RoadMap

DSpace 7 is available now!
See DSpace 7 Press Release and Release Notes for more information.

Ready to get started? Download DSpace 7 or Try out DSpace 7 (via our sandbox site or Docker quick install).
Roadmap updates in progress

This Roadmap is currently being updated based on the 7.x releases. Until this warning is removed, the below information is likely outdated.
The Roadmap for the 7.x releases can be found at What features are coming in a later 7.x release?
Specific Release Information

If you are looking for information regarding specific releases of DSpace, especially past releases, also see our Releases page.

- Background
- How to get involved
- Short-Term (Priority 1) - Available in DSpace 7.x
- Medium Term (Priority 2) - Not yet available in 7.x
- Longer Term Features / Priorities
- Integrations

Background

This Technical RoadMap is based on the DSpace 2015-18 Strategic Plan - Technology and the Use Case Analysis produced by the DSpace RoadMap Working Group in April/May of 2015. The DSpace RoadMap Working Group consists of the members of DCAT, Committers and Steering Group: Tim Donohue (Lead, DuraSpace), Stuart Lewis (Edinburgh), Bram Luyten (@mire), Jonathan Markow (DuraSpace), Michele Mennielli (CINECA), Richard Rodgers (MIT), Ryan Steans (Texas Digital Library), Maureen Walsh (Ohio State).

While it is a living document, it was initially presented at both OR15 (week of June 8), and OAi9 (week of June 15).

- A video screencast of this 2015 talk is available at: http://youtu.be/qlPAeqCptxw
- Slides are also available at: http://www.slideshare.net/tdonohue/dspace-technology-roadmap-201516

Since 2015, ongoing Strategic Planning activities have taken place in DSpace Steering & Leadership Group Meetings.

How to get involved

- If you’d like to help us achieve specific goals in this RoadMap, please get in touch via our Mailing Lists or Slack.
- DSpace is primarily supported by community code contributions. If you’d like to contribute code, please see our Code Contribution Guidelines.
- If you are interested in discussing a specific topic, we do have DSpace Interest Groups and DSpace Working Groups available.
- Other general ways of contributing to DSpace (both technical and non-technical) can be found at How to Contribute to DSpace

Short-Term (Priority 1) - Available in DSpace 7.x

The DSpace 7 is available now, see Release Notes. Latest status on other 7.x releases can be found at DSpace Release 7.0 Status. Features that have already been included in 7.x are listed below

<table>
<thead>
<tr>
<th>Priority 1 Features</th>
<th>Design / Notes</th>
<th>Related Technical Strategic Goal(s)</th>
<th>Core?</th>
<th>Complexity</th>
<th>Use Cases</th>
<th>Included in 7.x?</th>
</tr>
</thead>
</table>
| Single User Interface. | DSpace currently maintains two user interfaces in parallel (JSPUI and XMLUI). To replace these two user interfaces, we are building a new, single, out-of-the-box user interface on Angular.io.  
- Introducing the New DSpace User Interface (at OR16)  
- DSpace UI Prototype Challenge  
- Design - Single UI Project | Goal 2: Lean and flexible | x | High | For reference:  
- End User Use Cases  
- Admin UI Use Cases  
Also see: DSpace 7 UI Project Plan Language Summary | ✓ |
| Standards-based REST API | DSpace’s current REST API, while functional, is limited in features and does not follow current best practices for RESTful APIs.  
To support the new, single user interface (on Angular.io), we are building / designing a new REST API that follows modern best practices such as HATEOAS, API/ES, and using the HAL response format.  
The new REST API is being built using Spring technologies (Boot, MVC, and HATEOAS). | Goal 3: Can be “extended” and Goal 4: Integration with external services | x | High | New REST Contract (work in progress):  
- https://github.com/DSpace /Rest7Contract | ✓ |
<table>
<thead>
<tr>
<th><strong>Goal</strong></th>
<th><strong>Details</strong></th>
<th><strong>Priority</strong></th>
<th><strong>Implementation Status</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Goal 1:</strong> Fundamentals of IR</td>
<td>DS-3041 - Getting issue details... STATUS</td>
<td>Medium</td>
<td>DS-3455 - Getting issue details... STATUS</td>
</tr>
<tr>
<td><strong>Goal 2:</strong> Lean and flexible</td>
<td>Not all use cases met</td>
<td>Medium</td>
<td>(Not all use cases met)</td>
</tr>
<tr>
<td><strong>Goal 3:</strong> Integrate with other systems</td>
<td>Admin UI - Generate Handle before committing item</td>
<td>Medium</td>
<td>(Not all use cases met)</td>
</tr>
<tr>
<td><strong>Goal 4:</strong> Integrations - Integrations that significantly lower the effort to fill DSpace with content, both from in-house systems and third-party content</td>
<td>Admin UI - Update with embargo functionality</td>
<td>Medium</td>
<td>(Not all use cases met)</td>
</tr>
</tbody>
</table>

