
PolyRename

Andrew Simmons, Ethan Smith

Feb 14, 2021

CONTENTS

1	Contents	3
1.1	Concepts	3
1.2	Usage	3
1.3	For Developers	6
1.4	Changelog	7

Welcome to the documentation for [PolyRename](#), a cross-platform, bulk-file rename tool.

CONTENTS

1.1 Concepts

1.1.1 Transformation

Transformations are small, atomic operations that modify a filename in a simple way. For example, the *Suffix* transformation appends text to the end of a file's name, while the *Replace Extension* transformation replaces the selected files' extension with a specified extension.

1.1.2 Pipeline

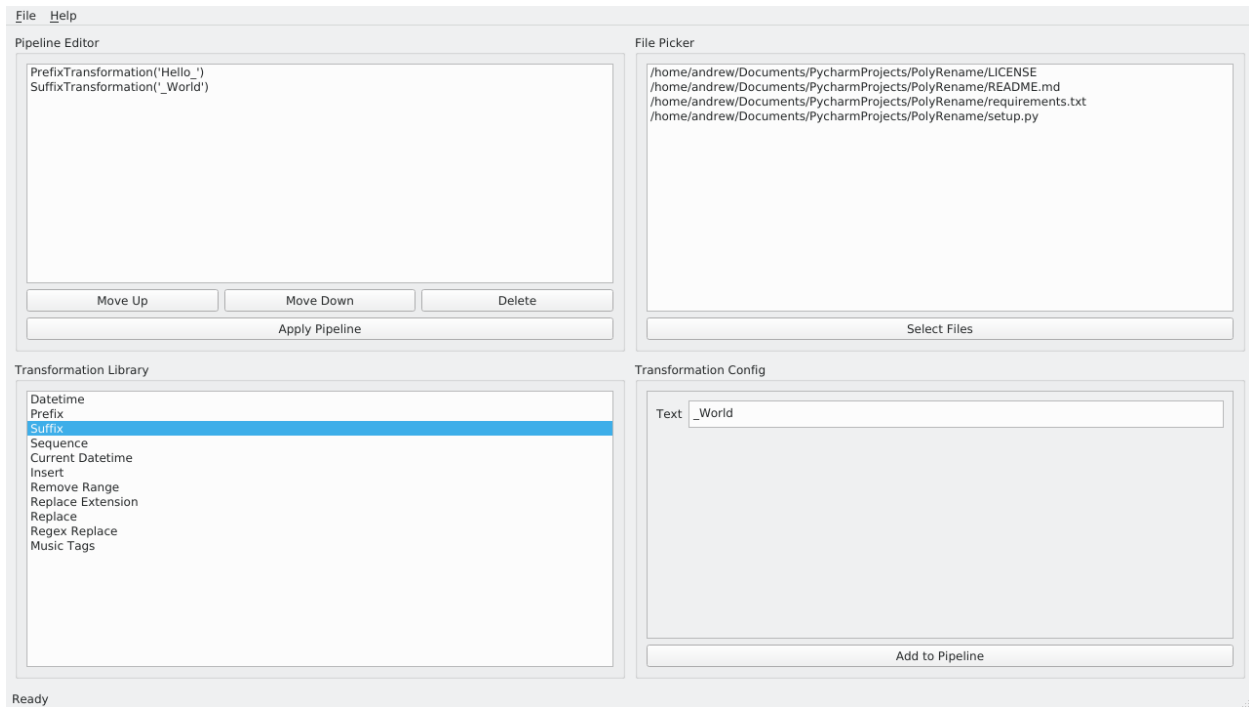
The pipeline is an ordered sequence of transformations which are applied to a file sequence. The first transformation is applied, then the second is applied to the output of the first, and so on. This allows the user to construct advanced rename operations out of the simple transformations.

