Search Engine Optimization

Please be aware that individual search engines also have their own guidelines and recommendations for inclusion. While the guidelines below apply to **most** DSpace sites, you may also wish to review these guidelines for specific search engines:

- "Indexing Repositories: Pitfalls and Best Practices" talk from Anurag Acharya (co-creator of Google Scholar) presented at the Open Repositories 2015 conference
- Google Scholar Inclusion Guidelines
- Bing Webmaster Guidelines

Ensuring your DSpace is indexed

Anyone who has analyzed traffic to their DSpace site (e.g. using Google Analytics or similar) will notice that a significant (and in many cases a majority) of visitors arrive via a search engine such as Google or Yahoo. Hence, to help maximize the impact of content and thus encourage further deposits, it is important to ensure that your DSpace instance is indexed effectively.

DSpace comes with tools that ensure major search engines (Google, Bing, Yahoo, Google Scholar) are able to easily and effectively index all your content. However, many of these tools provide some basic setup. Here's how to ensure your site is indexed.

For the optimum indexing, you should:

1. Keep your DSpace up to date. We are constantly adding new indexing improvements in new releases
2. Ensure your DSpace is visible to search engines.
3. Ensure your proxy is passing X-Forwarded headers to the User Interface
4. Ensure the user interface is using server-side rendering *(enabled by default)*
5. Ensure the sitemaps feature is enabled. *(enabled by default)*
6. Ensure your robots.txt allows access to item "splash" pages and full text.
7. Ensure item metadata appears in HTML headers correctly.
8. Avoid redirecting file downloads to Item landing pages
9. Turn OFF any generation of PDF cover pages
10. As an aside, it's worth noting that OAI-PMH is generally not useful to search engines. OAI-PMH has its own uses, but do not expect search engines to use it.

Keep your DSpace up to date

We are constantly adding new indexing improvements to DSpace. In order to ensure your site gets all of these improvements, you should strive to keep it up-to-date. For example:

- As of DSpace 7.0, Sitemaps are enabled by default (see below)
- As of DSpace 5.0, the DSpace robots.txt file now includes references to Sitemaps by default (see https://github.com/DSpace/DSpace/issues/5302) , and also blocks known bad bots (see https://github.com/DSpace/DSpace/issues/5701).
- As of DSpace 4.0, DSpace has provided several enhancements, which were requested by the Google Scholar team. These included providing users (and web indexers) a way to browse content by the date it was added to DSpace (see https://github.com/DSpace/DSpace/issues/4851), ensuring the "dc.date.issued" field is set more accurately (see https://github.com/DSpace/DSpace/issues/4850), and enhancing the logic behind the "citation_pdf_url" HTML <meta> tag (see https://github.com/DSpace/DSpace/issues/4852)
- As of DSpace 1.7, DSpace has improved how its Item-level metadata is made available to Google Scholar. For the 1.7.0 release, the DSpace Developers worked directly with the Google Scholar developers, to ensure DSpace is generating the "citation_*" HTML "<meta>" tags (i.e. Highwire Press tags) that Google Scholar recommends in their Indexing Guidelines.
- As of DSpace 1.5, DSpace has support for sitemaps (both simple HTML pages of links, as well as the sitemaps.org protocol). It also includes item metadata in the HTML HEAD element of item display pages, ensuring that the metadata can be effectively indexed no matter what changes you might have made to your DSpace's layout or style.
- As of DSpace 1.4, DSpace has support for the "if-modified-since" HTTP header. This basically means that if an item (or bitstream therein) has not changed since the last time a search engine's crawler indexed it, that item/bitstream does not have to be re-retrieved, sparing your server.

Additional minor improvements / bug fixes have been made to more recent releases of DSpace.

Ensure your DSpace is visible to search engines

First ensure your DSpace instance is visible, e.g. with: https://www.google.com/webmasters/tools/sitestatus

If your site is not indexed at all, all search engines have a way to add your URL, e.g.:

- Google: http://www.google.com/addurl
- Yahoo: http://siteexplorer.search.yahoo.com/submit
- Bing: http://www.bing.com/docs/submit.aspx

Ensure your proxy is passing X-Forwarded headers to the User Interface

Some HTML tags important for SEO, such as the "citation_pdf_url" tag, require the full URL of your site. The DSpace user interface will automatically attempt to "discover" that URL using HTTP Headers.

