

How to Contribute to DSpace

Are you a new developer looking for how to get started with DSpace? Take a look at our [New Developers Hub](#)

Contributing to DSpace Software

- [How to "Hack" the DSpace Community](#)
- [Ways To Contribute and Participate](#)
- [Platforms for Contribution and Participation](#)
- [How To Contribute Ideas or Suggest New Features](#)
 - [1. Make Your Request Known to DSpace Developers](#)
 - [2. Advertise your Request to Others & Help Us Find a Volunteer Developer or Two](#)
 - [3. Respond to the Formal Review of your Request \(as necessary\)](#)
 - [4. Keep in Touch about the Request](#)
 - [5. Once your Request is Accepted into DSpace](#)
- [How To Contribute Code or Development Time](#)
 - [Where to Obtain the Source Code](#)
 - [Our Development Policies / Mantra](#)
 - [Looking for a Project to Work on?](#)
 - [Submitting Code Changes](#)
 - [Copyright and Licensing of Code Contributions](#)

This page provides links to resources/notes on how you can contribute to DSpace, whether you are a repository manager or a developer.

How to "Hack" the DSpace Community

This presentation was given at Open Repositories 2014 in Helsinki, Finland. It provides some tips on how and why you may wish to contribute to DSpace:

<https://www.slideshare.net/todonohue/hack-d-spacecommunityor14>

Ways To Contribute and Participate

You do not have to just contribute code! There are other ways you can contribute:

- **Communicate** - Use the [Mailing Lists](#), this Wiki and the [Slack](#) to communicate with the community
- **Congregate** - Attend user groups, conferences, library events, developer meetings - and any other venue where DSpace users meet to share information and ideas. If you are a developer (or just interested in developer discussions), join the weekly DSpace [Developer Meetings](#). If you are a repository manager, you may wish to volunteer for the [DSpace Community Advisory Team](#) or attend their meetings / online discussions.
- **Test** - Download and try out beta releases; provide bug reports, experiences, feedback. Our [DSpace Demo Server](#) provides a place to test the latest and greatest version of DSpace. If you find a bug, report it via our [Issue Tracker](#)
- **Develop** - Contribute bug fixes, new features, developer cycles. Contributing code is far easier than you might think! See the [How To Contribute Code](#) section below and the [Code Contribution Guidelines](#) for more details.
- **Translate** - Translate the DSpace user interface into your language. See [DSpace 7 Translation - Internationalization \(i18n\) - Localization \(l10n\)](#) for more details.
- **Prototype** - The best way to gain support for an idea is to build and share prototype code. If you'd like to share existing prototypes, see the [Code Contribution Guidelines](#) for more details.
- **Deploy** - Share your experiences in deploying DSpaces in different organizations and situations, at large and small scales
- **Support** - Become active members on the [mailing lists](#), answer others' queries and help solve their technical problems
- **Experiment** - Take the system for a spin, try it out with different types of content and scenarios; tell everyone what you find. Again, the [DSpace Demo Server](#) provides a place to experiment with the latest and greatest version of DSpace. (If you are running a larger, scalability test experiment on the Demo Server, please let us know by emailing the 'dspace-devel' [mailing list](#))
- **Donate content and metadata** - To test and experiment with DSpace, free test collections unencumbered by restrictive usage rights are needed. Contact us via the [mailing lists](#) if you have content to donate for testing.
- **Request new features / Share ideas** - Is there something that you really need out of DSpace or isn't working right? Request new features / improvements or report bugs via our [Issue Tracker](#). You can also vote on existing features, or add your own comments/suggestions. Both of these can help developers decide which issues are the most important to the community. See the below section on [How To Contribute Ideas or Suggest New Features](#) for more details.
- **Help Improve Documentation** - Our DSpace Documentation is now managed directly via a new section of our Wiki: [DSpace Documentation](#). Although normal Wiki users cannot **edit** that area of the Wiki, you can always add comments for additions/changes/suggestions. If you are interested in contributing more formally, volunteer to help via one of the [mailing lists](#), and we can add you to our Documentation Team and provide you with access rights to edit/improve the Documentation directly.
- Let us know if there's a way we can ease the process of contributing to DSpace
- Don't be shy! Contributions don't have to be 100% polished or perfect; no one will think any the less of you. "Share early, share often" is a well-known open source mantra. The sooner you contribute something, the sooner others can help with the polishing, and you no longer have to maintain the customization against the evolving core DSpace platform, since it will be part of the platform!

