Feedback on New NLS Suggestions

(J23635) 17-JUL-74 07:52; Title: (Unrecorded) Title: Author(s): N, Dean Meyer/NDM; Distribution: /MDK([ACTION]) JCN([INFO-ONLY]) JHB([INFO-ONLY]) RLL([INFO-ONLY]) SRL([INFO-ONLY]) ; Sub-Collections: SRI-ARC; Clerk: NDM; Origin: (MEYER, RESP,NLS;2,), 17-JUL-74 07:39 NDM ;####;

Feedback on New NLS Suggestions

Compliments to the compilers of a rather complete list of problems with the New NLS.

Feedback on New NLS Suggestions

I endorse the suggestions made in Parts One and Two of (kudlick,newnls,1:why) with the following exceptions/additions: 1

I do not like the current implimentation of Split Window. I rarely want the split dead center; and it's a bother to have to split, then move the boundary. I would like to see a return to "Split Window at BUG CONFIRM" as in the old system. 2

The Line Processor bug which truncates each statement display to 72 characters must be fixed (23569,). 2a

I suggest "Set Terminal-type". 3

I object strongly to Load Program also running it. I need more control of the steps in programming-debugging than that. If the running is an option on the order of "Load Program LSEL and run it? Y/N CONFIRM", I would agree. 4

The command "Detach Subsystem" does not work. 5

When a User Program/subsystem (like MESSAGE) is loaded, the "WARNING -- no entry to program" for the L10 part misleads the user to thinking something is wrong. That message must be eliminated. 6

When a user programmed subsystem is attached, where the subsystem keyword is different from the program name (although this will be avoided in the future), the keyword, not the program name, should be given in the message "Subsystem XXX Attached". 6a

I strongly disagree with "unadvertised" single command keyword commands; we should be proud to advertise everything we do. If something is not in keeping with our philosophy, then either there's a better way to do it or else a modification of the general philosophy is in order. The suggested changes do not warrant a general modification of our verb-noun philosophy. 7

I use that generalization frequently in demonstrating how simple the system is to use. People ask "does one have to learn 200 different commands?" 7a

In TNLS, if I put two character searches (with an apostrophy) in an address expression, it shouldn't go to the beginning of the statement for the second search; it should search from the CM resulting from the DAE so far onward. 8

Actually I'm not sure if it always blows it; I was going to the second occurance of a given character, so both searches were for the same character (that shouldn't matter, though). 8a

Feedback on New NLS Suggestions

I weakly endorse the seperation of file handling commands (don't feel it's urgent) as described in (kudlick,newsubs,1:why). I was not particularly attracted to the seperation of the terminal control commands, but have no grounds for objection other than efficiency of the expert. 9

Donation for a Needy Straight

(J23636) 17-JUL-74 14:21; Title: Author(s): Elizabeth J. (Jake) Feinler/JAKE; Distribution: /SRI-ARC; Sub-Collections: SRI-ARC; Clerk: JAKE;

JAKE 17-JUL-74 14:21 23636

Donation for a Needy Straight

I am constructing a 'thing' and I need pieces of denim,,,,so am asking all of you to make the supreme sacrifice and give me any old jeans, kids jeans, cut-offs , legs of cut-offs, fallen off pockets, or what have you. I will accept any degree of togetherness from new (god forbid) to transparent threadbare, and any size, shape, or state of dirtiness. Give til it hurts to make a little old lady happy!!!!! 1

Other uses for Process Command Forms feature of New NLS

(J23637) 17-JUL-74 14:37; Title: Author(s): Elizabeth J. (Jake) Feinler/JAKE; Distribution: /SRI-ARC; Sub-Collections: SRI-ARC; Clerk: JAKE;
Origin: <FEINLER>SCENARIOS.NLS;2, 17-JUL-74 14:34 JAKE ;

Other uses for Process Command Forms feature of New NLS

In using the new Process Command Forms feature of newnls, it occurred to me that this could be used very effectively for a teaching tool to be used for building scenarios and for online training. I envision a short written document giving the pupil a little background on an NLS feature, such as journal for instance. This could be constructed fairly easily from HELP. Then he would be told (or it could happen automatically with editor's hidden links) how to start a process that would run through the various commands needed to demonstrate the correct way to submit a journal article (or whatever). Simple scenarios could be expanded into more complex ones in a logical easy-step fashion. 1

The whole arrangement is ideally suited to this because the pupil can carry the piece of paper with him and use it again for reference. There is also the possibility of having a SCENARIO feature in the HELP system. And so on. Anyway Dirk, Kirk, Jeanne and Dick can catch my enthusiam - I leave it to you to figure out the details. Just thought I would throw it out for consideration. 2

Help BUG--in DNLS, in work

(J23638) 18-JUL-74 14:47;;;;; ; Title: Author(s): Jeanne M. Beck/JMB; Distribution: /FDBK([ACTION]) HGL([ACTION]) EKM([ACTION]) ; Sub-Collections: SRI-ARC; Clerk: JMB;

JMB 18-JUL-74 14:47 23638

Help BUG--in DNLS, in work

Help gives you a message to use the MORE command when there's nothing else to menu, i,e, display says "empty" if you do say "More", Try Showing: Useroptions Show All, What is the reason for this? 1

Phone log, 18 Jul 74: Rick Witwer re forthcoming NIOSH visitors

(J23639) 18-JUL-74 16:08;;;;; ; Title: Author(s): Douglas C. Engelbart/DCE; Distribution: /JCN([ACTION] Jim: Let me know if you'd rather these visits be handled in some special way.) ; Sub-Collections: SRI-ARC; Clerk: DCE;

Phone log, 18 Jul 74: Rick Witwer re forthcoming NIOSH visitors

Rick Witwer, SRI MSD, called. Among the new hires they've made for their new NIOSH contract is a Dr. Fred Clayton, who has recently been a branch chief in the National Library of Medicine, apparently specializing in toxicology material. Clayton will be visiting SRI-Menlo 29 to 31 July, and Rick would like to have him get acqainted with ARC -- he says that Clayton would likely have quite a bit of contact with the NIOSH people who might be subscribing to our Utility. We tentatively are set up for Monday 29 July at 1500. 1

Also, Witwer mentioned that Dr. Vernon Rose, Director, Office of Research and Standards Development, NIOSH, might visit SRI in the first week of August, and if so Rick would also like to arrange for Rose to have another visit at ARC (he visited with Witwer on 14 Mar 74, see -- 22651,). Witwer judged that Rose would be ready for specific discussion about his shop's potential use of the Utility. 2

Since JCN plans to be on vactation between 29 July and 9 August, I left it with Rick that he would contact me for specific visiting arrangements. 3

For other relevant contact reports, see (22664,) regarding a discussion with Margaret Whittlesey about the preparation of the proposal for this NIOSH contract. Also, early in June, Norton and Meyer had an accidental meeting with a NIOSH site-inspection team at SRI-DC -- an impromptu demonstration ensued (GJOURNAL, 23213,). And several weeks ago I had a phone talk with Witwer about the general plans for getting the new SRI/NIOSH project launched, and about the prospects later in the summer to begin discussing AKW Utility subscription for either or both the SRI project and the NIOSH group at NIH (GJOURNAL, 23466,). 4

New version of NLS brought up 18-JUL-74

(J23640) 18-JUL-74 17:19;;;;; ; Title: Author(s): Kirk E. Kelley/KIRK; Distribution: /JCN([ACTION] SRL IS still doing FEEDBACK??) SRI-ARC([INFO-ONLY]) ; Sub-Collections: SRI-ARC; Clerk: KIRK;

New version of NLS brought up 18-JUL-74

A new version of NLS has been brought up at ARC containing all of those bug fixes Susan Lee has been promising for so long. It also contains various changes. 1) The Sendmail subsystem Initializes whenever you Goto or Execute it. You can no longer startup where you left off unless you are Quitting back into The sendmail subsystem. 2) The first time you update a file using the new system, the < USERNAME, FILENAME, >, DATE TIME IDENT;;;; part of the origin statement will appear two times seperated by four semicolons. You may edit your origin statement to Delete the second one that follows the four semicolons. After the first update, this anomaly will not occur again. please continue to send your feedback to FDBK via Sendmail and FEEDBACK via sndmessage. 1

Process Commands used as tutorials.

(J23641) 18-JUL-74 17:47;;;;; ; Title: Author(s): Kirk E. Kelley/KIRK; Distribution: /SRI-ARC([INFO-ONLY]) ; Sub-Collections: SRI-ARC; Clerk: KIRK;

Process Commands used as tutorials.

I have caught the enthusiasm from JAKE's (23637,), I think these would be useful activated by a link in the Help system. It would make it much more "interactive" ... an excellant idea that might solve our current problem with examples. 1

TABLE OF CONTENTS

INTRODUCTION

FTPFRK is an assembly-language program designed and implemented expressly to provide other Tenex programs with a convenient mechanism for manipulating files on other ARPANET hosts by means of the Network-standard File Transfer Protocol (FTP) (see RFC 542 -- 17759,).

FTPFRK is designed to run beneath an applications program in an inferior fork, and to communicate with its superior via the inter-fork protocol described in this document. FTPFRK is not a subsystem, and therefore has no user command language.

FTPFRK provides the applications program with convenient primitives for performing what are in many cases complex operations in (among others) the following areas:

1) Manipulation of distant files (e.g., deleting and renaming files).
2) Transfer of files between the local system and a distant system (e.g., retrieving, storing, and appending to distant files).
3) Manipulation and transfer of whole groups of Tenex files (e.g., retrieving all of the files named *.SAV;*).
4) Delivery of Network and local mail (e.g., formatting, delivering, and queuing mail files).
5) Interleaving operations at two or more hosts.

This document is a programmer's guide to FTPFRK and provides all of the information required to use its services. To obtain a copy of the source file, contact Jim White (JEW) at SRI-ARC?

INVOKING AN FTPFRK PRIMITIVE

FTPFRK will perform on behalf of its superior fork, any of a variety of atomic operations called "primitives". Each primitive has a name, takes zero or more arguments which provide the specifics of the operation to be performed, and returns an indication of its success or failure. Whenever a primitive fails, FTPFRK returns a diagnostic message suitable for presentation to a human user.

> Note: to locate the description of a particular primitive while reading this document on-line, the NLS user need only jump to the statement named by the primitive's op code. Thus, for example, JUMP NAME login (either typed or bugged) takes the user to the description of the LOGIN primitive.

The name of each primitive is a 1-5 character upper-case ASCII op code which the superior fork passes to FTPFRK left-adjusted (and padded on the right with zero bits) in FTPFRK's AC 0. A summary of these op codes is given in (opcodes:z). The arguments required by a primitive are also ASCII character strings, which, like the op code, are (in the simplest case) stored by the superior fork in FTPFRK's ACs. The rules for argument transmission are detailed in the next section of this document.

Once the op code and arguments have been stored in FTPFRK's ACs, FTPFRK's superior fork starts FTPFRK at its entry point. FTPFRK then examines the op code and arguments and executes the primitive. Upon completion of the primitive, FTPFRK halts, leaves information about the outcome of the primitive in its ACs, and waits for its superior to interpret the outcome and restart FTPFRK for the next primitive.

> After the first primitive, the superior may, if it chooses, simply resume FTPFRK with the then-current PC for all subsequent primitives.

When FTPFRK halts, its AC 0 contains an outcome code. A non-negative code indicates that the primitive was executed successfully. A code of -1 indicates that the primitive failed, in which case one of the diagnostics listed in the appendix "Error Messages" (see -- diags) is supplied as an ASCIZ string (i.e., an ASCII string terminated with a NUL), left-adjusted in FTPFRK's ACs 1-15.

SUPPLYING ARGUMENTS TO PRIMITIVES

Arguments to FTPFRK primitives are supplied by the superior fork in a uniform manner, regardless of the primitive selected or the type of argument supplied. Each argument has a "value" upon which the primitive ultimately acts. The superior fork transmits to FTPFRK not the argument's value, but rather a "handle" to the argument. FTPFRK then derives the argument's value from the handle on the basis of the latter's "appearance".

THE VALUE OF AN ARGUMENT

The value of an argument is an ASCII character string of (in principle) arbitrary length and content. In practice, both length and content are limited first by the superior fork's choice of handle and finally by the primitive itself (e.g., the value of an argument offered as a filename must obey the appropriate filename syntax conventions).

THE ARGUMENT HANDLE

An argument handle is an ASCIZ string whose maximum length is dependent upon the handle's "location", and whose content (or appearance) specifies the algorithm to be employed in deriving the argument's value.

THE HANDLE'S LOCATION

Argument handles may be stored by FTPFRK's superior fork in either its or its inferior's address space, but the handles to all arguments for a single primitive must have the same location (i.e., must all be in the same address space). In preparing for each primitive, the superior stores in FTPFRK's AC 1, the number of arguments supplied (Bits 32-35) and their location (Bit 0).

Note: Bits 1-31 of AC 1 are currently unused and should be zero.

LOCAL ARGUMENTS

An argument is said to be "local" (to FTPFRK) if its handle is passed to FTPFRK in its address space (specifically, in its ACs). If bit 0 of AC 1 is set to zero, then 0-1 argument handles are assumed by FTPFRK to have been stored left-adjusted in its ACs 2-15. The handle for a local argument thus has a maximum length of 14*5-1, or 69 characters (excluding the terminating NUL).

REMOTE ARGUMENTS

An argument is said to be "remote" (to FTPFRK again) if FTPFRK must retrieve its handle from the superior's address space. If bit 0 of AC 1 is set to one, then 0-14 argument handles are assumed to await retrieval by FTPFRK in the superior's address space. The required number of byte pointers are assumed left adjusted in FTPFRK's ACs 2-15 (a byte pointer whose left half contains -1 is given the usual, Tenex interpretation, i.e., 440700). The handle for a remote argument has a somewhat arbitrary %remlen -character maximum length imposed upon it.

Note: certain system parameters (e.g., timeout periods, maximum lengths) whose values are subject to change are denoted in the body of this document by a symbolic name whose first character is a percent sign ('%). The current value of any such parameter can be found either by consulting the appendix entitled "Current Values of System Parameters", or while on-line, by jumping to the statement of the same name. Thus JUMP NAME remlen (either typed or bugged) takes the user to the value of %remlen.

Before the superior can transmit remote arguments, it must grant FTPFRK access to its address space (i.e., it must apply the EPCAP JSYS to FTPFRK's fork with Bit 9 of AC 3 set to one).

THE HANDLE'S APPEARANCE

The handle's appearance (i.e., the syntax of the ASCIZ string which constitutes the handle) governs its interpretation and thus the manner in which the argument's value is derived. Every argument handle falls into one of two major classes, depending upon whether its first character is %argesc or not.

LITERAL ARGUMENTS

An argument whose handle does NOT have %argesc as its first character is called a "literal" argument, and its value and handle (dropping the terminating NUL) are one in the same (the simplest of derivations).

The value of a literal argument is (obviously) subject to the length restrictions imposed by FTPFRK upon argument handles (69 characters for local %remlen for remote arguments).

Any valid Tenex filename, for example, can be transmitted as a local literal argument, but many other types of arguments cannot. Most argument types can be transmitted as remote literal arguments, but a few cannot (e.g., the text of lengthy pieces of mail).

To overcome such length limitations, a second class of arguments, described below, is defined.

SYMBOLIC ARGUMENTS

An argument whose handle HAS %argesc as its first character is called a "symbolic" argument. The value of a symbolic argument is derived from the handle on the basis of the SECOND character of the handle.

The handle of a symbolic argument has the general form:

```
%argesc <operator> <operand> NUL
```

OPERATOR is a single character and OPERAND is a (possibly null) character string. The value of the argument is derived by FTPFRK from OPERATOR and OPERAND. The following types of symbolic arguments are defined:

FILE ARGUMENTS (OPERATOR = %filarg)

OPERAND is taken to be the name of a local sequential text file which contains the value of the argument. The entire contents of the file are taken as the argument value, which can therefore be of effectively arbitrary size.

COMPLEX ARGUMENTS (OPERATOR = %cpxarg)

OPERAND is taken to be the name of a local sequential text file containing the values of the next zero or more arguments for the primitive. A complex argument is thus not a single argument at all, but rather a whole set of arguments, each of which is treated by FTPFRK as if it had been separately transmitted.

The argument file contains the values of an arbitrary number of arguments, each preceeded and followed by instances of an arbitrary delimiter. Each delimiter-value-delimiter combination may be preceeded by an arbitrary number of formatting characters (specifically SP, TAB, CR, LF, and EOL). The first non-formatting character is taken as the delimiter for the argument, and the value of the argument is taken to be the string of ASCII characters between the delimiter and its next occurrence.

The special delimiter %unmdel may be used for the last argument in the file, in which case the terminating delimiter is neither expected nor sought; the rest of the file is taken as the argument's value. The last argument in the file may therefore contain all 128 ASCII characters, and need not incur the overhead of a delimiter search.

DEGENERATE SYMBOLIC ARGUMENTS (OPERATOR = %argesc)

The value of the argument is obtained by concatenating OPERATOR with OPERAND.

An argument whose value begins with %argesc and which could otherwise be passed as a literal argument, must be passed instead as a degenerate symbolic argument to avoid misinterpretation by FTPFRK.

This is simply an example of the familiar trick of doubling an escape character to get it through.

NOP ARGUMENTS (OPERATOR = %noparg)

A nop argument is effectively a "non-argument": it does not count toward the required number of arguments for the primitive. The OPERAND is ignored and hence might just as well be null.

THE NOP PRIMITIVE

A special NOP primitive is defined whose only function is to pre-supply arguments for a subsequent "target" primitive. Any primitive may be preceeded by zero or more occurences of the NOP primitive, each of which supplies the next zero or more arguments for the target primitive. If any arguments remain to be supplied when the target primitive is invoked, these final arguments must be supplied with it.

NOP can be used to beat the one-argument limitation for local arguments or to effectively mix arguments of different locations in a single primitive.

GROUP PRIMITIVES

TENEX GROUP DESCRIPTORS

(grpdesc)Tenex permits both individual files and whole groups of files to be designated with the same, concise syntax. Whenever a filename field has the value '* (an asterisk), the implication is that a group of files, rather than a single file is denoted. A filename with an asterisk in one or more of its fields is called a "file group descriptor", and, when used in reference to a group of existing files, stands for all those files whose filenames match the descriptor in all fields except those marked with '*.

Conceptually, then, a group descriptor specifies a template or mask through which candidate filenames are viewed. The template is opaque wherever a field is specified as '*, and tranparent elsewhere. Whenever the group descriptor and a candidate filename are identical when viewed through the mask, the candidate is included in the group.

FTPFRK GROUP PRIMITIVES

FTPFRK provides a set of primitives for manipulatng such groups of local or distant files. Of course, in the case of distant files, their use is appropriate if and only if the distant host is a Tenex system.

SPECIFYING THE FILE GROUP

In any such group primitive, the file group to be manipulated (e.g., deleted, renamed, transferred between hosts) -- denoted by "group", "distant.group", "source.group", and so forth in the descriptions that appear later on in this document -- can be specified by the user in either or two ways:

1) The value of the argument which denotes the group can be the name of a local sequential text file, called a "group file", containing a list of the filenames which comprise the group to be manipulated.

(grpsyntax)The group file contains zero or more filenames, each terminated by EOL or the character sequence CR LF, and each of the form:

[dev:] [<dir>] name [.ext] [;ver] [;T] [;P protection] [;A account]

Note: brackets surround optional fields.

Although temporary, protection, and account fields may be present, they are ignored by the group primitive. The end of the filename list is signified either by the end of the file or by the string "? Not found.".

