Chapter 13: Connecting to Remote Systems

Several utilities are available to enable you to transfer files between systems, to log into other systems on which you have an account, and to execute commands remotely. These features are described in this chapter.

Note that to use *any* of the communication methods with a VMS node, the VMS node must be running software to allow tcp/ip communication. All Fermilab VMS machines are licensed to run TGV's **MultiNet** software, which provides the necessary tcp/ip transport capability.

13.1 Transferring Files

13.1.1 ftp

ftp (File Transfer Protocol) can be used to transfer files to and from a remote system on which you have an account, and to manipulate the remote file system. The command format is:

`% ftp [-n] [-i] [`*`host`*`]`

If you specify ***host*** on the command line, you will be prompted for your (remote) username and password, and login will proceed. The option **`-n`** inhibits autologging on those systems on which it is enabled. The option **`-i`** eliminates prompting by the system, which can be tiresome when you're transferring many files.

Some commands available in **ftp** are described here; for a complete list, see the man pages for **ftp**.

`user` ***username password***	Identification for the remote system. If auto-log is enabled, you will be prompted automatically for the username and password. If you don't specify the password, it will prompt you.
`bin` or **`binary`**	Set transfer mode to binary image transfer.
`ascii`	Set transfer mode to ASCII (for text files).
`put` ***local_file*** [***remote_file***]	Transfer a local file to the remote system. If the remote filename is not specified, the local filename is used.
`mput` ***local_files*** [***remote_files***]	Similar to **`put`**, but you can use standard wildcard characters to transfer a series of files at a time.
`get` ***remote_file*** [***local_file***]	Transfer a remote file to the local system. If the local file name is not specified, it is given the name on the remote machine.
`mget` ***remote_files*** [***local_files***]	Similar to **`get`**, but you can use standard wildcard characters to transfer a series of files at a time.

`help` [*command*]	Display information about the meaning of ***command***. If no command is specified, **ftp** displays a list of the commands.
`quit`	Terminate the **ftp** session with the remote server and exit **ftp**.
`open` *host*	Establish connection with remote host. Needed if ***host*** was not specified on command line.
`close`	Close connection with remote host and return to **ftp** prompt. Doesn't exit **ftp**.
`dir` [*remote_directory*] [*local_file*]	Print listing of the directory contents of the remote directory and optionally put the output in ***local_file***.
`cd` *remote_directory*	Change working directory on the remote machine to ***remote_directory***.
`lcd` [*directory*]	Change working directory on the local machine. If no directory is specified, your home directory is used.
`delete` *remote_file*	Delete the file ***remote_file*** on the remote system.
`!` *shell_command*	Run a shell command without exiting **ftp**.

The following is an example **ftp** session where first one file, then a few files, are transferred to a UNIX machine from the FNALV cluster. The username is unnecessary if the assumed one is correct. The password, as usual, is not displayed.

```
<fsui01> ftp fnalv

Connected to fnalv.fnal.gov.
220 fnalv1.fnal.gov MultiNet FTP Server Process 3.4(14) at Wed 3-Jan-96 11:41AM-
CST
Name (fnalv:aheavey): 331 User name (aheavey) ok. Password, please.
Password:
230 User AHEAVEY logged into USR$ROOT17:[AHEAVEY] at Wed 3-Jan-96 11:42AM-CST, j
ob 25a26688.
ftp> bin
ftp> get newfile
200 Port 159.217 at Host 131.225.68.16 accepted.
150 ASCII retrieve of USR$ROOT17:[AHEAVEY]NEWFILE.;1 started.
226 Transfer completed.  25 (8) bytes transferred.
local: newfile remote: newfile
25 bytes received in 0.14 seconds (0.18 Kbytes/s)
ftp> lcd psfiles
Local directory now /afs/fnal.gov/files/home/room3/aheavey/psfiles
ftp> mget c*
mget cmdconcp.frame.ps? y
200 Port 159.219 at Host 131.225.68.16 accepted.
150 ASCII retrieve of USR$ROOT17:[AHEAVEY]CMDCONCP.FRAME$5NPS;1 started.
226 Transfer completed.  182118 (8) bytes transferred.
local: cmdconcp.frame.ps remote: cmdconcp.frame.ps
182118 bytes received in 0.69 seconds (2.6e+02 Kbytes/s)
mget cmdconcp.ps? n
mget codemgmt.frame.ps? y
200 Port 159.224 at Host 131.225.68.16 accepted.
150 ASCII retrieve of USR$ROOT17:[AHEAVEY]CODEMGMT.FRAME$5NPS;1 started.
226 Transfer completed.  81133 (8) bytes transferred.
local: codemgmt.frame.ps remote: codemgmt.frame.ps
81133 bytes received in 0.32 seconds (2.5e+02 Kbytes/s)
ftp> quit
221 QUIT command received. Goodbye.
<fsui01>
```

