

Campaign Storage file system

NCAR Campaign Storage is a resource for medium-term storage of project data, typically for three to five years, by NCAR labs and universities that have project allocations.

Campaign Storage is accessible a number of ways that are described below:

- through the Globus web and command-line interfaces
- from the data-access nodes, for Globus transfers and managing data holdings
- from the Casper cluster to facilitate data analysis and visualization workflows

Files stored on this system are not backed up. Users are responsible for replicating any data they feel should be stored at an additional location as backup.

Files that are deleted or overwritten cannot be recovered.

Page contents

- [Globus transfers](#)
- [Data-access nodes](#)
- [Casper use](#)
- [Data retention policy](#)
- [Allocations](#)
 - [NCAR labs](#)
 - [Universities](#)
- [Reports](#)
- [Automated data compression](#)
 - [Tool and accounting behavior](#)
 - [Compression details](#)
 - [Examples: commands and output](#)

Globus transfers

The Globus mapped collection established for the file system is **NCAR Campaign Storage**. How to make transfers to and from that collection is documented here:

[Globus file transfers](#)

How to make transfers using the command line interface also is covered in detail in this tutorial:

[Using Globus v5 at NCAR \(tutorial\)](#)

CAVEAT: The Globus interface for transferring data does not handle symbolic links and does not create symbolic links on a destination endpoint.

Data-access nodes

The Campaign Storage file system is mounted on the [data-access nodes](#) as **/glade/campaign** to:

- enable users to manage file and directory permissions using POSIX commands.
- facilitate transfers of *small files* to and from GLADE spaces such as **/glade/scratch** and **/glade/work**.

Casper use

The Campaign Storage file system can be accessed from the Casper cluster as **/glade/campaign** so users are able to:

- read and write data directly from their data analysis and visualization workflows.
- submit batch scripts to migrate data to the Campaign Storage resource.

Data retention policy

Campaign Storage is designed to provide medium-term storage for project data, typically for three to five years. While data will not be purged automatically after five years, retaining data longer will reduce the capacity for storing additional, new data. Users are expected to monitor their holdings, remove files that are no longer needed, and move necessary data to other storage options for longer-term preservation.

NCAR researchers are expected to collaborate with CISL's Digital Asset Services Hub (log in to [Sundog](#)) to develop data migration plans for storage needs that exceed five years.

University researchers are expected to transfer their project data to their home institutions or other alternative storage repositories within one year of their NSF grant expiring. CISL will not award storage space for researchers to carry data forward from one grant to another.

Allocations

NCAR labs

Each NCAR lab has an allocation of Campaign Storage space and the labs manage how those allocations are used.

Users who have questions related to lab allocations should contact the lab's own allocation representative.

Universities

University users can request Campaign Storage space through the [NCAR Resource Allocation System](#) as supplements to their project allocations. Requests must include detailed justification for the amount of space requested.

Because NCAR is not currently funded to provide long-term data storage services to the university community, university users' requests for these allocations are prioritized based on the following factors.

Higher priority is given to requests if:

- You have an active project, supported by an active NSF award, for using Cheyenne.
- Your request is for a period of no more than three (3) months and to support migrating of your data to your home institution.

Lower priority is given to requests if:

- Your need relates to satisfying external requirements or promises – to a publisher or agency, for example – to retain data for extended periods.

Any data requiring longer storage should be migrated to your home institution or to another appropriate repository.

Reports

The Systems Accounting Manager ([SAM](#)) provides overall summary information about the use of Campaign Storage allocations and other allocations.

CISL is developing additional tools for use in allocation management.

Automated data compression

Campaign Storage has an automated data compression feature for long-duration data sets. Our compression policy targets files that are 180 days old or older and 100MB in size or larger for "z" compression using [IBM Spectrum Scale file system mechanisms](#) (details below).

The action is transparent to the user – that is, no changes to metadata timestamps or reported size occur, and subsequent reads of the data proceed as usual. During a read, the compressed data are sent to the file system client and then transparently uncompressed for application use.

Tool and accounting behavior

The number of blocks reported consumed by the file will change. Note the following tool-specific behavior:

Tool	Output
ls -l	shows original (uncompressed) file size
stat	shows compressed number of blocks, original file size
du	shows compressed file sizes. du --apparent-size shows original (uncompressed) size
gladequota	shows project space usage <i>after</i> compression, as do SAM reports

Individual data sets can be excluded from the compression algorithm, if necessary. To discuss this option, please submit a request through the [NCAR Research Computing help desk](#).

