RunningTests

Running Tests

This page describes how to run Lucene/Solr tests and various options you can use.

Using Ant

Running the tests on trunk and branch_4x requires Ant version 1.8.2 or higher. The 3.x releases would only work with Ant version 1.7 and would not work with 1.8 or later.

Tests run using the RandomizedTesting library.

For full usage instructions, run `ant test-help`. A subset of those instructions follows here.

Basic Operations

Running all tests

To run all tests, just run `ant test`.

You can run tests at the project, product, module, or sub-module level:

- At the top-level (containing `lucene/` and `solr/`), all Lucene and Solr tests will be run.
- Under `lucene/`, all Lucene tests will be run.
- Under `lucene/core/`, only Lucene core tests will be run.
- etc.

Running only a specific test class

To run a specific test class, use `-Dtests.class`

`tests.class` filters full class names (including package) using a shell-like glob pattern.

For example, to run only `TestIndexWriter`: `ant test -Dtests.class="*.TestIndexWriter"`

(The quotations around the glob pattern prevent shell expansion.)

`-Dtestcase=Xyz` is an alias of `-Dtests.class="*.Xyz"`

Running only a specific test method

To run just a single method in a test, use `-Dtests.method`

`tests.method` filters method names using a shell-like glob pattern.

For example: `ant test -Dtests.class="*.TestIndexWriter" -Dtests.method="test*Count"`

`testmethod` is an alias of `tests.method`.

Running all tests in a specific package

To run tests in a specific package, use `-Dtests.class`

For example: `ant test -Dtests.class="*.search.*"`

Test Options

The following Lucene/Solr-specific options (not handled by the RandomizedTesting library) can be used to modify the behavior of tests.

From ant, you can set these on the command line by using `-D`.

From your IDE, you can set these by adding jvm arguments with `-D`.

variable	description	default value	example
tests.verbose	increases verbosity of tests. Use this to get (potentially very loud) test output	false	`-Dtests.verbose=true`
tests.infostream	separately toggle infostream output from indexwriter and checkindex	tests. verbose	`-Dtests.infostream=true`

tests.codec	sets the FlexibleIndexing codec for the test to use	random	`-Dtests.codec=SimpleText`
tests. postingsformat	sets the specific PostingsFormat to use for Lucene40 codec	random	`-Dtests.postingsformat=Pulsing`
tests.locale	sets the JVM default locale for a test to run under	random	`-Dtests.locale=ru_RU`
tests.timezone	sets the JVM default timezone for a test to run under	random	`-Dtests.timezone=Australia/Lindeman`
tests.directory	sets the lucene directory impl for a test to use	random	`-Dtests.directory=FSDirectory`
tests.multiplier	multiplier to increase the work done by random tests	1	`-Dtests.multiplier=3`
tests.showSuccess	show stdout/err even when test is successful	false	`-Dtests.showSuccess=true`