Topology Troubleshooting

When developing new parsers (or making other changes to the parser or enrichment topologies), it's important to be able to troubleshoot problems when things don't act as expected. This blog will cover techniques for troubleshooting the various Metron components. It is assumed that you have followed the last few sections on adding the new data source. Before starting, make sure your IDE is setup by following the instructions here.

- Troubleshooting Parser Topologies
    - Create some sample data
    - Create an integration test
    - Set some breakpoints
    - Run the test
- Troubleshooting Enrichment Topologies

Troubleshooting Parser Topologies

The integration testing framework can be a very effective way to troubleshoot topologies because they not only allow you to test parser logic (which is hopefully being done in an accompanying parser unit test) but also the related configuration files. Adding an integration test is highly recommended regardless of whether detailed troubleshooting is needed or not. The following steps describe the process for setting up and stepping through an integration test. The Squid parser that was created in Part 1 of the Metron Tutorial Fundamentals blog series will be used an example.

Create some sample data

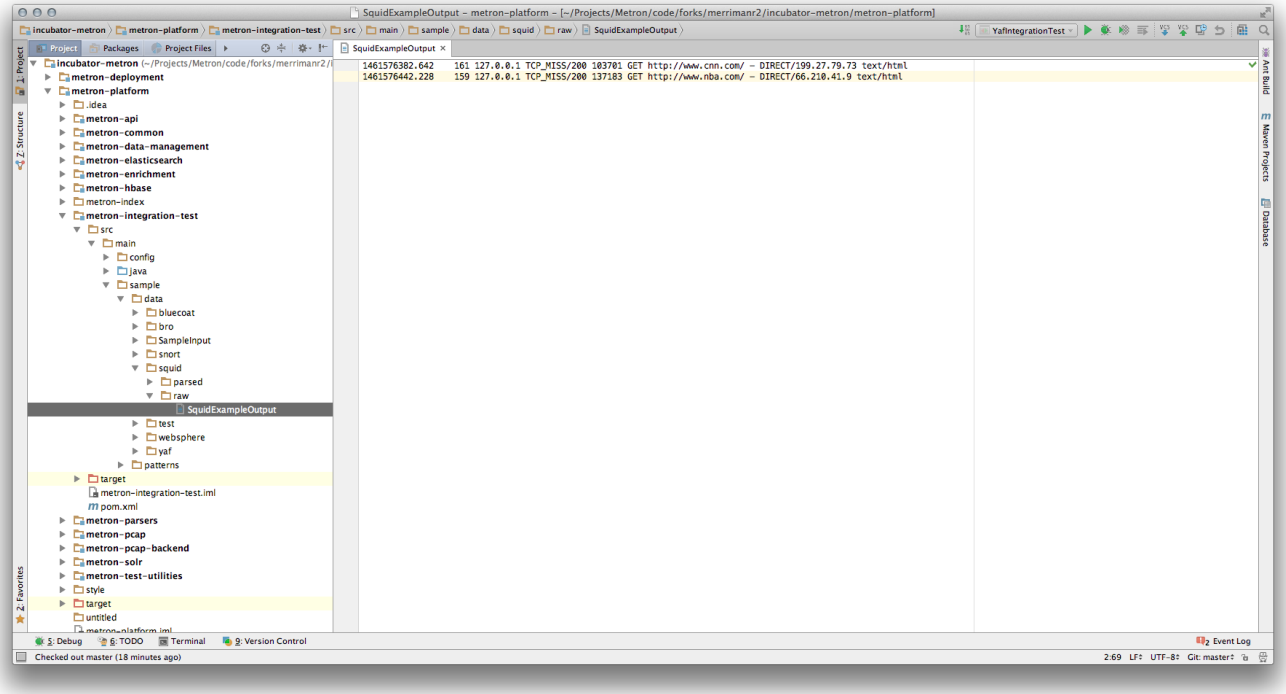
The first step is to create some sample data. Initially this can be just a couple lines, similar to what was used to write a unit test. Parser integration tests can automatically locate sample data as long as the paths follows these patterns:

/incubator-metron/metron-platform/metron-integration-test/src/main/sample/data/<sensor type>/raw (for raw data)

/incubator-metron/metron-platform/metron-integration-test/src/main/sample/data/<sensor type>/parsed (for parsed data)

