# Java Media Framework Installation Guidelines and Examples

## Prerequisites

An installation of Java SDK is required. You may find the latest releases here <http://www.oracle.com/technetwork/java/javase/downloads/index-jsp-138363.html>.

This installation can be combined with a development environment, such as Netbeans <http://www.oracle.com/technetwork/java/javase/downloads/jdk-netbeans-jsp-142931.html>. Other development environments such as Eclipse etc may be also used.

Upon the installation of Java SDK, you have to update the path in the environmental variables of your computer, adding the bin folder (e.g. C:\Program Files\Java\jdk1.6.0\_23\bin).

The next step is to download and install the Java Media Framework (the latest version is JMF 2.1.1e) from the following link <http://www.oracle.com/technetwork/java/javase/download-142937.html>.

Upon the installation of JMF, you have to update the path in the environmental variables of your computer, adding the bin folder (e.g. C:\Program Files\JMF2.1.1e\bin).

Compilation and execution of the developed programs may be done through the application environment or manually (javac command for compilation and java command for execution).

In the latter case (manual editing), you can use an editor like SciTE <http://www.scintilla.org/SciTE.html>.

## JMF examples

If you do not have experience with Java, you may find useful information as well as Java tutorials in the following links:

* <http://download.oracle.com/javase/tutorial/>
* <http://journals.ecs.soton.ac.uk/java/tutorial/TOC.html>
* <http://math.hws.edu/javanotes/>

In the following link <http://www.oracle.com/technetwork/java/javase/solutions-137579.html>, you may find several examples using JMF in order to get used with it.

Furthermore, we are going to examine the following examples within the lab.

### Playing multimedia files

In the folder JMF\_examples you will find several .java files as well as their correlated build and run files.

The first example is “simple\_jmf\_player.java”. In this example an instance of a JMF player is created, and a multimedia file that is given as argument in the corresponding run file is loaded in the player. Specific functions such as stop, pause and play are supported. Error messages are shown in case of failures.

The second example is “simple\_jmf\_player2.java”. In this example, a control panel is added with specific buttons (play, pause, stop). The functionality of each button is specified. For each button, an ActionListener is declared. In case of an event (pressing a button) the corresponding action is taken.

For example in the following part, try to remove the comment and see the difference when pressing the “play” button:

*if(command.equalsIgnoreCase("play"))*

*{*

*// p.setMediaTime(new Time(0));*

*p.start();*

*}*

The third example is “simple\_jmf\_player3.java”. In this example, two players with the corresponding panels and media locators are instantiated. The layout of each panel and component is defined. Check the differences in case you remove the comments in each of the following lines:

*else if (evt instanceof EndOfMediaEvent) {*

*//p.setMediaTime(new Time(0));*

*//p2.setMediaTime(new Time(0));*

*//p.start();*

*//p.close();*

*//System.exit(0);*

*} else if (evt instanceof SizeChangeEvent) {*

*}*

The fourth example is “simple\_jmf\_player4.java”. In this example, a new functionality is added related to the capability of grabbing specific frames from the video. Remove all the comments that have the tag “grab frame” and run this example. When you press the button “grab” the current frame of the video is saved in jpeg format in the same folder that you have saved your java files. This is done through the function saveJPG (several options are available such as the definition of the quality of the picture). Check the process of adding a new button in the panel and the way of specification of its functionality upon an event.

### Streaming multimedia files

In the folder RTP\_Streaming you will find a complete example for the setup of a client server application based on JMF. The server transmits multimedia files through the web and the clients are able to view them on their desktop.

The source files for the transmitter and the receiver are available on the corresponding folders. For both the transmitter and the receiver, the IP addresses and ports for the transmission are defined. Transmission is based on the RTP protocol. A processor is created for the specified multimedia files under transmission and an RTP session is created in order to transmit the output of the processor to the specified IP address and port. Similarly, in the receiver, an RTP manager handles the incoming RTP sessions, instantiates the player and plays the multimedia files.

Note: In the corresponding run files, the IP addresses and ports, as well as the multimedia file under transmission have to be declared.