
Stanford Research Institute (415) 326-6200
Menlo Park, California 94025 Computer Ident: RLL

Harvey G. Lehtman
Systems Programmer
Augmentation Research Center
Stanford Research Institute (415) 326-6200
Menlo Park, California 94025 x4174

James E. White
Systems Programmer
Augmentation Research Center
Stanford Research Institute (415) 326-6200
Menlo Park, California 94025

Kirk Kelley

test

Augmentation Research Center

Stanford Research Institute (415) 326-6200 x3506

Menlo Park, California 94025

Susan G. Roetter

Research Analyst - User Development

Augmentation Research Center

Stanford Research Institute
Menlo Park, California 94025

(415) 326-6200

Elizabeth K. Michael

Systems Programmer

Augmentation Research Center

Stanford Research Institute (415) 326-6200

333 Ravenswood Avenue, Menlo Park, California 94025 U.S.A.

Douglas C. Engelbart

Director

Augmentation Research Center

stanford research institute (415) 326-6200

333 ravenswood ave., menlo park, california 94025

Douglas C. Engelbart

Director

Augmentation Research Center

stanford research institute (415) 326-6200

333 ravenswood ave., menlo park, california 94025

test

James C. Norton

Assistant Director

Augmentation Research Center

Stanford Research Institute (415) 326-6200

333 Ravenswood Avenue, Menlo Park, California 94025 U.S.A.

James C. Norton

Assistant Director

Augmentation Research Center

Stanford Research Institute (415) 326-6200

333 Ravenswood Avenue, Menlo Park, California 94025 U.S.A.

James C. Norton

Assistant Director

Augmentation Research Center

Stanford Research Institute (415) 326-6200

333 Ravenswood Avenue, Menlo Park, California 94025 U.S.A.

James C. Norton

Assistant Director

Augmentation Research Center

Stanford Research Institute (415) 326-6200

333 Ravenswood Avenue, Menlo Park, California 94025 U.S.A.

Kenneth E. Victor

Systems Programmer

Augmentation Research Center

stanford research institute (415) 326-6200

333 ravenswood ave., menlo park, california 94025

Ann C. Weinberg Research Associate

test

Augmentation Research Center

stanford research institute (415) 326-6200

333 ravenswood ave., menlo park, california 94025

Jeanne M. Beck

Research Associate - User Development

Augmentation Research Center

stanford research institute (415) 326-6200

333 ravenswood ave., menlo park, california 94025

Jon Postel Technical Staff

Augmentation Research Center

stanford research institute (415) 326-6200

333 ravenswood ave., menlo park, california 94025

Stanford Research Institute (415) 326-6200

N. DEAN MEYER

RESEARCH ANALYST, AUGMENTATION RESEARCH CENTER

333 Ravenswood Avenue, Menlo Park, California 94025 U.S.A.

Stanford Research Institute (415) 326-6200

N. DEAN MEYER

RESEARCH ANALYST, AUGMENTATION RESEARCH CENTER

333 Ravenswood Avenue, Menlo Park, California 94025 U.S.A.

POOH 21-JAN-75 16:56 25164

test

(J25164) 21-JAN-75 16:56;;;; Title: Author(s): Ann Weinberg/POOH;
Distribution: /POOH([ACTION]) ; Sub-Collections: SRI-ARC; Clerk:
POOH; Origin: < WEINBERG, CARDS,NLS;5, >, 21-JAN-75 08:26 POOH
;;;; ####; ####;

testtwo

this is

testtwo

Robert L. Belleville
Research Engineer
Augmentation Research Center
Stanford Research Institute (415) 326-6200
Menlo Park, California 94025

Robert N. Lieberman
COMPUTER SCIENTIST
AUGMENTATION RESEARCH CENTER
Stanford Research Institute (415) 326-6200
Menlo Park, California 94025 Computer Ident: RLL

Robert N. Lieberman
COMPUTER SCIENTIST
AUGMENTATION RESEARCH CENTER
Stanford Research Institute (415) 326-6200
Menlo Park, California 94025 Computer Ident: RLL

Harvey G. Lehtman
Systems Programmer
Augmentation Research Center
Stanford Research Institute (415) 326-6200
Menlo Park, California 94025 x4174

James E. White
Systems Programmer
Augmentation Research Center
Stanford Research Institute (415) 326-6200
Menlo Park, California 94025

Kirk Kelley

testtwo

Augmentation Research Center

Stanford Research Institute (415) 326-6200 x3506

Menlo Park, California 94025

Susan G. Roetter

Research Analyst - User Development

Augmentation Research Center

Stanford Research Institute
Menlo Park, California 94025

(415) 326-6200

Elizabeth K. Michael

Systems Programmer

Augmentation Research Center

Stanford Research Institute (415) 326-6200

333 Ravenswood Avenue, Menlo Park, California 94025 U.S.A.

Douglas C. Engelbart

Director

Augmentation Research Center

stanford research institute (415) 326-6200

333 ravenswood ave,, menlo park, california 94025

Douglas C. Engelbart

Director

Augmentation Research Center

stanford research institute (415) 326-6200

333 ravenswood ave,, menlo park, california 94025

testtwo

James C. Norton

Assistant Director

Augmentation Research Center

Stanford Research Institute (415) 326-6200

333 Ravenswood Avenue, Menlo Park, California 94025 U.S.A.

James C. Norton

Assistant Director

Augmentation Research Center

Stanford Research Institute (415) 326-6200

333 Ravenswood Avenue, Menlo Park, California 94025 U.S.A.

James C. Norton

Assistant Director

Augmentation Research Center

Stanford Research Institute (415) 326-6200

333 Ravenswood Avenue, Menlo Park, California 94025 U.S.A.

James C. Norton

Assistant Director

Augmentation Research Center

Stanford Research Institute (415) 326-6200

333 Ravenswood Avenue, Menlo Park, California 94025 U.S.A.

Kenneth E. Victor

Systems Programmer

Augmentation Research Center

stanford research institute (415) 326-6200

333 ravenswood ave., menlo park, california 94025

Ann C. Weinberg Research Associate

testtwo

Augmentation Research Center

stanford research institute (415) 326-6200

333 ravenswood ave., menlo park, california 94025

Jeanne M. Beck

Research Associate - User Development

Augmentation Research Center

stanford research institute (415) 326-6200

333 ravenswood ave., menlo park, california 94025

Jon Postel Technical Staff

Augmentation Research Center

stanford research institute (415) 326-6200

333 ravenswood ave., menlo park, california 94025

Stanford Research Institute (415) 326-6200

N. DEAN MEYER

RESEARCH ANALYST, AUGMENTATION RESEARCH CENTER

333 Ravenswood Avenue, Menlo Park, California 94025 U.S.A.

Stanford Research Institute (415) 326-6200

N. DEAN MEYER

RESEARCH ANALYST, AUGMENTATION RESEARCH CENTER

333 Ravenswood Avenue, Menlo Park, California 94025 U.S.A.

POOH 21-JAN-75 17:13 25165

testtwo

(J25165) 21-JAN-75 17:13;;;; Title: Author(s): Ann Weinberg/POOH;
Distribution: /POOH([ACTION]) ; Sub-Collections: SRI-ARC; Clerk:
POOH;   ####; Origin: < WEINBERG, CARDS.NLS;5, >, 21-JAN-75 08:26
POOH ;;;;   ####; ####;

DCE 21-JAN-75 18:18 25166

Visit Log: Larry MacKechnie, Australian Post Office, 21 Jan 75

Returning Thursday and Friday, 23, 24 Jan. Apparently probing telecommunications, teleconferencing, etc. In Australia, the Post Office runs all of the communicaions systems (no private telegraph or telephone companies). Will arrange a short talk by him if enough people inform Jeanne Leavitt that they would be intersted. I'd like for RWW and JCN to be sure to meet him.

