CommitterInfo

General Information for Committers

This wiki is for collection useful information for Solr Committers

Guides

- http://wiki.apache.org/lucene-java/ReleaseTodo
- Website_Update_HOWTO
- CommitPolicy
- NightlyBuilds

Things to Double Check Before Any Commit

1. Did you add update CHANGES.txt ?
2. Do the tests all pass ?
3. Are there any warnings from "ant javadoc" ?
4. Does this commit affect anything in the tutorial ? ... If so, did you update the tutorial ?
5. Did you change anything in `src/site` ? ... If so did you follow the steps in "Website_Update_HOWTO" so this commit includes changes to the published version of the site in `site/` ?
6. See also http://wiki.apache.org/lucene-java/PatchCheckList

Suggested Subversion Configuration Options

These are some recomended options for your ~/.subversion/config file...

```
[miscellany]
enable-auto-props = yes
[auto-props]
KEYS = svn:eol-style=native
README = svn:eol-style=native;svn:keywords=Date Author Id Revision HeadURL
INSTALL = svn:eol-style=native;svn:keywords=Date Author Id Revision HeadURL
*.java = svn:eol-style=native;svn:keywords=Date Author Id Revision HeadURL
*.html = svn:eol-style=native;svn:keywords=Date Author Id Revision HeadURL
*.xml = svn:eol-style=native;svn:keywords=Date Author Id Revision HeadURL
*.jsp = svn:eol-style=native;svn:keywords=Date Author Id Revision HeadURL
*.xsl = svn:eol-style=native;svn:keywords=Date Author Id Revision HeadURL
*.xslt = svn:eol-style=native;svn:keywords=Date Author Id Revision HeadURL
*.properties = svn:eol-style=native;svn:keywords=Date Author Id Revision HeadURL
*.txt = svn:eol-style=native;svn:keywords=Date Author Id Revision HeadURL
.htaccess = svn:eol-style=native;svn:keywords=Date Author Id Revision HeadURL
*.png = svn:mime-type=image/png
*.pdf = svn:mime-type=application/pdf

# use LF for shell scrips since even cygwin wants LF
*.sh = svn:eol-style=LF;svn:keywords=Date Author Id Revision HeadURL
```

SVN Merging

See Subversion Merge on the Lucene wiki.

Generating New Analysis Factories

Anytime the Lucene jars are updated, we should attempt to generate Factories for any new Tokenizer or TokenFilter classes. To do this, run `ant stub-factories -Dstub.src.path="./src:../lucene/contrib:../lucene/src/java"` Where the "stub.src.path" property is set to be a "path like list" of java source directories that should be crawled when looking for classes to make factories for.

This task will first use reflection on all of the `dist` jars to find known subclasses of Tokenizer or TokenFilter which do not already have Factories and write this list to `build/need-stub-factories.txt`. It will then crawl the directories specified in `stub.src.path` looking for the corresponding source files to determine what the constructor parameter names are (this info is not available in the byte code) for each class, and generate a new factory for it.

Classes with no special constructor parameters will get full fledged factories. Classes with special params will get abstract "stub" factories that will not compile without human editing.

Once `ant stub-factories` is finished executing. Use `ant clean test` to check for stubs that need human intervention. Any stub factories that you want to postpone implementing can be deleted (they can always be created later). Any stub factory that should be ignored, and not attempted later, can be added to the list of exceptions in the `stub-factories` ant task definition (grep for "place to put special case classes we want to ignore")