

Release Coordinating

This page gives some notes on how to go about coordinating a DSpace release.

- [The Release Team](#)
- [DSpace development](#)
- [What is a Release Team expected to do or not do?](#)
- [Example Tasks](#)
- [Things to Do at the Start of a Release Cycle](#)
- [Tips](#)

The Release Team

The release team is selected from volunteers from the Committer Group. Each release team selects a "team lead", but all members of the release team are expected to help with release coordination duties.

DSpace development

DSpace is a community driven open source project. This means that the requirements for new features are driven by the community, and the actual code developments are contributed by the community. Therefore much of the role of the release team is based around working closely with the community to ensure that requirements are documented, and code is contributed.

What is a Release Team expected to do or not do?

Generally speaking, the release team helps coordinate the next release of DSpace. This may include (not an exhaustive list):

- Help keep code development around a release organized ([DSpace JIRA](#) is a great resource for keeping organized)
 - e.g. Ensure each new feature or bug fix has a developer assigned to it
 - e.g. If a developer needs help/feedback, help locate that help/feedback from the Committers or the community
- Help communicate information about the Release to DSpace Community
- Report on release status updates during DSpace [Developer Meetings](#)
- Ask for additional help/support when a release requires it

The Release Team is **NOT** expected to:

- Make all decisions themselves (*FALSE*: The Release Team should use the Committers Group to help make decisions. But, the Release Team can veto an idea, or ask for a re-vote, if it doesn't seem plausible for the release.)
- Be an expert on every feature in the release (*FALSE*: The Release Team should be aware of who the feature experts are, and be able to delegate questions to them as necessary)
- Do everything themselves (*FALSE*: The Release Team should delegate tasks to the Committers Group or others as necessary. In addition, DuraSpace provides additional support available on request.)

Example Tasks

Listed below are example tasks that a Release Team may need to undertake (or delegate to someone else to handle). Some of the tasks may include:

- Determine, along with the development community, whether a major or minor release is required. Minor releases include bug fixes and minor, non-db-schema-altering changes; mostly minor releases can be installed by replacing the DSpace code. Major releases include major functional enhancements, configuration changes and db schema alterations.
- Determine, along with the committer group, the requirements for the next release. If it is a minor release, this may be as simple as fixing any bugs that have been reported in JIRA. For a major release, you will likely want to consult the community via surveys, emails, or JIRA voting to determine what features should be included. Final determination is often done in conjunction with the rest of the committers.
- Work with the development community and committer group to find volunteers to develop new features.
- Work with the committer group to find volunteers to claim PRs, code review / test them, and merge into the codebase
- Determine a feature freeze date. In a normal release cycle, there will hopefully be a constant influx of patches waiting to be rolled into the relevant release. The feature freeze date, therefore, need not be a long way in advance. It's useful if the date is a Friday, as this allows trailing time-zones to catch up before work in earnest can start on code integration on the Monday.
- Announce to dspace-devel, dspace-tech and dspace-general lists the date of the feature freeze for the coming release, requesting patches and bug fixes.
- The job of the release team is not to test all the patches and fix all the bugs, but to help others in doing so. Advise the list of any important events in development and liaise with patch authors to ensure smooth development.

- Once the freeze date has occurred, hold a [Testathon](#). Encourage the community to take part in the testathon to make the release less buggy. Ask for volunteers to run test instances of the new version to allow people who do not have access to a server to be able to test it.
- Following the testathon, the release may be ready, or it may have significant bugs, in which case bug fixing is required, and another release candidate. Work with the community to find volunteers to fix bugs.
- The release team is responsible for building the release package and uploading it to SourceForge. See [Release Procedure](#) for information on how to do this. The release team should also email dspace-devel, dspace-tech and dspace-general to announce the release.

Things to Do at the *Start* of a Release Cycle

- Review prerequisite software (DBMS, Java, etc.) version requirements with the developer community, and adjust them as required. Keep the documentation of this up-to-date.
- Review dependency (e.g. log4j) version requirements and adjust as required. Tools such as `mvn dependency:analyze` can help. Also remember non-Java things such as Javascript libraries. This could be a big job, so ask for volunteers to help.

In general, try to establish a stable up-to-date foundation for development and integration early on.

Tips

- It is never too early to start thinking about the release.
- It is never too early to start talking about the release. People are busy and need reminding.
- Try and create a provisional list of major features as soon as possible, even though it will undoubtedly change throughout the year.
- Engage with the developers and gently push for development and commit timescales. Just talking may help speed the process.
- A commit date for a major feature of one week before the feature freeze is too late, it will inevitably result in a postponement of the feature freeze date. All code needs review and probably rework, developers need to recognise that and plan accordingly.
- Don't be afraid to badger people. It goes with the role and people expect it.
- Don't be afraid to ask for help.