RunNutchInEclipse0.9

Run Nutch In Eclipse on Linux and Windows nutch version 0.9

This is a work in progress. If you find errors or would like to improve this page, just create an account [UserPreferences] and start editing this page 🙂

Tested with

- Nutch release 0.9 and 1.0
- Eclipse 3.3 - aka Europa
- Java 1.6
- Ubuntu (should work on most platforms though)
- Windows XP

Before you start

Setting up Nutch to run into Eclipse can be tricky, and most of the time you are much faster if you edit Nutch in Eclipse but run the scripts from the command line (my 2 cents). However, it's very useful to be able to debug Nutch in Eclipse. But again you might be quicker by looking at the logs (logs /hadoop.log)...

Steps

For Windows Users

If you are running Windows (tested on Windows XP) you must first install cygwin

Download cygwin from http://www.cygwin.com/setup.exe

Install cygwin and set PATH variable for it.

It's in control panel, system, advanced tab, environment variables and edit/add PATH

I have in PATH like:

C:\Sun\SDK\bin;C:\cygwin\bin

If you run "bash" in Start->RUN->cmd.exe it should work.

Then you should install tools from Microsoft website (adding 'whoami' command).

Example for Windows XP and sp2

http://www.microsoft.com/downloads/details.aspx?FamilyId=49AE8576-9BB9-4126-9761-BA8011FABF38&displaylang=en

Then you can follow rest of these steps

Install Nutch

- Grab a fresh release of Nutch 0.9 - http://lucene.apache.org/nutch/version_control.html
- Do not build Nutch now. Make sure you have no .project and .classpath files in the Nutch directory

Create a new java project in Eclipse

- File > New > Project > Java project > click Next
- Name the project (Nutch_Trunk for instance)
- Select "Create project from existing source" and use the location where you downloaded Nutch
- Click on Next, and wait while Eclipse is scanning the folders
- Add the folder "conf" to the classpath (third tab and then add class folder)
- Go to "Order and Export" tab, find the entry for added "conf" folder and move it to the top. It's required to make eclipse take config (nutch-default.xml, nutch-final.xml, etc.) resources from our "conf" folder not anywhere else.
- Eclipse should have guessed all the java files that must be added on your classpath. If it's not the case, add "src/java", "src/test" and all plugin "src /java" and "src/test" folders to your source folders. Also add all jars in "lib" and in the plugin lib folders to your libraries
- Set output dir to "tmp_build", create it if necessary
- DO NOT add "build" to classpath

Configure Nutch

- See the Tutorial

- Change the property "plugin.folders" to "./src/plugin" on $NUTCH_HOME/conf/nutch-defaul.xml [You can change the property to run both from command line and eclipse to "<value>./src/plugin,plugins</value>"]

- Make sure Nutch is configured correctly before testing it into Eclipse 😉
- Three sample necessary configuration files: nutch-default.xml (the value of http.agent.name and the value of plugin.folder were edited), nutch-site.xml (the value of plugin.folder was edited) and crawl-urlfilter.txt (MY.DOMAIN.NAME was edited).

Missing org.farng and com.etranslate

Eclipse will complain about some import statements in parse-mp3 and parse-rtf plugins (30 errors in my case). Because of incompatibility with the Apache license, the .jar files that define the necessary classes were not included with the source code.

Download them here:

http://nutch.cvs.sourceforge.net/nutch/nutch/src/plugin/parse-mp3/lib/

http://nutch.cvs.sourceforge.net/nutch/nutch/src/plugin/parse-rtf/lib/

Copy the jar files into src/plugin/parse-mp3/lib and src/plugin/parse-rtf/lib/ respectively. Then add the jar files to the build path (First refresh the workspace by pressing F5. Then right-click the project folder > Build Path > Configure Build Path... Then select the Libraries tab, click "Add Jars..." and then add each .jar file individually).

Build Nutch

If you setup the project correctly, Eclipse will build Nutch for you into "tmp_build". See below for problems you could run into.

Create Eclipse launcher

- Menu Run > "Run..."
- create "New" for "Java Application"
- set in Main class

```
org.apache.nutch.crawl.Crawl
```

- on tab Arguments, Program Arguments

```
urls -dir crawl -depth 3 -topN 50
```

- in VM arguments

```
-Dhadoop.log.dir=logs -Dhadoop.log.file=hadoop.log
```

- click on "Run"
- if all works, you should see Nutch getting busy at crawling 🙂

Java Heap Size problem

If you find in hadoop.log line similar to this:

```
2009-04-13 13:41:06,105 WARN  mapred.LocalJobRunner - job_local_0001
java.lang.OutOfMemoryError: Java heap space
```

You should increase amount of RAM for running applications from eclipse.

Just set it in:

Eclipse -> Window -> Preferences -> Java -> Installed JREs -> edit -> Default VM arguments

I've set mine to

```
-Xms5m -Xmx150m
```

because I have like 200MB RAM left after runnig all apps

-Xms (minimum ammount of RAM memory for running applications) -Xmx (maximum)

Debug Nutch in Eclipse

- Set breakpoints and debug a crawl
- It can be tricky to find out where to set the breakpoint, because of the Hadoop jobs. Here are a few good places to set breakpoints:

```
Fetcher [line: 371] - run
Fetcher [line: 438] - fetch
Fetcher$FetcherThread [line: 149] - run()
Generator [line: 281] - generate
Generator$Selector [line: 119] - map
OutlinkExtractor [line: 111] - getOutlinks
```

If things do not work...

Yes, Nutch and Eclipse can be a difficult companionship sometimes 😉

eclipse: Cannot create project content in workspace

The nutch source code must be out of the workspace folder. My first attempt was download the code with eclipse (svn) under my workspace. When I try to create the project using existing code, eclipse don't let me do it from source code into the workspace. I use the source code out of my workspace and it work fine.

plugin dir not found

Make sure you set your plugin.folders property correct, instead of using a relative path you can use a absolute one as well in nutch-defaults.xml or may be better in nutch-site.xml

```
<property>
  <name>plugin.folders</name>
  <value>/home/....../nutch-0.9/src/plugin</value>
```

No plugins loaded during unit tests in Eclipse

During unit testing, Eclipse ignored conf/nutch-site.xml in favor of src/test/nutch-site.xml, so you might need to add the plugin directory configuration to that file as well.

Unit tests work in eclipse but fail when running ant in the command line

Suppose your unit tests work perfectly in eclipse, but each and everyone fail when running **ant test** in the command line - including the ones you haven't modified. Check if you defined the **plugin.folders** property in hadoop-site.xml. In that case, try removing it from that file and adding it directly to nutch-site.xml

Run **ant test** again. That should have solved the problem.

If that didn't solve the problem, are you testing a plugin? If so, did you add the plugin to the list of packages in plugin\build.xml, on the test target?

classNotFound

- open the class itself, rightclick
- refresh the build dir

debugging hadoop classes

- Sometime it makes sense to also have the hadoop classes available during debugging. So, you can check out the Hadoop sources on your machine and add the sources to the hadoop-xxx.jar. Alternatively, you can:
  - Remove the hadoopXXX.jar from your classpath libraries
  - Checkout the hadoop brunch that is used within nutch
  - configure a hadoop project similar to the nutch project within your eclipse
  - add the hadoop project as a dependent project of nutch project
  - you can now also set break points within hadoop classes lik inputformat implementations etc.

Original credits: RenaudRichardet