DCE 21-JAN-75 18:18 25166

Visit Log: Larry MacKechnie, Australian Post Office, 21 Jan 75

Larry had written ahead to arrange a visit with us. He is making a tour in the U.S. He is mainly pursuing techniques of teleconfering, but the APO has much broader activities and potential application interests in what we do than just teleconferencing. He was here for about an hour late this afternoon. I gave him a quick look at what we have, including letting him watch me use DNLS to put together the following listing of reference documentation. He took the pile of documents back to the Cabana. He has tomorrow relatively free and will try to read through as much as he can. He'll come back on Thursday, 1000 (with a non-technical friend). 1

List of documents that he took with him: 2

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

D. C. Engelbart, COORDINATED INFORMATION SERVICES for a DISCIPLINE- OR MISSION-ORIENTED COMMUNITY, paper presented at the Second Annual Computer Communications Conference, San Jose, California, 24 January 1973. (Journal, dated 12 Dec 72 -- Mjournal, 12445,1: xhmz) 2b

D. C. Engelbart, DESIGN CONSIDERATIONS FOR KNOWLEDGE WORKSHOP TERMINALS, paper presented at the National Computer Conference, New York City, June 1973. (Journal -- 14851,) 2c

D. C. Engelbart, R. W. Watson, J. C. Norton, THE AUGMENTED KNOWLEDGE WORKSHOP, paper presented at the National Computer Conference, New York City, June 1973. (IJOURNAL, 14724,) 2d

Charles H. Irby, DISPLAY TECHNIQUES FOR INTERACTIVE TTEXT MANIPULATION< Paper presented in xxxxx 2e

Donald I. Andrews, "Line Processor: A Device For Amplification of Display Terminal Capabilities for Text Manipulation," ... 2f

D. C. Engelbart, AUGMENTING HUMAN INTELLECT: A CONCEPTUAL FRAMEWORK, SRI Project AFOSR-3223, October 1962 (XDOC -- 3906,) 2g

D. C. Engelbart and ARC Staff, ADVANCED INTELLECT-AUGMENTATION TECHNIQUES, Final Report on NASA Contract NAS1-7897, July 1970 (XDOC -- 5140,) 2h

Augmentation Research Center, NETWORK INFORMATION CENTER AND COMPUTER AUGMENTED TEAM INTERACTION, Technical Report on project RADC-TR-71-175, June 1971 (Journal, dated 1 July 71 -- 8277,) 2i

Visit Log: Larry MacKechnie, Australian Post Office, 21 Jan 75

Augmentation Research Center, "Output Processor Users' Guide," 23 Aug 73, (Journal -- 12209,) 2j

Richard W. Watson, "Knowledge Workshop Terminl Systems" 2k

Dirk H. Van Nouhuys, Publication Support for the Jovial J73 Progamming Language Specification Document," (21883,) 2l

TNLS-8 Primer (23911,) 2m

Basic TNLS-8 Course (24207,) 2n

Richard W. Watson, National Software Works Developments (23352,) 2o

DCE 21-JAN-75 18:18 25166

Visit Log: Larry MacKechnie, Australian Post Office, 21 Jan 75

(J25166) 21-JAN-75 18:18;;;; Title: Author(s): Douglas C. Engelbart/DCE; Distribution: /JML([ACTION]) RWW([ACTION]) JCN([ACTION]) SRI-ARC([INFO-ONLY]) ; Sub-Collections: SRI-ARC; Clerk: DCE;

KIRK 21-JAN-75 20:49 25170

Announcing a new version of NLS: NLS-8,7 21-JAN-75

A note about the list of NLS changes
> The following changes in NLS were gleaned from <NLS, TASKS, NIC-NLS-CHANGES>. They were written by the programmer who made the change and were not originally intended for public consumption. I have deleted the most criptic notes (the ones I couldn't figure out) as well as ones that would have no noticable user effect. This is a complete list of changes that would be noticable to a user are listed. I made minor changes but did not have the time to check with the programmer to see what he/she meant. "Features added or changed" have been separated from "Bug fixes". If this version works out correctly, it will be moved to Office-1 in a few weeks. This file can be used as a source for Application's announcement document to those somewhat less sophisticated users. 1

Features added or changed. 2

New CML
> should not be noticible to users unless they have non-category one userprograms <see -- documentation, help, userprograms,dt> with a CML interface. The new CML facilitates the split of NLS into a front-end and a back-end by passing numbers instead of commandwords. Those who have personal user-programs that need to be converted, should see DSM or KLM who have a program that does it. 2a

JDH 13 DEC don't put page break before first line of print (E vspc) 2b

HGL 6 DEC Modified help parsefunctions to prompt correctly with fixed intepreter. Added "dumprompt" for the same reason. 2c

JDH 6 DEC TNLS print pagination (E vwspc) 11 inches instead of 11 1/2. 2d

JDH 6-DEC deflinmax to 60 (sets linmax in userop page) instead of 63. currently used only by tnls print, could also be used by quickprint. 2e

HGL 5 DEC Changed messages put out at ?-time to avoid overflowing strings on IMLACs with long vectors. Messages changed are "<CTRL-Q> for HELP" and "<CTRL-S> for SYNTAX" to "<CTRL-Q>: HELP" and "<CTRL-S>: SYNTAX". 2f

Changed help error message form "Item not found" to "[item]? Try a synonym" which stays on the screen 2g

Made help searches faster hgl 2 DEC Does lookup with type nxtname rather than call on seqsch which does namingrp. 2h

Announcing a new version of NLS: NLS-8,7 21-JAN-75

KJM 20 Nov, -- when blank lines and stmt numbers on the right are both on, and the statement number requires a separate line, no extra blank line should be printed on an output quickprint. It does now, but won't with this fix. 2i

KIRK 18 NOV -- changed load file command to add the characters "-rem" to the end of the local remote file name instead of to the beginning. 2j

Bugs fixed. 3

dsm 8-JAN-75 Fixed bug which would not let one load a subsystem which had a command word which was the same as the subsystem name. 3a

jdh 7-JAN-75 Changed user-options reset recognition mode to level 2 command 3b

dsm 6 JAN 75 Fixed bug in the run program command that used the wrong program number with user subsystems loaded 3c

kirk 3 JAN Increased local string escname for statementnames to 100 characters instead of 40 to fix "string to long" bug with a line of dashes. 3d

kirk 2 JAN Fixed so that loading a file from another site puts the file in your connected directory. 3e

