

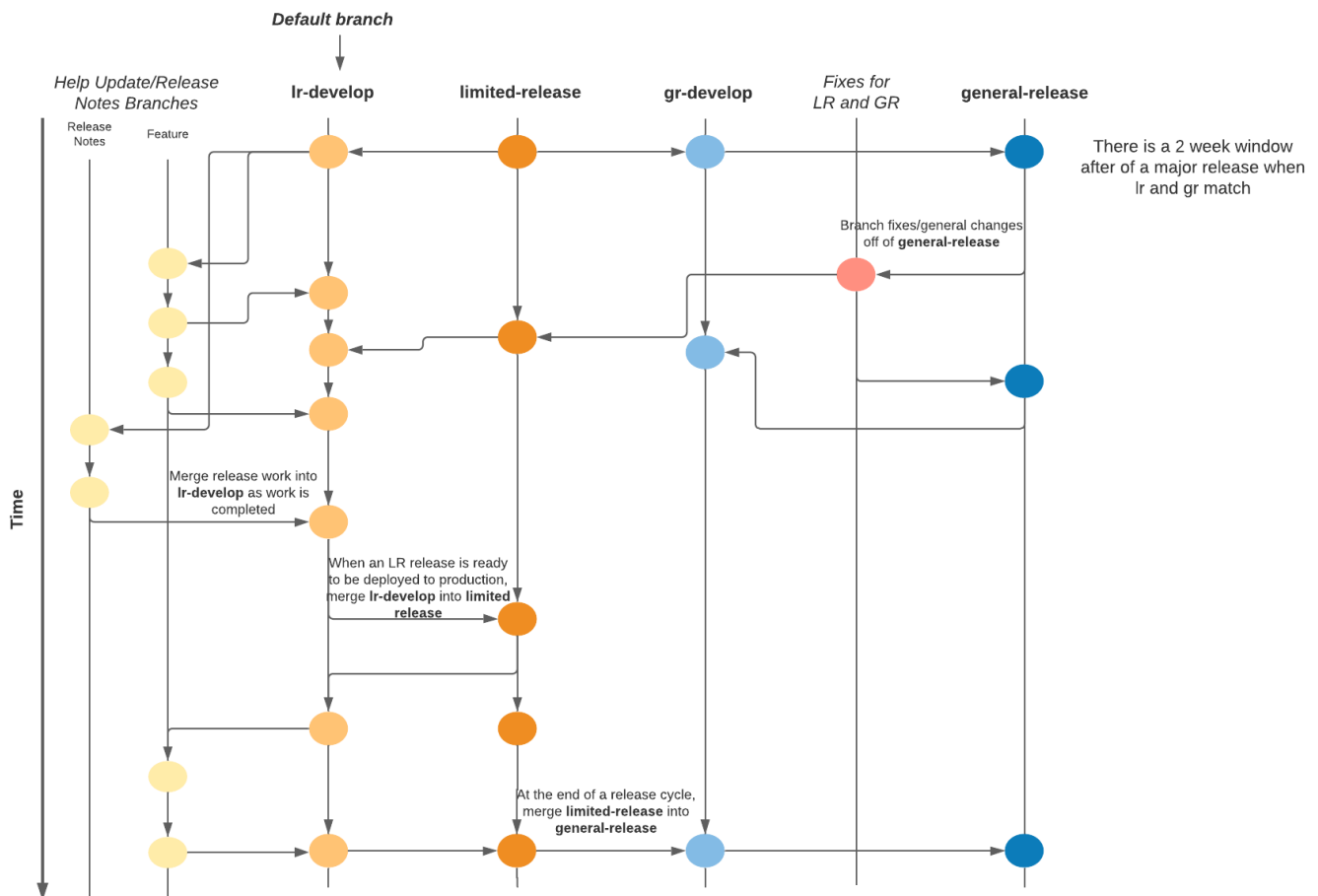
Git Branch and Merge Strategy (Online Help)

NOTE: This page is for the Safety writing team and completely unrelated to the dev team branching strategy. This page describes the GitLab repository for the online help website, [VaultSafetyDocumentation](#).