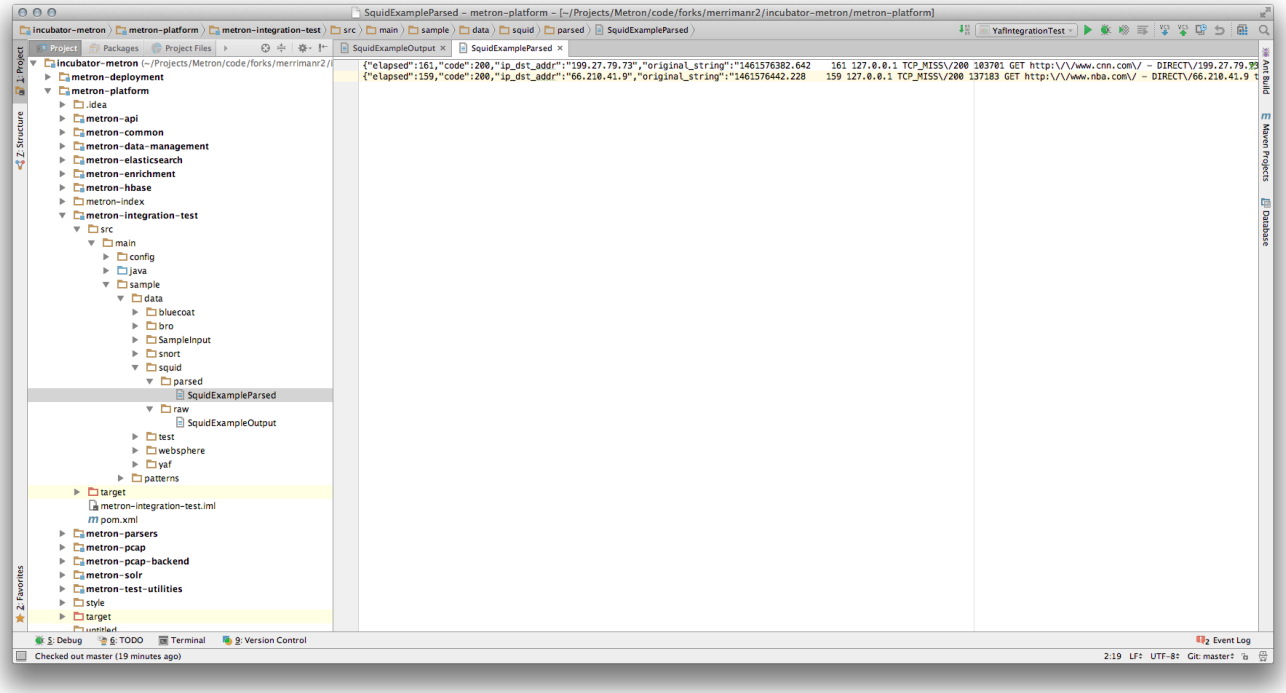
Create a file called SquidExampleOutput in /incubator-metron/metron-platform/metron-integration-test/src/main/sample/data/squid/raw with the following lines:

```
1461576382.642    161 127.0.0.1 TCP_MISS/200 103701 GET http://www.cnn.com/ - DIRECT/199.27.79.73 text/html
1461576442.228    159 127.0.0.1 TCP_MISS/200 137183 GET http://www.nba.com/ - DIRECT/66.210.41.9 text/html
```



