

Cutting a Release

This page describes the process for creating a new CORD release. At a high-level, the process has two steps:

1. For a major release, create a support branch on all the CORD repositories (e.g., *cord-3.0*)
2. Cut an individual release version on the support branch (e.g., *3.0.1*, *3.0.0-rc1*).

Create a Support Branch

The first step to cutting a release is to create a support branch (e.g., *cord-3.0*) on all the CORD repositories. This is done by an automated process, which can be triggered as follows:

1. Create the support branch on the manifest repository using the [Gerrit UI](#). (Note: you need to be a Gerrit admin to do this.)
 - a. *Branch Name*: specify the name of your support branch
 - b. *Initial Revision*: *master*
2. Check out the new manifest branch.
 - a. Edit *default.xml*; where it says *revision="master"*, change *master* to the name of your support branch.
 - b. Make changes to the *Jenkinsfile* and *default.xml* to remove repositories not included in the release. For example, the ONOS apps have their own release schedules. (See **Notes on 4.0 Release** below.)
 - c. Check in the modified *Jenkinsfile* and *default.xml* to the branch.
3. Edit the configuration of the [release-build job in Jenkins](#). (Note: you need to be a Jenkins admin to do this.)
 - a. In the **Branch Sources** section, add the new branch to the list of branches in the *Filter by name (with wildcards)* box.
4. Trigger the *release-build* Jenkins job for the new support branch.

After the *release-build* job is triggered, it should create the support branch on all repositories in the manifest. The support branch is created from *HEAD* of each repository's *master* branch. The support branch can then be checked out using the *-b* argument to the *repo init* command, for example:

```
$ repo init -u https://gerrit.opencord.org/manifest -b cord-3.0
```

Cut a Release

A release uses the manifest to specify a specific commit for each repository on the support branch. For example, if *3.0.0-rc1* and *3.0.1* are releases on the *cord-3.0* support branch, each would have its own manifest specifying a different set of commits.

To cut a release:

1. Run a job that creates */var/lib/jenkins/temp/manifest-<branch>.xml* on the Jenkins host. Physical POD builds will create this file.
2. Trigger the *release-build* Jenkins job for your support branch. This will send an email to the approvers listed in the *Jenkinsfile* in the support branch's *manifest* repository. Currently the approvers are Ali Al-Shabibi, Andy Bavier, and Larry Peterson.
3. An approver clicks the link in the email and enters the tag for the release (e.g., *3.0.0-rc1*)

At this point the *release-build* job will create a branch in the *manifest* repository for the release. The release manifest will contain commits for each repository corresponding to *HEAD* of the repository's support branch.

Additional Manual Steps

The steps in this section should be automated, but for now need to be performed manually:

- Change *build/scripts/cord-bootstrap.sh* so that the default branch is the release branch: <https://gerrit.opencord.org/#/c/6148/>
- Update the version in *xos_gui/package.json*: <https://gerrit.opencord.org/#/c/6152/>
- Update the *defaultbranch* in the *pod-configs/.gitreview* file
- Create the directory */var/www/guide/<version>* on the wiki machine
 - `$ sudo chown jenkins.jenkins /var/www/guide/<version>`
 - `$ sudo chmod 755 /var/www/guide/<version>`
- Run a documentation-publish job with parameter *branch=<version>*
- Modify the apache virtual server file to create all aliases used to reach the new documentation

Notes on 4.0 Release

The 4.0 release featured downloading published versions of the ONOS apps rather than building them from source. Since the ONOS apps are now handled like external packages, it was decided to [remove them from the manifest](#). The ONOS app repositories were also added to the `IGNORE_LIST` in the *Jenkinsfile* so that the *cord-4.0* branch would not be created on them.