

[Skip to content](#)

How to Create Extensions

Creating an extension takes only a few steps:

1. Open the directory containing the add-on code or theme file.
2. Add a [blender_manifest.toml](#) file with all the required meta-data (`name`, `maintainer`, `...`).
3. Use the [Blender command-line tool](#) to build the extension `.zip` file.

How to publish to the [Blender Extensions Platform](#):

- [Install from Disk](#) to test if everything is working well.
- [Upload the .zip file](#) (this step requires Blender ID).

The extension will be held for [review](#), and published once the moderation team approves it.

Extension files

An extension is shared as a `.zip` archive containing a manifest file and other files. The expected files depend on the extension type.

Add-on extension

[Add-ons](#) need at least the manifest and an `__init__.py` file, while more complex add-ons have a few different `.py` files or wheels together.

```
my_extension-0.0.1.zip
├─ __init__.py
├─ blender_manifest.toml
└─ (...)
```

Theme extension

A theme extension only needs the manifest and the `.xml` theme file.

```
my_extension-0.0.1.zip
├─ blender_manifest.toml
└─ theme.xml
```

Note

Extensions can optionally have all its files inside a folder (inside the archive). This is a common behavior when saving a repository as ZIP from version-control platforms.

Manifest

A manifest is a file with all the meta-data required for an extension to be processed. This example is a good starting point to the `blender_manifest.toml` that should be inside the `.zip`.

```
schema_version = "1.0.0"

# Example of manifest file for a Blender extension
# Change the values according to your extension
id = "my_example_extension"
version = "1.0.0"
name = "My Example Extension"
tagline = "This is another extension"
maintainer = "Developer name <email@address.com>"
```

```

# Supported types: "add-on", "theme"
type = "add-on"

# # Optional: link to documentation, support, source files, etc
# website = "https://extensions.blender.org/add-ons/my-example-package/"

# # Optional: tag list defined by Blender and server, see:
# # https://docs.blender.org/manual/en/dev/advanced/extensions/tags.html
# tags = ["Animation", "Sequencer"]

blender_version_min = "4.2.0"
# # Optional: Blender version that the extension does not support, earlier versions are supported
# # This can be omitted and defined later on the extensions platform if an issue is found.
# blender_version_max = "5.1.0"

# License conforming to https://spdx.org/licenses/ (use "SPDX:" prefix)
# https://docs.blender.org/manual/en/dev/advanced/extensions/licenses.html
license = [
    "SPDX:GPL-3.0-or-later",
]
# # Optional: required by some licenses.
# copyright = [
#     "2002-2024 Developer Name",
#     "1998 Company Name",
# ]

# # Optional: list of supported platforms. If omitted, the extension will be available in all
# platforms = ["windows-x64", "macos-arm64", "linux-x64"]
# # Other supported platforms: "windows-arm64", "macos-x64"

# # Optional: bundle 3rd party Python modules.
# # https://docs.blender.org/manual/en/dev/advanced/extensions/python_wheels.html
# wheels = [
#     "./wheels/hexdump-3.3-py3-none-any.whl",
#     "./wheels/jsmin-3.0.1-py3-none-any.whl",
# ]

# # Optional: add-ons can list which resources they will require:
# # * files (for access of any filesystem operations)
# # * network (for internet access)
# # * clipboard (to read and/or write the system clipboard)
# # * camera (to capture photos and videos)
# # * microphone (to capture audio)
# #
# # If using network, remember to also check `bpy.app.online_access`
# # https://docs.blender.org/manual/en/dev/advanced/extensions/addons.html#internet-access
# #
# # For each permission it is important to also specify the reason why it is required.
# # Keep this a single short sentence without a period (.) at the end.
# # For longer explanations use the documentation or detail page.
#
# [permissions]
# network = "Need to sync motion-capture data to server"
# files = "Import/export FBX from/to disk"

```

```
# clipboard = "Copy and paste bone transforms"

# # Optional: advanced build settings.
# # https://docs.blender.org/manual/en/dev/advanced/extensions/command_line_arguments.html
# [build]
# # These are the default build excluded patterns.
# # You only need to edit them if you want different options.
# paths_exclude_pattern = [
#     "__pycache__/",
#     "/.git/",
#     "/*.zip",
# ]
```

Required values:

blender_version_min:

Minimum supported Blender version - use at least 4.2.0.

id:

Unique identifier for the extension.

license:

List of [licenses](#), use [SPDX license identifier](#).

maintainer:

Maintainer of the extension.

name:

Complete name of the extension.

schema_version:

Internal version of the file format - use 1.0.0.

tagline:

One-line short description, up to 64 characters - cannot end with punctuation.

type:

“add-on”, “theme”.

version:

Version of the extension - must follow [semantic versioning](#).

Optional values:

blender_version_max:

Blender version that the extension does not support, earlier versions are supported.

website:

Website for the extension.

copyright:

Some licenses require a copyright, copyrights must be “Year Name” or “Year-Year Name”.

tags:

List of tags. See the [list of available tags](#).

platforms:

List of supported platforms. If omitted, the extension will be available in all operating systems. The available options are [“windows-x64”, “windows-arm64”, “macos-x64”, “macos-arm64”, “linux-x64”]

wheels:

List of relative file-paths [Python Wheels](#).

permissions:

Add-ons can list which resources they require. The available options are *files*, *network*, *clipboard*, *camera*, *microphone*. Each permission should be followed by an explanation (short single-sentence, up to 64 characters, with no end punctuation).

Optional values for “build”:

These values are only used by the [build](#) sub-command.

paths:

A list of file-paths relative to the manifest to include when building the package.

paths_exclude_pattern:

A list of file-path patterns to exclude include when building the package.

The pattern matching is compatible with [gitignore](#).

Note that setting this value isn’t supported when `paths` is also declared.

If the `[build]` table isn’t declared the following default is used:

```
[build]
paths_exclude_pattern = [
  "__pycache__/",
  ".*",
  "*.zip",
]
```

Reserved:

These values **must not** be declared in a TOML and are reserved for internal use.

- `[build.generated]`

Note

All the values present in the manifest file must be filled (i.e., cannot be empty, nor text `" "`, nor list `[]`).

If you don’t want to set one of the optional values just exclude it from the manifest altogether.

Command-line

Extensions can be built, validated & installed via command-line.

To build the package defined in the current directory use the following commands:

```
blender --command extension build
```

See [build](#) docs.

To validate the manifest without building the package:

```
blender --command extension validate
```

You may also validate a package without having to extract it first.

```
blender --command extension validate add-on-package.zip
```

See [validate](#) docs.

See also

[Extensions Command Line Arguments](#).

Third party extension sites

If you want to host the extensions yourself, see the [Creating an Extensions Repository](#) docs.

[Previous](#)
[Creating Extensions](#)

Copyright © : This page is licensed under a CC-BY-SA 4.0 Int. License

[No](#)
[Extension Licen](#)

Made with [Furo](#)
Last updated on 2025-05-10

[View Source](#)
[View Translation](#)
[Report issue on this page](#)