

Running Cassandra In Eclipse

Eclipse is open source. Download Eclipse from <http://www.eclipse.org/downloads/>. There is no need for the Enterprise Edition (EE) version of Eclipse. Hence "Eclipse IDE for Java Developers" is good enough.

Cassandra is using Git for version control. In this tutorial we will checkout Cassandra from it.

The previous version of this guide used the Subclipse (<http://subclipse.tigris.org/>) Eclipse Subversion plugin. However, over time, the instructions became out-of-date and confusing - if you are currently using Subclipse or Subversive successfully with Cassandra be aware as of December 2011 Cassandra was switched to Git.

For the moment, we will use a command-line Subversion client.

Checkout and setup Cassandra

Preconditions: JDK6, Ant (<http://ant.apache.org/>), Eclipse, Git.

These instructions were originally tested on Ubuntu 11.04, Eclipse Indigo SR1, on the trunk code shortly after the release of Cassandra 1.0.

From the console, checkout the code using Git. Here we assume you are checking out the latest trunk, but browse <http://git-wip-us.apache.org/repos/asf/cassandra.git> for all available versions...

```
git clone http://git-wip-us.apache.org/repos/asf/cassandra.git cassandra-trunk

cd cassandra-trunk

ant build

ant generate-eclipse-files
```

The `ant build` step may take some time as various libraries are downloaded. This step will do a whole lot of good things, eg. generate the CLI grammar with ANTLR, generate Thrift RPC code...

It is important that you generate the Eclipse files with Ant before trying to set up the Eclipse project. In the previous version of this tutorial, we executed the Ant tasks after importing the code into Eclipse, which tends to confuse Eclipse mightily, leaving you stuck with about 2000 compiler errors that you can't get rid of.

Setup Eclipse

*Start Eclipse.

*Click the `File->New->Java Project` menu

*Enter "cassandra-trunk" as the project name (assuming you checked the code out into the folder cassandra-trunk as shown above. If your project name doesn't match the top-level folder name you may get problems).

*Untick "Use default location" and browse to the top-level folder you just checked out (cassandra-trunk).

*Click "Next"

*Verify that the Default Output Folder is `cassandra-trunk/build/classes/main`. Eclipse normally defaults to `bin` which we DO NOT want for Cassandra.

*Click "Finish"

(The Eclipse console will show a few lines of output as the build process runs).

That's it. There should be no errors shown in the Problems view or the Package Explorer. In fact, problem markers (Errors and warnings in the Problems pane and in the Package Explorer) will not be working at all, because the custom builder replaces the default Java Builder. This should probably be fixed...

...in the meantime, you can add back the default Java Builder alongside the custom Ant Builder. Close Eclipse, edit `.project` to add the following next to the existing `buildCommand` tag, and save the file

```
<buildCommand>
  <name>org.eclipse.jdt.core.javabuilder</name>
  <arguments>
  </arguments>
</buildCommand>
```

Then restart Eclipse, and refresh the project (F5). You should then see plenty of warnings (about 2400 at the time of writing) in the Problems view.