The formatting characters SP, TAB, CR, LF, and FF may be used for that purpose anywhere within the file, and will be ignored by the group primitive. EOL and the character sequence CR LF will be similarly ignored whenever they appear as formatting characters (as opposed to filename delimiters). In addition, any characters appearing between the characters %grpcms and %grpcme will be treated as comments, and, along with the delimiters, ignored.

The literal escape character %litesc, when prefixed to what would otherwise be a field delimiter in one of the filenames, a comment delimiter, or a formatting character, causes it to be interpreted as simple filename text.

Because of its flexible format, a group file may be any one of the following:

a) A file created via the LDIR primitive, which generates a local directory listing.
b) A file created via the DDIR primitive, which generates a distant directory listing.
c) One of the "outcome files" from a previous group operation (see OUTCF).
d) A file created via the LTRNS primitive, which creates a second group file by "translating" each filename in a first.
e) A file created by a program other than FTPFRK (e.g., TECO).

2) The value of the argument which denotes the group can be simply a group descriptor (distinguished from case 1 above by the presence of asterisks(s)).

In this case, FTPFRK itself generates the necessary grou= file by means of either the LDIR or DDIR primitive, whichever is appropriate; uses it to execute the primitive; and deletes it (via the LELM primitive) when the group primitive is complete.

DERIVING A SECOND FILENAME FROM THE FIRST

Most group primitives require a second filename to pair with each filename in the group. For example, the GDREN primitive, which renames each distant file in a specified group, requires both current and proposed filenames for each file to be renamed. Similarly, any group transfer primitive (e.g., retrieve or store) requires both source and destination filenames. GDDEL (the group delete primitive) is the most obvious counter example; it requires only a single filename as an argument.

(masking)In those cases where a filename pair is required for each group member, the second filename is generated by FTPFRK by passing the first, obtained from the group file, through a group descriptor (a mask) which the superior fork provides as an additional argument to the group primitive.

The generated filename is identical to the mask, except in those fields that are asterisks, where the corresponding field in the first filename is used instead. Thus the group containing the three filenames:

```
FTPFRK.FAI
FTPFRK.REL
FTPFRK.SAV
```

when passed through the mask "JIMSPGM.*" generates:

```
JIMSPGM.FAI
JIMSPGM.REL
JIMSPGM.SAV
```

Both the source filename and the mask are of the form:

[dev:] [<dir>] name [.ext] [;ver] [;T] [;P protection] [;A account]

If corresponding fields of the source filename, the mask, and the generated filename be denoted by f(i), m(i), and g(i) respectively, then the following rules govern the process by which one filename is derived from another via the mask:

1) If m(i) is '*, then g(i) = f(i), even if f(i) is null.
2) Otherwise, g(i) = m(i), even if m(i) is null.
3) Temporary, protection, and account fields, even if present in f and/or m, will be absent in g.

DETERMINING THE OUTCOME OF GROUP PRIMITIVES

The implementation of each group primitive by FTPFRK involves the invokation of a corresponding single-file primitive for each file in the group. Each such "sub-primitive" succeeds or fails independently for each group member. Whenever the sub-primitive fails, a diagnostic explaining the nature of the failure is available and may be of use to FTPFRK's superior.

Furthermore, the superior may wish to apply subsequent group primitives only to those group members which are successfully processed by the current group primitive, and/or apply special group primitives to those group members which fail the current group primitive. To do this, the superior must be able to identify those group members for which the primitive succeeded, and those for which it failed.

To make options like these available to the programmer, FTPFRK generates two local sequential text files, called "outcome files", as the result of each group operation. The first contains the names of the (zero or more) group members which were successfully processed, and the second the names of those for which the primitive failed, along with the text of the diagnostic explaining the failure.

(outcexam)The failure outcome file generated by a group primitive looks like the following:

```
% DRTR (test,*.transferred;*,TENEX) started MON 3 JUN 74
1647:23 %
   <WHITE>ANDMSG.NLS;2 % No distant system is open. %
   <WHITE>FINREPORT.NLS;4 % " %
   <WHITE>FRKDOC.NLS;25 % " %
   <WHITE>FRKDOC.NLS;24 % " %
   <WHITE>GROUPRESPONSES.NLS;2 % " %
   <WHITE>JEW.NLS;218 % " %
   <WHITE>JSUBSERV.NLS;5 % " %
   <WHITE>MHJSPAPER.NLS;33 % " %
   <WHITE>MHJSTERSE.NLS;1 % " %
   <WHITE>MHJSVERBOSE.NLS;2 % " %
   <WHITE>NSWPROP.NLS;21 % " %
   <WHITE>QMR.NLS;3 % " %
% DRTR comple¯ed MON 3 JUN 74 1647:25 %
```

A simple quote (") as the diagnostic (as in the example above) implies that the group member failed for the same reason as the previous group member.

Each failure outcome file contains the name (i.e., the op code) of the sub-primitive applied to each group member, the arguments of the group primitive, the start and completion dates and times, the name of each group member which failed, and a diagnostic for each. Everything but the filenames themselves are distinguished as comments by their placement between %grpcms and %grpcme.

Success outcome files are identical to failures files, except that the diagnostics are absent.

Outcome files conform to the syntax requirements (see --grpsyntax) of group files themselves. Therefore, an OUTCOME FILE generated by one group primitive can be employed as the GROUP FILE for a subsequent group primitive. This is a very important and useful property of outcome files.

It is also possible, using the OUTCF primitive, to cause FTPFRK to employ a single outcome file for a series of group primitives, with the result of each successive group primitive appended to the results of previous ones. Using this feature, it is extremely easy for the programmer to generate a list of those group operations which failed during the session (or series of sessions); with sufficient information to determine each operation that failed, and the date, time, and cause of the failure; and suitable for output on a line printer.

It is also possible to disable the generation of one or both types of outcome files entirely.

FILE TRANSFER MODES

(xfermode)Each file transfer primitive requires as one of its arguments a "transfer mode" that governs the format in which the file is transmitted through the Network. The following transfer modes are defined:

ASCII

The ascii transfer mode provides a means for transferring sequential text files between unlike hosts (e.g., between the local Tenex system and an IBM 360). Files are converted for transmission to a Network-standard intermediate representation by the source system, and then converted to the distant system's own internal format before storage.

Ascii transfers are also valid between like hosts (i.e., if the distant host is also a Tenex system) and will produce the correct results (provided the source file is a sequential text file), but in such cases the "tenex" transfer mode described below is much more efficient.

TENEX

A tenex transfer, as its name suggests, is in general appropriate only when the distant host is a Tenex system. Any file can be transmitted in this mode, and for all but holey files, it is the most efficient one. Files are shipped to and stored in the destination system in their internal format.

Tenex transfers are in general valid even when the distant host is NOT a Tenex system, although a particular distant host may choose to reject them, but the transmitted files will probably be interpretable within the distant system only by specially written programs which understand the internal format of the original source file, since no conversion will have been performed.

If the distant system is being used only as an archiving facility (i.e., if no distant program need ever manipulate the files stored there), then the Tenex transfer mode (despite its in-this-case misleading name) is probably most appropriate.

COMPRESSED

A compressed transfer is valid only if the distant system is SRI-ARC or OFFICE-1. Files are compressed in the source system (i.e., SSAVEd) before transmission through the Network, and then restored to their original form (via GET) before storage in the destination system. Any file can be transmitted in this mode, and for holey files, it is usually the most efficient one.

Since the File Transfer Protocol makes no provision for the transmission of structured (i.e., non-sequential) files, distant Tenex systems, for tenex transfers, convert holey files to sequential files by substituting a page of zeros for each missing page in the source file's map. This procedure, although it works, leads to excess transmission time, and excess storage requirements in the destination system.

A compressed transfer, on the other hand, produces a destination file identical to the source file, which therefore requires no unnecessary disk space for storage. It also minimizes Network transmission time. Compressed tranfers gain these advantages, however, at the cost of added processing time in both the source and destination hosts and are hopefully only a short-term solution to the problem.

Compressed transfers can also be applied to files that are NOT holey, but the tenex transfer mode described above is more efficient in such cases.

Compressed transfers are provided primarily for use in transferring NLS files and their partial copies. Therefore compressed transfers have the following additional properties:

1) the byte size and count of the destination file are always set to zero (as they are in all NLS files)

2) if the source filename's extension is either "NLS" or "PC", and some additional conditions upon the file's contents are met, Tenex directory numbers stored in the header of the NLS file are converted to strings, since the same directory in the destination host may be assigned a different directory number there.

APPROPRIATE

Specifiying an approprate transfer effectively gives FTPFRK freedom to employ whatever transfer mode -- either ascii, tenex, or compressed -- it judges to be most appropriate.

FTPFRK currently employs the following selection algorithm. If the distant host is not a Tenex system, an ascii transfer is performed. Otherwise, a tenex transfer is performed, unless the source file's extension is NLS or PC, in which case a compressed transfer is selected.

By selecting the appropriate transfer mode, the superior fork relieves itself of all responsibility for choosing the transfer mode. It need know neither the type of file being transmitted, nor the nature of the destination host. Appropriate transfers also make possible the efficient transfer of both NLS and non-NLS files in a single group operation, with FTPFRK selecting the transfer mode on a per-file basis.

Whenever a transfer mode is required as an argument to a primitive, one of the character strings "ASCII", "TENEX", "COMPRESSED", or "APPROPRIATE" should be supplied. Like all other arguments whose values are keywords, the keyword must be in u=per-case.

DESCRIPTION OF PRIMITIVES

All of the primitives currently offered by FTPFRK are described in the followng sections. The primitives have been partitioned, for purposes of documentation, into the following categories:

Specifying FTPFRK Parameters
Connecting to a Distant System
Manipulating Distant Files
Manipulating Groups of Distant Files
Manipulating Local Files
Manipulating Groups of Local Files
Local File Utilities
Local File Group Utilities
Transferring Files Within the Local System
Transferring Groups of Files Within the Local System
Transferring Files Between the Local and Distant Systems
Transferring Groups of Files Between the Local and Distant Systems
Transferring Files Between Two Distant Systems
Transferring Groups of Files Between Two Distant Systems
Manipulating Directories
Obtaining Network-Related Information
Local and Distant Mail
Manipulating State Records

The description of each primitive has the following format:

Function of primitive
op code (argument 1, argument 2, ... argument n)

A detailed description of the primitive's function and of the arguments it requires.

SPECIFYING FTPFRK PARAMETERS

INTRODUCTION

The primitives described in this section control FTPFRK as a whole, by changing FTPFRK's state and by manipulating a variety of parameter settings.

PRIMITIVES

(BEGIN)Initialize for FTPFRK session
BEGIN ()

This must be the very first primitive invoked by FTPFRK's superior after creation of the fork. BEGIN initializes all the necessary program variables in preparation for the session.

(END)Terminate FTPFRK session
END ()

This should be the very last primitive invoked by FTPFRK's superior before killing the fork. The primitive releases all resources acquired by FTPFRK during the session.

END immediately followed by BEGIN effectively resets FTPFRK to its original state, as does the SYSRS primitive described below.

(SYSRS)Reset FTPFRK session
SYSRS ()

This primitive resets FTPFRK, releasing all resources it acquired during the session, and returns it to its initial state. The effect of SYSRS is identical to that obtained by executing the primitives BEGIN and END in succession.

(SOCK)Set contact socket for distant systems
SOCK (distant.socket)

This primitive specifies the socket number on which FTPFRK, in all subsequent OPEN primitives, is to expect the distant systems' FTP server processes to be listening. The File Transfer Protocol specifies a Network-wide standard for this number, which FTPFRK takes as the default, but non-production implementations are often offered on other sockets. The SOCK primitive permits the superior to access such experimental implementations.

DISTANT.SOCKET must be a decimal integer in the range /0, 2**32-1/. Since the Initial Connection Protocol (ICP) (see -- 7101,) requires that the socket number be odd, FTPFRK will force the low-order bit of the socket number's internal, binary representation to one.

If DISTANT.SOCKET has the value "" (i.e., if it is null), the contact socket number is reset to its Network-standard, FTPFRK default value.

(OUTCF)Specify a local file to receive group operation outcome information
OUTCF (type.of.outcome, filename, append.or.not)

This primitive makes the local sequential text file FILENAME available for use by FTPFRK for the recording of information about the outcome of subsequent group primitives.

The class of outcome information to be posted in the outcome file by FTPFRK is specified by TYPE.OF.OUTCOME, which may have either of the following values:

"SUCCESS"

The specified file is to receive the filenames of all those group members to which the group sub-primitive is successfully applied.

"FAILURE"

The specified file is to receive the filenames of all those group members to which the group sub-primitive is UNsuccessfully applied.

The results of each successive group operation will be appended to the outcome file if APPEND.OR.NOT is "YES", or, if APPEND.OR.NOT is "NO", either written as the next higher version of the file (if no version number is specified in FILENAME) or written over the results of the previous group operation in a single file.

A single invokation of the OUTCF primitive specifies the outcome file for one of the two classes of outcomes, and overrides the previous specification for that same class. Two outcome files, one for successes and one for failures, are in effect simultaneously.

If FILENAME is "", the specified class of outcome information for subsequent group primitives is discarded and thus goes unreported.

(DEBUG)Enable or disable the typeout of debug information
DEBUG (on.or.off)

This primitives enables (if ON.OR.OFF is "ON") or disables (if ON.OR.OFF is "OFF") the output of certain debug information, including all protocol interchanges with the distant system, to FTPFRK's primary output device.

(NOP)Pre-supply FTPFRK argument(s)
NOP (argument.1, argument.2, ... argument.n)

This primitive supplies zero or more arguments (ARGUMENT.1, ARGUMENT.2, etc.) for a subsequent target primitive. Aside from this function, the primitive is a nop.

Any primitive may be preceeded by zero or more occurences of the NOP primitive, each of which supplies the next zero or more arguments for the target primitive. If any arguments remain to be supplied when the target primitive is invoked, these final arguments must be supplied with it.

CONNECTING TO A DISTANT SYSTEM

INTRODUCTION

The primitives described in this section provide access to distant file systems, and include primitives for opening and closing a logical connection to the distant system, and for logging in.

PRIMITIVES

(OPEN)Open a distant file system
OPEN (host)

This primitive opens a logical connection to the file system at host HOST. The format of HOST is described in connection with the VHOST prmitive (see -- hostsyntax). OPEN is illegal if a distant system is already open.

(LOGIN)Login at the distant system
LOGIN (user, password, account)

This primitive establishes the user's identity at the distant system for both billing and file access purposes (in general), and specifies the default working directory (which may be overridden with either DFDIR or another LOGIN).

If the distant host is a Tenex system, USER must, of course, be a distant directory name, except that often the USER "ANONYMOUS" (accompanied by any PASSWORD) is also recognized.

(CLOSE)Close the distant file system
CLOSE ()

This primitive breaks the logical connection to a distant system established by a previous OPEN (or ICP) primitive. If no distant system is open, the primitive is a NOP.

(ICP)Connect to an arbitrary server process
ICP (host, socket, byte.size)

This primitive establishes a logical connection to an abitrary server process at host HOST using the Network's Initial Connection Protocol (see -- 7101,). The format of HOST is described in (hostsyntax).

SOCKET is the server process' primary contact socket, specified in decimal, and must be in the range [0,2**32-1], in accordance with Host-Host Protocol (see -- 8246,). BYTE.SIZE is the byte size which is to characterize each of the two simplex Network connections established as a result of the primitive, and must be in the range [1,255], again in accordance with Host-Host Protocol.

MANIPULATING DISTANT FILES

INTRODUCTION

The primitives described in this section manipulate files residing in the distant file system most recently opened with the OPEN primitive. Most, if not all distant systems require that the LOGIN primitive have been issued.

PRIMITIVES

(DREN)Rename a distant file
DREN (current.name, proposed.name)

This primitive renames the distant file CURRENT.NAME to be PROPOSED.NAME.

(DDEL)Delete a distant file
DDEL (filename)

This primitive deletes the distant file FILENAME. If the distant host is a Tenex system, the file will be marked for deletion, but not actually removed from the directory nor its disk space released. It is not possible to undelete a distant file nor to explicitly expunge a distant directory via FTPFRK.

MANIPULATING GROUPS OF DISTANT FILES

INTRODUCTION

The primitives described in this section manipulate entire groups of files residing in the distant file system most recently opened with the OPEN primitive. The distant host must be a Tenex system for any of these primitives to be successfully applied, and most such systems require that the LOGIN primitive have been issued.

The outcome of each primitive is reported on a per-file basis under control of the OUTCF primitive. The group primitive itself will fail only if the basic group primitive mechanism fails for some reason.

PRIMITIVES

(GDREN)Rename a distant file group
GDREN (current.group, proposed.mask)

This primitive renames each distant file "f" in CURRENT.GROUP to have the name generated by passing "f" through PROPOSED.MASK.

(GDDEL)Delete a distant file group
GDDEL (group)

This primitive deletes each distant file in GROUP. If the distant host is a Tenex system, each file will be marked for deletion, but not actually removed from the directory nor its disk space released. It is not possible to undelete a distant file group nor to explicitly expunge a distant directory via FTPFRK.

MANIPULATING LOCAL FILES

INTRODUCTION

The primitives described in this section manipulate files residing in the local file system; no distant system need be open.

PRIMITIVES

(LREN)Rename a local file
LREN (current.name, proposed.name)

This primitive renames the local file CURRENT.NAME to be PROPOSED.NAME.

(LDEL)Delete a local file
=ldel (filename)

This primitive deletes the local file FILENAME. The file will be marked for deletion, but not actually removed from the directory nor its disk space released. It is not possible to undelete a local file nor to explicitly expunge a local directory via FTPFRK.

MANIPULATING GROUPS OF LOCAL FILES

INTRODUCTION

The primitives described in this section manipulate entire groups of files residing in the local file system; no distant system need be open.

The outcome of each primitive is reported on a per-file basis under control of the OUTCF primitive. The group primitive itself will fail only if the basic group primitive mechanism fails for some reason.

PRIMITIVES

(GLREN)Rename a local file group
GLREN (current.group, proposed.mask)

This primitive renames each local file "f" in CURRENT.GROUP to have the name generated by passing "f" through PROPOSED.MASK.

(GLDEL)Delete a local file group
GLDEL (group)

This primitive deletes each local file in GROUP. Each file will be marked for deletion, but not actually removed from the directory nor its disk space released. It is not possible to undelete a local file group nor to explicitly expunge a local directory via FTPFRK.

LOCAL FILE UTILITIES

INTRODUCTION

The primitives described in this section manipulate files residing in the local file system, performing a variety of utility functions; no distant system need be open.

PRIMITIVES

(LELM)Eliminate a local file
LELM (filename)

This primitive marks the local file FILENAME for deletion and releases each disk page in its map. The file will not be actually removed from the local directory, nor the local directory expunged. The file's byte size and count remain unchanged.

(LSUB)Create a copy of a local file with EOL replaced by CRLF
LSUB (source.filename, destination.filename)

This primitive creates a second local file DESTINATION.FILENAME identical to the local file SOURCE.FILENAME, except that wherever EOL occurs in the source file, CR LF is substituted in the destination. The source file is treated as a simple sequential text file, and the destination file that is created has the same characteristics.

(LPOR)Create a portrayed version of a local file
LPOR (source.filename, destination.filename)

This primitive creates a second local file DESTINATION.FILENAME identical to the local file SOURCE.FILENAME, except that wherever an "invisible" character occurs in the source file, a string of visible characters -- the "name" of the invisible character -- is substituted in the destination. The source file is treated as a simple sequential text file, and the destination file that is created has the same characteristics.

