HowToSearchLucene

These are my notes on our practical experiences of getting Lucene search working with Lenya. Use at your own risk!

Introduction

If you haven't already, take a look at the Lucene documentation at http://lenya.apache.org/docs/1_2_x/components/search/lucene.html and http://lenya.apache.org/docs/1_2_x/how-to/search.html.

I followed this and initially search seemed to be working only to fail with .search(): EXCEPTION: java.lang.IndexOutOfBoundsException: Index: 111, Size: 7 (after re-crawling / indexing multiple times - I'd added the Ant crawl / index tasks to our crontab hourly).

The .search(): EXCEPTION: java.lang.IndexOutOfBoundsException: Index: 111, Size: 7 et al. errors

This occurs if you configure searching for the live area but then try to test it out whilst in the authoring area! We don't have search configured for the authoring area, but presumably if you have this error shouldn't occur.

Overview

Lucene needs indexes to be generated for search to function. Ant (a Java-based build tool similar in function to "make" which Lenya itself uses too) is used to:

- Crawl our live site (usually), i.e., retrieve all our documents as seen by the end-user.
- Generate indexes from the crawled documents for Lucene to use.

The Ant commands to do this are:

- ant -f {path1}/crawl_and_index.xml -Dcrawler.xconf={path2}/crawler-live.xconf crawl
- ant -f {path1}/crawl_and_index.xml -Dcrawler.xconf={path2}/lucene-live.xconf index
where
- {path1} is the full path to {pub}/../../bin
- {path2} is the full path to {pub}/config/search directory.
(the full paths are long so I've left them out for clarity - see the Lucene documentation for them).

The crawl_and_index.xml file

This is an Ant project (i.e., "Makefile") file. It's fairly self-explanatory and not of much interest (unless you want to extract text from pdfs - n.b., it's probably not a good idea to activate the pdf indexing based on Adobe's live conversion page!). You can add new targets here and add appropriate ant commands to call them.

The relevent targets ("Make rules") are:

- init (reads search.properties and local.search.properties files to set up paths and sets up java class / lib paths)
- crawl
- index
- indexOneDocument
- search
- various other (some unimplemented) targets to e.g., extract text from pdfs

The only arguments the crawl and index targets use are the crawler.xconf and lucene.xconf properties (i.e., variables) which will be the paths to our crawler-live.xconf and lucene-live.xconf files respectively (since we specify them on the Ant command line).

The search.properties file

This currently just sets the webapp.dir.

The crawler-live.xconf file

The only relevant lines are:

```
<base-url href="http://127.0.0.1/default/live/index.html"/>
```

This is your publication's live home page. I've removed the :8888 port number as we use port 80 instead.

```
<scope-url href="http://127.0.0.1/default/live/"/>
```

This limits where the crawler will crawl.

```
<uri-list src="../../work/search/lucene/uris.txt"/>
```

The pathname of the list of documents crawled. (i.e., {pub}/work/search/lucene/uris.txt).

```
<htdocs-dump-dir src="../../work/search/lucene/htdocs_dump"/>
```

The pathname of the directory where the actual documents once crawled are stored. (i.e., {pub}/work/search/lucene/htdocs_dump).

It can be useful to inspect uris.txt to make sure that only the documents you want indexed are being crawled. N.b., Remember that we're only crawling documents in the live area so don't forget to make sure your documents have actually been published first.

If all went well, after executing the crawl target, you should be able to see the uris crawled in uris.txt and the crawled documents will be in htdocs_dump/{pubname}/{area}/, e.g., in htdocs_dump/default/live.

The lucene-live.xconf file

I'm not sure why the following line is:

```
<htdocs-dump-dir src="../../content/live"/>
```

rather than:

```
<htdocs-dump-dir src="../../work/search/lucene/htdocs_dump/default/live"/>
```

The robots.txt file

Disallow any documents youbuild/lenya/webapp/lenya/content/search don't want indexed here. N.b., Lucene seems not to dump documents if it can't match the specifically configured fields (presumably those in lenyadocs.xconf?). This is handy as it e.g., ignores our newsfeed by default.

Extraneous white space characters in search result links

Having successfully got search working and figured out why the Java errors were occurring, we were then presented with search result links with extraneous white space (spaces and %0A newlines). These are coming from the select in the uri template in search-and-results.xsl.

The (global) search-and-results.xsp in build/lenya/webapp/lenya/content/search (the publication search-and-results.xsp seems to be ignored) generates the uri element. For some reason, the parent, filename and querystring attributes are being selected as blank (this is where the whitespace comes from). The lurl content is selected OK.

Fixed by changing

```
<xsl:value-of select=".">
```

to

```
<xsl:value-of select="normalize-space(.)">
```

No excerpt available problem / links wrong

Add

```
<map:transform src="pubs/default/lenya/xslt/search/searchfixer.xsl"/>
```

to src/webapp/lenya/lucene.xmap after

```
<map:transform src="xslt/search/sort.xsl"/>
```

In searchfixer.xsl, change the uri template to:

```
<xsl:template match="uri">
 <uri><xsl:value-of select="@parent"/>_<xsl:value-of select="../fields/language"/>.html</uri>
</xsl:template>
```

The lenyadocs.xconf file

A cursory look at this file seems to indicate that you can tell Lucene exactly which elements to index in your documents. This enables you to base searches on e.g., just title, just body text, just Dublin Core or whatever. Coupled with Ant targets aimed at, say, image metadata, very complex searching is possible.