

# The Elk Developer's Tutorial

Claes Buckwalter  
clabu@itn.liu.se

2005-01-28

# Contents

1	Overview .....	1
2	Requirements.....	2
2.1	Java 1.4.x.....	2
2.2	Ant 1.6.x .....	2
2.3	Jakarta Tomcat 5.0.x.....	2
3	Getting Elk's Source Code .....	3
3.1	Downloading Elk from web site .....	3
3.2	Checking out Elk from CVS .....	3
4	Building Elk .....	4
5	Running Elk .....	5
5.1	Deploying Elk's web application .....	5
5.2	Sending JMF to Elk .....	5
5.3	Reference JDF Process .....	5
5.4	Troubleshooting.....	6
5.5	Known Issues.....	6
6	References.....	7

# 1 Overview

Elk is a Java-based open source application framework that defines the services needed by a piece of software that implements the role of a *JDF device*. Elk also consists of a reference device, which is an implementation of the services defined by the framework. The reference device is a J2EE web application. More information about the Elk project can be found at <http://elk.itn.liu.se>.

This tutorial is for developers who want to start developing with the Elk framework and the reference device. This tutorial describes:

1. How to retrieve Elk's source code
2. How to build Elk using Ant
3. How to deploy the Elk reference device web application in Tomcat

## 2 Requirements

### 2.1 Java 1.4.x

Java 1.4.x [1] or later is required to build Elk and run the reference implementation. Elk has been tested with J2SE 1.4.2.

### 2.2 Ant 1.6.x

The Elk source distribution comes with an Ant build script. Elk's Ant build script compiles the Elk source code and automatically bundles the necessary files into a deployable J2EE web application. Ant is an open source build tool similar to make. Ant can be downloaded from [2].

### 2.3 Jakarta Tomcat 5.0.x

The Elk reference device is a J2EE web application that can be deployed in a servlet container, such as the open source *Jakarta Tomcat* [3]. Ant can be used to build the Elk reference device web application that can then be deployed in Tomcat without any additional configuration of Elk or Tomcat.

Elk has been tested with Tomcat 5.0.x but should work with newer versions of Tomcat and other servlet containers as well.

No configuration is needed for Tomcat. Simply download the latest binary distribution of Tomcat 5.0.x and unpack it to a directory of your choice.

## 3 Getting Elk's Source Code

Elk's source code can be download from the Elk web site or checked out from CIP4's CVS repository.

For normal development the source distribution of Elk is what you need, see section 3.1. However, if you need access to the most up to date Elk source code you have to check it out from CIP4's CVS repository, see section 3.2.

### 3.1 Downloading Elk from web site

The easiest way to get Elk's source code is to download the latest source distribution (`elk-date-src.zip` or `elk-date-src.tar.gz`) from [4]. Select the file with the newest date.

### 3.2 Checking out Elk from CVS

Downloading Elk from CIP4's CVS server ensures that you have the latest available version of the Elk source code. Elk's CVS module is found under the path `jdF/elk` in CIP4's CVS repository. How to access CIP4's CVS repository is described in [5].

There are several free CVS clients [6] available that can be used to check out the Elk source code from CIP4's CVS repository. One option is to use the excellent CVS client built into the Eclipse IDE.

## 4 Building Elk

The Elk source distribution comes bundled with an Ant build script that builds the Elk reference device web application and the Elk test tools.

1. Open a command line console and go to the directory where you unpacked Elk. Run `ant -p` for a list of the available build targets.
2. Run `ant war` to build the Elk reference device web application. The result of the build is a WAR file `target/elk.war`. This web application is preconfigured and ready to be deployed in a servlet container such as *Jakarta Tomcat* [3].
3. Run `ant sendjmf` to build the Elk test tool for sending JMF messages. The result of the build is a JAR file `dist/sendjmf.jar`.

## 5 Running Elk

Elk's reference device is a J2EE web application that can be deployed in a J2EE servlet container. The Elk reference device web application has been tested with Jakarta Tomcat 5.0.x but it should work with other servlet containers as well. This section describes how to deploy and use the Elk reference device web application in Tomcat.

### 5.1 Deploying Elk's web application

1. Download and install/unpack Tomcat 5.0.x.
2. Build the web application using Ant: `ant war`
3. Copy the resulting WAR file `target/elk.war` to Tomcat's web application directory `$CATALINA_HOME/webapps`.
4. Start Tomcat by running `$CATALINA_HOME/bin/startup.sh` or `startup.bat` depending on your platform.
5. Open a web browser and enter the address <http://localhost:8080/elk/jmf>. If the page displays a HTTP status message "*HTTP Status 501 - HTTP GET not implemented*" the Elk web application is up and running. If not, see the section below on troubleshooting.

### 5.2 Sending JMF to Elk

You send JMF messages to the Elk reference device using the URL:

<http://localhost:8080/elk/jmf>. *localhost* should be replaced with the servers IP address or hostname if you are accessing Elk remotely.

1. Send a `KnownMessages` JMF query to Elk to find out what it can do.
2. Then use any of the JMF commands or queries returned to interact with Elk.

Elk currently implements a limited set of JMF messages. Support for additional JMF messages and other functionality is continuously added to the Elk reference device so be sure to check the Elk web site (<http://elk.itn.liu.se>) for updates.

### 5.3 Reference JDF Process

The Elk reference device currently implements the *JDF Approval* process. To submit a job to Elk do the following:

1. Send a `SubmitQueueEntry` JMF command to Elk
2. Elk will place the job in its queue
3. Elk will run the job. Once the job is started it takes takes 10 seconds.
4. Finished jobs are returned in either of two ways:
  - a. Sending a `ReturnQueueEntry` JMF command to the URL specified in the job's `ReturnJMF`
  - b. Sending the job's JDF job ticket to the URL specified in the job's `ReturnURL`

## **5.4 Troubleshooting**

If you run into problems with the Elk reference device the first place to look for information about what could be wrong is in Elk's log file and the servlet container's log files:

1. `elk.log`  
Elk's log file is located in the directory from where you started Tomcat. Most likely it will be in `$CATALINA_HOME/bin`. By modifying Elk's web application configuration you can change where this log file is written.
2. `$CATALINA_HOME/log`  
Tomcat's log files are written to this directory. The file `catalina.out` contains useful information if something went wrong during startup.

## **5.5 Known Issues**

1. The HTTP header Content-Length is not set in the HTTP response sent from Elk.
- 2.

## 6 References

---

1 Java 2 Platform, Standard Edition (J2SE), <http://java.sun.com/j2se/downloads/index.html>

2 *Ant*, <http://ant.apache.org>

3 *Jakarta Tomcat*, <http://jakarta.apache.org/tomcat>

4 *Elk distributions*, <http://elk.itn.liu.se/maven/elk/distirubtions/>

5 Guide for CIP4 Open Source,  
[http://www.cip4.org/intern/help/guide\\_for\\_cip4\\_open\\_source.pdf](http://www.cip4.org/intern/help/guide_for_cip4_open_source.pdf)

6 *MacCvsX*, <http://cvsgui.sourceforge.org>