LPOR gives the indicated names to the following invisible characters:

```
(chareps)BEL (↑G)  -- "<BEL>"
CR  (↑M)   -- "<CR>"
CR LF      -- "<CRLF>" CR LF
  (in that sequence)
DEL (177)  -- "<DEL>"
EOL (37)   -- "<EOL>" CR LF
ESC (33)   -- "<ESC>"
FF  (14)   -- "<FF>"
FS  (34)   -- "<FS>"
GS  (35)   -- "<GS>"
LF  (↑J)   -- "<LF>"
NUL (00)   -- "<NUL>"
RS  (36)   -- "<RS>"
TAB (↑I)   -- "<TAB>"
↑A - ↑Z    -- "<↑A>" "<↑B>" etc.
  (except as noted above)
```

All other characters are copied to the destination file unchanged.

LOCAL FILE GROUP UTILITIES

INTRODUCTION

The primitives described in this section manipulate entire groups of files residing in the local file system, performing a variety of utility functions; no distant system need be open.

The outcome of each primitive is reported on a per-file basis under control of the OUTCF primitive. The group primitive itself will fail only if the basic group primitive mechanism fails for some reason.

PRIMITIVES

(GLELM)Eliminate a local file group
GLELM (group)

This primitive marks each local file in GROUP for deletion and releases each disk page in its map. The files will not be actually removed from the local directory, nor the local directory expunged. The byte size and count of each file remain unchanged.

(GLSUB)Create a copy of a local file group in which EOL is replaced by CRLF
GLSUB (source.group, destination.mask)

This primitive creates for each local file "f" in SOURCE.GROUP, a second local file whose name is generated by passing "f" through DESTINATION.MASK, identical to it, except that wherever EOL occurs in the source file, CR LF is substituted in the destination. Each source file is treated as a simple sequential text file, and each destination file that is created has the same characteristics.

(GLPOR)Create a portrayed version of a local file group
GLPOR (source.group, destination.mask)

This primitive creates for each local file "f" in SOURCE.GROUP, a second local file whose name is generated by passing "f" through DESTINATION.MASK, identical to it, except that wherever an "invisible" character occurs in the source file, a string of visible characters -- the "name" of the invisible character -- is substituted in the destination. Each source file is treated as a simple sequential text file, and each destination file that is created has the same characteristics. The names given to the various invisible characters are listed in (chareps .d:z).

TRANSFERRING FILES WITHIN THE LOCAL SYSTEM

INTRODUCTION

The primitives described in this section transfer copies of local files to other locations within the local system; no distant system need be open.

PRIMITIVES

(LCPY)Replicate a local file
LCPY (source.filename, destination.filename)

This primitive creates a second local file DESTINATION.FILENAME which is identical to the local file SOURCE.FILENAME in content, byte size, and byte count.

(LAPP)Append a copy of one local file to another
LAPP (source.filename, destination.filename)

This primitive appends a copy of the local file SOURCE.FILENAME to the local file DESTINATION.FILENAME. Both files are treated as simple sequential files, and the byte size of the source is assumed to be the same as that of the destination.

TRANSFERRING GROUPS OF FILES WITHIN THE LOCAL SYSTEM

INTRODUCTION

The primitives described in this section transfer copies of entire groups of local files to other locations within the local system; no distant system need be open.

The outcome of each primitive is reported on a per-file basis under control of the OUTCF primitive. The group primitive itself will fail only if the basic group primitive mechanism fails for some reason.

PRIMITIVES

(GLCPY)Replicate a local file group
GLCPY (source.group, destination.mask)

This primitive creates for each local file "f" in SOURCE.GROUP, a second local file whose name is generated by passing "f" through DESTINATION.MASK, identical to the source file in content, byte size, and byte count.

(GLAPS)Append a copy of a local file group to local files
GLAPS (source.group, destination.mask)

This primitive appends a copy of each local file "f" in SOURCE.GROUP to a second local file whose name is generated by passing "f" through DESTINATION.MASK. Source and destination files are treated as simple sequential files, and the byte size of the source is assumed to be the same as that of its destination.

Note: GLAPS and GLAPD (described below) are identical, except that in the former the destination filenames are derived from the source filenames, and in the latter the reverse it true, i.e., source filenames are derived from destination filenames.

(GLAPD)Append copies of local files to a local file group
GLAPD (source.mask, destination.group)

This primitive appends to each local file "f" in DESTINATION.GROUP, a copy of a second local file whose name is generated by passing "f" through SOURCE.MASK. Source and destination files are treated as simple sequential files, and the byte size of the source is assumed to be the same as that of its destination.

Note: GLAPD and GLAPS (described above) are identical, except that in the former the source filenames are derived from the destination filenames, and in the latter the reverse it true, i.e., destination filenames are derived from source filenames.

TRANSFERRING FILES BETWEEN THE LOCAL AND DISTANT SYSTEMS

INTRODUCTION

The primitives described in this section transfer a copy of a file between the local file system and the distant system most recently opened with the OPEN primitive. Most, if not all distant systems require that the LOGIN primitive have been issued.

Each primitive requires that the user specify a TRANSFER.MODE that governs the format in which the file is transmitted through the Network. TRANSFER.MODE may have any of the values listed in (xfermode .d):

PRIMITIVES

(DRTR)Create a local copy of a distant file
DRTR (distant.filename, local.filename, transfer.mode)

This primitive retrieves a copy of the distant file DISTANT.FILENAME and stores it in the local system as LOCAL.FILENAME. The transfer mode is specified by TRANSFER.MODE (see -- xfermode).

(DSTR)Create a distant copy of a local file
DSTR (distant.filename, local.filename, transfer.mode)

This primitive transmits a copy of the local file LOCAL.FILENAME to the distant system and stores it there as DISTANT.FILENAME. The transfer mode is specified by TRANSFER.MODE (see -- xfermode).

(DAPP)Append a copy of a local file to a distant file
DAPP (distant.filename, local.filename, transfer.mode)

This primitive transmits a copy of the local file LOCAL.FILENAME to the distant system and appends it to the distant file DISTANT.FILENAME. The transfer mode is specified by TRANSFER.MODE (see -- xfermode).

The local file is assumed to be a simple sequential file. If the distant host is a Tenex system, the distant file is assumed to have the same characteristics, and the byte size of the source file is assumed to be the same as that of the destination.

TRANSFERRING GROUPS OF FILES BETWEEN THE LOCAL AND DISTANT SYSTEMS

INTRODUCTION

The primitives described in this section transfer copies of entire groups of files between the local file system and the distant system most recently opened with the OPEN primitive. The distant host must be a Tenex system for any of these primitives to be successfully applied, and most such systems require that the LOGIN primitive have been issued.

Each primitive requires that the user specify a TRANSFER.MODE that governs the format in which the file is transmitted through the Network. TRANSFER.MODE may have any of the values listed in (xfermode .d):

The outcome of each primitive will be reported on a per-file basis under control of the OUTCF primitive. The group primitive itself will fail only if the basic group primitive mechanism fails for some reason.

PRIMITIVES

(GDRTR)Create a local copy of a distant file group
GDRTR (distant.group, local.mask, transfer.mode)

This primitive retrieves a copy of each distant file "f" in DISTANT.GROUP and stores it in the local system with the name generated by passing "f" through LOCAL.MASK. The transfer mode for the entire group of files is specified by TRANSFER.MODE (see -- xfermode).

(GDSTR)Create a distant copy of a local file group
GDSTR (distant.mask, local.group, transfer.mode)

This primitive transmits a copy of each local file "f" in LOCAL.GROUP to the distant system, and stores it with the the name generated by passing "f" through DISTANT.MASK. The transfer mode for the entire group of files is specified by TRANSFER.MODE (see -- xfermode).

(GDAPS)Append a copy of a local file group to distant files
GDAPS (distant.mask, local.group, transfer.mode)

This primitive transmits a copy of each local file "f" in LOCAL.GROUP to the distant system and appends it to the distant file whose name is generated by passing "f" through

DISTANT.MASK. The transfer mode for the entire group of files is specified by TRANSFER.MODE (see -- xfermode).

Each local file is assumed to be a simple sequential file. If the distant host is a Tenex system, each distant file is assumed to have the same characteristics, and the byte size of each source file is assumed to be the same as that of its destination.

Note: GDAPS and GDAPD (described below) are identical, except that in the former the destination filenames are derived from the source filenames, and in the latter the reverse it true, i.e., source filenames are derived from destination filenames.

(GDAPD)Append copies of local files to a distant file group
GDAPD (distant.group, local.mask, tranfer.mode)

This primitive transmits to the distant system and appends to each distant file "f" in DISTANT.GROUP, a copy of the local file whose name is generated by passing "f" through LOCAL.MASK. The transfer mode for the entire group of files is specified by TRANSFER.MODE (see -- xfermode).

Each local file is assumed to be a simple sequential file. If the distant host is a Tenex system, each distant file is assumed to have the same characteristics, and the byte size of each source file is assumed to be the same as that of its destination.

Note: GDAPD and GDAPS (described above) are identical, except that in the former the source filenames are derived from the destination filenames, and in the latter the reverse it true, i.e., destination filenames are derived from source filenames.

TRANSFERRING FILES BETWEEN TWO DISTANT SYSTEMS

INTRODUCTION

The primitives described in this section transfer a copy of a file between two distant systems, each previously opened with the OPEN primitive, and each described by a labeled state record (see "Manipulating State Records" -- 7T). Most, if not all distant systems require that the LOGIN primitive have been issued.

Neither of the state records upon which the primitive is to operate need be mounted when the primitive is invoked, and which state is left mounted upon completion of the primitive cannot be predicted.

Each primitive requires that the user specify a TRANSFER.MODE that governs the format in which the file is transmitted through the Network. TRANSFER.MODE may have any of the values listed in (xfermode .d) except "COMPRESSED".

PRIMITIVES

(NCPY)Negotiate the transfer of a copy of a file from one distant system to another
NCPY (source.state, source.filename, destination.state, destination.filename, transfer.mode)

This primitive transmits a copy of the distant file SOURCE.FILENAME at the distant system implied by SOURCE.STATE to the distant system implied by DESTINATION.STATE, where it is assigned the name DESTINATION.FILENAME. The transfer mode is specified by TRANSFER.MODE (see -- xfermode).

(NAPP)Negotiate the appending of a copy of a file at one distant system to a file at another distant system
NAPP (source.state, source.filename, destination.state, destination.filename, transfer.mode)

This primitive transmits a copy of the distant file SOURCE.FILENAME at the distant system implied by SOURCE.STATE to the distant system implied by DESTINATION.STATE, where it appends it to the distant file DESTINATION.FILENAME. The transfer mode is specified by TRANSFER.MODE (see -- xfermode).

If the source host is a Tenex system, the source file is assumed to be a simple sequential file. If the destination host is a Tenex system, the destination file is assumed to have the same characteristics, and the byte size of the source file is assumed to be the same as that of the destination file.

TRANSFERRING GROUPS OF FILES BETWEEN TWO DISTANT SYSTEMS

INTRODUCTION

The primitives described in this section transfer copies of entire groups of files between two distant systems, each previously opened with the OPEN primitive, and each described by a labeled state record. Both distant hosts must be Tenex systems for any of these primitives to be successfully applied, and most such systems require that the LOGIN primitive have been issued.

Neither of the state records upon which the primitive is to operate need be mounted when the primitive is invoked, and which state is left mounted upon completion of the primitive cannot be predicted.

Each primitive requires that the user specify a TRANSFER.MODE that governs the format in which each file in the group is transmitted through the Network. TRANSFER.MODE may have any of the values listed in (xfermode .d) except "COMPRESSED".

The outcome of each primitive will be reported on a per-file basis under control of the OUTCF primitive. The group primitive itself will fail only if the basic group primitive mechanism fails for some reason.

PRIMITIVES

(GNCPY)Negotiate the transfer of a copy of a file group from one distant system to another
  GNCPY (source.state, source.group, destination.state, destination.mask, transfer.mode)

This primitive transmits a copy of each distant file "f" in SOURCE.GROUP at the distant system implied by SOURCE.STATE to the distant system implied by DESTINATION.STATE, where it is assigned the name generated by passing "f" through DESTINATION.MASK.

The transfer mode for the entire group of files is specified by TRANSFER.MODE (see -- xfermode).

(GNAPS)Negotiate the appending of a copy of a file group at one distant system to files at another distant system
  GNAPS (source.state, source.group, destination.state, destination.mask, transfer.mode)

This primitive transmits a copy of each distant file "f" in SOURCE.GROUP at the distant system implied by SOURCE.STATE to the distant system implied by DESTINATION.STATE, where it is appended to the file whose name is generated by passing "f" through DESTINATION.MASK

If the source host is a Tenex system, each source file is assumed to be a simple sequential file. If the destination host is also a Tenex system, each destination file is assumed to have the same characteristics, and the byte size of each source file is assumed to be the same as that of its destination file.

The transfer mode for the entire group of files is specified by TRANSFER.MODE (see -- xfermode).

Note: GNAPS and GNAPD (described below) are identical, except that in the former the destination filenames are derived from the source filenames, and in the latter the reverse it true, i.e., source filenames are derived from destination filenames.

(GNAPD)Negotiate the appending of copies of files at one distant system to a file group at another distant system
GNAPD (source.state, source.mask, destination.state, destination.group, transfer.mode)

This primitive appends to each distant file "f" in DESTINATION.GROUP at the distant system implied by DESTINATION.STATE, a copy of the file, at the distant system implied by SOURCE.STATE, whose name is generated by passing "f" through SOURCE.MASK.

If the source host is a Tenex system, each source file is assumed to be a simple sequential file. If the destination host is also a Tenex system, each destination file is assumed to have the same characteristics, and the byte size of each source file is assumed to be the same as that of its destination file.

The transfer mode for the entire group of files is specified by TRANSFER.MODE (see -- xfermode).

Note: GNAPD and GNAPS (described above) are identical, except that in the former the source filenames are derived from the destination filenames, and in the latter the reverse it true, i.e., destination filenames are derived from source filenames.

MANIPULATING DIRECTORIES

INTRODUCTION

The primitives described in this section manipulate and examine local directories, and directories at the distant system most recently opened with OPEN. Most distant systems require (of DDIR) that the LOGIN primitive have been issued.

PRIMITIVES

(DFDIR)Set the default directory for distant files
DFDIR (directory)

This primitive sets the default working directory at the distant system. All subsequent distant filenames which do not explicitly specify a distant directory are taken to reside in directory DIRECTORY.

DFDIR does not establish access to DIRECTORY; it simply specifies it as a default, to be applied to those distant filenames in which a directory field does not explicitly appear.

This default specification is overridden by a subsequent LOGIN primitive or by another DFDIR.

(LDIR)Create a local file that contains a local directory listing
LDIR (local.directory.mask, destination.filename)

This primitive creates a local sequential text file DESTINATION.FILENAME containing a list of the local filenames implied by LOCAL.DIRECTORY.MASK

The implication algorithm is the same as that described in (grpdesc).

Each filename in the file has directory, name, extension, version, temporary, protection, and account fields, and is terminated with CR LF.

(DDIR)Create a local file that contains a distant directory listing
DDIR (distant.directory.mask, destination.filename)

This primitive creates a local sequential text file

DESTINATION.FILENAME containing a list of the distant filenames implied by DISTANT.DIRECTORY.MASK.

If the distant host is a Tenex system, the implication algorithm is the same as that described in (grpdesc), and each filename in the file contains directory, name, extension, version, temporary, protection, and account fields. In any case, each filename is terminated by CR LF.

Most distant Tenex systems prohibit an asterisk ('*) in the directory field of DISTANT.DIRECTORY.MASK.

(LTRNS)Translate a local group file
LTRNS (source.filename, mask, destination.filename)

This primitive creates a second local file DESTINATION.FILENAME by passing each filename in the local file SOURCE.FILENAME through MASK. The contents of the source file must conform to the syntax requirements of (grpsyntax), and the masking algorithm is that described in (masking).

The destination file created is a sequential text file containing the list of translated filenames, each terminated by CR LF, and contains no extraneous material (e.g., formatting characters or comments).

OBTAINING NETWORK-RELATED INFORMATION

INTRODUCTION

The primitives described in this section test hypotheses about the state of the Network or parameters related to it.

PRIMITIVES

(VHOST) Verify the syntax of a purported host name
VHOST (host)

This primitive verifies that HOST conforms to the FTPFRK syntax requirements for host names.

(hostsyntax) A host name can be a standard (i.e, official) host name, a nickname known to the local Tenex monitor, or a decimal host address. Host names and nicknames must be specified in upper-case.

(VNET) Verify that Tenex is operationally connected to the ARPANET
VNET ()

This primitive verifies that the local Tenex system is logically connected to the Network and that the subnet is functioning.

LOCAL AND DISTANT MAIL

INTRODUCTION

The primitives described in this section format and deliver mail to both local and distant users.

PRIMITIVES

(LMAIL)Send a copy of a suitably formatted local file as mail to a local user
LMAIL (user, filename)

This primitive mails the contents of the local file FILENAME created by FMTML to the specified local user USER. USER must be a local directory name.

(DMAIL)Send a copy of a suitably formatted local file as mail to a distant user
DMAIL (user, filename)

This primitive mails the contents of the local file FILENAME created by FMTML to the specified distant user USER in the distant system most recently opened with the OPEN primitive. If the distant host is a Tenex system, USER must, in most cases, be a distant directory name.

(QMAIL)Queue a suitably formatted local file for later delivery as mail to a user
QMAIL (filename, staging.directory, host, user)

This primitive queues in the local directory STAGING.DIRECTORY, a copy of the local file FILENAME created by FMTML, for later delivery by a local background process to the local or distant USER at host HOST.

HOST must either be "", in which case USER is taken to be a local directory name, or conform to the syntax requirements of (hostsyntax), in which case USER is taken to be a distant user at host HOST. If the distant host is a Tenex system, USER must, in most cases, be a distant directory name.

(FMTML)Create a local file suitable for mailing
FMTML (filename, author, author.host, title, header, text)

This primitive creates a local sequential text file FILENAME, suitable for transmission as mail to a local or distant user, from the following components:

1) The TEXT of the message (any character string, which may include CR LF's).
2) Its TITLE (any character string not containing CR LF).
3) The name of the AUTHOR (any character string not containing CR LF).
4) The author's host AUTHOR.HOST, with syntax as described in (hostsyntax).
5) Any additional information to be included literally in the HEADER of the mail file. Specifically, zero of more elements of the form:

```
<keyword> ': <text> <CRLF>
```

FMTML creates a mail file whose format is in accord with Network standards, currently specified by RFC 561 (see --18516,). The following is an example of the kind of file currently generated by FMTML:

```
From: white(JEW) at SRI-ARC
Date: 3 JUN 1974 1649-PDT
Subject: Example of the FMTML Primitive
Note: This is an optional field.

This is the text
    of the message.
```

The mail file above was generated by the primitive:

```
FMTML (filename, "white(JEW)", "ARC", "Example of the
FMTML Primitive", "Note: This is an optional field.CRLF",
"This is the textCRLF   of the message.")
```

MANIPULATING STATE RECORDS

INTRODUCTION

The primitives described in this section, by providing a mechanism for maintaining several distant file systems open simultaneously:

1) Permit requests to several distant systems to be interleaved.

2) Make possible a set of primitives (already described) which transfer files directly between two distant hosts.

In supporting these two activities, the notion of a "state record" is introduced. Conceptually, a state record is a labeled container which holds all of the information that FTPFRK generates internally and must maintain to deal intelligently with a distant system. Several such containers can exist simultaneously. At any point in time, one container is "mounted" and actively in use, and the others are on a shelf awaiting use.

NOTE CAREFULLY: All of the primitives described in this document that require that a distant system be open (or that themselves open or close one), implicitly manipulate the mounted state.

State labels are ASCII character strings whose case is significant (e.g., "ARCHIVER", "Archiver", and "archiver" are valid and different labels).

Primitives are provided to label a state record, relabel it, unlabel it, mount it, shelve it, discard it entirely, or clear off the entire shelf.

