

Creating a Release



From Registry version 0.9 forward, and for all versions of all other products, [Semantic Versioning](#) will be followed.



Unless otherwise noted, all steps apply to both Match and Registry releases.

Release Candidate (##.0-rc.#)

The steps for the Release Candidates are the same as for the Standard Releases, up to and including the tagging of the release. Do not update the public links, and only announce informally.

Note that `Config/VERSION` (for Registry) and `config/VERSION` (for Match) should not include the rc label, as this will break `upgradeVersionShell`.

Standard Release (##.0)

1. Check for [latest version of dependent libraries](#) and update if needed.
2. Review open issues in JIRA for the release. All issues should be resolved (with code committed) or moved to a future release.
3. For Registry: Update `Config/VERSION`, if not already updated, and make sure the version is defined in `UpgradeVersionShell::$versions`.
4. For Match: Update `config/VERSION`, if not already updated. Make sure the version is defined in `UpgradeVersionCommand::$versions`.
5. Check for commits to the most recent hotfix branch that *were not* released in a patch release. (This may happen if, for example, the latest patch release was canceled with the intent to skip to the next feature release instead.) If the most recent hotfix commit is not in develop, merge the hotfix branch into develop.
6. For Registry: on the develop branch, update the LABEL in the files `container/aws/buildspec/aws_buildspec.yml` and `container/aws/buildspec/aws_buildspec_manifest.yml` to the intended release number.
7. For Match: on the develop branch, update the LABEL in the file `container/aws_buildspec.yml` to the intended release number.
8. Merge develop into main (assuming you're already following both)

```
$ git switch develop
$ git pull origin develop
$ git switch main
$ git pull origin main
$ git merge develop
(Make sure everything looks OK)
$ git push origin main
```

9. Tag the release:

```
$ git tag -a 0.8.5 -m 'COManage Registry v0.8.5'
$ git push origin 0.8.5
```

10. Notify the lead of the packaging/container team of the new tag.
11. The lead of the packaging/container team [builds new container images](#).
12. Wait for notification from the packaging/container team that packages/images are ready.
13. Test the release, including upgrading QA servers and installing/deploying from scratch.
14. Update download link at [Directory Installation](#), [Registry Installation - Source](#), or [Match Installation - Source](#).
15. Update [TAP release page](#).
16. Review [Registry Data Model](#) or [Match Data Model](#) and update the status of tables as appropriate.
17. In JIRA, flag the release as "Released" with today's date.
18. Update [Release History](#).
19. Update [Home](#), including the *Download link and About This Release*.
20. Update the [COManage Product Roadmap](#).
21. If there are any unrelated bug fixes in the prior hotfix series (##-1.#), simultaneously release that as the "last maintenance release" of the previous series.
22. Announce (to `comanage-users` and `comanage-announce`). If appropriate and approved, credit additional funding sources.
23. Begin the [Synchronous Code Update Checkpoint](#) process.

Bug Fix Release (##.#)

A bug release is intended to quickly fix bugs discovered with a standard release. These releases are indicated using a revision number after the related standard release number. ~~There is not necessarily a corresponding JIRA release.~~

1. Review open issues in JIRA for the release. All issues should be resolved (with code committed) or moved to a future release.
2. Update `Config/VERSION`, if not already updated, and make sure the version is defined in `UpgradeVersionShell::$versions`.
3. Update the LABEL in the file `container/aws/buildspec/aws_buildspec.yml` to the intended release number.
4. Merge changes from the [hotfix release branch](#) into main *and* develop.
 - a. ⚠️ If main has moved on to a new minor release, do *not* merge into either branch. You may wish to cherry pick commits if appropriate.
5. Tag the release from the *hotfix* branch (though you can also use main if you merged the hotfix branch into main):

```
git tag -a 0.8.5 -m 'CManage Registry v0.8.5'
$ git push origin 0.8.5
```

6. Notify the lead of the packaging/container team of the new tag.
7. The lead of the packaging/container team [builds new container images](#).
8. Wait for notification from the packaging/container team that packages/images are ready.
9. Test the release, including upgrading QA servers and installing/deploying from scratch.
10. Update download link at [Directory Installation](#), [Registry Installation - Source](#), or [Match Installation - Source](#).
11. Update [TAP release page](#).
12. If appropriate, flag the release in JIRA as "Released" with today's date.
13. Create the next bug fix release in JIRA ([#. #. #+1](#), unless there's a simultaneous feature release, in which case [#. #+1.1](#)).
14. Update [Release History](#).
15. Update [Home](#), *including the Download link and About This Release*.
16. Update the [CManage Product Roadmap](#) (if appropriate).
17. Announce (to [comanage-users](#) and [comanage-announce](#)). If appropriate and approved, credit additional funding sources.