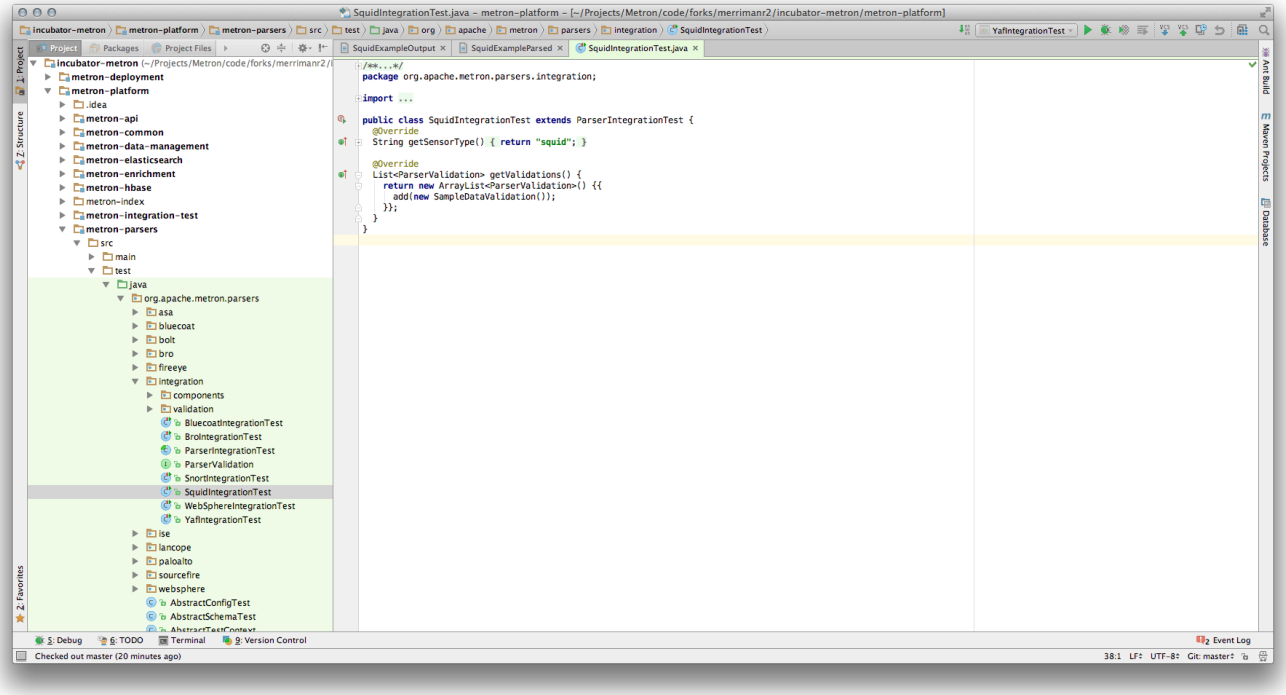
The parser integration test will test the accuracy of the parser topology by comparing raw data against expected parsed data. Create a file called SquidExampleParsed in /incubator-metron/metron-platform/metron-integration-test/src/main/sample/data/squid/parsed with the following lines:

{"elapsed":161,"code":200,"ip_dst_addr":"199.27.79.73","original_string":"1461576382.642   161 127.0.0.1 TCP_MISS\/200 103701 GET http:\/\/www.cnn.com\/ - DIRECT\/199.27.79.73 text\/html","method":"GET","bytes":103701,"action":"TCP_MISS","ip_src_addr":"127.0.0.1","url":"cnn.com","timestamp":1461576382642,"source.type":"squid"}
{"elapsed":159,"code":200,"ip_dst_addr":"66.210.41.9","original_string":"1461576442.228   159 127.0.0.1 TCP_MISS\/200 137183 GET http:\/\/www.nba.com\/ - DIRECT\/66.210.41.9 text\/html","method":"GET","bytes":137183,"action":"TCP_MISS","ip_src_addr":"127.0.0.1","url":"nba.com","timestamp":1461576442228,"source.type":"squid"}