PRIMITIVES

(SHELV)Shelve the mounted stat#
SHELV (label)

This primitive labels the mounted state LABEL, replacing any existing label, and shelves it. A virgin, unlabeled state is left mounted.

If the mounted state is already labeled, it is unnecessary to explicitly shelve it before mounting a second state, since the saving of the previous state is implied in the MOUNT primitive.

(MOUNT)Mount a previously shelved state
MOUNT (label)

This primitive terminates or, if its labeled, shelves the mounted state and mounts the previously labeled state LABEL. If the requested state is already mounted, the primitive is a nop.

(RESET)Reset the mounted state
RESET ()

This primitive terminates the mounted state, leaving it labeled (if labeled) but in its original, virgin condition.

(DSCD)Discard a labeled state
DSCD (label)

This primitive terminates the previously labeled state LABEL, whether currently mounted or shelved, and discards it. If the state was mounted, a virgin, unlabeled state is left in its place.

(LBL)Label the mounted state
LBL (label)

This primitive assigns the label LABEL to the mounted state, replacing any label which might have been previously assigned to it.

(RELBL)Relabel a labeled state
RELBL (current.label, proposed.label)

This primitive assigns the new label PROPOSED.LABEL to the previously labeled state CURRENT.LABEL, whether currently mounted or shelved.

(UNLBL)Unlabel the mounted state

```
UNLBL ()
```

This primitive unlabels the mounted state. If the mounted state has no label, the primitive is a nop.

(CLEAR)Clear the state shelf

```
CLEAR ()
```

This primitive terminates and discards every shelved state, and unlabels the mounted state.

EXAMPLES

The following scenarios are simple examples of FTPFRK's use. In the examples, FTPFRK primitives are represented in the following format:

```
opcode (argument 1, argument 2, ... argument n)
```

The value of an argument is sometimes enclosed in quotes for readability.

Virtually no error checks are performed in the scenarios. In any real application, of course, the successful completion of each primitive should be verified before preceeding with the next.

In each example, the "local system" is SRI-ARC.

Example 1. Retrieving a distant file

The following scenario retrieves a copy of the file <WHITE>FTPFRK.NLS from OFFICE-1.

```
%initialize for FTPFRK session%
   BEGIN ()
%open OFFICE-1's file system%
   OPEN (OFFICE-1)
%login as WHITE%
   LOGIN (white, secret, 606)
%retrieve a copy of the file%
   DRTR (ftpfrk.nls, ftpfrk.nls, COMPRESSED)
%close OFFICE-1%
   CLOSE ()
%terminate the FTPFRK session%
   END ()
```

Example 2. Moving a group of files to a distant system

The following scenario moves all of the FTPFRK-related files from SRI-ARC to BBN.

```
%initialize for FTPFRK session%
   BEGIN ()
%open BBN's file system%
   OPEN (BBN)
%login as WHITE%
   LOGIN (white, jimbo, 203)
%disable generation of group operation error file%
   OUTCF (FAILURE, ""  NO)
%copy all of the FTPFRK-related files to BBN%
   GDSTR (*.*;*, ftpfrk.*;*, TENEX)
%delete the sources for the files that were successflly copied%
   GLDEL ([suc])
%close BBN%
   CLOSE ()
%delete outcome file%
   LELM ([suc])
%terminate the FTPFRK session%
   END ()
```

Example 3. Sending mail

The following scenario delivers to Postel at USC-ISI, or if necessary queues for later delivery by the system, a message whose text is stored in the local file MSG.TXT. A copy of the message is also deposited in the author's mailbox.

```
%initialize for FTPFRK session%
   BEGIN ()
%format the mail to be sent%
   FMTML (mail, WHITE, ARC, "Draft of Support Protocol Strategy",
     "NOTE: Add your changes and distribute.CRLF", ↑S↑Fmsg)
%send the mail to POSTEL@ISI and leave the author a copy%
   LMAIL (white, mail)
   IF OPEN (ISI)
      THEN DMAIL (postel, mail)
      ELSE QMAIL (mail, net, ISI, postel)
   CLOSE ()
%cleanup%
   LELM (msg)
   LELM (mail)
%terminate the FTPFRK session%
   END ()
```

Example 4. Moving a file between two distant systems

The following scenario moves the file FTPFRK.SAV from OFFICE-1 to BBN.

```
%initialize for FTPFRK session%
   BEGIN ()
%make preparations at the source host%
   OPEN (OFFICE-1)
   LOGIN (white, secret, 606)
   SHELV (SRC)
%make preparations at the destination host%
   OPEN (BBN)
   LOGIN (white, jimbo, 203)
   LBL (DST)
%move the file%
   NCPY (SRC, ftpfrk.sav, DST, ftpfrk.sav, TENEX)
   MOUNT (SRC)
   DDEL (ftpfrk.sav)
%close both distant systems%
   CLEAR ()
%terminate the FTPFRK session%
   END ()
```

Example 5. A final example

The following scenario retrieves from BBN the files listed in the file WORKLIST.TXT at OFFICE-1, reporting any difficulties to WHITE@OFFICE-1.

```
%initialize for FTPFRK session%
   BEGIN ()
%fetch list of files to be retrieved%
   OPEN (OFFICE-1)
   LOGIN (white, secret, 606)
   DRTR (worklist.txt, worklist, TENEX)
   SHELV (MASTER)
%disable generation of group operation success file%
   OUTCF (SUCCESS, "", NO)
%retrieve requested files%
   OPEN (BBN)
   LOGIN (white, jimbo, 203)
   GDRTR (worklist, <*>*.*;*, APPROPRIATE)
   CLOSE ()
%acknowledge request%
   MOUNT (MASTER)
   FMTML (reply, System, ARC, "File Retrieval Request
     Acknowledgment", "NOTE: The files you requested be
     transferrred have been successfully retrieved, except for any
     listed below.CRLF", ↑S↑F[abr])
   DMAIL (white, reply)
   CLOSE ()
%cleanup%
   LELM (worklist)
   LELM (reply)
   LELM ([abr])
%terminate the FTPFRK session%
   END ()
```

APPENDICES

FTPFRK LIMITATIONS

1) Long files, in the strict Tenex sense (i.e., files which have one or more pages with numbers greater than decimal 511), are not supported by FTPFRK.

2) As of this writing, negotiated file transfers (i.e., transfers between two distant systems) fail because the FTP command PASV is not implemented by most FTP server processes.

3) Compressed transfers are illegal in append primitives, and in negotiated transfers.

CURRENT VALUES OF SYSTEM PARAMETERS

FTPFRK ARGUMENTS

(unmdel)Argument file unmatched delimiter %unmdel = ↑U
(argesc)Symbolic argument prefix %argesc = ↑S
(filarg)File argument operator %filarg = ↑F
(cpxarg)Complex argument operator %cpxarg = ↑X
(noparg)NOP argument operator %noparg = ↑N
(remlen)Maximum length of a remote argument %remlen = 3000 characters
Maximum length of an immediate argument = 69 characters
(argcnt)Maximum number of arguments required by any primitive %argcnt = 6
Maximum number of local arguments per primitive = 1
Maximum number of remote arguments per primitive = 14
(winsiz)Number of FTPFRK address space pages for mapping remote, file, and complex arguments %winsiz = 256

GROUP OPERATIONS

(grpcms)Character which marks start of comment in group file %grpcms = %
(grpcme)Character which marks end of comment in group file %grpcme = %

MAIL

(lmlmxl)Maximum size of a mail file destined for a local user %lmlmxl = 80,000 characters

STATES

(lblmxl)Maximum length of state label %lblmxl = 40 characters
(maxsta)Maximum number of simultaneously shelved states %maxsta = 12

TIME LIMITS

Maximum wait time for completion of ICP = 15 seconds
Maximum wait time for delivery of FTP command to distant system = 10 seconds
Maximum wait time for return of FTP reply = 300 seconds

WORK FILES

<FTPFRK-LISTING>.TXT
This is the work file created by group primitives whenever

the group is specified by a file group descriptor, i.e., when FTPFRK is required to generate the group file. The file is deleted with the LELM primitive.

[FTPFRK-PACKED].SAV

This is the work file which holds the compressed version of a file transmitted from the local system by a compressed transfer. The file is deleted with the LELM primitive.

MISCELLANEOUS

(litesc)Literal escape character %litesc = ↑V
(replen)Maximum length of an FTP reply %replen = 200 characters
(fnlen)Maximum length of a Tenex filename %fnlen = 60 characters
Maximum length of an FTPFRK diagnostic = 74 characters

DEFAULTS

Typeout of debug information = ON (iff DDT is loaded)
Group operation success outcome filename = [SUC].TXT
Group operation success outcome file append.or.not = YES
Group operation failure outcome filename = [ABR].TXT
Group operation failure outcome file append.or.not = YES

Extension for local filenames = TXT

Except as noted above, all fields of local filenames (and distant filenames whenever the distant host is a Tenex system) are subject to the normal, system defaults.

SUMMARY OF PRIMITIVES

SPECIFYING FTPFRK PARAMETERS

Initialize for FTPFRK session
    BEGIN ()
Terminate FTPFRK session
    END ()
Reset FTPFRK session
    SYSRS ()
Set contact socket for distant systems
    SOCK (distant.socket)
Specify a local file to receive group operation outcome information
    OUTCF (type.of.outcome, filename, append.or.not)
Enable or disable the typeout of debug information
    DEBUG (on.or.off)
Pre-supply FTPFRK argument(s)
    NOP (argument.1, argument.2, ... argument.n)

CONNECTING TO A DISTANT SYSTEM

Open a distant file system
    OPEN (host)
Login at the distant system
    LOGIN (user, password, account)
Close the distant file system
    CLOSE ()
Connect to an arbitrary server process
    ICP (host, socket, byte.size)

MANIPULATING DISTANT FILES

Rename a distant file
    DREN (current.name, proposed.name)
Delete a distant file
    DDEL (filename)

MANIPULATING GROUPS OF DISTANT FILES

Rename a distant file group
    GDREN (current.group, proposed.mask)
Delete a distant file group
    GDDEL (group)

MANIPULATING LOCAL FILES

Rename a local file

```
      LREN (current.name, proposed.name)
   Delete a local file
      LDEL (filename)
```

MANIPULATING GROUPS OF LOCAL FILES

```
   Rename a local file group
      GLREN (current.group, proposed.mask)
   Delete a local file group
      GLDEL (group)
```

LOCAL FILE UTILITIES

```
   Eliminate a local file
      LELM (filename)
   Create a copy of a local file with EOL replaced by CRLF
      LSUB (source.filename, destination.filename)
   Create a portrayed version of a local file
      LPOR (source.filename, destination.filename)
```

LOCAL FILE GROUP UTILITIES

```
   Eliminate a local file group
      GLELM (group)
   Create a copy of a local file group in which EOL is replaced by
   CRLF
      GLSUB (source.group, destination.mask)
   Create a portrayed version of a local file group
      GLPOR (source.group, destination.mask)
```

TRANSFERRING FILES WITHIN THE LOCAL SYSTEM

```
   Replicate a local file
      LCPY (source.filename, destination.filename)
   Append a copy of one local file to another
      LAPP (source.filename, destination.filename)
```

TRANSFERRING GROUPS OF FILES WITHIN THE LOCAL SYSTEM

```
   Replicate a local flE GROUP
      GLCPY (source.group, destination.mask)
   Append a copy of a local file group to local files
      GLAPS (source.group, destination.mask)
   Append copies of local files to a local file group
      GLAPD (source.mask, destination.group)
```

TRANSFERRING FILES BETWEEN THE LOCAL AND DISTANT SYSTEMS

Create a local copy of a distant file
  DRTR (distant.filename, local.filename, transfer.mode)
Create a distant copy of a local file
  DSTR (distant.filename, local.filename, transfer.mode)
Append a copy of a local file to a distant file
  DAPP (distant.filename, local.filename, transfer.mode)

TRANSFERRING GROUPS OF FILES BETWEEN THE LOCAL AND DISTANT SYSTEMS

Create a local copy of a distant file group
  GDRTR (distant.group, local.mask, transfer.mode)
Create a distant copy of a local file group
  GDSTR (distant.mask, local.group, transfer.mode)
Append a copy of a local file group to distant files
  GDAPS (distant.mask, local.group, transfer.mode)
Append copies of local files to a distant file group
  GDAPD (distant.group, local.mask, transfer.mode=/

transferring files between two distant systems

negotiate the transfer of a copy of a file from one distant system to another
  NCPY (source.state, source.filename, destination.state, destination.filename, transfer.mode)
Negotiate the appending of a copy of a file at one distant system to a file at another distant system
  NAPP (source.state, source.filename, destination.state, destination.filename, transfer.mode)

TRANSFERRING GROUPS OF FILES BETWEEN TWO DISTANT SYSTEMS

Negotiate the transfer of a copy of a file group from one distant system to another
  GNCPY (source.state, source.group, destination.state, destination.mask, transfer.mode)
Negotiate the appending of a copy of a file group at one distant system to files at another distant system
  GNAPS (source.state, source.group, destination.state, destination.mask, transfer.mode)
Negotiate the appending of copies of files at one distant system to a file group at another distant system
  GNAPD (source.state, source.mask, destination.state, destination.group, transfer.mode)

MANIPULATING DIRECTORIES

Set the default directory for distant files
  DFDIR (directory)

Create a local file that contains a local directory listing
    LDIR (local.directory.mask, destination.filename)
Create a local file that contains a distant directory listing
    DDIR (distant.directory.mask, destination.filename)
Translate a local group file
    LTRNS (souce.filename, mask, destination.filename)

OBTAINING NETWORK-RELATED INFORMATION

Verify the syntax of a purported host name
    VHOST (host)
Verify that Tenex is operationally connected to the ARPANET
    VNET ()

LOCAL AND DISTANT MAIL

Send a copy of a suitably formatted local file as mail to a local user
    LMAIL (user, filename)
Send a copy of a suitably formatted local file as mail to a distant user
    DMAIL (user, filename)
Queue a suitably formatted local file for later delivery as mail to a user
    QMAIL (filename, staging.directory, host, user)
Create a local file suitable for mailing
    FMTML (filename, author, author.host, title, header, text)

MANIPULATING STATE RECORDS

Shelve the mounted state
    SHELV (label)
Mount a previously shelved state
    MOUNT (label)
Reset the mounted state
    RESET ()
Discard a labeled state
    DSCD (label)
Label the mounted state
    LBL (label)
Relabel a labeled state
    RELBL (current.label, proposed.label)
Unlabel the mounted state
    UNLBL ()
Clear the state shelf
    CLEAR ()

ERROR MESSAGES

(diags)<JSYS error message>

The diagnostic returned by FTPFRK may be a system error message obtained from the monitor following a JSYS error.

<error message from distant system>

The diagnostic returned by FTPFRK may be an error message generated by the distant system.

<error message>. ...

Any diagnostic, whether relayed from the monitor, from a distant system, or generated by FTPFRK itself, which ends with "..." has been truncated because it exceeds the maximum allowable length.

A distant system's already open.

Either the OPEN or ICP primitive, each of which establishes a logical connection to a distant system, has been issued while a distant system was already open. The proper course of action is to either close the open system, or shelve that state and begin a new one.

Address space window overflow.

More than %winsiz pages are required for mapping in remote, file, and complex arguments. If the superior fork's calling sequence is correct, FTPFRK must be patched or reassembled.

Bad use of *.

A group primitive has encountered a Tenex filename in which '* has been used improperly, i.e., it appears along with other characters in some field of the filename, rather than comprising the entire field.

Bulk mail is not deliverable locally.

A text file presented for delivery as mail to a local user exceeds %lmlmxl characters in length, the somewhat arbitrary maximum length thought appropriate for storage in a Tenex mail system and thus allowed by FTPFRK. If the current maximum length is too restrictive, FTPFRK must be patched or reassembled.

Compressed negotiated transfers unsupported.

Compressed transfers are not supported for file transfers between two distant systems.

Connection closed during transmission.

The Network connection on which a file was being transmitted between the local and distant hosts was, for some reason, broken during the course of the transmission. The distant system may have crashed, or the Network failed. It's probably necessary to break connections with the distant system entirely via CLOSE, and start afresh.

Data error on Network connection.

An error was detected while transmitting or receiving data from the distant system on one of the physical connections between the two hosts. The distant system may have crashed, or the Network failed. It is probably necessary to break connections with the distant system entirely via CLOSE, and start afresh.

Distant host closed control connection.

One of the Network connections on which control information is transmitted between the local and distant hosts has, for some reason, been broken. The distant system may have crashed, or the Network failed. It is probably necessary to break connections with the distant system entirely via CLOSE, and start afresh.

Distant host has been timed out.

An action required of the distant system's FTP server process failed to occur within FTPFRK's timeout period. The distant system may have crashed, or the Network failed. It is probably necessary to break connections with the distant system entirely via CLOSE, and start afresh.

Distant host is disconnected from the Net.

A logical connection between the local system and the distant host cannot, at the moment, be established, since the distant host is not operationally connected to the Network. Wait until the host reconnects itself to the Network.

Duplicate label.

The state record label proposed in SHELV, LBL, or RELBL is already assigned to a previously labeled state record, either mounted or shelved. State labels must be unique.

FTP reply code out of range.

The distant FTP server process replied to an FTPFRK command with a reply code which was not in the range [000,999], in violation of the File Transfer Protocol.

FTP reply too long.

FTPFRK received a command reply from the distant FTP server process that exceeds %replen characters in length. If the reply is legitimate, FTPFRK must be patched or reassembled.

FTPFRK argument list overflow.

More than %argcnt arguments were supplied for a primitive; no primitive requires or will accept that many arguments. The superior fork's FTPFRK calling sequence is in error.

Filename syntax error.

The group file for a group operation contains one or more filenames whose format violates the syntax requirements specified in (grpsyntax). The group file has been incorrectly constructed or lists files in a non-Tenex system.

Filename too long.

A local filename exceeds %fnlen characters in length. No valid Tenex filename is that long.

First reply from FTP server not a greeting.

The File Transfer Protocol requires that the distant FTP server process issue a spontaneous reply whose code is 000 as soon as the logical connection between it and the local system is established. The reply received had an inappropriate reply code. The distant server process has thus violated the File Transfer Protocol.

Group filename too long.

The group file for a group operation contains one or more filenames whose length exceeds %fnlen characters. No valid

Tenex filename is that long. The group file has been incorrectly constructed or lists files in a non-Tenex system.

Group outcome filename too long.

The group outcome filename specified in OUTCF exceeds %fnlen characters in length. No valid Tenex filename can be that long.

HOSTN monitor table doesn't exist.

The monitor table called HOSTN, which contains Network-related information required by FTPFRK, does not exist in the monitor. FTPFRK is thus incompatible with the release of Tenex on which it is being run.

HSTNAM monitor table doesn't exist.

The monitor table called HSTNAM, which contains Network-related information required by FTPFRK, does not exist in the monitor. FTPFRK is thus incompatible with the release of Tenex on which it is being run.

ICP failure (socket number not read).

In either the OPEN or ICP primitive, the establishment of Network connections between the local system and the distant host failed in mid-stream. In particular, the ICP initial connection was opened but no socket number was sent by the distant server process. If re-issuing the primitive yields the same results, the distant FTP server process can be assumed to be malfunctioning.

IMPHRT monitor table doesn't exist.

The monitor table called IMPHRT, which contains Network-related information required by FTPFRK, does not exist in the monitor. FTPFRK is thus incompatible with the release of Tenex on which it is being run.

Inferior fork died during unpack.

An FTPFRK bug; notify a systems programmer.

Invalid ICP byte size.

The byte size specified for the Network connections to be established by the ICP primitive is not in the range [1,255] and would therefore be in violation of Host-Host Protocol (see -- 8246,).

Invalid socket number.