KJM 9-jan-75 Fixed bug in Line Processor error recording; error data file MUST already exist (user can't create it). 3f

hgl 13 DEC fixed Page default change to "on" that messed up help. 3g

Help initialization changed: hgl 12 DEC This one is for NIC, not REL as before!!! 3h

hgl 12 DEC Changed help calling sequence to take three parameters so we can check if we are being called recursively. On recursive calls check under universal for second level command word, then upon return, if not found put the user at the description of the subsystem he is in. If command found under universal, put her at that command description. Fixes loop bug discovered when DB didn't have description of command under subsystem. 3i

ekm fixed bug in calc to allow user to specify an accumulator as input to an arithmetic operation. 3j

hgl 9 DEC Moved calls on dismes(0) to before calls on qdisp to remove messages remaining on screen before display creation. 3k

KIRK 21-JAN-75 20:49 25170

Announcing a new version of NLS: NLS-8,7 21-JAN-75

hgl 9 DEC Truncate promptstr if too long (>= 20 characters.) 3l

hgl 6 DEC Changed max prompt string length. (Currently uses stack space. Should also truncate if too long!) 3m

hgl 5 DEC Truncate string if too long to fit in column. 3n

dsm 5-dec-74 Fixed force case branch bug 3o

dsm 4-dec Four files were changed to fix bugs in the priht command 3p

jdh 3-DEC Changed "HEARALD" in User-Options to a second level command. 3q

JDH 27 NOV set name delims to () when delivery inserts "action", "info", and "author" branchs. Correction so they are set only when the new statment is inserted. 3r

r1b2 21 NOV [DOC's note: does this fix questionmark problems in IMLAC DNLS?] Transposed two groups of statements in this procedure in order to make the IMLAC version of NLS work. Thoroughly pragmatic solution inasmuch as we don't know why it works other than realizing that there is some order dependency in the sout of the 13B and the call on dismes. NOTE: Does this transposition affect anything else??? E.g., what about when 132 comes up? 3s

DSM 21 NOV fixed bug in calculator turned up by command word changes. 3t

HGL 15 NOV Changed action when BC and BW recognized by command parser. (Primarily to prevent bad stuff happening in HELP.) 3u

EKM 14-NOV-74 08:26 24500 output remote printer
Message: Output remote printer now works at SRI-ARC and will be fixed at Office-1 tonight. 14-NOV-74 Susan tested it vigorously for us at RADC. It wants tip number followed by port number (both decimal). Jim would you notify any of your users that might need to know this?
*****Note: [INFO-ONLY] *****

3v

KIRK 21-JAN-75 20:49 25170

Announcing a new version of NLS: NLS-8.7 21-JAN-75

(J25170) 21-JAN-75 20:49;;;; Title: Author(s): Kirk E. Kelley/KIRK;
Distribution: /SRI-ARC([INFO-ONLY]) ; Sub-Collections: SRI-ARC;
Clerk: KIRK;

KIRK 21-JAN-75 20:53 25171

Sendmail subsystem command 'Number Assign' doesn't work

When I say "Send the mail after using the "Numer Assign" command, Sendmail says "Number reserved by someone else!". This happened three times in a row. The first time I said "Goto Base" and did some editing before I "Quit" back to sendmail and said "Send". 1

KIRK 21-JAN-75 20:53 25171

Sendmail subsystem command 'Number Assign' doesn't work

(J25171) 21-JAN-75 20:53;;;; Title: Author(s): Kirk E. Kelley/KIRK; Distribution: /BUGS([ACTION]) FEED([ACTION]) JDH([INFO-ONLY]) ; Sub-Collections: SRI-ARC BUGS; Clerk: KIRK;

KIRK 21-JAN-75 20:55 25172

LIT typein bug

Backspacing at the end of a literal typein line in DNLS causes most of the previous line break at a funny place and appear on the next line when typing forward enough to reach the margin. All DNLS users should know what I'm talking about. 1

KIRK 21-JAN-75 20:55 25172

LIT typein bug

(J25172) 21-JAN-75 20:55;;;; Title: Author(s): Kirk E. Kelley/KIRK; Distribution: /FEED([ACTION]) ; Sub-Collections: SRI-ARC; Clerk: KIRK;

DSM 21-JAN-75 23:36 25173

Sysgd Lives!

The file (nls,sysgd,) has been recreated. It contains a named statement for each NLS procedure. This statement contains a link pointing to the current source code for the procedure and gives a brief description of the procedure's arguements and function. It can be used as your "external names link file" which allows you to "Jump (to) Name External" on any nls procedure name. 1

DSM 21-JAN-75 23:36 25173

Sysgd Lives!

(J25173) 21-JAN-75 23:36;;;; Title: Author(s): David S. Maynard/DSM; Distribution: /SRI-ARC([INFO-ONLY]) ; Sub-Collections: SRI-ARC; Clerk: DSM;

RLL 22-JAN-75 00:04 25174

Sug: cutting down the typing for TNLS users for JUMP File Return command

One of the complaints often heard is the verbose print outs TNLS gives the users. Here is one spot which would be nice to eliminate some typing. When using th jump file return command and one selects the file (which is printed out) the file name (as a link) is again printed (per usual when going to another file) It would be nice to supress this second printout. I realize this might be very hard if not impossible since the command and the loading of the file are different processes. lso it would meaa some inconsistency. In any case this suggestion is on record for future considerations (right gang??) Rob

1

RLL 22-JAN-75 00:04 25174

Sug: cutting down the typing for TNLS users for JUMP File Return command

(J25174) 22-JAN-75 00:04;;;; Title: Author(s): Robert N. Lieberman/RLL; Distribution: /FEED([ACTION]) ; Keywords: suggestion; Sub-Collections: SRI-ARC; Clerk: RLL;

POOH 22-JAN-75 08:38 25175

help Suggestions and Criticisms

Thanks for the feedback Jeanne. The definition for O>K> has been rewritten and I think it takes care of your suggestion. The information about start up is not yet done, but I will be sure to include your suggestions when I get there. You can look in xhelp for the def of ok and I think it will be in help the next time a new one is brought up. 1

POOH 22-JAN-75 08:38 25175

help Suggestions and Criticisms

(J25175) 22-JAN-75 08:38;;;; Title: Author(s): Ann Weinberg/POOH;
Distribution: /JMB([ACTION]) FEED([INFO-ONLY]) ;
Sub-Collections: SRI-ARC; Clerk: POOH;

JAKE 22-JAN-75 09:08 25177

New system announcement - a real win for users

Find Kirk's response to my (and probably others) request for notification of new system changes a real win. Thanks to all concerned. Don't think the brevity is a particular problem since the person involved is named and can be asked about any particular detail. Outside users might need a little more clarification. 1