**Single Approval Workflow system.**

DSpace currently has two approval workflow systems:

1. Basic/Traditional Approval Workflows. These are enabled by default and provide up to three approval steps: Approve/Reject, Approve/Reject/Edit, or Edit. These are described in more detail in the Functional Overview/Workflows section.
2. Configurable Workflow (KXMLUI only, requires migrating all Basic/Traditional workflows).

We should consolidate on a single Approval Workflow system, likely the Configurable Workflow, with sane defaults.

**Single built-in Statistical Engine (SOLR Statistics)**

DSpace currently has three built-in statistical engines, one based on Apache Solr (default), another based on Elasticsearch (optional), and a third Legacy statistics (which parses logs). DSpace should only provide one out-of-the-box, built-in statistical engine (Solr Statistics). If and all others should be removed (and as necessary, their features merged into one).

- Remove Elasticsearch Usage Statistics, see DS-3455 (Completed)
- Disable/Remove Legacy (log based) Statistics, see DS-3456

**Lower the effort to deposit content (via integrations).**

(Some use cases met by new deposit UI, featuring “external sources” of metadata. However, NOT all use cases implemented yet.)

The DSpace deposit process should integrate more closely with external data sources, in order to automatically populate (or suggest) data on deposit. We should also investigate whether some integrations may allow opportunities for Administrator’s to autopopulate DSpace from trusted, external content sources.

- **Design:** Lower effort to deposit

**Easy and Intuitive Deposit Interface**

(See new deposit UI)

DSpace’s deposit interface could use enhancement for a better overall user experience with regards to usability and accessibility.

- **Dependent on/related to “Single User Interface”**
- **Primary use case description:** Easy and Intuitive Deposit Interface
- **May need further definition around what sorts of changes will make deposit easier?**

**Relationships between Objects**

(Related between items is new supported via Configurable Entities, including new Author Entity objects related to their publications. Other object relationship use cases not yet met.)

DSpace should support defining relationships between objects (especially at the Item and Bitstream level). Some object relationships are definable via metadata, but should be respected by the DSpace UI. But, in other situations, it may also necessitate the creation of new types of Objects (e.g. Author objects as Authors of an Item rather than simply textual metadata).

- **Primary use case description:** Structure - Relationships between objects
- **This feature should also investigate whether there are opportunities to better integrate or utilize the new RDF-based interface to describe/define object relationships within DSpace.**

**Goal 5:** Fundamentals of IR

---

**Medium Term (Priority 2) - Not yet available in 7.x**
As "priority 2" features, these features are not yet available in DSpace 7.x, **but are still seen as a priority in an upcoming release**. Nonetheless, we'd encourage community members to volunteer to help achieve any of these features. If there is community interest in moving one or more of these features forward, that feature may be moved up to "priority 1" for the next release. These features are ordered in terms of importance, but are all considered to be lower importance than the "priority 1" features listed above.

