

LexEVS Per-segment Search Implementation

Document Information

Author: Craig Stancl, Scott Bauer, Cory Endle
Email: Stancl.craig@mayo.edu, bauer.scott@mayo.edu, endle.cory@mayo.edu
Team: LexEVS
Contract: S13-500 MOD4
Client: NCI CBIIT
National Institutes of Health
US Department of Health and Human Services

Lucene Per-segment search

Per-segment searching is largely kept under the covers but since LexEVS employs a number of customizations of parameters, particularly filter customizations, many queries may break and need refactoring. At the same time a number of objects have new names, apis and other changes that will require refactoring and updating to an as yet undetermined amount. Some needed changes are documented in recent migration guides and deprecation lists:

- http://lucene.apache.org/core/2_9_4/api/core/deprecated-list.html
- https://lucene.apache.org/core/4_0_0/MIGRATE.html
- https://lucene.apache.org/core/5_0_0/MIGRATE.html

The use of IndexReader in particular will not change, but filters and queries passed into this may break or otherwise be unusable. Read this article for more information: <http://blog.thetaphi.de/2012/02/is-your-indexreader-atomic-major.html>

Updated Objects will include

- ChainedFilter replaced with BooleanFilter
- HitCollector is replaced by Collector or SimpleCollector
- MultiSearcher will likely be replaced by something like IndexSearcher(MultiReader)
- The interface Searchable has gone away and it's former child classes such as IndexSearcher will have to take it's place.
- Searcher is also gone.
- TermEnum is replaced by TermsEnum
- FieldSelector is replaced by FieldVisitor, which has a very different API.

Updated Objects that have new API's include:

- DocIdSetIterator
- IndexWriter
- IndexReader (optimizing methods can no longer be called on this.)
- TermsFilter (addTerm() method has gone away and terms are added in the constructor.)