JAKE 22-JAN-75 09:08 25177

New system announcement - a real win for users

(J25177) 22-JAN-75 09:08;;;; Title: Author(s): Elizabeth J. (Jake) Feinler/JAKE; Distribution: /NPG([INFO-ONLY]) ; Sub-Collections: SRI-ARC NPG; Clerk: JAKE;

POOH 22-JAN-75 09:18 25178

the process sendmailform command

If you are trying to send a file which you have specified and then use the insert status form command to edit something, when you use the process sendmailform command, it gets hung up on the file name. It seems that the enitre origin statement is printed and the process command can only take what is in angle brackets. I'm not sure I have explained this very well so if you have any questions, please come ask. thanks ann. 1

POOH 22-JAN-75 09:18 25178

the process sendmailform command

(J25178) 22-JAN-75 09:18;;;; Title: Author(s): Ann Weinberg/POOH; Distribution: /FEED([ACTION]) ; Sub-Collections: SRI-ARC; Clerk: POOH;

SRI-ARC Mid Point NSW Status Report 1

INTRODUCTION 2

We are involved in three task areas, the NSW PDP 11 based Frontend; Protocol design, documentation, and implementation; and tool building and modification. Each of these three areas is discussed below. A fourth area, dealing with computer support that affects the other three is broken out separately. 2a

The discussion lists accomplishments todate, problems along the way worth mentioning, work yet to be done and general worries and concerns requiring NSW management attention that are outside our control. 2b

We have tried to be open and honest about where we are at so that NSW management can plan accordingly. As indicated in spots below, we are behind where we would feel most comfortable and wanted to be by now, but basically feel we are in good shape for the July 1 date if there are no big surprises in the next 2-4 months. We still hope to be able to run NLS from the Frontend without the Works Manager so that there will be something for NSW users on July 1 in case there are delays in total system integration. 2c

Some general things that we have learned about working on a large new multi-client, multi-contractor, network distributed system are worth pointing out for future reference: 2d

1) We think the NSW system will be a better system as a result of the multi party approach because of the increased cross fertilization. 2d1

2) Design and requirements documentation is needed for adequate communication. This documentation is needed at both the overview and detailed technical levels. More resources should be acknowledged by client and contractors as required in this area and budgeted for accordingly. We feel we have performed well at intermediate to technical levels in our design documentation, but wish we had more resources for higher level documents. Design documentation has slowed us behind where we would like to be in several subtask areas and feel that we should have estimated more time for this type of important work. In the future we think that we will assume that about 20% of technical resources should be involved in this type of activity. 2d2

3) More resources should be budgeted to handle new tasks that only show up as the work proceeds, we think we all did well to see clearly the main problems and tasks at proposal time and to

budget appropriately for them with some small amount of safety factor. However, we think the client should recognize that the initially uncommitted resource factor should be larger. 2d3

4) We should allow more resources in the future for system integration design issues, communication, and mutual education etc. 2d4

It is assumed that readers of this document are familiar with NSW concepts and work generally. 2e

PROTOCOLS 3

INTRODUCTION 3a

Although as indicated below, we are about one month behind where we would feel most comfortable to be at this point, we feel that SRI-ARC's Protocol work will be completed on schedule. Under the section on worries there are some serious concerns about protocol application package implementations that we are not responsible for. Failure to deal with these concerns immediately will impact delivery of the NSW system. 3a1

PROCEDURE CALL PROTOCOL (PCP) 3b

DESIGN 3b1

PCP has been designed, documented, critiqued, and modified accordingly. Version 2 documentation has been out for a month or two; version 3 will not be issued until we having a running implementation on both Tenex and ELF. In the meantime, an on-line file <NLS>PCPV2CHANGES.TXT, accessible via FTP, contains design changes -- new features required by NSW, and bugs and deficiences encountered during implementation. Interest in the PCP work is widespread and growing; over 50 people have asked to be placed on the distribution list for PCP documentation. 3b1a

IMPLEMENTATION 3b2

We're behind schedule on implementation, having first gone through an unexpected design iteration which delayed coding progress. We had hoped the Tenex implementation would be up and running by now; it looks now like the end of FEB before it will be up. 3b2a

The ELF implementation has not been started, but for the most part it's the very same code currently being written for Tenex (recompiled, of course). 3b2b

YET TO DO 3b3

Complete Tenex implementation and debug 3b3a

Re-write low-level inter-process communication code for ELF 3b3b

Debug ELF implementation 3b3c

Write Version 3 documentation 3b3d

PROTOCOL PACKAGES (BATCH JOB, FILE, and NVT) 3c

DESIGN 3c1

All have been designed, documented, critiqued, and modified accordingly. Version 2 documentation has been out for a month or two; version 3 will not be issued until we having a running implementation on both Tenex and ELF. 3c1a

IMPLEMENTATION 3c2

No implementation has yet begun (SRI is not contractually responsible for these implementations); NSW is therefore no doubt behind schedule by any standards. 3c2a

We've had some difficulty in conveying the concepts embodied in the package descriptions; the preparation of additional documentation and question answering will slow implementation. More high level concept documentation would have helped here. 3c2b

YET TO DO 3c3

Implement for Tenex and debug (responsibility lies with BBN?) 3c3a

Implement for Elf and debug. 3c3b

Responsibility lies with ADR; we're not sure that ADR really realizes that. In particular, ADR must implement a Batch Job Package, and a File Package, both of which run on ELF but are really no more than interfaces to B4700 services. We're not sure they realize that they have to implement precisely these primitives, not just a set of primitives from which these can be constructed. Also, ADR may have (want) to implement the NVT Package to handle whatever B4700 console inferface is to be provided. 3c3b1

After both Tenex and Elf are running, issue final documentation (Version 3) 3c3c

WORRIES 3d

We are still uncertain about division of responsibility on FILE and NVT package implementation (for Tenex); how much is BBN going to do? 3d1

MCA is asking enough questions and of a kind, that we know that they understand relationship of Protocols to their task; we're not getting the same kind of questions from (or feeling about) ADR and/or Gunter. However, we haven't had time (and don't know enough about ELF) yet to write a PCPELFINT document, which ADR needs. 3d2

We're hearing expectations/suggestions for user capabilites which, as far as we know, are not being assigned to people for design and implementation. 3d3

E.g. the ability for a user to request that a particular cataloged procedure be run, as opposed to a request to invoke a particular batch system and then request the cataloged procedure via appropriate JCL. 3d3a

We don't have the "white box" document and the main WM design document from MCA which would be useful to us. 3d4

RJE and Batch job scenarios still seem up in the air in peoples minds and there does not yet seem to be wide agreement or understanding of what is required and who is responsible for what. 3d5

FRONTEND 4

INTRODUCTION 4a

We are pushing to have the first version of the Frontend up and running by March 1 when our computer goes away, although this may be somewhat optimistic. April 1 is probably more realistic. 4a1

Implementation strategy is to make the Frontend operational on the PDP 10 first, as we are committed to having an operational version for both machines, and there are presently better debugging facilities on the PDP 10. We will then recomplie and move to the PDP 11. 4a2

