Datalad Cheat Sheet

Basic command structure

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

General options

Command options

-d/--dataset: A path that points to the root of the dataset an operation is performed on. ^ points to the top-most 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

Global options

The datalad invocation has its own options. They need to be specified prior to the command specification.

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

Command line interface

Need help?
Type -h/--help after any command and get a help page!

Dataset operations

create -d -D -f

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

```
datalad create [-c <config-proc] \
        PATH
```

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

save -d -m -R -r

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

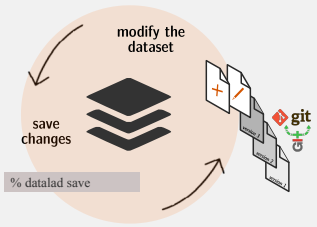
```
datalad save -m "did XY" file1
```

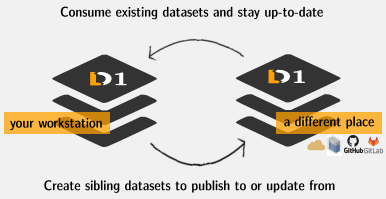
status -d -R -r

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

```
datalad status [--annex <mode>] \
        [PATH]
```

```
datalad status
```





get -d -D -R -r

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

```
datalad get [-s/--source <label>] \
        PATH
```

```
datalad get file_xyz directory_1
```

install -d -D -R -r

Install an existing dataset from path/url/open data collection (///). With -g/--get-data, all dataset content is obtained with a get operation. Providing -d installs a dataset as a subdataset.

```
datalad install -s/--source URL/PATH \
        [-g/--get-data] \
        [PATH]
```

```
datalad install -s \
https://github.com/datalad/datalad.git repos/datalad
```

update -d -R -r

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

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

siblings -d -R -r -D

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

```
datalad siblings <action> --url <url>\
        [-s <siblingname>] \
        [--publish-depends]
```

```
datalad siblings add -s new --url some/path
```

publish -d -R -r -f

Publish a dataset to a known sibling (--to) and specify level of data-transfer with --transfer-data {auto|none|all}. --since allows to specify commit/tag from which to look for changes to publish.

```
datalad publish [--to <sibling>] \
        [--since <since>] \
        [--transfer-data]
```

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

unlock -d -R -r

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 [PATH]
```

```
datalad unlock my_data_file
```

drop -d -R -r

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 [--nocheck] \
        PATH
```

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

uninstall -d -R -r

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] \
        PATH
```

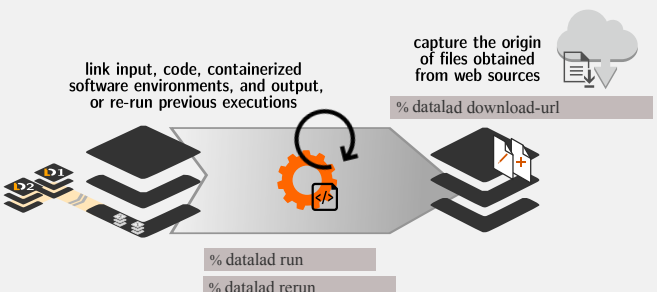
```
datalad uninstall --nocheck subds/
```

remove -d -m -R -r

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] \
        PATH
```

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



Reproducible execution and provenance capture

% datalad run-procedure

run -m -d

Run an arbitrary shell command & record its impact. Only creates a record if it modifies the dataset. -i/--input is retrieved with get and -o/--output is unlocked. Requires clean dataset status or --explicit.

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

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

rerun -d -m

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

```
datalad rerun [--since HASH] \
        [--onto HASH] HASH
```

```
datalad rerun 1f334e5
```

run-procedure -d

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

```
datalad run-procedure <NAME> \
        [--discover]
```

```
datalad run-procedure cfg_yoda
```

download-url -d -m

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 url [-O PATH]\
        o/--overwrite]
```

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

Concepts in brief

Dataset nesting

DataLad datasets can contain other DataLad datasets, enabling arbitrarily deep nesting inside of a dataset. Each individual dataset is a modular component with a stand-alone history. A superdataset only registers the version (via commit hash) of the subdataset. A dataset knows its installed subdatasets, but has no way of knowing about its superdataset(s). To apply commands not only to the dataset the action is performed in but also in subdatasets, run commands recursively, i.e. with -r/--recursive.

DataLad configuration

Within a dataset the following files contain configurations for DataLad, Git-annex, and Git: .git/config, .datalad/config, .gitmodules, .gitattributes. All but .git/config are version controlled and can be distributed with a dataset. The git config command can modify all but .gitattributes. .gitattributes contains rules about which files to annex based on file path, type and/or size. Environment variables for configurations override options set in configuration files.

DataLad extensions

DataLad extensions are additional Python packages that provide (domain-specific) functionality and new commands. The installation is done with standard Python package managers, such as pip, and beyond installation of the package, no additional setup is required. To install a DataLad extension, use `$ pip install <extension-name>`.

DataLad procedures

DataLad procedures are algorithms that alter datasets in certain ways. They are used to automate routine tasks such as configurations, synchronizing datasets with siblings, or populating datasets. `datalad run-procedure --discover` finds available procedures, `datalad run-procedure <name>` applies a given procedure to a dataset.

Python interface

All of DataLad's user-oriented commands are exposed via `datalad.api`. Any command can be imported as a stand-alone command like this:

```
>>> from datalad.api import <COMMAND>
```

Alternatively, to import all commands, one can use

```
>>> import datalad.api as dl
```

and subsequently access commands as `dl.get()`, `dl.install()`, ...

Want to know more?
Check out the DataLad Handbook at
handbook.datalad.org!

-f -m -r -d -D -R