

[Skip to content](#)


[  (../..../assets/logo-letter.svg) ](../..../ "uv")

uv

Locking and syncing

Initializing search

[ uv ](<https://github.com/astral-sh/uv> "Go to repository")

[  (../..../assets/logo-letter.svg) ](../..../ "uv") uv

[ uv ](<https://github.com/astral-sh/uv> "Go to repository")

\* [ Introduction ](../..../)

\* [ Getting started ](../..../getting-started/)

Getting started

\* [ Installation ](../..../getting-started/installation/)

\* [ First steps ](../..../getting-started/first-steps/)

\* [ Features ](../..../getting-started/features/)

\* [ Getting help ](../..../getting-started/help/)

\* [ Guides ](../..../guides/)

## Guides

- \* [ Installing Python ]([../../guides/install-python/](#))
- \* [ Running scripts ]([../../guides/scripts/](#))
- \* [ Using tools ]([../../guides/tools/](#))
- \* [ Working on projects ]([../../guides/projects/](#))
- \* [ Publishing packages ]([../../guides/package/](#))
- \* [ Integrations ]([../../guides/integration/](#))

## Integrations

- \* [ Docker ]([../../guides/integration/docker/](#))
- \* [ Jupyter ]([../../guides/integration/jupyter/](#))
- \* [ GitHub Actions ]([../../guides/integration/github/](#))
- \* [ GitLab CI/CD ]([../../guides/integration/gitlab/](#))
- \* [ Pre-commit ]([../../guides/integration/pre-commit/](#))
- \* [ PyTorch ]([../../guides/integration/pytorch/](#))
- \* [ FastAPI ]([../../guides/integration/fastapi/](#))
- \* [ Alternative indexes ]([../../guides/integration/alternative-indexes/](#))
- \* [ Dependency bots ]([../../guides/integration/dependency-bots/](#))
- \* [ AWS Lambda ]([../../guides/integration/aws-lambda/](#))
- \* [ Concepts ]([../](#))

## Concepts

- \* [ Projects ]([../](#))

## Projects

- \* [ Structure and files ]([../layout/](#))
- \* [ Creating projects ]([../init/](#))
- \* [ Managing dependencies ]([../dependencies/](#))
- \* [ Running commands ]([../run/](#))
- \* Locking and syncing [ Locking and syncing ]([./](#)) Table of contents
  - \* Creating the lockfile
  - \* Exporting the lockfile
  - \* Checking if the lockfile is up-to-date
  - \* Upgrading locked package versions
- \* [ Configuring projects ]([../config/](#))
- \* [ Building distributions ]([../build/](#))
- \* [ Using workspaces ]([../workspaces/](#))
- \* [ Tools ]([../../tools/](#))
- \* [ Python versions ]([../../python-versions/](#))
- \* [ Resolution ]([../../resolution/](#))
- \* [ Caching ]([../../cache/](#))
- \* [ Configuration ]([../../configuration/](#))

## Configuration

- \* [ Configuration files ]([../../configuration/files/](#))
- \* [ Environment variables ]([../../configuration/environment/](#))
- \* [ Authentication ]([../../configuration/authentication/](#))
- \* [ Package indexes ]([../../configuration/indexes/](#))
- \* [ Installer ]([../../configuration/installer/](#))

\* [ The pip interface ]([../..../pip/](#))

## The pip interface

- \* [ Using environments ]([../..../pip/environments/](#))
- \* [ Managing packages ]([../..../pip/packages/](#))
- \* [ Inspecting packages ]([../..../pip/inspection/](#))
- \* [ Declaring dependencies ]([../..../pip/dependencies/](#))
- \* [ Locking environments ]([../..../pip/compile/](#))
- \* [ Compatibility with pip ]([../..../pip/compatibility/](#))
- \* [ Reference ]([../..../reference/](#))

## Reference

- \* [ Commands ]([../..../reference/cli/](#))
- \* [ Settings ]([../..../reference/settings/](#))
- \* [ Troubleshooting ]([../..../reference/troubleshooting/](#))

## Troubleshooting

- \* [ Build failures ]([../..../reference/troubleshooting/build-failures/](#))
- \* [ Reproducible examples ]([../..../reference/troubleshooting/reproducible-examples/](#))
- \* [ Resolver ]([../..../reference/resolver-internals/](#))
- \* [ Benchmarks ]([../..../reference/benchmarks/](#))
- \* [ Policies ]([../..../reference/policies/](#))

## Policies

- \* [ Versioning ]([../..../reference/policies/versioning/](#))
- \* [ Platform support ]([../..../reference/policies/platforms/](#))
- \* [ License ]([../..../reference/policies/license/](#))

## Table of contents

- \* Creating the lockfile
- \* Exporting the lockfile
- \* Checking if the lockfile is up-to-date
- \* Upgrading locked package versions

1. [ Introduction ]([../..../](#))
2. [ Concepts ]([../..../](#))
3. [ Projects ]([../](#))

## # Locking and syncing

### ### Creating the lockfile

The lockfile is created and updated during uv invocations that use the project environment, i.e., ``uv sync`` and ``uv run``. The lockfile may also be explicitly created or updated using ``uv lock``:

```
$ uv lock
```

### ### Exporting the lockfile

If you need to integrate uv with other tools or workflows, you can export `uv.lock`` to `requirements.txt`` format with `uv export --format requirements-txt``. The generated `requirements.txt`` file can then be installed via `uv pip install``, or with other tools like `pip``.

In general, we recommend against using both a `uv.lock`` and a `requirements.txt`` file. If you find yourself exporting a `uv.lock`` file, consider opening an issue to discuss your use case.

### ### Checking if the lockfile is up-to-date

To avoid updating the lockfile during `uv sync`` and `uv run`` invocations, use the `--frozen`` flag.

To avoid updating the environment during `uv run`` invocations, use the `--no-sync`` flag.

To assert the lockfile matches the project metadata, use the `--locked`` flag.

If the lockfile is not up-to-date, an error will be raised instead of updating the lockfile.

You can also check if the lockfile is up-to-date by passing the `--check`` flag to `uv lock``:

```
$ uv lock --check
```

This is equivalent to the `--locked` flag for other commands.

### ### Upgrading locked package versions

By default, uv will prefer the locked versions of packages when running `uv sync` and `uv lock` with an existing `uv.lock` file. Package versions will only change if the project's dependency constraints exclude the previous, locked version.

To upgrade all packages:

```
$ uv lock --upgrade
```

To upgrade a single package to the latest version, while retaining the locked versions of all other packages:

```
$ uv lock --upgrade-package <package>
```

To upgrade a single package to a specific version:

```
$ uv lock --upgrade-package <package>==<version>
```

In all cases, upgrades are limited to the project's dependency constraints. For example, if the project defines an upper bound for a package then an upgrade will not go beyond that version.

#### Note

uv applies similar logic to Git dependencies. For example, if a Git dependency references the ``main`` branch, uv will prefer the locked commit SHA in an existing ``uv.lock`` file over the latest commit on the ``main`` branch, unless the ``--upgrade`` or ``--upgrade-package`` flags are used.

January 31, 2025

[Back to top](#) [ [Previous Running commands](#) ]([../run/](#)) [ [Next Configuring projects](#) ]([../config/](#))

Made with [ [Material for MkDocs Insiders](https://squidfunk.github.io/mkdocs-) ](<https://squidfunk.github.io/mkdocs->



material/)

[ ](https://github.com/astral-sh/uv "github.com") [  
](https://discord.com/invite/astral-sh "discord.com") [  
](https://pypi.org/project/uv/ "pypi.org") [ ](https://x.com/astral\_sh  
"x.com")