

# DSpace Release 1.7.0 Notes

## Version 1.7.0

DSpace 1.7.0 was officially released to the public on December 17, 2010.

DSpace 1.7.0 can be downloaded immediately at either of the following locations:

- SourceForge: <https://sourceforge.net/projects/dspace/files/>
- SVN: <https://svn.duraspace.org/dspace/dspace/tags/dspace-1.7.0/>

Recommended to Upgrade to DSpace 1.7.1

We recommend all DSpace 1.7.0 users upgrade to DSpace 1.7.1, to receive several bug fixes along with a medium-level security fix. For more information see:

- [DSpace 1.7.1 Announcement](#)
  - [DSpace Release 1.7.1 Notes](#)
- 1 [New features in DSpace 1.7](#)
  - 2 [Performance Improvements](#)
  - 3 [Bug Fixes](#)
  - 4 [Changes](#)
  - 5 [Removed / Deprecated](#)
  - 6 [Organizational Details](#)
    - 6.1 [Release Coordination](#)
    - 6.2 [Timeline and Proceeding](#)

**DSpace 1.7.0 is a scheduled, "time-based" release.** In order to decrease delays in releasing new features and increase transparency, the DSpace Developers scheduled 1.7.0 in advance and based its features on what we were able to complete within that timeframe. Despite the fact that 1.7.0 had a much tighter timeframe than previous major releases, the developers have managed to include some significant new features, numerous bug fixes and performance improvements.

Scheduling releases benefits us all as it should decrease the delays in releasing new features, and increase the transparency of the development process. The DSpace Developers feel that these benefits will far outweigh the cost of potentially having fewer major features in a given DSpace release. We hope the DSpace Community will also realize the immediate benefits, which should allow them to receive new features more quickly, rather than potentially waiting years for the next major release of the software. The DSpace Developers hope to continue this trend of "time based" releases with all future releases.

## New features in DSpace 1.7

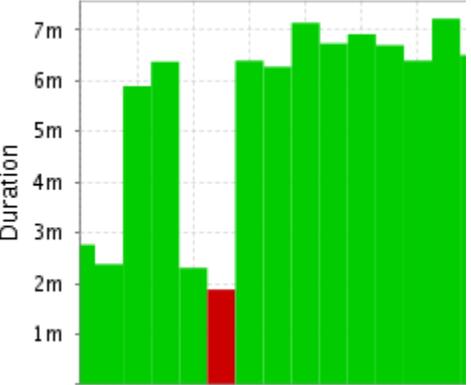


Mirage, a clean and professional looking theme for XMLUI.

dri2xhtml-alt, and xmlui theme development framework that eases XMLUI theme development.

See Mirage in action at <http://demo.dspace.org/xmlui>. For a video demo see: [http://www.youtube.com/watch?v=Pq3d\\_oD-4aM](http://www.youtube.com/watch?v=Pq3d_oD-4aM)

Mirage was contributed by @mire, NV.

<p><b>Discover</b></p> <p>Author</p> <ul style="list-style-type: none"> <li><a href="#">doe, john (100)</a></li> <li><a href="#">doe, jane (52)</a></li> <li><a href="#">smith, john (51)</a></li> <li><a href="#">smith, jane (50)</a></li> <li><a href="#">ackerman, samuel (25)</a></li> <li><a href="#">avery, jason (25)</a></li> <li><a href="#">beecher, tim (25)</a></li> <li><a href="#">brown, charles m. (25)</a></li> <li><a href="#">brown, charlie (25)</a></li> <li><a href="#">calvin, john (25)</a></li> <li><a href="#">... View More</a></li> </ul>	<p>Discover, a faceted browsing and searching interface that gives a deeper and more intuitive look at repository contents. See Discovery in action at <a href="http://demo.dspace.org/xmlui">http://demo.dspace.org/xmlui</a>. For more information, watch the video <a href="#">introduction to Discovery</a>.</p> <p>Discovery was contributed by <a href="#">@mire, NV</a>.</p>
	<p>Archival Information Package (AIP) <a href="#">Backup &amp; Restore</a> process. Allows for a backup of DSpace into a generic METS-based structure, that can be used to migrate DSpace content to another system that supports AIP's (DSpace or non-DSpace). This backup and restore functionality also allows one to backup to cloud storage services like <a href="#">Dura Cloud</a>, though it could just as easily be used to backup to tape or a hard drive.</p> <p>Added by Tim Donohue (DuraSpace)</p>
<p><b>Task:</b></p> <p><input type="text" value="Check for Required Metadata"/></p> <p><input type="button" value="Perform"/> <input type="button" value="Queue"/> <input type="button" value="Return"/></p>	<p><a href="#">Curation System</a>, a framework for building and running tasks to help a Curator preserve and improve your repository contents. Tasks can be run on communities, collections, and items through the command line for cron-tasks, or through the User Interface for admins. The initial tasks available are:</p> <ul style="list-style-type: none"> <li>* Profile Bitstream Formats -- counts the number of bitstreams that share the same file format extension.</li> <li>• Virus Scan -- inspect the bitstreams with a virus scanner (ClamAV) to detect if they contain viruses</li> <li>• Check for Required Metadata -- checks that item metadata has values for all fields marked as required in the input-form</li> </ul> <p>Added by Richard Rodgers (MIT)</p>
	<p>Automated <a href="#">Unit Testing</a> of core code -- helps the developers ensure that DSpace is as bug free and stable as possible. Unit Testing coupled with continuous integration on our bamboo server allows us to validate every change to the DSpace code base. Thus letting us know immediately if something changed broke another feature.</p> <p>Added by Pere Villega, a product of <a href="#">DSpace Summer of Code2</a> 2010 (mentor Stuart Lewis).</p>
	<p>Improved Google Scholar metadata exposure. Additional citation_ tags have been exposed to allow the Google Scholar crawler to find better associate repository metadata and PDF content.</p> <p>Added by Sands Fish, Richard Rodgers (MIT) and Peter Dietz (Ohio State)</p>
	<p>PowerPoint text extraction, for searching within PowerPoint slides</p> <p>Added by Keith Gilbertson (Georgia Tech)</p>
	<p>Top 10 Most Visited items list, available for the overall site.</p>

