Skip to content

Python Wheels

TODO

- Guidelines for wheel selecting the version to use.
- Finalize a policy for how conflicting versions of a wheel are handled.

Python wheels (`*.whl`) are the standard way of distributing Python modules. They are supported in Blender to make self-contained Python Extension

Guidelines

- By convention, always locate the files under `./wheels/`.

Requirements

- Wheels must be bundled unmodified from Python's package index.
- Wheels must include their dependencies.
- Wheels filenames must match Python's binary distribution specification: see docs. *Wheels downloaded from Python's package index will follow t convention.*
- Use forward slashes as path separators when listing them on the manifest.

How to Bundle Wheels

Python wheels (`*.whl`) can be bundled using the following steps.

Downloading Wheels

Download the wheel to the directory `./wheels/`.

For wheels that are platform independent this example downloads `jsmin` :

```
pip wheel jsmin -w ./wheels
```

For wheels that contain binary compiled files, wheels for all supported platforms should be included:

This example downloads `pillow` - the popular image manipulation module.

```
pip download pillow --dest ./wheels --only-binary=:all: --python-version=3.11 --platfo
pip download pillow --dest ./wheels --only-binary=:all: --python-version=3.11 --platfo
pip download pillow --dest ./wheels --only-binary=:all: --python-version=3.11 --platfo
```

The available platform identifiers are listed on pillow's download page.

Update the Manifest

In `blender_manifest.toml` include the wheels as a list of paths, e.g.

```
wheels = [
    "./wheels/pillow-10.3.0-cp311-cp311-macosx_11_0_arm64.whl",
    "./wheels/pillow-10.3.0-cp311-cp311-manylinux_2_28_x86_64.whl",
    "./wheels/pillow-10.3.0-cp311-cp311-win_amd64.whl",
]
```

Now installing the package will extract the wheel into the extensions own `site-packages` directory.

Running

Once the extension has been installed you can check the module is being loaded by importing it in the Python console and printing it's location:

```
import PIL
print(PIL.__file__)
```

Platform Builds

Wheels can severely impact the size of an extension. To mitigate this, it is possible to build different extension zip files for each unique required platform.

For this you need to use the `--split-platforms` option from the build command.

```
blender --command extension build --split-platforms
```

Example

Manifest file excerpt:

```
id = "my_addon_with_wheels"
version = "1.0.0"

platforms = ["windows-x64", "macos-x64"]
wheels = [
   "./wheels/pillow-10.3.0-cp311-cp311-macosx_11_0_arm64.whl",
   "./wheels/pillow-10.3.0-cp311-cp311-manylinux_2_28_x86_64.whl",
   "./wheels/pillow-10.3.0-cp311-cp311-win_amd64.whl",
]
```

Generated files from `--split-platforms`:

- my_addon_with_wheels-1.0.0-windows_x64.zip
- my_addon_with_wheels-1.0.0-macos_x64.zip

Note

Even though there is a Linux-only wheel present, no Linux zip file is generated. This happens because the `platforms` field only has Mac and Windows.

Previous
Add-ons

Ne
Creating an Extensions Reposito

View Source
View Translation
Report issue on this page