

# DSpace 2013 Vision Document

Long-term Vision vs. DSpace 4.0 / 5.0

 In case it is not evident, this Vision Document describes a longer-term vision for the DSpace platform. It does NOT have any relationship to the [DSpace 4.0 Release](#) in 2013 or the DSpace 5.0 release scheduled for end of 2014. Both DSpace 4.0 and 5.0 will just involve improvements to the existing DSpace platform. This Vision Document describes DSpace software's "aspirations" or the overarching goals which DSpace will strive to achieve.

Latest Vision Related Work

 The latest Vision-related work (as of June 2014) is occurring at:

- Next steps overview and notes (including links to Vision Survey results): [DSpace 2013 Vision Discussions](#)
- Product Planning (based on Vision and Vision Survey results): [DSpace 2014 Product Plan](#)
- Also see the [Governance Model](#) and [Product Planning Process](#)
  
- [DSpace 3-5 Year Vision Statement](#)
- [OR13 Presentation on Vision Statement](#)
- [DSpace Use Cases / Features \(Very Rough Draft / Notes\)](#)
  - [Primary / Core Features](#)
  - [Possible Extensions / Plugins](#)
  - [Likely Extensions / Plugins](#)

## DSpace 3-5 Year Vision Statement

A presentation version of this Vision Statement is also available at: <https://www.dropbox.com/s/ij3j77iuit0h1k/DSpace%20vision.pdf>

The following statements describe, at a high level, what goals the DSpace open source product should strive to meet.

1. DSpace will focus on the fundamentals of the modern "Institutional Repository" use case. We are striving to meet the IR needs of the next 5-10 years.
2. DSpace will be "lean", with agility and flexibility as primary goals.
3. DSpace will include a "core" set of functionality that can be "extended" (think plugins) or have "hooks" (integration points) to complimentary services/tools
4. DSpace will be designed in such a way that it can be easily/quickly configured to integrate with new & future tools/services in the larger digital scholarship "ecosystem"
5. DSpace will support low-cost, hosted solutions and deployments (by featuring an easy, "just works" setup)

For details about about possible fundamental IR use cases, please see the "DSpace Use Cases / Features" section below.

## OR13 Presentation on Vision Statement

An overview of the draft DSpace 3-5 Year Vision (as presented at the [Open Repositories 2013](#) conference) is provided in the following "DSpace 2013 RoadMap & 3-5 Year Vision" screencast by Tim Donohue of DuraSpace.

The slides from the screencast and the OR13 talk are also available at: <http://www.slideshare.net/todonohue/dspace-roadmap-vision-2013-or13>

## DSpace Use Cases / Features (Very Rough Draft / Notes)

What follows is a (rough) listing of core use cases & features that DSpace should strive to meet, based on the "3-5 year vision statement" detailed above.

These lists are by no means final, they are just an initial brainstorm of organizing use cases and features into three main categories: "Primary / Core Features", "Possible Extension/Plugins", "Likely Extensions/Plugins". The goals are to generate discussion and attempt to determine which use cases /features are fundamental to all DSpace users and which could be provided by a (third party or centrally maintained) plugin/extension.

Other Repository Use Case Lists for Possible Reference

 Recently, other repository systems have done some similar Use Case / Requirements lists. There may be some overlap in these use cases that could be helpful in our brainstorms / analysis.

- [Fedora Futures / Fedora 4 Use Cases](#) list (sub-pages of this page describe each use case in more detail)
  - Higher level requirements / goals also listed on [Fedora 4 Roadmap](#)
- Hydra's list of [IR Solution Requirements / Use Cases](#)

## Primary / Core Features

The following features seem to be primary use cases of DSpace. Therefore, these features should likely be immediately available in "out-of-the-box" DSpace and should require no extra installation/setup:

- Basic Functionality:

