

January 1975

```
# HALF DUPLEX
# HELP
# HOSTAT
# INITIATE
# LEAVE
# LINK
# LIST
# LOCATE
# LOGOUT
# NEED
# NETNEWS
# NETSTAT
# PROEDIT
# PROLIST
# PURGE
# QFD
# QUIT
# RECEIVE
# REFUSE
# RELEASE
# RENAME
# RESET
# RUN
# SERVERS
# SITES
# SNDMSG
# START
# TELCONN
# TENXSTAT
# TERMINATE
# TIMECONSTANT
# TRSTAT
# TYPE
# UNDELETE
# USE
# WHERE
# WHO
#
# _DESCRIBE (command, term or ALL) RSEEXEC
#
# RSEEXEC
#
# The Resource Sharing Executive is an evolutionary multi-computer
# executive program. It provides an environment in which the range
# of many features found on a single-Host time sharing system are
# extended beyond the boundaries of a single Host to encompass many
# Hosts on the ARPANET.
#
# At present RSEEXEC includes facilities for inter-Host user-user
# interaction (see descriptions for WHO, WHERE, SITES, LINK,
# SNDMSG), for managing "multi-Host" file directories (see
```

January 1975

```
# descriptions of ENTER and BIND) and for controlling multiple
# "jobs" on several Hosts (see descriptions for TRANSACTION and
# INITIATE). In addition, the RSEEXEC serves as a command language
# interpreter for TIP users.
#
# The DESCRIBE command can be used to obtain descriptions of all
# (accessible) RSEEXEC commands and, in addition, the following
# terms:
#
#     BOUND-DEVICE, COMPONENT-DIRECTORY, COMPOSITE-DIRECTORY,
#     FILE-NAMES, INTERRUPT-CHARACTERS, MULTI-IMAGE-FILES,
#     PRIMARY-DIRECTORY, PROFILE, TRANSACTION, BUGCHK
#
# (TIP users accessing RSEEXEC via the TIP "@n" command can use only
# a subset of the RSEEXEC commands; they can obtain descriptions of
# only those commands (and related terms) they have access to.)
#
# The user interested in the design philosophy of RSEEXEC and its
# implementation is referred to the paper "A Resource Sharing
# Executive for the ARPANET", Proceedings of 1973 National Computer
# Conference and Exposition, also NIC #14689).
#
# _DESCRIBE (command, term or ALL) ALL
#
# ACQUIRE (Component Directory) Component1 ... Componentn <cr>
# Use of this command is limited to users who have gained access to
# the file system features of RSEEXEC via the ENTER command. Used
# to add the files in the component directories specified to the
# composite directory. A directory need not be ACQUIRED in order
# to reference files in it or at the corresponding site. However,
# file name recognition and completion will work only for files
# that are local or in ACQUIRED directories. See also descriptions
# for COMPOSITE-DIRECTORY and COMPONENT-DIRECTORY.
#
# APPEND (file) FILE1 (to file) FILE2 <cr>
# Use of this command is limited to users who have gained access to
# the file system features of RSEEXEC via the ENTER command.
# Changes FILE1 by appending FILE2 to it.
#
#     BIND (device) DEVICE-NAME (to site) SITE-NAME <cr>
# or
#     BIND (device) DEVICE-NAME (to site) TIP-NAME <cr>
#         (Port #) number <cr>
# Use of this command is limited to users who have gained access to
# the file system features of RSEEXEC via the ENTER command.
#
# Associates the device named with the Host named such that
# subsequent references to the device name refer to the device at
# the Host specified. For example, the sequence:
#
#     _BIND LPT MITRE-TIP
```

January 1975

```
#      (Port #) 1
#      _LIST TEST.DATA
#
# would produce a listing of file FOO.DATA at device port 1 on the
# MITRE-TIP. For binding to a TIP Port to work properly the TIP
# port in question must be set to "wild".
#
# BOUND-DEVICE
# The user can use the BIND command to specify that subsequent use
# of a particular device name is to refer to that device at a
# specific site. Such a device is said to be "bound" to that site.
# For example, the sequence of commands:
#
#      _BIND LPT USC-ISI <cr>
#      _COPY REPORT.DRAFT LPT: <cr>
#      _LIST PROGRAM.SOURCE <cr>
#
# first binds the line printer to ISI and then causes two listings
# to be produced by the ISI line printer.
#
# BREAK<cr> Breaks terminal links (see LINK).
#
# BUGCHK
# RSEEXEC contains a considerable number of internal consistency and
# redundancy checks. If RSEEXEC detects a malfunction it prints the
# message:
#
#      _BUGCHK at NNNNN
#
# and continues. Such messages are useful for debugging purposes
# and recurring BUGCHK messages, together with the circumstances
# under which they occur, should be reported to Bob Thomas
# (BTHOMAS@BBN) or Paul Johnson (JOHNSON@BBN).
#
# COMPONENT-DIRECTORY One of the file directories that may be
# included in the user's composite directory. The syntax for
# component directories is:
#
#      [Site]<Directory>
#
# e.g., [NIC]<JONES>. There is an entry in the user's profile for
# each component directory. The user may control which component
# directories are, at any given time, included in his composite
# directory via the ACQUIRE and RELEASE commands and the
# subcommands of the ENTER command.
#
# COMPOSITE-DIRECTORY
# The collection of file directories specified in a user's profile
# define his composite directory. The "contents" of the composite
# directory are the union of the "contents" of the component
# directories specified in the profile. Pathnames without site and
```

```
# directory qualification are interpreted with respect to the
# user's composite directory. The ENTER command uses information
# in the profile to gather sufficient information to construct (a
# local copy) the user's composite directory. See also
# descriptions for PROFILE and FILE-NAMES.
#
# CONTINUE <cr>
# Resumes execution interrupted by previous ^C.
#
# COPY (file) FILE1 (to new file) FILE2 <cr>
# Use of this command is limited to users who have gained access to
# the file system features of RSEEXEC via the ENTER command. Makes
# a copy of FILE1 which is named FILE2.
#
# DELETE (file) FILE <cr>
# or
# DELETE (file) FILE1 ... FILEn <cr>
# Use of this command is limited to users who have gained access to
# the file system features of RSEEXEC via the ENTER command.
# Deletes the file(s) specified. Files which have been deleted but
# not expunged may be "undeleted" by the UNDELETE command. Deleted
# files are automatically expunged at LOGOUT.
#
# DESCRIBE (command, term or ALL) command<cr>
# or
# DESCRIBE (command, term or ALL) ALL<cr>
# Describes any (or all) command(s). In addition, DESCRIBE
# can be used to describe certain "terms" such as RSEEXEC.
#
# DEVSTAT <cr>
# Lists the (binding) status of all devices.
#
# DIRECTORY <cr>
# or
# DIRECTORY , <cr>
# _subcommand <cr>
# .
# .
# .
# <cr>
# or
# DIRECTORY FILE1 ... FILEn <cr>
# Use of this command is limited to users who have gained access to
# the file system features of RSEEXEC via the ENTER command. Prints
# information (as modified by subcommands, if any) about the
# file(s) specified or, if none are specified, all files in the
# user's composite directory. The subcommands are:
#
#     VERBOSE <cr>
#     --MULTI-IMAGE FILES <cr>
#     --SITES sitel ... siten <cr>
```