<table>
<thead>
<tr>
<th>Priority 2 Features</th>
<th>Design / Notes</th>
<th>Related Technical Strategic Goal(s)</th>
<th>Core?</th>
<th>Complexity</th>
<th>Use Cases</th>
<th>Work in Progress?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configurations in Admin User Interface</td>
<td>DSpace should support the modification of most configurations/settings from the Administrative User Interface, instead of requiring such configurations be tweaked from command line.</td>
<td>Goal 5: Low cost, &quot;just works&quot;</td>
<td>x</td>
<td>High</td>
<td>- Structure - Associate Separate Properties with Each DSpace Community - Admin UI - Changing / Editing OAI crosswalks - Admin UI - Configurations in the admin UI - Admin UI - Configure and manage browse indexes - Admin UI - Rebuild the discovery index - Admin UI - Alter messages easier - Integration - Integration with external authentication / authorization system - Admin UI - Manage input forms - Structure - Check Bitstream names against allowed file name pattern - Admin UI - Apply an input-form to the collection through the GUI - Admin UI - DSpace should accommodate basic index normalization - Admin UI - System Alerts via Admin UI - Structure - Format checking of data entry in input forms - Integrations - Authentication through Multiple Mechanisms</td>
<td></td>
</tr>
</tbody>
</table>
### Longer Term Features / Priorities

*While these features are unscheduled at this time, we still encourage volunteers to begin to analyze or tackle them. If there is interest in moving one or more of these features forward in the nearterm, we can immediately schedule it for an upcoming release. These features are unordered, but are all considered priorities for DSpace moving forward.*

