CERN
Tape Archive

Disk File Metadata for Tape Files

Restoring, Replicating, Migrating

Lasse Tjernæs Wardenær, Technical Student

CERN, IT Department, Storage Group, Tape Archive and Backups Section

Common Metadata

EOS Namespace
- EOS disk file location
- Location of archive file in CTA
- Archive file arrangement
- Validation

CTA
- Archive file location
- Location of disk files in EOS
- Archive file arrangement
- Validation

eos -j file info


```
{
  "atime": 1675949520,
  "atime_ns": 0,
  "btime": 0,
  "btime_ns": 0,
  "checksumtype": "adler",
  "checksumvalue": "3c777549",
  "ctime": 1675949520,
  "ctime_ns": 0,
  "detached": false,
  "etag": "\"1153293524205043712:3c777549\"",
  "fxid": "100152599",
  "gid": 0,
  "id": 4296353177,
  "inode": 1153293524205043700,
  "layout": "replica",
  "locations": [
    {
      "fsid": 65535,
      "fstpath": "/does_not_exist/00068e43/100152599",
      "geotag": "",
      "host": "localhost",
      "mountpoint": "/does_not_exist",
      "schedgroup": "tape.0",
      "status": "down"
    }
  ],
  "mode": 502,
  "mtime": 1675949520,
  "mtime_ns": 0,
  "name": "file3",
  "nlink": 1,
  "nstripes": 1,
  "path": "/eos/ctapublicpps/tabtest/injectionToolTest/file3",
  "pid": 4294967631,
  "size": 1811,
  "uid": 0,
  "xattr": {
    "eos.btime": "1675949520",
    "sys.archive.file_id": "4303580659",
    "sys.archive.storage_class": "tabtest"
  }
}
```


cta-admin --json tapefile


```
[
  {
    "af": {
      "archiveId": "4303580659",
      "storageClass": "tabtest",
      "creationTime": "1675939585",
      "checksum": [
        {
          "type": "ADLER32",
          "value": "3c777549"
        }
      ],
      "size": "1811"
    },
    "df": {
      "diskId": "4296353177",
      "diskInstance": "eosctapublicpps"
      "ownerId": {
        "uid": 98119,
        "gid": 4
      },
      "path": ""
    },
    "tf": {
      "vid": "L86769",
      "copyNb": 1,
      "blockId": "30181",
      "fSeq": "27"
    }
  }
]
```

When Do We Need to Change Disk File Metadata?

- Restoring a file from the recycle bin
- Moving a file from one EOSCTA instance to another
- Migrating to CTA
- Moving tape files from one archive route to another
- Going from single to dual copy

Important Details About The Next Tools

- Need root access in EOS namespace
  - gRPC access from CTA Frontend to EOS
  - Intended to be used by CTA admins
  - **Access rights should be set up before use and removed after use**
- Tools are not atomic
- Tools are idempotent