The socket number specified for the distant server process in the ICP primitive is not in the range [0,2**32-1] and would therefore be in violation of Host-Host Protocol (see -- 8246,).

JOBNAM monitor table doesn't exist.

The monitor table called JOBNAM, which contains information required by FTPFRK, does not exist in the monitor. FTPFRK is thus incompatible with the release of Tenex on which it is being run.

LHOSTN monitor table doesn't exist.

The monitor table called LHOSTN, which contains Network-related information required by FTPFRK, does not exist in the monitor. FTPFRK is thus incompatible with the release of Tenex on which it is being run.

Label too long or null.

A state record label exceeds FTPFRK's self-imposed, assembly-parameter maximum of %lblmxl characters.

Long files are not supported.

A local file designated by the current primitive is a long file in the strict Tenex sense, i.e., it has a page numbered greater than 511 in its page table. Long files are not supported by FTPFRK.

Missing delimiter.

An FTPFRK (complex) argument file contains an argument that is missing a right delimiter. That is, the end of the argument file was encountered before the matching right delimiter was found. The argument file has been improperly constructed.

Missing error state pop.

An FTPFRK bug; notify a systems programmer.

Missing error state push.

An FTPFRK bug; notify a systems programmer.

NETRDY monitor table doesn't exist.

The monitor table called NETRDY, which contains Network-related information required by FTPFRK, does not exist in the monitor. FTPFRK is thus incompatible with the release of Tenex on which it is being run.

Name filename field unspecified.

A filename extracted from the group file, or the mask specified for the group operation contains no name field. The name field of a Tenex filename can be neither defaulted nor specified as null.

No distant system is open.

The mounted state does not designate an open distant system, but must for this primitive. Either OPEN a distant file system or MOUNT the desired state record before re-attempting the primitive.

No previous state.

An FTPFRK bug; notify a systems programmer.

No reply code in FTP response.

The reply from the distant system's FTP server process is not prefixed by a decimal reply code as required by the File Transfer Protocol.

No shelf space available.

Only %maxsta state records may be shelved simultaneously. An existing state must be discarded.

No such FTPFRK operation.

The op code specified is not one of those specified in (opcodes:z). No such primitive exists.

No such answer.

One of the arguments to the current primitive must, but does not have, either the value "YES" or "NO" (in upper-case).

No such group operation outcome file.

The first argument to the OUTCF primitive, which specifies the type of outcome to be recorded in the specified file must, but does not have, either the value "SUCCESS" or "FAILURE" (in upper-case).

No such host.

A non-existent host was specified as an argument. Host names must conform to the syntax specified in (hostsyntax).

No such label.

No such state record, either shelved or mounted, exists.

No such staging directory.

QMAIL requests that the mail file be staged for later delivery in a non-existent local directory. Directory names may be specified in either upper- or lower-case, so look elsewhere for the problem.

No such switch setting.

One of the arguments to the current primitive must, but does not have, either the value "ON" or "OFF" (in upper-case).

No such transfer type.

Only "ASCII", "TENEX", "COMPRESSED", and "APPROPRIATE" (in upper-case) are valid transfer types. See (xfermode).

Only one interval timer permitted.

An FTPFRK bug; notify a systems programmer.

SNAMES monitor table doesn't exist.

The monitor table called SNAMES, which contains information required by FTPFRK, does not exist in the monitor. FTPFRK is thus incompatible with the release of Tenex on which it is being run.

The IMP is down.

The subnet is not functioning. If everything at the local host is in order, consult the Network Control Center.

The IMP is going down.

The local system is in the process of being logically disconnected from the Network. Consult the operator.

The Network is turned off.

At the moment, the local system is not operationally connected to the Network. Consult the operator.

Too many FTPFRK arguments.

More than 1 local argument or 14 remote arguments have been claimed for the current primitive. It's impossible to transmit that many argument handles in the specified location.

Unexpected EOF in group file.

The contents of the group file are syntactically in error.

Unknown system error.

The JSYS error with which the current primitive failed has no ASCII error message associated with it.

Unlooked up system error.

An FTPFRK bug; notify a systems programmer.

Wrong number of FTPFRK arguments.

The number of arguments specified for the current primitive is inappropriate for that primitive. Either more arguments than required were specified, or one or more arguments were missing.

ALPHABETICAL LISTING OF PRIMITIVES

Legend:

```
O = Should a distant file system be open?
    Y = Yes
    N = No
    - = Irrevelant
L = Should the user be logged in at the distant system?
    Y = Yes
    N = No
    - = Irrevelant
S = Does the primitive implicitly apply to the mounted state?
    Y = Yes
    N = No
```

(opcodes)

O	L	S	OP	ARGUMENTS
N	N	N	BEGIN	()
-	-	N	CLEAR	()
-	-	Y	CLOSE	()
Y	Y	Y	DAPP	(distant.filename, local.filename, transfer.mode)
Y	Y	Y	DDEL	(filename)
Y	Y	Y	DDIR	(distant.directory.mask, destination.filename)
-	-	N	DEBUG	(on.or.off)
Y	Y	Y	DFDIR	(directory)
Y	-	Y	DMAIL	(user, filename)
Y	Y	Y	DREN	(current.name, proposed.name)
Y	Y	Y	DRTR	(distant.filename, local.filename, transfer.mode)
-	-	N	DSCD	(label)
Y	Y	Y	DSTR	(distant.filename, local.filename, transfer.mode)
-	-	N	END	()
-	-	N	FMTML	(filename, author, author.host, title, header, text)
Y	Y	Y	GDAPD	(distant.group, local.mask, transfer.mode)
Y	Y	Y	GDAPS	(distant.mask, local.group, transfer.mode)
Y	Y	Y	GDDEL	(group)
Y	Y	Y	GDREN	(current.group, proposed.mask)
Y	Y	Y	GDRTR	(distant.group, local.mask, transfer.mode)
Y	Y	Y	GDSTR	(distant.mask, local.group, transfer.mode)
-	-	N	GLAPD	(source.mask, destination.group)
-	-	N	GLAPS	(source.group, destination.mask)
-	-	N	GLCPY	(source.group, destination.mask)
-	-	N	GLDEL	(group)
-	-	N	GLELM	(group)
-	-	N	GLPOR	(source.group, destination.mask)
-	-	N	GLREN	(current.group, proposed.mask)
-	-	Y	GLSUB	(source.group, destination.mask)
Y	Y	N	GNAPD	(source.state, source.mask, destination.state, destination.group, transfer.mode)
Y	Y	N	GNAPS	(source.state, source.group, destination.state, destination.mask, transfer.mode)
Y	Y	N	GNCPY	(source.state, source.group, destination.state, destination.mask, transfer.mode)
N	N	Y	ICP	(host, socket, byte.size)
-	-	N	LAPP	(source.filename, destination.filename)
-	-	Y	LBL	(label)
-	-	N	LCPY	(source.filename, destination.filename)
-	-	N	LDEL	(filename)
-	-	N	LDIR	(local.directory.mask, destination.filename)
-	-	N	LELM	(filename)
-	-	N	LMAIL	(user, filename)
Y	-	Y	LOGIN	(user, password, account)
-	-	N	LPOR	(source.filename, destination.filename)
-	-	N	LREN	(current.name, proposed.name)

-	-	N	LSUB	(source.filename, destination.filename)
-	-	N	LTRNS	(source.filename, mask, destination.filename)
-	-	N	MOUNT	(label)
Y	Y	N	NAPP	(source.state, source.filename=@ destination.state, destination.filename, transfer.mode)
Y	Y	N	NCPY	(source.state, source.filename, destination.state, destination.filename, transfer.mode)
-	-	N	NOP	(argument.1, argument.2, ... argument.n)
N	N	Y	OPEN	(host)
-	-	N	OUTCF	(type.of.outcome, filename, append.or.not)
-	-	N	QMAIL	(filename, staging.directory, host, user)
-	-	N	RELBL	(current.label, proposed.label)
-	-	Y	RESET	()
-	-	Y	SHELV	(label)
-	-	N	SOCK	(distant.socket)
-	-	N	SYSRS	()
-	-	Y	UNLBL	()
-	-	N	VHOST	(host)
-	-	N	VNET	()

ASSEMBLY LANGUAGE CALLING SEQUENCE

The following is a suggested FTPFRK assembly-language calling sequence.

```
%create an inferior fork%
   movsi a,200000     ;create
   cfork              ; the
   jrst  error        ;  fork
   hrrzm a,fh         ;save fork handle
%load FTPFRK%
   movsi a,100001     ;fetch JFN
   hrroi b,[asciz/<SYSTEM>FTPFRK.SAV/]
   gtjfn              ; for FTPFRK
   jrst  error        ;  SAV file
   hrrzm a,jfn        ;map
   hrl   a,fh         ; FTPFRK
   get                ;  into fork
%allow remote arguments%
   hlrzs a            ;fetch FTPFRK's existing
   rpcap              ; enabled capabilities
   tlo   c,1b9        ;add to them the ability
   epcap              ; to map this fork
%initialize FTPFRK for session%
   hrroi a,[ascii/BEGIN/] ;op code
   setzm acs+1        ;zero arguments
   pushj p,prim       ;execute BEGIN primitive
%open BBN file system%
   hrroi a,[asciz/BBN/] ;store pointer to host name
   movem a,acs+2      ; as first argument
   move  a,[1b0+1]    ;one
   movem a,acs+1      ; remote argument
   hrroi a,[ascii/OPEN/] ;op code
   pushj p,prim       ;execute OPEN primitive
...
%terminate FTPFRK session%
   hrroi a,[ascii/END/] ;op code
   setzm acs+1        ;zero arguments
   pushj p,prim       ;execute END primitive
%cleanup and quit%
   move  a,fh         ;kill
   kfork              ; inferior fork
   move  a,jfn        ;release JFN for
   rljfn              ; FTPFRK
   jfcl               ;  SAV file
   haltf              ;halt
```

message subsystem error.

(J23656) 21-JUL-74 20:39;;;; Title: Author(s): Robert N. Lieberman/RLL; Distribution: /NDM([ACTION]) ; Sub-Collections: SRI-ARC; Clerk: RLL;

message subsystem error.

tried your message file today, gag, smash, cough, 'NO such Version' message appeared after th move message <CA> <BUG> <CA> command, tried several times with same results, did an expunge (at tenex level) and tried again, success, again did a snd for a test and got the ''no such version; message, apparently something is no beingexpunged and is causing a version mess up. 1

(J23657) 22-JUL-74 07:55;;;;   Title: Author(s): Michael D. Kudlick/MDK; Distribution: /SRL([ACTION]) ; Sub-Collections: SRI-ARC; Clerk: MDK;

Susan --- 1

PR's locations are: 1a

Systems Applications Inc
950 Northgate
San Rafael Calif 94903
(415) 472-4011 1b

2120 Pacific #208
San Francisco 94115
563-2959 1c

agreement with Kirk

(J23658) 22-JUL-74 09:44;;;; Title: Author(s): Richard W, Watson/RWW; Distribution: /KIRK([ACTION]) CHI([ACTION]) EKM([ACTION]) KEV([ACTION]) DSM([ACTION]) HGL([ACTION]) MDK([ACTION]) ; Sub-Collections: SRI-ARC; Clerk: RWW;

agreement with Kirk

I agree with Kirk that force case is the more commonly used "f" command and should be first level, in fact thats what we agreed to in the meeting on command language changes. Whoever is implementing it please set is tha way. Thanks 1

(J23659) 22-JUL-74 10:58;;;; Title: Author(s): Michael D. Kudlick/MDK; Distribution: /JHB([ACTION]) ; Sub-Collections: SRI-ARC; Clerk: MDK;

Jim ... 1

I have resigned today from SRI-ARC, effective August 3rd. My new position is Asst Professor at Univ of San Francisco Computer Science Dept. 2

It has been a real pleasure working with you, Jim, and I am very sorry that that relationship is terminating. 3

Elaine and I look forward to maintaining frequent contact with you and Maria. We hope the friendship continues to grow from its enjoyable beginnings. 4

Our location will continue to be 122 Liberty, 648-0306. Please let us know when you have settled in to your new home. 5

... Mike Kudlick 6

(J23660) 22-JUL-74 11;07;;;;   Title: Author(s): Michael D.
Kudlick/MDK; Distribution: /SRL([ACTION]) ; Sub-Collections:
SRI-ARC; Clerk: MDK;

Susan --- 1

Just a brief note to let you know that I resigned today from SRI-ARC, effective August 3rd. My new employer is the Univ of San Francisco (computer science dept). 2

I really enjoyed working with you, Susan, and am sorry that that relationship is ending. I hope our paths cross again in the future. 3

... Mike Kudlick 4

New NLS BUG: Y/N won't take yes for an answer

(J23661) 22-JUL-74 13:05;;;; Title: Author(s): Jeanne M. Beck/JMB; Distribution: /FDBK([ACTION]) DSM([ACTION]) CHI([ACTION]) ; Sub-Collections: SRI-ARC; Clerk: JMB;

New NLS BUG: Y/N won't take yes for an answer

Please fix soon; it makes our documentation lie.

JMB 22-JUL-74 13:05 23661

New NLS BUG: Y/N won't take yes for an answer

In the Substitute command, when I get to the field prompted "Y/N:" typing y gets a questionmark, 1

Superwatch Average Graphs for Week of 6/30/74

(J23662) 22-JUL-74 13:26;;;; Title: Author(s): Susan R, Lee/SRL; Distribution: /JCN([INFO-ONLY]) RWW([INFO-ONLY]) DCE([INFO-ONLY]) JCP([INFO-ONLY]) DVN([INFO-ONLY]) JAKE([INFO-ONLY]) DLS([INFO-ONLY]) WRF([INFO-ONLY]) DSM([INFO-ONLY]) ; Sub-Collections: SRI-ARC; Clerk: SRL; Origin: < LEE, WEEK6/30GRAPHS.NLS;1, >, 22-JUL-74 13:22 SRL ;;;;####;

Superwatch Average Graphs for Week of 6/30/74

```
TIME PLOT OF AVERAGE IDLE TIME FOR WEEK OF 6/30/74
x axis labeled in units of hr:min,  xunit = 30 minutes                 1

     82,5           *
     75,0  *****   **                                    *
     67,5  ****** ***                                   **
     60,0  ****** ****                                 ***
     52,5  ****** ****                             **  ***
     45,0  ***********                             *******
     37,5  ***************                      ** *******
     30,0 *****************         *   *       **********
     22,5 ******************        *  ***   *************
     15,0 ****************** *** ** ** ****  *************
      7,5 ************************************************
      0,0 *************************************************
          +'''''''''+'''''''''+'''''''''+'''''''''+'''''''''
         0:00      5:00     10:00     15:00     20:00                  1a

TIME PLOT OF AVERAGE NUMBER OF GO JOBS FOR WEEK OF 6/30/74
x axis labeled in units of hr:min,  xunit = 30 minutes                 2

      4,0                                  *
      3,5                             *  ***
      3,0                    **    *  *******
      2,5                   *****************
      2,0                  ******************
      1,5                  *********************
      1,0             *****************************  *
      0,5        *    *********************************
      0,0 *************************************************
          +'''''''''+'''''''''+'''''''''+'''''''''+'''''''''
         0:00      5:00     10:00     15:00     20:00                  2a

TIME PLOT OF AVERAGE PER CENT OF CPU TIME CHARGED TO USER ACCOUNTS
FOR WEEK OF 6/30/74
x axis labeled in units of hr:min,  xunit = 30 minutes                 3

     53,9                     * *  *       *
     46,2                   ******** *** ****  ** *
     38,5                  ************************
     30,8             *****************************
     23,1        *    *********************************
     15,4        *    *********************************
      7,7  *   **** *************************************
      0,0 *************************************************
          +'''''''''+'''''''''+'''''''''+'''''''''+'''''''''
         0:00      5:00     10:00     15:00     20:00                  3a
```

Superwatch Average Graphs for Week of 6/30/74

TIME PLOT OF AVERAGE NUMBER OF USERS FOR WEEK OF 6/30/74
x axis labeled in units of hr:min, xunit = 30 minutes 4

```
 11                   **             **
 10                   ****     **   ***
  9                  ********* ** *****
  8                 *******************
  7                 *******************
  6             *************************
  5             **************************
  4            ****************************** **
  3  *  **     *********************************** *
  2 ************************************************
  1 ************************************************
  0 *************************************************
    +,,,,,,,,,+,,,,,,,,,+,,,,,,,,,+,,,,,,,,,+,,,,,,,,,
   0:00      5:00     10:00     15:00     20:00
```
4a

TIME PLOT OF AVERAGE NUMBER OF NETWORK USERS FOR WEEK OF 6/30/74
x axis labeled in units of hr:min, xunit = 30 minutes 5

```
  4                *
  3             *****
  2            *******
  1  *   *     ********  **********         * *
  0 *************************************************
    +,,,,,,,,,+,,,,,,,,,+,,,,,,,,,+,,,,,,,,,+,,,,,,,,,
   0:00      5:00     10:00     15:00     20:00
```
5a

TIME PLOT OF AVERAGE PER CENT OF SYSTEM USED IN OLDDNLS FOR WEEK OF 6/30/74
x axis labeled in units of hr:min, xunit = 30 minutes 6

```
2,0                                      *
0,0 *************************************************
    +,,,,,,,,,+,,,,,,,,,+,,,,,,,,,+,,,,,,,,,+,,,,,,,,,
   0:00      5:00     10:00     15:00     20:00
```
6a

Superwatch Average Graphs for Week of 7/7/74

(J23663) 22-JUL-74 13:39;;;; Title: Author(s): Susan R, Lee/SRL; Distribution: /JCN([INFO-ONLY]) RWW([INFO-ONLY]) DCE([INFO-ONLY]) JCP([INFO-ONLY]) DVN([INFO-ONLY]) JAKE([INFO-ONLY]) DLS([INFO-ONLY]) WRF([INFO-ONLY]) DSM([INFO-ONLY]) ; Sub-Collections: SRI-ARC; Clerk: SRL; Origin: < LEE, WEEK7/7/74GRAPHS,NLS;2, >, 22-JUL-74 13:37 SRL ;;;;####;

Superwatch Average Graphs for Week of 7/7/74

```
TIME PLOT OF AVERAGE IDLE TIME FOR WEEK OF 7/7/74
x axis labeled in units of hr:min,  xunit = 30 minutes               1

     75,0  *     *
     67,5  ** *  ***                                   ***
     60,0  ****  ***                                   ***
     52,5  ***********                                ****
     45,0  ***********                               *****
     37,5  ***********                           **  *****
     30,0 ************                         ***********
     22,5 ************** *                    ************
     15,0 *****************         *         ************
      7,5 ******************** * * ***  **  **************
      0,0 *************************************************
          +'''''''''+'''''''''+'''''''''+'''''''''+'''''''''
         0:00      5:00     10:00     15:00     20:00                1a
```

```
TIME PLOT OF AVERAGE NUMBER OF GO JOBS FOR WEEK OF 7/7/74
x axis labeled in units of hr:min,  xunit = 30 minutes               2

      4,5                     *   *
      4,0                     *   *  *** * **
      3,5                     ********** ****
      3,0                  ** ****************
      2,5                 ********************
      2,0               ***********************
      1,5             *************************    **
      1,0             *******************************
      0,5    * **   ***********************************
      0,0 *************************************************
          +'''''''''+'''''''''+'''''''''+'''''''''+'''''''''
         0:00      5:00     10:00     15:00     20:00                2a
```

```
TIME PLOT OF AVERAGE PER CENT OF CPU TIME CHARGED TO USER ACCOUNTS
FOR WEEK OF 7/7/74
x axis labeled in units of hr:min,  xunit = 30 minutes               3

     61,6                             **
     53,9                  ** ****** **** ***
     46,2             * * *********************
     38,5             ***************************  *
     30,8             *******************************
     23,1    * **    *********************************
     15,4   *****   ***********************************
      7,7  ***********************************************
      0,0 *************************************************
          +'''''''''+'''''''''+'''''''''+'''''''''+'''''''''
         0:00      5:00     10:00     15:00     20:00                3a
```

