

Gadi Scratch File Expiry Commands

Version: PUBLIC (20 Apr 2022)

Gadi's /scratch file system is subject to an expiry policy that will automatically remove files that haven't been accessed in more than 100 days. Using the `nci-file-expiry` tool you can get a listing of files that are within 14 days of their expiry time, and a listing of files that have recently expired but as still recoverable. You can also use this tool to request the recovery of such files – however please note that it is not an immediate process, and that all such requests are logged.

Listing files that will soon expire

```
$ nci-file-expiry list-warnings --help
usage: nci-file-expiry list-warnings [-h] [--json] [--filesystem FS [FS ...]]
                                     [--project P [P ...]]

optional arguments:
  -h, --help            show this help message and exit
  --json                generate JSON instead of default table output
  --filesystem FS [FS ...], -f FS [FS ...]
                       filter search to filesystem(s) FS
  --project P [P ...], -p P [P ...]
                       filter search to project(s) P
```

Running `nci-file-expiry list-warnings` with no additional arguments will generate a list of all files owned by you that are within 14 days of their expiry time. You can filter this output using the `--filesystem` and `--project` flags as needed. For example, `nci-file-expiry list-warnings --project ab12` will filter to just files located within the directories of project `ab12`.

Listing files that have recently expired ("been quarantined")

```
$ nci-file-expiry list-quarantined --help
usage: nci-file-expiry list-quarantined [-h] [--json]
                                         [--filesystem FS [FS ...]]
                                         [--project P [P ...]]

optional arguments:
  -h, --help            show this help message and exit
  --json                generate JSON instead of default table output
  --filesystem FS [FS ...], -f FS [FS ...]
                       filter search to filesystem(s) FS
  --project P [P ...], -p P [P ...]
                       filter search to project(s) P
```

As for the `list-warnings` case, you can filter the output of `list-quarantined` via `--filesystem` and `--project` flags.

Requesting recovery of a quarantined file

```
$ nci-file-expiry recover --help
usage: nci-file-expiry recover [-h] UUID PATH

positional arguments:
  UUID          UUID of quarantine record to recover
  PATH          location to store recovered object

optional arguments:
  -h, --help    show this help message and exit
```

If a file you require has been recently removed by this expiry policy, search for it in the output of `list-quarantined`. If there, it means that you can still recover it. To do so, you can use the `recover` option to `nci-file-expiry` to request this. When requesting recovery, you need to provide the ID of the corresponding record from the output of `list-quarantined`, and the path to where you would like to recover it to. This destination path must be on the same filesystem and be under the same project as the file when it was quarantined.

Once you've requested the recovery of a file, it will go into a queue to be processed at some point in the future. You can check the status of a request using the `status` command:

```
$ nci-file-expiry status --help
usage: nci-file-expiry status [-h] [--json]
                               [--id UUID | --between TIMESTAMP TIMESTAMP | --days N]

optional arguments:
  -h, --help            show this help message and exit
  --json                generate JSON instead of default table output
  --id UUID             UUID of recovery request
  --between TIMESTAMP TIMESTAMP
                        limit results to those active between timestamps
  --days N             limit results to those active within the last N days
                        (default: 14)
```