# Making a Collection Searchable

We're treating the search system as a part of Fedora. Each Fedora server has the associated search system on the same machine. However, an index may be copied between servers for testing (e.g., copy the production index to the development server to test new SRU features).

## Primary Process

1. Ingest a sample object into Fedora.
2. Update gSearch config:
    a. Check that all fields are being mapped properly into the Lucene index. This is controlled by *CVS*/infrastructure/fedoragsearch /foxmlToLucene.xslt. You can test by tracking down the actual FOXML file from Fedora, or by actually indexing the object (see below).
    b. You most likely want to update the list of fields that make up the keyword index.
    c. When you're done making changes, commit to CVS, and upload a copy to the working location: *fedora_tomcat_dir*/webapps /fedoragsearch/WEB-INF/classes/config/index/Lucene.
3. Ensure that an empty directory exists for the Lucene index, /fedora/gsearchIndex-test
4. Re-generate the Lucene index. From *fedora_dir*/webapps/fedoragsearch/client/bin, run these commands, checking the output index (with Luke) at each step:
    a. fgsoperations fedora.dlib.indiana.edu:9090 updateIndex createEmpty
    b. fgsoperations fedora.dlib.indiana.edu:9090 updateIndex fromPid *sample_pid*
    c. fgsoperations fedora.dlib.indiana.edu:9090 updateIndex fromFoxmlFiles
5. From the fedora account's home directory, run the **horrible hack** IndexUpdater. There is currently a bug in gsearch that prevents it from adding XML to fields in the Lucene index. IndexUpdater is a separate process that runs to add these fields to the index. We will want to address this problem as soon as the gsearch source code is realeased.
    - java -cp /usr/local/tomcat/webapps/SRW/WEB-INF/lib/lucene-core-1.9.1.jar:. IndexUpdater
6. You should now have a working index in /fedora/gsearchIndex-test. You can use this to replace the existing index in /fedora/gsearchIndex-dev, and test it with the SRU-dev server. (Change the config of the SRU Server as needed.)
7. Once you're certain the new index works properly, you can use it to replace the primary index /fedora/gsearchIndex.

## To update the search index for a single collection

This is a lousy process, and there must be a better way to do it, but at least it works:

1. Run the collection's listMembers disseminator, ensuring that the value for *max* is larger than the size of the collection.
2. Save the output XML file.
3. Parse the XML file to obtain a list of PIDs.
4. Transfer the list of PIDs to the machine where indexing is done. Run a command like this to generate a list of re-indexing commands:
```
for line in `cat collectionMembers.xml`; do echo "/usr/local/fedora/server/jakarta-tomcat-5.0.28/webapps
/fedoragsearch/client/bin/fgsoperations rhyme.dlib.indiana.edu:9090 updateIndex fromPid $line"; done
>script.sh
```
5. Make the script file executable (and add `#! /bin/sh` to the top).
6. Run the script.
7. Run the IndexUpdater from above. Note that IndexUpdater still works on only a full index, but it is possible to merge multiple Lucene indexes.
8. You should now have a working index in /fedora/gsearchIndex-test. You can use this to replace the existing index in /fedora/gsearchIndex-dev, and test it with the SRU-dev server. (Change the config of the SRU Server as needed.)
9. Once you're certain the new index works properly, you can use it to replace the primary index /fedora/gsearchIndex.

## Search system 2.2

We are attempting to take advantage of the automatic indexing allowed by Fedora 2.2. Here's the current status:

- Fedora 2.2 is running on development and production machines
- The latest version of fedoragsearch is installed on the development machine.

Problem: Indexing full MODS/DC documents.

- We need to remove usage of IndexUpdater.
- The way fedoragsearch performs indexing doesn't lend itself to simply embedding MODS/DC within an IndexField element
    - The parser would need some rewrites to pick up this content
    - Namespaces would be lost
- The entire MODS/DC document can be stored within a CDATA section
    - The foxmlToLucene.xsl file has been updated to do this
    - Unfortunately, namespace declarations at the top of the XSLT file aren't placed in the embedded MODS/DC, because the parser thinks they already exist (it doesn't know it is outputting into CDATA).
        - Therefore, the needed namespace is placed explicitly, which causes lower-level elements to contain unneeded namespaces.
- The SRU server can handle either escaped MODS/DC documents (indexed by fedoragsearch with a CDATA) or unescaped documents (indexed by IndexUpdater).