Common Errors

artifact:pom error

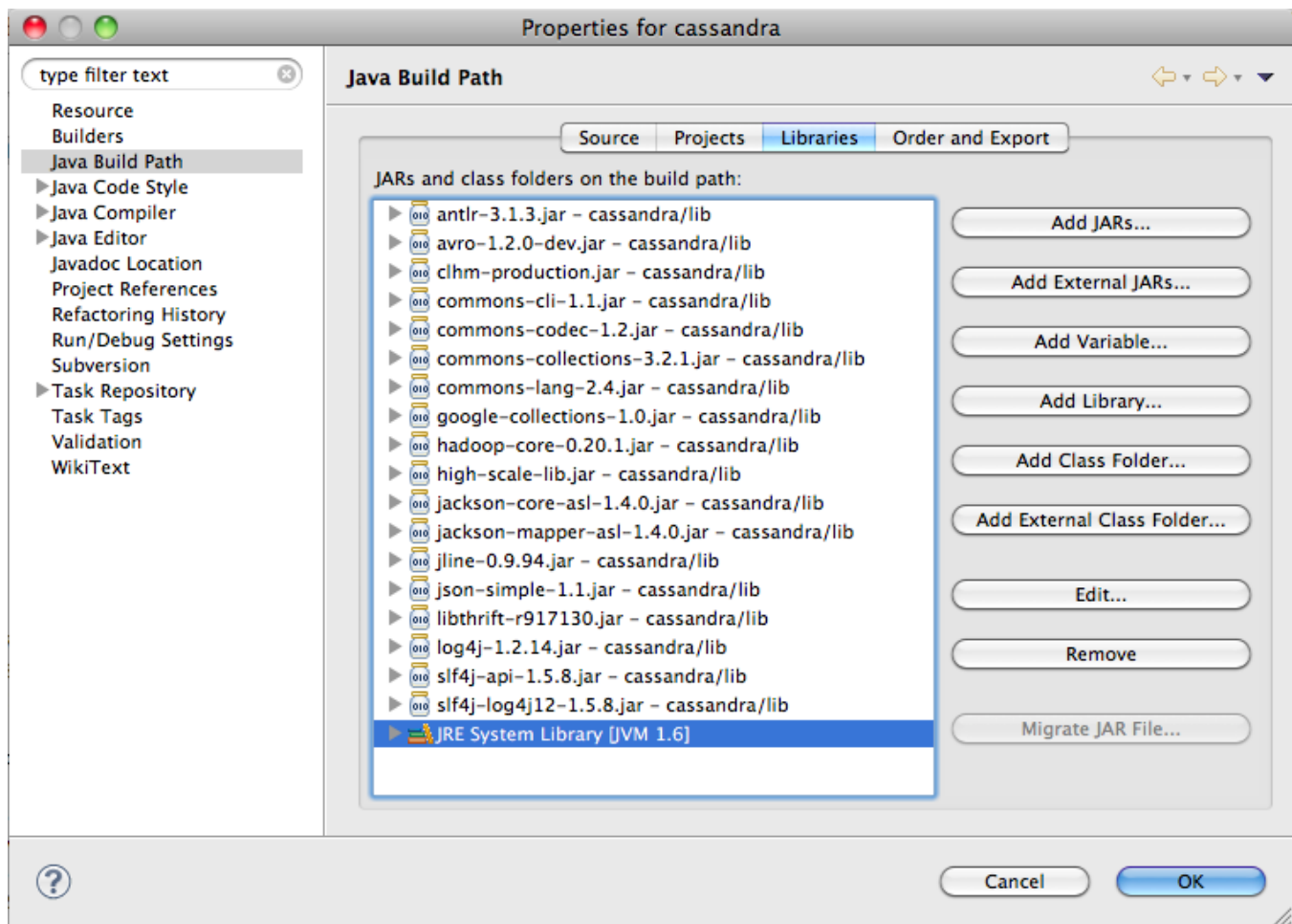
If you get the error `artifact:pom doesn't support the "groupId" attribute` when running `ant build`, then make sure your copy of Ant is up to date, and try downloading the latest Maven Ant Tasks from <http://maven.apache.org/ant-tasks/download.html> and add the jarfile to your Ant installation's `lib` directory.

Access Restriction

Some Eclipse users have complained about the following error message:

Access restriction: The method `getDuration()` from the type `GcInfo` is not accessible due to restriction on required library `/System/Library/Frameworks/JavaVM.framework/Versions/1.6.0/Classes/classes.jar`.

This is because some parts of the Cassandra project are using stuff from the `com.sun.*` package and Eclipse's "default behaviour" (bug?) is to report that as an error, IntelliJ IDEA does not. By simply reploting the "JRE System Library" this will hopefully disappear. To do that, 1) go to your build path settings for your cassandra project in Eclipse, and under the libraries tab, select "JRE System Library" and click the "Remove" button 2) re-add the JRE System Library using "Add Library" (Thanks Todd Bloose for this excellent solution!)



Still "Access restrictions" problems? Another solution to fix this is to change the "severity level" from error to e.g. warning for this. To do this right click on your cassandra project -> "Properties" -> "Java Compiler" -> "Error/Warnings" and locate "Forbidden reference (access rule)" under "Deprecated and restricted API"

ForbiddenReference-12.png!

Now the errors should be gone and you are ready to create a run/debug configuration for cassandra.

StackOverflowError

If you get a [StackOverflowError](#) while building Cassandra, you may need to upgrade to a more recent version of Ant (at least 1.8.1) - see <https://issues.apache.org/jira/browse/CASSANDRA-2640>. To upgrade Ant, you may need to upgrade to a recent version of Eclipse, as older versions (e.g. Helios SR2) don't appear to offer an upgrade via "Check for Updates" or "Install New Software".