Compression details

When a file is considered for compression, the algorithm tests compression of chunks of the file. If the realized compression efficiency of a given chunk is at least 10%, it is then stored compressed on disk.

The compression status of a file can be queried via the **mmlsattr** command. Follow this example:

```
/usr/lpp/mmfs/bin/mmlsattr -L filename
```

A file has been compressed if the **mmlsattr** output:

- includes "Misc attributes: COMPRESSION" – which indicates that the file was targeted for compression.
- does not include "flags: illCompressed" – which indicates the file was targeted or deferred but not yet compressed.

Several output examples are provided below.

User-driven manual compression is also possible before the automated policy is triggered if desired via the **mmchattr** command:

```
/usr/lpp/mmfs/bin/mmchattr [-I defer] compression z filename
```

- z: best compression (Campaign Storage default)
- If deferred, the file will be compressed during the next Campaign Storage policy execution rather than instantly.

Examples: commands and output

Run du, ls, stat for an original uncompressed file

```
$ du -h 1GB.dat && du -h --apparent-size 1GB.dat && ls -lh 1GB.dat && stat 1GB.dat
1000M 1GB.dat
1000M 1GB.dat
-rw-r-----+ 1 benkirk csg 1000M Mar  9 10:08 1GB.dat
  File: '1GB.dat'
  Size: 1048576000 Blocks: 2048000    IO Block: 8388608 regular file
Device: 2dh/45d Inode: 1006073884  Links: 1
Access: (0640/-rw-r-----)  Uid: (38057/ benkirk)   Gid: ( 1564/      csg)
Access: 2022-03-09 10:08:00.479563000 -0700
Modify: 2022-03-09 10:08:01.486585235 -0700
Change: 2022-03-09 10:08:01.486585235 -0700
Birth: -
```

Request compression manually

```
$ /usr/lpp/mmfs/bin/mmchattr --compression z 1GB.dat
```

Run du, ls, stat for a compressed file (note that metadata dates are not changed)

```
$ du -h 1GB.dat && du -h --apparent-size 1GB.dat && ls -lh 1GB.dat && stat 1GB.dat
104M 1GB.dat
1000M 1GB.dat
-rw-r-----+ 1 benkirk csg 1000M Mar  9 10:08 1GB.dat
  File: '1GB.dat'
  Size: 1048576000 Blocks: 212992    IO Block: 8388608 regular file
Device: 2dh/45d Inode: 1006073884  Links: 1
Access: (0640/-rw-r-----)  Uid: (38057/ benkirk)   Gid: ( 1564/      csg)
Access: 2022-03-09 10:08:00.479563000 -0700
Modify: 2022-03-09 10:08:01.486585235 -0700
Change: 2022-03-09 10:08:01.486585235 -0700
Birth: -
```

List file attributes to verify a compressed file

```
$ /usr/lpp/mmfs/bin/mmlsattr -L 1GB.dat
file name:          1GB.dat
metadata replication: 2 max 2
data replication:    1 max 2
immutable:          no
appendOnly:         no
flags:
storage pool name:   DATA
fileset name:        csg
snapshot name:
creation time:       Wed Mar  9 10:08:00 2022
Misc attributes:     ARCHIVE COMPRESSION (library z)
Encrypted:           no
```

Request deferred compression of a file

```
$ /usr/lpp/mmfs/bin/mmchattr -I defer --compression z 1GB_deferred.dat
```

Note that deferred compression is recommended when manually requesting compression for a large number of files. In this case, the **mmchattr** command will return immediately, and the file compression will occur at the next regularly scheduled system interval.

List file attributes (note that "illcompressed" indicates the compression has not yet been applied)

```
$ /usr/lpp/mmfs/bin/mmlsattr -L 1GB_deferred.dat
file name:          1GB_deferred.dat
metadata replication: 2 max 2
data replication:    1 max 2
immutable:          no
appendOnly:         no
flags:              illcompressed
storage pool name:   DATA
fileset name:        csg
snapshot name:
creation time:       Wed Mar  9 10:07:17 2022
Misc attributes:     ARCHIVE COMPRESSION (library z)
Encrypted:           no
```