<table>
<thead>
<tr>
<th>Features</th>
<th>Design / Notes</th>
<th>Related Technical Strategic Goal(s)</th>
<th>Core?</th>
<th>Complexity</th>
<th>Use Cases</th>
</tr>
</thead>
<tbody>
<tr>
<td>Non-functional: Module Framework and Registry</td>
<td>DSpace needs a clear definition of what constitutes a “DSpace module”, so that third-parties can create, maintain and distribute their own “modules” as add-ons to DSpace, and distribute them via a public “registry”.</td>
<td>Goal 3: Can be “extended”</td>
<td>x</td>
<td>High</td>
<td></td>
</tr>
<tr>
<td>Hierarchical Metadata Support</td>
<td>DSpace should support hierarchical metadata formats (e.g. MODS)</td>
<td>Goal 1: Fundamentals of IR</td>
<td>x</td>
<td>High</td>
<td></td>
</tr>
<tr>
<td>Simplify Community and Collection Hierarchy</td>
<td>DSpace’s required hierarchy of Communities and Collections should be replaced with a more flexible module of supporting Collections &amp; Sub-Collections.</td>
<td>Goal 2: Lean and flexible</td>
<td>x</td>
<td>High</td>
<td></td>
</tr>
<tr>
<td>Enhance Basic Statistical Reporting</td>
<td>DSpace’s existing Statistical Reporting user interfaces are rudimentary and do not provide enough useful views/reports/exports of the underlying captured data. These should be enhanced based on modern use cases/needs.</td>
<td>Goal 1: Fundamentals of IR</td>
<td>x</td>
<td>Medium</td>
<td></td>
</tr>
<tr>
<td>Enhance Basic Preservation</td>
<td>DSpace’s existing basic preservation functions (format identification, checksum verification, etc) could use enhancement based on modern tools and services.</td>
<td>Goal 1: Fundamentals of IR</td>
<td>x</td>
<td>Medium</td>
<td></td>
</tr>
</tbody>
</table>
| Non-functional: Easier Installation | While DSpace's installation process continues to improve, it is still highly manual in nature. We should investigate ways to provide an "easy installer" which could potentially even auto-install and configure prerequisites such as Tomcat, Postgres, etc.  
- Dependent on some Priority 1 features (e.g. Single UI, Configs in Admin UI) | Goal 5: Low cost, "just works" | x |  
- There was an older Installer Prototype which attempted some of this, but was not successful. It may be worth revisiting that prior work/experience, if just for use cases.  
- Ability to perform an installation from a CD/DVD or offline is also of importance to institutions where network speeds are lower (mentioned at OAI9 in Geneva in 2015)  
- Vision: Installing DSpace could mean simply installing its UI. Once the UI is installed, you would be prompted to optionally install plugin(s) which enable OAI, SWORD, REST, etc. |
| --- | --- | --- | --- | --- |
| Non-functional: Easier Upgrade Process | While DSpace's upgrade process continues to improve, it is still complex with regards to local customizations or theme changes. We should investigate of simplifying the upgrade process, by alleviate the manual process of merging in local changes/tweaks where possible.  
- Dependent on some Priority 1 features (e.g. Single UI, Configs in Admin UI)  
- Could be helped by "Theme management in Admin UI" | Goal 5: Low cost, "just works" | x |  
- Goal 3: Can be "extended"  
- Stats/Metrics - Advanced Statistics  
- Others under "uc-stats-metrics" label |
| Advanced Usage Statistics | DSpace currently only provides basic statistical reporting functionality. We should investigate ways to provide enhanced, highly detailed reports for users who need them. Likely, this will be implemented as an optional module available via a module registry.  
- Likely an optional module or external integration, therefore it may be dependent on a "Module Framework" being developed  
- Ranked very highly in 2013-14 Vision Survey | Goal 3: Can be "extended" | High |  
- Admin UI - Apply an existing theme to a collection or community through the GUI  
- Admin UI - Theme templates  
- Admin UI - Theme management  
- Admin UI - Alter messages easier  
- End User - Visual collection browsing |
| Theme management in Admin UI | While DSpace currently supports some basic theming in the XMLUI interface, management of such themes is performed via command line configurations and tweaks. We should find a way to make Themes selectable and tweakable via the Admin UI, like other modern web facing systems.  
- Dependent on "Single User Interface"  
- Could be highly beneficial to "Easier Upgrade Process" in that theme management is a huge pain point of the existing upgrade process  
- Ranked lower in 2013-14 Vision Survey | Goal 5: Low cost, "just works" | High |  
- Admin UI - Apply an existing theme to a collection or community through the GUI  
- Admin UI - Theme templates  
- Admin UI - Theme management  
- Admin UI - Alter messages easier  
- End User - Visual collection browsing |
| Enhance Versioning | DSpace provides an optional, basic Item versioning capability. However, this versioning capability is limited in nature, and not yet fully integrated into all DSpace tools and interfaces. We should find a way to enhance item versioning, and make it more embedded into the system (and API) overall.  
- Ranked lower in 2013-14 Vision Survey | Goal 1: Fundamentals of IR | x |  
- Structure - Manual Edit of Existing Items  
- Structure - Automated Retention of All changes to Items  
- Structure - Manual Creation of "New Editions" of an Item  
- Structure - Generated provenance for all added bitstreams |
| Replace Bundle object with Metadata (or similar) | In DSpace, a “Bundle” object is just a simple grouping of Files (Bitstreams) (e.g. “THUMBNAIL”, “TEXT”). Rather than being its own object, a “bundle” could be more easily replaced with metadata on individual Bitstreams (e.g. dc.type=“thumbnail”), or even relationships between Bitstreams (i.e. derivative objects).
- Primary use case description: Structure - purpose of bundle layer
- May also be dependent on or related to “Relationships between Objects” project
- Also see alignment with PCDM, which simply uses metadata on files | Goal 2: Lean and flexible | x | Medium | - Structure - purpose of bundle layer
- Structure - Describe Individual Bitstream within an Item
- Structure - Support for derivative objects |
| Advanced Preservation | DSpace currently only provides basic preservation tools (format identification, checksum verification, etc) out-of-the-box. We should investigate providing enhanced preservation options (or integrations) for users who need it. Likely, this will be implemented as an optional module available via a module registry.
- Likely an optional module or external integration, therefore it may be dependent on a “Module Framework” being developed
- Ranked lower in 2013-14 Vision Survey | Goal 3: Can be "extended" | High | - Admin UI - Advanced Preservation - Format Migration
- Admin UI - Advanced Preservation - Format characterization
- Others under "uc-preservation" label |
| Enhance Search / Browse System | While DSpace’s search/browse system continues to improve, there are still ways we can improve the user experience of finding items within DSpace.
- See various unmet use cases labeled “Search/Browse”
- Dependent on “Single Search / Browse System” feature being completed, to ensure we can more easily implement new Search/Browse use cases
- Some use cases also dependent on “Single User Interface” being completed | Goal 1: Fundamentals of IR | x | End User - Intelligent rendering of URLs in item Metadata
- Admin UI - Trigger the re-index of a collection
- End User - Clear distinction between different types of contributor involvement
- End User - Discovery and retrieval of content on mobile devices
- End User - Search for items
- End User - Browse
- End User - Enhanced access to related items
- Admin UI - Configure and manage browse indexes
- Admin UI - Rebuild the discovery index
- Structure - Create the ability to place “dynamic collections” (pre-faceted view of a collection) within the community hierarchy
- Admin UI - Run media filters
- Admin UI - Clear the OAI Cache
- Admin UI - Rebuild the OAI index
- Admin UI - DSpace should accommodate basic index normalization
- Admin UI - Customise which metadata fields to facet upon, search upon |
DSpace’s access control system (which manages authentication and authorization) is a custom solution that is now nearly as old as the software platform itself. While it functions well enough as-is, replacing it with a comparable third party, open source AuthN/AuthZ system would simplify our codebase and ongoing maintenance.