January 1975

```
#      _DELETED-FILES <cr>
#
# See also descriptions of COMPOSITE-DIRECTORY and
# MULTI-IMAGE-FILES
#
#   ENTER (name) NAME (affiliation) AFFL (RSEEXEC password) PWRD <cr>
# or
#   ENTER (name) NAME (affiliation) AFFL (RSEEXEC password) PWRD , <cr>
#   _subcommand <cr>
#
#   .
#
#   .
#
#   .
#
#   <cr>
#
# Grants access to distributed file system features of RSEEXEC after
# constructing a composite directory for the user from his profile.
# If the user does not have a permanent profile (e.g., hasn't
# previously used the ENTER command or has chosen not to have
# RSEEXEC maintain a permanent profile for him) RSEEXEC will acquire
# the information necessary to construct a profile for him. NAME
# is the name the user has chosen to be known by to RSEEXEC; AFFL
# is his affiliation (e.g., AMES, NIC, ARPA - at present an
# arbitrary string); PWRD is his RSEEXEC password chosen when his
# permanent profile was created. W A R N I N G : RSEEXEC
# distinguishes between upper and lower case letters in the RSEEXEC
# password. The user may use the ENTER subcommands to control
# which components of his profile are acquired for his composite
# directory. The subcommands are:
#
#   ALL all components are acquired
#   NONE no components are acquired
#   ACQUIRE COMPONENT1 ... COMPONENTn
#           components 1 through n are acquired
#
# See also the descriptions for PROFILE, COMPONENT-DIRECTORY and
# COMPOSITE-DIRECTORY.
#
# ESCAPE (Character is) CNTL-CHAR <cr>
# Sets the "return from transaction escape character" to the
# control character specified. The escape character is initially
# ^Z. See also the descriptions for INTERRUPT-CHARACTERS and USE.
#
# EXEC<cr>
# Runs the standard TENEX EXEC; to return to RSEEXEC use the EXEC
# QUIT command. If he has previously ENTERed the user has the
# option of reacquiring the local component(s) of his composite
# directory when he returns to RSEEXEC from an inferior EXEC. This
# is useful if he has added or deleted files while using the EXEC.
#
# EXPUNGE (deleted files) <cr>
# Irretrievably removes deleted files from the user's composite
# directory.
```

January 1975

```
# FILE-NAMES
# The RSEEXEC extends the syntax for TENEX file names to include a
# Host component. The syntax for file pathnames is:
#
# [HOST]DEVICE:<DIRECTORY>NAME.EXTENSION;VERSION
#
# Where HOST is either the string "LOCAL" or the name of an ARPANET
# TENEX. Partial pathnames may be used within RSEEXEC. For
# example, whenever the site, device and directory fields are
# omitted, the user's composite directory is used as a default. At
# present the TENEX "*" convention may be used only with local
# files. The user must have a profile entry for a site before he
# can access files at that site. A Component Directory for a site
# need not be ACQUIRED in order to reference files at the site or
# in the Component Directory. However, file name recognition and
# completion will work only for files that are local or in an
# ACQUIRED directory. See description for PROFILE.
#
# FILE-TRANSFER-EXAMPLES
# The NEED command is a convenient way to move files from one or
# more Hosts to a specified destination Host: (assume in the
# following that the local Host is BBNA): To move a group of files
# to the local primary directory:
#
# _NEED (files) F1 F2 ... FN <cr>
# e.g.,
# _NEED (files) [ISI]<SUBSYS>NETSTAT.SAV [BBNB]<TENEX-132>SCHED.MAC
# [ISI]<SUBSYS>NETSTAT.SAV
# [BBNB]<TENEX-132>SCHED.MAC
#
# To move a group of files to a specified directory at the local Host:
# _NEED (files) F1 F2 ... FN <cr>
# -(in Component Directory) CD <cr>
# e.g.,
# _NEED (files) [ISI]<SUBSYS>NETSTAT.SAV [BBNB]<TENEX-132>SCHED.MAC ,
# NE=-(in Component Directory) [BBNA]<TENEX-132>
# [ISI]<SUBSYS.NETSTAT.SAV
# [BBNB]<TENEX-132>SCHED.MAC
#
# To move a group of files to another Host:
# _NEED (files) F1 F2 ... FN , <cr>
# -(in Component Directory) CD <cr>
# e.g.,
# _NEED (files) [ISI]<SUBSYS>NETSTAT.SAV [BBNB]<TENEX-132>SCHED.MAC ,
# -(in Component Directory) [ARC]<JONES>
# [ISI]<SUBSYS>NETSTAT.SAV
# [BBNB]<TENEX-132>SCHED.MAC
#
# Note that component directories need not be "ACQUIRED" to move to
# or from them. However, file name recognition and completion will
# only work for files that are local or in ACQUIRED directories.
```