How to Add gRPC Access

1. Add gRPC key in EOS

```
$ eos vid set map -grpc key:<key> vuid:2 vgid:2
```

2. Add gRPC key to client side

Each line in /etc/cta/eos.grpc.keytab represents a key

<instance> <host>:<port> <key>

3. Add gRPC gateway in EOS

```
$ eos -r 0 0 vid add gateway <hostname> grpc
```

How to Remove gRPC Access

1. Remove gRPC key in EOS

```
$ eos vid rm vid rm grpc:<key>:uid
$ eos vid rm vid rm grpc:<key>:gid
```

2. Remove gRPC key to client side

Remove line in /etc/cta/eos.grpc.keytab

<instance> <host>:<port> <key>

3. Add gRPC gateway in EOS

```
$ vid remove gateway <hostname> grpc
```

Restoring a File From The Recycle Bin

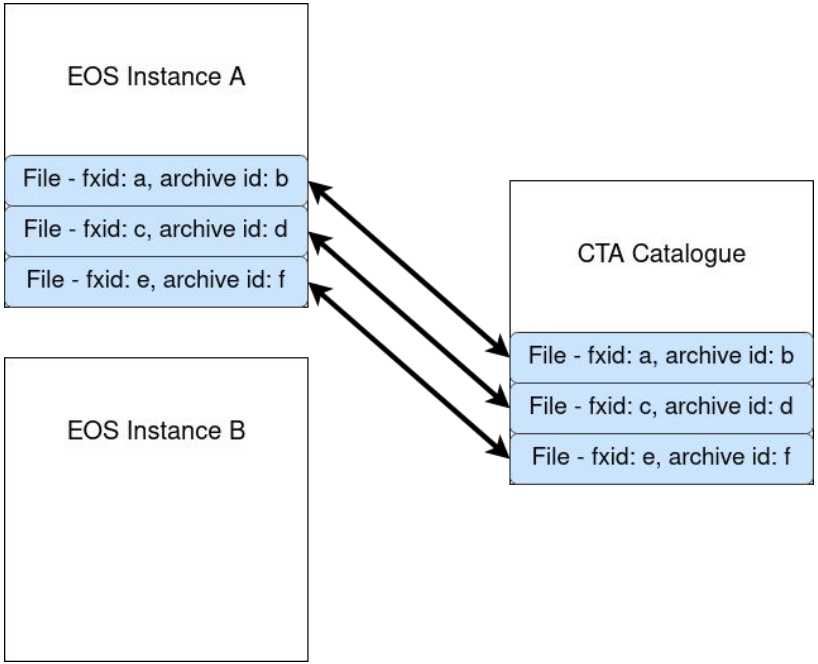
cta-restore-deleted-files documentation

Link to last year's presentation

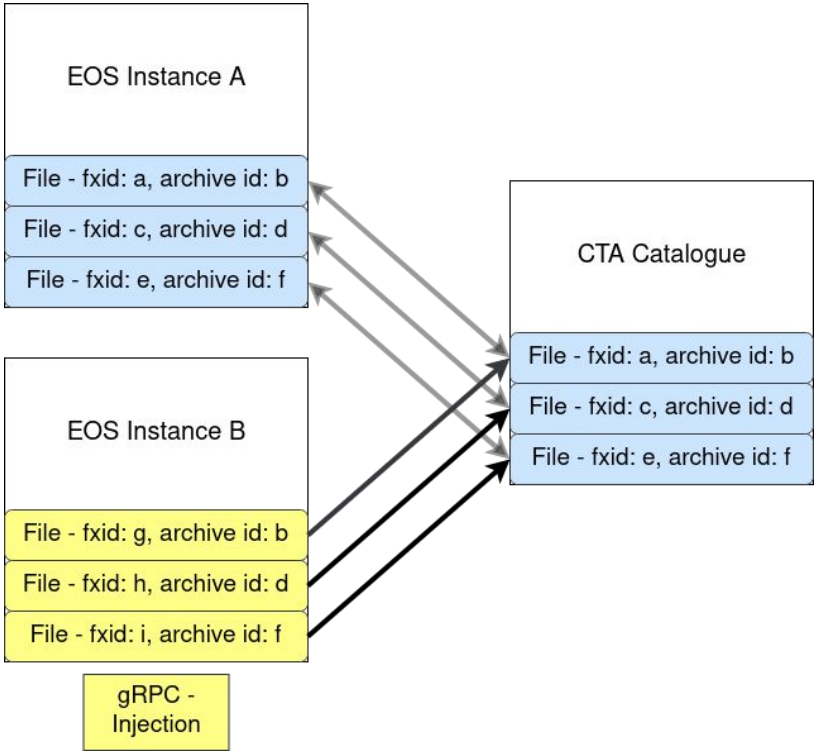
Moving Files Between EOSCTA Instances

cta-eos-namespace-inject documentation

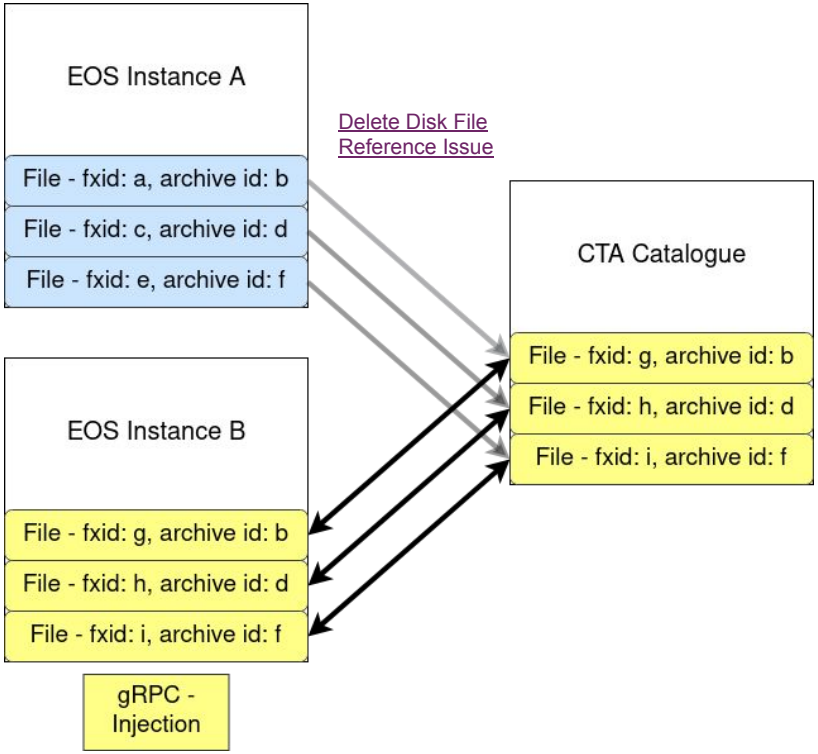
Moving Files Between EOSCTA Instances - 1



Moving Files Between EOSCTA Instances - 2



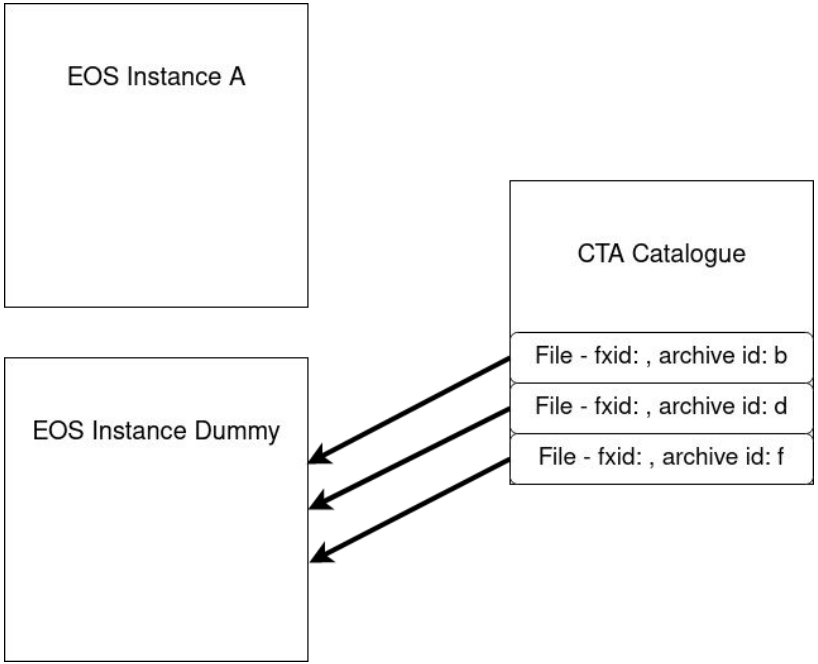
Moving Files Between EOSCTA Instances - 3