COMMAND META LANGUAGE 4b

The Command Meta Language (CML) and compiler are being modified to reflect the need for NSW contructs and these have been documented. 4b1

L 10 LANGUAGE COMPILER 4c

The L 10 compiler for the PDP 11 has been written and is in the process of final checkout. We are roughly on schedule on manpower estimated here but somewhat behind time wise as resources were shifted to other Frontend tasks. Some improvements in the language (coroutines, improved signal, and call return mechanisms for example) were made during the conversion process and implemented in the PDP 10 version as well. The PDP 10 version runs. Reasons for choice of L 10 were documented. We anticipate no problems here. 4c1

COMMAND LANGUAGE INTERPRETER 4d

The Command Language Interpreter has been written and partially checked out with portions of NLS. 4d1

OPERATING SYSTEM INTERFACE 4e

The Operating System Interface (OSI) has been designed and documented and is beginning implementation. 4e1

WORRIES 4f

We still have some worries about the Virtual memory features of ELF getting done on time and having the right capabilities. There is a new release of ELF out that we need to get into soon to see what is there. 4f1

It is important that the Works Manager, Protocols, and NLS be in a state early in April to allow integration of the whole system to begin. 4f2

TOOL TASKS 5

INTRODUCTION 5a

We are 4-6 weeks behind where we would like to be at this point given the quite ambitious list of things we want to accomplish with NLS by July. One of the key tasks was to get the new NLS 8 system up and operational, as it contained in its organization the foundation for future work, and improvemets and additions to the user interface. This task was accomplished, but all the nitty gritty little details involved in such a major transition required more resources than

anticipated. There is a lesson in this for the NSW, in that many important details that will need attention and resources will only be clear once we have real users running on the system. 5a1

We should probably get some guidance on the priority of the remaining tasks so that if in a month or so it should appear that we will not be able to have all tasks completed by July we can have the really important one completed. NLS, in any case, will be operational fully integrated into the NSW environment by July. 5a2

NLS 5b

NLS 8 OPERATION ON OFFICE-1 5b1

Task completed and turned over to ARC Applications Group. 5b1a

CREATING AN INITIAL SET OF NSW TOOLS 5b2

SEPARATE THE BACKEND AND FRONTEND PROCESSES 5b2a

This task involves insertion of NLS as a tool fully embedded in the NSW environment and is important as a task in its own right and as a test bed for many NSW ideas. What must be done is split the NLS source code into Backend and Frontend files, identify variables and routines apparently needed by both processes and decide what to do about them. This task is almost complete. 5b2a1

WRITE THE GRAMMARS FOR THESE TOOLS 5b2b

The grammars for the NLS editor (currently the base subsystem) and for commands that are now part of supervisor have been rewritten and are being used by to test the Frontend. 5b2b1

The remaining grammars are scheduled for completion by March 1. 5b2b2

REWRITE THE BACKEND CODE TO SATISFY THE REQUIREMENTS OF THE PROCEDURE CALL PROTOCOL AND NSW TOOL INTERACTION PROTOCOLS 5b2c

The document defining the NLS tool Backend execution routines is almost complete and will be reviewed this week. 5b2c1

We will begin coding the execution routines to interface with PCP as soon as the design has been reviewed. We hope

to have at least the initial tool, the NLS editor, running by March 1. 5b2c2

REWRITE BACKEND FILE CODE TO USE OSI. 5b2d

This task is mentioned below. 5b2d1

DEFINE NSW HELP SYSTEM AND DATA STRUCTURE 5b2e

We are currently writing the definition of a NSW Help data base for use with the NLS based Help tool. We expect to be finished at least by March 1. We expect online Help material for Works Manager and other tools to utilize this capability until a fancier NSW Help facility can be funded and constructed. 5b2e1

We are writing a detailed description of the modifications to the current NLS implementation to make Help work in the NSW environment. We believe these changes are minor. 5b2e2

USER PROFILE SYSTEM IN THE NSW ENVIRONMENT NOT DEFINED 5b2f

The User Profile for the tools that are derived from NLS should work much as it does now. It is unclear how this fits into the overall NSW, Works Manager, Frontend user profile and other tools. The concept of the User Profile is an NSW wide one and the type of data needed in it is more general than that needed by just NLS. This task seems to be a hole between SRI and MCA and needs management attention. 5b2f1

IDENT SYSTEM NOT DEFINED 5b2g

This area was not included in our charter. For the first year of NSW we propose to use the current NLS Ident file and implementation. Ultimately there should only be Works Manager data bases in this domain. This is another hole that needs attention. 5b2g1

COBOL PROGRAMMING IN NLS AND REMOTE JOB ENTRY 5b3

COBOL USERS' TRAINING PACKAGE 5b3a

We have discussed this briefly with the documentation group. The main task seems to be to define and relate NLS file structure and basic editing commands to the structure required by COBOL. Working with an Air Force

COBOL program this weekend leads us to believe the relationship is straight forward. 5b3a1

We expect to transfer recommendations for the training package to documentation for writing by February 20. 5b3a2

INTERFACE TO NSW REMOTE JOB ENTRY (RJE) 5b3b

Tasks here are still somewhat uncertain as the whole RJE Batch needs still have some lack of definiteness. What is envisioned here is whatever is required to use NLS to create a source file, JCL file and get them into the NSW RJE world. We consider this and important task and hope this area gets clarified soon by all the parties involved. 5b3b1

Note the July 1 delivery is also dependent on Works Manager and connection to the Burroughs 4700 being available for final testing at an appropriate date. 5b3b2

FILE STRUCTURE AND NEW NLS ENTITIES 5b4

MODIFICATION OF THE EXISTING NLS FILE STRUCTURE 5b4a

The modified file system has been designed and documented. New routines are needed to manipulate properties and subtrees. The Operating System Interface (OSI), with the appropriate file primitives, will not be available until the end of March. . 5b4a1

We are scheduled to begin coding 1/20 in a version not requiring OSI and to have it running and available to the graphics system by 2/15. The code will then be inserted into the OSI. 5b4a2

NEW NLS HEADING ENTITITY 5b4b

The file structure and OSI will be able to accommodate the 'Heading Property' by July 1. However, the many questions of how this is to be portrayed on various classes of terminals, what the output processer, both hardcopy and COM, is to do with it, etc. are still undefined. 5b4b1

As it is possible to handle headers through Output Processor directives this task has been given a relatively low priority, at this point, we happen to feel that this is an important task. We need some guidance here. 5b4b2

DOCUMENT PRODUCTION: THE OUPUT PROCESSOR AND NLS 5b5

We have been working with George Lithograph to enable Output Processor generated tapes for COM to run on their Singer 6000. This was done both to provide an alternative source of COM supply and to understand better the problems of moving between COM suppliers. 5b5a

We have been looking at the problem of obtaining input from offline cassette type devices. We now have the capability to run input from cassette devices through the high speed Line Processor lines and a PDP 10 program that ADR could easily transcribe for the PDP 11 that supports ICP Termicette, Techtran, and Three-Phoenix Cartridge devices and can be easily extended to others in the future. 5b5b

