

## Main commands

<b>dbt build</b>	Builds and tests all selected resources (models, seeds, snapshots, tests)
<b>dbt clean</b>	Deletes all folders specified in the clean-targets list specified in dbt_project.yml, like target.
<b>dbt compile</b>	Compiles (but does not run) the models in a project
<b>dbt clone</b>	Clones selected nodes from the specified state to the target schema(s)
<b>dbt debug</b>	Debugs dbt connections and projects
<b>dbt deps</b>	Downloads dependencies for a project
<b>dbt init</b>	Initializes a new dbt project (dbt Core only)
<b>dbt list</b>	Lists resources defined in a dbt project
<b>dbt parse</b>	Parses a project and writes detailed timing info
<b>dbt retry</b>	Retries the last run dbt command from the point of failure
<b>dbt run</b>	Runs the models in a project
<b>dbt run-operation</b>	Invokes a macro
<b>dbt seed</b>	Loads CSV files into the database
<b>dbt show</b>	Previews table rows post-transformation
<b>dbt snapshot</b>	Executes "snapshot" jobs defined in a project
<b>dbt source</b>	Provides tools for working with source data (including validating that sources are "fresh")
<b>dbt test</b>	Executes tests defined in a project

## Main Flags

<b>--debug</b>	Shows debug-level info in the terminal
<b>--fail-fast</b>	Makes dbt exit immediately if a resource fails
<b>--full-refresh</b>	Causes table and seeds to be recreated
<b>--help</b>	Shows available commands and arguments
<b>--log-level</b>	Limits the info in terminal to the set log level
<b>--profiles-dir</b>	Sets the path for profiles.yml
<b>--project-dir</b>	Sets the path for dbt_project.yml
<b>--record-timing-info</b>	Saves performance profiling information to a file
<b>--store-failures</b>	Makes dbt store failed rows in a table
<b>--threads</b>	Specifies the number of threads to use in the run
<b>--vars</b>	Supplies variables to the project
<b>--version</b>	Shows dbt version
<b>--warn-error</b>	Makes warnings act like errors

## dbt docs

<b>dbt docs generate</b>	Generates your projects' documentation website. Use the <b>--no-compile</b> argument to skip re-compilation
<b>dbt docs serve</b>	Starts a webserver to serve your documentation locally. Use the <b>--port</b> to choose the port

## Node selection

### Syntax overview

command	argument(s)
<b>run</b>	<b>--select</b> { <b>-s</b> }, <b>--exclude</b> , <b>--selector</b> , <b>--defer</b>
<b>test</b>	<b>--select</b> { <b>-s</b> }, <b>--exclude</b> , <b>--selector</b> , <b>--defer</b>
<b>seed</b>	<b>--select</b> { <b>-s</b> }, <b>--exclude</b> , <b>--selector</b>
<b>snapshot</b>	<b>--select</b> { <b>-s</b> }, <b>--exclude</b> , <b>--selector</b>
<b>list</b>	<b>--select</b> { <b>-s</b> }, <b>--exclude</b> , <b>--selector</b> , <b>--resource-type</b>
<b>compile</b>	<b>--select</b> { <b>-s</b> }, <b>--exclude</b> , <b>--selector</b>
<b>freshness</b>	<b>--select</b> { <b>-s</b> }, <b>--exclude</b> , <b>--selector</b>
<b>build</b>	<b>--select</b> { <b>-s</b> }, <b>--exclude</b> , <b>--selector</b> , <b>--resource-type</b> , <b>--defer</b>
<b>docs generate</b>	<b>--select</b> { <b>-s</b> }, <b>--exclude</b> , <b>--selector</b>

### Specifying resources

The **--select** flag accepts one or more args. Each arg can be one of compatibility.

- A packages name
- A model name
- A fully-qualified path to a directory of models
- A selection method (path, tag, config, test\_type, test\_name: etc

Examples:

- \$ dbt run --select my\_dbt\_project\_name - all models in your project
- \$ dbt run --select my\_dbt\_model - a specific model
- \$ dbt run --select path.to.my.models - all models in a specific directory
- \$ dbt run --select my\_package.some\_model - a specific model in a specific package
- \$ dbt run --select tag:nightly - models with the "nightly" tag
- \$ dbt run --select path/to/models - models contained in path/to/models
- \$ dbt run --select path/to/my\_model.sql - a specific model by its path

### Excluding models

dbt provides an **--exclude** flag with the same semantics as **--select**. Models specified with the **--exclude** flag will be removed from the set of models selected with **--select**.

Example:

- \$ dbt run --select my\_package.\*+ --exclude my\_package.a\_big\_model+

## Graph operators

Plus operator(+)

- \$ dbt run --select my\_model+ - select my\_model and all children
- \$ dbt run --select +my\_model - select my\_model and all parents
- \$ dbt run --select +my\_model+ - select my\_model, and all of its parents and children

N-plus operator

- \$ dbt run --select my\_model+1 - select my\_model and its first-degree children
- \$ dbt run --select 2+my\_model - select my\_model, its first-degree parents, and its second-degree parents ("grandparents")
- \$ dbt run --select 3+my\_model+4 - select my\_model, its parents up to the 3rd degree, and its children down to the 4th degree

At operator(@)

- \$ dbt run --models @my\_model - select my\_model, its children, and the parents of its children

Star operator(\*)

- \$ dbt run --select finance.base.\* - run all of the models in models/finance/base

## Methods examples

tag	dbt run -s "tag:nightly"
source	dbt run -s "source:snowplow+"
resource_type	dbt list -s "resource_type:test"
path	dbt run -s "models/staging/github"
package	dbt run -s "package:snowplow"
config	dbt run -s "config.materialized:incremental"
test_type	dbt test -s "test_type:generic"
test_name	dbt test -s "test_name:unique"
state	dbt run -s "state:modified" --state path/to/artifacts
exposure	dbt run -s "+exposure:weekly_kpis"
metric	dbt build -s "+metric:weekly_active_users"
result	dbt run -s "result:error" --state path/to/artifacts
source_status	dbt build -s "source_status:fresher"
group	dbt run -s "group:finance"
access	dbt list -s "access:public"
version	dbt list -s "version:latest"
unit_test	dbt list -s "unit_test:*"

## Set operators

Unions (space-delineated)

- \$ dbt run --select +snowplow\_sessions +fct\_orders \$ dbt run --select +my\_model - run snowplow\_sessions, all ancestors of snowplow\_sessions, fct\_orders, and all ancestors of fct\_orders

Intersections (comma-separated)

- \$ dbt run --select +snowplow\_sessions,fct\_orders - run all the common ancestors of snowplow\_sessions and fct\_orders

- \$ dbt run --select 3+my\_model+4 \$ dbt run --select marts.finance,tag:nightly - run models that are in the marts/finance subdirectory and tagged nightly

## State

Some methods require a manifest file to compare the current state of the project with another state, like the state of a previous invocation or the state of the project in production.

The path of this manifest can be passed using the **--state** flag.

dbt build -s "state:modified+" --defer --state path/to/artifacts

## Defer

Defer allows you build your project without having to build upstream resources. It requires a state. It is commonly used for Slim CI:

dbt build -s "state:modified+" --defer --state path/to/artifacts