
Installing additional Python packages

Arbitrary third-party Python packages can be added to the MeVisLab Python environment using pip (package installer tool for Python).

The recommended way to install Python packages is via the PythonPip module in MeVisLab. The help to this module also describes some limitations regarding the installation of packages that are using C extensions.



Note

On Windows, if you did install MeVisLab in the default location, you need to be administrator to install Python packages (when installing to a package from the SDK like MeVis/ThirdParty).

You should do this with a freshly started MeVisLab, since DLLs loaded from used Python packages can't be updated.

For your own Python modules that should be shared between MeVisLab modules you can use the `Modules/Scripts/python` directory in each MeVisLab package, which is added to the Python path automatically if it exists. Python files in this directory can be imported in the usual way.

Installing Python packages with PythonPip

Starting with MeVisLab 3.7, PythonPip also allows to install Python packages to other MeVisLab packages, preferably to a user's own package. (Some tinkering with the Python path of the pip tool is involved, so your mileage may vary.) This has the advantage that no administrator rights are necessary for installation, and that the original SDK installation can be left untouched.

On the other hand this can now lead to version conflicts when installing different versions of the same Python package to different (active) MeVisLab packages. You should usually avoid this situation, but if it can't be avoided you should be aware of the order in which Python packages are found: The Python packages installed to a user's MeVisLab package added via the Preferences dialog take precedence over the MeVisLab packages added via the preferences file, which in turn take precedence over the MeVisLab packages installed with the SDK. Amongst these groups no precedence is defined. PythonPip will indicate it when Python packages are available multiple times.

Python packages are installed to the top-level `site-packages` directory of a MeVisLab package when using this feature, with the notable exception of the MeVis/ThirdParty package, where it uses the default `site-packages` directory of the Python installation provided with the MeVisLab SDK instead.

If you want to use the Python packages installed to your own MeVislab package outside of the MeVisLab context (i.e., with a stand-alone Python), you need to add the required `site-packages` directory in Python with `site.addsitedir()` first.

Installing Python packages from the command line

It is also possible to install Python packages from the command line. You need to use the Python executable delivered with the MeVisLab installation for this.

This will only install to the default Python `site-packages` directory.

Example instructions for installing PyTorch on Windows

1. Open a command prompt as administrator
2. Go to: <installation_directory>\Packages\MeVis\ThirdParty\Python.

If you have only one MeVisLab version installed you should be able to do

```
cd %MLAB_ROOT%\MeVis\ThirdParty\Python
```

3. Run one of the following commands:

- Newest CPU-only version:

```
MeVisPython -m pip install torch torchvision
```

- PyTorch 1.10.0 for CUDA 10.2 (adapt to your desired version):

```
MeVisPython -m pip install torch==1.10.0+cu102 torchvision==0.11.1+cu102 -f https://download
```

4. If your network environment requires a proxy use the --proxy argument. E.g.

```
MeVisPython -m pip install --proxy http://<proxy_location>:<proxy_port> torch torchvision
```

Or set the http_proxy and https_proxy environment variables to the same effect.



Note

You can execute the same installation instructions with the PythonPip module by copying everything after "install" in the above instructions into the top-most input field.

Example instructions for installing PyTorch on Linux

1. The instructions for Linux are very similar to Windows. The default installation folder usually just doesn't need administrator rights.
2. Go to: <installation_directory>/Packages/MeVis/ThirdParty/Python/Release/bin.

(This only installs this for the Release mode of MeVisLab. If needed, you can repeat the steps also the in the Debug directory: <installation_directory>/Packages/MeVis/ThirdParty/Python/Debug/bin.

3. Run command:

```
./MeVisPython -m pip install torch torchvision
```

(The only difference to Windows is prepending "./" to execute MeVisPython from the current directory.

4. If you need to use a proxy, the required environment variables are perhaps already set, otherwise the same instructions as for Windows apply.