A simplified Interface to the Output Processor is being designed and will be there on July 1. 5b5c

The tasks below need some priority guidance. 5b5d

New NLS entities (see heading entity above) 5b5d1

Full justification 5b5d2

Permit a set of files to be processed with a single command to produce one document. 5b5d3

Changes to NLS editing system 5b5d4

Tabs 5b5d4a

Underlining 5b5d4b

Training Development in the document production area will include 3 tasks. 5b5e

A document providing guidelines to trainers 5b5e1

Preliminary training sessions 5b5e2

Consulting 5b5e3

GRAPHICS AND THE GRAPHICS WORKSTATION 5b6

DATA STRUCTURES FOR GRAPHICS 5b6a

This is essentially the new NLS file system explained above. 5b6a1

USER INTERFACE 5b6b

The interface planned for the July delivery is designed and, to some extent, implemented. 5b6b1

The entire graphics area is one that presents many needs for expansion and added capabilities after July 1. We feel that the graphics capability delivered in July will be useful but will need additional work. 5b6b2

THE GRAPHICS WORKSTATION 5b6c

The basic hardware for the graphics workstation is here and checked out. Line processor and NLS terminal control code has been written and debugged. 5b6c1

The 19 inch Tektronix display and the copier are expected to arrive on April 7 and should work without further software modifications. 5b6c2

PROOFING COM ON THE TEKTRONIX 5b6d

Doubtful by July 1. 5b6d1

SEQUENTIAL FILE INTERFACE TO NLS 5b7

This entire area has been under discusion for about a year and many of the problems have been identified in detail and solutions proposed. For the first year of NSW the specific file interfaces to be implemented are: 5b7a

Interface to NSW standard file formats to NLS files and back. 5b7a1

Interface between the Journal system and the SNDMSG system 5b7a2

Interface between the Journal and the mail facility offered by the NSW Works Manager (what is actually going to happen here?) 5b7a3

Interface between NLS files and the RJE system 5b7a4

We will begin work on these tasks on March 1. 5b7b

We have taken several tapes of text prepared on various text editors and input them into NLS. Each one had a different format and set of conventions for the tape that was not documented and could only be determined by a raw tape dump.

This has clearly demonsrated the need for some standards in this area. A suggested standard was sent out to the network community in December and is being modified from further experience for recommendation to the NSW. 5b7c

INTERFACE FOR THE INEXPERIENCED USER 5b8

SPECIALIZED, LIMITED, INTERACTIVE, TASK-ORIENTED MODULES 5b8a

These include tools for reading and sending mail, possibly calendar maintenance, etc. A scenario has been written for sending the mail and design outlines have been prepared in other areas. These will be distributed in the near future for review by members of the NSW steering committee. As more NSW users get into using NLS in various forms there will undoubtedly be changes and additions that people will want and thee should be some spare resources for this. 5b8a1

OFFLINE INPUT AND EDITING SYSTEM 5b8b

This task includes writing documentation and a scenario and approximately 1 week of programming. 5b8b1

TRAINING PACKAGES 5b8c

Some written scenarios and other documentation will be prepared to aid secretaries use the facilities. 5b8c1

WORRIES 5b8d

As indicated in specific tasks above. 5b8d1

OTHER TOOLS 5c

There seems to be a need for a number of additional small tools and grammars for miscellaneous tasks uncovered as NSW work has been going on. These need to be assigned to some group(s) to get done by July. 5c1

COMPUTER SUPPORT 5d

ARC is about to go through a major transition from having its own machine to accessing its computer power out on the net. We think (at least I do, others here have doubts) this is healthy, but there are still some loose ends and there will be other problems before a new steady state settles in. One of the tasks involved will be to get NLS running on the BBN machine, moving all our files etc. These tasks will take time. One

general benefit will be that both DNLS and TNLS will then run under standard Tenex. 5d1

Tenex time 5d2

We have a purchase order in the mill for computer time at BBN after March 1. Since we do not have a contract yet, that is causing some problem with our purchasing people, but BBN seems willing to allow a contingent order up to Feb 15. Hope we have a contract by then. If there is any slipup here we are in serious trouble as we are well past the point in the NSW development cycle where we can function without heavy computer support. 5d2a

PDP 11 AND OTHER HARDWARE 5d3

This is an attempt to list equipment status and point out problems that that need attention, mostly by us. The problems are listed in some detail so that you can draw any conclusions or concerns appropriate to the Gunter installation also. 5d3a

GETTING AN 11 SYSTEM OPERATIONAL BY MARCH 1 5d3b

Unless we do something to have a driver written for the DEC IMP 11A coming, we will have to use the 11 we already have as the network access machine. Retz does not feel responsible for this, does ADR? The machine has enough input ports, but it does not have a lineprinter. We do however have a 1200 baud Memorex (loan) and 1200 baud G.E Terminette serial printer coming that we could use in a pinch. 5d3b1

we need to establish desired operating up-time and determine if an operator(s) is needed. 5d3b2

PDP-11 THAT WE ALREADY HAVE: NSW DEVELOPMENT MACHINE 5d3c

Do not have a confirmed delivery date for back ordered equipment. 5d3c1

Do not have diagnostics assembled for ANTS IMP interface check out (We are working at converting these to run under ELF without a disk). 5d3c2

Do not have IMP side Network interface. 5d3c3

PDP-11 THAT IS COMING: NETWORK ACCESS MACHINE 5d3d

There is no driver on ELF for the DEC IMP 11A interface. 5d3d1

There are only 8 holes to plug terminals in. We need to order more. 5d3d2

It is not known how far away the Lineprinter can be operated. If we intend to have it in the same room as our old Data Products printer we need to investigate further. 5d3d3

DISPLAYS: FOR ARC AND CLIENTS 5d3e

Hazeltine could not deliver so we had to switch our order to Datamedia. We have been having problems with reliability on the Lear Siegler ADM II's,primarily keyboard problems. 5d3e1

LINEPROCESSORS: FOR ARC AND CLIENTS 5d3f

Everything looks fine here. 5d3f1

HARDCOPY TERMINALS: 5d3g

Everything looks fine here also 5d3g1

WORRIES 5d4

The main worry is that DEC and BBN will deliver all the needed hardware on time. The secondary worry is the software for the DEC IMP Interface driver and who is going to write it. 5d4a

RWW 22-JAN-75 14:21 25181

SRI-ARC Mid Point NSW Status Report

(J25181) 22-JAN-75 14:21;;;; Title: Author(s): Richard W. Watson/RWW; Distribution: /NPG([ACTION]) JBP([ACTION]) MEH([ACTION]) SRI-ARC([INFO-ONLY]) ; Sub-Collections: SRI-ARC NPG; Clerk: RWW; Origin: < WATSON, STATUS.NLS;9, >, 22-JAN-75 13:57 RWW ;;;;####;