1.2 Usage

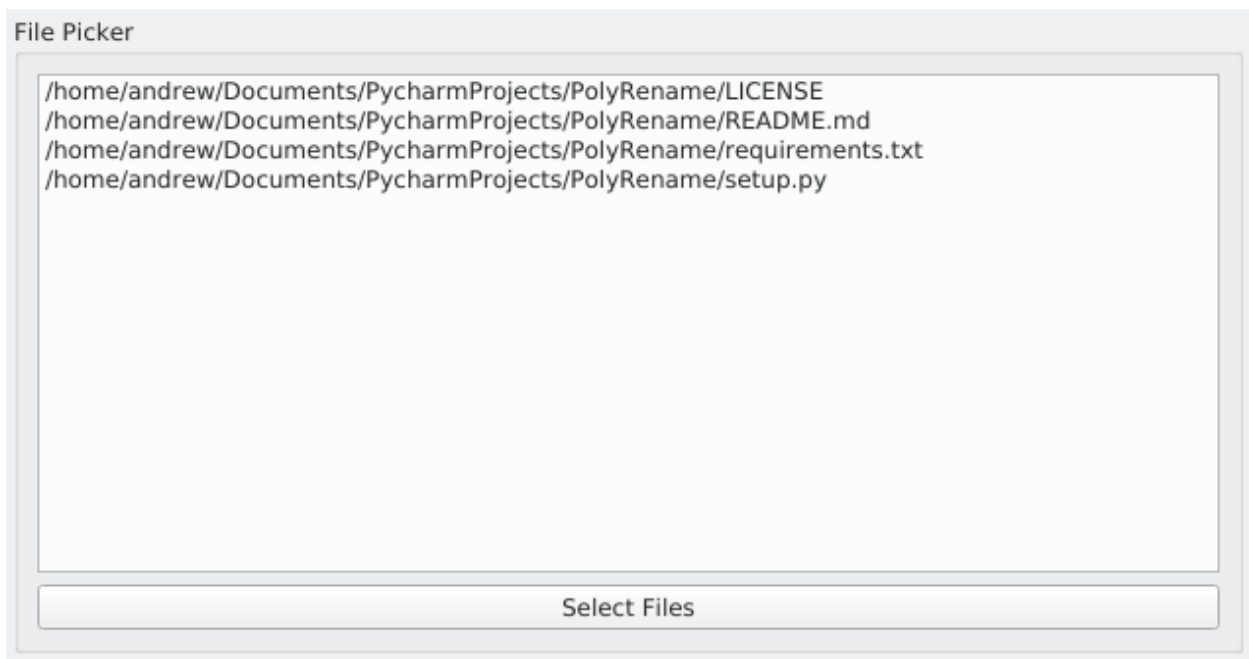
This section contains information on how to use PolyRename. Most users will want to view the *GUI* documentation.

1.2.1 GUI

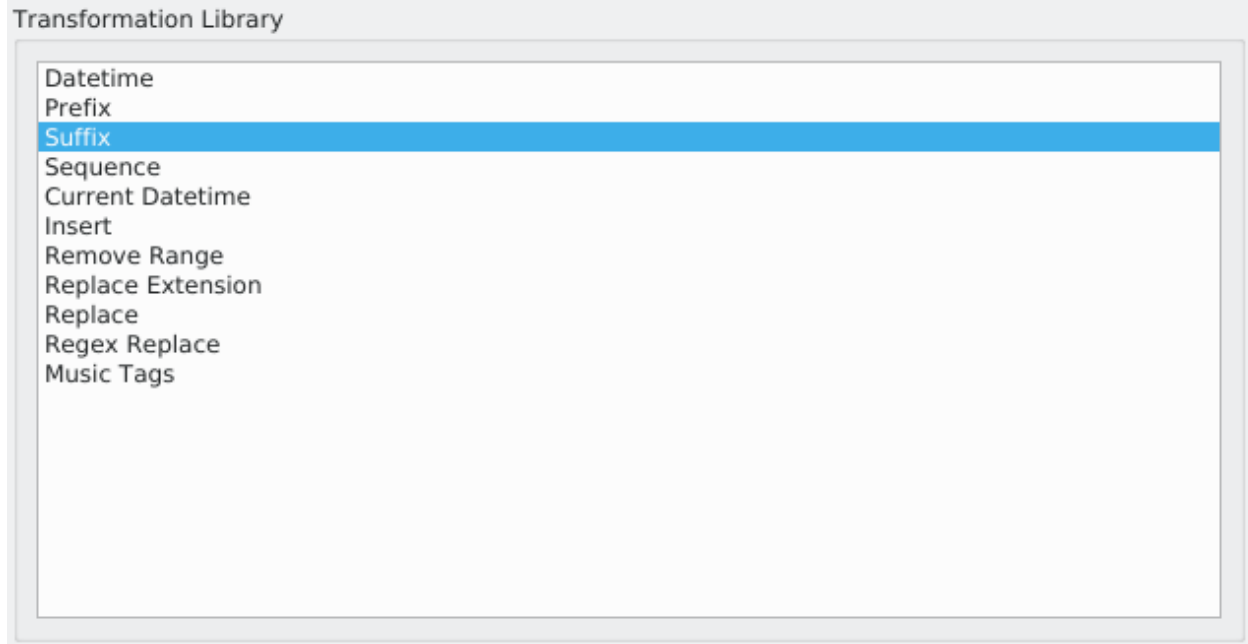
The user interface of PolyRename was designed to be simple to use, unlike many other bulk-file rename tools. It's GUI interface is divided into 4 quadrants.



