DeveloperTips

Tips for developers

Dependencies

Building Lucene requires that you have Apache Ant installed. Starting with 3.6, Apache Ivy is also required for fetching dependency jars to compile against.

Once you have ant working, you can use the "ant ivy-bootstrap" build target to have ivy automatically downloaded and installed for you in your ${user.home}/.ant/lib directory

Tips to configure IDEs

- For Lucene/Solr 9.0 and later, execute "gradlew tasks". The section "IDE tasks" will show you the tasks that can be used to open the project
- For Lucene/Solr 8.0 and earlier, "ant idea" or "ant eclipse" will create the project structure than be opened in the appropriate IDE

The lucene-dev user's list and slack solr-dev channel are good places to get further help.

Contribution guidelines

Before you start, send a message to the developer mailing list (Note: you have to subscribe before you can post), or file a bug report in Jira. Describe your proposed changes and check that they fit in with what others are doing and have planned for the project. Be patient, it may take folks a while to understand your requirements.

Modify the source code and add some (very) nice features using your favorite IDE.

But take care about the following points:

- Code compatibility:
  - All code to be included in 3.X releases should be compatible with Java 5.
  - All code to be included in 4.0 to 4.7.x releases should be compatible with Java 6.
  - All code to be included in 4.8 onwards and 5.x releases should be compatible with Java 7.
  - All code to be included in 6.x releases and on master branch should be compatible with Java 8.
- All public classes and methods should have informative Javadoc comments.
- Code should be formatted according to Sun's conventions with one exception:
  - indent two spaces per level, not four.
- Contributions should pass existing unit tests.
- New unit tests should be provided to demonstrate bugs and fixes (http://www.junit.org). Please make your test cases extend LuceneTestCase, which has additional checks including thread leaks for example and makes sure that assertions are enabled.
- The java source code is in the directory lucene/src/java and the java test code is in the directory lucene/src/test.

Please **do not**:

- reformat code unrelated to the bug being fixed: formatting changes should be separate patches/commits.
- comment out code that is now obsolete: just remove it.
- insert comments around each change, marking the change: folks can use subversion to figure out what's changed and by whom.
- make things public which are not required by end users.
- Add @author tags to your code. (Instead, give yourself credit in the CHANGES.txt file.)

Please **do**:

- try to adhere to the coding style of files you edit;
- comment code whose function or rationale is not obvious;
- update documentation (e.g., *package.html* files, this wiki, etc.)

Please use the *LUCENE-NNNN.patch* naming pattern when creating patches for uploading to JIRA. Once you create a new JIRA issue, note its name and use that name when naming your patch file. For example, if you are creating a patch for a JIRA issue named LUCENE-3456, then name your patch filename LUCENE-3456.patch. If you are creating a new version of an existing patch, use the existing patch's file name. JIRA will automatically "gray out" the old patch and clearly mark your newly uploaded patch as the latest.

Problems with Ivy?

Some people have reported problems getting Ivy to work when installed in ${user.home}/.ant/lib

The majority of these issues seem to be people using installations of ant packaged by linux disto providers that either do not provide a full ant install (ie: additional RPMs are required) or explicitly disable the ability to use "user lib" feature of ant in their configs, or have some other problems.

Suggested fixes for people who still have problems getting Ivy to work after running "ant ivy-bootstrap"...

- try using the "--noconfig" option when running ant
- try adding "rpm_mode=false" to your ~/.ant/ant.conf
- try removing "rpm_mode=true" from your /etc/ant.conf
- try installing a clean version of Apache Ant downloaded directly from the Apache mirrors
- if you installed ant from distribution packages (yum, apt, etc...) then try installing ivy from the same.

For more details:

- https://issues.apache.org/jira/browse/LUCENE-3946
- http://ant.apache.org/manual/running.html
- https://bugzilla.redhat.com/show_bug.cgi?id=810067
- https://bugzilla.redhat.com/show_bug.cgi?id=490542

Pre-commit check

Committers run this check before they commit something. But you can help them out by running it yourself first and fixing anything it catches.

Before constructing your patch, please run the top-level pre-commit check, which finds problems like tabs and @author tags in source files, broken links in javadocs, files not controlled by Subversion (a.k.a. "unversioned files"), etc.

To run the pre-commit checks from ant, run the following from the top-level directory – the directory containing `lucene/` and `solr/` – in your working copy:

```
ant precommit
```

Other Helpful Resources

The following resources may prove helpful when developing Lucene contributions. (These are not an endorsement of any specific development tools)

- Eclipse Galileo codestyle.xml file for Lucene's coding conventions
- Setting up the .classpath - copy the contents of eclipse.lucene.classpath file to your project's .classpath file. Note that you may need to change the location of ant.jar.
- IntelliJ, Eclipse, Maven: information for both Lucene and SOLR can be found at SOLR How To Contribute
- Additional information on running tests.