January 1975

```
# FULLDUPLEX<cr>
# Causes your terminal to be treated as fullduplex.
#
#   GET (Saved File) FILE <cr>
# *****fix description*****
# *****Not Implemented*****
#
#   HALF DUPLEX<cr>
# Causes your terminal to be treated as halfduplex.
#
#   HELP<cr>
# Prints a short help message.
#
#   HOSTAT<cr>
# Lists the status of network server hosts as maintained by the
# host survey program at MIT-DMCG.
#
#   INITIATE (transaction at) HOST-NAME (called) NAME <cr>
# Attempts to create a job for the user at the site specified. The
# job is known as NAME. The user will be notified when the
# transaction is ready for use. See also the descriptions for the
# USE, TERMINATE, TRSTAT and PURGE commands.
#
#   INTERRUPT-CHARACTERS
# The following characters are handled as terminal interrupts
# by RSEEXEC:
#
#   ^C (CRTL-C): interrupts the current activity, returning control
#                  to RSEEXEC. The CONTINUE command may be used to
#                  resume the interrupted activity. When a transaction
#                  is being USED, RSEEXEC transmits the ^C to the
#                  remote transaction.
#
#   ^T (CRTL-T): prints CPU and console time used in RSEEXEC
#                  session. When a transaction is being USED,
#                  RSEEXEC transmits the ^T to the remote transaction.
#
#   ^Z (CRTL-Z): enabled only when a transaction is being USED.
#                  Returns control from transaction to RSEEXEC.
#                  The ESCAPE command may be used to change the
#                  transaction escape character from ^Z to
#                  another control character.
#
#   ^P (CRTL-P): RSEEXEC "panic" escape. Intended for use when your
#                  terminal becomes "hung". It breaks all terminal
#                  links, clears terminal input and output buffers,
#                  and returns control to the top level EXEC. The
#                  EXEC CONTINUE command may be used to resume the
#                  RSEEXEC session. When resumed in this way the
#                  RSEEXEC acts as if the user had typed ^C.
```

January 1975

```
# LEAVE (distributed file system) <cr>
# Use of this command is limited to users who have gained access to
# the file system features of RSEEXEC via the ENTER command. Makes
# the distributed file system features of the RSEEXEC inaccessible.
# Inverse of ENTER.
#
# LINK (to tty #) number (at site) hostname<cr>
# or
# LINK (to tty #)<cr>
# "Links" your terminal to the terminal specified at the host
# specified such that the output for either terminal appears on
# both. If no hostname is given the local host is assumed and a
# local link will be made. The RSEEXEC comment character ";" should
# be used when LINKed to prevent RSEEXEC from interpreting dialogue
# as commands.
#
# Links are broken by the BREAK command or by quitting RSEEXEC.
#
# LIST (file) FILE <cr>
# or
# LIST (file) FILE1 ... FILEn <cr>
# Use of this command is limited to users who have gained access to
# the file system features of RSEEXEC via the ENTER command. Causes
# a listing(s) of the specified file(s) to be output to the line
# printer. The command:
#     COPY (file) FILE (to new file) LPT: <cr>
# will produce a listing without the formatting action of the
# LIST command.
#
# LOCATE (file) FILE (at Component Directory) COMPONENT-DIR <cr>
# Use of this command is limited to users who have gained access to
# the file system features of RSEEXEC via the ENTER command.
# Creates an image of FILE and places it in the Component Directory
# specified. See also the description for MULTI-IMAGE-FILE.
#
# LOGOUT<cr>
# Logs out from RSEEXEC and TENEX.
#
# MULTI-IMAGE-FILES
# The RSEEXEC treats files with the same pathname relative to a
# user's composite directory (i.e., identical name, extension and
# version components) as "images" of the same file. Such a file is
# said to be a multi-image file. Although the profile file (see
# description of USER PROFILE) is transparent to the RSEEXEC user,
# it is implemented as a multi-image file.
#
# NEED (file) FILE1 ... FILEn <cr>
#
# or
# NEED (file) FILE1 ... FILEn , <cr>
#   (in Component Directory) CD <cr>
```

January 1975

```
# Use of this command is limited to users who have gained access to
# the file system features of RSEEXEC via the ENTER command.
# Creates an image(s) of the file(s) specified in the Component
# Directory specified. If no Component Directory is specified, the
# image(s) is placed in the local primary directory. See also the
# description of MULTI-IMAGE-FILE.
#
# NETNEWS<cr>
# Prints the latest network news.
#
# NETSTAT<cr>
# Runs the standard TENEX NETSTAT subsystem which gives network
# status information.
#
# PRIMARY-DIRECTORY
# For each site for which there are Component Directories there is
# a Component Directory designated the Primary Component Directory
# for that site. The Primary Directory for a site must be a
# "login" (rather than "files only") directory. It is used as the
# basis for access control checks at the site. The Primary
# Directory for a site is set by the user either implicitly (as the
# first "login" directory for the site added to his profile) or
# explicitly (via the PRIMARY command of the profile editor). See
# also descriptions for PROFILE, PROEDIT and COMPONENT- DIRECTORY.
#
# PROEDIT <cr>
# Use of this command is limited to users who have gained access to
# the file system features of RSEEXEC via the ENTER command. Used
# to edit the user profile. PROEDIT subcommands are:
#
#      ADD      to add an entry to the profile
#      REMOVE   to remove an entry from the profile
#      LIST     to print the profile
#      CHANGE   to modify an existing profile entry
#      PRIMARY  to change the primary directory for a site
#      UPDATE   to make the edits permanent (see below also)
#      QUIT     to return to the RSEEXEC
#
# The syntax for a profile entry is:
#
#      [Site]<Directory>
#
# If successful, the ADD (REMOVE) command results in modification
# of the user's profile with the addition (removal) of the
# specified entry. In addition, if he has so chosen, his composite
# directory is modified by the addition (removal) of the
# appropriate files. If the user attempts to ADD an entry for a
# site for which the RSEEXEC server program is down, the entry will
# be marked to indicate that it has not been verified (i.e.
# name/password not checked) and that its files have not been added
# to the user's composite directory.
```

