Introduction to TELNET and FTP on UNIX

SYNOPSIS

This document is written with the novice user in mind. It describes the use of TCP/IP and FTP to transfer files to and from the UNIX operating system and another Internet attached host. It also describes the use of TELNET to communicate as a terminal to and from other Internet host systems.

INTRODUCTION

This document is broken into five major sections: this introduction; using the UNIX operating system as a local ftp host; using the UNIX operating system as a remote ftp host; using TELNET to the UNIX operating system; and using TELNET from the UNIX operating system to other hosts.

FTP (File Transfer Protocol) is a TCP/IP (Transmission Control Protocol/Internet Protocol) protocol for file transfer server and user functions. To participate, a host must be connected to the Internet via gateways and bridges. Transfers to and from hosts must be accomplished by other hosts which are also attached to the Internet and which support compatible FTP programs.

In order to connect to a remote host, you must know the Internet address or domain name of that remote host. Because AIX and some of the other UNIX systems at Cal Poly are cluster oriented with multiple CPUs, there are multiple names associated with each cluster. It is always best to use the site where the files are or will be stored. The AIX operating system’s domain names are:

Domain Names:
- cymbal.aix.calpoly.edu
- flute.aix.calpoly.edu
- harp.aix.calpoly.edu
- oboe.aix.calpoly.edu
- trumpet.aix.calpoly.edu
- tuba.aix.calpoly.edu
- violin.aix.calpoly.edu
- zeus.aix.calpoly.edu

TELNET allows you to communicate to an Ethernet attached host from an Ethernet attached machine running TELNET. Because of the UNIX software architecture, it is highly recommended that you use a version of TELNET which supports a vt100 terminal emulation. vt100 will give the user the greatest flexibility in using the UNIX operating system. DO NOT use a product which support the IBM 3270 block mode terminal operation as UNIX does not support it.

You may use any Internet capable host to TELNET to a UNIX system which is also on the Internet. If you do use screen mode, how does your TELNET host support the various function or application keys on the terminal type selected? Most PC TELNET programs are very flexible and provide you with a usable set of keys. We will assume, for the rest of this document, that you are on a properly configured TELNET host.

Everything else about the session is just like being logged onto the system with a real terminal. If you do have a complete vt100 emulation, the UNIX operating system should behave just as if you were using a vt100 class terminal. This includes the way in which UNIX supports the arrow keys.
TELNETING FROM UNIX  Most UNIX systems support a vt100 emulation as part of their TELNET software. There is also another variety of TELNET called tn3270 which allow you to telnet to an IBM block mode system which expects an IBM 3270 class terminal.

On the AIX system, the telnet program may be invoked in several ways, regardless of the target host, AIX will adjust to 3270 mode versus vt100 mode. However, the key mapping chosen may not be what is expected. Please refer to the key mapping section later in this chapter for further information on key mapping.

A. VT100 TERMINAL EMULATION

To telnet to another Internet attached system which uses a vt100 terminal emulation enter

  % telnet hostname [port_number] <CR>

where "hostname " is the name of the host you wish to connect to in the form "host . site . suf " . If you are logged into a machine at the same site as the one you are telnetting to, you may leave off the site and suffix. "port_number " is an optional port number for some services provided on some machines.

NOTE: Some sites may require the use of additional commands while using TELNET. The Cal Poly AIX system will send intermittent messages to the user to use the "play " command while using "telnet ". On AIX, the form to use to avoid these messages is:

  % play telnet hostname [port_number] <CR>

Once the terminal session starts, its as if you had connected to that remote system with a vt100 terminal.

Usually you will receive a connection message similar to:

  Trying . . .
  Connected to harp . aix . calpoly . edu.
  Escape character is ' ^] ' .

  telnet (harp . aix . calpoly . edu)

Note the "Escape character" (in this case "^]"), it is useful in breaking the connection at a later time should the connection get hung. Usually the escape character will be either "^]" (for telnet on AIX), "^T" (for tn on AIX), and "^C" (for tn3270 on AIX). Typing the escape character will result in a "telnet >" prompt (for "tn>" or "tn3270" on AIX depending on the command used to start the session). See the sub-section "DISCONNECTING A TELNET SESSION" later in this section for more information on how to close a TELNET session.

When you log off the remote system, you should receive the message

  Connection closed.

And then receive your normal UNIX system prompt.

B. 3270 TERMINAL EMULATION

When tn3270 is specified as the command, or in the case of AIX, any of the three TELNET commands (telnet, tn, or tn3270), and a IBG 3270 block mode capable host is being connected to, the user must also be aware of the keymapping necessary for there terminal session. With tn3270 on most UNIX
systems there is usually only one key mapping file to worry about. With AIX, the keymapping will vary depending on how the TELNET session was started.