Note that the CERN utility **zftp** (**z** for Zebra) is available for transferring FZ and RZ files (ntuple files) between systems[1]. To use **zftp**, it must be installed on the server system, and you need to set up CERN library during the session (**`setup cern`**).

13.1.2 rcp

Another way to transfer files is with the **rcp** utility (stands for **r**emote **cop**y). You can use this command if both local and remote hosts support **rcp** protocol. The basic format is:

`%` **`rcp [`*options*`]`** ***file1 file2***

where ***file1*** and ***file2*** are the source and destination filenames, respectively; or

`%` **`rcp [`*options*`]`** ***file1 file2 ... directory***

where the files listed are to be copied to the directory. On a UNIX system the format of a remote directory or filename is ***hostname*`:`*path***, and the **MultiNet** implementation is consistent with VMS usage, namely ***hostname*`::`*path*** (note the double colon). If only filenames are given, they are interpreted relative to your home directory. Both the source and the destination may be on nodes other than the current machine.

The available options are:

`-p`	Give each copy the same modification times, access times, and modes as the original file.
`-r`	Copy each subtree rooted at ***file1***, ***file2***, etc. (for those that are directories); in this case the destination (last argument) *must* be a directory.

rcp does not prompt for passwords. Your current local user name must exist on the remote host and allow remote command execution by **rsh**. **rsh** is a utility that connects to a remote host and either executes a specified command or logs you on via **rlogin** (see section 13.3).

More information is available in the **rsh** and **rcp** man pages.

When you are using **rcp** between VMS and UNIX, you must protect special characters from local interpretation by enclosing them in quotes. An example command to copy a file `$HOME/prog1.c` from UNIX to the VMS node FNALV is:

`%` **`rcp prog1.c fnalv:"[.c]prog1.c"`**

Here is an example showing how to copy a file from remote UNIX node FNSG01 to the home directory on the VMS node (**`$`** represents the VMS prompt):

`$ rcp fnsg01::"/usr/products/bufio/v1_00/src" bufio.src`

If you are using the C shell family, **rcp** will not work if your `.cshrc` or `.login` on UNIX, or your `LOGIN.COM` file on a remote VMS host, executes interactive or output-generating commands. You should check for non-interactive access, and exit before such commands are executed, as is done for a UNIX system in the FUE default `.cshrc` file. The lines in `.cshrc` that check this are the following:

```
# Place any items that you want executed even for non-interactive use here
# Skip if not interactive shell
if ( $?USER == 0 || $?prompt == 0 ) exit
```

1. This program is known to be likely to break in the AFS environment.

On a VMS system, check for **OTHER** (TCP/IP is not **NETWORK**):

$ IF (F$MODE () .EQS. "OTHER") THEN $ EXIT

Although you can specify a username and password with **rcp** (see the local man pages or **HELP**), you can also have a kind of "proxy login" by creating a special file (named `.rhosts`) on the remote machine containing the names of the systems and users on those system that should be allowed access.

