


DSpace 2014 Product Plan

This work has transitioned into a Strategic Plan and Technical Roadmap

 This "Product Plan" is more of an analysis of possible feature or use case gaps within the DSpace platform (based on the 2014 Vision Survey). It is NOT a true "Product Plan".

In addition, this document has been replace / reworked into a Strategic Plan for Technology and a corresponding Technical Roadmap. As of 2015, both are in progress and identified on the [Strategic Planning](#) page.

This analysis document has been left for historical reference, but please visit the [Strategic Planning](#) page for more recent strategic planning activities.

- [Candidate Features for DSpace](#)
- [Non-Functional Goals for DSpace](#)

Candidate Features for DSpace

The following table represents a list of candidate features (and corresponding use cases) that DSpace software may wish to implement. They have been sorted by average ranking from the 2014 DSpace Vision Survey results (see [Survey Average Importance](#)), in order to show which features are of highest importance to our community. Each feature has been grouped into a rough "Category" (see [Category descriptions](#)), based on the part of the overall system it relates to. Finally, we've performed a rough analysis (based on corresponding use cases) of whether each feature has been implemented/achieved within the current DSpace platform (see [Status explanations](#)).

Overall, this table provides a way to visualize many of the highly ranked "use case or feature gaps" which have not yet been met within DSpace, and which likely should be prioritized for future software releases.

Candidate Feature	Category* (see categories below)	Status* ✓ = mostly met, ⚠ = partially met, ✗ = not met	Survey Average Importance* 10 = very, 5 = moderately, 0 = not important	Use Cases
Create / manage files and metadata (as an Item)	Structure	✓	10.00	<ul style="list-style-type: none">• Structure - Create / manage files and metadata (as an Item)• Structure - Manual Submission of New Items• Structure - Manual Edit of Existing Items• Structure - Automated Deposit of New Items• Structure - Automated Update of Existing Items• Structure - Management of Deposits / Submissions
Community and Collection hierarchy (or generic containers)	Structure	✓ / ⚠ (no generic containers)	7.37	<ul style="list-style-type: none">• Structure - Community and Collection hierarchy (or generic containers)• Structure - Associate Separate Properties with Each DSpace Community
Create new versions of existing Items	Structure	⚠ (disabled by default, limited)	5.92	<ul style="list-style-type: none">• Structure - Automated Retention of All changes to Items• Structure - Manual Creation of "New Editions" of an Item
Support for derivative objects (e.g. thumbnails, coversheets)	Structure	⚠ (thumbnails & text extraction only)	6.52	<ul style="list-style-type: none">• Structure - Support for derivative objects

Metadata for all levels of object hierarchy (e.g. for Communities and Collections)	Structure	✗	6.99	<ul style="list-style-type: none"> Structure - Metadata for all levels of object hierarchy Structure - Describe Individual Bitstream within an Item Structure - Manage Lists, Controlled Vocabularies and Authority Control
Relationships between objects (e.g. Author items as Authors of an Item rather than textual metadata)	Structure	✗	8.17	<ul style="list-style-type: none"> Structure - Relationships between objects
Support for hierarchical metadata formats (e.g. METS / MODS)	Structure	✗	7.10	<ul style="list-style-type: none"> Structure - Support for hierarchical metadata formats (e.g. METS / MODS)
Item approval workflows	Administrative	✓	7.88	<ul style="list-style-type: none"> Admin UI - Approval workflows
Item embargo facility	End User / Administrative	✓	8.64	<ul style="list-style-type: none"> Admin UI - Embargo functionality
Support for flexible licensing, including Creative Commons	End User	✓	8.44	<ul style="list-style-type: none"> End User - Flexible licensing
CRIS functionality (Current Research Information System: researcher pages and automated publication feeds from external data providers)	End User	✗ (though, DSpace-CRIS is available)	5.51	
Search and browse for Items	End User	✓	9.41	<ul style="list-style-type: none"> End User - Search for items End User - Browse
Easy and intuitive deposit mechanism for users	End User	⚠ (arguable)	8.86	<ul style="list-style-type: none"> End User - Easy and Intuitive Deposit Interface
Batch deposit (multiple item deposit via the user interface)	End User	✗	8.59	<ul style="list-style-type: none"> End User - Batch Deposit
Batch download (multiple item download as a single file)	End User	✗	6.91	
Authentication with common single-sign-on systems (e.g. LDAP, Shibboleth)	Integrations	✓	8.52	<ul style="list-style-type: none"> Integrations - Support External Authentication Systems Integrations - Authentication through Multiple Mechanisms
Most configuration takes place via the administrative user interface	Administrative	✗	7.61	<ul style="list-style-type: none"> Admin UI - Configuration edits to dspace.cfg Admin UI - Database Driven Configuration Property Management Admin UI - Change community/collection parent/child relationship Admin UI - Run batch load Admin UI - Run media filters Admin UI - Manage controlled vocabularies Admin UI - Configure and manage browse indexes
Template-driven user interface for easy branding	Administrative	✗	7.53	<ul style="list-style-type: none"> Admin UI - Theme templates
User interface and theme management takes place via user interface	Administrative	✗	6.47	<ul style="list-style-type: none"> Admin UI - Manage input forms Admin UI - Manage home page Admin UI - Theme management Admin UI - Theme templates