The user can prevent negotiation by specifying the emulation type with a "-e" parameter as follows:

```
% telnet -e 3270 machine.sit.e.suf <CR>
```

This will guarantee that the user is starting the session in 3270 mode only, no negotiation will take place. Please refer to the section entitled "TELNET KEY MAPPING" for more information on key mapping on your terminal type.

C. DISCONNECTING A TELNET SESSION

To disconnect the telnet session, you must use the Escape character given by the TELNET program at the time the connection is made. Usually the escape character will be either "^]" or "^T" and typing it will result in a "telnet>" prompt.

Once the prompt is received, type

```
telnet> close<CR>
```

to close the connection. When the connection is closed, you should return to the system prompt unless a host name was not specified on the the TELNET command line. If this is the case and you wish to leave the TELNET program enter

```
telnet> quit<CR>
```

and you will be returned to the system prompt.

TELNET KEY MAPPING

If the client and the server negotiate to use a 3270 data stream the keyboard mapping is determine by the following precedence:

- `$HOME/.3270keys` Specifies the user's 3270 keyboard mapping when `tn` or `telnet` command is invoked.
- `/etc/map3270` Specifies the user's 3270 keyboard mapping when `tn3270` command is invoked.
- `/etc/3270.keys` Specifies the base 3270 keyboard mapping for use with limited function terminals (vt100 and others).
- `/etc/tn3270.keys` Binds the terminal keys depending on the terminal type specified by the TERM environmental variable. This file gets loaded when the `tn3270` command is invoked.

The following table provides a list of the various key mappings based on a vt100 terminal setting for the TERM environmental variable and the various startup commands.

<table>
<thead>
<tr>
<th>3270 Function</th>
<th>vt100 Key Sequence</th>
</tr>
</thead>
<tbody>
<tr>
<td>Backtab</td>
<td><code>&lt;Esc&gt;&lt;TAB&gt;</code> or Ctrl-D</td>
</tr>
<tr>
<td>Backspace</td>
<td><code>&lt;BS&gt;</code> or Ctrl-H</td>
</tr>
<tr>
<td>Clear</td>
<td>Ctrl-L or Ctrl-Z</td>
</tr>
</tbody>
</table>

Table TCPIP-1: 3270 Key Mappings for a vt100 Terminal
Some of the following subcommands may be specified on the TELNET prompt (telnet>, tn>, or tn3270>). For a complete list of the available subcommands, refer to the "telnet" man page.

? [subcommand] <CR> provides help on telnet subcommands. If a subcommand is not specified, a list of the valid subcommands will be displayed.

close<CR> Closes the telnet session and returns to TELNET command mode.

display [argument] <CR> displays all of the set and toggle values if no argument is specified; otherwise, lists only those values that match the argument variable.

emulate terminal_type<CR> Overrides terminal type negotiation with the specified terminal type. Possible choices are:

? Prints help information.
3270 Emulates a 3270 terminal (AIX only).
none Specifies no emulation.
vt100 Emulates a DEC VT100 terminal.

open host [port]<CR> Opens a connection to the specified host.
**UNIX FTIP COMMAND SYNTAX**

The command syntax for IBM AIX/UNIX FTP commands is as follows:

```
ftp> ftp_sub-command param1 param2 ... paramN<CR>
```

where "ftp>" is the UNIX FTP sub-command prompt. "ftp_sub-command" is the FTP sub-command. The parameters are separated by spaces and their form depends on the order that they appear, the parameters abbreviation, and the presence of sub-parameters. Options always follow an open parenthesis at the end of the command line.

**TRANSMITTING FILES TO AND FROM UNIX**

The following is a basic scenario for transferring files to and from UNIX, as a local host, from another remote host.

---

**FTP WITH UNIX AS THE LOCAL HOST**

**A. UNIX FTP COMMAND SYNTAX**

The command syntax for IBM AIX/UNIX FTP commands is as follows:

```
ftp> ftp_sub-command param1 param2 ... paramN<CR>
```

where "ftp>" is the UNIX FTP sub-command prompt. "ftp_sub-command" is the FTP sub-command. The parameters are separated by spaces and their form depends on the order that they appear, the parameters abbreviation, and the presence of sub-parameters. Options always follow an open parenthesis at the end of the command line.

**B. TRANSMITTING FILES TO AND FROM UNIX**

The following is a basic scenario for transferring files to and from UNIX, as a local host, from another remote host.
These steps include making the FTP connection to the remote host. If the connection already exists, you do not need to establish it. If you already have an established connection, skip to "Listing Files on the Remote Host" later in this section.