January 1975

```
# If during the course of the user's RSEEXEC session the remote
# server comes up, the entry will automatically be verified and its
# files added to the users composite directory (if the user has
# indicated that he wants them). The user has the option of making
# his edits permanent via the UPDATE command or when he leaves the
# distributed file system environment via the LEAVE, QUIT or LOGOUT
# command. The profile editor prompt character is "*". See also
# the descriptions for PROFILE, COMPONENT-DIRECTORY, and
# COMPOSITE-DIRECTORY and PRIMARY-DIRECTORY.
#
# PROFILE
# A collection of user specific information and parameters
# maintained for the user by the RSEEXEC. At present, the
# information maintained includes an entry for each of the user's
# file directories: each entry consisting of Host name, directory
# name, password and account number or string. The profile editor
# (PROEDIT) can be used to add or delete entries from the profile.
# If a user chooses to have the RSEEXEC maintain a permanent record
# of his profile a file named:
#
#       ]-RSPRF-[ .NAME@AFFILIATION;1
#
# will be maintained in each directory named in the profile. This
# file is itself transparent to the RSEEXEC user. Images of the
# profile file are suitably protected: only the user himself may
# read or write it (its protection attribute is P770000); the
# passwords stored in it are encrypted (using the user's RSEEXEC
# password as a key). The QUIT, LEAVE and LOGOUT commands ask the
# user if he wishes to have a permanent profile.
#
# PROLIST <cr>
# Use of this command is limited to users who have gained access to
# the file system features of RSEEXEC via the ENTER command.
#
# Prints the contents of the user's profile.
#
# PURGE (transaction) NAME <cr>
# break Causes forced termination of a previously INITIATED job or
# TELCONN connection by breaking network connection with remote
# site. Intended for use only when TERMINATE fails. See also
# descriptions for INITIATE, TELCONN, USE, TERMINATE, and TRSTAT.
#
#   QFD (file) FILE <cr>
# or
#   QFD (file) FILE1 ... FILEn <cr>
# Use of this command is limited to users who have gained access to
# the file system features of RSEEXEC via the ENTER command.
#
# Prints a "quick" description of the file(s) specified.
#
# QUIT<cr>
```

January 1975

```
# Ends RSEXEC session.  
#  
# RECEIVE (links)<cr>  
# Sets terminal to accept links (default state).  
# Undoes a previous REFUSE command.  
#  
# REFUSE (links)<cr>  
# Sets terminal to refuse links.  
# Undone by a subsequent RECEIVE command.  
#  
# RELEASE (Component Directory) Component1 ... Componentn <cr>  
# Use of this command is limited to users who have gained access to  
# the file system features of RSEXEC via the ENTER command. Used  
# to remove from the composite directory the files in the specified  
# component directories.  
# See also the descriptions for COMPOSITE-DIRECTORY and  
# COMPONENT-DIRECTORY.  
#  
# RENAME (file) FILE1 (to be) FILE2 <cr>  
# Use of this command is limited to users who have gained access to  
# the file system features of RSEXEC via the ENTER command.  
# Changes the name of FILE1 to be FILE2.  
#  
# RESET <cr>  
# Similar to the RESET command of the TENEX EXEC.  
#  
# RSEXEC  
# The Resource Sharing Executive is an evolutionary multi-computer  
# executive program. It provides an environment in which the range  
# of many features found on a single-Host time sharing system are  
# extended beyond the boundaries of a single Host to encompass many  
# Hosts on the ARPANET.  
#  
# At present RSEXEC includes facilities for inter-Host user-user  
# interaction (see descriptions for WHO, WHERE, SITES, LINK,  
# SNDMSG), for managing "multi-Host" file directories (see  
# descriptions of ENTER and BIND) and for controlling multiple  
# "jobs" on several Hosts (see descriptions for TRANSACTION and  
# INITIATE). In addition, the RSEXEC serves as a command language  
# interpreter for TIP users.  
#  
# The DESCRIBE command can be used to obtain descriptions of all  
# (accessible) RSEXEC commands and, in addition, the following  
# terms:  
#  
# BOUND-DEVICE, COMPONENT-DIRECTORY, COMPOSITE-DIRECTORY,  
# FILE-NAMES, INTERRUPT-CHARACTERS, MULTI-IMAGE-FILES,  
# PRIMARY-DIRECTORY, PROFILE, TRANSACTION, BUGCHK  
#  
# (TIP users accessing RSEXEC via the TIP "@n" command can use only  
# a subset of the RSEXEC commands; they can obtain descriptions of
```

January 1975

```
# only those commands (and related terms) they have access to.)  
#  
# The user interested in the design philosophy of RSEEXEC and its  
# implementation is referred to the paper "A Resource Sharing  
# Executive for the ARPANET", Proceedings of 1973 National Computer  
# Conference and Exposition, also NIC #14689).  
#  
# RUN (Saved File) FILE <cr>  
# *****fix description*****  
# *****Not Implemented*****  
#  
# SERVERS<cr>  
# Prints a list of the sites which (at times) run RSEEXEC servers.  
# These sites must both be up and running the server to be  
# accessible from RSEEXEC.  
#  
# SITES (of user) username<cr>  
# Lists the sites (with RSEEXEC servers running) at which the  
# specified user is known.  
#  
# SNDMSG<cr>  
# Runs a subsystem for sending messages to other network users.  
# Messages can be delivered only if the destination site runs an  
# FTP server with the MAIL command implemented. Undelivered  
# messages will be deleted after a week.  
#  
# START <cr>  
# *****fix description*****  
# *****Not Implemented*****  
#  
# TELCONN (to site) HOST-NAME (with a connection called) NAME<cr>  
# or  
# TELCONN (to site) HOST-NAME (with a connection called) NAME,<cr>  
# _option <cr>  
# - .  
# - :  
# - .  
# - <cr>  
# Attempts to establish a TELNET connection to the specified HOST.  
# If successful, the user's terminal is connected to this network  
# communication path. To return to RSEEXEC type ^Z (CNTL Z). Use  
# of the connection may be resumed with the USE command. Default  
# socket is the logger (#1). An alternate socket, as well as  
# desired echo mode and terminal characteristics may be specified  
# in the second form of the command. Type ? in response to the  
# option prompt to list available options. See also the  
# descriptions of the USE and INITIATE commands.  
#  
# TENXSTAT<cr>  
# Prints status information for TENEX sites with RSEEXEC servers  
# running.
```

