

1 - Setting up Kurator-Akka

This page describes how to set up an environment for writing and running workflows for Kurator-Akka. These instructions are valid for release 0.2 and later of Kurator-Akka.

1. Check installed version of Java

Kurator-Akka requires Java version 1.7 or higher. To determine the version of java installed on your computer use the `-version` option to the java command. For example,

```
$ java -version
java version "1.7.0_67"
Java(TM) SE Runtime Environment (build 1.7.0_67-b01)
Java HotSpot(TM) 64-Bit Server VM (build 24.65-b04, mixed mode)
$
```

2. Download the Kurator-Akka jar

Kurator-Akka is distributed as a jar (Java archive) file that can be executed using the `java -jar` command. You can download the latest supported distribution from the [Software Releases](#) page.

To download the most recently built (but unsupported) preview snapshot of an upcoming distribution, follow the link in the sidebar at left to [Latest Build](#), click on the *Artifacts* tab, and download the *executable jar* artifact.

3. Install Kurator-Akka on your system

Once you have downloaded the Kurator-Akka jar, save the file in a convenient location and rename it to something like `kurator-akka.jar`. Kurator-Akka can now be run using the `java -jar` command. The jar file includes several sample workflow scripts which can be accessed from the command line using the classpath scheme.

Test your installation by running the `hello.yaml` workflow. Assuming `kurator-akka.jar` is in your current working directory type:

```
java -jar kurator-akka.jar -f classpath:/org/kurator/akka/samples/hello.yaml
```

If the Kurator-Akka jar is stored elsewhere, qualify `kurator-akka.jar` with the path to that file. If you stored the jar file in the `bin` subdirectory of your home directory (on a Unix platform), running Kurator-Akka would look something like this:

```
$ java -jar ~/bin/kurator-akka.jar -f classpath:/org/kurator/akka/samples/hello.yaml
Hello World!
$
```

Kurator-Akka can be run similarly on Windows platforms.

4. Extract the samples (optional)

If you would like to browse and edit the sample workflows included in the Kurator-Akka jar, type the following (qualifying the path to the Kurator-Akka jar as necessary) from a working directory and extract them to your filesystem:

```
jar xf kurator-akka.jar org/kurator/akka/samples
```

To run a script residing on the filesystem, you can use the file scheme:

```
$ java -jar kurator-akka.jar -f file:org/kurator/akka/samples/hello.yaml
Hello World!
```

The `file:` qualifier is optional, however. By default Kurator-Akka looks for workflows on your filesystem. So this will work, too:

```
$ java -jar ~/bin/kurator-akka.jar -f org/kurator/akka/samples/hello.yaml
Hello World!
```

5. Create an alias for Kurator-Akka (optional)

On Unix platforms (include OS X) you can create an alias for more conveniently running Kurator-Akka at the command line. For example, if you have saved the Kurator-Akka jar to the `bin` subdirectory of your home directory, the following bash command will create an alias for running Kurator-Akka simply by typing `kurator-akka` at the command prompt.

```
alias kurator-akka='java -jar ~/bin/kurator-akka.jar'
```

If you use `csh` or `tcsh` the command is:

```
alias kurator-akka java -jar ~/bin/kurator-akka.jar
```

The `hello.yaml` demo then can be run using the alias:

```
$ curator-akka -f classpath:/org/kurator/akka/samples/hello.yaml
Hello World!
$
```