Equipment Status Summary

EQUIPMENT STATUS 1

There are two parts to this document; the first is a short status summary with a list of things that are still hanging fire and some problems; the second is status details. 1a

SUMMARY: 2

AA. Getting an 11 system operational by March 1 2a

- Unless we do something before March 1 to resolve the DEC IMP operating system problem for the 11 coming we will have to use the 11 we already have as the access machine. The 11 we already have has enough input ports, but does not have a lineprinter. We do however have a 1200 baud Memorex (loan) and 1200 baud G.E Terminette serial printer coming that we could use in a pinch. 2a1

- we need to establish desired operating up-time and determine if an operator(s) is needed. 2a2

A. PDP-11 that we already have: NSW Experimental Machine 2b

- Do not have a confirmed delivery date for back ordered equipment. 2b1

- Do not have diagnostics assembled for ANTS IMP interface check out. 2b2

- Do not have IMP side Network interface. 2b3

B. PDP-11 that is coming: Network Access Machine 2c

- DEC will not have an IMP interface operating system available until May. 2c1

Our alternatives are: (my pref 1,2). 2c1a

1) adapt ELF or some other operating system 2c1a1

2) write our own 2c1a2

3) Spend some money to get the rest of the hardware together to debug and connect the NBS ANTS interface that we have on loan. (We are suppose to return it Feb 15, and will have to discuss with them if we want to keep it longer). 2c1a3

- There are only 8 holes to plug terminals in. 2c2

We must order more holes if we intend to plug in more than 8 terminals, which I think we do; cost $500/4 holes, up to 16. 2c2a

- It is not known how far away the Lineprinter can be operated. If we intend to put it in the same room as our old Data Products printer we need to investigate further. 2c3

C. Displays: For ARC and Clients 2d

- Everything looks fine here. Development will have 9 (Don included) and Applications 8 by March 1 for local use. 2d1

D. Lineprocessors: For ARC and Clients 2e

- Everything looks fine here also. Development will have 9 (Don included) and Applications 8 by March 1 for local use. 2e1

E. Hardcopy Terminals: For ARC and Clients 2f

- Everything looks fine here also. 2f1

STATUS: 3

A. PDP-11 that we already have: NSW Experimental Machine 3a

1) DEC equipment that is here and operating (P.O. B96273) 3a1

1	11/40-BA computer with tty	3a1a
1	KE11-E extended instruction set	3a1b
1	KW11-P programmable clock	3a1c
2	MF11-U 16K sen mem and logic	3a1d
3	KT11-D mem management	3a1e
1	H960-DA cabinet	3a1f
1	DD11-DA peripherial sys unit	3a1g

2) DEC equipment that is back ordered (P.O. B96273) 3a2

1	DU11-DA sync/Isoc com unit	3a2a
4	M-792 boot strap PROMs	3a2b
1	CR11 card reader	3a2c

1 TM11 9 chan mag tape unit and control 3a2d

3) PDP-11 side IMP interface 3a3

1 ANTs Network interface 3a3a

Recieved a couple of weeks ago. It came with Heathkit diagnostics that we must configure depending on our PDP-11 configuration. Ed has completed installation. He is now waiting for programmer help and has discussed assembling diagnostics to check out the device with KEV. 3a3b

KEV says that the ELF operating system which we intend to run has a utility driver that has already been developed which will drive the ANTs interface, presumably then there will be no development work needed here. 3a3c

4) IMP side IMP interface 3a4

Not installed yet. However it is available from BBN and has been given number-one priority for installation by ARPA. Mc Kinzie at BBN say that he expects installation to happen by early FEB, perhaps Lynch has a more specific date? Lynch and Watson have agreed to have it installed in our 316 IMP. 3a4a

Arrangements have been made with the AI group to use their 11 IMP port to check our 11 systems on the NET if need be while we are waiting to get ours. 3a4b

B. PDP-11 that is coming: Network Access Machine (P.O. B96684) 3b

1) DEC equipment (except Net interface) that will be delivered by Feb 1 latest. 3b1

1 11/40-BC computer with 30 cps tty 3b1a

1 KE11-E extended instruction set 3b1b

1 KW11-P programmable clock 3b1c

1 DH11-AA 16 chan prog async com mux 3b1d

2 DM11-DB terminal line adapter (4 chan ea) 3b1e

3 MF11-U 16K sen mem and logic 3b1f

4 MM11-U 16K sen mem boards 3b1g

1 KT11-D mem management 3b1h

Equipment Status Summary

1 LP11-WA 132 col 96 char line printer 3b1i

1 DU11-DA sync/Isoc com unit (single channel) 3b1j

4 M-792 boot strap loader PROMs 3b1k

2 DD11-A peripherial sys unit 3b1l

1 H960-DA cabinet 3b1m

2) PDP-11 side IMP interface (P.O. B96273) 3b2

1 DEC IMP interface 3b2a

Dec says delivery will be Feb 28 at the latest. It will come with check out diagnostics but no operating system software. Dec is developing operating software and intends to market, however they are quoting four month delivery and have not yet determined price. 3b2b

3) IMP side IMP interface 3b3

Available, part from Hawaii TIP, and part from BBN spares. Like the IMP side interface for the NSW machine it has also been given number-one priority by ARPA. Mc Kinzie at BBN will install as soon as possible in our 316 IMP. He has not yet confirmed an installation date. 3b3a

C. Displays: For ARC and Clients 3c

As of today: 3c1

Recieved ; 7 Delta Datas, 3 ADM-2, and 1 Hazeltine. 3c1a

Outstanding orders; 13 Data Media. 3c1b

In the mill ; 7 Data Media. (to replace Delta Datas as leases expire) 3c1c

1) For NSW (ARC Development) use (8 local, 1 remote) 3c2

3 DM (9229), expected delivery: end of Jan latest 3c2a

3 ADM2 (9229), Already here. 3c2b

Two of the three recieved do not work. They are under warrenty and must be returned to the factory for repairs. ICE, the service rep, is waiting delivery of two

replacement units which they expect to receive by this Fri. When they receive them they will come right out and recycle, so they say. 3c2b1

2 DM (750 D61), part batch of 5, expected delivery mid-feb latest. 3c2c

1 DD (20120), Already here and being used by Don Andrews. 3c2d

This display will be replaced with a DM when the lease expires. 3c2d1

2) For Office-1 (ARC Applications) use (8 local) 3c3

3 DM (750 D61), part batch of 5, expected delivery mid-Feb latest. 3c3a

3 DM (20120), part batch of 5, expected delivey: Jan 30 latest. 3c3b

2 DD (20120), Already here. 3c3c

One on loan to Hudson, other to MIT-S, expected return March 1 latest. Will be replaced with DMs as leases expire. 3c3c1

3) For Office-1 Client use (1 local, 6 remote) 3c4

Most Clients supply their own display or will hence this list does not reflect how many Lineprocessor Workstations are now (or will be) in use by clients. 3c4a

