
Armada

Release 0.0.1

Mike Bourbeau

Aug 07, 2020

CONTENTS

1	Getting Started	3
1.1	Installation	3
1.2	Projects Explorer	3
1.3	Asset Manager	3
1.4	Task Management	3
2	Development	5
2.1	Getting Started	5
2.2	Utils	6
2.3	Adding Software	6
2.4	Hooks	8
2.5	Multithreading	8
3	API Reference	9
3.1	mb_armada	9
3.2	mb_utils	9
4	Indices and tables	11

Important: Current dev: *Alpha*

Armada is the main pipeline tool from which all project files (Maya, Houdini, etc) are launched from.

Any files launched from Armada will be hooked into the pipeline, allowing the user access to a library of scripts, plugins, automated file versioning workflows, and asset management tools.

GETTING STARTED

1.1 Installation

This is how to install

1.2 Projects Explorer

Main file browser

1.3 Asset Manager

This is the asset manager maybe

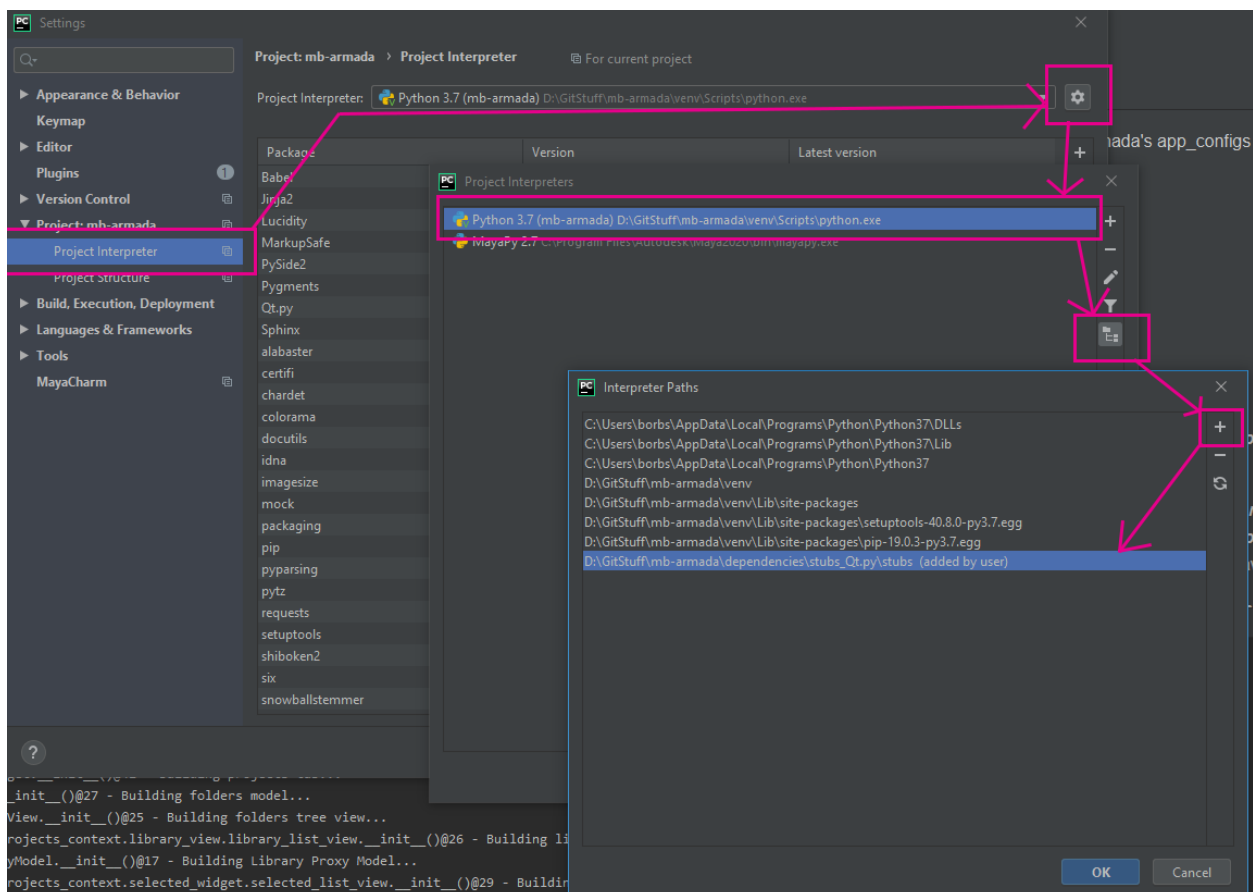
1.4 Task Management

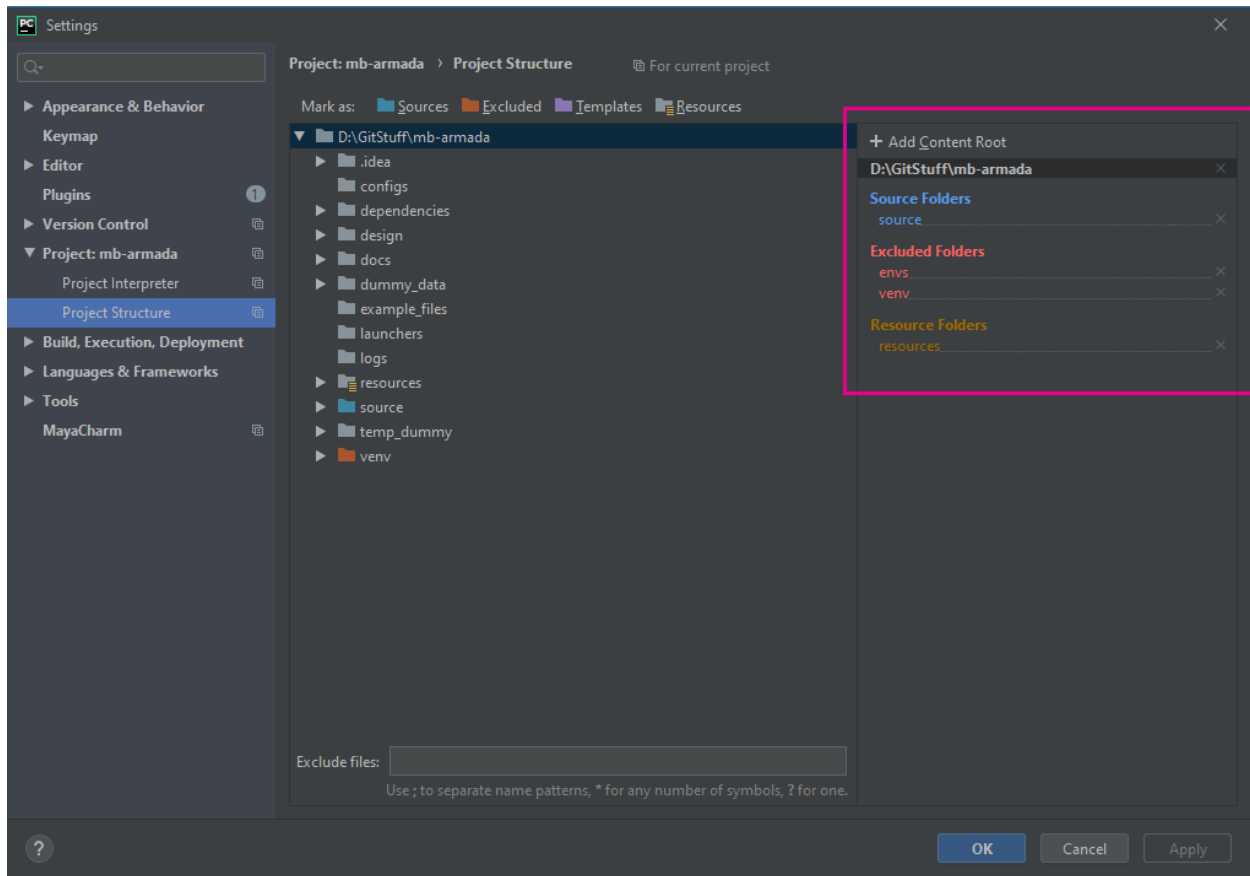
Asana browser window

DEVELOPMENT

2.1 Getting Started

- install requirements.txt
- add qtstubs to path
- Maya dev kit: <https://www.autodesk.com/developer-network/platform-technologies/maya>





2.2 Utils

2.2.1 Logger

2.2.2 Path Resolver

2.2.3 Data Resolver

2.3 Adding Software

1. **Create an app config file** `mb_armada/app_configs/software.json`
2. **Create an app path resolver** `mb_utils/path_resolver/resolver_templates/armada/apps/software/software_template.py`
3. **Create a launch hook** `mb_armada/hooks/launchers/software_hook.py` 3a. **Optional: Create pythonpath directories to launch other software or plugins at startup**
`mb_armada/hooks/launchers/apps/software/scripts` `mb_armada/hooks/launchers/apps/software/plugins`
4. Add app icon to resources

2.3.1 Configs

- An app config file should be placed in Armada’s app_configs directory and should be named like this: software.json.

