# TABLE OF CONTENTS

***STEP 1: CREATE A PULL REQUEST TO MERGE DEV INTO STABLE*** ..... 3
***STEP 2: TAG THE STABLE BRANCH FOR A RELEASE*** ..... 3
***STEP 3: TRIGGER THE RELEASE WORKFLOW*** ..... 3
***STEP 4: DOWNLOAD THE RELEASE*** ..... 4
***NOTES FOR RELEASES*** ..... 4
***CONCLUSION*** ..... 4

# 01.02. CREATING A RELEASE

Once you've finished customizing and testing your configurator, follow these steps to create a release version:

## STEP 1: CREATE A PULL REQUEST TO MERGE DEV INTO STABLE

1. Go to your forked repository on GitHub.
2. Navigate to the **Pull Requests** tab and click **New pull request**.
3. Set the base branch to stable and the compare branch to dev.
4. Review the changes in the pull request and click **Create pull request**.
5. Once the pull request is created, review and merge it into the stable branch.

## STEP 2: TAG THE STABLE BRANCH FOR A RELEASE

1. After merging, ensure you are on the stable branch:

```
git checkout stable
git pull origin stable
```

2. Create a version tag for the release:

```
git tag v1.0.0
git push origin v1.0.0
```

    1. Replace `v1.0.0` with your desired version number following semantic versioning conventions (e.g., `v1.1.0`, `v2.0.0`).

## STEP 3: TRIGGER THE RELEASE WORKFLOW

The push of a version tag (e.g., `v1.0.0`) to the stable branch will automatically trigger the

GitHub Actions workflow for creating a release. This workflow will:

- Package the application for both Windows (.exe) and macOS (.app).
- Upload the compiled files as assets to the release in GitHub.

## STEP 4: DOWNLOAD THE RELEASE

1. Once the release workflow is complete, go to the **Releases** section in your repository.
2. Find the newly created release (e.g., `v1.0.0`).
3. Download the compiled assets (`Configurator.exe` for Windows and `Configurator.app` for macOS).

### NOTES FOR RELEASES

- **Branch Management**: Keep the stable branch clean and only merge fully tested changes from the dev branch.
- **Versioning**: Use semantic versioning for tags (vX.Y.Z) to indicate major, minor, and patch updates.
- **Automated Workflow**: Ensure your GitHub Actions workflows (build.yml and others) are correctly configured for release tagging.

## CONCLUSION

By following these steps, you'll ensure a smooth release process and maintain a clear separation between development and production code.

From:
https://laswitchtech.com/ - **LaswitchTech**

Permanent link:
**https://laswitchtech.com/en/projects/ini-configurator/documentation/01/02/index**

Last update: **2024/12/05 08:38**