Search Engine crawler-friendly user interface	Integrations	✓	8.91	<ul style="list-style-type: none"> Integrations - Search Engine Optimization
Specialized content delivery mechanisms (media players, page turners, document viewers)	End User	✗	6.83	
Content streaming (video player)	End User	✗	6.92	<ul style="list-style-type: none"> End User - Streaming Video Content
Basic preservation (fixity checks)	Administrative	✓	8.33	<ul style="list-style-type: none"> Admin UI - Basic Preservation - Fixity Checking Admin UI - Basic Preservation - Format identification
Format identification and reporting	Administrative	⚠ (simplistic identification, no reporting)	7.00	<ul style="list-style-type: none"> Admin UI - Format reporting Admin UI - Basic Preservation - Format identification
Advanced preservation services (e.g. format migration)	Administrative	✗	6.61	<ul style="list-style-type: none"> Admin UI - Advanced Preservation - Format Migration Admin UI - Advanced Preservation - Format characterization
Persistent identifiers: handles	Integrations	✓	8.32	<ul style="list-style-type: none"> Integrations - Handle System Identifiers
Additional external persistent identifiers: DOIs, DataCite	Integrations	⚠ (basic support only)	7.56	<ul style="list-style-type: none"> Integrations - Persistent Identifiers other than Handles (DOI) Integrations - Support Standard Persistent Identifiers
Other external identifiers: ORCID	Integrations	✗ (some current funding / development)	7.22	<ul style="list-style-type: none"> Integrations - Personal Identifiers (ORCID) Integrations - ORCID Lookup during item submission Integrations - ORCID Lookup during item edit Integrations - Support external authorities (ORCID) in Authority Cache Integrations - Support for external identifiers (ORCID) in the CSV Batch edit
Basic statistics (item access counts, file downloads)	Stats/Metrics	⚠ (limited reporting)	9.03	<ul style="list-style-type: none"> Stats/Metrics - Basic Statistics
Advanced statistics (search terms, geographical locations)	Stats/Metrics	⚠ (limited reporting)	8.13	<ul style="list-style-type: none"> Stats/Metrics - Advanced Statistics
Altmetrics	Stats/Metrics	✗	7.30	
Open Access status tracking and compliance checking	Stats/Metrics / Integrations	✗	7.27	
Standard repository machine interfaces (e.g. OAI-PMH, SWORDv2, ResourceSync)	Integrations	✓ (but, no ResourceSync yet)	8.11	<ul style="list-style-type: none"> Integrations - Support Standard Repository Machine Interfaces (OAI-PMH, SWORD, ResourceSync)
REST API for building external applications	Integrations	✓ (beta)	7.56	<ul style="list-style-type: none"> Integrations - REST API

Feature Categories

Features have been grouped into rough "categories" as follows:

- Structure: Features that relate to the architecture/structure of content within DSpace
- Administrative: Features which are mostly administrative functions/activities
- End User: Features relating to how end users interact with the system (deposit, download, search)
- Integrations: Features which are either used by external systems (or programs) or interact themselves with an external system
- Stats/Metrics: Features relating to metrics or statistics around content and its usage

Status explanation

- ✓ - Arguably, DSpace meets *most* major use cases already. (However, there still may still be areas for improvement or enhancement.)
- ⚠ - DSpace meets some of the major use cases, but there are other major use cases which it does not yet meet.
- ✗ - DSpace does not really meet these use cases, yet.

Formula for "Survey Average Importance"

The "Survey Average Importance" represents average importance ranking of all **respondents** for a single feature from the 2013-14 DSpace Vision Survey (Survey results linked off of: [DSpace 2013 Vision Discussions](#))

- Very Important = 10 points
- Moderately important = 5 points
- Not important = 0 points

So, for example, "Community and Collection Hierarchy" had 50 respondents rank it "Very important" (10 points each), 37 rank it "Moderately important" (5 points each), and 6 rank it "Not important" (0 points).

This is an average score of $((10 \times 50) + (5 \times 37)) / (50 + 37 + 6) = 7.37$ average (which is midway between a "moderately important" and "very important" ranking)

A spreadsheet was used to calculate all average scores. See the attached [DSpace_Vision_Survey_Feature_Rankings.xls](#)

Non-Functional Goals for DSpace

The following table describes high-level, non-functional goals which DSpace software should strive to achieve. These "non-functional" goals do not map directly to features or functions within DSpace, and in fact, some goals may be subjective in nature. However, they are perceived as *highly important* in order to continue to maintain and grow our active, open source community of users and developers.

Non-functional Goal	Use Cases / Details
Easy to install / deploy	<ul style="list-style-type: none">• One click installer?• Pre-packaged, "drop-in", web applications (WARs)?• Related feedback on current DSpace installation/setup woes: http://gavialib.com/2014/04/could-your-software-be-taught-in-a-library-school/• Imbedded Servlet Engine• Relative Execution / No hardcoded path to dspace config
Easy to upgrade	<ul style="list-style-type: none">• "auto-upgrader" option? (script to upgrade, or it auto-upgrades data/configs when you load the new WAR, similar to Atlassian tools)• Store configuration properties/options in database
Scalability / Performance	Some Suggestions: <ul style="list-style-type: none">• Uploading/Processing content in DSpace should not be bound to size of content• DSpace should strive to operate in with constant memory usage footprint during all operations.• DSpace user experience should not be performance limited by number of Communities, Collection, Items and/or Bitstreams• Support Alternative Assetstore Technologies• Support Many Database Vendors
Attractive to new developers	<ul style="list-style-type: none">• Opportunities to learn / contribute• Adopt Current and Industry Standard Technologies• Good developer and technical documentation• Supportive community

Attractive to new Repository/Content Managers	<ul style="list-style-type: none"> • Good user documentation • Supportive community • Usage best practices / guidelines • More UI-Driven Admin Functionality
Avoid centrally maintaining duplicative codebases/functions	<ul style="list-style-type: none"> • Standardize on a single out-of-the-box UI <ul style="list-style-type: none"> ◦ other third-party UIs may be offered, but would not be centrally maintained • Centralize common UI Actions / Commands and consolidate into Business Tier.