

Solr Indexing Quick Guide

This guide will help you get up and running with a Fedora 4 instance whose updates are automatically indexed in a [Solr](#) repository. This guide glosses over many of the details and should be considered a starting point for testing this feature. The document assumes a [POSIX](#) operating system with [cURL](#), a text editor, [Java](#), [Git](#), and a download of [Apache Solr 4.6.0](#).

Install and Start Fedora 4

This guide assumes Fedora 4 is running on port 8080 (with JMS listening on port 61616).

Install, Configure and Start Solr

```
wget http://mirror.cogentco.com/pub/apache/lucene/solr/4.6.0/solr-4.6.0.tgz
tar -xzf solr-4.6.0.tgz
```

Edit `solr-4.6.0/example/solr/collection1/conf/solrconfig.xml`, un-commenting the `schemaFactory` element in lines 134-37, and commenting out the `schemaFactory` element in line 151, as shown below:

`solrconfig.xml`

```
<!-- To enable dynamic schema REST APIs, use the following for <schemaFactory>: -->

<schemaFactory class="ManagedIndexSchemaFactory">
  <bool name="mutable">true</bool>
  <str name="managedSchemaResourceName">managed-schema</str>
</schemaFactory>

<!-- When ManagedIndexSchemaFactory is specified, Solr will load the schema from
the resource named in 'managedSchemaResourceName', rather than from schema.xml.
Note that the managed schema resource CANNOT be named schema.xml. If the managed
schema does not exist, Solr will create it after reading schema.xml, then rename
'schema.xml' to 'schema.xml.bak'.

Do NOT hand edit the managed schema - external modifications will be ignored and
overwritten as a result of schema modification REST API calls.
When ManagedIndexSchemaFactory is specified with mutable = true, schema
modification REST API calls will be allowed; otherwise, error responses will be
sent back for these requests.

-->
<!-- <schemaFactory class="ClassicIndexSchemaFactory"/> -->
```

The `fcrepo-message-consumer SolrIndexer` implementation does not commit upon updates. In order to see the changes, you must configure Solr to have a `commit` strategy that is appropriate for your use. Resource removal events **do** trigger a commit.

solrconfig.xml

```
<!-- AutoCommit

    Perform a hard commit automatically under certain conditions.
    Instead of enabling autoCommit, consider using "commitWithin"
    when adding documents.

    http://wiki.apache.org/solr/UpdateXmlMessages

    maxDocs - Maximum number of documents to add since the last
              commit before automatically triggering a new commit.

    maxTime - Maximum amount of time in ms that is allowed to pass
              since a document was added before automatically
              triggering a new commit.
    openSearcher - if false, the commit causes recent index changes
                  to be flushed to stable storage, but does not cause a new
                  searcher to be opened to make those changes visible.

    If the updateLog is enabled, then it's highly recommended to
    have some sort of hard autoCommit to limit the log size.
-->
<autoCommit>
  <maxTime>${solr.autoCommit.maxTime:15000}</maxTime>
  <openSearcher>false</openSearcher>
</autoCommit>
```

Start Solr and verify that it is running at <http://localhost:8983/solr>.

```
cd solr-4.6.0/example
java -jar start.jar
```

Add the "uuid" field ("title" and "id" already exist).

```
curl -X POST -H "Content-Type: application/json" --data-binary "@solr-fields.json" "http://localhost:8983/solr/schema/fields"
```

solr-fields.json

```
[{"name": "uuid", "type": "text_general", "stored": "true", "indexed": "true"}]
```

Download, Build, Configure and Start fcrepo-message-consumer

```
git clone git@github.com:fcrepo4/fcrepo-message-consumer.git
```

Edit the configuration at `fcrepo-message-consumer/fcrepo-message-consumer-webapp/src/main/resources/spring/indexer-core.xml` to point to your Solr installation.

```

<!-- Solr Indexer START-->
<bean id="solrIndexer" class="org.fcrepo.indexer.solr.SolrIndexer">
  <constructor-arg ref="solrServer" />
</bean>
<!--External Solr Server -->
<bean id="solrServer" class="org.apache.solr.client.solrj.impl.HttpSolrServer">
  <constructor-arg index="0" value="http://${fcrepo.host:localhost}:${solrIndexer.port:8983}/solr/" />
</bean>
<!-- Solr Indexer END-->

<!-- Message Driven POJO (MDP) that manages individual indexers -->
<bean id="indexerGroup" class="org.fcrepo.indexer.IndexerGroup">
  <constructor-arg name="indexers">
    <set>
      <!--
        <ref bean="jcrXmlPersist"/>
        <ref bean="fileSerializer"/>
        <ref bean="sparqlUpdate"/> -->
      <!--To enable solr Indexer, please uncomment line below -->
      <ref bean="solrIndexer"/>
    </set>
  </constructor-arg>

  <!-- If your Fedora instance requires authentication, enter the
    credentials here. Leave blank if your repo is open. -->
  <constructor-arg name="fedoraUsername" value="${fcrepo.username:}" /> <!-- i.e., manager, tomcat, etc. -->
  <constructor-arg name="fedoraPassword" value="${fcrepo.password:}" />
</bean>

```

Start the application (in this case on port 9999).

```

mvn clean install -DskipTests
cd fcrepo-message-consumer-webapp
mvn -Djetty.port=9999 jetty:run

```

Create an Indexable resource

```

curl -v -X PUT -H "Content-Type: text/turtle" --data-binary "@object.rdf" "http://localhost:8080/rest/indexable0bject"

```

object.rdf

```

PREFIX dc: <http://purl.org/dc/elements/1.1/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX indexing: <http://fedora.info/definitions/v4/indexing#>

<> indexing:hasIndexingTransformation "default"; rdf:type indexing:Indexable; dc:title "This title will show up in the index."

```

Ensure that the records are committed to Solr (either through an explicit commit or waiting until the configured commit period is up) and then verify that they show up.