

# NFT

By Alice Potts

UCRL-PRES-203059

# What is it?

- Network File Transport (NFT)
- File transfer utility
- Available on open and secure LC production machines
- Dedicated server provides persistent, passwordless transfers with job tracking
- <http://www.llnl.gov/LCdocs/nft/>
- Man page
- Help command

# Architecture

- Dedicated server for persistence and job tracking
- Uses FTP daemons on sending and receiving machines to carry out actual file transfers
- All file transfers use FTP binary mode
- Automatically uses parallel file transfer when appropriate (based on availability and file size)
- Limit: 512Gbyte file size (use HTAR to store and manage large archive files)

# How To Run

- Need account on open or secure LC High Performance Storage System (HPSS)
- Account on open or secure LC production machine (e.g. blue, mcr, lilac)
- Usage: `nft [-i] [<commands>]`
- `-i` option causes NFT to ignore any `.nftrc` file in your home directory
- The `.nftrc` file can contain any NFT commands (example: `session new`)

# Example 1

```
apotts@lucy:/g/g0/apotts> nft "dir"
drwxr-s---    2 apotts    apotts          512 Apr 24 2003  Documents
drwxr-s---    2 apotts    storage         512 Oct  8 15:15  Projects
nft> quit
apotts@lucy:/g/g0/apotts>
```

```
apotts@lucy:/g/g0/apotts> nft "dir ; block ; quit"
drwxr-s---    2 apotts    apotts          512 Apr 24 2003  Documents
drwxr-s---    2 apotts    storage         512 Oct  8 15:15  Projects
apotts@lucy:/g/g0/apotts>
```

The block command prevents NFT from executing any more commands until all previously entered synchronous jobs have completed.

# Example 2

```
apotts@lucy:/g/g0/apotts> cat nfttest
dir
block
apotts@lucy:/g/g0/apotts> nft
nft> source nfttest
drwxr-s---  2 apotts  apotts      512 Apr 24 2003  Documents
drwxr-s---  2 apotts  storage    512 Oct  8 15:15  Projects
nft> quit
apotts@lucy:/g/g0/apotts>
```

The source command causes NFT to read and execute the NFT commands contained in the specified text file.

```
apotts@lucy:/g/g0/apotts> nft < nfttest
drwxr-s---  2 apotts  apotts      512 Apr 24 2003  Documents
drwxr-s---  2 apotts  storage    512 Oct  8 15:15  Projects
apotts@lucy:/g/g0/apotts>
```

# Example 3

```
apotts@lucy:/g/g0/apotts> nft
nft> dir
drwxr-s---    2 apotts    apotts          512 Apr 24 2003  Documents
drwxr-s---    2 apotts    storage          512 Oct  8 15:15  Projects
nft> dir :
drwxr-x---   15 apotts    storage        4096 Feb  3 15:58  Projects
-rw-r-----    1 apotts    apotts           39 Feb  4 13:05  nfttest
-rw-r-----    1 apotts    apotts           18 Feb  4 13:06  nfttest2
-rw-r-----    1 apotts    apotts           14 Feb  4 13:06  nfttest3
nft> put nfttest*
4/0. 39 bytes sent in 0.00 seconds (39.00 Kbytes/s) from
    /g/g0/apotts/nfttest to ~/nfttest
5/0. 18 bytes sent in 0.00 seconds (18.00 Kbytes/s) from
    /g/g0/apotts/nfttest2 to ~/nfttest2
6/0. 14 bytes sent in 0.00 seconds (14.00 Kbytes/s) from
    /g/g0/apotts/nfttest3 to ~/nfttest3
3.0. 3 entries copied (aggregate 0.01 Kbytes/s) /g/g0/apotts/nfttest*
nft> quit
apotts@lucy:/g/g0/apotts>
```

# Pathname Syntax

Example (host is mcr24): `mcr24:/Projects/*.c`

Example (host is local machine): `*.c`

Example (host is storage\*\*\*): `/Dir/file[0-9]`

Example (host is storage\*\*\*): `../Bproj/{f1,fx}`

Example (host is storage\*\*\*): `/Dir/{file?,xyz?}`

\*\*\* Except for commands put, get, lcd and log which have specific implied host specifications.