Superwatch Average Graphs for Week of 7/7/74

TIME PLOT OF AVERAGE NUMBER OF USERS FOR WEEK OF 7/7/74
x axis labeled in units of hr:min, xunit = 30 minutes 4

```
  13                     *
  12                     **           *
  11                    ***   *  *  ****
  10                 *******************
   9               **********************
   8               **********************
   7             ************************
   6             ************************    **
   5             *******************************
   4   **       **********************************
   3  ****** **************************************
   2  ***********************************************
   1 ************************************************
   0 *************************************************
     +'''''''''+'''''''''+'''''''''+'''''''''+'''''''''
    0:00      5:00     10:00     15:00     20:00
```
4a

TIME PLOT OF AVERAGE NUMBER OF NETWORK USERS FOR WEEK OF 7/7/74
x axis labeled in units of hr:min, xunit = 30 minutes 5

```
   5               *
   4             *****
   3             *****
   2            *******  ***   *
   1     *** **************************  *******
   0 *************************************************
     +'''''''''+'''''''''+'''''''''+'''''''''+'''''''''
    0:00      5:00     10:00     15:00     20:00
```
5a

TIME PLOT OF AVERAGE PER CENT OF SYSTEM USED IN OLDDNLS FOR WEEK OF 7/7/74
x axis labeled in units of hr:min, xunit = 30 minutes 6

```
 4.0                                  **
 2.0                              *   ***
 0.0 *************************************************
     +'''''''''+'''''''''+'''''''''+'''''''''+'''''''''
    0:00      5:00     10:00     15:00     20:00
```
6a

Superwatch Average Graphs for Week of 7/14/74

(J23664) 22-JUL-74 13:53;;;; Title: Author(s): Susan R, Lee/SRL; Distribution: /JCN([INFO-ONLY]) RWW([INFO-ONLY]) DCE([INFO-ONLY]) JCP([INFO-ONLY]) DVN([INFO-ONLY]) JAKE([INFO-ONLY]) DLS([INFO-ONLY]) WRF([INFO-ONLY]) DSM([INFO-ONLY]) ; Sub-Collections: SRI-ARC; Clerk: SRL; Origin: < LEE, WEEK7/14GRAPHS.NLS;2, >, 22-JUL-74 13:48 SRL ;;;;####;

Superwatch Average Graphs for Week of 7/14/74

TIME PLOT OF AVERAGE IDLE TIME FOR WEEK OF 7/14/74
x axis labeled in units of hr:min, xunit = 30 minutes 1

```
82,5    * **
75,0  ******* *
67,5  ******* *
60,0  **********
52,5  **********
45,0  ***********                                  *
37,5 ************                                ****
30,0 ************                         *** * *****
22,5 ************  *                      ***********
15,0 ************* ***                  *************
 7,5 ******************* **          ****************
 0,0 *************************************************
     +,,,,,,,,,+,,,,,,,,,+,,,,,,,,,+,,,,,,,,,+,,,,,,,,,
    0:00      5:00     10:00     15:00    20:00
```

1a

TIME PLOT OF AVERAGE NUMBER OF GO JOBS FOR WEEK OF 7/14/74
x axis labeled in units of hr:min, xunit = 30 minutes 2

```
 5,5                       *
 5,0                       *        *
 4,5                       ****  ****
 4,0                    ** ***********
 3,5                  ***************
 3,0              *   ******************
 2,5              *  ********************
 2,0              * **********************     *
 1,5             *************************   ****
 1,0             *********************************
 0,5           **************************************
 0,0 *************************************************
     +,,,,,,,,,+,,,,,,,,,+,,,,,,,,,+,,,,,,,,,+,,,,,,,,,
    0:00      5:00     10:00     15:00    20:00
```

2a

TIME PLOT OF AVERAGE PER CENT OF CPU TIME CHARGED TO USER ACCOUNTS
FOR WEEK OF 7/14/74
x axis labeled in units of hr:min, xunit = 30 minutes 3

```
61,6              *     *    ** * *
53,9              *  *******************
46,2             *************************   *
38,5             ************************* *******  *
30,8             ********************************** *
23,1            *************************************
15,4           **************************************
 7,7  ** *   ****************************************
 0,0 *************************************************
```

Superwatch Average Graphs for Week of 7/14/74

```
           +,,,,,,,,,+,,,,,,,,,+,,,,,,,,,+,,,,,,,,,+,,,,,,,,,
          0:00      5:00     10:00     15:00     20:00                3a
```

```
TIME PLOT OF AVERAGE NUMBER OF USERS FOR WEEK OF 7/14/74
x axis labeled in units of hr:min,  xunit = 30 minutes                4

        14                        *
        13                       ** *  *
        12                   ****** *  ******
        11                   ******** *******
        10                  ******************
         9                ********************
         8              **********************
         7             ************************
         6             ************************
         5            **************************    **
         4            ***********************************
         3  *      ****************************************
         2 ************************************************
         1 ************************************************
         0 *************************************************
           +,,,,,,,,,+,,,,,,,,,+,,,,,,,,,+,,,,,,,,,+,,,,,,,,,
          0:00      5:00     10:00     15:00     20:00                4a
```

```
TIME PLOT OF AVERAGE NUMBER OF NETWORK USERS FOR WEEK OF 7/14/74
x axis labeled in units of hr:min,  xunit = 30 minutes                5

         6                *
         5              ****
         4             *****
         3             ******
         2            ********      *  **
         1 *        ********************** *
         0 *************************************************
           +,,,,,,,,,+,,,,,,,,,+,,,,,,,,,+,,,,,,,,,+,,,,,,,,,
          0:00      5:00     10:00     15:00     20:00                5a
```

```
TIME PLOT OF AVERAGE PER CENT OF SYSTEM USED IN OLDDNLS FOR WEEK OF
7/14/74
x axis labeled in units of hr:min,  xunit = 30 minutes                6

       0,0 *************************************************
           +,,,,,,,,,+,,,,,,,,,+,,,,,,,,,+,,,,,,,,,+,,,,,,,,,
          0:00      5:00     10:00     15:00     20:00                6a
```

Requiescat in pacem

(J23665) 22-JUL-74 16:03;;;; Title: Author(s): Elizabeth J. (Jake) Feinler/JAKE; Distribution: /SRI-ARC([INFO-ONLY]) ; Sub-Collections: SRI-ARC; Clerk: JAKE;

Requiescat in pacem

The enterprise phone numbers and th NIC numbers (329-0740-1-2) have now been discontinued and the call directors will be removed soon. The sorter for the Xerox machine is already gone and the machine itself will go Monday, July, 29, 1974. The work Mil is doing to wrap up the catalogs takes precedence, but some of you may want to get a few last licks in on the Xerox machine before it departs. The copy centers can make reductions and multiple copies and the operator will place long distance phone calls for you after the demise. 1

newnls bugs and feedback

```
(J23666) 22-JUL-74 17:15;;;;  Title: Author(s): N. Dean Meyer/NDM;
Distribution: /FDBK( [ ACTION ] ) NDM( [ INFO-ONLY ] ) ;
Sub-Collections: SRI-ARC; I submitted this for you, ... Mike); Clerk:
MDK;        Origin: < KUDLICK, NDMBUGRPT.NLS;2, >, 22-JUL-74 17:12 MDK
;;;;####;
```

newnls bugs and feedback

Herewith some bugs and feedback on newnls, noted by NDM in (23635,): 1

The Line Processor bug which truncates each statement display to 72 characters must be fixed (23569,). 1a

The command "Detach Subsystem" does not work. 1b

When a User Program/subsystem (like MESSAGE) is loaded, the "WARNING -- no entry to program" for the L10 part misleads the user to thinking something is wrong. That message must be eliminated. 1c

When a user programmed subsystem is attached, where the subsystem keyword is different from the program name (although this will be avoided in the future), the keyword, not the program name, should be given in the message "Subsystem XXX Attached". 1c1

In TNLS, if I put two character searches (with an apostrophy) in an address expression, it shouldn't go to the beginning of the statement for the second search; it should search from the CM resulting from the DAE so far onward. 1d

Actually I'm not sure if it always blows it; I was going to the second occurance of a given character, so both searches were for the same character (that shouldn't matter, though). 1d1

Suggested Changes to Help System

(J23667) 23-JUL-74 13:11;;;; Title: Author(s): Richard W, Watson/RWW; Distribution: /CHI([INFO-ONLY]) HGL([INFO-ONLY]) EKM([INFO-ONLY]) DSM([INFO-ONLY]) KIRK([INFO-ONLY]) DVN([INFO-ONLY]) MDK([INFO-ONLY]) JMB([INFO-ONLY]) KEV([INFO-ONLY]) JCN([INFO-ONLY]) DCE([INFO-ONLY]) ; Sub-Collections: SRI-ARC; Clerk: RWW; Origin: < WATSON, HELP,NLS;2, >, 18-JUL-74 17:51 RWW ;;;;(JOHNSON, RWW,NLS;3,), 18-JUL-74 11:26 SLJ ;####;

Suggested Changes to Help System

The Help System 1

After much useful and interesting discussion between Mike, Harvey, Kirk, Dirk, Jeanne, Charles, Ken, and myself during the past two days with respect to the Help system, the following proposal seems to best meet the requirements of consistency, ease of learning, and ease of use. The proposal is a modification of one by Kirk (GJOURNAL,23633,). 2

1) Help is to be a command available from all subsystems rather than a subsystem. The Help command will be defaulted to repeat mode. A CD will exit the user from repeat mode. 2a

2) Help can be entered in one of three ways. 2b

a) typing two question marks in a row. (Harvey is not sure this can be implemented in CML, but we want to do it if possible). 2b1

b) typing ^Q 2b2

c) typing HELP as a command word. 2b3

The semantics of a) and b) are identical. 2c

The user is entered into the Help data base at a point appropriate to his current state, a definition and menu are printed. The user is then appropriately prompted as described below. 2c1

The semantics of c) are the following: 2d

When the user types HELP (as per his recognition mode), he will be prompted OK/T: in both TNLS and DNLS. If he hits OK, and he has not changed subsystems, since his last entry to HELP, then he is taken to his previous point else he is taken to the entry point in the data base for his current subsystem. 2d1

For typein it is as below. 2d2

Typein Semantics 2e

The user can type in a series of words that lead to a named mode in the database or a menu number. Either of these can be followed by a field indicating the type of view desired and an OK. He is not prompted for the view. Whether the view is a single special character or a colon followed by an alpha character is an open question. 2e1

Bug Semantics 2f

There are two types of entities to be bugged, menu items meaning "jump to item" and words meaning "jump to name", To distinguish between them three schemes have been proposed, I lean toward the first unless I hear strong arguement for either of the others, 2f1

(There is a question about what is to be done with multi-word "see alsos" if two bug marks are not used,) 2f1a

a) to jump to a menued item, you bug the menu number, but bugging any word means jump to word 2f1b

b) a bug anywhere in the top mode or some unnumbered lower node means jump to word; a bug in a numbered menu node means jump to that item, 2f1c

c) use one combination of mouse buttons for jump to item, another for jump to name, 2f1d

After a bug the user can enter a viewspecification as with typein followed by an OK, 2f2

Back 2g

A user can get a previous view by typing any number of -'s during typein; the system echoes (both DNLS and TNLS) the first line of definition to be printed and prompts Y/N: The system continues back to other views on N until a Y is typed, 2g1

More 2h

When the system fills the screen or has printed XX lines, it prompts "more Y/N", 2h1

Views 2i

The views to be printed include a one line "outline" view; an all line "full" view; a all line top node, one line menu nodes "definition" view, 2i1

The specificaions for how to handle "includes" is still open, 2j

Lineprocessor development time in manweeks

(J23668) 23-JUL-74 13:33;;;; Title: Author(s): Don I. Andrews/DIA;
Distribution: /RWW([INFO-ONLY]) ; Sub-Collections: SRI-ARC; Clerk:
DIA; Origin: < ANDREWS, NOTES.NLS;1, >, 23-JUL-74 13:13 DIA
;;;;####;

Lineprocessor development time in manweeks

Here is the (very rough) estimate of my time spent on Lineprocessor development. 1

Numbers represent manweeks and include startup time, design, implementation, debugging where applicable. 1a

(3) Display terminal search (reading, phoning,letters) 1b

(2) hardware - working with MEH, Rod 1c

(2) assembler/loader & PROM programmer on TEN 1d

(3) hardware test program(s) (both LP and TEN) 1e

(3) lyiteral collection algorithm & partial (85%) implementation 1f

(4) printer driver in LP & algorithm (several times) 1g

(2) printer program in TEN, mods to NLS 1h

(2) IMLAC-LP program (mouse and keyset on LP, not IMLAC) 1i

(2) mods to NLS for system restart button to work 1j

(20) main program development in LP 1k

was re-written several times due to hardware changes, mouse button problems, speed problems, lack of foresight etc. 1k1

Other involvements during same period 2

(2) writing NCC paper 2a

(1) NCC presentation/movie prepairation 2b

(1) NCC attendance 2c

(2) Display system considerations, frontend meeting & proposal, etc. 2d

(1) L10 changes 2e

(2) vacation 2f

(1) final report writing 2g

Bugs and Modifications

(J23669) 23-JUL-74 14:14;;;; Title: Author(s): David S. Maynard/DSM; Distribution: /NPG([ACTION]) ; Sub-Collections: SRI-ARC NPG; Clerk: DSM;

Bugs and Modifications

The file (nls,mods,1;xs) has a list of interesting and challenging modifications that have to be made to nls before September 1, Please look at it and move any tasks you have already done to the done branch and then choose one or more tasks which you would like to do, Hurry before all the really challenging tasks are gone, 1 file retun ring bug.

(J23670) 23-JUL-74 16:56;;;; Title: Author(s): Robert N. Lieberman/RLL; Distribution: /FDBK([ACTION]) ; Sub-Collections: SRI-ARC; Clerk: RLL;

file retun ring bug,

got the message 'insufficent space in file ring//' after a oad file and a jump to link command, Bad, 1 user prog suggestion: assignment of numbers to plex statements at beginning of statements

(J23671) 24-JUL-74 00:06;;;; Title: Author(s): Robert N. Lieberman/RLL; Distribution: /FDBK([ACTION]) NDM([ACTION]) ; Sub-Collections: SRI-ARC; Clerk: RLL;

user prog suggestion: assignment of numbers to plex statements at beginning of statements

How about a user prog that assins outline numbers to statemnts in a plex/branch/group under controlling viewspecs(levels only). Some debault couldbe used as the numbers (uppercase roman numbers, upper case roman letters, arabic numbers, lowercase roman letters, lowercase roman numbers.) There should be a simple command for just numering the plex with the given type (this could be a simple version of the overall command or aother command.) You can hhve a set mode command to set the kind of number or ask for it each ime therequest is made. 1

Position statement on RFCs

(J23672) 24-JUL-74 10:37;;;; Title: Author(s): Elizabeth J. (Jake) Feinler/JAKE; Distribution: /JCN([ACTION]) JEW([INFO-ONLY]) RWW([INFO-ONLY]) DCE([INFO-ONLY]) CHI([INFO-ONLY]) ; Sub-Collections: SRI-ARC; Clerk: JAKE;

JAKE 24-JUL-74 10:37 23672

Position statement on RFCs

Dave Crocker's frequent sndmsgs organizing handling of RFCs at Multics makes it even more imperative that we take a stand and announce our policy on this matter. May I please have some kind of feedback on this, Jim, before you leave again. I agree with Doug and Dick that if money is available for this, we should continue the service from NIC - not because we particularly want to expand services but primarily because it maintains a valuable collection here that is of use to both utility and research interests. Whether the decision is yes or no, however, we MUST make one. Our lack of response is making us look very bad. Realize life has been very hectic for you and this is not a criticism - it is a red flag, top of the stack, action message which I will be glad to follow through on when I know which direction you want to go. 1

Hot News Item!! a coming interview with the Wizard of Colorado

(J23673) 24-JUL-74 15:09;;;; Title: Author(s): Sandy L. Johnson, N. Dean Meyer/SLJ NDM; Distribution: /SRI-ARC([ACTION]) ; Sub-Collections: SRI-ARC; Clerk: SLJ; Origin: < JOHNSON, MERCURY.NLS;3, >, 24-JUL-74 15:04 SLJ ;;;;####;

SLJ NDM 24-JUL-74 15:09 23673

Hot News Item!! a coming interview with the Wizard of Colorado

[ACTION of course]

SLJ NDM 24-JUL-74 15:09 23673

Hot News Item!! a coming interview with the Wizard of Colorado

The first bi-centenial sun-in-leo celebration will be held closely yes, in varying degrees of yes, and next saturday, yes? 1

Here's what the mystical prognosis lady says: 2

At saturday, july 27, in the realm of 3 hours past noon, many tangents, swimming pool, as you like it, please feel, free, also pool table, and other gardens of earthly delight (i knew a girl like that once)... 2a

Please bring whatever you feel good about bringing, in whatever language, barbque is burning, so pot-luck and if you can help with some staples like paper plates, drinking utinsils et al let ms, miranda know thursday.. bring a towel if you want to enter pisces, and a musical instrument if other kinds of gliding sound more like it. You can choose to coordinate with or all surprise on what you want to. 2b

Yes, thank you mystical lady... And that's right folks, you really can't beat the deal at Meyer's Pleasure Palace! yata 3

```
Here's where:                                                          4

watch for   !  golden oak                                              5

"ISLANDS"  aY-----------\                                              6

           l!  !_<      !    _<75 Bear Gulch Drive, Portola Valley     7

           p!  !        !      851-0267                                8

           i!-----------/                      woodside rd (84)        9

           n!                                   !                      10

           e!    !                              !                      11

 ==========/!\===/!\===========================/!\===== 280            12

            !    s!                             !                      13

            !    a!                             !                      14

            !    n!                             !                      15

 -----------+    d!                             !                      16

            \-----+-----+--------------------------+------ alameda     17
```

Hot News Item!! a coming interview with the Wizard of Colorado

```
                 h!       \                         !                 18

                 i!        \santa                   !                 19

                 l!        !cruz                    !                 20

      stanford   l!        !                        !                 21

      shop cen  --!        !                        !                 22

 -----------------+-----+--+------------------------!------ el camino 23

                        !  !                        !                 24
```

Note that Golden Oak is a horseshoe; take the second Golden Oak (by the Islands in the road and the Alpine Swim and Tennis Club). 25

Outline Numbering User Program

(J23674) 24-JUL-74 15:36;;;; Title: Author(s): N. Dean Meyer/NDM;
Distribution: /RLL([INFO-ONLY]) FDBK([INFO-ONLY]) ;
Sub-Collections: SRI-ARC; Clerk: NDM;

Outline Numbering User Program

Robert: You really want the numbers in your file? you know the Output Processor ca do all that, right? It would be an easy thing for you or anyone to rght if that's really what you want, 1

Status of Search for Display Terminals for Lineprocessors

(J23675) 24-JUL-74 15:42;;;; Title: Author(s): Don I, Andrews/DIA; Distribution: /RWW([INFO-ONLY]) CHI([INFO-ONLY]) MEH([INFO-ONLY]) KEV([INFO-ONLY]) JCN([INFO-ONLY]) DCE([INFO-ONLY]) ; Sub-Collections: SRI-ARC; Clerk: DIA; Origin: < ANDREWS, NOTES.NLS;1, >, 24-JUL-74 15:06 DIA ;;;;####;

Status of Search for Display Terminals for Lineprocessors

Status of Display Survey 1

We are currently looking at three displays (in addition to Delta Data and Hazeltine) that we may use with Lineprocessors. The current opinion etc. of each: 1a

