Skip to content

Deploying Blender in Production

This page contains tips for setting up Blender in environments such as animation studios and schools.

These environments often have special requirements regarding security, automated deployment and customization.

Installing Blender

Blender downloads can be extracted to any directory on the system, as a self contained installation. Multiple Blender versions can co-exist on the same system, and deployment can be automated using standard file management tools.

New Blender versions may add, remove or change functionality that affects the results of production files. For a given project, it is advisable to use a sing LTS version of Blender. LTS versions receive bug fixes for two years.

Working Offline

For security or other reasons, workstation may not have internet access.

By default Blender does not access the internet, however this can be enabled in the System preferences with the Online Access option.

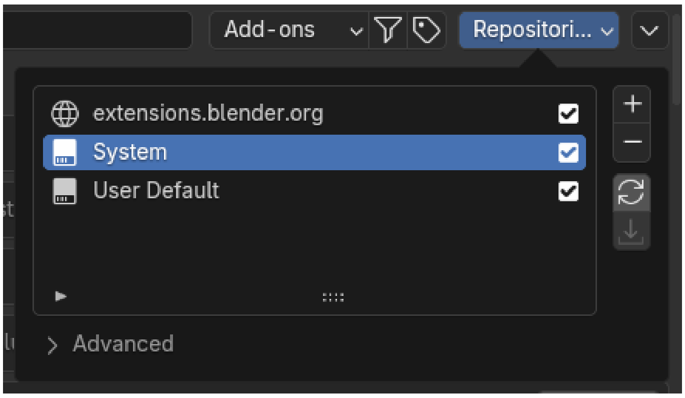
Working offline can be enforced by running with the `--offline-mode` command line argument. Users will then be unable to enable online access the preferences.

Note

Add-ons that follow this setting will only connect to the internet if enabled. However, Blender cannot prevent third-party add-ons from violating this rule

Bundling Extensions

When working offline or in a more controlled environment, it may be useful to provide a set of extensions to all users. For this there is a default read-only System repository. This repository could be located on a read-only network drive or in a system directory.



System repository

The `$BLENDER_SYSTEM_EXTENSIONS` environment variable controls the default location. This should point to a directory, within which a `system` directory should exist.

Extensions packages should be extracted in this `system` directory, with a resulting path like this:

```
$BLENDER_SYSTEM_EXTENSIONS/system/my-addon/blender_manifest.toml
```

In the Extensions preferences, it’s possible to manually set a custom directory for the default System repository, or to create multiple repositories.

Bundling Scripts

Besides extensions, it’s possible to bundle scripts for presets, application templates, legacy add-ons, as well as scripts run on startup.

Script directories can be manually added in the File Paths preferences. `$BLENDER_SYSTEM_SCRIPTS` can also be used to add script directories without modifying the preferences.

Script directories are expected to contain specific subdirectories like `presets`, `addons` and `startup` for different types of scripts. See Path Layout for a complete list.

Startup Scripts

The Blender Python API can be used to customize Blender. This includes changing preferences, changing the startup file and adding UI elements.

For example, a script can enable add-ons for every user.

```
$BLENDER_SYSTEM_SCRIPTS/startup/enable_addons.py
```

```python
def register():
    import addon_utils
    addon_utils.enable("my-addon")

def unregister():
    pass

if __name__ == "__main__":
    register()
```

Application Templates

Application Templates can be used to set up Blender for particular tasks or projects, separate from the default configuration. When creating a new file th user can choose the template.

The files are expected to be placed in the system script directories like this:

```
$BLENDER_SYSTEM_SCRIPTS/startup/bl_app_templates_system/MyTemplate/__init__.py
$BLENDER_SYSTEM_SCRIPTS/startup/bl_app_templates_system/MyTemplate/startup.blend
```

Legacy Add-ons

Add-ons that have not been converted to become an extension yet need to be placed in the `addons` script directory.

For example, an add-on could be located at:

```
$BLENDER_SYSTEM_SCRIPTS/addons/simple_addon.py
$BLENDER_SYSTEM_SCRIPTS/addons/complex_addon/__init__.py
```

Splash Screen

When Blender is configured for a particular studio or a project, it can be helpful to customize the splash screen so artists know which version they are running.

The `BLENDER_CUSTOM_SPLASH` environment variable replaces the entire splash image, while `BLENDER_CUSTOM_SPLASH_BANNER` on overlays a banner.

VFX Platform

Blender follows the VFX reference platform, which means it is able to run on the same systems as other VFX software and exchange image, volume and scene files with them.

Python Version

Python Version

Blender and the bpy module are only compatible with a single Python version. This makes it possible for add-ons and VFX software in general to only have to target a single Python version.

Blender bundles a complete Python installation and does not interact with the system Python by default. This can be changed with the `--python-use-system-env` command line argument, if care is taken to set up a compatible Python version.

Previous
Blender’s Directory Layout

Ne
Appendic

View Source
View Translation
Report issue on this page