January 1975

```
# TERMINATE (transaction) NAME <cr>
# Terminates a previously INITIATED job by sending it several ^C's
# and then logging it out. Also can be used to terminate a TELCONN
# connection. After termination, a TELCONN connection can no
# longer be USED.

#
# TIMECONSTANT (for net connections is) value<cr>
# Sets the time constant used for interactions with non-local
# RSEEXEC server programs. If the remote server does not respond
# within the specified time the interaction is aborted. Possible
# values are: RAPID (8 sec.), MODERATE (15 sec.), LETHARGIC (40
# sec.), and INFINITE (2 min.). The time constant is initially
# MODERATE (15 sec.).

#
# TRANSACTION
# A user can instruct the RSEEXEC to create a job for him at another
# site. Such jobs are called transactions. See descriptions of
# the INITIATE, TELCONN, USE, TERMINATE and PURGE commands.

#
# TRSTAT <cr>
# Prints status of previously INITIATED jobs and TELCONN
# connections. Possible status are:
#
# PENDING           INITIATED but login incomplete
# USEABLE            can be used via USE command
# USEABLE TELNET     TELCONN connection, can be used via USE
#                     command
# TERMINATION PENDING TERMINATED but logout incomplete
# TERMINATED         TERMINATED but not yet removed from
#                     RSEEXEC's transaction table

#
# TYPE (file) FILE <cr>
# or
# TYPE (file) FILE1 ... FILEn <cr>
# Use of this command is limited to users who have gained access to
# the file system features of RSEEXEC via the ENTER command. Prints
# the file(s) specified on the user's terminal. Identical to LIST
# command except for the use of the terminal. The command:
#   COPY (file) FILE (to new file) TTY: <cr>
# will copy a file to the user's terminal without the formatting
# action of the TYPE command.

#
# UNDELETE (file) FILE <cr>
# or
# UNDELETE (file) FILE1 ... FILEn <cr>
# Use of this command is limited to users who have gained access to
# the file system features of RSEEXEC via the ENTER command.
# Revives the DELETED file(s) specified by restoring the file(s) to
# normal status; e.g., they are once again accessible to RSEEXEC
# commands. The inverse of DELETE.
```

January 1975

```
# USE (transaction) NAME <cr>
# or
# USE (transaction) NAME, <cr>
# _option <cr>
#   .
#   .
#   .
# _<cr>
#
# Connects the user's terminal to a previously INITIATED job or to
# a previously established TELCONN connection. To return to RSEEXEC
# type ^Z (CNTL-Z); to transmit ^Z to the job type the two
# character sequence <^><Z>. In general typing the two character
# sequence <^><x> will transmit x if x is a non-alphabetic
# character and CNTL-x if s is an alphabetic.
#
# Thus to transmit ^P (the RSEEXEC "panic" escape) type <^><P>. If
# the user has ENTERED and uses ^Z to return to RSEEXEC from a
# transaction, he has the option of updating his composite
# directory to reflect any additions or deletions resulting from
# his USE of the transaction. The ESCAPE command may be used to
# change the transaction escape character from ^Z.
#
# If the second form of the command is used options concerning the
# network connection (echo modes, terminal characteristics, etc.)
# may be specified. To list the allowable options type "?" when
# prompted with " ". While USING any TELNET connection (created
# by either INITIATE or TELCONN) certain TELNET control functions
# may be invoked by typing the TELNET dynamic option escape
# character followed by a character indicating the desired
# function. The dynamic option escape character is initially ^D
# (CNTL D) but may be changed using the DYNAMIC option.
#
# The recognized commands are:
#
#   <^D><L> means to do Local echoing
#   <^D><R> means to do Remote echoing
#   <^D><T> means to Transmit accumulated chars immediately
#   <^D><B> means send TELNET Break character
#   <^D><S> means send the TELNET Synch sequence
#   Any other character will be transmitted as data.
#
# To send a <^D> character through the network, type <^><D>.
#
# WHERE (is user) username<cr>
# Lists all active jobs belonging to the specified user at all
# sites (with RSEEXEC servers running).
#
# WHO<cr> or WHO (at site) hostname<cr>
# Lists users with active jobs at specified (or all) network
# site(s) with RSEEXEC servers running.
```

January 1975

```
# _QUIT  
#  
# @
```

January 1975

Telnet User Guide

User Telnet (hereafter called Telnet) provides facilities for communicating with host computers via the ARPA network utilizing the TELNET protocol. The purpose of the Telnet program is threefold. It converts various terminals connected to TENEX into a standard type of terminal called a network virtual terminal (NVT) by interposing programs in the character streams between the terminal keyboard and printer and the terminal port on the host computer. Secondly, it provides information about the network to assist a user in establishing connections. Thirdly, it multiplexes the terminal among several remote jobs.

Telnet Command Interpreter

Instructions to the Telnet program are given via the Telnet Command Interpreter. When in command mode (see below), characters typed on the user's terminal are read by the Telnet command interpreter and decoded as commands to perform various actions by Telnet.

The Telnet command interpreter has two unique features. The command interpreter will refuse to hear anything it does not understand. With full-duplex terminals, this means that no echo will appear for characters which are not valid successors of the previous input. In any case, the character is ignored and a bell is typed out. The input stream that has already been typed is not forgotten however. Therefore, it is only necessary to type the correct character and not the complete command. This feature may be turned off with the "no fancy.command.interpret" command.

The other unique feature of the Telnet command interpreter is the use of question mark to discover what the command interpreter expects next. Typing a "?" at any time in command mode will elicit a list of words the command interpreter is expecting. Thus, typing a "?" when nothing has been typed will yield a list of all possible top-level commands. Typing "co?" will yield a list of all commands starting with "co". Typing "connection.to ?" will yield a list of possible arguments to the "connection.to" command.

