DataLad Cheat Sheet

DataLad is a data management and publication multitool based on Git and git-annex with a command line interface and a Python API. With DataLad, you can version control arbitrarily large data, share or consume data, record your data's provenance, and work computationally reproducible.

DataLad The Handbook

API

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

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 renderung

`-l/--log-level critical|error|warning|info|debug`
Set logging verbosity level

`-C PATH`
Execute the command in the specified directory

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

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

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] [--to-git]
[--amend] [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
```

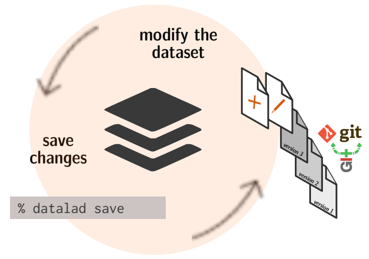
status

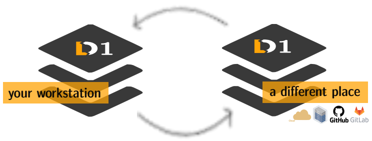
-d -R -r

```
[--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. For faster performance, tune `-e`,`-t`,`--untracked`.

```
datalad status
```




Consume existing datasets and stay up-to-date
D1
your workstation
D1
a different place
GitHub
GitLab
Create sibling datasets to publish to or update from

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

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

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

```
datalad clone ///openneuro
```

update

-d -R -r

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

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

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

create-sibling-*

-d -R -r

```
github|gitlab|gin|gitea|gogs
[ORG/]REPO [--private]
[-s NAME] [-api URL]
```

Create a dataset sibling on a repository hosting service under a given (organization)/repository name. Authentication is handled via access tokens (queried interactively at first try).

```
datalad create-sibling-gin -s gin mynewrepo
```

remove

-d -m -R -r

```
[--reckless <kill|modification|
availability|undead>] PATH
```

Remove datasets + contents, unregister from potential top-level datasets. Disable safety checks (e.g., for availability, unsaved content,...) with --reckless. PATH can not be the current directory.

```
datalad remove --reckless modification 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

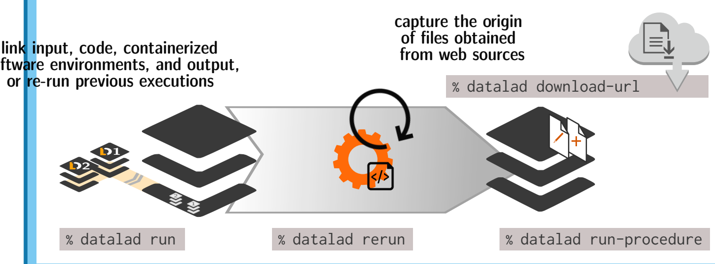
-d -R -r

```
[--reckless <mode>]
PATH
```

Drop content (remove data, retain symlink). Availability of at least 1 remote copy needs to be verified - disable with `--reckless availability`. Drops all contents if no PATH is given.

```
datalad drop -r --reckless availability dir/
```

Reproducible execution and provenance capture


link input, code, containerized software environments, and output, or re-run previous executions
capture the origin of files obtained from web sources
% datalad download-url
D1
D2
% datalad run
% datalad rerun
% datalad run-procedure

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
```

push

-d -R -r -f

```
[--to <sibling>] [--since <since>]
[--data <anything|nothing|auto|
auto-if-wanted>]
```

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

```
datalad push --to gin
```

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
```