- [Flow Overview](#)
- [Branches](#)
 - [Permanent Branches](#)
 - [Temporary Branches](#)
 - [New Features](#)
 - [Release Notes & RIA](#)
 - [Limited Release Notes](#)
 - [General Release Notes \(About the yyRx Release\) & RIA](#)
 - [Fixes and Enhancements \(Unrelated to New Features\)](#)
 - [For Both LR and GR](#)
 - [For LR Only](#)
 - [For GR Only](#)

Flow Overview

The following diagram shows the overall flow for branching and merging during release content development:



Branches

Permanent Branches

Branch Name	Description	Auto-Deployment Location
lr-develop (default)	<p>Content development for the limited release website. This is a branch we use to preview our latest changes before they go live on LR.</p> <p>Always branch off lr-develop for the following content updates:</p> <ul style="list-style-type: none"> • New Features • LR Release Notes <p>Once the content changes for a new feature or LR release notes is drafted and ready for review, merge the branch back into lr-develop.</p>	<p>Each merge or commit to this branch triggers an auto-deployment to:</p> <p>https://safetystage.veevavault.help/lr/ (Internal Staging Site)</p>
limited-release	<p>The stable version of the limited release website at safety.veevavault.help/lr. Only commit or merge changes to this branch if they are finalized and ready to go live.</p> <p>On the release date for a limited release, at approximately 6pm or at the time of the Vault Safety update, lr-develop is merged into limited-release and the limited release help updates are automatically deployed.</p>	<p>Each merge or commit to this branch triggers an auto-deployment to:</p> <p>https://safety.veevavault.help/lr/ (Public Production Site)</p>
gr-develop	<p>Content development for the general release website. This is a branch we use to preview changes before they go live on GR.</p> <p>Branch off of gr-develop for the following content updates:</p> <ul style="list-style-type: none"> • Help fixes or enhancements that should go on both LR <i>and</i> GR (typos, customer/internal feedback, missing information) • Adding or updating the RIA • Adding or updating the About the Release page or sub-pages (release notes) for a major release, such as 20R3 <p>Once the content changes for the above items are final, merge them back into gr-develop to preview the changes. Once the updates are ready to go-live, merge gr-develop into both general-release and limited-release.</p>	<p>Each merge or commit to this branch triggers an auto-deployment to:</p> <p>https://safetystage.veevavault.help/gr/ (Internal Staging Site)</p>
general-release ("master")	<p>The stable version of the limited release website at safety.veevavault.help/gr. Only commit or merge changes to this branch if they are finalized and ready to go live.</p> <p>On the release date for a general release, at approximately 6pm or at the time of the Vault Safety update, lr-develop is merged into both limited-release and general-release.</p>	<p>Each merge or commit to this branch triggers an auto-deployment to:</p> <p>https://safety.veevavault.help/gr/</p>

Temporary Branches

New Features

1. Create a new branch off of **lr-develop**.
Name the branch in a format similar to <release number>/<your name>/<feature>. For example, 20r2.2/emily/scheduled-follow-up-questionnaire
2. Draft your content updates in the branch you created in step 1.
3. When you've drafted the updates and want to stage them for preview, merge the updates back into **lr-develop**.
NOTE: If it's unclear whether a feature will make it into a limited release, wait before you merge with lr-develop (less risk to have to revert the merge if the feature gets pushed).

Release Notes & RIA

Limited Release Notes

1. Create a new branch off of **lr-develop**.
Name the branch in a format similar to <release number>/release-notes
2. Draft your content updates in the branch you created in step 1.
3. When you've drafted the updates and want to stage them for preview, merge the updates back into **lr-develop**.

General Release Notes (About the yyRx Release) & RIA

1. Create a new branch off of **gr-develop**.
Name the branch in a format similar to <release number>/gr-release-notes
2. Draft your content updates in the branch you created in step 1.
3. When you've drafted the updates and want to stage them for preview, merge the updates back into **gr-develop** AND **lr-develop**. Once they're ready to publish, merge the respective branches into **general-release** and **limited-release**.
4. Keep the branch you created in step 1 and use it to draft future updates. For future updates, repeat step 3.

Fixes and Enhancements (Unrelated to New Features)

For Both LR and GR

1. Branch off of **gr-develop**
Name the branch in a format similar to <release number>/<your name>/<fix>
2. Draft your content updates in the branch you created in step 1.
3. When you've drafted the updates and want to stage them for preview, merge the branch into both **gr-develop** AND **lr-develop**

For LR Only

Follow the same process as New Features, branching off of **lr-develop**.

For GR Only

This is super rare- generally, any updates applicable to GR also apply to LR.