

Scenario: User archives a file and has the disk copy marked as available for LRU garbage collection.

1. User creates a directory within the EOS namespace for files that are to go to tape.

```
eos mkdir /eos/tape_dir
```

2. The user associates the appropriate layouts and workflow with the directory.

```
# Disk files should start out as having 2 replicas  
eos attr set default=replica /eos/tape_dir
```

```
# The policy to migrate files and keep them in the disk pool
```

```
# if a default policy is used and a file is written, we call a  
URL with <params> defined by attributes explained later  
attr set sys.workflow.default  
= "onclosew:*/*:call:http://cta.cern.ch?<params>" /eos/tape_dir
```

```
# if the migration policy is used by the cta user and a file  
was read with a successful close, we register it in the tape  
location(space) and drop it from the default locations  
attr set sys.workflow.migrate  
= "oncloser:cta/cta:register:tape,oncloser:cta/cta:drop:default "  
/eos/tape_dir
```

```
# if a file was written by the cta user and the recall policy  
was used, nothing has to be done - one can just omit that!  
attr set sys.workflow.recall = "onclosew:cta/cta:none "  
/eos/tape
```

```
# Add a condition on the LRU policies that files cannot be  
garbage collected if they don't have a location in the tape  
space
```

TO BE DECIDED

3. The user associates the destination tape storage class with the directory by setting an extended attribute.

```
eos attr set tape_storage_class=daq_raw_2_copies
```

4. User runs a command-line tool to synchronously copy a file into EOSTAPE with no overwrite:

```
eos cp -k local_file /eos/murrayc3/tape_dir/tape_file
```

5. The command-line tool connects to the xrootd server with the mgm plugin and requests the destination file `tape_file` be opened for writing.
6. The mgm plugin asserts the destination file does not already exist.
7. The mgm plugin redirects the client to a disk FST.
8. The client writes the data and closes the file on the disk FST
9. The disk FST notifies the mgm of the close for write.
10. The mgm plugin connects to the CTA xrootd front-end and queues a request to archive the file. The request includes EOS instance, EOS inode, file size, file checksum and destination tape storage class.
11. The CTA xrootd front-end stores the archive request in the CTA object store.
12. A tape server pulls the need to mount a tape from the CTA object store and does so.
13. The tape server pulls the archive request from the CTA object store.
14. The tape server connects to the xrootd server with the mgm plugin and requests the source disk file based on inode be opened for reading.
15. The mgm redirects the tape server to the disk FST.
16. The tape server opens the file on the disk FST for reading.
17. The tape server reads blocks from the file on the disk FST and writes them to the tape.
18. The tape server closes the file on the disk FST.
19. After enough files have been written to tape, the tape server synchronously flushes the tape drive cache in order to guarantee that all of the files since the previous flush are now safely on tape.
20. The tape drive notifies the mgm that the file is safely on tape.

***xrdfs query <>?eos.workflow=migrate***

21. The mgm marks the file as being on tape.

EOS currently has a queue in the mgm for file conversions. A second queue for to run workflows can be added. Workflows would stay in the queue until they are successful or a retry policy expires them. This boils down to move virtual files

between directories representing a state. An operator could manually re-trigger things by just moving a virtual file back into a queue. Supporting workflows in this way will be trivial due to the fact that every low-level command is already implemented today in order to provide the current EOS support for “conversions/converters”.

### **Work items on EOS would be:**

- add virtual tape FST placeholder
- extend LRU to Workflow engine, define attribute syntax
- implement workflow actions
- add some CLI functions for comfort
- replace LRU full table scan ( find / ) implementation with a scalable LRU queue in the future for more efficient garbage collection  
( we can use REDIS/ArDB for that, it has an LRU set implementation and scales to billions of entries).

The workflow engine is not only something useful for the tape. It could be useful e.g. to have faster uploads only to CERN from the PIT and trigger the move of one replica from CERN to WIGNER later or under certain conditions. We can also implement the CERNBOX backup with this workflow syntax. There are many applications for that already today outside CTA.

### **Other remarks**

The availability of a tape file can be classified as on-line, near-line and off-line. On-line means the file is on disk, near-line means the file is only on tape, off-line means the file is only on tape and that tape is currently not available. The current CASTOR SRM system permits a user running a stat request to see all three classifications. It is now assumed that end users no longer need to distinguish between near-line and off-line files. If a file is currently only on tape it shall be reported as near-line even if the tape involved is currently not available.