



How To Create a Release Candidate

Author	Date	Comment
Nathan Rasch	Abc 20XX	Document inception

Table of Contents

<u>Purpose</u>	<u>3</u>
<u>Intended Audience</u>	<u>3</u>
<u>What is a Release Candidate?</u>	<u>3</u>
<u>Release Candidate Creation Process</u>	<u>3</u>
<u>Creating a New Release</u>	<u>3</u>
<u>Resolving Changes Made to an Existing Release Candidate</u>	<u>4</u>
<u>Finishing and Tagging the Release</u>	<u>4</u>

Purpose

The purpose of this document is to describe the steps needed to create a release candidate for submission to the Dev Ops team for deployment into the UAT and/or Production environment(s).

Intended Audience

The intended audience of this document is anyone who will be involved in creation and management of a release candidate. These groups are typically:

- Development
- Quality Assurance

What is a Release Candidate?

A release candidate is a bolus of code being considered for production deployment. The release candidate typically requires a full set of regression tests before being approved for production deployment.

Release Candidate Creation Process

Creating a New Release

1. Check out **master** and pull it
2. Check out **develop** and pull it
3. Execute the following command: *git flow release start <release version>*
 - a. *git flow release start 2.1.7*
4. Edit the **“.RELEASE_NUM”** file and update it with the release’s version number
 - a. Ex: 2.1.7
5. Edit the **“CHANGELOG.md”** file and add to the release notes
 - a. Note: You can use the instructions and previous release notes already in the file for examples on what a new release note entry should look like
6. Commit and push the **release/<release version>** branch
 - a. Note: The commit message be in the following format: *Preparing release <release version>*
 - i. Ex: *Preparing release 2.1.7*
7. Browse to Jenkins (<https://jenkins.companyx.com>)
 - a. Navigate to the **“CompanyX”** tab and select the **“companyx-branch-build”** job
 - b. Next, select **“Build with Parameters”** from the left hand navigation menu
 - c. Update the **“buildNum”** field with the release number which comes after **“release/”** in the branch name.
 - i. Ex: 2.1.7
 - d. Select **“release/”** from the **“branch”** select menu
 - e. Click on the **“Build”** button
8. Once the build finishes, click **“Console Output”** from the left hand navigation menu
 - a. Scroll to the bottom and record the release candidate job parameter
 - i. Ex: *release-2.1.7112*

9. Inform QC of the new release candidate job parameter via email
10. Fill out and submit the Dev Ops Change Management form with the release candidate job parameter requesting they push the release to UAT/PROD/etc.
 - a. Dev Ops Change Management form URL: <https://companyx.url.com>

Resolving Changes Made to an Existing Release Candidate

If/when a defect is found and mitigated in the release code (or other change), the following steps should be performed:

1. Check out and pull **develop** branch
2. Check out and pull **release/<release version>** branch
3. Merge the **develop** branch into the **release/<release version>** branch
4. Push the **release/<release version>** branch
5. Now repeat the “**Jenkins**” steps from above

Finishing and Tagging the Release

Once QC approves the release it should be finished and tagged:

1. Check out and pull the **master** branch
2. Check out and pull the **develop** branch
3. Check out and pull the **release/<release version>** branch
4. Execute the following commands:
 - a. *git flow release finish <release build number>*
 - i. Ex: *git flow release finish 2.1.7*
 - ii. Note that this command creates the git tag for you...
 - b. *git push*
 - i. *The previous git flow command should have left you on the **develop** branch, which you are pushing.*
 - c. *git checkout master*
 - d. *git push*
 - e. *git push origin v<release build number>*
 - i. Ex: *git push origin v2.1.7*

You're done!