- Create, Read, Update Delete (CRUD) on objects
  - Versioning of objects
  - Basic Search & Browse functionality
  - Basic Preservation functionality (e.g. Fixity checks)
  - Basic Statistics (or "hooks" into external statistics engines)
- Content Model
  - Should support a Community & Collection "like" hierarchy
    - Doesn't necessarily require Communities and Collections to be separate object types. They could just be a single "Container" type of object.
  - Items are the primary type of object. Items include File(s).
    - Note: Old concept of "bundles" may need rethinking.
  - *New object type*: Author objects (which hold metadata about authors/researchers in the system)
  - Persistent Identifier support for all objects
  - Support for object derivatives (e.g. thumbnail images)
  - More flexible relationships between objects
    - Including aggregations of objects & complex objects
- Metadata
  - Structured Metadata
  - Metadata should be at all levels of object hierarchy.
    - Administrative/Preservation Metadata at all levels, including on individual Files
  - Hierarchical Metadata formats should be supported
- Upload / Download of Content:
  - Self deposit & mediated (approval workflow based) deposit of content
  - Batch Deposit of content (from a UI)
  - Batch Download of content (from a UI)
  - Large File support for End Users
    - End Users should be able to upload and download large files themselves
- Access controls (Authentication & Authorization)
  - Authorization controls at all levels of object hierarchy
  - Also includes Embargo-style access controls
- User Interface Functionality
  - Single, default out-of-the-box User Interface
    - Preferably some sort of template-driven UI framework
  - User Interface should be "SEO Friendly"
  - Configuration Management takes place in the UI
  - UI Template/Theme Management takes place in the UI
- Standard Machine Interfaces
  - Some high priorities: OAI, SWORD, REST API
  - When possible, machine interfaces should be able to target content at any level (Community, Collection, Item, File, Author)
- Licensing support
  - Both deposit license and Creative Commons licensing
  - Enhanced "third party" licensing options, e.g. where a student deposits on behalf of a professor, or a librarian deposits ETDs on behalf of students. Essentially, cannot always assume that the person depositing content has sufficient rights to accept a "deposit license" on behalf of actual author(s). (Suggested by [LibSkrat via Twitter](#))
    - Possibly related: "Depositing under an Open Access mandate", where there may be a required (configurable) license agreement based on either local or national OA mandate.
- Ability to easily "hook" into external tools & services
  - e.g. something like Curation Tasks & other more robust ways to integrate with other tools/services

## Possible Extensions / Plugins

The following features may or may not be available in "out-of-the-box" DSpace. It's arguable whether these are primary use cases that DSpace should support. It's possible some of these features could be handled by plugins which you install in DSpace, or by "hooks" into external services/tools.

- Richer Licensing support (individual CC licenses on individual files)
- Support for Delivery of Media
  - Doc Viewers / Page Turners
  - Geospatial
  - Streaming content
- Alt-metrics (downloads, tweets, etc.)
- Support for small scale research data sets
  - Relationship back to publication (linked)
  - Also may include software programs
- Metadata extensibility
  - Stronger support for channeling user contributed metadata
  - Schema agnostic
- Compliance with Open Access directives (of various countries)
  - models to track with general worldwide OA directives
  - when possible, methods to check compliance
  - when possible, support for automated evaluation
- Improved Statistics (could be external, e.g. Google Analytics)
- Improved Support for External Identifiers (DOIs, Handles, etc.)
- Customized / Flexible UI support
  - e.g. Users should be able to change their Collection's "theme" or "template"

## Likely Extensions / Plugins

The following features seem to not be primary use cases/needs of DSpace. Therefore, these features would likely NOT be provided by "out-of-the-box" DSpace. They would either need to be implemented as plugins to DSpace, or DSpace would integrate into external services/tools which provide these features.

- Advanced Statistics engine
  - Instead should look towards better integration with Google Analytics, or other external statistics engines
- Advanced Preservation Activities
  - Instead should provide integration with external preservation tools / services (via Curation Tasks or something similar)
- Digital Publishing Activities
  - Instead, should provide integration with external publishing systems
- CRIS (Current Research Information System) functionality
  - Instead, DSpace should integrate with external CRIS systems, or offer a CRIS plugin.