Because most DSpace sites use some sort of proxy (e.g. Apache web server or Nginx or similar), this **requires** that the proxy be configured to pass along proper X-Forwarded-* headers, especially X-Forwarded-Host and X-Forwarded-Proto. For example in Apache HTTPD, you can do something like this:

```
# This lets DSpace know it is running behind HTTPS and what hostname is currently used
# (requires installing/enabling mod_headers)
RequestHeader set X-Forwarded-Proto https
RequestHeader set X-Forwarded-Host my.dspace.edu
```

Ensure the user interface is using server-side rendering

In DSpace 7, server-side rendering is *enabled by default (when running in production mode)*. However, it's important to ensure you do *not* disable it in production mode. Per the frontend Installation instructions, you MUST also be running your user interface in production mode (via either `yarn run serve:ssr` or `yarn start`).

Because the DSpace user interface is based on Angular.io (which is a javascript framework), you MUST have server-side rendering enabled (which is the default) for search engines to fully index your side. Server-side rendering allows your site to still function even when Javascript is turned *off* in a user's browser. Some web crawlers do not support Javascript (e.g. Google Scholar), so they will only interact with this server-side rendered content.

If you are unsure if server-side rendering (SSR) is enabled, you can check to see if your site is accessible when Javascript is turned **off.** For example, in Chrome, you should be able to do the following:

1. Open your site in the Chrome browser
2. Turn off (disable) Javascript using the Chrome instructions: https://developer.chrome.com/docs/devtools/javascript/disable/
3. Click reload in your browser window to reload your site.
    a. If SSR is enabled, then you will still see your site's contents. You should be able to browse & search the site. (Keep in mind, pages may take longer to load because every request requires SSR.) However, all dynamic menus or actions obviously will not work, as all pages will be static HTML.
    b. If SSR is disabled, then you will see a blank white page. You will not be able to see any content on your site.
4. Don't forget to re-enable Javascript after you are done testing (see link above, or just close that window & reopen a new one)

DSpace use Angular Universal for server-side rendering, and it's enabled by default in Production mode via our production environment initialization in src /environments/environment.production.ts:

```
// Angular Universal Settings
universal: {
  preboot: true,
  ...
},
```

For information, see "Universal (Server-side Rendering) settings" in User Interface Configuration

Ensure the sitemaps feature is enabled

As of DSpace 7, sitemaps are *enabled by default and automatically update on a daily basis.* This is the recommended setup to prefer proper indexing. So, there's nothing you need to do unless you wish to either change their schedule, or disable them.

In the dspace.cfg, the Sitemap generation schedule is controlled by this setting

```
# By default, sitemaps regenerate daily at 1:15am server time
sitemap.cron = 0 15 1 * * ?
```

You can modify this schedule by using the Cron syntax defined at https://www.quartz-scheduler.org/api/2.3.0/org/quartz/CronTrigger.html . Any modifications can be placed in your local.cfg.

If you want to disable this automated scheduler, you can either comment it out, or set it to a single "-" (dash) in your local.cfg

```
# This disables the automatic updates
sitemap.cron = -
```

Again, we **highly recommend** keeping them enabled. However, you may choose to disable this scheduler if you wish to define these in your local system cron settings.

Once you've enabled your sitemaps, they will be accessible at the following URLs:

- HTML Sitemaps: ${dspace.ui.url}/sitemap_index.html
- XML Sitemaps: ${dspace.ui.url}/sitemap_index.xml

So, for example, if your "dspace.ui.url = https://mysite.org" in your "dspace.cfg" configuration file, then the HTML Sitemaps would be at: "http://mysite.org /sitemap_index.html"

By default, the Sitemap URLs also will appear in your UI's `robots.txt` (in order to announce them to search engines):

```
# The URL to the DSpace sitemaps
# XML sitemap is listed first as it is preferred by most search engines
Sitemap: [dspace.ui.url]/sitemap_index.xml
Sitemap: [dspace.ui.url]/sitemap_index.html
```

The generate-sitemaps command

If you wanted to generate your sitemaps manually, you can use a commandline tool to do so.

