

Development Environment

This document describes how you can setup your development environment.

- [Prerequisites](#)
 - [Java 1.6](#)
 - [Maven 3.x](#)
- [Getting and Building Sources](#)
 - [Basic Build](#)
 - [Trying out Built Artifacts](#)
- [Setting up Eclipse](#)
 - [Basic Eclipse Setup](#)



The instructions provided in this document apply to Flume 1.x, the current trunk version. Instructions for setting up a development environment for version 0.9.x can be found [here](#).

Prerequisites

Java 1.6

Flume is written in Java and requires JDK 1.6 for build and runtime. Please download and install the [latest update of JDK 1.6](#). Even though having the Java executable in your path will suffice, it is better to explicitly set your `JAVA_HOME`. Once installed, make sure that is working by typing the following command:

```
$ java -version
java version "1.6.0_31"
Java(TM) SE Runtime Environment (build 1.6.0_31-b04-415-11M3635)
Java HotSpot(TM) 64-Bit Server VM (build 20.6-b01-415, mixed mode)
$
```

Maven 3.x

The build tool used for building and testing Flume is Apache Maven. Download the [latest update of Maven 3.x](#) and install it locally on your system. Once installed, make sure the maven executable `mvn` is in your path and test it out using the following command:

```
$ mvn -version
Apache Maven 3.0.3 (r1075438; 2011-02-28 09:31:09-0800)
Maven home: /usr/share/maven
Java version: 1.6.0_31, vendor: Apple Inc.
Java home: /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home
Default locale: en_US, platform encoding: MacRoman
OS name: "mac os x", version: "10.7.3", arch: "x86_64", family: "mac"
$
```

Getting and Building Sources

Basic Build

Use the following command to checkout the sources in a convenient location:

```
$ git clone http://git-wip-us.apache.org/repos/asf/flume.git flume
Cloning into 'flume'...
remote: Counting objects: 17984, done.
remote: Compressing objects: 100% (5382/5382), done.
remote: Total 17984 (delta 8728), reused 17900 (delta 8702)
Receiving objects: 100% (17984/17984), 33.72 MiB | 3.18 MiB/s, done.
Resolving deltas: 100% (8728/8728), done.
$
```

Once the checkout is complete, you can build these by using `mvn install` as follows from within the workspace root.

```

$ cd flume
$ mvn install -DskipTests
...
[INFO] Flume NG distribution ..... SUCCESS [17.448s]
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 1:03.655s
[INFO] Finished at: Sun May 27 11:43:59 PDT 2012
[INFO] Final Memory: 40M/117M
[INFO] -----
$

```

Note that specifying `*-DskipTests` to `mvn` results in the skipping of all unit tests. This may be a good thing if you want to quickly setup the development environment. However, in the general case when you are making or testing changes, you must always run the tests. You can do that by running `mvn install` without the `-DskipTests` option, or explicitly by running `mvn test`.

Trying out Built Artifacts

The module `flume-ng-dist` within the source tree packages the built artifacts so that you can try them out as a local installation. To do this, untar the distribution tar-ball in a convenient location and invoke the `bin/flume-ng` script as shown below.

```

$ mkdir /path/to/testdir
$ cd /path/to/testdir
$ tar zxvf /path/to/workspace/flume/flume-ng-dist/target/apache-flume-<version>-dist.tar.gz
...

$ cd apache-flume-<version>
$ bin/flume-ng
Error: Unknown or unspecified command ''

Usage: bin/flume-ng <command> [options]...

commands:
  help                display this help text
  agent               run a Flume agent
  avro-client          run an avro Flume client

global options:
  --conf,-c <conf>    use configs in <conf> directory
  --classpath,-C <cp> append to the classpath
  --dryrun,-d          do not actually start Flume, just print the command
  -Dproperty=value     sets a JDK system property value

agent options:
  --conf-file,-f       specify a config file (required)
  --name,-n            the name of this agent (required)
  --help,-h            display help text

avro-client options:
  --host,-H <host>     hostname to which events will be sent (required)
  --port,-p <port>     port of the avro source (required)
  --filename,-F <file> text file to stream to avro source [default: std input]
  --headerFile,-R <file> headerFile containing headers as key/value pairs on each new line
  --help,-h            display help text

Note that if <conf> directory is specified, then it is always included first
in the classpath.

$

```

Setting up Eclipse

Basic Eclipse Setup

To setup the flume project for development in Eclipse, you can use either the maven-eclipse plugin or m2eclipse plugin for eclipse. Here is how you would use the maven-eclipse plugin:

```
$ cd /path/to/workspace/flume
$ mvn install -DskipTests
$ mvn eclipse:eclipse -DdownloadSources -DdownloadJavadocs
...
[INFO] Flume NG distribution ..... SUCCESS [0.170s]
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 8.963s
[INFO] Finished at: Sun May 27 11:57:03 PDT 2012
[INFO] Final Memory: 22M/81M
[INFO] -----
$
```

Once this command completes successfully, you must add \$HOME/.m2/repository to the classpath in preferences and then you can import all the flume modules as interdependent projects into Eclipse by going to File > Import > General > Existing Projects into Workspace.