1. **START FTP ON UNIX HOST.**

   To start FTP on UNIX local host, simply enter the FTP command as follows:
   
   ```
   % ftp<CR>
   ```
   
   Typing the command "FTP " at the VM system prompt causes the VM FTP program to be executed.

   The IBM will respond with a message line as follows:
   
   ```
   AIX/UNIX TCP/IP FTP R1.2.1
   OPEN (name of foreign host):
   ```

2. **OPEN THE FTP SESSION.**

   To open the session, the user enters
   
   ```
   ftp_address<CR>
   ```
   
   "ftp_address" is either an Internet numeric address composed in a string resembling `xx.xx.xx.xx` (where each "xx" is in the range of 1 to 255) or an Internet Host Name. If you use an Internet Host Name, that name must be defined in Host tables within the local host.

   FTP will then establish a connection with the remote host if it can find it within the Internet. Once the connection is established, the user will be prompted for the user I.D. or Account as follows:
   
   ```
   USER (identify yourself to the host): account<CR>
   ```
   
   Enter the account number or user I.D. for your account on the remote host. Some FTP hosts allow you to log on as "anonymous" or "guest".

   If the remote host requires a password, FTP will then prompt the user
   
   ```
   331 Password required for account.
   Password: password<CR>
   ```
   
   Enter the password assigned to the account entered in the previous prompt.

3. **LISTING FILES ON THE REMOTE HOST**

   There are two commands available for listing files on the remote host, LS and DIR. LS will provide a brief list of the files in the current working directory on the remote host in a format similar to what the remote host would use for a local user. DIR will usually produce a more comprehensive list of files, again in a format that the remote host would use for a local user.

   To use the LS command we enter
   
   ```
   Ftp> ls<CR>
   ```

   FTP responds with output similar to the following (the examples here were performed using a UNIX remote host).
4. REMOTE HOST TO LOCAL HOST TRANSFER.

The first example of a file transfer involves the use of the get command. It assumes that the remote host file "remote_file" is an upper and lower case text file and that we want the same type of file created on UNIX. To perform this transfer we enter

```
ftp> get remote_file local_file<CR>
```

where "remote_file" is the name of the file on the remote host and "local_file" is a valid UNIX file name for our local UNIX host. If the remote file name is a valid UNIX file name and we wish to keep it, we may leave out the "local_file" entirely. For example:

```
ftp> get fred.txt fred.data<CR>
```

In the example the remote host file "fred.txt" will be transferred to the user's account with the name "FRED DATA".

5. LOCAL HOST TO REMOTE HOST TRANSFER.

The first example of a file transfer involves the use of the put command. It assumes that the local host file "local_file" is an upper and lower case text file and that we want the same type of file created on the remote host. To perform this transfer we enter

```
ftp> put local_file remote_file<CR>
```

Where "local_file" is the name of the file on UNIX and "remote_file" is a valid file name for your remote host. If the remote file name is to be the same as the UNIX name and we wish to keep it, we may leave the second parameter entirely.

6. LISTING FILES ON THE REMOTE HOST.

There are two commands available for listing files on the remote host, ls and dir. ls will provide a brief list of the files in the current working directory on the remote host in a format similar to what the remote host would use for a local user. dir will produce a more comprehensive list of files, again in a format that the remote host would use for a local user.

To use the LS command we enter

```
ftp> ls<CR>
```

FTP responds with output similar to the following (the examples here were performed using a UNIX remote host).

```
.cshrc
.history
.login
.logout
file1. ext
file2. ext
```
7. DELETING FILES ON THE REMOTE HOST.

During a transfer, you may find it necessary to delete files on the remote host. This may be due to disk quotas or accidentally transferring the wrong file. UNIX FTP allows you to delete one file at a time on the remote host. To delete a file on the remote host enter

```bash
ftp> delete file1.new
```

Where "file1.new" is the name of the file stored on the remote host that we want to delete.

8. CLOSING THE CURRENT CONNECTION TO THE REMOTE HOST.

When all of the files have been transferred to the local host, you will close the connection to the remote host as follows:

```bash
ftp> close
```

This has the effect of logging you out of the remote host.

The system will then respond with

```
221 Goodbye.
```

9. EXITING FTP

To exit FTP we enter

```bash
ftp> quit
```

When FTP returns control to UNIX you will see the system prompt.

---

### FTP WITH UNIX AS THE REMOTE HOST

The UNIX FTP Server supports a variety of commands from local FTP hosts. Most FTP hosts generate these commands based upon commands similar to those discussed in the previous section above.

