

# Packaging the File Renamer Application

This guide explains how to package the File Renamer application as an executable for both Windows and Linux platforms.

## Prerequisites

Before packaging, make sure you have the following installed:

- Python 3.6 or higher
- pip (Python package installer)

## Method 1: Using PyInstaller (Recommended)

PyInstaller is a popular tool that bundles Python applications into stand-alone executables.

### Installation

```
pip install pyinstaller
```

### Packaging for Windows

1. Open a command prompt
2. Navigate to the directory containing your script
3. Run the following command:

```
pyinstaller --name FileRenamer --windowed --icon=icon.ico --add-data "icon.ico;." file_renamer.py
```

The packaged application will be available in the `dist/FileRenamer` directory.

### Packaging for Linux

1. Open a terminal
2. Navigate to the directory containing your script
3. Run the following command:

```
pyinstaller --name FileRenamer --windowed file_renamer.py
```

The packaged application will be available in the `dist/FileRenamer` directory.

## Method 2: Using cx\_Freeze

cx\_Freeze is another tool for freezing Python scripts into executables.

### Installation

```
pip install cx_Freeze
```

## Create a setup.py file

Create a file named `setup.py` in the same directory as your script with the following content:

```
import sys
from cx_Freeze import setup, Executable

# Dependencies
build_exe_options = {
    "packages": ["os", "re", "tkinter", "datetime", "threading", "queue"],
    "excludes": [],
    "include_files": []
}

# Base
base = None
if sys.platform == "win32":
    base = "Win32GUI"

setup(
    name="FileRenamer",
    version="1.0",
    description="File Renaming Application",
    options={"build_exe": build_exe_options},
    executables=[Executable("file_renamer.py", base=base, icon="icon.ico")]
)
```

## Building the Executable

### For Windows:

```
python setup.py build
```

### For Linux:

```
python3 setup.py build
```

The packaged application will be available in the `build` directory.

## Adding an Icon

For both methods, you'll need an icon file:

1. For Windows, use a `.ico` file
2. For Linux, use a `.png` file

Make sure to place the icon file in the same directory as your script before packaging.

## Creating a Desktop Shortcut

### Windows

A desktop shortcut is automatically created when you use the `--create-shortcut` option with PyInstaller:

```
pyinstaller --name FileRenamer --windowed --icon=icon.ico --add-data "icon.ico;" --create-s
```

### Linux

After installing, create a `.desktop` file in `~/.local/share/applications/` with the following content:

```
[Desktop Entry]
Name=File Renamer
Exec=/path/to/executable/FileRenamer
Icon=/path/to/icon.png
Type=Application
Categories=Utility;
```

## Troubleshooting

- If you encounter missing module errors, add them to the `packages` list in your PyInstaller command or `cx_Freeze` setup
- For tkinter issues on Linux, make sure you have `python3-tk` package installed
- If the application works but the GUI doesn't appear, ensure you're using the `--windowed` flag with PyInstaller