The command interpreter provides command completion whenever a terminator is typed (full-duplex terminals only) and an exact match is achieved with some command or a unique initial substring is typed. Command completion may be suppressed with the "concise" command. Terminators are space, comma, alt-mode, and carriage return. Terminators are often not distinguished and are thus equivalent. Where necessary, comma is used to separate list items, space terminates a command or option and signals the desire to specify more options, carriage return ends a command unless more information is necessary. Altmode is the same as

January 1975

space except that it will cause command completion in those modes where it is normally suppressed.

Command/Remote Mode

As mentioned above, characters typed on the terminal keyboard may be used in two ways: either as commands to Telnet, or as input to the remote host. The choice is made on the basis of whether Telnet is in remote mode or command mode. In command mode, characters typed on the terminal keyboard are read by the Telnet command interpreter and decoded as commands to perform various actions. TELNET is initially in command mode and will revert to command mode whenever the Telnet escape character (see below) is typed.

The opposite of command mode is remote mode. In remote mode, characters typed on the keyboard (with certain exceptions) are not examined by Telnet at all, but are merely passed on to the remote host computer. Remote mode is normally entered after any command is executed when the current connection exists. The "local.mode" command may be used to defeat this. The effect of the "local.mode" command is cancelled by the "remote.mode" command or by the "connection.to" or "retrieve.connection" commands.

Escaping Back to Command Mode

At any time, typing the Telnet escape character (initially control-Z (SUB)) will cause Telnet to stop whatever it is doing and return to command mode. Occasionally, a slight delay may be experienced due to the need to clean up whatever was happening at the time. Telnet announces the switch to command mode by the appearance of a sharp sign "#" at the left margin. Telnet also indicates the transition out of command mode by the appearance of another sharp sign followed by a new line.

WARNING: If you have control-Z anywhere in your programming, you
should change your escape character for Telnet to other than
control-Z to avoid mishaps.

Making a Connection

There are two ways to make a connection. Typing "connection.to <host> [<qualifiers>]" or simply typing "<host> [<qualifiers>]" will cause a connection attempt to be made. If successful, the connection will be said to be complete and the terminal will be placed in remote mode. If unsuccessful, the connection will be said to be "incomplete because ---" with a reason given; also if

January 1975

the remote host is down, a line is typed telling why and for how long. By terminating the host name with a space, one or more qualifiers may be specified. Ordinarily socket 1 is assumed. Thus without a qualifier, the connection will be made to the "logger" on the remote system. By using an octal number as a qualifier, the connection will be made to the socket so specified. A set of names is available for specifying the socket desired. This set consists of names for all the standard sockets.

The "wait" qualifier may be used to camp-on the connection. This qualifier causes Telnet to repeat the attempt to connect in the event of a failure until it finally succeeds. An initial failure causes a message to that effect to be printed. When the attempt finally succeeds, bells are typed out to wake the user up. The attempt to connect may be aborted by typing the Telnet escape character.

The "load.settings.from..." qualifier (possibly qualified with "no") may be used to cause (inhibit) the mode flags to be initialized from the mode file. When inhibited, the current modes are used.

The "name.for.connection" qualifier may be used to specify a name for this connection other than the one assigned by Telnet. A name for the connection may also be given later by the "name.for.current.connection" command.

Disconnecting

The "disconnect" command is used to close the current connection. This will not necessarily log you out from the remote host so you should perform the logout procedure for that host before disconnecting. The disconnect command takes an optional argument specifying the name of a particular connection to be disconnected. See multiple connections and connection names below.

In the event that the network connections are severed by a network failure, the message "IO error for connection <name>" is printed, the connection is disconnected, and Telnet reverts to command mode. This may happen even if the error occurs on a connection which is not current. If the remote host initiates a disconnect, a message to that effect is printed and the same action is taken.

If the remote host on the current connection stops responding
when input is being sent, a line is typed, "Host not responding
on connection xxx." (In this case the connection is not lost.)

January 1975

- # When the remote host resumes operating, the user is informed:
- # "Service restored on connection xxx."

Echo Control

Telnet allows several options concerned with echoing. Echos may be generated by the terminal, by Telnet, or by the remote host. Telnet determines if the terminal is generating echoes when started by examining the mode word for the terminal. The "terminal.type.is" command may be used to change this.

If the terminal is echoing, then Telnet will do everything possible to cause the remote host to not generate echoes, and Telnet will not generate echoes itself. If the terminal is not generating echoes, then Telnet determines whether it should echo or not by information in the mode file (if any) or by the "echo remote"/"echo local" commands, or by information sent from the remote host.

Telnet keeps the remote host informed about how echoing is being done and if the remote host is suitably equipped, it will follow along. If not, then the user will have to give commands to the remote host to achieve the proper echoing. Telnet also will respond to commands from the remote host concerning who should be echoing. If Telnet believes the terminal is doing its own echoing, it will respond to any request from the remote host to not echo by an "I'll echo" command.

Line Buffering and End of Line Conventions

Telnet provides an optional line buffer for use with line-oriented operating systems. In this mode, characters typed in remote mode are stored in a local buffer up through an end of line. Prior to the end of line, the currently buffered line may be edited using control-A (SOH) or control-H (BS) to delete characters, control-X (CAN) to delete everything, and control-R (DC2) to retype the current contents. Telnet always converts the TENEX EOL into the NVT EOL. TENEX in turn converts a carriage return into the TENEX EOL. Thus typing a carriage return will cause the buffered line to be transmitted. Linefeed may also be used to terminate a line. In this case, the transmitted line will end with only linefeed, not the NVT EOL.

Telnet provides an optional linefeed echo for carriage return. If the remote host provides a linefeed also, then the echo generated by Telnet should be suppressed with the "echo no linefeed.for.carriage.return" command. In remote echo mode, Telnet generates no echos whatsoever. In this mode, all echos must be provided by the remote host.

Status Commands

Several status commands are available for discovering facts about the network. None of these commands will affect the state of the current connection. The status commands include where.am.I, status.of, netstatus, and socket.map. These commands are summarized below.