- JSON file data

```
{
  "working_dir": "dir_name",
  "extension": "ext"
}
```

- **working_dir**: The folder name of the directory that the software typically saves to

- Example: Maya’s working_dir is called “scenes”

- **“extension”**: Some programs can save in multiple file formats (binary, ascii). For these programs it’s useful to restrict use

- Example: Maya has binary and ascii formats for its save files.

- **Ascii is good for working files because things can become corrupt and having the ability to read and edit the file**

- * Binary isn’t editable, but the file size is smaller. This makes it a good format for asset exporting.

2.3.2 Resolver Templates

- A template file in mb_util’s resolver templates path that contains all the extra directories required for a software

- Example: Maya’s working directory, scenes, has many sibling directories such as images, assets, renderData, etc.

- With these templates you can add your own custom directories on top of the defaults.

2.3.3 Launch Hooks

- A launch hook runs the software and sets all the necessary environment variables from within the hook

- Some software will allow you to add paths to

- * If so add a software folder to hook/app and add any necessary folders such as “scripts” or “plugins”

- You can create your own environment variables here

2.4 Hooks

2.5 Multithreading

API REFERENCE

3.1 mb_armada

3.1.1 mb_armada.gui

mb_armada.gui.tasks_context.tasks_tab

mb_armada.gui.tasks_context.tasks_widget

3.1.2 mb_armada.hooks

mb_armada.hooks.apps

mb_armada.hooks.copy_path_hook

mb_armada.hooks.make_dirs_hook

mb_armada.hooks.open_explorer_hook

3.1.3 mb_armada.resource

3.2 mb_utils

3.2.1 mb_utils.mb_logger

3.2.2 mb_utils.path_resolver

INDICES AND TABLES

- genindex
- modindex
- search