Migrating to CTA

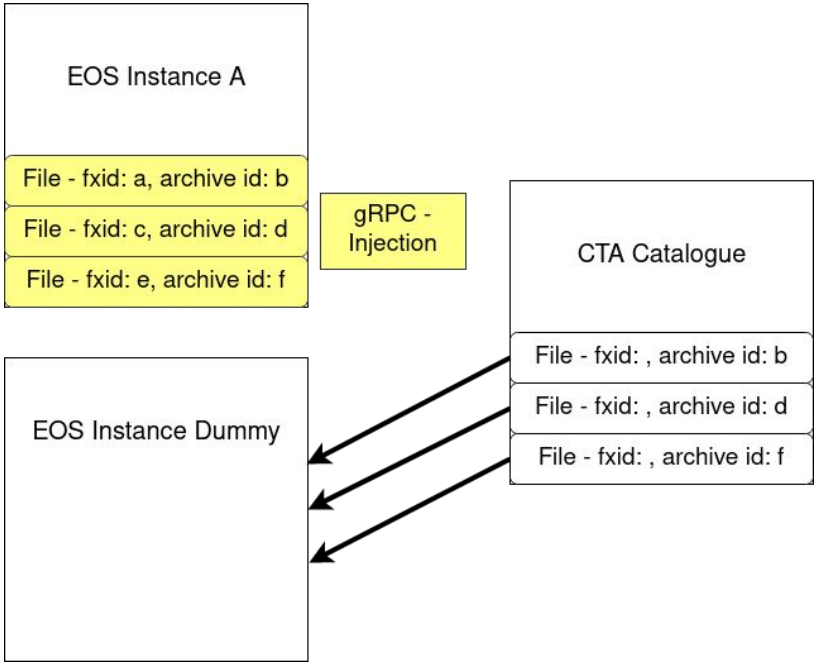
cta-eos-namespace-inject documentation

Injecting a File Into EOSCTA - 1

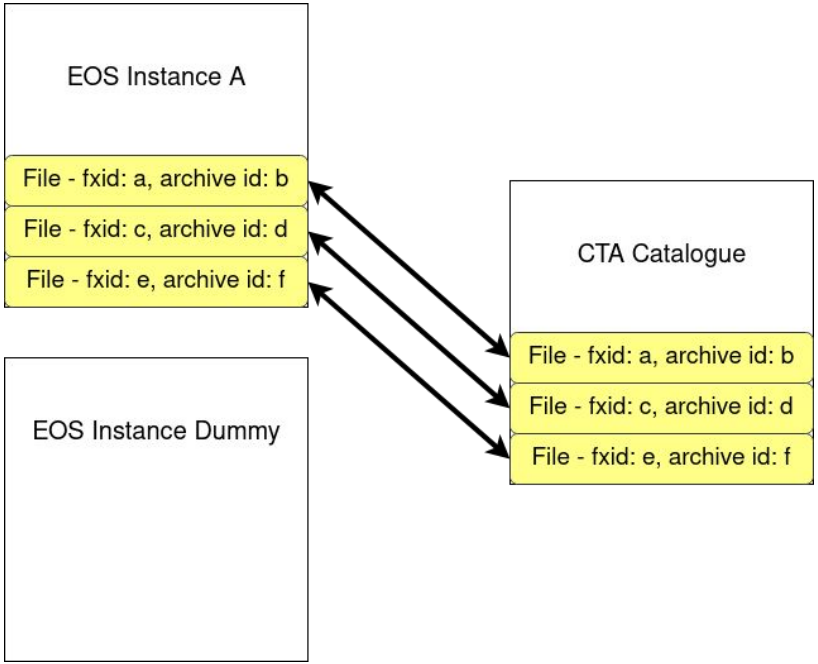


cta-admin diskinstance add –name <name> –comment <comment>

Injecting a File Into EOSCTA - 2



Injecting a File Into EOSCTA - 3



Setting Attributes in EOS

- **Before injecting a file into EOS**
  - Set directory attributes
  - Create folder structure
- **After injecting file into EOS**
  - Set missing file attributes
    - Not inherited automatically when injected