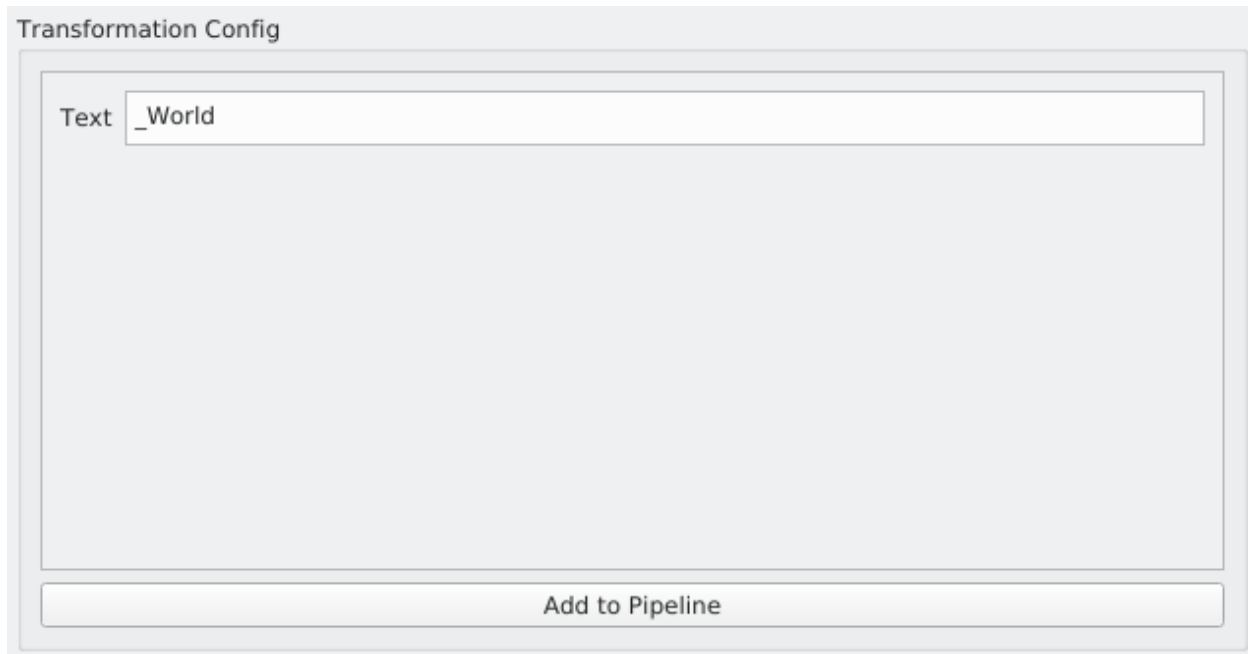
At the top right is the File Picker. The file picker is used to select files on the user's computer, and to visualize which files (and in what order) are currently selected.



