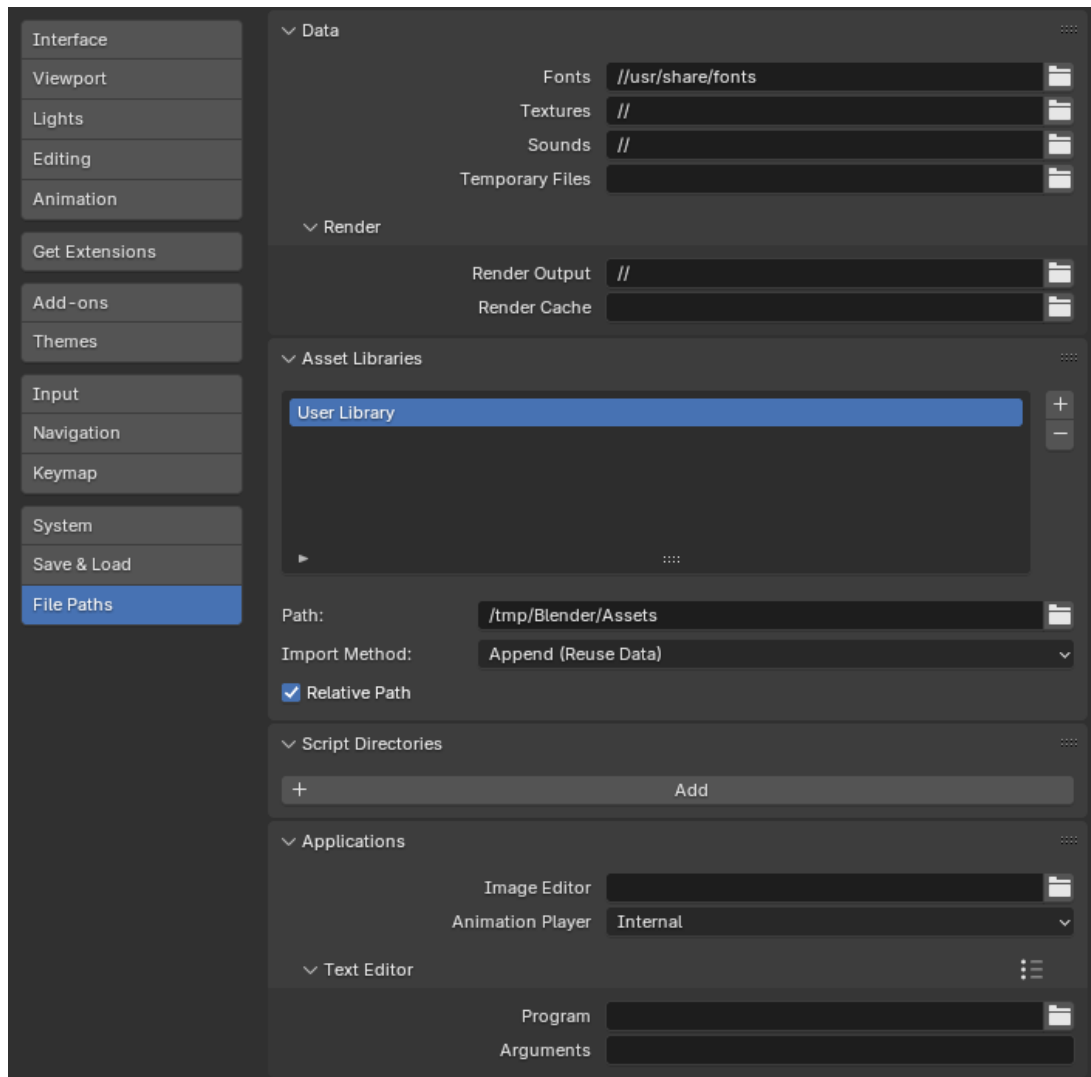


[Skip to content](#)

# File Paths

The *File* section in *Preferences* allows you to configure auto-save preferences and set default file paths for blend-files, rendered images, and more.

Locations for various external files can be set for the following options:



Preferences File Paths section.

## Hint

The default path `//` refers to the folder of the currently open blend-file (see [Relative Paths](#) for details).

## Data

### Fonts

Default location to browse for [text object](#) font files.

### Textures

Default location to browse for image textures.

### Sounds

Default location to browse for sound files.

### Temporary Files

The directory for storing temporary save files. The path must reference an existing directory or it will be ignored and the systems temporary directory will be used instead. When left blank, the systems temporary directory will be used (see [Temporary Directory](#) for details).