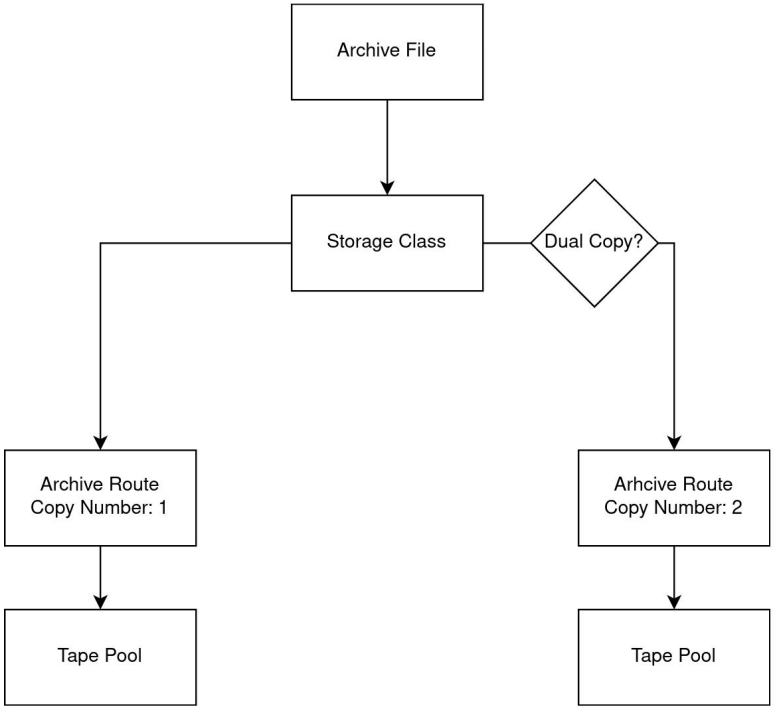
```
EOS Console [root://localhost] |/eos/ctapublicpps/tabtest/injectionToolTest/> attr ls .
sys.acl="u:98119:rwx+dp,z:!u!d,u:0:+u"
sys.archive.storage_class="tabtest"
sys.attr.link="/eos/ctapublicpps/proc/cta/workflowtabtest"
sys.eos.btime="1630056730.842891480"
sys.forced.checksum="adler"
sys.link.workflow.sync::abort_prepare.default="proto"
sys.link.workflow.sync::archive_failed.default="proto"
sys.link.workflow.sync::archived.default="proto"
sys.link.workflow.sync::closew.default="proto"
sys.link.workflow.sync::closew.retrieve_written="proto"
sys.link.workflow.sync::create.default="proto"
sys.link.workflow.sync::delete.default="proto"
sys.link.workflow.sync::evict_prepare.default="proto"
sys.link.workflow.sync::prepare.default="proto"
sys.link.workflow.sync::retrieve_failed.default="proto"
```

```
EOS Console [root://localhost] |/eos/ctapublicpps/tabtest/injectionToolTest/> attr ls file3
eos.btime="1675949520"
sys.archive.file_id="4303580659"
sys.archive.storage_class="tabtest"
```

Changing the Storage Class

cta-change-storage-class documentation

What Is a Storage Class and Why Change It?



