No File Found
Software Deployment Plan

Latest update: 2023-11-01

1. System Requirements

Support operating system:

- Windows 10 and above
- macOS 13 Ventura and above
- Linux SuSE Leap 15 and above
- Rocky 8
- Ubuntu14 and above
- Fedora 37 and above

Required dependencies:

- Windows
  - none
- MacOS
  - none
- Linux
  - RPM package manager

2. Deployment Strategy Summary

Our software will be deployed as a standalone application written in PyQT6. We will make use of the PyInstaller tool to package our application into executable files for each supported operating system. The final distribution will be available as:

- An .exe file for Windows.
- A .dmg file for MacOS.
- An .rpm package for Linux distributions that support the RPM package manager, and a .deb package for those that use the DEB package manager (like Ubuntu).

3. Installation Package Contents

3.1 Required source or compiled files

- no_file_found.exe
- no_file_found.dmg
- no_file_found.rpm

3.2 Required third-party components

- PyQT6 runtime libraries

3.3 Required graphical assets, configuration and other non-program files

- Application icons and imagery
- Default configuration file (if any)
- Theme or skin files (if applicable)

3.4 Documentation files to be provided

- README.md: Contains general information about the software and links to detailed documentation.

3.5 Development files and components that must be excluded

- Any debugging tools or logs
- Source code (unless open source)
- Test files and test data
- py cache

4. Additional Code Required for Deployment

- A script to automate the creation of .rpm and .deb packages for Linux distributions.
- A script to clean up any temporary files created during the installation process.

5. Deployment Tasks

1. Compile and test the PyQT6 application in a development environment.
2. Use PyInstaller to create executables for each OS.
3. Create .rpm and .deb packages for Linux distributions.
4. Ensure all necessary assets and files are bundled.
5. Write/update documentation files.
6. Conduct internal testing of the installation packages on supported operating systems.
7. Fix any identified bugs or issues.
8. Finalize the installation packages.
9. Distribute the installation packages to users via a github release.

6. Deployment Test Plan

1. Installation Test: Ensure that the software installs correctly without any errors on all supported operating systems.
2. Functional Test: Verify that all functionalities of the software work as expected on different platforms.
3. Uninstallation Test: Check that the software uninstalls completely..