

# LoggingInDefaultJettySetup

<!-- This page refers to logging technology present in Solr examples before version 4.3.0. See [SolrLogging](#) for information about newer versions.

In the default Jetty setup (in the solr example/ directory), Solr log messages are automatically printed to stdout (stderr?). Here is a quick recipe for changing that, or other logging settings:

## Creating a logging.properties file

First, create a file called logging.properties and put it in your example/ directory. It should look something like this:

### Logging to a file

```
# Default global logging level:
.level = INFO

# Write to a file:
handlers = java.util.logging.FileHandler

# Write log messages in XML format:
# Use java.util.logging.SimpleFormatter to log like Solr logs to the screen by default
java.util.logging.FileHandler.formatter = java.util.logging.XMLFormatter

# Log to the current working directory, with log files named solrxxx.log
java.util.logging.FileHandler.pattern = ./solr%u.log
```

You can find a description of the different [FileHandler](http://java.sun.com/j2se/1.4.2/docs/api/java/util/logging/FileHandler.html) properties at <http://java.sun.com/j2se/1.4.2/docs/api/java/util/logging/FileHandler.html>

### Logging to the console

The above writes beautiful XML-formatted messages suitable for automated processing to a log *file*, but I wanted a quick way to just keep from having all those INFO messages come out to the console when running the example code. Here's an alternative logging.properties file for doing that. NOTE: there are still a few INFO level messages that come out (1/2 dozen or so) which I'd guess are from Jetty

```
.level = WARNING

# Write to the console:
handlers = java.util.logging.ConsoleHandler
```

## Telling Java about logging.properties

Now we need a way to tell the Java virtual machine to look for this configuration file. The way to do it is to set the java.util.logging.config.file system property to point to your file.

One way to do this is to set the variable on the command-line each time you start java. So instead of starting Solr with

```
java -jar start.jar
```

you'll use

```
java -Djava.util.logging.config.file=logging.properties -jar start.jar
```

Another option is to modify example/etc/jetty.xml. Inside the outermost Configure tag, i.e. inside

```
<Configure id="Server" class="org.mortbay.jetty.Server">
</Configure>
```

you'll add the following:

```
<Call class="java.lang.System" name="setProperty">  
  <Arg>java.util.logging.config.file</Arg>  
  <Arg>logging.properties</Arg>  
</Call>
```

If you do this, you can start java normally, without the -D option.

## Notes

We've specified the logging.properties file without an explicit path, which means Java will look for it in the current working directory. Thus you'll need to start java from the example/ directory or your change won't work. (You can specify an explicit path if you'd like, in which case you could start java from wherever you'd like.)

This will change where the messages logged by Solr go, but it won't change where messages logged by Jetty go.