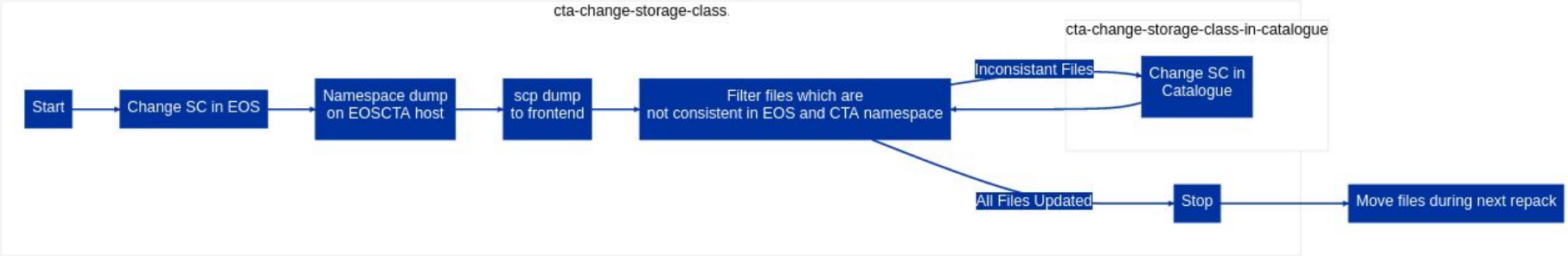
cta-admin --json storageclass ls


```
{
  "name": "sctest",
  "nbCopies": "1",
  "vo": "IT",
  "creationLog": {
    "username": "lwardena",
    "host": "cta-frontend01",
    "time": "1674032105"
  },
  "lastModificationLog": {
    "username": "lwardena",
    "host": "cta-frontend01",
    "time": "1674032105"
  },
  "comment": "change storage class test"
},
{
  "name": "sctest2",
  "nbCopies": "2",
  "vo": "IT",
  "creationLog": {
    "username": "lwardena",
    "host": "cta-frontend01",
    "time": "1675864194"
  },
  "lastModificationLog": {
    "username": "lwardena",
    "host": "cta-frontend01",
    "time": "1675930071"
  },
  "comment": "change storage class test"
}
```