Special Characters

Several commands are available to send characters which do not appear on the terminal. "Code" takes an octal (decimal if preceded by "D", hexadecimal if preceded by "H") argument and sends the character with that code. The word "code" may be omitted and just the argument typed. "Control" takes a character argument and sends the corresponding control character (the low order five bits of the character) is sent. The "!break!" command sends the NVT break character which is mapped by some systems into the equivalent of the attention, quit or break key which appears on some terminals.

To facilitate operation with systems requiring frequent use of special characters or lower/upper case graphics which a particular terminal may lack (e.g. 33 Teletypes have no lower case), case shift characters may be defined for upper/lower character/lock shifts and characters may be defined which will translate into attention or break (NVT 201), and the synch sequence. The "case.shift.prefix.for", "attention.character=", and "synch.character=" commands are available to independently set each of these characters. In addition, a character may be defined ("quote.prefix" command) to be a single character quote. The character following this character is always sent regardless of any special action it may otherwise have.

If possible, case shift characters will be used to indicate the case of both input and output. Thus the case shift characters may not be echoed when typed but rather before the output.

All special characters are listed by the "current.modes.are" command. This includes the escape character and the clear output buffer character.

January 1975

Leaving Telnet

To leave Telnet, it is first necessary to return to command mode by typing the escape character. This is because while in remote mode all characters except the escape character are passed on to the remote host or modify characters passed to the remote host. Once in command mode, you may return to the EXEC by typing control-C (ETX) or by using the "quit" command. Continuing from the EXEC will resume with no loss. The "logout" command will disconnect from any remote job and logout your local job. The "exec" command will start up an inferior EXEC under Telnet. From this inferior EXEC, it is possible to perform assemblies or any other task involving the running of subsystems. The "run" command allows an arbitrary program to be run in an inferior fork of Telnet. The "run" may be interrupted by the Telnet escape character.

Multiple Connections

Telnet provides a facility for multiplexing a user's terminal among several remote jobs thus allowing several simultaneous activities. This is done by giving a name for each connection as it is created. The user may specify the name, or Telnet will default the name to a number. The "retrieve.connection..." command causes the named connection to be made current and remote mode to be entered. Non-current connections remain active, but any output received is buffered until that connection again becomes active. Terminal input goes only to the currently active connection.

The name of the current connection may be changed after it is established by means of the "name.for.current.connection" command. The name so specified may be up to 6 characters in length and must be unique.

Typescript

```
# Telnet provides a means of saving on a file a copy of the
# typescript for a session. This is useful for producing hard copy
# of the session when using a scope terminal or for producing
documentation of procedures or demonstrations. Telnet always
keeps a typescript on the temporary file "TELNET.TYPESCRIPT;T" in
the connected directory. The "typescript.to.file" command may be
used to specify a different file. The typescript consists of a
nearly exact copy of what appears on the terminal with the
exception of that which occurs during the execution of the "exec"
or "run" or "ddt" commands. "Nearly" refers to slight
differences in the spelling of file names in certain Telnet
# commands. For privacy, the typescript file is given a protection
```

that allows no access to anyone but "self".

Diverting Output

The output stream may be diverted to some other file with the "divert.output.to.file" command. While diverting output, Telnet sends all output to the indicated file and sends a line to the terminal only when the terminal's output buffer is empty. Thus the terminal monitors the transmission of the stream to the file. The diverted output consists only of characters from the remote host. Telnet commands and responses do not appear in the diverted information. This mode is useful as a primitive file

transfer mechanism or to allow printing of large amounts of
terminal output to be done with the lineprinter. It is cancelled
by "no divert.output ...".

#

#

#

Input from a File

#

The input stream to a remote job may be taken from a file instead
of the local terminal by means of the command
"take.input.stream.from.file". Telnet blocks terminal input to
the connection current when the file is specified, and transmits
characters from the named file (echoing as usual according to
current modes). However, input to other connections and in
command mode is from the user's terminal. When the given file
reaches EOF the file is closed and released, and input reverts to
the terminal. The user may also manually cancel file input by
escaping to command mode and giving "no take.input ...". This
mode is useful for routine sequences performed in the remote job.
Note that a connection must be established and current when input
to it is diverted to a file.

January 1975

Telnet Command Summary

Connection.to <host> or host name

Performs ICP to connect to the indicated host. Options are available for specifying initial connection socket name or number, and initializing modes from the mode file via the following subcommands. Note that if <host name> is used as a command, only the name of a server host may be given (e.g., BBN-TENEX). The argument for "Connection.to" may be any host name or an octal host number.

<octal number>

An ICP is performed to connect to the indicated service socket. Normally socket 1 is assumed.

Logger

Sets socket to 1.

Wait

The connection attempt is repeated until successful.

Name.for.connection.is <name>

Sets the name for this connection as specified.

[no] load.settings

Determines whether to use current mode settings or to load new ones from the mode file.

Disconnect <cr>

Disconnects the current connection. This will not necessarily log you out from the remote host.

January 1975

Perform the necessary operations before disconnecting.

Disconnect <name>

Disconnects the connection with the specified name.

Net.exec

Connects to BBN socket 15600031 where-in the
RSEXEC (Resource-Sharing Executive) is found.

Status.of <host>

Performs ICP with the indicated host and prints its status.

Echo.mode.is

Sets echo mode according to the following subcommand.

[no] remote

Turns off echoes generated by Telnet and signals the remote computer to generate echoes. Some hosts are not yet equipped to handle this signal and may require additional action to cause the remote computer to generate echoes. If Telnet believes it is connected to a local half-duplex terminal, it will complain about remote echoes but do it anyway.

[no] local

Turns on Telnet generated echoes and signal the remote computer to not generate echoes. Note that Telnet never generates echoes for terminals it believes have local echo of their own.

