



10 ways to reduce your tax bill

Amit Nithianandan
Senior Search Engineer
Zvents Inc.

04-15-2010

Solr Eclipse- Running Apache Solr in Eclipse.



Amit Nithianandan
Senior Search Engineer
Zvents Inc.



04-15-2010

Presentation Outline

- About Me
- Brief Intro to Solr
- Brief Intro to Eclipse
- Running Solr in Eclipse
 - What you need to accomplish this.
 - Demo to show final outcome.
- Solr + Eclipse Possibilities.
- Q&A

About Me

- BS/MS Computer Science- Virginia Tech (Go Hokies!).
- Previously, worked at Northrop Grumman TASC as Java Software Engineer.
- Currently, Senior Search Engineer at Zvents Inc.
 - We help people find fun things to do.
 - Platform powers local [event] search for a nationwide network of partners (mainly newspapers) including some large national sites such as Bing.com and YP.com.
- Experience with Solr and Lucene (mostly Solr).
 - Zvents search is powered by Solr.
- Authored a few articles for Lucid Imagination, one of which I am presenting today.

Introduction to Solr

- Open source Java based search platform.
 - Blazing fast due to caching at various levels (among other reasons).
 - Uses Lucene at its core.
 - Supports REST like API with various output formats (XML, JSON, Ruby Hash etc).
 - Customizable via configs and plugin architecture.
- Supports full text search, highlighting, clustering, faceted search etc.
- Can import data from a variety of sources (RDBMS, RSS, PDF – via Tika, etc).

Introduction to Eclipse

- Free, open source Java based IDE.
 - Plugin based built using OSGI.
- Consequently, IDE is extremely feature rich supporting many languages, technologies.
 - Supports rich code complete, refactoring, version control tools, code searching, method call tree viewing etc.
 - Can install plugins to deploy to web containers, query databases, graphically edit HTML/XML.
- Consists of perspectives, views, and editors.
 - Perspective ==> collection of views + editors arranged depending on task being completed.



Have you done this?

- Download a version of Solr and extract it.
- Read the wiki/getting started documentation and tutorial
- `cd example/`
- `java -jar start.jar`
- Visit <http://localhost:8983/solr> and start playing around.

Enter Eclipse.....

- Eclipse is great for developing Java (and other language) applications
- Not only can you develop and debug standard Java applications but you can also do the same for web based ones with the help of a few plugins.
- Now your entire search engine project for your company/client can reside within Eclipse making it easy to develop, test, debug, and share with others via version control, all within one IDE.

Pre-Requisites.

- Latest Eclipse
 - For those new to Eclipse, I recommend getting the **smallest** distribution possible for your platform and then add plugins as needed else it may consume lots of memory.
 - 64 bit Windows users, there is a distribution but it's not obvious from the main Eclipse downloads page.
 - Visit <http://download.eclipse.org/eclipse/downloads/>
- Eclipse SVN plugin, Subclipse (currently 1.6.x)
- Eclipse Jetty plugin, RunJettyRun

Basic Steps

- Checkout trunk (
<http://svn.apache.org/repos/asf/lucene/dev/trunk>
)
 - Looks like it now contains both Solr and Lucene in one place which is awesome!
- Download .classpath and .project files from http://filebox.vt.edu/users/anithian/solr_lucene_eclipse.zip
 - Place these files in the folder you just checked out. Refresh your Eclipse Project.
- Setup a new RunJettyRun runtime configuration for Solr and Run!

Demo

The screenshot displays the Eclipse IDE interface with the 'Run Configurations' dialog box open. The dialog is titled 'Create, manage, and run configurations' and is used to launch a web application with Jetty. The configuration is named 'solr-lucene-trunk'.

Run Configurations Dialog Details:

- Name:** solr-lucene-trunk
- Project:** solr-lucene-trunk
- Ports:** HTTP 8983, HTTPS (empty)
- Keystore:** C:\Users\ANithian\.keystore
- Password:** changeit, **Key Password:** changeit
- Web Application Context:** /solr
- WebApp dir:** solr/src/webapp/web

The background shows the Package Explorer with the project structure for 'solr-lucene-trunk' and the Console window at the bottom.

Recap

- Pulled the latest trunk of Solr/Lucene
- Used Eclipse to setup the build paths
- Used RunJettyRun to execute Solr from within Eclipse.
- Launched in Debug Mode to step through some code to inspect variables.

Endless Possibilities...

- This is just a basic integration of Solr and Eclipse from a trunk development/debugging perspective.
 - Read my blog post or article on LucidImagination for running stable, release Solr in Eclipse
- To really integrate Solr/Lucene with Eclipse:
 - New Solr Project Wizard → Creates folder structure with basic solrconfig and schema files
 - Custom solrconfig/schema editor UI?
 - Luke/Eclipse plugin to inspect Lucene index from Eclipse?

Back to those tax saving tips...

Eat Free
Pizza! :-)



Questions?
(Maybe Answers)

Resources

- Eclipse
 - www.eclipse.org
 - <http://download.eclipse.org/eclipse/download>
- Subclipse
 - <http://subclipse.tigris.org/>
- RunJettyRun
 - <http://code.google.com/p/run-jetty-run/>
- Apache Solr
 - <http://lucene.apache.org/solr>
- My Blog for detailed screenshots
 - <http://hokiesuns.blogspot.com>