

API `datalad [--GLOBAL-OPTION <opt. flag spec.>] COMMAND [ARGUMENTS] [--OPTION <opt. flag spec.>]`

Each datalad invocation can have two sets of options: general options are given first, command-specific ones go after the subcommand.

GLOBAL OPTIONS

- `-c KEY=VALUE` Set config variables (overrides configurations in files)
- `-f/--output-format default|json|json_pp|tailored` Specify the format for command result rendering
- `-l/--log-level critical|error|warning|info|debug` Set logging verbosity level

COMMAND OPTIONS

- `-d/--dataset` Dataset location: path to root, or ^ for superdataset|
- `-D/--description` A location description (e.g., "my backup server")
- `-f/--force` Force execution of a command (Dangerzone!)
- `-m/--message` A description about a change made to the dataset
- `-r/--recursive` Perform an operation recursively across subdatasets
- `-R/--recursion-limit <n>` Limit recursion to n subdataset levels

Dataset operations

create `-d -D -f`

```
[ -c <config-proc>
[ PATH ]
```

Create a new dataset from scratch. If executed within a dataset and the `-d/--dataset` flag, it is created as a subdataset.

```
datalad create -c yoda my_first_ds
```

save `-d -m -R -r`

```
[ -u/--updated
[ PATH ... ]
```

Save the current state of a dataset. Use `-u/--updated` to leave untracked files untouched, and `--to-git` to save modifications to Git instead of Git-annex.

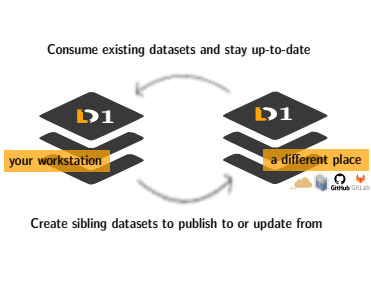
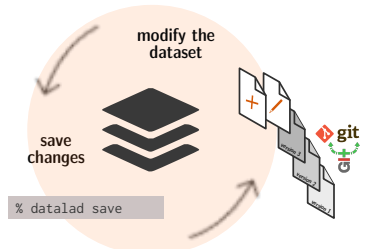
```
datalad save -m "did XY" file1
```

status `-d -R -r`

```
[ --annex <mode>
[ PATH ... ]
```

Report on the state of a dataset and/or its subdatasets. `--annex {None|basic|availability|all}` reports additional information on annex contents.

```
datalad status
```



get `-d -D -R -r`

```
[ -s/--source <label>
[ -n/--no-data ] PATH
```

Get dataset content (files/directories/subdatasets). Will get directory but not subdataset content recursively by default. Specify the label of a data source (e.g., sibling) with `-s/--source`.

```
datalad get file_xyz directory_1
```

clone `-d -D`
install `-d -D -R -r`

```
URL/PATH [DEST-PATH]
-s URL/PATH [DEST-PATH ...]
```

Install an existing dataset from path/url/open data collection (///). Providing `-d` installs a dataset as a subdataset. Install allows recursive operations.

```
datalad clone ///openneuro
datalad install -r -s ///openneuro
```

update `-d -R -r`

```
[ -s <siblingname>
[ --merge ]
```

Update a dataset from a sibling. Updates are by default on branch `remotes/origin/master`. Changes can be merged with `--merge`. Without `-s/--sibling`, all siblings are updated.

```
datalad update --merge -s origin
```

uninstall `-d -R -r`

```
[ --nocheck
PATH
```

Uninstall subdatasets. Availability of at least one remote copy needs to be verified - disable with `--nocheck`. PATH can not be the current directory.

```
datalad uninstall --nocheck subds/
```

remove `-d -m -R -r`

```
[ --nocheck
PATH
```

Remove datasets + contents, unregister from potential top-level datasets. Availability of at least one remote copy needs to be verified - disable with `--nocheck`. PATH can not be the current directory.

```
datalad remove --nocheck subds/
```

unlock `-d -R -r`

```
[ PATH ]
```

Unlock file(s) of a dataset to enable editing their content. If PATH is not provided, all files are unlocked. Requires datalad save to lock again afterwards.

```
datalad unlock my_data_file
```

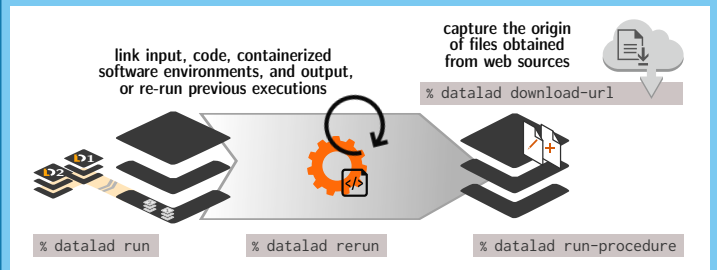
drop `-d -R -r`

```
[ --nocheck
PATH
```

Drop file content from dataset (remove data, retain symlink). Availability of at least one remote copy needs to be verified - disable with `--nocheck`. Drops all contents if no PATH is given.

```
datalad drop -r --nocheck dir_1/
```

Reproducible execution and provenance capture



siblings `-d -R -r -D`

```
query|add|remove|configure|enable
[ -s <siblingname> ] [ --url <url> ]
[ --publish-depends ]
```

Manage sibling configurations with either add, query (default), remove, configure, or enable. Provide a name with `-s`, a URL/path with `--url`, and publication dependencies with `--publish-depends`.

```
datalad siblings add \
-s different-place --url some/path
```

publish `-d -R -r -f`

```
[ --to <sibling>
[ --since <since>
[ --transfer-data auto|none|all ]
```

Publish a dataset to a known sibling and specify level of data-transfer with `--transfer-data`. `--since` allows to specify commit/tag from which to look for changes to publish.

```
datalad publish --transfer-data all
```

run `-m -d`

```
[ -i input ] [ -o output ]
[ --explicit ] <CMD>
```

Run arbitrary shell command and record its impact. Only creates record if dataset is modified. Gets any `-i/--input` and unlocks any `-o/--output`. Requires clean dataset or `--explicit`.

```
datalad run -m "rename" -i file \
-o file.txt "mv file file.txt"
```

rerun `-d -m`

```
[ --since COMMITISH ]
[ --onto COMMITISH ]
COMMITISH
```

Re-execute a previous run command identified by its hash, and save resulting modifications.

```
datalad rerun my-analysis-tag
```

run-procedure `-d`

```
[ --discover ]
<NAME> [ ARGS ... ]
```

Run prepared procedures (executables) on a dataset. To find available procedures, use `--discover` as the only argument, else specify the name of the procedure to run.

```
datalad run-procedure cfg_yoda
```

download-url `-d -m`

```
<URL> [ -O PATH ]
[ -o/--overwrite ]
```

Download, save, and record origin of content from websources. Specify a path to save under (`-O/--path`). `-o/--overwrite` enables overwriting existing files.

```
datalad download-url \
www.example.com/file -O file
```