Create an integration test

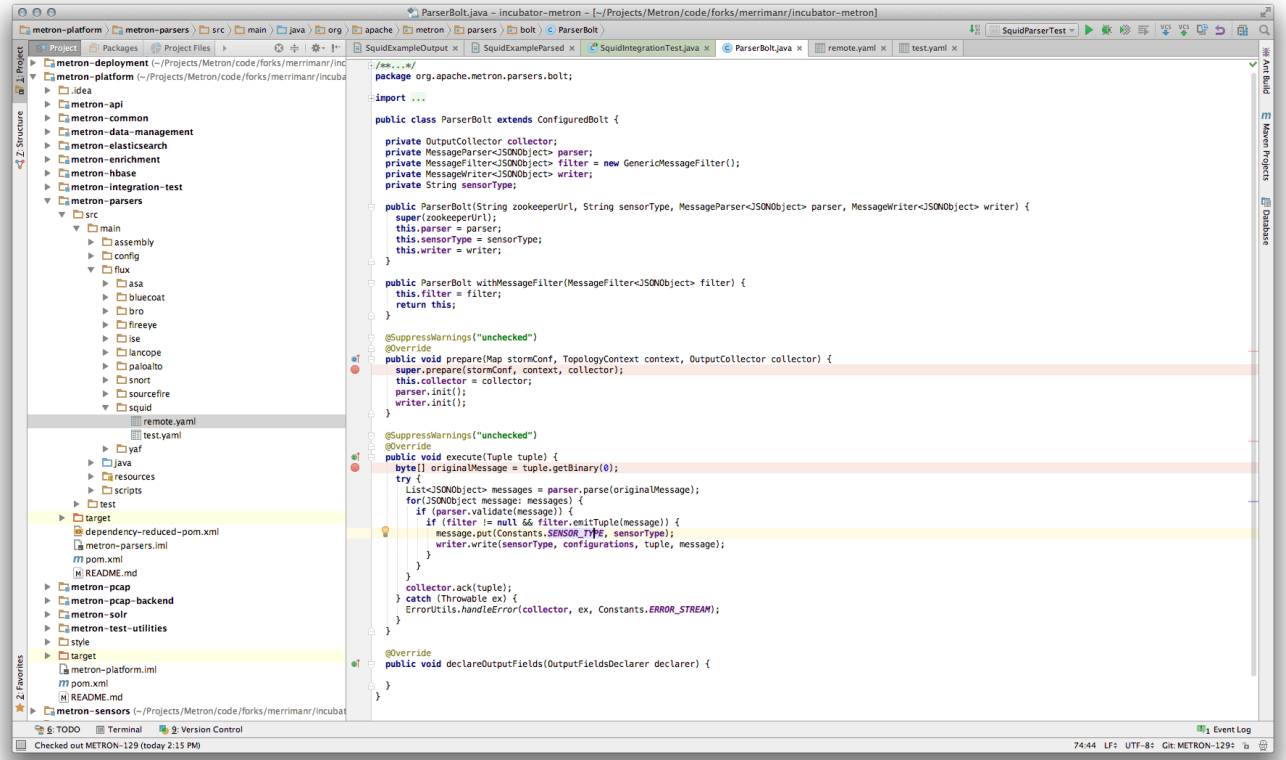
The infrastructure for running an integration test can easily be leveraged by extending the base parser integration test. Create a java class called SquidIntegrationTest in /incubator-metron/metron-platform/metron-parsers/src/test/java/org/apache/metron/parsers/integration that extends ParserIntegrationTest. The ParserIntegrationTest is an abstract class that requires a couple of methods to be implemented. These methods should be fairly intuitive (SnortIntegrationTest and YafIntegrationTest can be referenced as examples) and include providing the sensor type and validations that should be performed:



We will use the sample data created in previous steps for validation (can be copied from other parser integration tests, YafIntegrationTest for example). Now you are setup to run an integration test for the Squid parser.