Datamedia 2500 1a1

The West coast REP is on vacation, will arrange demo on 7/29 for ??. Stanford Med. Center has some that we may get to look at. 1a1a

Bad things we know so for: 1a1b

Insert line takes forever (30ms). This is not as bad as Delta's delete however. The delete line time is 1 ms!!!! That should make it look very snappy. 1a1b1

The coordinates are quite screwed up, but computable. We may be on the very hairy edge of running 9600 baud be cause of the extra computing required to transform integers into the correct coordinates. 1a1b2

There is evidently no way to show bug selections, other than the character smashing technique used on the Hazeltines. 1a1b3

Cost: Quote is $2280 each, singles. We may want some options not included in that. 1a1c

Availability: ?? 1a1d

LSI ADM-2 1a2

we will have a short demo on Aug. 9, 9:30 AM. 1a2a

Bad things we know about now: 1a2b

Insert line takes 17 ms. Delete line takes 23 ms. Not too good, BUT the thing has a 40 char buffer and claims are that it can catch up if it doesn't get too many inserts/deletes in a row. We will have to see what the net effect is. 1a2b1

VERY DUMB: The damn thing wants coordinates in order: Y, then X. This is quite a pain for the Lineprocessor. Don't know what the result will be, but we will make it work unless other bad looking things come up and we eliminate it. 1a2b2

There is evidently no way to show bug selections, other than the character smashing technique used on the Hazeltines. 1a2b3

Cost: $2600 includes upper/lower case. 1a2c

Availability: 90 days, they say. New model and backlog is expected. 1a2d

Infoton Vistar Plus 1a3

Looks for all the world like we can't edit from computer connection. The rep is looking into this for us. This one is out unless we hear good news from the rep. 1a3a

By the way, the word from Tektroniccs is that they will make us the modified 4023 terminals as follows;: 1b

$5500 each, with 10% discount on our order of 16 = $4950 each. 1b1

(We sent a letter of intent to buy 16). 1b1a

They will deliver something like two of them per week, beginning 12 weeks after receipt or order. 1b2

They are meeting our specs which means that bug selections will be shown as reverse video. So will standout text. Delete line time for that one is 3.75 ms. 1b3

Comments do not work in Help links

(J23676) 24-JUL-74 18:49;;;; Title: Author(s): Kirk E. Kelley/KIRK; Distribution: /HGL([ACTION]) FDBK([INFO-ONLY]) DVN([INFO-ONLY]) JMB([INFO-ONLY]) ; Sub-Collections: SRI-ARC; Clerk: KIRK;

Comments do not work in Help links

I think it must be trying to parse the comment, Jump to link works fine, An example is at <documentation,help,nouns>, 1 comments in links work

(J23677) 24-JUL-74 18:53;;;; Title: Author(s): Kirk E. Kelley/KIRK; Distribution: /HGL([INFO-ONLY]) JHB([INFO-ONLY]) DVN([INFO-ONLY]) ; Sub-Collections: SRI-ARC; Clerk: KIRK;

comments in links work

was I imagining things? 1

The <nls,mods,> file

(J23678) 25-JUL-74 12:38;;;; Title: Author(s): Kirk E. Kelley/KIRK; Distribution: /SRL([INFO-ONLY]) ; Sub-Collections: SRI-ARC; Clerk: KIRK;

The <nls,mods,> file

The nls programmers (HGL EKM DSM CJM) have created a file <nls, mods,> which sounds like it might contain your Changed Status items, I can't think of a good way to interface to it via FDBK except to place a link in FDBK to mods at the appropriate branch, Mods has a seperate Done branch to which the nls programmers are suposed to move items, It's not clear if people will create a branch with their initials and move the items they want to do to their branch until the item is done, If only we had a Completed Task command! Ah well, I really don't have the time to spend on the feedback-decision mechanism right now (certainly not the authorization), but the way things were decided was abominable, That meeting of six(?) people ... Better than nothing?? The outcome is in <kudlick,newnls,> if you haven't seen it, Though <kudlick,newnls,> is what mods is based on and the programmers are (have) actively made alot of the changes and it means substantial changes to help, it has not been journalized or otherwise made public, In fact, JMB and DVN were told not to look at it!! 1

I'm looking forward to meeting your friends when they come out to CA, (Command Accept; thats Carriage Return for all you TNLS people out there unless you've changed it the Useroptions SUBSYSTEM with the Control (characters) command, Show also: ...) I've been working on Help too long, Will you really be here in a week? 2

swimming wet water pool

(J23679) 25-JUL-74 12:52;;;; Title: Author(s): Sandy L, Johnson/SLJ; Distribution: /SRI-ARC([INFO-ONLY]) ; Sub-Collections: SRI-ARC; Clerk: SLJ;

swimming wet water pool

in these days of hot let yourself know that menlo aterton high school pool is open to the pubics from 1:30 to 4:00 monday through friday and 10:00 to 5:00 sat and sunday. this has been a public service announcement from radio station WSLJ. 1

Near-future plan for ARC Computer Services

(J23680) 25-JUL-74 13:20;;;; Title: Author(s): Douglas C, Engelbart/DCE; Distribution: /JCN([ACTION]) SRI-ARC([INFO-ONLY]) BC([INFO-ONLY]) ; Sub-Collections: SRI-ARC; Clerk: DCE;

Near-future plan for ARC Computer Services

This is my understanding of the agreed-upon, near-future plan for ARC Computer Services stemming from this morning's EMC meeting (i.e. RWW, JCN and me). Implementation and operation of the scheduling means is in JCN's bailiwick; what is finally set up, and when it is to take effect, will be announced by him. 1

This new access-control mode will operate for five or six weeks and then be reviewed. 1a

Perhaps everyone isn't aware of the computer-resource picture: As of July 1 our machine has been bought by ARPA for a special project; they will move it to a new site at NASA Ames Research Center next January. Until then, NSW has found fnds to contract with ARC (JCN's domain) to have the system maintained and operated as a more-or-less NSW resource (with some unspecified service rights due to the new-owner ARPA group). After January, it will prbably be to an expanded OFFICE-1 (or perhaps and OFFICE-2) that ARC will turn to for its NLS support -- and for instance NIC and NSW contracts have explicit budgets included for their buying the necessary computer support. 1b

The specific need is to assure fair distribution of resources under the transient conditions between July 1 and Dec 31 (when we have to make explict purchase of support for each of our ARC activities). We want also to be getting ourselves more aware of the pinched-service conditions that will then prevail, and to start now on the process of adapting our mode of work toward those conditions. 1c

In the past we have considered part of the IPTO R&D support money to be paying for our basic "bootstrapping" mode of life, in which all of ARC was learning to live with AKW support. We no longer have sponsorship for that mode of work; our computer support comes only as direct expense to projects, or to the overhead margin we can spare from our operating budget. We are lucky to have the relatively flexible resource-allocation situation for a six-month adaptation period. 1d

Assume the SRI-ARC machine will support approximately 15 working jobs -- and that we'll handle the access and scheduling on the basis of slots, group quotas, etc. 2

We are obligated to supply four slots to outside ARPA contractors, who will be using Fortran, probably TECO, etc. Their service-support level during the day is not advertised as being heavy; e.g. perhaps only one or two jobs at any one time, so it seem relatively safe for the time being to allocate 4 slots and see how they affect the load average. 3

Of the remaining slots, the following allocation will be made, with the designated person as controlling that group-allocation's usage: 4

RWW: NSW project (software, doc devel, etc.)	8	4a
JAKE: NIC	1	4b
JCN: Facility operation, Valid Applic New-NLS	1	4c
DCE: DCE, Demos, XDOC, whole-ARC overhead, etc.	1	4d
	---	4e
Total ARC slot assignment	11	4f

New basic scheduling and operating rules: 5

Off-Quota logins will be subject to limitation if necessary so that the NSW group can get its share of service (e.g., when 5 programmers would like to use the whole NSW-group "space" to be getting like a third of the machine's capacity); 5a

ELOG quota may also be cut back. 5b

A third "NLOG" (Network-access only) log-in status will be added, with a certain independent quota. This will enable Utility-support staff to access OFFICE-1 through our system; we assume that they put on only a small loading, and we will thus control this kind of log-in use separately from ELOG or OFF-QUOTA access to NLS on our machine. 5c

Computer service for holdover tasks, such as producing reports from our old contracts, finalizing the XDOC catalogs and indices, etc. will be fitted into OFFQUOTA usage, or by special arrangements with the above "slot-holders" for "borrowing" some slot time. 6

We will re-examine the mode in which documentation is developed, and consider where special clerical support and DEX (versions 1, 2, or intermediate 1.5) could enable us to get more mileage from reduced slot capacity. Here, the NSW "secretarial-support" task, and some of ARC's overhead investment in the DEX evolution, may cooperate in the coming months toward more efficiency in developing and updating text files. 7

bug: set external file and jump name extenal commands in te EDITOR.

(J23681) 25-JUL-74 16:08;;;; Title: Author(s): Robert N. Lieberman/RLL; Distribution: /FDBK([ACTION]) KEV([ACTION]) ; Sub-Collections: SRI-ARC; Clerk; RLL;

RLL 25-JUL-74 16:08 23681

bug: set eXternal file and jump name extenal commands in te EDITOR,

Ken : Charles witnessed the events,

bug: set external file and jump name extenal commands in te EDITOR.

the set external file name command in the EDITOR (not the userop subsystem) does not work as is discovered by trying a jump name external command. 1

The Connect to Tty command is a 'ding-a-ling'

(J23682) 25-JUL-74 16:50;;;; Title: Author(s): Jeanne M, Beck/JMB; Distribution: /FDBK([ACTION]) NPG([ACTION]) DIRT([INFO-ONLY]) ; Sub-Collections: SRI-ARC NPG DIRT; Clerk: JMB;

The Connect to Tty command is a 'ding-a-ling'

In writing up the NLS concept of linking, i.e, connecting terminals, and the Connect to Tty command, I find myself describing a feature that's really inscrutable and unusable.

The Connect to Tty command is a 'ding-a-ling'

Does it have to be this way? A SCENARIO: 1

Kirk, in DNLS, wants to link to JMB. Says "Connect to Tty". NLS feeds back "(number)". He has to goto Tenex & do a Where to find the job number. 1a

This is a minor inconvenience. 1a1

Next field makes him choose Input & output or Output only. Then an OK. "(ding-a-ling)" appears in the Tty window. He assumes JMB is refusing links. Goes to Tenex, which tells him JMB did not do a "Receive Connect." He walks to the next room where JMB is at her display. 1b

If JMB happens to be at home or across the country, stop here. 1b1

JMB's screen displays "(ding-a-ling)" in the tty-window. She has assumed it's there because of her normally bad typing. 1c

Sandy Johnson says she assumed that "ding-a-ling" was calling her a dummy. It always reminds me of the TIP's critical attitude, "BAD, CAN'T, etc". Anyway, JMB is not likely to interpret this as a link attempt if she knows she has not set Refuse Links in Tenex. 1c1

KIRK tells JMB she needs to do something with "receive". But that word gets a "?" from NLS. 1d

If the actors here happen not to be documenters, stop here. 1d1

KIRK & JMB read in their documentation something about a command called "Accept Connect (from display)". JMB says, "Yes, that obviously goes with the Connect to Display command for shared-screens; you don't need to do anything to accept a regular link". 1e

Well, JMB & the documentation lie. Any sensible person would just forget it and use Tenex's Link command. If the above is the way NLS's linking is to work it seems to me to be pointless to advertise it. 2

Suggestion: You should not have to do anything special to receive a Connect to Tty, only to refuse it (that's the Tenex principle). Maybe you should have to Accept a Connect to Display (shared-screen), but only if you are given some information from the system that you have to so something. 3

Bug in Jump to name external

(J23683) 25-JUL-74 17:10;;;; Title: Author(s): Kirk E. Kelley/KIRK; Distribution: /BUGS([ACTION]) ; Sub-Collections: SRI-ARC BUGS; Clerk: KIRK;

Bug in Jump to name external

Jump to name external does not search at the address you set in useroptions if the file has a different address set, 1

(J23684) 26-JUL-74 01:15;;;; Title: Author(s): James C. Norton/JCN;
Distribution: /DCE([INFO-ONLY]) RWW([INFO-ONLY]) DLS([
INFO-ONLY]) CKM([INFO-ONLY]) JDH([INFO-ONLY]) MEH([INFO-ONLY
]) JHB([INFO-ONLY]) SRL([INFO-ONLY]) MDK([INFO-ONLY]) WRF(
[INFO-ONLY]) JCP([INFO-ONLY]) RLL([INFO-ONLY]) ;
Sub-Collections: SRI-ARC; Clerk: JCN; Origin: (NORTON,
UTILQMR,NLS;2,), 12-JUL-74 20:42 JCN ; Title: ####;

Quarterly Management Report 1: RADC/ARPA Project 3074 to 18 Apr 74

ARPA Order Number: - Program: - 1

Title: WORKSHOP UTILITY SERVICE FOR RADC AND ARPA 1a

Contractor: Augmentation Research Center
Stanford Research Institute 1b

Date of Contract: 13 December 1973 1c

Amount of Contract: $ 689,039 1d

Contract Number: F-30602-74-C-0076 1e

Principal Investigator: James C, Norton
Phone (415) 326-6200, x2124 1f

Contract Expiration Date: 18 January 1975 1g

I RESEARCH PROGRAM AND PLAN 2

As per our proposal and contract, services are being provided in the following areas: 2a

Access during 16 hours each day, 6 days a week to a Workshop Utility computer service reached by users through the ARPANET, This service is being provided from "OFFICE-1," a PDP-10 TENEX system operated for SRI-ARC by Tymshare, Inc., a commercial supplier. The basic software system is NLS. 2b

Training as appropriate in the use of Display NLS (DNLS), Typewriter NLS (TNLS), and Deferred Execution (DEX) software subsystems, 2c

Technical assistance to RADC and ARPA "Workshop Architects" in the formulation, development, and implementation of augmented knowledge work procedures within offices selected by RADC and ARPA, 2d

Technical assistance to help set up and assist selected ARPA groups, who share a special discipline or mission orientation to use the Workshop Utility Services and to develop procedures, documentation, and methodology for their purposes, 2e

II MAJOR ACCOMPLISHMENTS 3

FACILITY CHECKOUT IN DECEMBER, CONTINUING SERVICE STARTED IN JANUARY 3a

Pager problems delayed the start of service. A replacement pager was supplied by BBN, solving the problem and permitting service to commence 18 January 1974. 3a1

During the first full month of operation (February): uptime was 99.71 % for OFFICE-1. Overall uptime was 99.45 % with net downtime. This is on the 16 hour, 6 day availablity basis. 3a2

During March, uptime was 98.3 % for OFFICE-1. Overall uptime was 97.9% with net downtime. 3a3

During April, uptime was 96.6 % for OFFICE-1. Overall uptime was 96.3% with net downtime. This level resulted from hardware problems experienced mainly during a Saturday in April. 3a4

SUMMARY OF OFFICE-1 USE 3b

Reference set [1] shows use by each Utility Organization, by each user, for each of the first three and 1/2 months of operation. Both CPU usage and connect times are shown. 3b1

REPORTS BY TYMSHARE 3c

Reference set [2] provides STATUS reports furnished by Tymshare. 3c1

Reference set [3] provides UPTIME reports furnished by Tymshare. 3c2

ARCHITECTS' TRAINING COURSE 3d

RADC and ARPA Workshop Architects (D.L.Stone and C.K.McLindon) attended a week-long initial meeting at SRI-ARC with others in that role in January. A second session is planned for Summer '74.
See [4] (21319,1:wyn) 3d1

NETWORK INFORMATION CENTER (NIC) USERS SHIFTED TO OFFICE-1 3e

About 50 site directories are in use at Office-1. with files transferred from SRI-ARC. This transfer process appeared to run smoothly, an important goal in our overall Utility startup plan. 3e1

SUPPORT OF THE DEIS PROJECT UNDER THE ARPA NMRO ENERGY PROGRAM,
See [5] SRI PROPOSAL ISU 74-25 (21447,1:wyn) 3f

We have provided on-line documentation and communication support for two Energy Problem Analysis Centers (EPAC) offices (Menlo Park, Calif, and Arlington, Va,) and the ARPA Nuclear Monitoring Research Office (NMRO), 3f1

INITIAL SEISMIC DATA MANAGEMENT SYSTEM (SDMS) USE PLANNING WITH NMRO AND MIT-LL SEISMIC DISCRIMINATION GROUP,
See [6] SRI PROPOSAL ISU 74-52 (21883,1:gwy) 3g

We have commenced NLS training and applications-use consultation with the staff associated with Seismic Data Management System development at MIT, Lincoln Labs, 3g1

ARPA EXECUTIVE STAFF DIRECTORIES SET UP AT OFFICE-1: 3h

About 30 directories have been set up at Office-1 for ARPA executive staff use, These are being used initially for SNDMSG, RD, READMAIL, TECO backup, In the future we will be working toward gradual introduction of NLS methodology into selected ARPA offices,
See [7] (22371,1:wyn) 3h1

ADDITIONAL PLANNING WITH ARPA STAFF 3i

Planning discussions were held with ARPA staff concerning additional use by people in other ARPA-sponsored programs such as the Computer-Based Instruction and the National Software Works efforts, 3i1

CONSULTATION WITH RADC STAFF ON LARGE DOCUMENT PRODUCTION 3j

We have assisted RADC in the production of the JOVIAL Manual, to be produced by RADC using NLS and COM processes, 3j1

III PROBLEMS ENCOUNTERED 4

None requiring Government action, 4a

IV FISCAL STATUS 5

Estimated expenditures and commitments to date are $290,000 excluding computer and other lease commitments, Estimated funds required to complete the work are $399,039, 5a

The estimated date of completion of work is January 18, 1975, 5b

V ACTION REQUIRED BY THE GOVERNMENT 6

The balance of funding up to contractual limit should be completed soon. 6a

VI NEXT QUARTER PLANS 7

We will continue to provide computer and people support service in an increasingly more effective manner from Office-1 throughout the balance of this initial contract, while expanding the facility and staff to provide service to additional subscriber organizations. 7a

We plan to organize the Utility operations and business planning for smoother day-to-day operation and orderly growth. 7b

We will continue to study system performance to determine what adjustments in the hardware or software should be made for improvement and to get an idea of what to add in the next expansion step. 7c

We will work with the RADC and ARPA Workshop Architects to select offices for gradual introduction to NLS and associated methodology. 7d

We are still developing our training methods and will continue to do so, while training RADC and ARPA-selected users.

7e

VII REFERENCES:

8

1. USE BY EACH UTILITY ORGANIZATION: 1974
January (23558,) February (23559,) March (23560,) April (23561,)

2. STATUS REPORTS FURNISHED BY TYMSHARE: 1974
February (23563,) March (23564,) April (23565,)

3. UPTIME REPORTS FURNISHED BY TYMSHARE: 1974
January (23598,) February (23567,) March (23599,) April (23566,)

4. Architects' Intensive Seminar at ARC
January 10-15, 1974 (21319,)

5. Augmented Knowledge Workshop Support
for the ARPA/SRI DEIS (21447,)

6. Augmented Knowledge Workshop Support
for the ARPA Seismic Data Management System (21883,)

7. Information for New ARPA Users of OFFICE-1 (22371,)

Approved by:

J. C. Norton, Principal Investigator

A demo message to try the journal out

(J23685) 26-JUL-74 02:04; Title: Author(s): Geoffrey S. Goodfellow/GSG; Distribution: /GSG; Sub-Collections: NIC; Clerk: GSG;

GSG 26-JUL-74 02:04 23685

A demo message to try the journal out

This is a demo try to enter a journal message to myself, 1

Pachadermy

(J23686) 26-JUL-74 17:27;;;; Title: Author(s): Dirk H. Van Nouhuys/DVN; Distribution: /JI([INFO-ONLY]) ; Sub-Collections: SRI-ARC; Clerk: DVN;

Pachadermy

Glad to hear the elephants are still active, 1

Query as to what 'ecod' does.