WARNING: Keep in mind, you do NOT need to run these manually in most situations, as sitemaps are autoupdated on a regular schedule (see documentation above)

```
# Commandline option (run from the backend)
[dspace]/bin/dspace generate-sitemaps
```

This command accepts several options:

Option	meaning
-h --help	Explain the arguments and options.
-s --no_sitemaps	Do not generate a sitemap in sitemaps.org format.
-b -no_htmlmap	Do not generate a sitemap in htmlmap format.

You can configure the list of "all search engines" by setting the value of `sitemap.engineurls` in `dspace.cfg`.

Create a good robots.txt

As of 7.5, DSpace's robots.txt file can be found in the UI's codebase at "src/robots.txt.ejs". This is an "embedded javascript template" (ejs) file, which simply allows for us to insert variable values into the "robots.txt" at runtime. It can be edited as a normal text file.

The trick here is to minimize load on your server, but without actually blocking anything vital for indexing. Search engines need to be able to index item, collection and community pages, and all bitstreams within items – full-text access is critically important for effective indexing, e.g. for citation analysis as well as the usual keyword searching.

If you have restricted content on your site, search engines will not be able to access it; they access all pages as an anonymous user.

Ensure that your robots.txt file is at the top level of your site: i.e. at http://repo.foo.edu/robots.txt, and NOT e.g. http://repo.foo.edu/dspace/robots.txt. If your DSpace instance is served from e.g. http://repo.foo.edu/dspace/, you'll need to add /dspace to all the paths in the examples below (e.g. /dspace/browse-subject).

NEVER BLOCK THESE PATHS

Some URLs can be disallowed without negative impact, but be ABSOLUTELY SURE the following URLs can be reached by crawlers, i.e. DO NOT put these on Disallow: lines, or your DSpace instance might not be indexed properly.

- `/bitstreams`
- `/browse/*` (UNLESS USING SITEMAPS)
- `/collections`
- `/communities`
- `/community-list` (UNLESS USING SITEMAPS)
- `/entities/*`
- `/handle`
- `/items`

Example good robots.txt

DSpace 7 comes with an example robots.txt file (which is copied below). As of 7.5, this file can be found at "src/robots.txt.ejs" in the DSpace 7 UI. This is an "embedded javascript template" (ejs) file, which simply allows for us to insert variable values into the "robots.txt" at runtime. It can be edited as a normal text file.

The highly recommended settings are uncommented. Additional, optional settings are displayed in comments – based on your local configuration you may wish to enable them by uncommenting the corresponding "Disallow:" line.

```
# The URL to the DSpace sitemaps
# XML sitemap is listed first as it is preferred by most search engines
# NOTE: The <%= origin %> variables below will be replaced by the fully qualified URL of your site at runtime.
Sitemap: <%= origin %>/sitemap_index.xml
Sitemap: <%= origin %>/sitemap_index.html

##########################
# Default Access Group
# (NOTE: blank lines are not allowable in a group record)
##########################
User-agent: *
# Disable access to Discovery search and filters; admin pages; processes; submission; workspace; workflow &
profile page
Disallow: /search
Disallow: /admin/*
Disallow: /processes
Disallow: /submit
Disallow: /workspaceitems
Disallow: /profile
Disallow: /workflowitems

# Optionally uncomment the following line ONLY if sitemaps are working
# and you have verified that your site is being indexed correctly.
# Disallow: /browse/*
#
# If you have configured DSpace (Solr-based) Statistics to be publicly
# accessible, then you may not want this content to be indexed
# Disallow: /statistics
#
# You also may wish to disallow access to the following paths, in order
# to stop web spiders from accessing user-based content
# Disallow: /contact
# Disallow: /feedback
# Disallow: /forgot
# Disallow: /login
# Disallow: /register

# NOTE: The default robots.txt also includes a large number of recommended settings to avoid misbehaving bots.
# For brevity, they have been removed from this example, but can be found in src/robots.txt.ejs
```

WARNING: for your additional disallow statements to be recognized under the `User-agent: *` group, they *cannot be separated by white lines* from the declared `user-agent: *` block. A white line indicates the start of a new user agent block. Without a leading user-agent declaration on the first line, blocks are ignored. Comment lines are allowed and will not break the user-agent block.