(See https://bugs.eclipse.org/bugs/show_bug.cgi?id=302296 for recent history of the efforts to get Eclipse up to this version of Ant!)

Variable references non-existent resource

If you get the following (probably when refreshing the project after first setting it up):

```
Errors occurred during the build.
Errors running builder 'Integrated External Tool Builder' on project 'Cassandra-trunk'.
Variable references non-existent resource : ${workspace_loc:/cassandra-trunk/build.xml}
Variable references non-existent resource : ${workspace_loc:/cassandra-trunk/build.xml}
```

This probably means that the project name you chose in Eclipse doesn't match the top-level directory (must match exactly, including capitalisation). Go to **Project->Properties->Builders**, select **Cassandra-Ant-Builder**, **Edit**, and **fix the Build File and Base Directory to match the name in Eclipse** (the easiest way is just to browse to the right file/folder).

Can't find javadoc.exe

On Windows, at least, if you try to run an Ant target that invokes javadoc (such as the `artifacts` target), then you may get an error like:

```
Javadoc failed: java.io.IOException: Cannot run program "javadoc.exe": CreateProcess error=2, The system cannot find the file specified
```

Like it says, Eclipse can't find Javadoc.exe because it's not on the path.

The simplest fix seems to be to go to **External Tools Configurations**, and edit your configuration for the Cassandra build.xml: go to the **JRE** tab and make sure that you have selected a **JDK** not a **JRE**.

See <http://blog.darevay.com/2008/12/running-javadoc-ant-task-from-eclipse/> for some discussion of this issue and some alternative fixes.

Other compile errors

If Eclipse still complains about compile errors it may be because `'src'` (and not `'src/java'`) is added as source folder. To fix this I recommend to remove `'src'` from build path and add `'src/java'` to the build path.

Unit Tests

Question - can JUnit tests be run through Eclipse, or should they only be run using the Ant build target?

Note that some of the unit tests use Java `assert` as well as JUnit `assertEquals()` etc, so they need to be run with assertions enabled in the JVM (using `-ea`). If you just run them directly in Eclipse/JUnit without assertions enabled, you may get false test passes.

Run Cassandra

Click **"Run"** -> **"Run Configurations..."**. Select `org.apache.cassandra.thrift.CassandraDaemon` as your Main class, make sure that your cassandra project is selected in the **"Project"** field. Under the **Arguments** tab you can specify VM arguments. Below is my complete VM arguments list for Cassandra 0.6:

```
-Dstorage-config=/Users/schildmeijer/Documents/workspace/cassandra/conf/
-Dcassandra-foreground
-ea -Xmx1G
```

Below are another set of VM arguments for running Cassandra 0.7+ in Windows. Notice that for config file property has changed to `cassandra.config` from `storage-config` in 0.6 and that all the files need to have `"file:"` prepended, for example:

```
-Dcassandra.config=file:C:\Users\Joaquin\workspace\cassandra-3\conf\cassandra.yaml
-Dcassandra-foreground
-ea -Xmx1G
-Dlog4j.configuration=file:C:\Users\Joaquin\workspace\cassandra-3\conf\log4j-server.properties
```

or

```
-Dcassandra.config=file:/home/david/programming/java/cassandra/conf/cassandra.yaml
-Dcassandra-foreground
-ea -Xmx1G
-Dlog4j.configuration=file:/home/david/programming/java/cassandra/conf/log4j-server.properties
```

Make sure to change the storage-config property so it defines the path to your storage-conf.xml file.

Add the following VM arguments if you want to use nodetool

```
-Djava.rmi.server.hostname=$JMX_HOST
-Dcom.sun.management.jmxremote.port=$JMX_PORT
-Dcom.sun.management.jmxremote.ssl=false
-Dcom.sun.management.jmxremote.authenticate=false
```

replace \$JMX_HOST and \$JMX_PORT with your own configurations.

