Sequential File Renamer

A cross-platform utility that makes batch renaming files simple, consistent, and powerful.

Overview

Sequential File Renamer allows you to rename multiple files with a consistent naming pattern, automatically adding sequential numbering with adaptive padding. It's ideal for organizing photos, documents, and any collection of files that need a uniform naming scheme.

Features

- **Sequential Renaming with Adaptive Padding**
  - Single digit padding for fewer than 10 files (photo_1.jpg)
  - Double digit padding for 10-99 files (photo_01.jpg)
  - Triple digit padding for 100+ files (photo_001.jpg)
- **File Selection Flexibility**
  - Process entire directories at once
  - Select specific files to rename
  - Work with files across different folders
- **Extension Handling**
  - All extensions automatically converted to lowercase
  - Normalization of non-standard extensions to common formats
  - Example: .JPEG → .jpg, .Tiff → .tif, .Markdown → .md
- **Additional Options**
  - Add date prefix in YYYYMMDD format
  - Filter files by extension

Installation

Prerequisites

- Python 3.6+
- Tkinter (usually included with Python)
- Pydantic (automatically installed by installer)

Linux Installation

```
# Extract the package, then:
cd file-renamer
chmod +x install.sh
./install.sh
```

For system-wide installation:

```
sudo ./install.sh
```

Windows Installation

Simply download and run the Windows executable (.exe file).

Manual Installation Steps

1. Clone this repository:

   ```
   git clone https://github.com/yourusername/file-renamer.git
   ```

2. Navigate to the directory:

   ```
   cd file-renamer
   ```

3. Install required dependencies:

   ```
   pip install pydantic
   ```

4. Run the application:

   ```
   python3 main.py
   ```

Usage Guide

Step 1: Select Files

Choose one of two selection methods:

- **Directory Mode**:
  1. Click “Browse Directory…”
  2. Select a folder containing files to rename
  3. Choose “Process entire directory”
- **File Selection Mode**:
  1. Click “Select Files…”
  2. Choose individual files (use Ctrl/Shift for multiple selections)
  3. Choose “Process selected files only”

Step 2: Configure Options

1. **Enter Base Name**
   - Type the text you want to use as the base name (e.g., “photo”)
   - Result: “photo_1.jpg”, “photo_2.jpg”, etc.
2. **Optional Settings**
   - Include Date: Adds current date (YYYYMMDD_) as prefix
   - Normalize Extensions: Converts uncommon extensions to standard ones
   - Filter by Extension: Specify which file types to process (e.g., “jpg,png,txt”)

Step 3: Preview and Apply

1. Click “Generate Preview” to see how files will be renamed
2. Review the changes in the preview panel
3. Click “Rename Files” to apply the changes

Extension Normalization

When enabled, these file extensions are automatically converted:

Original	Normalized	Original	Normalized
.jpeg	.jpg	.mpeg	.mpg
.tiff	.tif	.mov	.mp4
.htm	.html	.text	.txt
.markdown	.md	.midi	.mid

Examples

Basic Sequential Renaming:

```
Before:                 After (base name "photo"):
IMG_1234.jpg   →        photo_1.jpg
20050123.jpg   →        photo_2.jpg
scan0003.jpg   →        photo_3.jpg
```

With Date Prefix:

```
Before:                 After (base name "vacation"):
DSC_9876.jpg   →        20240311_vacation_1.jpg
IMG_2468.jpg   →        20240311_vacation_2.jpg
```

With Extension Normalization and Case Conversion:

```
Before:                 After (base name "image"):
picture.JPEG   →        image_1.jpg
photo.Tiff     →        image_2.tif
file.TXT       →        image_3.txt
```

Code Structure

The application is organized into several modules for better maintainability:

```
file-renamer/

  main.py                  # Main application entry point
  models.py                # Data models with Pydantic
  file_operations.py       # File handling logic
  ui_components.py         # UI components
  __init__.py                # Package definition
  icon.png                   # Application icon (PNG format)
  icon.ico                   # Application icon (Windows format)
  LICENSE                    # License file
  README.md                  # Documentation
  install.sh                 # Installation script
  uninstall.sh               # Uninstallation script
```

Each module has a specific responsibility: - **main.py**: Contains the main application class that ties everything together - **models.py**: Contains all data models using Pydantic - **file_operations.py**: Contains file handling and renaming logic - **ui_components.py**: Contains reusable UI components

Uninstallation

To remove the application:

```
# For user installation:
file-renamer-uninstall

# For system-wide installation:
sudo file-renamer-uninstall
```

You can also uninstall directly from the source directory:

```
cd file-renamer
chmod +x uninstall.sh
./uninstall.sh
```

Building from Source

Building for Linux

```
# Install PyInstaller
pip3 install pyinstaller

# Navigate to the source directory
cd file-renamer

# Build the executable
pyinstaller --name FileRenamer --windowed --onefile main.py
```

Building for Windows

```
# Install PyInstaller
pip install pyinstaller

# Navigate to the source directory
cd file-renamer
```

```
# Build the executable
pyinstaller --name FileRenamer --windowed --onefile --icon=icon.ico main.py
```

Troubleshooting

Common Installation Issues

- **Command not found**: If you get "command not found" errors, make sure your PATH includes the bin directory
- **Missing Python/Tkinter**: Ensure Python 3 and Tkinter are installed on your system
- **Permission errors**: Check if you need to run with sudo for system-wide installation
- **Application doesn't appear in menu**: Try logging out and back in to refresh the application menu

Runtime Issues

- **File access errors**: Ensure the application has read/write permissions for the selected directory and files
- **UI glitches**: Some themes may not display correctly; try changing your system theme
- **Regular expression errors**: Double-check your regex patterns for syntax errors
- **"No files found" error**: Make sure you've either selected files or chosen a directory with files in it
- **"Cannot rename file" error**: Check if another application has the file open or locked

License

This project is licensed under the MIT License - see the LICENSE file for details.

Contributing

Contributions are welcome! Please feel free to submit a Pull Request.