
conda Documentation

Release

Anaconda, Inc.

Mar 11, 2020

Contents

1	Conda	3
2	Conda-build	5
3	Miniconda	7
4	Help and support	11
5	Contributing	13
6	Conda license	17

Package, dependency and environment management for any language—Python, R, Ruby, Lua, Scala, Java, JavaScript, C/C++, FORTRAN, and more.

Conda is an open source package management system and environment management system that runs on Windows, macOS and Linux. Conda quickly installs, runs and updates packages and their dependencies. Conda easily creates, saves, loads and switches between environments on your local computer. It was created for Python programs, but it can package and distribute software for any language.

Conda as a package manager helps you find and install packages. If you need a package that requires a different version of Python, you do not need to switch to a different environment manager, because conda is also an environment manager. With just a few commands, you can set up a totally separate environment to run that different version of Python, while continuing to run your usual version of Python in your normal environment.

In its default configuration, conda can install and manage the thousand packages at repo.anaconda.com that are built, reviewed and maintained by Anaconda®.

Conda can be combined with continuous integration systems such as Travis CI and AppVeyor to provide frequent, automated testing of your code.

The conda package and environment manager is included in all versions of Anaconda and Miniconda.

Conda is also included in [Anaconda Enterprise](#), which provides on-site enterprise package and environment management for Python, R, Node.js, Java and other application stacks. Conda is also available on [conda-forge](#), a community channel. You may also get conda on [PyPI](#), but that approach may not be as up to date.

CHAPTER 1

Conda

CHAPTER 2

Conda-build

Miniconda

Miniconda is a free minimal installer for conda. It is a small, bootstrap version of Anaconda that includes only conda, Python, the packages they depend on, and a small number of other useful packages, including pip, zlib and a few others. Use the `conda install` command to install 720+ additional conda packages from the Anaconda repository.

See if [Miniconda](#) is right for you.

3.1 Windows installers

Table 1: Windows

Python version	Name	Size	SHA256 hash
Python 3.7	Miniconda3-Windows-64-bit	351.6 MiB	1701955cd637d1dad5a84958fd470649b79de973d1570541eb52857664b5056c
	Miniconda3-Windows-32-bit	352.2 MiB	ca74cb6eb0731db2b972c0fb512e29661a84c3f01ac6133121b4372eb1c41f46
Python 2.7	Miniconda2-Windows-64-bit	250.9 MiB	8647c54058f11842c37854edeff4d20bc1fbdad8b88d9d34d76fda1630e64846
	Miniconda2-Windows-32-bit	248.7 MiB	0d106228d6a4610b599df965dd6d9bb659329a17e3d693e3274b20291a7c6f94

3.2 MacOSX installers

Table 2: MacOSX

Python version	Name	Size	SHA256 hash
Python 3.7	Miniconda3-MacOSX64-bit bash	350.3 MiB	d1fca4f74f9971c27220122723843f6c879a5d13ff59c01fca17ef62a1576732
	Miniconda3-MacOSX64-bit pkg	361.3 MiB	f3ede3a58d82fb5dcbca52d291a9edb5cd962d84d823a20693dd4bb27506cdd0
Python 2.7	Miniconda2-MacOSX64-bit bash	239.4 MiB	0db8f4037e40e13eb1d2adc89e054dfb165470cc77be45ef2bf9cb31c8b72f39
	Miniconda2-MacOSX64-bit pkg	247.8 MiB	fcc30b2e18f7a292b34b2e24ad855786a66423f860157fa2b77e48b6392f0abb

3.3 Linux installers

Table 3: Linux

Python version	Name	Size	SHA256 hash
Python 3.7	Miniconda3-Linux64-bit	81.1 MiB	957d2f0f0701c3d1335e3b39f235d197837ad69a944fa6f5d8ad2c686b69df3b
	Miniconda3-Linux32-bit	362.7 MiB	f387eded3fa4ddc3104b7775e62d59065b30205c2758a8b86b4c27144adafcc4
Python 2.7	Miniconda2-Linux64-bit	246.0 MiB	383fe7b6c2574e425eee3c65533a5101e68a2d525e66356844a80aa02a556695
	Miniconda2-Linux32-bit	239.0 MiB	2e20ac4379ca5262e7612f84ad26b1a2f2782d0994facdec28e0baf51749979