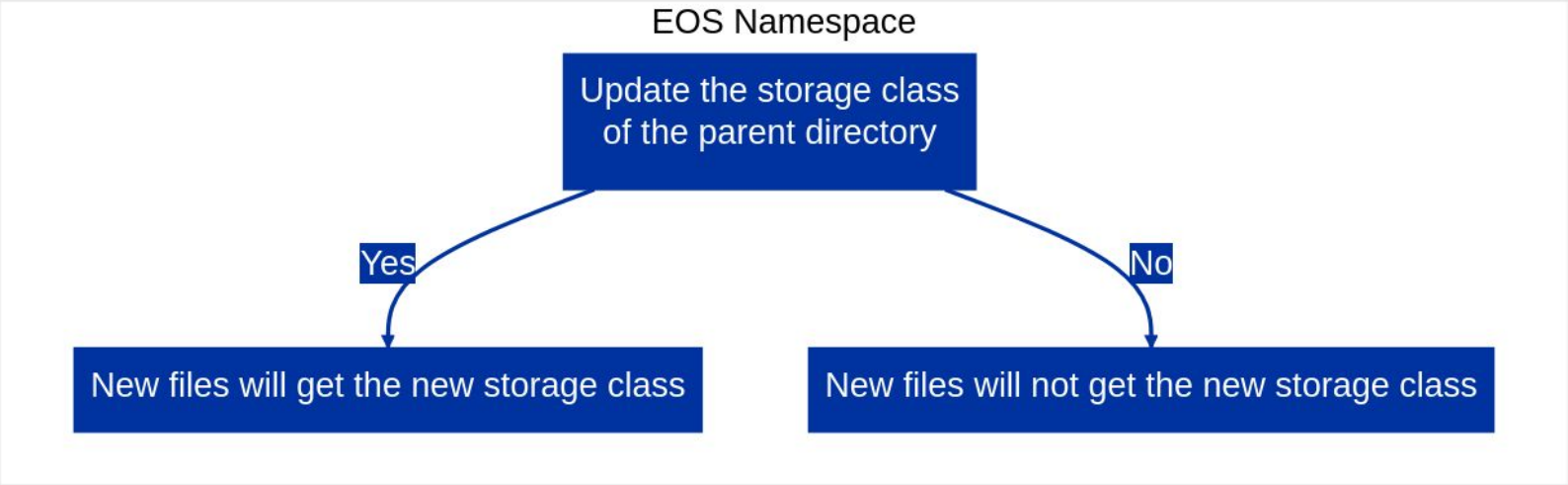

cta-admin --json archiveroute ls


```
{
  "storageClass": "sctest2",
  "copyNumber": 1,
  "tapepool": "test_lasse",
  "creationLog": {
    "username": "lwardena",
    "host": "cta-frontend01",
    "time": "1676020213"
  },
  "lastModificationLog": {
    "username": "lwardena",
    "host": "cta-frontend01",
    "time": "1676020213"
  },
  "comment": "Lasse: storage class test"
},
{
  "storageClass": "sctest2",
  "copyNumber": 2,
  "tapepool": "test_lasse2",
  "creationLog": {
    "username": "lwardena",
    "host": "cta-frontend01",
    "time": "1675932186"
  },
  "lastModificationLog": {
    "username": "lwardena",
    "host": "cta-frontend01",
    "time": "1675932186"
  },
  "comment": "storage class tool test"
}
```

Changing the Storage Class



What Happens If New Files Are Archived



Summary

- **Three tools with similar workflows**
  - cta-restore-deleted-files
  - cta-eos-namespace-inject
  - cta-change-storage-class
- **The updates are not atomic**
  - Idempotent
- **gRPC is required with write access in EOSCTA**
  - Remember to remove the access after finishing the updates

Documentation

EOSCTA Docs

Search

GitLab

EOSCTA Docs
- Introduction
- CTA Design
- Software Development
- Deployment Guide
- File Lifecycle
  - Introduction
  - Checksums
  - Archival
  - Retrieval
  - Deletion
  - Repack
  - Reconciliation
  - Zero-length files
  - Recycle-bin
  - Restoring
  - Garbage Collection
  - Change Storage Class
  - EOS Namespace Injection
- Database
- CTA Catalogue Schema
- CTA statistics
- Queues and Scheduling
- Authentication and Security
- RAO
- Tape Infrastructure
- EOS
- CASTOR

EOS Namespace Injection

Table of contents
- Introduction
- Procedure

Introduction

The `cta-eos-namespace-inject` tool is a general purpose tool for injecting files that are on tape with valid CTA catalogue metadata into EOS. Two examples of usage are:

1. Migration to CTA, where the catalogue has been populated with the correct metadata.
2. Moving a file from one EOS instance to another.

Procedure

The metadata migration to CTA involves two main parts:

1. Populate the CTA catalogue with all tape-related metadata. Set a dummy instance name when populating the CTA catalogue before using the tool. If you are moving a file from one EOS instance to another, setting a dummy is not necessary.
2. Run the `cta-eos-namespace-inject` tool for injection of files into EOS.

To use the tool call:

```
cta-eos-namespace-inject --json/-j <json-path>
```

where the JSON file is a text file with one JSON object for each line. The tool will check if the metadata for a file is consistent with what is found in the CTA catalogue. Example of JSON file:

```
{"eosPath": <path>, "diskInstance": <instance>, "archiveId": <id>,
"size": <size>, "checksumType": <checksum>, "checksumValue": <value>}
{"eosPath": <path>, "diskInstance": <instance>, "archiveId": <id>,
"size": <size>, "checksumType": <checksum>, "checksumValue": <value>}
{"eosPath": <path>, "diskInstance": <instance>, "archiveId": <id>,
"size": <size>, "checksumType": <checksum>, "checksumValue": <value>}
```