Performance Improvements

Performance and Scalability improvements. The code has been thoroughly analyzed by a suite of code quality tools to find blatant errors and omissions, more efficient ways of doing things, and implementing general best practices in the code. Additionally, numerous immeasurable performance gains have been made with regard to item ingestion and indexing speed. This was tested by adding a sample-data-generator, in which 400,000 items were added to a repository already containing 100,000 items, where by the total length of time to ingest items reduced to items per second, as opposed to seconds per item. Adding so many items would previously taken weeks or more, but the latest performance feat was done in 10 hours on a laptop. See more at [DS-707](#).

**Many thanks go out to Graham Triggs (BioMed Central) for many sleepless weeks to vastly overhaul many weak links.**

In addition to that, many other general improvements are:

- Reducing the cost of browse prunes
- SOLR is using autoCommit to reduce resource exhaustion
- SOLR has an optimization function runnable from the command-line for crontasks to essentially "defragment" the solr index.
- Item bitstream sorting/ordering can be specified according to sequence or name
- Moving the documentation into the Confluence wiki so that workload can be divided, and that the documentation is improved.

## Bug Fixes

Major Bug Fixes include:

- Batch Metadata Import will now validate metadata fields in CSV's
- Restricted items / metadata is better protected from exposure via web services: OAI
- File handle leak in ItemImporter closed. Fixes issues when max\_files\_open exceeded on some systems.
- Database connections released when no longer needed in xmlui BitstreamReader. Fixes problem getting connections from the database pool while simultaneously downloading multiple large files.

## Changes

For a full list of all changes (new features, improvements, and bug fixes), please visit the [Changes in DSpace 1.7.0](#) section of the new wiki-based [DSpace Documentation](#).

## Removed / Deprecated

Most command line scripts that have historically resided in `[dspace]/bin/` were deprecated in 1.6.x, and are now removed in 1.7.0. They have been replaced with the [configurable command launcher](#), which eases the cross platform development of scripts. Full details of the discussion are at: <http://jira.dspace.org/jira/browse/DS-646>.

The old way will no longer work, as the task scripts have been removed:

```
[dspace]/bin/create-administrator
```

The functionality is all performed by the centralized DSpace launcher, e.g.:

```
[dspace]/bin/dspace create-administrator
```

Calling a command by its full classname still works by adding dsrun before the classname.

```
[dspace]/bin/dspace dsrun org.dspace.administer.CreateAdministrator
```

## Organizational Details

### Release Coordination

- Release Coordinator: Peter Dietz, Ohio State University Libraries
- Release Coordinator: Tim Donohue, DuraSpace

### Timeline and Proceeding

**Release Timeline:**

-  *August 13, 2010* : Milestone 1 - "Feature Decision Day"
-  *October 22, 2010* : Feature Freeze
-  *October 29, 2010* : Final Documentation "Due Date"
-  *November 5, 2010* : Release Candidate 1
-  *November 8-19, 2010* : [1.7 Testathon](#)
-  *December 3, 2010* : Release Candidate 2

-  December 6-15, 2010 : Final Testing / Bug Fixing
-  December 17, 2010 : Final Release

Release Process needs to proceed according to the following Maven release process: [Release Procedure](#)

## Postponed for a Future Release

The following projects were considered for 1.7, but were not stable enough to be included. They need further review and development from the stakeholders before they are suitable for widespread use, they may be considered for a future release of DSpace. The next release they will be reconsidered for is 1.8

- REST API - Using standard web services to CRUD DSpace Objects. A product of previous GSOC.
- SWORD Client for DSpace – (Robin Taylor, and possibly Richard Jones & Stuart Lewis)
  - would allow DSpace to push/submit content to other SWORD enabled repositories
  - For closed & open access repositories – add a button to transfer content from a closed to an open repository.
- [CGIProposal](#)(Richard Rodgers/MIT)
  - would allow for type-based submission processes (e.g. Theses/Dissertations could have different submission steps than articles/papers).
  - based on the [Item type based submission patch](#) picked up by Robin Taylor (initially a GSoC project)
- Context Guided Ingest – define an interface, where any submission code can write "attributes" and can retrieve those again later on. Can add any new attributes/values that you want for your submission code. Could be serialized to XML (using input-forms.xml) OR have an implementation of that service that stores in DB (recommended). JPA2?
  - seems similar to SimpleStorage Service (user centered storage of state info) – Mark Diggory.
- Rewrite of Creative Commons licensing (MIT)
  - would improve upon the features of the current CC licensing submission step
  - Currently only against XMLUI from MIT
  - Legacy problem – do we update old license to new or not? Currently MIT runs 'split version' with old licenses looking like old, and new look like new.