

EclipsePlugIn

Hadoop Eclipse Plug-in

- [JIRA MAPREDUCE-1262](#) has the latest status for this plugin. The JIRA contains a compiled Plugin JAR you can use for Hadoop 0.20.0 and 0.20.1*

[JIRA MAPREDUCE-1280](#) contains a version of the plugin that works with hadoop 0.20.2 and eclipse 3.5/3.6

The Hadoop Eclipse Plug-in provides tools to ease the experience of Map/Reduce on Hadoop. Among other things, the plug-in provides support to:

- create Mapper, Reducer, Driver classes;
- browse and interact with distributed file systems;
- submit jobs and monitor their execution.

Requirements

To ensure a safe behavior of the plug-in, you should consider the following facts and recommendations:

- The plug-in has been tested on Eclipse 3.2+ only, using Java5+ compliant JVMs only.
- To compile jobs on your development host, you need Hadoop jars. In most cases, you only need `hadoop-X.Y-core.jar`. Make sure you build your jobs with the same version of Hadoop as the version of Hadoop your execution environment currently runs.
- Make sure you when you compile your jobs that the Java classes you generate are compatible with the JVM environment of your execution environment. A frequent issue is to have a Java5 execution environment that cannot executes your jobs because your jobs have been compiled with a Java6 compliance level.

More will come here soon.

Overview

Here is an overview of the Eclipse environment for Hadoop:

`drawing:EclipseMain`

The environment is accessible through the "Map/Reduce perspective" (the blue elephant icon in the top-right side of the main window). To open this perspective, select the menu: Window, Open Perspective, Other, and finally Map/Reduce. This perspective is roughly a Java edition environment extended with:

- a view named "Map/Reduce locations" which lists Hadoop locations (the view at the bottom of the main window),
- a file browser for the distributed file systems associated to each Hadoop locations (on the left side).

Hadoop Map/Reduce locations

The location view allows the creation, edition and deletion of Map/Reduce locations.

To create a new location click on the "New Hadoop location..." button in the view toolbar or in the contextual menu.

`drawing:LocationContextualMenu`

A wizard pops up and asks for the location parameters.

`drawing:NewLocationWizard`

You must at least fill the following entries:

- the location name (avoid punctuation marks),
- the masters addresses: hostname or IP address and TCP port numbers for the Map/Reduce master (the [JobTracker](#)) and for the Distributed File System (the [NameNode](#)).

The Map/Reduce and the DFS masters are colocated by default (i.e. run on the same host).

A SOCKS proxy can be configured if you cannot access the Map/Reduce location directly because your machine is not directly connected to the location. See section "How to use SOCKS proxies" for more details.

Not implemented yet: user name, validation button, load from file button.

How to use SOCKS proxies

To set up a local proxy through an ssh server PROXY at port 21080, for example:

```
ssh -D 21080 PROXY
```

Note that when using a SOCKS proxy in a local client like the Eclipse plugin, you should ensure that your Hadoop cluster does not inherit your proxy settings, or the Hadoop daemons won't be able to communicate with each other. To override proxy settings in the Hadoop nodes, add the following property setting to each node's `hadoop-site.xml`:

```
<property>
  <name>hadoop.rpc.socket.factory.class.default</name>
  <value>org.apache.hadoop.net.StandardSocketFactory</value>
  <final>true</final>
  <description>
    Prevent proxy settings set up by clients in their job configs from affecting our connectivity.
  </description>
</property>
```

The standard socket factory produces RPC sockets with direct connections (ie, without going through any proxies), and the "final" attribute prevents your job configuration from overriding this property's value.

How to build and install the plug-in

To build the Eclipse plug-in, you need the Hadoop source files and a working Eclipse environment (version 3.3+). When compiling Hadoop, the Eclipse plug-in will be built if it finds the Eclipse environment path in the ant property "eclipse.home". The build framework looks for this property in `/${hadoop-src-root}/src/contrib/eclipse-plugin/build.properties` and in `$/HOME/eclipse-plugin.build.properties`.

A typical `$/HOME/eclipse-plugin.build.properties` file would contain the following entry: `eclipse.home=/path/to/eclipse`

Then the plug-in should be built when compiling Hadoop: `ant clean package` (from the `/${hadoop-src-root}` directory), which will produce `/${hadoop-src-root}/build/contrib/eclipse-plugin/hadoop-${version}-eclipse-plugin.jar`

To install the generated plug-in in your Eclipse environment, remove first all previous versions of the plug-in from your Eclipse environment and copy the `hadoop-${version}-eclipse-plugin.jar` file generated as described above in your `/${eclipse.home}/plugins/` directory. When you restart Eclipse, the Map/Reduce perspective should be available.