cta-eos-namespace-inject

cta-restore-deleted-files

EOSCTA Docs

Search

EOSCTA Docs
- Introduction
- CTA Design
- Software Development
- Deployment Guide
- File Lifecycle
  - Introduction
  - Checksums
  - Archival
  - Retrieval
  - Deletion
  - Repack
  - Reconciliation
  - Zero-length files
  - Recycle-bin
  - Restoring
  - Garbage Collection
  - Change Storage Class
  - EOS Namespace Injection
- Database
- CTA Catalogue Schema
- CTA statistics
- Queues and Scheduling
- Authentication and Security
- RAO
- Tape Infrastructure
- EOS
- CASTOR

Restoring files in the recycle bin

The tool `cta-restore-deleted-files` can restore deleted files from cta that match the given criteria. The tool can recover files that have been deleted by a user, by EOS or by an operator using cta-admin tf rm. This tool should only be used as a last resort.

There are tree scenarios for disk file recovery:

1. The file has been deleted in the EOS namespace and in the CTA catalogue This happens in the case of a normal file removal from the user (eos rm). The strategy is to reinject the file metadata in EOS and restore the file entry in the cta catalogue (with a new diskFileId).
2. The file has been deleted in the CTA catalogue but the EOS diskFileId remains the same. This can happen during a disk draining. The file is kept in the EOS namespace but the entry is removed from the Catalogue. The strategy is to just restore the CTA Catalogue entry.
3. The file has been deleted in the CTA catalogue but the EOS diskFileId changed. This happens during the conversion of a file from a space to another. The strategy is to recover the CTA Catalogue entry and update the diskFileId to the one that corresponds to its EOS entry.

Since the tool restore the files in both the EOS and CTA namespaces, there is no atomicity guarantee. Specifically, a file may be restored in the EOS namespace but the same restoration may fail in the CTA catalogue.

The tool comunicates with the EOS instances using grpc. As such, it requires an access token to these instances with write permissions. It uses the standard cta configuration files `/etc/cta/cta-cli.conf` and `/etc/cta-frontend-xrootd.conf`. The key tab path is read from the `cta.ns.config` entry in the frontend configuration file. (TODO: maybe we should add a command line option to choose this file).

Assuming the `daemon` user of an eos instance already has root priviledged, you can create an

EOSCTA Docs

Search

GitLab

EOSCTA Docs
- Introduction
- CTA Design
- Software Development
- Deployment Guide
- File Lifecycle
  - Introduction
  - Checksums
  - Archival
  - Retrieval
  - Deletion
  - Repack
  - Reconciliation
  - Zero-length files
  - Recycle-bin
  - Restoring
  - Garbage Collection
  - Change Storage Class
  - EOS Namespace Injection
- Database
- CTA Catalogue Schema
- CTA statistics
- Queues and Scheduling
- Authentication and Security
- RAO
- Tape Infrastructure
- EOS
- CASTOR

Change Storage Class

Table of contents
- Introduction
- Procedure

Introduction

The `cta-change-storage-class` tool changes the storage class for a file in both the EOS namespace and in the CTA Catalogue. Two examples of usage are:

1. Changing the storage class for specified files.
2. Going from single to dual copy.

Procedure

The workflow when using the tool is as follows:

1. Create a storage class or make sure that the new storage class exists.
2. If new files that are going to tape should inherit the new storage class from the parent directory, theextended attribute `sys.archive.storage_class` of the directory must be set manually.
3. Use `cta-change-storage-class` to change the storage class in the EOS namespace and in the CTA catalogue.
4. Perform a repack

To use the tool call:

```
cta-change-storage-class --id/-I <archiveFileID> | --json/-j <path>
--storageclassname/-n <storageClassName> [--frequency/-t <eosRequestFrequency>]
```

where the JSON file is a text file with one JSON object for each line. The tool will check if the archive ID exists in the CTA catalogue. Example of JSON file:

cta-change-storage-class

CERN

Tape Archive