# Directory Commands

## Syntax:

<code>cd</code>	<code>[host][pathname]</code>	<i>changes directory on specified host</i>
<code>cdup</code>	<code>[host]</code>	<i>changes to parent directory on specified host</i>
<code>lcd</code>	<code>[pathname]</code>	<i>changes directory on local host machine only</i>
<code>pwd</code>	<code>[-a]   [host]</code>	<i>displays current working directory on specified host</i>

## Examples:

<code>nft&gt; cd</code>		<i>changes to your home directory on storage</i>
<code>remote wd is ~</code>		
<code>nft&gt; cd subdirectory</code>		<i>changes to subdirectory in current directory on storage</i>
<code>remote wd is ~/subdirectory</code>		
<code>nft&gt; cd mcr24:subdirectory</code>		<i>changes to subdirectory in current directory on mcr24</i>
<code>remote host mcr24: wd is ~/subdirectory</code>		

# Directory Commands continued

nft> lcd subdirectory *changes to subdirectory in current directory on local host*

local wd is /g/g0/apotts/subdirectory

nft> pwd *displays current working directory on storage*

remote wd is ~/subdirectory

nft> pwd : *displays current working directory on local host*

local wd is /g/g0/apotts/subdirectory

nft> pwd -a *displays current working directory on all hosts*

local wd is /g/g0/apotts/subdirectory

remote wd is ~/subdirectory

mcr24: wd is ~/subdirectory

# Directory Commands continued

Following are equivalent:

cd :..

cdup :

lcd ..

**Note:** If you specify a nonexistent pathname, you will get the usual confirmation message before receiving an error message. Subsequent use of PWD will also report the nonexistent directory without complaint. You must overtly reset the working directory with another use of CD to a real location.

# Environment Commands

## Syntax:

help [command]	<i>List commands or displays help text for specified command</i>
clobber	<i>Allows overwriting of existing files during file transfers</i>
noclobber	<i>Prevents overwriting of existing files during file transfers (default)</i>
log pathname	<i>NFT begins recording all your input to and output messages from NFT to the specified log file on your local machine</i>
clog	<i>Closes the NFT log file</i>
session nn   new	<i>Closes your former NFT session (session 0 is the default) and opens either the specified session number nn (0 to 99) or a new unused session number. NFT numbers jobs in each session with an increasing sequence that ignores all other sessions (so each session may have an unrelated job numbered 13)</i>

# Environment Commands continued

## Syntax:

sync

*NFT runs all subsequent commands strictly in series (default), preserving order in which you submitted them*

async

*NFT runs all subsequent commands in parallel, allowing any job to run in any order as soon as resources are available*

open [host]

*Changes remote host from storage to specified host*

close

*Restores NFT's remote host to default (storage)*

verbose mask

*Specifies which changes of job state NFT will report to you (default is 76, which displays errors, aborts, and transfer statistics)*

status

*Displays NFT "environment variables"*

quit

*Quit NFT. You can quit NFT before your "jobs" have completed and they will continue to run. You can come back and check the status of the jobs using the RPT command.*

# Environment Commands continued

## Example:

```
nft> status
  Connected to storage as apotts.
  Session: 0.
  Verbose: 76 (decimal), 4c
(hexidecimal).
  Clobber: no.
  Job Execution mode: Synchronous.
  Group construct: closed.
  Input from: standard-in.
  Output to standard-out: yes.
  Output to log file: no.
nft>
```

# Environment Commands continued

## Example (synchronous mode):

```
nft> put file  
dir
```

Since default mode is synchronous, the 'dir' command will not be executed until the 'put' command is finished.

## Example (session stealing):

Start an instance of NFT in one window (by default it will get session 0.)

Start a second instance of NFT in another window (it also will by default get session 0.) It has just "stolen" session 0 from the first instance. If you try and use the first instance of NFT you will get an error.

You can prevent this by putting the SESSION NEW command into your .nft.rc file in your home directory. You will however have to remember the session number it gives you if you wish to come back later and check on job statuses.

# Environment Commands continued

Example (asynchronous mode):

```
nft> async  
nft> mkdir temp  
nft> cd temp  
nft> put file
```

Dangerous! New directory 'temp' may not exist when cd or put commands are performed!



