

Checking File Integrity

Category: File Transfers

Lou2 Note:

This article is currently being edited to reflect the [changes to Lou2](#) which take effect on December 6, 2012. A finalized version will be posted soon.

It is a good practice to check that your data are complete and accurate before and after a file transfer. A common way for checking data integrity is to compute a checksum of the data.

The easiest way to verify the integrity of file transfers is to use the NAS-developed [Shift](#) tool for the transfer with the `--verify` option enabled. Shift will automatically checksum the data at the source and destination to detect corruption as part of the transfer. If corruption is detected, partial file transfers/checksums will be performed until the corruption is rectified.

For example:

```
pfe20% shiftc --verify $HOME/foo /nobackupp2/username
lou% shiftc --verify /nobackupp2/username/foo $HOME
your_localhost% sup shiftc --verify foo pfe:
```

Besides Shift, there are multiple algorithms and programs that one can use for computing a checksum. A good checksum algorithm will yield a different result with high probability when the data is accidentally corrupted. If the checksums obtained before and after the transfer match, the data is almost certainly not corrupted.

On NAS HECC systems, the following programs are available:

sum

Computes a checksum using BSD sum or System V sum algorithm; also counts the number of blocks (1 KB-block or 512 B-block) in a file

cksum

Computes a cyclic redundancy check (CRC) checksum; also counts the number of bytes in a file

md5sum

Computes a 128-bit MD5 checksum which is represented by a 32-character hexadecimal number

For example:

```
%ls -l foo
-rw----- 1 username groupid 67358 Nov 15 11:49 foo
```

```
%sum foo
50063    66
```

```
%cksum foo
269056887 67358 foo
```

```
%md5sum foo
cfe0fc62607e9dc6ea0c231982316b75  foo
```

md5sum is more reliable than **sum** or **cksum** for detecting accidental file corruption, as the chances of accidentally having two files with identical MD5 checksum are extremely small. It is installed by default in most Unix, Linux, and Unix-like operating systems. Users are recommended to compute the **md5sum** of a file before and after the transfer.

The following example shows that the file *foo* is complete and accurate after the transfer based on its **md5sum**.

```
pfe20% md5sum foo
cfe0fc62607e9dc6ea0c231982316b75  foo
```

```
pfe20% scp foo local_username@your_localhost:
```

```
your_localhost%md5sum foo
cfe0fc62607e9dc6ea0c231982316b75  foo
```

See **sum**, **cksum**, **msum**, and **md5sum** man pages for more information.

See [Using mtar to Create or Extract Tar Files on Lustre](#) for more information on **mtar**.

Article ID: 243

Last updated: 17 Dec, 2012

Data Storage & Transfer -> File Transfers -> Checking File Integrity

<http://www.nas.nasa.gov/hecc/support/kb/entry/243/?ajax=1>