3.4 Installing

- See hashes for all Miniconda installers.
- [Verify your installation.](#)
- [Installation instructions.](#)

3.5 Other resources

- [Miniconda with Python 3.7 for Power8 & Power9](#)
- [Miniconda with Python 2.7 for Power8 & Power9](#)
- [Miniconda Docker images](#)
- [Miniconda AWS images](#)
- [Archive and MD5 sums for the installers](#)
- [conda change log](#)

These Miniconda installers contain the conda package manager and Python. Once Miniconda is installed, you can use the conda command to install any other packages and create environments, etc. For example:

```
$ conda install numpy
...
$ conda create -n py3k anaconda python=3
...
```

There are two variants of the installer: Miniconda is Python 2 based and Miniconda3 is Python 3 based. Note that the choice of which Miniconda is installed only affects the root environment. Regardless of which version of Miniconda you install, you can still install both Python 2.x and Python 3.x environments.

The other difference is that the Python 3 version of Miniconda will default to Python 3 when creating new environments and building packages. So for instance, the behavior of:

```
$ conda create -n myenv python
```

will be to install Python 2.7 with the Python 2 Miniconda and to install Python 3.7 with the Python 3 Miniconda. You can override the default by explicitly setting `python=2` or `python=3`. It also determines the default value of `CONDA_PY` when using `conda build`.

Note: If you already have Miniconda or Anaconda installed, and you just want to upgrade, you should not use the installer. Just use `conda update`.

For instance:

```
$ conda update conda
```

will update conda.

- Get free community support with our [Google group](#).
- Paid [support](#), [training](#) and [consulting](#) options are available. Our teaching philosophy is that the best way to learn is with hands-on experience addressing real-world problems. Courses are available to individuals online, at numerous sites, or in-house at your place of business. We also offer consulting services for the analysis, management and visualization of scientific and business data or optimizing your processing workflows on modern hardware and GPUs.
- Support is included in purchases of [Anaconda Enterprise](#).

4.1 Join the conda email group

Join the mailing lists for both [Anaconda](#) and [conda](#). Ask questions, answer questions, discuss ways to use conda, request new features and submit any other comments you may have.

4.2 Contribute recipes to GitHub

Submit new recipes to the GitHub repository at <https://github.com/conda/conda-recipes>. When you build a new conda recipe for a program not yet listed in the repository, use a GitHub pull request to submit it to the repository, so that others using that program in the future can reuse it. Recipes are welcome for programs that use any license, such as GPL, BSD, MIT or Apache, and all of the recipes in this repository are released into the public domain.

4.3 Report bugs

If you find a bug, search to see if it has already been reported, and then report it if no one else has.

Issues with the conda documentation are tracked on GitHub at <https://github.com/conda/conda-docs/issues>.

For issues with conda, conda-build, [repo.anaconda.com](https://github.com/conda/conda), anaconda.org, and specific conda packages, please see *New Issues*.

4.4 Presentations and blog posts

- [Packaging and Deployment with conda—Travis Oliphant](#).
- [Anaconda Developer Blog](#).

4.5 Conda announcements list

Conda Announcements is a low-traffic email list for news and updates directly from the conda core team. By providing your email address below, you consent to join Conda Announcements. It is not a marketing list. We never sell, give away or distribute your email address to third parties.

- No more than 1 email per week, usually less.
- Project announcements relevant to all conda users.
- No corporate marketing hype.
- No spam.

For convenience, we use MailChimp to manage subscriptions, including [archived announcements](#).

Upon clicking the “Consent and Subscribe” button, you will receive an email that contains a link to confirm your subscription. You must click this link to activate your subscription. You can opt-out at any time by clicking the “Unsubscribe” link included in any email we send through this announcement list, or by contacting privacy@anaconda.com.