### FTP COMMAND SUMMARY

What follows is two lists of FTP Sub-commands for TCP/IP under UNIX. The first set is based on UNIX being the local host. The second set is based on UNIX being the remote host. More detailed descriptions and additional commands may be found in the man page "ftp".

#### A. UNIX AS THE LOCAL HOST

- `!command<CR>`: Pass a UNIX command to the local environment for processing (must be the full command name, not a synonym).
- `?subcommand<CR>`: Displays a help message describing the subcommand. If no subcommand is provided, a list of valid subcommands is displayed.
- `account password<CR>`: Send accounting information to the remote host.
append local_file [remote_file] <CR>
Append the local file "local_file" to the remote file "remote_file".

ascii <CR>
Set transfer type to ASCII (for text files).

bell <CR>
Sounds a bell after the completion of each file transfer.

binary <CR>
Set transfer type to binary variable length records.

bye <CR>
Ends the FTP session and exits the FTP command.

cd newpath <CR>
Changes the remote host working directory on the remote host to the path specified by "newpath".

close <CR>
Close the current connection to a remote host.

delete filename <CR>
Delete the file specified by "filename" on the remote host.

dir [filename] <CR>
Display a detailed remote host file list for the working directory. If a file is specified with "filename", info for that file will be specified.

ebcdic <CR>
Sets file transfer mode to EBCDIC (a file will result with the EBCDIC character set.

get remote_file [local_file] <CR>
Retrieves the file specified by the "remote_file" parameter on the remote host and stores it under the name specified by the "local_file" parameter on the local host.

hash <CR>
Toggles hash sign (#) printing. A hash mark is printed every 1024 bytes on most UNIX systems. This is a good way to determine that the transfer is still in progress.

help <CR>
Displays information about UNIX FTP subcommands.

help subcommand <CR>
Gives detailed help on the "subcommand".

ls [filename] <CR>
Displays the remote host file list for the working directory. If a file is specified with "filename", info for that file will be specified.

mdelete [filename [filename [...]]] <CR>
Deletes the named file(s) on the remote host.

mget [filename [filename [...]]] <CR>
Gets one or more files from the remote host. If the filename is not acceptable to the local host, the file is not transferred.

mode [ stream | block ] <CR>
Sets file-transfer mode. The default is stream.
put [filename [filename [...]]]<CR>  
Places one or more files on the remote host. If the filename is not acceptable to the local host, the file is not transferred. If the file already exists on the foreign host, the name of the transferred file will be changed to make it unique if SUNIQUE has been used.

open ip_address<CR>  
Open a connection to the host specified by "ip_address ". The "ip_address " may be it's name or IP address.

prompt<CR>  
Toggles interactive prompting. Default is to prompt for each file for any mutli-file command.

put local_file [remote_file]<CR>  
Send the file "local_file " to the remote host. The file will be stored under "remote_file " on the remote host if specified.

pwd<CR>  
Displays the name of the current working directory on the remote host.

quit<CR>  
Terminates the execution of FTP.

quote string<CR>  
Sends the FTP command specified with the "string " parameter directly to the remote host for processing.

remotehelp [subcommand]<CR>  
Gives help offered by the remote host for FTP commands it supports.

rename old_filename new_filename<CR>  
Renames the files listed with the "old_filename " parameter on the remote host to the corresponding names listed with the "new_filename " parameter.

rmdir remote_directory<CR>  
Removes the remote directory specified at the remote host.

runique<CR>  
Toggle the creation of unique file names for local destination files during get and mget subcommands.

sendport<CR>  
Toggles PORT commands on the remote host.

site [parameters]  
Send information to the remote host to provide services specific to the foreign host. "parameters " are dependant on the foreign host.

status<CR>  
Displays current status of the FTP command as well as the status of the subcommands.

sunique<CR>  
Specifies that if the file exists on the remote host, the transferred file will be renamed to "rfilename1 " from "rfilename " on the PUT or MPUT commands.
system<CR>                        Print the name of the operating system on the remote host.
type ascii<CR>                   Specifies the transfer type as ascii, which is the default.
type ebcidi<CR>                  Sets the file transfer type to ebcidi c.
type [image|binary]<CR>          Specifies the transfer type as image. This has the same effect as the binary command.
user username [password] [account]<CR>  Transmits login information to the remote host after a connection has been established.
verbose<CR>                      Toggles verbose mode. Defaults to on.

B. UNIX AS THE REMOTE HOST

All of the commands described in the previous section may be sent from another machine to the UNIX FTP server while it is in the remote host role.

HELP                                    Gives help information.