# CS25610 / CSM5610 Worksheet 5 1999-00

**Dave Price**

April 2, 2000

## 1 Introduction

This worksheet is intended to provide some experience of using the Java Media Framework (JMF). As has already been explained in an email, rather than getting you to create applications from scratch, I'm going to ask you to run some of my applications which have been discussed in lectures, observe the facilities that the JMF provides and perhaps fix a few of my bugs!

You will very much need to consult your lecture notes, the listings I have given out, and on-line resources linked from the course materials web site at

http://www.aber.ac.uk/~dcswww/Dept/Teaching/Courses/CS25610/

to complete this work.

## 2 Background Information

This worksheet asks you to run some Java programs that make use of the JMF. I get you to run most of the programs on the workstation at which you are sitting, but I ask you to run one or two remotely on a Sun running Solaris so that you can see some differences.

### 2.1 Making the JMF available

To make the JMF available for your programs to use, you will need to add some information to your environment. As you all use the BASH shell, I've written a small bash script that can either be added to the end of your .bashrc file, or can be run using the "source" command. I've left the script in a file called /dcs/dap/jmf.bashrc and you can cause it to effect your environment by typing

```
     source /dcs/dap/jmf.bashrc
```

Or, as i've mentioned, you could edit this onto the end of your .bashrc file in your home directory. I would recommend the first option at the moment until we are certain that the script cannot have any adverse side effect. It has been tested, but you never know.....

### 2.2 Starting Windows on Stonkin

Stonkin is a server machine sitting in our computer room. It is a “real” Sun Sparc running Solaris, rather then being an Intel machine running Solaris. On Stonkin, the JMF with the Sparc “performance pack” is available and so some extra bits of the JMF are available there which are not available on the “pure Java” version which is used on the Solarium machines.

When, in the text below, I ask you to start a window on stonkin, type the following command

```
rcmd stonkin /usr/dt/bin/dtterm
```

Once the window is displayed you will again need to source the jmf.bashrc file if it has not been added to your .bashrc file.

### 2.3 The Java/JMF programs for you to run

I have made copies of all of the programs in some directories in the filestore in the directory

```
/dcs/dap/CS25610/JMF
```

The directory /dcs/dap/CS25610/JMF/Players contains Java programs and a couple of movie files. Copy all of these into a directory in your own filestore. The directories /dcs/dap/CS25610/JMF/Capture and /dcs/dap/CS25610/JMF/Conversion contain just Java programs, make copies of these too.

The final directory, /dcs/dap/CS25610/JMF/Movies, contains various samples of movies that we happen to have. Do NOT copy these, I’ll ask you to use them directly from my filestore.

## 3 Your Tasks

### 3.1 Using SimplePlayer.java

#### 3.1.1 Running on your Intel/Solaris Workstation

Take a copy of the file SimplePlayer.java into your filestore. Make sure you have sourced the jmf.bashrc file (see above) and then compile the player.

Make sure the directory in which you are working also contains the movie file sample.mov. Now run the compiled program. All being well, a window should pop up, a video should play with sound (headphones available) and then you should be able to quit by pressing the appropriate button.

As the program is starting and the video is playing, look carefully in the window from where you ran the program. You will see a stream of messages output by my program to document interesting events and actions that occur. The output will show the progress of the system and the sequence of JMF controller events that are occurring. Try to match what you see with what you would expect based upon my lectures.

You will notice that an early line documents that the JMF is preparing a file called jmf.log in your directory. The JMF always write such a file. When you have exited the SimplePlayer, read through the jmf.log file. It documents the assembling of the player, make sure you understand what it says.

### 3.1.2 Running Remotely on stonkin

Start a new window on stonkin as described above. If necessary, source the /dcs/dap/jmf.bashrc file in that new window. Change to the directory in which you are working. Now run SimplePlayer again. (You will NOT need to re-compile, 'cuz it's Java and remember "write once, run anywhere" :-) ).

You will notice that you get no sound. Why not ?? Check the output display in the window in which the software was started. You should be able to locate the problem. You will also find information in the jmf.log file that will have been created.

### 3.1.3 Back on your local workstation with kayaker.avi

Now, return to the windows on your local workstation. Open SimplePlayer.java in an editor and locate the line that specifies the filename of the movie. Alter it to refer to the file kayaker.avi

Recompile and re run the program. Now, you should this time hear sound, but where's the video display?