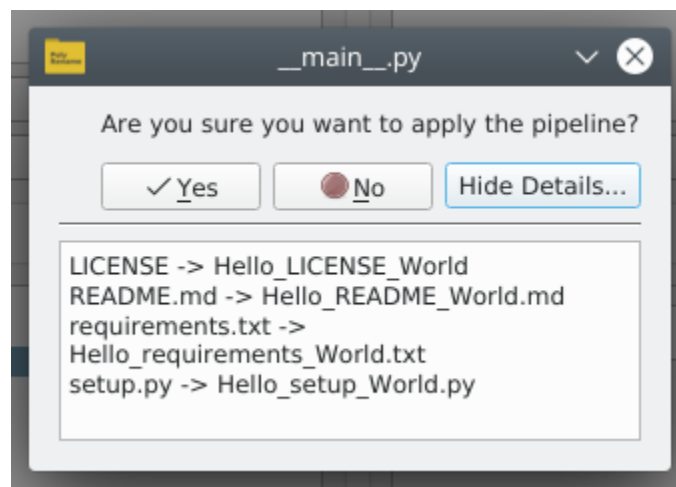
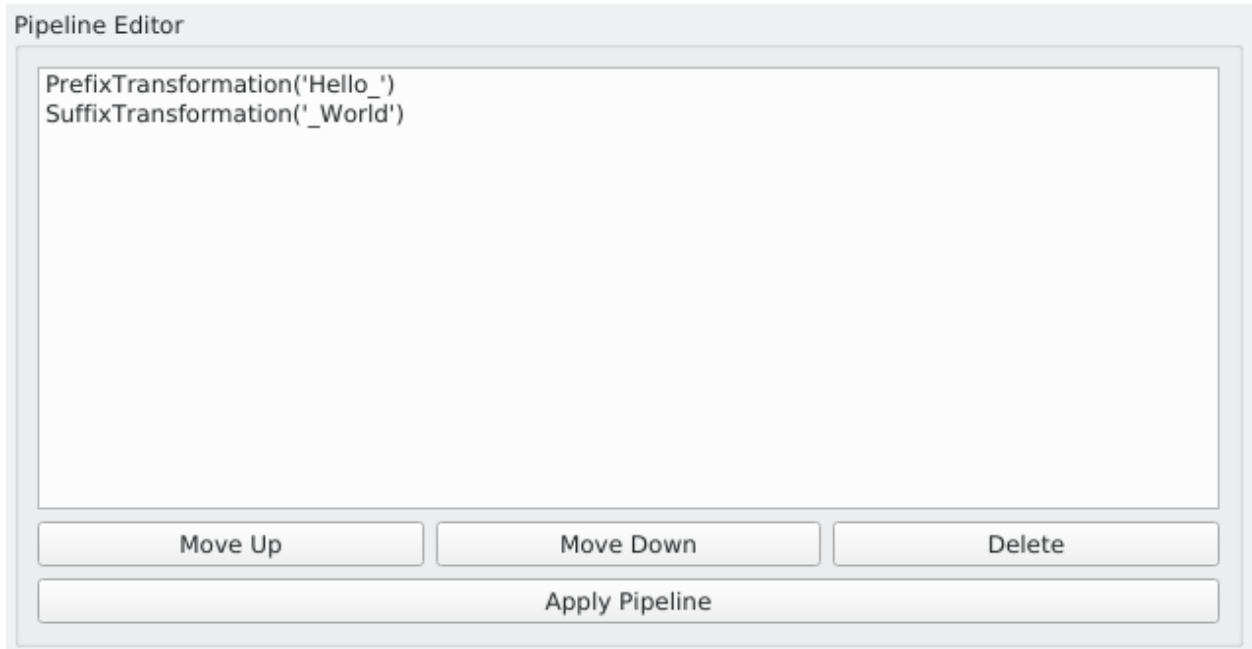
At the bottom left is the Transformation Library. The transformation library is automatically populated with the available transformations. From this interface, the user can select a transformation to begin configuring that transformation.



At the bottom right is the Transformation Config. This interface allows the user to configure the selected transformation before appending the configured transformation to the pipeline. This interface performs error checking to ensure that the user has entered a value for all required fields, and that all entered values are of the expected datatype.



Finally, at the top left is the Pipeline Editor. The pipeline editor shows you the current state of your pipeline, and allows you to modify your pipeline. Modification actions include moving a configured transformation up or down in the pipeline's sequence or deleting a configured transformation from the pipeline. Finally, this interface allows the user to apply a pipeline to their selected file sequence. Upon pressing the "Apply Pipeline" button, a preview of the resulting file sequence modifications is shown.



1.3 For Developers

This section contains information on how PolyRename is structured, along with how to extend PolyRename through its *Plugin* system.

1.3.1 Git

Branches

PolyRename follows the [Gitflow](#) branching model. The basics are that development is done in the *develop* branch, and *master* is used for stable release code.

1.3.2 Plugin

Transformations can be user-created and added to PolyRename. To do this, place a .py file in the `polyrename/transformation/` package, and modify `polyrename/transformation/__init__.py` such that your newly created transformation is imported.

Transformations must do the following to be considered valid:

- Define a class which inherits from *Transformation*
- Define a static variable *schema* which conforms to the defined specification
- Define a `__init__` method which takes an argument for each option defined in the schema in the same order as the schema defines it
- Define a **resolve** method which takes a file sequence as an argument, and returns a file sequence
- Define a `__repr__` method. This will be used to display the configured transformation in the Pipeline Editor

See the existing transformations for reference when creating your own.

1.4 Changelog

1.4.1 1.1.0 (in development)

- Drop support for Python 3.5
- Use Poetry for dependency management

1.4.2 1.0.1 (2019-07-26)

- Use MediaFile instead of pytag in the `music_file` transformation

1.4.3 1.0.0 (2019-07-26)

- Initial Release