Apache Lucene - Lucene Contrib

Table of contents

1. Lucene Contrib

The Lucene Java project also contains a workspace, Lucene Contrib (formerly known as the Lucene Sandbox), that is open both to all Lucene Java core committers and to developers whose commit rights are restricted to the Contrib workspace; these developers are referred to as "Contrib committers". The Lucene Contrib workspace hosts the following types of packages:

- Various third party contributions.
- Contributions with third party dependencies - the Lucene Java core distribution has no external runtime dependencies.
- New ideas that are intended for eventual inclusion into the Lucene Java core.

Users are free to experiment with the components developed in the Contrib workspace, but Contrib packages will not necessarily be maintained, particularly in their current state. The Lucene Java core backwards compatibility commitments (see http://wiki.apache.org/lucene-java/BackwardsCompatibility) do not necessarily extend to the packages in the Contrib workspace. See the README.txt file for each Contrib package for details. If the README.txt file does not address its backwards compatibility commitments, users should assume it does not make any compatibility commitments.

See Contrib CHANGES for changes included in the current release.

You can access the current trunk Contrib repository at http://svn.apache.org/repos/asf/lucene/java/trunk/contrib/.

1.1. analyzers

Contributed Analyzers, Tokenizers, and Filters for various uses and languages.

See analyzers javadoc

1.2. ant

Ant task to create Lucene indexes.

See ant javadoc

1.3. benchmark

The benchmark contribution contains tools for benchmarking Lucene using standard, freely available corpora.

See benchmark javadoc

1.4. collation

CollationKeyFilter/Analyzer and ICUCollationKeyFilter/Analyzer - can be used as an efficient replacement for Locale sorting and Locale range queries as well as Locale-specific normalization

See collation javadoc

1.5. db

Provides integration with Berkley DB.

See db javadoc

1.6. highlighter

A set of classes for highlighting matching terms in search results.

See highlighter javadoc

1.7. fast-vector-highlighter

An alternative set of classes for highlighting matching terms in search results that relies on stored term vectors. This highlighter can be much faster than the standard highlighter, especially on large fields.

See fast-vector-highlighter javadoc

1.8. instantiated

RAM-based index that enables much faster searching than RAMDirectory in certain situations.

See instantiated javadoc

1.9. lucli

An application that allows Lucene index manipulation from the command-line.

See lucli javadoc

1.10. memory

High-performance single-document main memory index.

See memory javadoc

1.11. misc

A variety of miscellaneous files, including QueryParsers, and other alternate Lucene class implementations and tools.

See misc javadoc

1.12. queryparser

A new Lucene query parser implementation, which matches the syntax of the core QueryParser but offers a more modular architecture to enable customization.

See queryparser javadoc

1.13. regex

Queries with additional regex matching capabilities.

See regex javadoc

1.14. remote

Classes to help use Lucene with RMI.

See remote javadoc

1.15. snowball

Pre-compiled versions of the Snowball stemmers for Lucene.

See snowball javadoc

1.16. spatial

Classes to help with efficient distance based sorting.

See spatial javadoc

1.17. spellchecker

Provides tools for spellchecking and suggestions with Lucene.

See spellchecker javadoc

1.18. surround

A QueryParser that supports the Span family of queries as well as pre and infix notation.

See surround javadoc

1.19. swing

Swing components designed to integrate with Lucene.

See swing javadoc

1.20. wikipedia

Tools for working with wikipedia content.

See wikipedia javadoc

1.21. wordnet

Tools to help utilize wordnet synonyms with Lucene

See wordnet javadoc

1.22. xml-query-parser

A QueryParser that can read queries written in an XML format.

See xml-query-parser javadoc