



- blender.org
- code.blender.org

User:Ideasman42/BlenderAsPyModule

Log in

< User:Ideasman42

- User page
- Discussion
- View source
- History

Page

- What links here
- Related changes
- Permanent link

From BlenderWiki

Overview

The official **blender.org** embeds a Python interpreter (CPython 3.x). This makes sense from the perspective of a user, who's primary needs are to have an application which can be extended with scripts.

However from the perspective of a Python developer, it can be useful to being Blender into your existing scripts and access its feature set.

The option to build Blender as a Python module is not officially supported, in the sense Blender.org isn't distributing it along with regular releases, Currently, its a build option you can enable, for your own use.

Rationale

This is a build option to be able to import blender into python and access its modules

Possible uses include:

- rendering animations.
- image processing using Blender's compositor.
- video editing (using Blender's sequencer).
- importers, exporters (convert 3D file formats).
- development, *accessing bpy from Python IDE's and debugging tools for example.*
- automation.

This is mainly limited to features which can be usable in background mode, so you cant for instance do OpenGL preview renders.

Building

assuming you have a CMake out-of-source build setup, see:

http://wiki.blender.org/index.php/Dev:Doc/Building_Blender/Linux/Ubuntu/CMake

OSX prerequisites: get Python3.4-Framework from Python.org and install it.

Change these CMake options from the defaults:

```
WITH_PYTHON_INSTALL=OFF  
WITH_PLAYER=OFF  
WITH_PYTHON_MODULE=ON
```

Now build blender normally with 'make install'

Everything should build as normal except in the cmake directory you will have ./bin/bpy.so instead of ./bin/blender

Installation

Linux

To make this available for python copy these files to your site-packages by running

```
make install
```

Or if you like to do it manually

```
cp ./bin/bpy.so /usr/lib/python3.4/site-packages/  
cp --recursive ./bin/2.71 /usr/lib/python3.4/site-packages/
```

Windows

```
copy bin\bpy.pyd C:\Python34\Lib\site-packages\  
copy bin\*.dll C:\Python34\Lib\site-packages\  
del C:\Python34\Lib\site-packages\python34.dll  
xcopy /E bin\2.71 C:\Python34\
```

OSX

After compiling and "make install", copy needed files to your python framework

```
cp ./bin/bpy.so /Library/Frameworks/Python.framework/Versions/3.4/lib/python3.4/site-packages/  
cp -R ./bin/2.71 /Library/Frameworks/Python.framework/Versions/3.4/lib/python3.4/site-packages/
```

*Note. Unlike on *nix C:\Python34\2.71 is not in the site packages, this is because of a difference in how Windows finds the scripts path and should eventually be fixed.*

Testing

This allows 'bpy' to be imported from python or other applications/IDE's which embed python, eg:

```
python -c "import bpy ; bpy.ops.render.render(write_still=True)"
```

This runs in background mode and has similar restrictions to running a script:

```
blender --background --python test.py
```

Troubleshooting

- The Python version requirements are the same with building a regular blender binary (if Blender is using Python3.4 then there is NO WAY to use another version - 2.7/3.2/3.4 will all fail).
- On Windows, you probably won't want to use a debug build, since this requires a debug python installation (python34_d.dll rather than python34.dll), so while it can be made to work, it's more trouble.

Retrieved from "<http://wiki.blender.org/index.php/User:Ideasman42/BlenderAsPyModule>"

Contents

- 1 Overview
- 2 Rationale
- 3 Building
- 4 Installation
 - 4.1 Linux
 - 4.2 Windows
 - 4.3 OSX
- 5 Testing
- 6 Troubleshooting



Wiki

- [Report a wiki bug](#)
- [Wiki Guidelines](#)
- [Special pages](#)
- [Categories](#)
- [Popular pages](#)
- [New files](#)
- [New pages](#)
- [Recent changes](#)