Platforms for Contribution and Participation

- This Wiki - Help making this Wiki a useful, concise and up-to-date information source by
 - supplying content
 - correcting content

- deleting obsolete content
- restructuring or otherwise improving content
- [The Mailing Lists](#)- Take an active part in the discussion on DSpace.
 - share your thoughts about DSpace
 - ask questions
 - provide feedback
- [Official DSpace Documentation](#)
 - add comments to existing docs, to help enhance the existing content
 - volunteer to help improve content or otherwise enhance it
- [DSpace GitHub](#) - GitHub allows for collaborative & social code development! You can create your own GitHub account (it's free) to get started, and then "fork" our repository to start creating your own modifications (which can be submitted back to us via a GitHub "Pull Request"). For much more information see [Development with Git](#).
- [Issue Tracker](#)
 - bug reports
 - feature requests
 - add your own comments
 - "watch" existing issues (you will receive an email any time a new comment is added or the issue status is updated)
 - volunteer to work on a feature/issue (just add a comment to that issue)
- [Slack](#) - an informal way to discuss ideas and ask questions. You can also help others who need some immediate help.

How To Contribute Ideas or Suggest New Features

We always welcome new ideas or suggestions for new features. However, there are a few things to keep in mind which may increase the chances of your feature making it into DSpace!

1. Make Your Request Known to DSpace Developers

You should submit your idea or feature request to our [Issue Tracking System \(GitHub\)](#). However, before going through the process of submitting your ideas, it's always best to search the Issue Tracker to see if others have already requested this feature. If someone else has requested this feature, you can add your ideas as "comments", or "vote" for that feature to be added/implemented.

If this feature hasn't been requested by anyone else, submit a request! (Don't worry, if it ends up being a duplicate request, we'll let you know). Your request will enter the "queue", where it will get a thorough review.

A few things to keep in mind when providing a request:

1. Always provide us with as many details as you can. A paragraph description is good, but a few paragraphs with some sample use cases is even better. A single sentence description is often very hard to work with, and we may need to ask you for more information before we understand what you are asking for.
2. If you have use cases or local needs, please describe them to us. Use cases really help developers understand why this feature is important. In addition, use cases may help us locate other institutions with similar needs (who may be willing to help us develop this feature).
3. **Expect that we probably will need to ask you a few questions.** Even with detailed descriptions/use cases, chances are we will need to follow up with you later for a few more details, or to make sure we understand exactly what you are asking for. So, please be willing to respond to questions or requests for additional details. Anytime someone comments on your newly created issue, you will receive an email from the Issue Tracker system.

2. Advertise your Request to Others & Help Us Find a Volunteer Developer or Two

All of our DSpace Developers/Committers are volunteers. Let's repeat that: ***All of our DSpace Developers/Committers are volunteers.*** This means the core DSpace Development Team (i.e. Committers) don't always have control over how much time they can spend on developing new features for DSpace. In many cases, the Committers can only work on new features which are *of interest to their local institution/university*.

Therefore, even if the Committers agree that a new feature seems worthwhile, oftentimes, we need to first find an **institution** that is willing to provide developer time towards that feature.

You may be able to help us expedite this process! Here's what you can do to help:

- If you have a local developer who has time to work on this feature, let us know when you submit the issue (or add a comment later). If a local developer can already work on the issue/feature request, we may just need to approve the code (see [Code Contribution Guidelines](#), for details on our DSpace Code Approval/Acceptance processes).
- If you know of other institutions with similar needs, tell them to "vote" for your issue request in the [Issue Tracker](#), or add their own use cases /support as comments. Also, if any of them have a developer with time to develop the feature, let us know!
- If you are not sure if other institutions may have this need, you can promote your issue by sending an email to 'dspace-community' or 'dspace-tech' [mailing lists](#), asking for others' feedback. Hopefully, others can add comments/suggestions or even point us in the direction of an interested volunteer developer.
- Even if we cannot find an interested developer in the community, the DSpace Committers will review your request and see if one or more Committers have time to devote towards the work. In some cases, a Committer may be able to convince their institution of the importance of the new feature (again, sample use cases are helpful to convince institutions of a feature's importance).

