UpgradingTikaInSolr

Idiot's Guide to Upgrading Tika in Apache Solr

This is written as a relative outsider to Apache Solr development. It will appear painfully rudimentary to devs on Solr, and it assumes less-than-black-belt-familiarity with git...so be it.

Many thanks to Uwe Schindler for walking me through these steps several times.

Prereqs

1. python3 must be installed and callable as python3
2. perl must be installed
3. ivy must be installed: `ant ivy-bootstrap`
4. Fork lucene-solr and clone your fork.
5. Hint: if you're using Intellij, run `ant idea` **before** opening the project in Intellij
6. Getting Solr built the first time can take a long time.
7. Check out your local fork
8. Make sure you're up to date
    a. `git remote add upstream https://github.com/apache/lucene-solr.git`
    b. `git fetch upstream`
    c. `git pull upstream master`
    d. `git push master`

Phase 1

1. create new branch, e.g `jira/solr-11701`
2. `mvn dependency:tree` on the newly released Apache Tika and **MEMORIZE** it
3. upgrade all dependencies in `lucene/ivy-versions.properties` – make sure that they are in alphabetical order
4. add any new licenses in `solr/licenses` – must include a -LICENSE-XYZ.txt and -NOTICE.txt file for every jar
5. update anything new in `solr/contrib/extraction/ivy.xml`
6. `ant clean` (out of nervous habit) and then run the unit tests in `contrib/dataimporthandler-extras` and `contrib/extraction`
7. Fix any problems in the source code, and this can include `XLSXResponseWriter` which relies on Apache POI.
8. `ant clean-jars jar-checksums`
9. `git add` new .sha1 files in solr/licenses and lucene/licenses and `git rm` old .sha1 files
10. `ant precommit`
11. Receive immediate errors that you missed something and go back two steps; repeat `ant precommit` as needed, waiting 15-20 minutes each time ... if you didn't break something obvious.
12. In my environment, `ant precommit` eventually ends in errors about broken links in html. This means you are successful!!!
13. Run `ant test` for kicks. Something will likely break. Try to figure out if it is caused by anything you did or just a flaky build. Bonus points if the test failure is reproducible and you report it/fix it.

Phase 2: Integration Testing Solr

To test that you've gotten most of the dependencies right, why not run DIH on Tika's test documents?

1. Build the Solr dist: `cd solr/` and `ant package`
2. Unzip your shiny new Solr and create a collection from: https://github.com/tballison/tika-addons/tree/main/solr-tika-integration/src/configs
3. `bin\solr start`
4. Copy the files from tika-parsers/src/test/resources/test-documents ... make sure to remove ucar files: *.nc, *.hdf, *.fb2, *.he5 – these wreak havoc with the data importer
5. Navigate to the Solr admin window->`Dataimport`.
6. Close your eyes, cross your fingers, pray to your appropriate diet(y|ies) or not, and press `Execute`
7. Watch the command window to see if there were any catastrophic missing class problems
8. Go to logs to see if there are any show stoppers for exceptions.
9. When this completes, go to `Query` and check how many documents are actually indexed
10. Compare the number of documents in Solr to the number you'd get if you ran `java -jar tika-app.jar -i <input_dir> -o <output_dir>`

In addition to DIH, the above configs are also set up to work with the ExtractingHandler.

You can run either the SolrJ client (https://github.com/tballison/tika-addons/blob/main/solr-tika-integration/src/main/java/org/tallison/indexers/SolrJIndexer.java) or the

Curl wrapper https://github.com/tballison/tika-addons/blob/main/solr-tika-integration/src/main/java/org/tallison/indexers/CurlIndexer.java

Make sure to set the source directory appropriately and the solr-collection name correctly for your test files and Solr collection. Note that these indexers do not process files recursively.

Phase 3: Submit a Pull Request

1. When everything looks good, commit your changes and submit a PR

Phase 4: Reflect, Rejoice, Work

1. Reflect on:
    a. The tedium to get the dependencies right and the risks of not getting them right
    b. The ever present risks of jar hell by integrating Tika into Solr
    c. The seductive belief that Tika won't break Solr, when we know it will eventually, and we should really be keeping Tika out of Solr if at all possible...and yet maintain the awesome easy-to-get-started-ness of the current integration.
2. Rejoice that Tika is being refactored out of Solr in 9x.
3. Work towards whatever solution allows for an easy, out of the box extraction process for binary files