That should be it. After running the newly created run configuration you should see something like this in the console view.

```
11/10/22 21:52:31 INFO service.AbstractCassandraDaemon: JVM vendor/version: Java HotSpot(TM) 64-Bit Server VM/1.6.0_27
11/10/22 21:52:31 INFO service.AbstractCassandraDaemon: Heap size: 96468992/1431699456
11/10/22 21:52:31 INFO service.AbstractCassandraDaemon: Classpath: C:\Users\David\eclipse_workspace\cassandra-trunk\build\classes\main; [...]
11/10/22 21:52:31 INFO utils.CLibrary: Unable to link C library. Native methods will be disabled.
11/10/22 21:52:31 INFO config.DatabaseDescriptor: Loading settings from file:C:/Users/David/eclipse_workspace/cassandra-trunk/conf/cassandra.yaml
11/10/22 21:52:31 INFO config.DatabaseDescriptor: DiskAccessMode 'auto' determined to be mmap, indexAccessMode is mmap
11/10/22 21:52:31 INFO config.DatabaseDescriptor: Global memtable threshold is enabled at 455MB
11/10/22 21:52:31 INFO config.DatabaseDescriptor: Couldn't detect any schema definitions in local storage.
11/10/22 21:52:31 INFO config.DatabaseDescriptor: Found table data in data directories. Consider using the CLI to define your schema.
11/10/22 21:52:31 INFO commitlog.CommitLogSegment: Creating new commitlog segment /var/lib/cassandra/commitlog/CommitLog-1319316751680.log
11/10/22 21:52:31 INFO commitlog.CommitLog: No commitlog files found; skipping replay
11/10/22 21:52:31 INFO service.StorageService: Cassandra version: 1.0.0-SNAPSHOT
11/10/22 21:52:31 INFO service.StorageService: Thrift API version: 19.18.0
11/10/22 21:52:31 INFO service.StorageService: Loading persisted ring state
11/10/22 21:52:31 INFO service.StorageService: Starting up server gossip
11/10/22 21:52:31 INFO db.ColumnFamilyStore: Enqueuing flush of Memtable-LocationInfo@779185335(192/240 serialized/live bytes, 4 ops)
11/10/22 21:52:31 INFO db.Memtable: Writing Memtable-LocationInfo@779185335(192/240 serialized/live bytes, 4 ops)
11/10/22 21:52:31 INFO db.Memtable: Completed flushing \var\lib\cassandra\data\system\LocationInfo-h-1-Data.db (300 bytes)
11/10/22 21:52:31 INFO net.MessagingService: Starting Messaging Service on localhost/127.0.0.1:7000
11/10/22 21:52:31 INFO service.StorageService: This node will not auto bootstrap because it is configured to be a seed node.
11/10/22 21:52:31 WARN service.StorageService: Generated random token 83227525456023497602384283663789453783. Random tokens will result in an unbalanced ring; see http://wiki.apache.org/cassandra/Operations
11/10/22 21:52:31 INFO db.ColumnFamilyStore: Enqueuing flush of Memtable-LocationInfo@753505921(53/66 serialized/live bytes, 2 ops)
11/10/22 21:52:31 INFO db.Memtable: Writing Memtable-LocationInfo@753505921(53/66 serialized/live bytes, 2 ops)
11/10/22 21:52:32 INFO db.Memtable: Completed flushing \var\lib\cassandra\data\system\LocationInfo-h-2-Data.db (163 bytes)
11/10/22 21:52:32 INFO service.StorageService: Node localhost/127.0.0.1 state jump to normal
11/10/22 21:52:32 INFO service.StorageService: Bootstrap/Replace/Move completed! Now serving reads.
11/10/22 21:52:32 INFO utils.Mx4jTool: Will not load MX4J, mx4j-tools.jar is not in the classpath
11/10/22 21:52:32 INFO thrift.CassandraDaemon: Binding thrift service to localhost/127.0.0.1:9160
11/10/22 21:52:32 INFO thrift.CassandraDaemon: Using TFastFramedTransport with a max frame size of 15728640 bytes.
11/10/22 21:52:32 INFO thrift.CassandraDaemon: Using synchronous/threadpool thrift server on localhost/127.0.0.1 : 9160
11/10/22 21:52:32 INFO thrift.CassandraDaemon: Listening for thrift clients...
```

Note: You may find that you receive an error like:

```
java.io.IOException: java.io.IOException: unable to mkdirs /var/lib/cassandra/data
```

when running `org.apache.cassandra.thrift.CassandraDaemon`. This is because the `cassandra.yaml` file contains this path. Change these paths to a path which you have access to.

Try running the `cassandra-cli` as per [CassandraCli](#), and you should be able to connect to `localhost/9160`.

Problems? contact me (rschildmeijer) in #cassandra on irc ([freenode](#)).

<https://c.statcounter.com/9397521/0/fe557aad/1/> | stats