- See various unmet use cases regarding “Access Control”
- Also see

DS-1566 - Getting issue details...  STATUS

Enhance Approval Workflow capabilities

While directly related to the "Single Approval Workflow system" feature (above), DSpace’s workflow system(s) do not yet meet all community use cases for approval workflows.

This feature is a placeholder for enhancing the Approval Workflow system based on the Workflow Use Cases (see list in the “Use Cases” column). However, it is likely somewhat dependent on standardizing on a single Workflow system.

Goal 1: Fundamentals of IR

Integrations

While integrations are very important to DSpace, these integrations projects are “unscheduled” as of yet. Some of these integrations are “ongoing activities” (e.g. search engine optimization), while others require further definition, or their design may be affected by one or more of the candidate features listed above. Nonetheless, if there is interest in moving one or more of these features forward in the near term, we can immediately schedule it for an upcoming release. These integrations are unordered, but are all considered priorities for DSpace moving forward.

<table>
<thead>
<tr>
<th>Integration</th>
<th>Design / Notes</th>
<th>Related Technical Strategic Goal(s)</th>
<th>Core?</th>
<th>Complexity</th>
<th>Use Cases</th>
</tr>
</thead>
</table>
| Integration with external authentication / authorization system | Also related to Integrations - Authentication through Multiple Mechanisms
Already exists to some extent (LDAP, Shibb, etc). But, would be nice to move to a third-party, standardized AuthN/AuthZ solution.

DS-1566 - Getting issue details...  STATUS | Goal 4: Integration with external services | x | Medium |

Integrations that increase the exposure of content stored into DSpace in external systems

Related: Integrations - Linking to repository content through a learning management system

Goal 4: Integration with external services

Persistent Identifiers other than Handles (DOI)

Related to Integrations - Use of multiple sorts of Direct Object Identifiers
Also related Integrations - Handle System Identifiers

DS-2153 - Getting issue details...  STATUS

Goal 4: Integration with external services

x
<table>
<thead>
<tr>
<th>Category</th>
<th>Description</th>
<th>Goal</th>
<th>Integration Status</th>
</tr>
</thead>
</table>
| **Personal Identifiers (ORCID) (Enhanced)**  | • Related to Integrations - Support external authorities (ORCID) in Authority Cache  
• Also related Integrations - Support for external identifiers (ORCID) in the CSV Batch edit  
• ORCIDs exist in DSpace 5.x, but these use cases suggest enhancements to that system | Goal 4: Integration with external services | x                    |
| **Streaming Video Content**                  | • Provide an ability to stream video in-browser (likely using third-party integrations or tools)  
• Related to End User - Visitor can play media hosted on an external streaming media server | Goal 5: Low cost, "just works" | (Basic version exists in 7.0) |
| **Integrating with third party document streaming services** | • Provide an ability to stream/view (textual) documents (e. g. PDF) in-browser | Goal 5: Low cost, "just works" |                                |
| **Streaming Image Server**                   | • Provide an ability to stream/view/zoom/pan images in-browser  
• Related to End User - Image file display (pan, zoom, size options) | Goal 5: Low cost, "just works" |                                |
| **Search Engine Optimization**               | • Search Engine Optimization  
• While this exists in DSpace, SEO is an ongoing activity and can always be improved upon | Goal 4: Integration with external services | x                    |