# Directory Manipulation Commands

## Syntax:

<code>mkdir [host]pathname</code>	<i>Creates specified directory on specified host</i>
<code>rmdir [-R] [host]pathname</code>	<i>Removes specified empty directory on specified host, or alternatively recursively (-R) removes specified directory and its contents</i>
<code>dir [-options] [host][pathname]</code>	<i>Lists contents of specified directory in long format</i>
<code>ls [-options] [host][pathname]</code>	<i>Lists contents of specified directory in short format (unless you add detail with options)</i>
<code>chmod [-R] rights [host]pathname</code>	<i>Changes the access rights or “mode” of the specified file or directory (only octal number allowed for rights). Option -R specifies recursion.</i>
<code>chgrp [-R] group [host]pathname</code>	<i>Changes the group membership of the specified file or directory. Option -R specifies recursion.</i>

# Directory Manipulation Commands continued

## Example:

```
nft> dir
drwxr-s---    2 apotts    apotts          512 Apr 24 2003  Documents
drwxr-s---    2 apotts    storage         512 Oct  8 15:15  Projects
nft> dir Projects
-rw-r-----    1 apotts    storage    9121280 Jan 29 23:38
    nft20040128.tar
-rw-r-----    1 apotts    storage    9123840 Jan 29 23:40
    nft20040129.tar
nft> mkdir TestDir
nft> dir
drwxr-s---    2 apotts    apotts          512 Apr 24 2003  Documents
drwxr-s---    2 apotts    storage         512 Oct  8 15:15  Projects
drwxr-s---    2 apotts    apotts          512 Feb  9 15:23  TestDir
```

# Directory Manipulation Commands continued

## Example:

```
nft> dir :  
drwxr-x--- 15 apotts storage 4096 Feb 3 15:58 Projects  
nft> mkdir :testdir  
nft> dir :  
drwxr-x--- 15 apotts storage 4096 Feb 3 15:58 Projects  
drwxr-xr-x 2 apotts apotts 4096 Feb 9 15:24 testdir  
nft> rmdir TestDir  
8.0. 1 entry removed ~/TestDir  
nft>
```

# File Manipulation Commands

## Syntax:

`put [-dR] sourcepath [sinkpath]`

*Copies specified file(s) from local machine to storage.*

*Note: host is not specified for source or sink. If sinkpath is not specified, the sourcepath is duplicated as the sinkpath. For example, for “put a/b/c/file” then path “a/b/c” needs to exist in current storage directory*

`get [-dR] sourcepath [sinkpath]`

*Copies specified file(s) from storage to local machine.*

*Note: host is not specified for source or sink. If sinkpath is not specified, the sourcepath is duplicated as the sinkpath. For example, for “get a/b/c/file” then path “a/b/c” needs to exist in current local directory*

`cp [-dR] [host]sourcepath [host][sinkpath]`

*Copies specified file(s) from source host to sink host.*

`-d`      *deletes source file after successful transfer of file*

`-R`      *specifies recursion*

# File Manipulation Commands

## Syntax:

ren[ame] [host]sourcepath [host]sinkpath

*Renames the file(s) specified by sourcepath to the name specified by sinkpath. Source and sink host specifications must be the same.*

del[ete] [-R] [host]pathname

*Deletes the file(s) specified by pathname from the specified host. Option -R specifies recursion. Note: delete and rmdir are equivalent commands when -R option is specified*

# File Manipulation Commands continued

## Examples:

```
nft> put nfttest*
4/0. 39 bytes sent in 0.00 seconds (39.00 Kbytes/s) from /g/g0/apotts/nfttest to
~/nfttest
5/0. 18 bytes sent in 0.00 seconds (18.00 Kbytes/s) from /g/g0/apotts/nfttest2 to
~/nfttest2
6/0. 14 bytes sent in 0.00 seconds (14.00 Kbytes/s) from /g/g0/apotts/nfttest3 to
~/nfttest3
3.0. 3 entries copied (aggregate 0.01 Kbytes/s) /g/g0/apotts/nfttest*
nft> dir -R :testdir
.:
-rw-r-----  1 apotts  apotts      5102 Feb  9 15:40 file1
drwxr-x---   2 apotts  apotts      4096 Feb  9 15:40 subdir

./subdir:
-rw-r-----  1 apotts  apotts      5102 Feb  9 15:40 file2
nft> put -R testdir
10/0. 5102 bytes sent in 0.00 seconds (5.10 Mbytes/s) from /g/g0/apotts/testdir/file1
to ~/testdir/file1
11/0. 5102 bytes sent in 0.00 seconds (5.10 Mbytes/s) from
/g/g0/apotts/testdir/subdir/file2 to ~/testdir/subdir/file2
9.0. 2 entries copied (aggregate 1.47 Kbytes/s) /g/g0/apotts/testdir
```