## Render

### Render Output

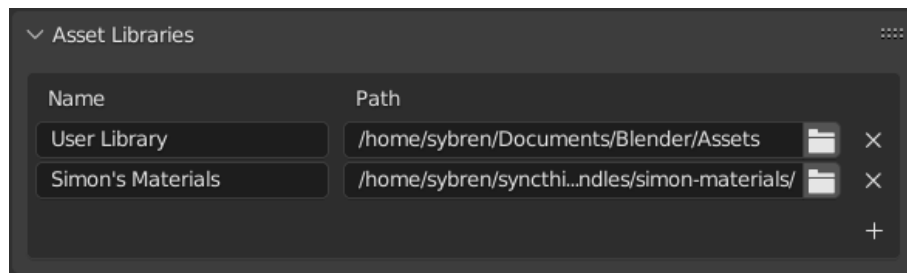
Where rendered images/videos are saved.

### Render Cache

The location where cached render images are stored.

## Asset Libraries

Name and on-drive directory paths of asset libraries. To make Blender aware of an asset library, add it to this list. The name is for your reference only, and will appear in asset library selectors. The path should point to the location of the asset library.



Name and Location of asset libraries in the Preferences.

To create a new asset library, just create an empty directory and add it to the [List View](#). Any asset from any blend-file contained in that directory (or subdirectories thereof) will appear in the [Asset Browser](#).

### Import Method

Determines how data is managed when an asset is imported, unless overridden by the [Asset Browser](#).

#### Link

The asset will be linked to the current blend-file, and thus be read-only. Later changes to the asset file will be reflected in all files that link it :

#### Append

All of the asset and all its dependencies will be appended to the current file. Dragging a material into the scene three times will result in three independent copies. Dragging an object into the scene three times will also result in three independent copies.

“Dependencies” in this case means everything the asset refers to. For an object, this can be its mesh and materials, but also other objects used by modifiers, constraints, or drivers.

Since the file now has its own copy of the asset, later changes to the asset file will not be reflected in the file it’s appended to.

### Append (Reuse Data)

*Specific to the Asset Browser.*

The first time an asset is used, it will be appended, including its dependencies, just like described previously. However, Blender will keep track of where it originated, and the next time the asset is used, as much data as possible will be reused. Dragging a material into the scene three times will only load it once, and just assign the same material three times. Dragging an object into the scene three times will create three copies of the object, but all copies will share their mesh data, materials, etc.

Since the file now has its own copy of the asset, later changes to the asset file will not be reflected in the file it’s appended to.

### Relative Path

Use relative path when linking assets from this asset library.

## Script Directories

Additional locations to search for Python scripts.

Each path can be given a *Name* to signify to purpose of that script directory.

By default, Blender looks in several directories (platform dependent) for scripts. By adding a user script path in the preferences an additional directory is

used. This can be used to store your own scripts and add-ons independently of the current Blender version.

You will need to create specific subfolders in this path which match the structure of the `scripts` folder found in Blender's installation directory.

The following subdirectories will be used when present:

#### **startup/**

Modules in this folder will be imported on startup.

#### **addons/**

Legacy add-ons located here will be listed in the add-ons preferences.

#### **modules/**

Modules in this folder can be imported by other scripts.

#### **presets/**

Presets in this folder will be added to existing presets.

#### Hint

For add-ons it is now recommended to use a local extension repository if you wish to define additional locations to install and manage them.

To make use of these you will need to define them as [extensions](#).

#### Note

You have to restart Blender for all changes to the users scripts to take effect.

## Applications

### Image Editor

The path to an external program to use for image editing.

### Animation Player

The program used for playing back rendered animations via [View Animation](#).

By default this is set to *Internal* which uses Blender's built-in [animation player](#).

This has the advantage that all image formats supported by Blender can be played back and no 3rd party application needs to be installed.

## Text Editor

### Program

Command to launch the text editor when using [Edit Externally](#), either a full path or a command in `$PATH`. Use the internal editor when left blank

### Arguments

Defines the specific format of the arguments with which the text editor opens files.

The supported expansions are as follows:

- `$filepath`: The absolute path of the file.
- `$line`: The line to open at (Optional).
- `$column`: The column to open from the beginning of the line (Optional).
- `$line0` & `$column0` similar to the above but they start at zero.

Example: `-f $filepath -l $line -c $column`

## Development

Only visible when [Developer Extras](#) are enabled.

### I18n Branches

The path to the `/branches` directory of your local SVN translation copy, to allow translating from the UI.

## Known Limitations

### Permissions on Windows

Be sure that you have the right privileges for running the executable accessing the path defined. On Windows for instance, if the option “Run this program as an administrator” is enabled for the executable, it will lead to a failure to open the editor due to a limitation within the OS User Account Control. Running a program with elevated privileges is potentially dangerous!

[Previous](#)  
[Save & Load](#)

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

Made with [Furo](#)

Last updated on 2025-05-10

[No](#)  
[Experimenter](#)

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