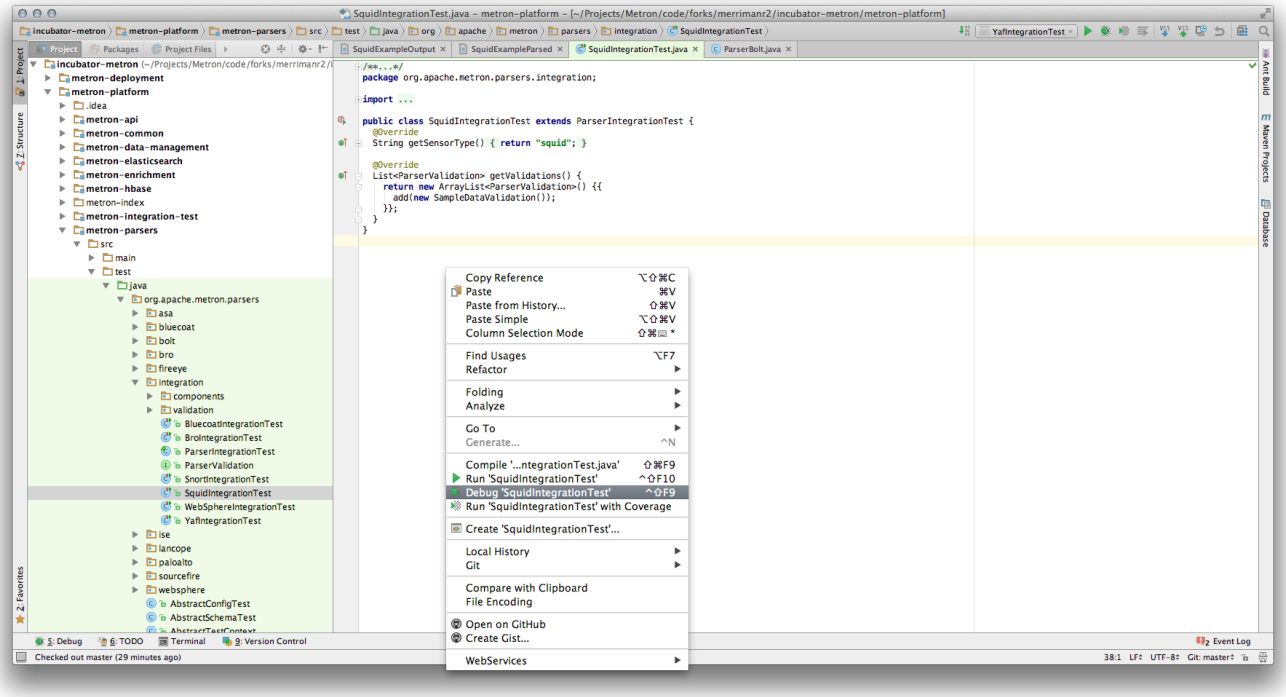
Set some breakpoints

Adding break points in the ParserBolt.prepare and ParserBolt.execute methods should provide a good starting point to troubleshooting parser topologies:

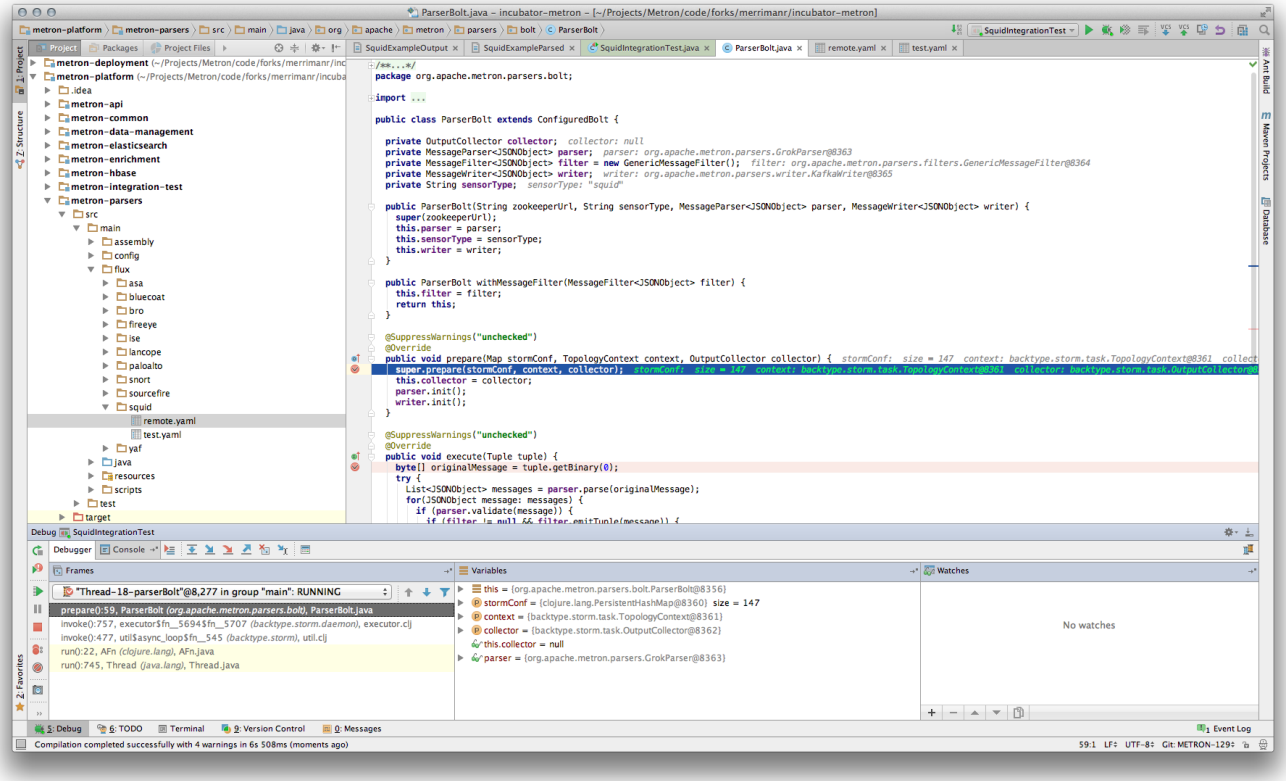


Run the test

Now run the integration test in Debug mode by simply right-clicking inside the integration test and selecting "Debug 'SquidIntegrationTest'":



You should now be able to step through the parser topology and see exactly what's going on:



Troubleshooting Enrichment Topologies

Coming soon...