3. Respond to the Formal Review of your Request (as necessary)

Each Feature Request is guaranteed to get a formal review by at least one of two groups (possibly both):

1. The DSpace Committers - They review every feature request or bug report that comes into the system, often in weekly [Developer Meetings](#). Note: Because of occasional backlogs, it is sometimes possible there may be a delay of several weeks/months before your request will get a formal review by the Committers. However, it is worth noting that your request will be immediately emailed to the 'dspace-tickets' [mailing list](#), where individual developers may provide immediate feedback before the formal review takes place. So, you may not need to wait that long for immediate response, or even for a developer to volunteer to work on that feature.
 - For more information on which issues the Committers are planning to review in their next meeting, see the currently posted agenda on the [Developer Meetings](#) page. Please Note: The Committers always review issues in the order they are received. However, as noted above, all new feature requests are emailed to 'dspace-devel', so if there is immediate interest amongst developers, you may find you will receive more immediate feedback and/or questions.
2. The [DSpace Community Advisory Team](#) - They review and request additional feedback about any new feature request. This is a team of Repository Managers (or similar) who provide additional feedback to the Committers on new features.

After a formal review, a comment will be added to your request (it will also generate an email to you). This comment will detail the results of the initial review, along with any questions that came up. If you have time, please respond to these questions, or encourage others to do so. These questions are often asked to help us determine more about the request, so that we can ensure we have a common understanding.

4. Keep in Touch about the Request

Let us know if you need updates on the Feature Request's status. Just add a comment to your issue, requesting the latest status, and we'll try to get back to you as soon as we can.

There are many different reasons why a Feature Request may not have had any recent activity:

1. We may have a backlog of requests, and just haven't gotten to a formal review yet.
2. We may need to find a developer (or committer) who has time to develop this feature. In these cases, if we can locate other institutions who may be interested, that can often help in the search for a volunteer developer.
3. We may be waiting for the answer to one or more questions posed in earlier comments. If we need more clarification, we can let you know.
4. We may be currently performing a "Code Review" on any submitted code, to ensure it is safe & stable enough to release in DSpace. For more information on our DSpace Code Approval Processes, see [Code Contribution Guidelines](#)
5. It's also possible that there are one or more developer(s) actively working on the feature, but that the work is not yet in a "completed" state.

5. Once your Request is Accepted into DSpace

If your request is formally accepted into DSpace, you'll receive an email as soon as we "Close" or "Resolve" the request in our Issue Tracker. At that point in time, the Issue Tracker will also be updated to state which version of DSpace this new feature will be released in.

Once that version of DSpace is released, your name (and a link back to your initial feature request) will appear in our [Version History](#) section of our DSpace Documentation. You will also be added to our list of all known [DSpaceContributors](#). This is our way of ensuring you receive recognition for your contributions to DSpace!

How To Contribute Code or Development Time

Where to Obtain the Source Code

If you plan to develop new features for DSpace, we recommend forking and cloning our Source Code via Github:

- DSpace GitHub repository: <https://github.com/DSpace/DSpace>
- DSpace 7 UI GitHub repository: <https://github.com/DSpace/dspace-angular>

More information on using DSpace + GitHub is at: [Development with Git](#)

Development Hints/Tips



More hints/tips on developing with DSpace are available in the following locations:

- [Developer Guidelines and Tools](#) - includes hints / tips on using many common IDEs (IDEA, NetBeans, Eclipse) with DSpace
- [Code Contribution Guidelines](#) - includes our guidelines for submitting patches or new features, and our code approval processes

Our Development Policies / Mantra

For more information on our DSpace Code Approval/Acceptance process (i.e. how to get your code accepted in DSpace), please see our [Code Contribution Guidelines](#).

The overriding mantra is **share early, share often**. Here are a few things to consider:

- **Please be sure to share your plans with the DSpace community on the 'dspace-devel' list (or via one of the weekly [Developer Meetings](#)) before embarking on any sizable development effort.** This will ensure you achieve your goals in a way that is consistent with the DSpace architecture and plans of the rest of the community. This will minimize the chances of a scenario where you have invested a large amount of time and effort into a body of code that does not fit in with the DSpace architecture or the consensus of the community, meaning that you need to spend further time refactoring your code or worse, 'forking' the code.
- **Develop incrementally;** try and implement and contribute a basic form of your feature as soon as possible, rather than aiming to implement a complete and 'polished' solution. This will help ensure you're on the right track with regards to the rest of the DSpace community and platform. The sooner your code is part of the core code base, the less time you will have to spend 'chasing' the main code base, i.e. keeping your changes up-to-date with that core code base.