This is OK:

```
User-agent: *
# Disable access to Discovery search and filters; admin pages; processes
Disallow: /search
Disallow: /admin/*
Disallow: /processes
```

This is **not OK**, as the two lines at the bottom will be completely ignored.

```
User-agent: *
# Disable access to Discovery search and filters; admin pages; processes
Disallow: /search

Disallow: /admin/*
Disallow: /processes
```

To identify if a specific user agent has access to a particular URL, you can use this handy robots.txt tester.

For more information on the robots.txt format, please see the Google Robots.txt documentation.

Ensure Item Metadata appears in the HTML HEAD

It's possible to greatly customize the look and feel of your DSpace, which makes it harder for search engines, and other tools and services such as Zotero, Connotea and SIMILE Piggy Bank, to correctly pick out item metadata fields. To address this, DSpace includes item metadata in the <head> element of each item's HTML display page.

```
<meta name="DC.type" content="Article" />
<meta name="DCTERMS.contributor" content="Tansley, Robert" />
```

If you have heavily customized your metadata fields away from Dublin Core, you can modify the service which generates these elements by modifying https://github.com/DSpace/dspace-angular/blob/main/src/app/core/metadata/metadata.service.ts

Google Scholar Metadata in HTML HEAD

In addition to Dublin Core <meta> tags in the HTML HEAD, DSpace also includes Google Scholar specific metadata fields in each item's HTML display page.

```
<meta property="citation_author" content="Tansley, Robert; Donohue, Timothy"/>
<meta property="citation_title" content="Ensuring your DSpace is indexed" />
```

These meta tags are the "Highwire Press tags" which Google Scholar recommends. If you have heavily customized your metadata fields, or wish to change the default "mappings" to these Highwire Press tags, you may do so by modifying https://github.com/DSpace/dspace-angular/blob/main/src/app/core/metadata/metadata.service.ts (see for example the "setCitationAuthorTags()" method in that service class)

Much more information is available in the Configuration section on Google Scholar Metadata Mappings.

Avoid redirecting file downloads to Item landing pages

Make sure that you never redirect "direct file downloads" (i.e. users who directly jump to downloading a file, often from a search engine) to the associated Item's splash/landing page. In the past, some DSpace sites have added these custom URL redirects in order to facilitate capturing statistics via Google Analytics or similar.

While these URL redirects may seem harmless, they may be flagged as cloaking or spam by Google, Google Scholar and other major search engines. This may hurt your site's search engine ranking or even cause your entire site to be flagged for removal from the search engine.

If you have these URL redirects in place, it is highly recommended to remove them immediately. If you created these redirects to facilitate capturing download statistics in Google Analytics, you should consider upgrading to DSpace 5.0 or above, which is able to automatically record bitstream downloads in Google Analytics (see https://github.com/DSpace/DSpace/issues/5454) without the need for any URL redirects.

Turn OFF any generation of PDF cover pages

While DSpace offers a PDF Citation Cover Page option, this option may affect your content's visibility in search engines like Google Scholar. Google Scholar (and possibly other search engines) specifically extracts metadata by analyzing the contents of the first page of a PDF. Dynamically inserting a custom cover page can break the metadata extraction techniques of Google Scholar and may result in all or much of your site being dropped from the Google Scholar search engine.

For more information, please see the "Indexing Repositories: Pitfalls and Best Practices" talk from Anurag Acharya (co-creator of Google Scholar) presented at the Open Repositories 2015 conference.

In general, OAI-PMH is not useful to Search Engines

Feel free to support OAI-PMH, but be aware that in general it is not useful for search engines:

- No reliable way to determine OAI-PMH base URL for a DSpace site.
- No standard or predictable way to get to item display page or full text from an OAI-PMH record, making effective indexing and presenting meaningful results difficult.
- In most cases provides only access to simple Dublin Core, a subset of available metadata.
- **NOTE:** Back in 2008, Google officially announced they were retiring support for OAI-PMH based Sitemaps. So, OAI-PMH will no longer help you get better indexing through Google. Instead, you should be using the DSpace 'generate-sitemaps' feature described above.

T