

## 📦 Installation

This section describes how to install the PDS Deep Archive. Because the instructions vary markedly based on platform, it's divided into two sections:

- Unix and Unix-like systems, including Linux and macOS
- Windows systems

### Unix and Unix-Like Systems

Follow the instructions in this section to install the Deep Archive software onto Unix and Unix-like systems. This includes operating systems such as Linux, FreeBSD, OpenBSD, NetBSD, etc., as well as Apple Macintosh systems running macOS.

For users of Windows systems, see the next section, below.

### Requirements

Prior to installing this software, ensure your system meets the following requirements:

- **Python 3**. This software requires Python 3.9, 3.10, or 3.11. Python 2 will absolutely *not* work, and indeed Python 2 came to its end of life on the first of January, 2020. Run `python --version`, or `python3 --version`, to check what is installed.
- `libxml2` version 2.9.2; later 2.9 versions are fine. Run `xml2-config --version` to find out.
- `libxslt` version 1.1.28; later 1.1 versions are OK too. Run `xslt-config` to see.

Consult your operating system instructions or system administrator to install the required packages. For those without system administrator access and are feeling anxious, you could try a local (home directory) **Python 3** installation using [Miniconda](#).

### Doing the Installation on Unix

## ! Note

Some things to be aware of regarding examples below:

- The octothorp characters `#` below indicate comments and need not be typed in.
- The location of where you choose to create a Python virtual environment is entirely your preference.
- The examples below should be seen only as suggestions. Invoking command lines below are demonstrative.
- Please consult your system documentation for the appropriate invocations for your operating system, command shell (or “terminal”), and so forth.

The easiest way to install this software is to use [Pip](#), the Python Package Installer. If you have Python on your system, you probably already have Pip; you can run `pip --help` or `pip3 --help` to check.

It's best install the PDS Deep Archive into a [virtual environment](#) so it won't interfere with—or be interfered by—other packages. To do so:

```
# Example assumes bash command shell. For others, consult shell documentation.
mkdir -p $HOME/.virtualenvs
python3 -m venv $HOME/.virtualenvs/pds-deep-archive
source $HOME/.virtualenvs/pds-deep-archive/bin/activate
pip3 install pds.deeparchive
```

You can then run `pds-deep-archive --help` or `pds-deep-registry-archive --help` to get a usage message and ensure it's properly installed.

## ! Note

The above commands will install last approved release from the Python Package Index (“[Cheeseshop](#)”). The latest, cutting edge release is posted at the Test Package Index, but these releases may not be fully confirmed to be operational. If you like taking risks, run the following to create a new virtual environment and install the latest development version of the software:

```
mkdir -p $HOME/.virtualenvs
python3 -m venv $HOME/.virtualenvs/pds-deep-archive
source $HOME/.virtualenvs/pds-deep-archive/bin/activate
pip3 install --index-url https://test.pypi.org/simple --extra-index-url
https://pypi.org/simple`` pds.deeparchive
```

## Windows Installation

To install the Deep Archive software on Windows comprises the following steps:

1. Installing Python 3.11 for Windows
2. Creating a “virtual environment” to contain an isolated instance of Python 3.11
3. Installing LXML 4.9.0 for Python 3.11 into the virtual environment
4. Installing the PDS Deep Archive into the virtual environment

The remainder of this section details these steps.

### Installing Python for Windows

Python 3.11 (and specifically Python 3.11—no later or earlier versions will work) will need to be installed onto your Windows system. There are several ways to get Python 3.11:

- The “Microsoft Store” app
- Directly from <https://python.org/>
- Using [Anaconda](#)
- Using [Miniconda](#)

Use whatever is the most familiar to you. If you’re not sure, the Microsoft Store app is probably the easiest. To use the Microsoft Store to install Python 3.11, do the following:

1. In the Windows taskbar’s search box or Start Menu, open Microsoft Store.
2. In the search box at the top, type `Python 3.11`
3. In the list of matching results, press the “Get” button next to Python 3.11.



Tip

If you're on a managed system, you may need to ask your system administrator to install Python 3.11 for you.

Next, confirm that it's properly installed by opening Windows PowerShell and starting Python 3.11 from the command-line. Use the Windows taskbar search box or Start Menu to launch Windows PowerShell, then type `python3.11` and press Enter.

### ❗ Note

If you installed Python from <https://python.org/> or using Anaconda or Miniconda, the command you enter may be `python3` or even simply `python` instead of `python3.11`.

### ❗ Tip

If entering the `python3` or `python` commands opens the Microsoft Store instead, you may need to turn off “application execution aliases”. To do so, open the Settings app, choose Apps → Advanced App Settings → App Execution Aliases. In this list, look for “App Installer `python.exe`” and “App Installer `python3.exe`” and slide both switches to “off”.

Once you see Python's `>>>` prompt, press CTRL+Z then press Enter to exit Python.

## Creating the “Virtual Environment”

Python supports the notion of “virtual environments”, which are small installations of Python that are isolated from the system's installation. This enables you to install software for different Python applications without interfering dependencies or conflicts. We recommend creating a virtual environment for the Deep Archive software.

To do so, open Windows PowerShell (as above) and at the prompt, type the following command (then press Enter):

```
python3.11 -m venv pds
```

### ❗ Note

If you installed Python from <https://python.org/> or using Anaconda or Miniconda, you may need to replace `python3.11` with `python3` or even simply `python`.

This will create a subfolder in the current directory called `pds` which contains the virtual environment. Next, you'll need to "activate" the virtual environment by entering the following command (then press Enter):

```
.\pds\Scripts\activate.ps1
```

Your PowerShell prompt will change to show `(pds)` at the front, indicating that the virtual environment is now active.

## Installing LXML 4.9.0

Because the Deep Archive manipulates and parses XML files, the "LXML" API for Python must now be installed into the virtual environment. In the same Windows PowerShell with the `(pds)` prompt, enter the following command (then press Enter):

```
pip install https://download.lfd.uci.edu/pythonlibs/archived/lxml-4.9.0-cp311-cp311-win_amd64.whl
```

This will download and install LXML version 4.9.0 for Python 3.11 for 64-bit Intel/AMD processors for Windows.

## Installing PDS Deep Archive

Finally, you can install the PDS Deep Archive. As of this writing, version 1.5.0 or later is recommended for Windows. To install it, enter the following command in the same Windows PowerShell with the `(pds)` prompt (then press Enter):

```
pip install pds.deeparchive~=1.5.0
```

Feel free to change the version number in the command as needed.

You can then run `pds-deep-archive --help` or `pds-deep-registry-archive --help` to get a usage message and ensure it's properly installed.

## Upgrading the Software

To check and install an upgrade to the software, run the following command in your virtual environment (on Unix and Unix-like systems):

```
source $HOME/.virtualenvs/pds-deep-archive/bin/activate  
pip install --upgrade pds.deeparchive
```

Or on Windows in PowerShell:

```
.\pds\Scripts\activate.ps1  
pip install --upgrade pds.deeparchive
```

### Note

The same admonitions mentioned earlier about command line invocations also apply to the above examples.