4 DD (20120), part batch of 7 3c4b

Three are in use now by ARPA. We will ship the fourth one early next week. 3c4b1

These will be replaced with DM as leases expire. 3c4b2

2 DM (20120), for NSRDC use, part batch of 5, expected delivery: Jan 30 latest. 3c4c

1 HAZ (20120), for local NIC use. 3c4d

This display is already here and on temporary loan to Mike Placko for SRI Office-1 use. It will be returned as

soon as Mike receives his, which he expects Jan 30 latest. 3c4d1

D. Lineprocessors: For ARC and Clients 3d

As of today: 3d1

ARC constructed: 3 3d1a

Cybernex constructed complete: 5 3d1b

Orders outstanding: 22 3d1c

1) For NSW use (8 local, 1 remote) 3d2

7 (30114), part batch of 10, 3d2a

Part of these have already been received and are in various stages of check-out. Rod expects to have a few completed by end of this week and rest by end of month. 3d2a1

2 (ARC) ARC constructed, 3d2b

One in use by Don Andrews and the other is in the Tasker display area being used by Belleville for DEX and graghics development. 3d2b1

2) For Office-1 ARC Applications use (8 local) 3d3

1 (30114), part batch of 10, 3d3a

In the Tasker console area connected to the 208A modem connected to the Tymshare TIP. 3d3a1

2 (30114) part batch of 10, 3d3b

Already received. In various stages of check out, which will be completed by Rod by end of Jan. 3d3b1

5 (20120) part batch of 5. 3d3c

Status same as above. 3d3c1

3) For Office-1 Client use (7 remote, 1 local, 1 spare) 3d4

1 (20120) slated for BRL use, part batch of 5. 3d4a

Construction costs will be transfered to the BRL contract. 3d4a1

Equipment Status Summary

2 (20120) slated for NSRDC use, part batch of 5. 3d4b

Construction costs will be transfered to NSRDC contract, These units have been recieved and have been checked out by Rod. As soon as we recieve two DM (expected next week) we will ship to NSRDC. 3d4b1

1 (20120) Slated for NIC use, part batch of 5. 3d4c

4 (20120) ARPA office, 3 installed, other will be shipped next week. 3d4d

1 (20120) Floater, part batch of 5. 3d4e

E. Hardcopy Terminals: For ARC and Clients 3e

As of today: (35) : 21 for ARC, 17 for clients. 3e1

Received: 3e1a

TI 725	: 11		3e1a1
TI 720	: 15		3e1a2
TI 735	: 4		3e1a3
Execuports	: 2		3e1a4
G.E 1200	: 1	(for local Appl use)	3e1a5
G.E 300	: 2	(canceling as lease expires)	3e1a6

On order: (3) 3e1b

TI 735	: 2	(for Watson, Postel)	3e1b1
Memorex 1200	: 1	(for local Appl, use)	3e1b2

MEH 22-JAN-75 14:54 25182

Equipment Status Summary

(J25182) 22-JAN-75 14:54;;;; Title: Author(s): Martin E. Hardy/MEH;
Distribution: /JCN([ACTION]) DCE([ACTION]) RWW([ACTION])
SRI-ARC([INFO-ONLY]) ; Sub-Collections: SRI-ARC; Clerk: SLJ;
Origin: < HARDY, STAT-DRAFT,NLS;5, >, 22-JAN-75 13:25 HDW ;;;;####;

JBP 22-JAN-75 15:58 25183

Comments on NVTP

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. 1

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). 2

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. 3

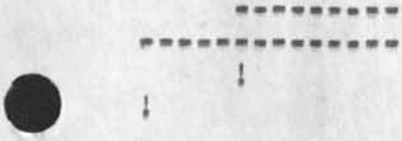
(We note that at times the same machine that supports the FE may be used in a non-NSW context to communicate with other machines, including those that support the WM or those that are also TBHs, using other protocols. This does not alter our basic simplifying principle since those other communications protocols and programs are completely independent of the NSW.) 3a

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). 4

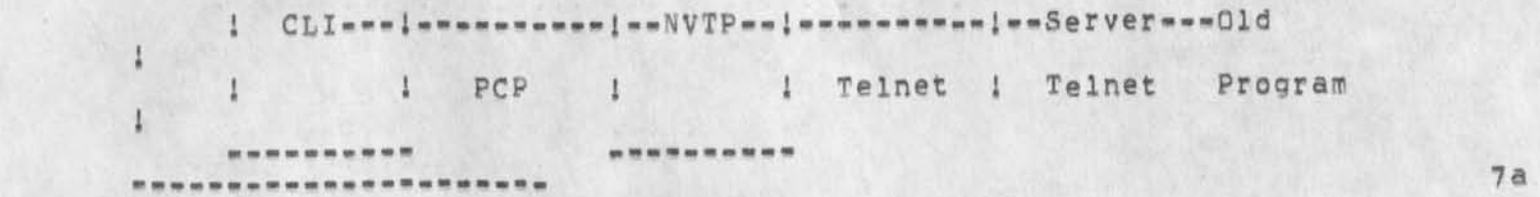
The TBHs support applications programs (e.g. text editors, compilers, reformatters, ...) called tools. These tools are or will be constructed with the NSW in mind, and will expect to communicate via PCP. Other applications programs, here called "Old Programs", were constructed to communicate only with 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. 5

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. 6

Case 1 7



Comments on NVTP



7a

Notes:

7b

CLI only does PCP calls.

7b1

NVTP merely copies data.

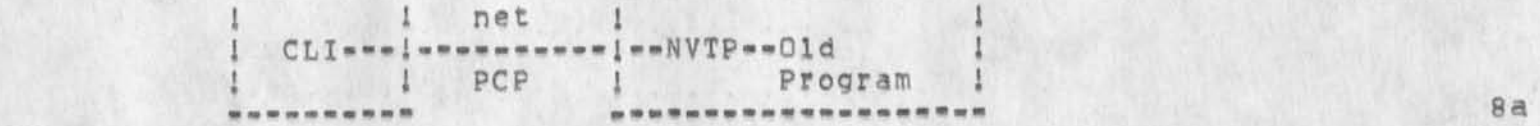
7b2

Server Telnet acts as controlling teletype to Old Program.

7b3

Case 2

8



8a

Notes:

8b

CLI only does PCP calls.

8b1

NVTP acts as controlling teletype to Old Program.

8b2

NVTP is a SMALL extension of Server Telnet program.

8b3

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

9

Most of Telnet's control options are for controlling aspects of the users 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 case 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.

9a

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. 9ai

An alternative communication strategy for Old programs has been suggested that would have the FE communicate with the Old Program using Telnet protocol. 10

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 are no known implementations 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. 10a

JBP 22-JAN-75 15:58 25183

Comments on NVTP

(J25183) 22-JAN-75 15:58;;;; Title: Author(s): Jonathan B.
Postel/JBP; Distribution: /JBP([INFO-ONLY]) ; Sub-Collections:
SRI-ARC; Clerk: JBP; Origin: < POSTEL, NVTP-COMMENTS,NLS;8, >,
21-JAN-75 16:42 JBP ;;;;####;