(J23687) 26-JUL-74 18:39; Title: Author(s): Geoffrey S. Goodfellow/GSG; Distribution: /JEW; Sub-Collections: NIC; Clerk: GSG;

Query as to what 'ecod' does.

Hi, I head from Jeff Peters that uyou were the one to query about 'ECOD' that runs under sysjob, and how to collects network data. If you could send me any information on it, it would be appreciated, or I'll just get in touch with you the next time I'm over your way. Geoff 1

NSW software plan for 29-July to 1-October-74

(J23689) 28-JUL-74 16:35;;;; Title: Author(s): Charles H. Irby/CHI; Distribution: /KJM([ACTION]) DSM([ACTION]) KEV([ACTION]) JEW([ACTION]) EKM([ACTION]) DIA([ACTION]) HGL([ACTION]) CHI([ACTION]) RWW([INFO-ONLY]) KIRK([INFO-ONLY]) DVN([INFO-ONLY]) JMB([INFO-ONLY]) JDH([INFO-ONLY]) NDM([INFO-ONLY]) ; Sub-Collections: SRI-ARC; Clerk: CHI; Origin: < NLS, PLAN.NLS;1, >, 28-JUL-74 16:28 CHI ;;;;####;

CHI 28-JUL-74 16:35 23689

NSW software plan for 29-July to 1-October-74

The following is a proposed NSW software plan for the time period from 29-july to 1-oct. It is assumed that Dirk, Kirk, and Jeannie are working madly to get NLS documentation done for the oct-1 new nls at office-1. This file will be maintained as (nls, plan,). People should feel free to make planning notes here. We should attempt to keep it at least two months ahead of us. -- Charles. 1

29-july --> 1-aug 1a

chi, kev, ekm, kjm, dsm : madly edit nls source code to partially implement the frontend-backend split. This involves rearranging code and data declarations into distinct frontend and backend source files. This is principly being done to flush out some of the problems Associated with the split. Files like INTNLS will actually have some code changes to allow for frontend initialization and backend initialization. Generate a new sysgd when you are through. 1a1

JEW: produce a proposed frontend-backend interface with remote procedure call as communication mechanism. This should be reviewed by others and a version should be sent to ISI by 1-aug, along with a list of callable procedures (see below). 1a2

dsm, hgl, ndm: prepare a list of initial backend procedures to be callable from the frontend (and to be used by user programs). This list will be sent to ISI by 1-august with some reasonable documentation of each procedure's function and usage. NDM has a reasonable starting list (intended for user program usage). It is better that the list be too long than too short for the time being. I recommend liberal use of sysgd generator and xref program to help you. 1a3

dia: try to get LP printer stuff squared away. 1a4

hgl: also start working on new help interface. This should be considered a background item along with DEX 1.5. 1a5

1-Aug --> 1-sep 1b

chi, kev, ekm, hgl, dsm, kjm: make as many of the approved changes (DSM will distribute list) as we can to nls. at 15-aug, try to tell documentation and applications people how many of the changes we will really promise by 1-sep. on 1-sep stop work on nls changes for office-1 nls. turn over sources to jdh (he should be familiarizing himself with the source code during this month and we should offer him any assistance we can). nls will actually come up on 1-oct if documentation people are ready. 1b1 jew: build backend-frontend interface routines for one-fork and two-fork configurations. 1b2

dia: develop lp programs for data media, infoton (if they are still in the running), and adm. get demos to play with. make final recomendation by 1-sep. Order should go out asap after recomendation is made (LP's could be ordered sooner once we know how many terminals we need). Also, become familiar with BBN's bcpl/tenex interface package. It may be usable by us with little or no changes (speak to kev about this). Also order NSW pdp-11. 1b3

everyone: start learning bcpl as a background task. Most of us should also become familiar with ELF from a user programming point of view. In addition KEV should be trying to get ELF up on our pdp-11 and learning how to load user progs into a running ELF and how to debug such a program. 1b4

HGL, DSM : finalize list of procedures for user program / frontend usage (please consider the thuroughness of the argument checking that these routines must do). I will try to guarantee that these routines do not change their funchion or calling sequence for a long time so that user programs may be written using these routines and need not be changed because of new NLS releases. 1b5

1-sep --> 1-oct 1c

chi, kev, ekm, hgl, dsm, kjm, jew: make frontend-backend split. This might proceed most expeditiously if we split into two groups: one trying to build a Frontend and one trying to build a Backend. These two groups should try not to share routines or data, but some will be inevitable. These however should be carefully understood and documented. Some code will have to change during this month but we should shoot for at least a running backend (the simpler of the two, in my opinion) by 1-oct for ISI first usage. 1c1

CHI, DIA: begin redesign of CML interpreter and CML language. 1c2

JEW, JP: JEW help JP get acclimated to ARC. Both start writing protocol packages for ELF and TENEX needed for frontend-backend communication. 1c3

kev: help belville get on board (NLS, TENEX, L10, BCPL, ELF). Bob may be very helpful at the ELF support level given his IMLAC experience. 1c4

ECOD

(J23691) 29-JUL-74 15:36;;;; Title: Author(s): James E, (Jim) White/JEW; Distribution: /GSG([INFO-ONLY]) ; Sub-Collections: SRI-ARC; Clerk: JEW;

ECOD

Geoff-- I don't know where you heard about ECOD. It's a little Network server program I wrote about two years ago for Dave Retz of SCRL. "ECOD" stands for "Echo or Discard", and the single-user-at-a-time server process acquires a user via ICP, accepts opcode-length-text commands from him, and either echos or discards the command, depending upon its op code. Don't even remember the details of the protocol, and don't understand why you would be interested in the pgm in the first place. --Jim 1

Modifications Planned to NLS for OFFICE-1 before October 1st,

(J23692) 29-JUL-74 16:15;;;; Title: Author(s): David S. Maynard/DSM; Distribution: /DSM([ACTION]) CHI([ACTION]) KEV([ACTION]) EKM([ACTION]) HGL([ACTION]) KJM([ACTION]) RWW([INFO-ONLY]) KIRK([INFO-ONLY]) JCN([INFO-ONLY]) DVN([INFO-ONLY]) JMB([INFO-ONLY]) JDH([INFO-ONLY]) ; Sub-Collections: SRI-ARC; Clerk: DSM;

Modifications Planned to NLS for OFFICE-1 before October 1st.

The following is a list of the bugs to fix, and the modifications to be made before we bring the new system up at OFFICE-1.The Tasks are ordered in the approximate order in which they will be implemented by the developement staff. The developent staff will implement as many of these as is possible before the OFFICE-1 NLS is frozen on Sept 1st. An accurate estimate of exactly how far down the list we will be by Sept 1st will be published by the development staff by August 15. The priority of each task (first digit of statement name) is a product of the relative importance of the task (middle character of statement name, a=most important, c=least important) and the difficulty of the task (last digit of statement name, 1= least difficult, 3=most difficult). The tasks listed below include all of the tasks in category one and some of the tasks from category two in <GJOURNAL,23653,>. The numbers immediately following the statement name are references to the SID of the associated statement in the aforementioned journal document. 1

(BUGS) 2

(3c1) Fix update to "always" update origin. 2a

(3c1) Fix substitute command to accept a "y" for answer. 2b

(MODS) 3

(1a1) -020- -035- Help instead of or in addition to Goto Help. Make the command Help available from any subsystem (feedback,fdbk,01104) 3a

(1a1) -029- Unavailable alternatives should not appear after question mark. For example, Load Busy file. 3b

(1a1) -030- Change the default setting for ESC in TNLS to be ESC. (feedback,fdbk,02906) 3c

(1a1) -042- The error msg
"Exceed Capacity"
should be changed to
" NLS syst error: string too long "
to eliminate confusion with disk allocation being exceeded. (feedback,fdbk,02765) 3d

(1a1) Make the user dialog better for the "output to terminal" command and add an output to file option which outputs to a sequential file from the output processor. 3e

(2a2) -0101- When viewspecs B and l or g are used the view on the screen and on a printout are different and they should be the same. (feedback,fdbk,03284), (feedback,fdbk,02731) 3f

(2a2) -0102- Numbers on the right look different on an output quickprint than on a display: both output media should have identical formatting appearance (feedback,fdbk,02761) 3g

(2a2) -053- Must have simple DEX available in new nls. 3h

(2a2) -062- Typing ^U before an address was also disliked by many. The preferred prompt would be B:/A:. (feedback,fdbk,02664), (feedback,fdbk,02777), (feedback,fdbk,02790 3i

This will be changed, together with a new set of "DSEL SSEL LSEL" def'ns , namely: 3i1

```
              TNLS           DNLS
   DSEL       A              <bug> / A
   SSEL       A / [T]        <bug> / A / [T]
   LSEL       T / [A]        <bug> / T / [A]
```
3i2

(2a2) -094- Altmode should cause filename recognition for a file in Programs directory without typing programs first when loading a program. (feedback,fdbk,03018) 3j

(2b1) -019- In Print Structure commands, when a link with viewspecs is used as an address, the viewspecs should affect the printout. (feedback,fdbk,03273) 3k

(2b1) -021- New command: "print file <ca>" (obeying default viewspecs, without upsetting current CM and without upsetting current viewspecs): equivalent in concept to print branch zero 3l

(2b1) -022- New command: "print rest <ca>" (constrained as in proposed "print file" command) equivalent to and replacing current "print <ca>" 3m

(2b1) -036- The error msg
  "illegal text entity"
should be changed to
  "invalid ... selection"
where "..." should be the appropriate entity such as text, group, statement 3n

(2b1) -043- When someone tries to load a file which is protected, he should not get the message that the file cannot be opened; he should get the msg "File protected from unauthorized access" 3o

(2b1) -044- It seems inconsistent that in Sendmail you type "SH" to do (Sh)ow Status but anywhere else you would have to type "SHS" for (Sh)ow (S)tatus. (feedback,fdbk,03340) i.e Change "Status" to a Command word in Sendmail and in the Ident subsystems. 3p

(2b1) -052- Have an option in Output Quickprint to put NO heading on any pages except for the string "Page #" at top-right on each page. (feedback,fdbk,03263) i.e. implement option and change CML to be O[utput] Q[uickprint] (OK / N[o headers] (OK / REST) / REST); where REST = C[opies] / F[ile] / A[ppend] ; 3q

(2b1) -055- Default herald length should be 4 instead of 3. (feedback,fdbk,02647) 3r

(2b1) -068- needs further study - Editor subsystem is too big resulting in unnecessary alphabetic conflicts. Suggested replacement: Editor, File-Handler, and Terminal Handler. See (kudlick,newsubs,1:why) for a preliminary description; a more up-to-date description is forthcoming. Development will study this proposal and make a recommendation. 3s

(2b1) -072- Expert-expert should not be the default recognition for new users. (feedback,fdbk,02714) 3t

TNLS = Demand
DNLS = Exp/Exp 3t1

This is up to Applications. Development needs a final decision before Sept. 1st. 3t2

(2b1) Change the CML to replace the entities "window" and "boundary" by the single entity "edge". i.e. S[plit window] H[orizontally] will become I[nsert] E[dge] etc. 3u

(2b1) change the SENDMAIL Command DONE to "SEND (the mail)" 3v

(2b1) change the SENDMAIL Commands SEND FOR ACTION and SEND for INFO to D[istribute] A[ction (copies to)] and D[istribute] I[nformation (copies to)] 3w

(2b1) fix the bug in sendmail which will not let just specify a <ca> for a null distributiion list. 3x

This will either be implemented by: 3x1

1) adding a type LIT to the CML which would require typein (no bugging allowed) or 3x1a

2)specifing a special character (ESC?) which means null if it is the first character of a typein. 3x1b

(3a3) -0115- Userprograms are too difficult for users to invoke. Why not have commonly used ones such as inmes and letter instituted as a regular subsystem and simply say Goto subsystem message or whatever. (feedback,fdbk,03335) Other suggestions: 3y

One "load program" command should accomplish all of the following events: 3y1

set the buffer size appropriate to the program being loaded,
load the program,
OPTIONALLY run it 3y1a

In addition, the "Run Program" command should still be there. 3y2

At the least, the Load Program command should automatically set Buffer Size. (feedback,fdbk,03261) 3y3

Recommendation: Make G[o to subsystem] M[essage] load the appropiate subsystem (increasing buffer size if possible) and then do the GOTO. 3y4

(3a3) -014- Repeating prompts is unnecessary in TNLS: cleanup TNLS echoing. i.e. Update C: File OK:/C C: Compact OK:, the two C's are not necessary. (feedback,fdbk,02828) 3z

(3a3) -027- Valid alternatives should be available in Help exactly as stated in the response to questionmark, especially such things as <tab>, <insert>, etc., as well as ANY other response. (feedback,fdbk,03191) 3a@

(We acknowledge that this may require changes to the Help/Query command recognition algorithm.) 3a@1

(3c1) -0103- Let words containing hyphens break at the hyphen when it is at the end of a line. (feedback,fdbk,02775) 3aa

(3c1) -012- Change "Show File Marker" to be "Show Marker" 3ab

(3c1) -035- Make the Jump commands available from any subsystem 3ac

(3c1) -039- [dsm] Setting name delimiters should change the Statement Signature. (feedback,fdbk,02717) 3ad

(3c1) -041- Since a person is told a file is bad, he should also be told it is good. "file verify in progress" is an o.k. message, akin to "output quickprint in progress". (feedback,fdbk,02764) 3ae

(3c1) -047- <tab> is listed as an alternative in response to questionmark, but if not typed in the proper context it's responded to by "Illegal Search Type". (feedback,fdbk,02651) This err msg should be changed to "<tab> valid only to repeat a previous search" 3af

(3c1) -050- The validity of the characters used for name delimiters should be checked as they are typed in. (feedback,fdbk,02718) 3ag

(3c1) -056- The herald should be settable as an option to zero length leaving just a prompt. (feedback,fdbk,02907) 3ah

(3c1) -058- Put the "Process Commands Branch" command in the Editor subsystem. 3ai

(3c1) -092- Allow viewspec "o" and "p" to be set before the completion of freeze and release commands and let "release all" OPTIONALLY result in viewspec "p". (feedback,fdbk,03249) 3aj

(4b2) -0105- Show Return Stack (feedback,fdbk,03390), (feedback,fdbk,03159) 3ak

(4b2) -015- The space is not echoed to TNLS users when second-level commands are typed in expert-expert. Our suggestion is to echo it. (feedback,fdbk,02711), (feedback,fdbk,02712) 3al

(4b2) -016- When using Jump to Link command with viewspecs specified in the link, let viewspecs be manually set to SUPPLEMENT those in the link. (feedback,fdbk,03271) 3am

for example, "jump (to) link SELECTION VIEWSPECS <confirm>" 3am1

Moreover, implement this capability in ALL jump commands. 3am2

i.e. implement this in the J[ump to] R[eturn] and J[ump to] F[ile] R[eturn] commands. 3am3

(4b2) -034- All DNLS "jump" commands should be also available as commands in TNLS. 3an

(4b2) Make a new viewspec which would turn all indenting off putting all text left justified regardless of structure. 3ao

(6b3) -069- CONFIRM should work for recognition so that <sp> or <esc> is not required when an entire command has been typed. (feedback,fdbk,02709) i.e. have CA as a right delimiter but not swallowed by the CML. 3ap

(6c2) -0111- Since ;filter; is defined as a viewspec in a link it should be a valid viewspec whenever the prompt V: appears. (feedback,fdbk,02741) 3aq

(6c2) -059- Resolve the present bug in newnls that makes it impossible to "jump to name" in the identfile when the name is enclosed in single quotes. i.e. implement by removing the first single quote ' from the last names in the ident file. 3ar

(6c2) Review TNLS CALCULATOR and DNLS CALCULATOR 3as

(6c3) -023- Rather than having left-over prompts at the top of the screen, such as "Replace Text at through by through", display the actual text typed, following and on the same line as the respective prompts. Need an appropriate symbol for a bug mark (possibly the word or character if text; the statement number, if structure; or a symbol such as "<bug>"). This would more closely approximate TNLS. (feedback,fdbk,01927), (feedback,fdbk,03236) Development feels that the current implementation of noisewords () is OK, we may display "<bug>" for bug selections. 3at

(9c3) -065- After typing a space and one character, a backspace character should result in your being able to type another second-level command. (feedback,fdbk,03151) 3au

(9c3) -070- Have a new user option to set the escape charater (and its echo) to be other than "<sp>". 3av

(DONE) 4

(1) -012- Have "Set Case" be as follows instead of it's present wording: 4a

Force-case Character
Force-case Text
...
Force-case Mode

4a1

(1) -025- In response to question-mark, alphabetize the alternatives. (feedback,fdbk,03189) 4b

(1) -037- Change "record session" to "start recording-session" to correspond to "stop recording-session". (feedback,fdbk,03258) 4c

(1) -048- We'd also like to see <insert> and <repeat> listed as alternatives at top level. (Most users will type <control-e> and <control-b> respectively, but these are user-settable options. Therefore use of the global terms insert and repeat is preferred. But this also means that the HELP system data base must respons to "show <repeat>" etc. 4d

(1) -049- Other control characters should echo as it's done now

(e,g,, <lf> echoes via its action, not via a visible), This replaces the recommendation given in (feedback,fdbk,02772) 4e

(1) -051- In TNLS, the Show Feedback command should show the feedback mode as well as length and indenting, (feedback,fdbk,03156) 4f

(1) -057- L: field shoul be optionally terminate-able by <ca> 4g

(1) -060: when searching with a Content Analyzer Pattern, why not have the compile command automatically institute the program, (feedback,fdbk,03286) 4h

It does this now, 4h1

(1) -067- "jump to ,fr" and "jump to ,r" need not be changed to pause and give stack information to TNLS users, but they certainly shouldn't go away, 4i

(1) -071- - no this will not be done - When multiple characters or words are allowed, the noiseword should be Character(s) or Word(s) etc, (feedback,fdbk,02900) 4j

(1) -074- Discuss the following commands, choose simpler wording 4k

Split window Horizontally,
Split window vertically
Simulate Terminal-type 4k1

One suggestion is: 4k2

split-window horizontally
split-window vertically
terminal-type 4k3

An alternate suggestion (by NDM) for simulating terminal type is "set terminal-type" 4k4

An alternate suggestion (by NDM) for the window commands is to do away with the "split" verb and "window" noun, and instead use something like 4k5

Insert Boundary (Horizontally / Vertically) at (BUG / Center of BUG) using window BUG CONFIRM
[note this still offers the split at center as an option] 4k5a

Move Boundary from BUG to BUG CONFIRM 4k5b

Delete Boundary at BUG keeping view at BUG CONFIRM 4k5c

Supporting arguments for NDM's suggestion are in (Gjournal,30932,1:why) 4k5d

(2) -026- When the first or second command word is a phrase, list the entire phrase as an alternative in response to question-mark. For example, (d)elete (a)ll markers, preferably hyphenated. (feedback,fdbk,03248) See (Gjournal,23646,1:why) for suggestions by Applications. 4l

(2) -031- Have Control-O stop printing alternatives after a question mark has been typed. (feedback,fdbk,02796) 4m

(2) -038- In TNLS a space should be fedback after a prompt and a CR before a herald and prompt. (feedback,fdbk,02648) (We know of no case where this isn't done already.) 4n

(2) -054- Fewer colons would look better, i.e. A/[T]: rather than A:/[T]:. (feedback,fdbk,02649) 4o