RunNutchInEclipse1.0

Run Nutch In Eclipse on Linux and Windows nutch version 1.0

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 1.0
- Eclipse 3.3 (Europa) and 3.4 (Ganymede)
- Java 1.6
- Ubuntu (should work on most platforms though)
- Windows XP and Vista

Before you start

Setting up Nutch to run into Eclipse can be tricky, and most of the time it is 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. Sometimes examining the logs (logs/hadoop.log) is quicker to debug a problem.

Steps

For Windows Users

If you are running Windows (tested on Windows XP) you must first install cygwin. Download it from http://www.cygwin.com/setup.exe

Install cygwin and set the PATH environment variable for it. You can set it from the Control Panel, System, Advanced Tab, Environment Variables and edit /add PATH.

Example PATH:

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

If you run "bash" from the Windows command line (Start > Run... > cmd.exe) it should successfully run cygwin.

If you are running Eclipse on Vista, you will need to either give cygwin administrative privileges or turn off Vista's User Access Control (UAC). Otherwise Hadoop will likely complain that it cannot change a directory permission when you later run the crawler:

```
org.apache.hadoop.util.Shell$ExitCodeException: chmod: changing permissions of ... Permission denied
```

See this for more information about the UAC issue.

Install Nutch

- Grab a fresh release of Nutch 1.0 or download and untar the official 1.0 release.
- Do not build Nutch yet. 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 (Right-click on the project, select "properties" then "Java Build Path" tab (left menu) and then the "Libraries" tab. Click "Add Class Folder..." button, and select "conf" from the list)
- Go to "Order and Export" tab, find the entry for added "conf" folder and move it to the top (by checking it and clicking the "Top" button). This is required so Eclipse will take config (nutch-default.xml, nutch-final.xml, etc.) resources from our "conf" folder and not from somewhere else.
- Eclipse should have guessed all the Java files that must be added to your classpath. If that'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
- Click the "Source" tab and set the default output folder to "Nutch_Trunk/bin/tmp_build". (You may need to create the tmp_build folder.)
- Click the "Finish" button
- DO NOT add "build" to classpath

Configure Nutch

- See the Tutorial

- Change the property "plugin.folders" to "./src/plugin" on $NUTCH_HOME/conf/nutch-default.xml
- Make sure Nutch is configured correctly before testing it into Eclipse 😉

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. If that does not work, you may try clicking "Add External JARs" and the point to the two the directories above).

Two Errors with RTFParseFactory

If you are trying to build the official 1.0 release, Eclipse will complain about 2 errors regarding the RTFParseFactory (this is after adding the RTF jar file from the previous step). This problem was fixed (see NUTCH-644 and NUTCH-705) but was not included in the 1.0 official release because of licensing issues. So you will need to manually alter the code to remove these 2 build errors.

In RTFParseFactory.java:

1. Add the following import statement: `import org.apache.nutch.parse.ParseResult;`
2. Change

```
public Parse getParse(Content content) {
```

to

```
public ParseResult getParse(Content content) {
```

1. In the getParse function, replace

```
return new ParseStatus(ParseStatus.FAILED,
                               ParseStatus.FAILED_EXCEPTION,
                               e.toString()).getEmptyParse(conf);
```

with

```
return new ParseStatus(ParseStatus.FAILED,
                 ParseStatus.FAILED_EXCEPTION,
                e.toString()).getEmptyParseResult(content.getUrl(), getConf());
```

1. In the getParse function, replace

```
return new ParseImpl(text,
                         new ParseData(ParseStatus.STATUS_SUCCESS,
                                       title,
                                       OutlinkExtractor.getOutlinks(text, this.conf),
                                       content.getMetadata(),
                                       metadata));
```

with

```
return ParseResult.createParseResult(content.getUrl(),
                            new ParseImpl(text,
                                    new ParseData(ParseStatus.STATUS_SUCCESS,
                                            title,
                                            OutlinkExtractor.getOutlinks(text, this.conf),
                                            content.getMetadata(),
                                            metadata)));
```

In TestRTFParser.java, replace

```
parse = new ParseUtil(conf).parseByExtensionId("parse-rtf", content);
```

with

```
parse = new ParseUtil(conf).parseByExtensionId("parse-rtf", content).get(urlString);
```

Once you have made these changes and saved the files, Eclipse should build with no errors.

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 🙂

Debug Nutch in Eclipse (not yet tested for 0.9)

- 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 😉

Java Heap Size problem

If the crawler throws an IOException exception early in the crawl (Exception in thread "main" java.io.IOException: Job failed!), check the logs/hadoop.log file for further information. If you find in hadoop.log lines similar to this:

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

then 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 running all apps

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

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.

NOTE: Additional note for people who want to run eclipse with latest nutch code

If you are getting following exception - org.apache.nutch.plugin.PluginRuntimeException: java.lang.ClassNotFoundException: org.apache.nutch.net.urlnormalizer.basic.BasicURLNormalizer

1. Execute 'ant job' (which is the default) after downloading nutch through SVN
2. Update "plugin.folders" (under nutch-default.xml) to build/plugins (where ant builds plugins)
3. If it still fails increase your memory allocation or find a simpler website to crawl.

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.

Failed to get the current user's information

On Windows, if the crawler throws an exception complaining it "Failed to get the current user's information" or 'Login failed: Cannot run program "bash"', it is likely you forgot to set the PATH to point to cygwin. Open a new command line window (All Programs > Accessories > Command Prompt) and type "bash". This should start cygwin. If it doesn't, type "path" to see your path. You should see within the path the cygwin bin directory (e.g., C:\cygwin\bin). See the steps to adding this to your PATH at the top of the article under "For Windows Users". After setting the PATH, you will likely need to restart Eclipse so it will use the new PATH.

Original credits: RenaudRichardet

Updated by: Zeeshan