[no] linefeed.for.carriage.return

January 1975

TENEX translates carriage return to EOL, Telnet sends the EOL as the TELNET EOL (i.e. carriage return-linefeed). For some systems, the TELNET EOL is translated into carriage return. For these systems, the appropriate echo is carriage return. Other systems translate the TELNET EOL into carriage return-linefeed. For these systems the appropriate echo is carriage return-linefeed. This subcommand causes the latter echo to be generated.

[no] control.character.echo.for <list of characters>

Turns on local echoes for the indicated control characters. Normally only control-G,J, and M (bell, linefeed, and carriage return) are enabled.

Terminal.type.is

Allows the user to change Telnet's opinion of his terminal according to the following subcommands. Each command may be preceded by the word "no" to negate its meaning.

Half-duplex

Terminal generates its own echoes.

Full-duplex

Terminal does not generate its own echoes.

[no] lower.case

The terminal has lower case characters.

Local.mode

If connected, this command prevents Telnet from returning to remote mode after each command.

Remote.mode

If connected, this command causes Telnet to return to remote mode after each command. If not connected, it does nothing.

No

May appear before some commands to reverse their action.

Current.modes.are

Prints the state of connection terminal mode flags, and all special characters.

[no] character.mode

Causes each character typed to be transmitted as it is typed.

[no] line.buffer

Causes Telnet to accumulate a line of text before transmitting. A line ends on linefeed or EOL or altnode (esc). The line may be edited with control-a, x, and r.

[no] raise

Causes lower case letters to be transmitted as their upper case equivalents.

[no] lower

Causes upper case letters to be transmitted as their lower case equivalents.

[no] transparent.mode

Causes all characters to pass through Telnet and TENEX untouched. This is needed for special terminals such as the IMLAC using special character stream protocols.

January 1975

[no] case.shift.prefix.for

Allows the specification of the four case shift characters according the following four subcommands.

Lock.lower.case

Same as the "Lower" command. Subsequent upper case input will be converted to lower case.

Char.lower.case

Converts the following letter to lower case.

Lock.upper.case

Same as "Raise" command. Subsequent lower case input will be converted to upper case.

Char.upper.case

Converts the following character to upper case.

[no] unshift.prefix

Causes all following characters to be unshifted. I.e. undoes both an upper case lock and a lower case lock.

[no] quote.prefix

Causes the following character to be transmitted without regard to any special significance it may have.

[no] synch.character

The specified character will be converted to the TELNET synch sequence. The TELNET synch sequence

January 1975

is used to cause the remote host examine its input stream to the current point for any special characters (interrupts, attentions etc.). All non-special may be thrown away.

[no] attention.character

The specified character will be converted to the TELNET break or attention character. This character is equivalent to the attention, quit, or break key on certain terminals and may be necessary for using some systems. The !Break! command generates the same character.

Concise

Turns off automatic command completion. Saves typeout at the expense of readability.

Verbose

The opposite of concise.

[no] fancy.command.interpret

Commands are checked character by character. If a character does not fit, it is ignored and not echoed (full duplex terminals only).

[no] divert.output.stream.to.file

Causes all subsequent output from the remote computer to be written on the specified file. Use "No divert..." to stop this.

[no] take.input.stream.from.file

Causes subsequent input to the remote host on the current connection to be read from the specified file; input to other connections and in command mode is still from the user's terminal. File is automatically closed and released at EOF; user may force this by "No take.input...", after escaping to command mode.

January 1975

[no] typescript.to.file

A record of the session is kept on a file including both input and output. This is useful for providing hard copy with scope terminals.

Escape.character=

The specified character becomes the Telnet escape character. This character must be a TENEX interrupt character. "?" will type what these are.

WARNING: If you have anywhere in your programming
a control-Z you should change your escape
character in TELNET to other than control-Z to
avoid mishaps.

Clear.output.character=

The specified character becomes the clear output buffer character. Typing this character generates an interrupt which causes the terminal output buffer and any accumulated output to be cleared.

Help

Prints the file <SYSTEM>TELNET.HELP on the user's terminal.

Netstatus

Runs <SUBSYS>NETSTAT.SAV.

Socket.map

Prints a list of all current connection on the system. Optional arguments may be used to select a particular host and a particular connection state.

Run

Runs the specified file. Like the EXEC's run command.

Quit

Returns from Telnet to the superior fork (usually the EXEC). May be continued with no loss.

Logout

Logs out the local job (not the remote one). Requires confirmation with a carriage return.

Reset

Re-initializes Telnet producing an essentially virgin copy.

Ddt

Enters ddt. If ddt is not loaded, this will result in an unexpected interrupt. No harm is done if this happens.

Exec

Starts up an inferior EXEC under Telnet. This EXEC may be used like an ordinary EXEC to run subsystems etc without disturbing any existing connections. The Telnet escape character will return to Telnet however.

Code

Transmits the character specified by the argument. The argument is taken as an octal number unless preceded by "d" for decimal or "h" for hexadecimal. The argument may be preceded by "o" for octal.

The "code" command argument may be used as a command by itself and will cause the indicated code to be transmitted.

!break!

Transmits the TELNET break character.

January 1975

!synch!

Transmits the TELNET synch sequence. Occasionally the "!"synch!" command will work where the synch character will not since the command bypasses the buffering which may interfere with the use of the synch character.

Write.modes.for.host

Causes the current mode flags to be saved on the <SYSTEM>TELNET.MODES file under the specified host. Requires write access to the file and is thus not available to ordinary users.

Retrieve.connection.under.name

Retrieves the connection previously saved under the specified name.

Wait.for.any.active.connection

Used with multiple connections to wait for and switch attention to the next connection that has any output waiting. Useful when several independent tasks are being run and you wish to know when one completes and switch to that task.

Where.am.I

Prints a summary of the local job, system, user, terminal and the remote host and socket.

[no] Signal.waiting.output

Causes all non-current connections to print a message when output becomes available.

Host.names

Lists all current host names with corresponding
octal host numbers.

List.connections

January 1975

Lists the name, local socket, foreign host, and foreign socket of all connections.

Flush.host

Marks all connections to the specified host as dead and sends a reset to that host. Requires wheel or operator special capability.

An initial semi-colon causes the remainder of the line to be ignored. Useful for comments or typing to links.