5.1 New Issues

If your issue is a bug report or feature request for:

- **a specific conda package:** please file it at <https://github.com/ContinuumIO/anaconda-issues/issues>
- **anaconda.org:** please file it at <https://github.com/Anaconda-Platform/support/issues>
- **repo.anaconda.com:** please file it at <https://github.com/ContinuumIO/anaconda-issues/issues>
- **commands under** `conda build`: please file it at <https://github.com/conda/conda-build/issues>
- **commands under** `conda env`: please file it at <https://github.com/conda/conda/issues>
- **all other conda commands:** please file it at <https://github.com/conda/conda/issues>

5.2 Development Environment, Bash

To set up an environment to start developing on conda code, we recommend the following steps:

1. Fork the conda/conda repository, clone it locally anywhere you choose (an isolation miniconda will be set up within the clone directory), and set up `git remote` to point to upstream and fork. For detailed directions, see below.

- 1a. Choose where you want the repository located (not location of existing conda)

```
CONDA_PROJECT_ROOT="$HOME/conda"
```

- 1b. Clone the project, with `upstream` being the main repository. Make sure to click the `Fork` button above so you have your own copy of this repo.

```
GITHUB_USERNAME=kalefranz  
git clone git@github.com:$GITHUB_USERNAME/conda "$CONDA_PROJECT_ROOT"
```

(continues on next page)

(continued from previous page)

```
cd "$CONDA_PROJECT_ROOT"
git remote --add upstream git@github.com:conda/conda
```

2. Create a local development environment, and activate that environment

```
. dev/start
```

This command will create a project-specific base environment at `./devenv`. If the environment already exists, this command will just quickly activate the already-created `./devenv` environment.

To be sure that the conda code being interpreted is the code in the project directory, look at the value of `conda location`: in the output of `conda info --all`.

3. Run conda's unit tests using GNU make

```
make unit
```

or alternately with pytest

```
py.test -m "not integration and not installed" conda tests
```

or you can use pytest to focus on one specific test

```
py.test tests/test_create.py -k create_install_update_remove_
↪smoketest
```

5.3 Development Environment, Windows `cmd.exe` shell

In these steps, we assume `git` is installed and available on `PATH`.

1. Choose where you want the project located

```
set "CONDA_PROJECT_ROOT=%HOMEPATH%\conda"
```

2. Clone the project, with `origin` being the main repository. Make sure to click the `Fork` button above so you have your own copy of this repo.

```
set GITHUB_USERNAME=kalefranz
git clone git@github.com:conda/conda "%CONDA_PROJECT_ROOT%"
cd "%CONDA_PROJECT_ROOT%"
git remote --add %GITHUB_USERNAME% git@github.com:%GITHUB_USERNAME%/
↪conda
```

To be sure that the conda code being interpreted is the code in the project directory, look at the value of `conda location`: in the output of `conda info --all`.

3. Create a local development environment, and activate that environment

```
.\dev\start
```

This command will create a project-specific base environment at `.\devenv`. If the environment already exists, this command will just quickly activate the already-created `.\devenv` environment.

5.4 Conda Contributor License Agreement

In case you're new to CLAs, this is rather standard procedure for larger projects. [Django](#) and even [Python](#) itself both use something similar.

Conda is released under the 3-clause BSD license with the following terms:

(c) 2017 Continuum Analytics, Inc. (dba Anaconda, Inc.). <https://www.anaconda.com>. All Rights Reserved

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name of Continuum Analytics, Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS “AS IS” AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL CONTINUUM ANALYTICS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

6.1 Dependencies

versioneer.py is Public Domain.

The ProgressBar package is redistributed under the BSD 3-clause license as per the terms of its license:

6.2 ProgressBar License

You can redistribute and/or modify this library under the terms of the GNU LGPL license or BSD license (or both).

6.2.1 LGPL

progressbar - Text progress bar library for python. Copyright (C) 2005 Nilton Volpato

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

6.2.2 BSD

progressbar - Text progress bar library for python Copyright (c) 2008 Nilton Volpato

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name of the author nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.