

Table of Contents

CernLbdDiskManagement	1
This is the LHCb CERN LBD group page describing the management of locally available disk space.....	1
z5:cernlbd.....	1
Group afs space.....	1
- Setting more Afs permissions.....	1
lhcbt3.....	2
- removal notes:.....	2
- Files you can't remove.....	3

CernLbdDiskManagement

This is the LHCb CERN LBD group page describing the management of locally available disk space

Our resources are advertised as volatile and maintained with a fair-use policy. If space becomes limited people should be contacted to voluntarily remove old files, then the manager can if really necessary arbitrarily remove changes on top of that, since the system is advertised as volatile.

The instructions here are in addition to the instructions on CernLbdDiskSpace.

The scripts for management, and other things are kept in backed up afs space:

`/afs/cern.ch/lhcb/group/lbd/scripts/`, which is softlinked to `/afs/cern.ch/project/lbcern/scripts`

z5:cernlbd

z5:cernlbd holds the access list for the 1TB disk space.

```
pts mem z5:cernlbd #see the member list
pts adduser <someuser> z5:cernlbd #add someuser to the list
pts removeuser <someuser> z5:cernlbd #remove someone from the list
```

Note that at the moment only JoelClosier can do this. Although perhaps Loic Brarda and Hubert Degaudenzi are also able to, to be checked.

Group afs space

1TB of space is mounted in 10 volumes in: `/afs/cern.ch/lhcb/project/lbcern`

```
afs_admin create -u rlambert -q 100G /afs/cern.ch/project/lbcern/vol2 s.lbcern.vol2 #create a volume
afs_admin set_quota /afs/cern.ch/project/lbcern/vol1 100G #change the quota of a volume
fs sa /afs/cern.ch/project/lbcern/vol2 z5:cernlbd write #give the right permissions
afs-admin list_project lbcern #to list the amount of space for the project
python /afs/cern.ch/project/lbcern/scripts/check1TB.py #to check the usage and the usage per user
```

z5:cernlbd is the access list for this volume. All users should make directories with their user name. If there is some top-level directory without a user name it should be moved/deleted.

For more on acl see here http://docs.openafs.org/Reference/1/fs_setacl.html

- Setting more Afs permissions

give the permission to change permissions to a user for their own directory

```
fs sa /afs/cern.ch/project/lbcern/vol*/<user> <user> rlidwka
```

To propagate permissions from one place to another use:

```
fs copy_acl <one_place> <another>
```

Directories created within a parent directory inherit the parent's permissions

Then if they want to let someone else See/read/write in the directories, the upper directories need to be 'lookup'-able

```
fs sa /afs/cern.ch/project/lbcern/vol*/<user>/<something> <user-or-group> write
fs sa /afs/cern.ch/project/lbcern/vol*/<user> <user-or-group> l
fs sa /afs/cern.ch/project/lbcern/vol* <user-or-group> l
```

e.g. <user-or-group> cern:z5

Since directories created within a parent directory inherit the parent's permissions, one quick way to apply permissions to subdirectories is to temporarily rename them, then copy them recursively.

```
mv /afs/cern.ch/project/lbcern/vol*/<user>/<something> /afs/cern.ch/project/lbcern/vol*/<user>/_t
cp -r /afs/cern.ch/project/lbcern/vol*/<user>/_tmp /afs/cern.ch/project/lbcern/vol*/<user>/<some
```

To set a tree with new acl you can do :

```
find <mytopdir> -type d -exec fs sa {} cern:z5 read \;
```

lhcbt3

There are two types of files which can be on the lhcbt3: user files, and central files. In general there are two spying scripts, `check30TB.py` and `castorQuota.py`, each with many options to look in specific directories, etc.

- `check30TB.py`: poles the castor dump of all files from last night to categorize all files which were on the lhcbt3 then
- `check30TB_slow.py`: uses `nsls -Rl` and `stager_qry` to find files on lhcbt3, this is up-to-date but much slower
- `castorQuota.py` uses `nsls -Rl` and `stager_qry` to categorise files from a given user, again this is up-to-date but slow

Each script has a `--help` flag to tell you about the other options

- removal notes:

Once you have the list of files to remove, remove them individually with:

```
stager_rm -M <existing castor file> -S lhcbt3
```

Or remove a big list with:

```
stager_rm -f <file with a list in it> -S lhcbt3
```

or similarly:

```
/afs/cern.ch/project/lbcern/scripts/lhcbt3_rm.py <username or directory or file with a list in it>
```

Removing a user file potentially deletes the file forever, so it's nice to get the users to do that themselves, but any centrally produced or grid-resident file is by definition backed-up and can be removed arbitrarily.

- Files you can't remove

If a user stages a file from a different user's home directory, then they won't have the ability to unstage them, neither will the owner unless they are also on the access list for lhcbt3, so they are stuck. Only Joel can remove them, or the castor support who certainly could do it, but if you ask them will reply with the annoying:

Unfortunately, as service policy, CASTOR Operations does not preform delete operations on files.

-- RobLambert - 19-Nov-2010

This topic: Main > CernLbdDiskManagement

Topic revision: r11 - 2011-08-16 - RobLambert



Copyright &© 2008-2019 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.

Ideas, requests, problems regarding TWiki? Send feedback