TestingSolr

How to run Solr unit tests

If you're going to be using or modifying Solr's source, unit tests are critical as they help uncover potential downstream runtime errors that you might encounter, and they help to isolate problems in any new functionality you're adding. The question is, how do you run Solr's unit tests? Read on below to find out.

Ant and Junit

Solr uses the Ant build system and leverages Junit as its testing harness plugin. A description of these technologies is outside the scope of this page, but we recommend you go check out each individual project as they have their share of testing tutorials. This guide is meant as a quick and dirty path to testing out Solr using what's been chosen to test it.

Running all tests

```
ant test
```

Yes, it's that simple.

Running specific unit tests

For full usage instructions, run:

```
 ant test-help
```

A subset of those instructions follows here.

Run a specific test method:

```
 ant -Dtestmethod=testEqualsLOOKUP3 -Dtestcase=TestHash clean test
```

Run all test methods in a single test class using:

```
 ant -Dtestcase=TestDistributedSearch clean test
```

Run tests inside a package and its sub-packages:

```
 ant -Dtests.class="org.apache.solr.handler.*" clean test
```

`-Dtests.class` selects full class names with a shell-like glob pattern. `-Dtestcase=XXX` is an alias for `-Dtests.class=*.XXX`

`-Dtests.method` selects method names with a shell-like glob pattern. `-Dtestmethod=XXX` is an alias for method name **prefixes**: `-Dtests.method=XXX*`

Running tests from Eclipse

Set the current working directory for the tests to src/test/test-files. In Eclipse, that can be done by going to the Run/Debug Configuration -> Arguments tab. In the Working directory select "other" and enter the path to the test-files directory.