# File Manipulation Commands

## continued

### Examples:

```
nft> cp -d :testdir/file1
23.0. 5102 bytes sent in 0.00 seconds (5.10 Mbytes/s) from /g/g0/apotts/testdir/file1
to ~/file1
23.0. 1 entry copied /g/g0/apotts/testdir/file1
nft> dir :testdir
drwxr-x---  2 apotts  apotts      4096 Feb  9 15:40 subdir
nft> mkdir testdir
nft> ren file1 testdir
27.0. 1 entry renamed ~/file1
nft> dir
drwxr-s---  2 apotts  apotts      512 Apr 24 2003  Documents
drwxr-s---  2 apotts  storage     512 Oct  8 15:15  Projects
drwxr-s---  2 apotts  apotts      512 Feb  9 16:22  testdir
nft> dir testdir
-rw-r-----  1 apotts  apotts     5102 Feb  9 16:20  file1
nft> del -R testdir
30.0. 2 entries deleted ~/testdir
```

# Job Reporting and Aborting Commands

## Syntax:

`abt [[n[-m]] | [-opt]]`

*Aborts your most recent NFT request ("job") by default, or aborts a specific job number n, or the range of jobs n-m inclusive, or all members of the specified job class option opt (-a, -l, -h, -x). Only incomplete jobs can be aborted.*

`clr [[n[-m]] | [-opt]]`

*Clears the status for your most recent NFT request ("job") by default, or clears a specific job number n, or the range of jobs n-m inclusive, or all members of the specified job class option opt (-a, -c, -o, -e, -k). Only the status of completed jobs (successfully or unsuccessfully) can be cleared.*

`rpt [[n[-m]] | [-opt]]`

*Reports the current status of your most recent NFT request ("job") by default, or the status of the specific job n, or the range of jobs n-m inclusive, or all members of the specified job class option opt (-a, -l, h, -x, -c, -o, -e, -k).*

- a all jobs in current session
- i incomplete jobs in current session (held and active)
- h held jobs in current session (incomplete jobs in the scheduling queue)
- x active jobs in current session (incomplete jobs currently running)
- c complete jobs in current session (that have successfully or unsuccessfully completed running)
- o successfully completed (okay) jobs in current session
- e unsuccessfully completed (error) jobs in current session
- k aborted jobs in current session



# Job Reporting and Aborting Commands Continued

## Example:

```
nft> dir
drwxr-s---  2 apotts  apotts      512 Apr 24 2003  Documents
drwxr-s---  2 apotts  storage     512 Oct  8 15:15  Projects
nft> rpt -a
1.0. done. ~
nft> clr -a
nft> rpt -a
*** no jobs.
nft> put nfttest*
rpt -a
1.0.  Waiting for 3 subordinate jobs to finish.
2/0. start /g/g0/apotts/nfttest
3/0. start /g/g0/apotts/nfttest2
4/0. start /g/g0/apotts/nfttest3
abt -a
1.0. aborted ~/nfttest*
2/0. aborted ~/nfttest
4/0. aborted ~/nfttest3
3/0. aborted ~/nfttest2
nft> dir
abt
5.0. aborted ~
nft>
```

# Job Reporting and Aborting Commands Continued

## Example:

**Important Note:** Abt command without any options specified aborts last command submitted. If you perform a file transfer command (put, get, cp) with more than one file specified (with wild card specification), you may not get the behavior you expect!!!

```
nft> put nfttest*
abt
6.0. aborted ~/nfttest*
nft> put nfttest*
rpt -a
1.0. aborted ~/nfttest*
2/0. aborted ~/nfttest
3/0. aborted ~/nfttest2
4/0. aborted ~/nfttest3
5.0. aborted ~
6.0. aborted ~/nfttest*
7.0. Waiting for 3 subordinate jobs to finish.
8/0. start /g/g0/apotts/nfttest
9/0. start /g/g0/apotts/nfttest2
10/0. start /g/g0/apotts/nfttest3
abt
8/0. 4 bytes sent in 0.00 seconds (4.00 Kbytes/s) from /g/g0/apotts/nfttest to ~/nfttest
10/0. aborted ~/nfttest3
9/0. 18 bytes sent in 0.00 seconds (18.00 Kbytes/s) from /g/g0/apotts/nfttest2 to ~/nfttest2
7.0. error. storage.llnl.gov: 2 entries copied, 1 aborted (aggregate 0 Kbytes/s) /g/g0/apotts/nfttest*
nft>
```