13.1.3 The .rhosts File

To use many of the file transfer utilities, you need to set up what is known as a `.rhosts` file on the remote machine. This file must include a list of the names of the systems to be allowed remote access without login, and the usernames, where each system-username combination is separated by a space. If the username is left off, it is assumed to be the same as on the current system. If there are problems with name servers, you can be most assured that your address will be understood if you specify three lines for each connection requested:

1) one line with the complete Internet address,

2) one with the Internet number[1], and

3) one with the local name.

An example `.rhosts` file showing a single connection in the three specified formats:

```
fnsg01.fnal.gov
131.225.8.178
fnsg01 judy
```

When writing a `.rhosts` file on a UNIX host to allow access via remote commands (known as *r-commands*[2]) from a VMS cluster, you must include an entry for *each* cluster member and username from which you might access the UNIX host.

The following command line is an example of the command to copy a `FINGER.PLN` file from an account on FNALA to the file `.plan` in the same person's account on the UNIX node FNSG01. In order for this example to work, there must be a file named `.rhosts` on FNSG01 containing a line `fnala.fnal.gov`, and the username must be the same in both places.

$ RCP FINGER.PLN FNSG01::.PLAN

where **$** is the VMS prompt.

1. Use **nslookup** ***node*** to get the number. On VMS, you must enter **MULTINET NSLOOKUP** ***hostname***.
2. r-commands execute on remote hosts, and use `.rhosts` for authentication. They are variants of UNIX commands without the "r", and include for example **rsh** (**r**emote **sh**) and **rlogin** (**r**emote **login**).

13.2 Logging in to Other Systems

13.2.1 telnet

telnet is a remote terminal protocol which allows you to establish a connection to a login server at a remote site. **telnet** in the UNIX world is analogous to **`SET HOST`** in the VMS world. You can connect to other UNIX machines, or you can connect to VMS machines if the appropriate software is installed on the system (e.g., with **MultiNet** for which Fermilab has a site license). The format is:

`% ` **`telnet [`*`host`*`]`**

If you include the host, you are prompted for the username[1] and password. If you do not include the host, you go into command mode. You can enter **`help`** at this point for help on the commands.

13.2.2 rlogin

Another way of logging into a remote computer from another system on the network is by use of the **rlogin** utility. One difference between **telnet** and **rlogin** is that if the communications package is configured for it and you create an appropriate `.rhosts` file (see section 13.1.3), you can log in to the remote system using **rlogin** without typing the password. The format from a UNIX machine is:

`% ` **`rlogin`** ***rhost*** **`[-l`** ***username*****`]`**

where:

rhost	is the remote host to which you want to log in
username	is the username if the name is different on the remote system than the originating system

From a VMS machine, the format is:

`$ RLOGIN /USERNAME=`*`username rhost`*

AFS If your system uses AFS and you experience permission problems after using **rlogin**, run **`pagsh`** followed by **`klog`**. Please see the discussion of these commands in section 7.3.1.

13.3 Executing Commands Remotely: rsh

You can execute commands on a remote system and have the output displayed on your terminal with **rsh**. **rsh** stands for **r**emote **sh**ell. In order for this to work, there must be an appropriate `.rhosts` file on the remote machine (see section 13.1.3).

The format is:

`% ` **`rsh`** ***rhost*** **`[-l`** ***username*****`]`** ***command***

where:

1. Some systems may not prompt you for a username; they will use your current login id. If you need to log in with a different username, enter a carriage return at the password prompt. Then the system will prompt you for a username.

rhost is the remote host on which the command is to be executed

username is the username on the remote system if different than the originating system

command is the command to be executed on the remote system

The command may be a single command or a shell script to be executed on the remote system. Metacharacters in the command must be quoted for proper interpretation on the remote machine.

The **MultiNet** implementation on VMS is:

$ RSHELL[/INPUT=*infile*][/OUTPUT=*outfile*][/USERNAME=*username*]– ***hostname commandline***

VMS

The **MultiNet** implementation does not handle metacharacters like the UNIX implementation does.

The **/INPUT** and **/OUTPUT** options in the **MultiNet** implementation allow you to execute a task on a UNIX machine which reads and writes files on VMS.

If the command is omitted, you will be logged in to the remote host using **rlogin**.