To package your Python application as a .deb file and ensure it runs in the background, you'll need to follow a multi-step process. This involves creating a specific directory structure for the Debian package and, importantly, creating a systemd service file to manage the background execution.

Here's a detailed breakdown of the steps:

### Part 1: Modify the Python Script for Background Execution

The original script with the GUI is not designed to run headless (without a graphical interface). When you run it as a background service, it won't have a display to show the window. To solve this, you need to separate the wallpaper-changing logic from the GUI. The GUI will be used for configuration, and the background service will read that configuration.

**Step 1: Create a configuration file.**

The GUI needs a place to save the user's settings (the wallpaper folder and the duration). A simple text file or a JSON file is perfect for this. Let's use a config.json file.

**Step 2: Modify the Python script.**

The script will now have two modes of operation:

1. **GUI Mode**: When run normally, it will show the GUI, allow the user to select the folder and duration, and then save these settings to config.json.
2. **Service Mode**: When a special flag (e.g., --service) is provided, it will read the settings from config.json and start the wallpaper changing loop *without* a GUI.

**Updated wallpaper\_changer.py:**

Python

import os

import random

import time

import subprocess

import json

import tkinter as tk

from tkinter import filedialog, simpledialog, messagebox

import sys

CONFIG\_FILE = os.path.expanduser("~/.wallpaper\_changer\_config.json")

LOG\_FILE = os.path.expanduser("~/.wallpaper\_changer.log")

def log(message):

"""Simple logging function for the background service."""

with open(LOG\_FILE, "a") as f:

f.write(f"{time.strftime('%Y-%m-%d %H:%M:%S')} - {message}\n")

def set\_wallpaper(image\_path):

"""Sets the wallpaper for the Cinnamon desktop environment."""

try:

command = ['gsettings', 'set', 'org.gnome.desktop.background', 'picture-uri', f'file://{image\_path}']

subprocess.run(command, check=True)

log(f"Wallpaper set to: {image\_path}")

return True

except FileNotFoundError:

log("Error: gsettings command not found. Please ensure it's installed and in your PATH.")

return False

except subprocess.CalledProcessError as e:

log(f"Error: Failed to set wallpaper: {e}")

return False

def run\_service(wallpaper\_folder, duration\_minutes):

"""The main wallpaper changing loop for the background service."""

log("Starting wallpaper changer service...")

while True:

try:

wallpapers = [os.path.join(wallpaper\_folder, f) for f in os.listdir(wallpaper\_folder)

if f.lower().endswith(('.png', '.jpg', '.jpeg', '.gif', '.bmp'))]

if not wallpapers:

log("Error: No valid image files found in the selected folder.")

time.sleep(60) # Wait a bit before checking again

continue

random\_wallpaper = random.choice(wallpapers)

set\_wallpaper(random\_wallpaper)

except Exception as e:

log(f"An unexpected error occurred: {e}")

time.sleep(duration\_minutes \* 60)

class WallpaperChangerApp:

def \_\_init\_\_(self, master):

self.master = master

master.title("Linux Mint Wallpaper Changer")

self.config = self.load\_config()

# UI Elements

self.folder\_label = tk.Label(master, text=f"Folder: {self.config.get('wallpaper\_folder', 'Not set')}")

self.folder\_label.pack(pady=10)

self.select\_button = tk.Button(master, text="Select Wallpaper Folder", command=self.select\_folder)

self.select\_button.pack(pady=5)

self.duration\_label = tk.Label(master, text=f"Duration: {self.config.get('duration\_minutes', 10)} minutes")

self.duration\_label.pack(pady=10)

self.duration\_button = tk.Button(master, text="Set Duration (mins)", command=self.set\_duration)

self.duration\_button.pack(pady=5)

self.save\_button = tk.Button(master, text="Save Settings", command=self.save\_config)

self.save\_button.pack(pady=10)

def load\_config(self):

"""Loads configuration from a JSON file."""

if os.path.exists(CONFIG\_FILE):

with open(CONFIG\_FILE, 'r') as f:

return json.load(f)

return {"wallpaper\_folder": None, "duration\_minutes": 10}

def save\_config(self):

"""Saves configuration to a JSON file."""

with open(CONFIG\_FILE, 'w') as f:

json.dump(self.config, f, indent=4)

messagebox.showinfo("Settings Saved", "Configuration has been saved.")

log("GUI: Settings saved.")

def select\_folder(self):

folder\_path = filedialog.askdirectory()

if folder\_path:

self.config["wallpaper\_folder"] = folder\_path

self.folder\_label.config(text=f"Folder: {os.path.basename(folder\_path)}")

messagebox.showinfo("Folder Selected", f"The wallpaper folder has been set to:\n{self.config['wallpaper\_folder']}\n\nRemember to click 'Save Settings'!")

def set\_duration(self):

try:

new\_duration = simpledialog.askinteger("Set Duration", "Enter duration in minutes:",

initialvalue=self.config.get('duration\_minutes', 10),

minvalue=1)

if new\_duration is not None:

self.config["duration\_minutes"] = new\_duration

self.duration\_label.config(text=f"Duration: {self.config['duration\_minutes']} minutes")

except ValueError:

messagebox.showerror("Invalid Input", "Please enter a valid number.")

