Startup How-To

@version@ (@date@)

by Dieter Wimberger

Table of contents

1. About

This document describes how to start the telnetd with the given main() or from within your own application.

2. Basics

For starting up the telnetd and the required number of listeners, you have two choices:

1. using the existing main() in the TelnetD class
2. instantiating and activating the daemon from within your application.

Either way you will have to provide the appropiate configuration properties as well as shell implementations (see deployment and configuration index (../deployment/)).

3. Using the existing main()

Assemble/prepare the properties that resemble your desired configuration and issue:

```
java -classpath telnetd.jar:commons-logging.jar:log4j.jar
net.wimpi.telnetd.TelnetD -D -Dlog4j.configuration=<your URL for
log4j.properties> <your URL for telnetd.properties>
```

Logically you will need a JRE/JDK installed (1 and above should work) and the java VM binary in your PATH.

4. Bootstrapping from within your application

It will be required to take the following steps:

1. Assemble/prepare the properties that resemble your desired configuration.
2. Create a *TelnetD* instance using one of the given factory methods in the *TelnetD* class (see API docs).
3. Start the daemon calling *start()*

An example is given below, *props* represents a single *Properties* instance containing all properties for your desired configuration:

```
//1. create singleton instance
TelnetD daemon = TelnetD.createTelnetD(props);
//2.start serving
daemon.start();
```

5. Starting through ant (build script)

You can as well use ant to start the daemon invoking ant with the `runit` target.