- **Obtain the DSpace code using GitHub.** This will make code management *much* easier. It's very simple to do; see [Developer Guidelines and Tools](#) and [Development with Git](#).
- Read [Code Contribution Guidelines](#) to ensure you are following DSpace conventions. This page also gives you a sense of the DSpace Code Approval processes.

Looking for a Project to Work on?

Do you have a developer (or two) with some extra time? Are you looking for ways that you can help the community and improve your local DSpace?

Please take a look at our issues lists

Please, take a look at our current listing of "good first issue" tickets. Any help you can provide would be much appreciated!

- UI "good first issue" tickets: <https://github.com/DSpace/dspace-angular/labels/good%20first%20issue>
- Backend "good first issue" tickets: <https://github.com/DSpace/dspace/labels/good%20first%20issue>

(If none are currently listed, as on Slack or via email and we'll find a smaller ticket for you to start with)

But, before you get started, please make sure to do the following:

1. Add a comment to the Feature/Improvement you plan to work on, letting us know you will work on it.
2. If you'd like more input on the feature/improvement, or potential requirements, post your questions and/or plans as a comment as well.
3. Make sure your developer is following our [Code Contribution Guidelines](#). If you have questions about any guidelines, or want some early feedback/suggestions from developers, please get in touch with us on the '[dspace-devel](#)' listserv. We'd be glad to help make suggestions on ways in which to implement the new feature, and the earlier you get in touch, the earlier we can give you feedback on whether there's anything you may need to change before we can accept it as part of out-of-the-box DSpace. See also the Code Review Process on the [Code Contribution Guidelines](#) page.
4. If you run into any "gray areas", ask questions! If it's a development issue, contact the Developers via the '[dspace-devel](#)' listserv. If it's a policy issue or requires feedback from Repository Managers, get in touch with the [DSpace Community Advisory Team](#), as they can help you query the community for feedback and/or provide you with their immediate opinions

Most of all: Thanks! The more individuals/institutions can give back to DSpace, the better the software is for everyone!

Submitting Code Changes

See [Code Contribution Guidelines](#) for guidelines that all submissions must adhere to. That page also describes the general process for how a patch/contribution gets accepted into DSpace. The mechanics of creating a patch file are described in [Developer Guidelines and Tools](#).

Copyright and Licensing of Code Contributions

In the words of the PostgreSQL Global Development Group, which also uses the BSD license, "The simplest explanation of the licensing terms is that you can do whatever you want with the product and source code as long as you don't claim you wrote it or sue us." The BSD License under which DSpace is made available does not require you to make your changes public or open-source. It does allow for proprietary commercial use, and for DSpace-derived creations to be incorporated into proprietary commercial products. Works based on DSpace may even be released under a proprietary license (but still must maintain the license requirements).

You are encouraged, but not obligated, to share your contributions with the DSpace community. If you choose to do so, you will need to sign over copyright and intellectual property rights of your code to DuraSpace, to be distributed via the BSD license. DuraSpace is a 501c(3) non-profit established to be the legal guardian of the code and to remain mission centric on providing free and open source software for management and archiving of digital works. Also, your code cannot rely on any non-BSD compatibly licensed code.

The BSD license means there is no advantage to be gained by your university (or anyone) retaining copyright, and that by having different copyright holders of different sections of the code, we will be rendered inflexible regarding copyright and licensing in the future, we do ask that you transfer copyright of your modifications to DuraSpace.

You will receive full acknowledgment for contributing the code; so we do encourage you to incorporate your enhancements to DSpace's functionality for everyone to benefit. You will also see benefits since you will neither have to re-incorporate the changes with new versions of DSpace, nor maintain this code solely yourself!

If your code contribution uses third-party products/tools, you should also double-check that they use a compatible open source license. Compatible licenses are listed at: [Licensing of Contributions](#) section of the [Code Contribution Guidelines](#) page.