if \_\_name\_\_ == "\_\_main\_\_":

if "--service" in sys.argv:

# Run in service mode

if os.path.exists(CONFIG\_FILE):

with open(CONFIG\_FILE, 'r') as f:

config = json.load(f)

folder = config.get("wallpaper\_folder")

duration = config.get("duration\_minutes")

if folder and os.path.isdir(folder) and duration:

run\_service(folder, duration)

else:

log("Error: Configuration file is invalid or incomplete. Please run the GUI to configure the settings.")

else:

log("Error: Configuration file not found. Please run the GUI to configure the settings.")

else:

# Run in GUI mode

root = tk.Tk()

app = WallpaperChangerApp(root)

root.mainloop()

### Part 2: Create the .deb Package

This is the most complex part. We'll use a manual approach to create the Debian package structure.

**Step 1: Set up the project directory.**

Create a main directory for your package. The name should be packagename-version.

Bash

mkdir wallpaper-changer\_1.0-1

cd wallpaper-changer\_1.0-1

**Step 2: Create the Debian control directory.**

This directory contains metadata about your package.

Bash

mkdir DEBIAN

Inside the DEBIAN directory, create a file named control.

DEBIAN/control

Package: wallpaper-changer

Version: 1.0-1

Architecture: all

Maintainer: Your Name <your.email@example.com>

Depends: python3, python3-tk

Description: A simple wallpaper changer for Linux Mint.

This application provides a GUI to configure a wallpaper folder and

duration, and then runs as a background service to change the wallpaper

randomly.

**Step 3: Create the installation and service files.**

We need to place our Python script and a systemd service file in the correct directories so they get installed properly.

Create the directory structure for installation:

Bash

mkdir -p usr/local/bin

mkdir -p usr/lib/systemd/user

* usr/local/bin: This is a standard location for local executables. We'll place our main script here.
* usr/lib/systemd/user: This is where user-level systemd service files are stored. We'll place a service file here to make our script run in the background.

Now, copy your modified Python script into the usr/local/bin directory.

Bash

cp /path/to/your/wallpaper\_changer.py usr/local/bin/

Make the script executable:

Bash

chmod +x usr/local/bin/wallpaper\_changer.py

**Step 4: Create the systemd service file.**

Create a file named wallpaper-changer.service inside usr/lib/systemd/user.

usr/lib/systemd/user/wallpaper-changer.service

[Unit]

Description=Linux Mint Wallpaper Changer

After=graphical.target

[Service]

Type=simple

ExecStart=/usr/bin/python3 /usr/local/bin/wallpaper\_changer.py --service

Restart=always

TimeoutSec=10

[Install]

WantedBy=default.target

**Explanation of the systemd file:**

* [Unit]: Describes the service. After=graphical.target ensures it starts after the graphical desktop environment is ready, which is crucial for our gsettings command to work.
* [Service]: Defines how to run the service.
  + Type=simple: The process will be the main process of the service.
  + ExecStart: The command to execute. We use python3 to run our script and pass the --service flag.
  + Restart=always: This is important. If the script crashes for any reason, systemd will automatically restart it.
* [Install]: Defines how the service is enabled. WantedBy=default.target means the service should be started when the user logs in.

**Step 5: Create post-installation scripts.**

We need a script to enable and start the systemd service after the package is installed.

Create a file named postinst inside the DEBIAN directory.

DEBIAN/postinst

Bash

#!/bin/bash

# Enable and start the systemd user service after installation

systemctl --user daemon-reload

systemctl --user enable wallpaper-changer.service

systemctl --user start wallpaper-changer.service

exit 0

Also, create a prerm script to stop and disable the service when the package is uninstalled.

DEBIAN/prerm

Bash

#!/bin/bash

# Stop and disable the systemd user service before removal

systemctl --user stop wallpaper-changer.service

systemctl --user disable wallpaper-changer.service

exit 0

Make sure these scripts are executable:

Bash

chmod 755 DEBIAN/postinst DEBIAN/prerm

### Part 3: Build the .deb Package

Now that the directory structure is complete, you can build the .deb file.

From the parent directory of wallpaper-changer\_1.0-1, run the following command:

Bash

dpkg-deb --build wallpaper-changer\_1.0-1

This will create a wallpaper-changer\_1.0-1.deb file in the current directory.

### Part 4: Install and Use the Package

1. **Install the .deb package:**

Bash

sudo dpkg -i wallpaper-changer\_1.0-1.deb

(Note: You will need to install python3-tk as a dependency first if it's not already installed: sudo apt-get install python3-tk).

1. **Configure the service:**
   * Find the application in your application menu (or run wallpaper\_changer.py).
   * Use the GUI to select your wallpaper folder and set the duration.
   * Click "Save Settings". This creates the ~/.wallpaper\_changer\_config.json file.
2. **Check the service status:**

Bash

systemctl --user status wallpaper-changer.service

You should see that the service is active (running).

The service will now run in the background, reading the configuration you set and changing your wallpaper automatically. It will start automatically every time you log in to your desktop.