Advanced Search

This form allows you to perform an advanced search. You only need to fill in one field below. This can be any field. If you select "not" as your match criteria, you must select one other field.

	Any field:	contains ⬍	
AND	Title:	contains ⬍	
AND	Creator:	contains ⬍	
AND	Description:	contains ⬍	
AND	Collection:	is	All collections ⬍
AND	Mediatype:	is	All mediatypes ⬍
AND	Custom fie ⬍	contains ⬍	
AND	Custom fie ⬍	contains ⬍	
AND	Custom fie ⬍	contains ⬍	
AND	Date:	YYYY ⬍ MM ⬍ DD ⬍	
AND	Date range:	YYYY ⬍ MM ⬍ DD ⬍ TO YYYY ⬍ MM ⬍ DD ⬍	

Search

Advanced Search returning JSON, XML, and more

This will return results in the format of your choice.

Query:

Fields to return (pick one or more):

(optional) Sort results by:

avg_rating
backup_location
btih
call_number
collection
contributor
coverage
creator
date
description
downloads
external-identifier
foldoutcount
format
genre
headerImage
identifier
imagecount
indexflag
language
licenseurl
mediatype
members
month
name
noindex
num_reviews
oai_updatedate
publicdate
publisher
related-external-id
reviewdate
rights
scanningcentre
source
stripped_tags
subject
title
type
volume
week
year

Number of results: 50

Page: 1

JSON format:	●
XML format: save to file: ☑	○
HTML table:	○
CSV format: (show/hide help)	○
RSS format: ("Fields to return" ignored)	○

Search

Help with CSV and Excel (show/hide help)

Instructions for using the advanced search engine to create reports for items pertaining to books...

Notes

- You can try constructing a search query and hitting "Search" in the top "Advanced Search" section and then coming "back" to this section to copy that query into the "Query:" box.
- "downloads" is the number of downloads for content in an item
- "month" is the number of downloads in the last 30 days
- "week" is the number of downloads in the last 7 days
- We have a **known bug** in our site **with using " TO " in things like titles/descriptions**. If you run into a problem and need to use " TO ", use it lowercase and you will avoid this bug.

Dates and ranges

The following dates can be used for range queries:

- addeddate
- createdate
- date
- indexdate
- publicdate
- reviewdate
- updatedate

The special date field, "oai_updatedate", can be used to mean all of the following dates combined:

- addeddate
- indexdate
- publicdate
- reviewdate
- updatedate

The rationale for "oai_updatedate" date merging is to allow something like the Open Archives Initiative protocol get time-sortable lists of updated or added items.

- addededate -- should be time the item was initially added to archive
- indexdate -- should be last time item had a change that updated our search engine
- publicdate -- should be, for *most* items, the time after its first derive is done (ie: has "public formats" for a /details/ page)
- reviewdate -- updated to be the most recent time of all reviews (updated on new reviews and review changes)
- updatedate -- mostly the time of the last /editxml/ page submit for an item

Example date ranges: `updatedate:[2007 TO 2008]` `createdate:[2007-02-01 TO 2007-02-11]`

For dates like "sponsordate" that are not able to do ranges, they can still use * wildcards, for example: `sponsordate:200802*`

Example queries

```
publicdate:[2008-02-01 TO 2008-03-01] AND contributor:smithsonian
sponsordate:200802* AND mediatype:texts  indexdate:[2008-02-01T00:12:00Z TO 2008-03-01T00:23:59Z]
```

Explanation of Terms, Operators, Queries, and Grouping

Here, you will find an explanation on how the syntax to the search engine works.

Our site uses the **Apache Lucene** opensource search engine library and uses its lucene query syntax. A briefer explanation of the syntax follows.

Terms:
The search engine supports two types of terms, single terms and phrases.
A single term is a single word such as "test" or "hello".
A phrase is a group of words such as "duck and cover".

Field Specific queries:
If you want to restrict your search to a specific part of the metadata, you can append in front the term a Field Shortcut, such as:

```
title:"Duck and Cover"
```

will look for "Duck and Cover" only in the title of the items. For more information on Field Shortcuts, see section about Field Shortcuts Expansion
NOTE: searching in only a restricted set is better than using the default set for two reasons:

1. it’s much faster
2. the results are much more relevant

Boolean Operators:
By default, the operator is AND, which means that the query:

```
test hello
```

is exactly similar to test AND hello, which means that documents containing the term "test" AND the term "hello" will be returned.
The query:

```
test OR hello
```

will return documents either containing the term "test" OR the term "hello". The query:

```
test AND NOT hello
```

will return documents that contain the term "test" but not the term "hello" NOTE: using the NOT operator, be sure to supply at least one term without the NOT, or search will fail (the query NOT test is not valid).

Range Queries:

range queries allow you to search for documents whose field match values in between two bounds, for example:

```
downloads:[1000 TO 2000]
```

will return items whose download count is between 1000 and 2000. using [] will include the bounds in the search, and () will exclude the bounds from the search If you don’t want to specify a bound, you can use the special keyword "null":

```
downloads:[10000 TO null]
```

will search for items with more that 10000 downloads.

Boosting a term:

Boosting allows you to control the relevance of a document by boosting its term. For example, if you are searching for war gulf and you want the term "gulf" to be more relevant boost it using the ^ symbol along with the boost factor next to the term. You would type:

```
war gulf^4
```

This will make documents with the term gulf appear more relevant.

Grouping:

The search engine supports using parentheses to group clauses to form subqueries, for example:

```
(war OR gulf) AND tank
```

you can also use this feature within a field:

```
title:(war OR gulf)
```

Fuzzy Queries:

If you are not sure how to spell a word you can ask the search engine to try to figure it out for you. To do this, append a ~ after the term to apply it on, for example:

```
buttonwood~
```

will find the words buttonwood, as well as cottonwood and buttonware. NOTE: this query involves a lot of computing, so use it wisely and expect to wait a little for results to come in!