Check the messages in the window and check jmf.log. Why did we not get video?

### 3.1.4 And on stonkin with kayaker.avi

Now, in your window on stonkin, run the edited version of SimplePlayer that is accessing kayaker.avi. You won't get sound for the same reasons as you discovered a couple of sections ago, but what happens to video ?? why?? Check the output and the jmf.log file.

## 3.2 Using SimpleControls.java

### 3.2.1 Investigating the Control Panel

Compile and run the program SimpleControls.java. You will notice that you now have a control panel displayed on the screen. You need to click the "play" arrow at the bottom left of the Player display to start the movie.

Hopefully, video and and audio both play fine. Check the output and the jmf.log file and make sure you understand that what they say matches to what you see and hear.

Now investigate the effect of the controls. They behave somewhat like a domestic VCR. This is the default controlpanel for this player. It is possible for a programmer to also add his own controls instead if desired. Investigate the effect of clicking with first the right button and then the left button on all the components of the controlpanel. You should find a way of popping up an information display that supplies information such as codecs in use, bandwidth, framerate and so on, along with other features. I've also put a large "Plugin Viewer" button so you can get a visual display of the player components and their interconnection.

### 3.2.2 Using SimpleControls.java from Browser.java

You will find another program, Browser.java in the filestore. Compile that and then run it. It provides a simple window requesting a media locator for a file. Files must be specified in the format

```
file:sample.mov
```

looking a bit like a url. In fact, you can actually supply web URLs as well. There are some sample media files on the web on the following URLs.

```
http://www.aber.ac.uk/~dap/cs25610/hide.au
http://www.aber.ac.uk/~dap/cs25610/test.au
http://www.aber.ac.uk/~dap/JMF/sample.mov
```

Look at the output you get and the jmf.log file created, when running each of these, or when you provide file: media locators for the other media files in your filestore.

## 3.3 WaitPlayer.java

WaitPlayer uses a slightly different approach to deal with the asynchronous events used by the JMF. Try compiling and running WaitPlayer. You will also notice that, although it does not "autoplay", it does always display the first frame of the movie. Look at the Java code and see if you can spot why.

You will also notice that I've added a "Rewind and Play" button at the top of the display. What happens if you click it while the movie is playing? Look at the code and discover why it behaves like that, and how it achieves the rewind and play action when allowed.

## 3.4 Capturing and Conversion

As I've said in lectures, unfortunately the Intel/Solaris student workstations cannot conduct capturing as Sun have not yet released a Performance Pack version of the JMF for Intel/Solaris. However, there are related things you can do.

### 3.4.1 Investigating the PlugIns and CaptureDevices

Try compiling LookPlugins.java and running it on both your Intel/Solaris workstation and remotely on stonkin. Is the output the same ? if not, why not?

Try compiling ListDevices.java and running that on the two environments. What happens and why?

### 3.4.2 Media Conversion

As you saw above, the movie kayaker.avi would not show video on your Intel/Solaris workstation. Use Browser to remind yourself what happened. Use the Plugin Viewer to see what components have been used to make the player and thus what media streams must be included in the movie.

Now, look at the code of Convert.java. You will notice that it is clearly trying to make efforts to convert the kayaker.avi file into a different format. Compile and run the code on your Intel/Solaris workstation.

What happens?

It probably apparently executed quite quickly. Try to play the newfile.mov file that it created. Does it work? No? What went wrong then ? Look at the output that was generated and the jmf.log file produced. Have you spotted the problems?

Now, start a window on stonkin and run Convert.java again in that. What happens?? Click with the left button on the right hand symbol that looks a bit like frames of a film. An information box will pop up. Investigate what that is telling you. Spotted it ?? Yes, its running, but taking an absolute age. If you have got the patience it will eventually convert the file into a new format. It will probably take five minutes or more!

I cooked one earlier :-) and if you access the URL

```
http://www.aber.ac.uk/~dap/JMF/newfile.mov
```

using (say) Browser, then you should be able to both watch and hear the canoe going over the waterfall.

### 3.5 And a bit of fun to end

I've managed to locate a set of movies provided as samples with Windows 95. I've placed them in a directory and the files can be accessed via the medialocators

```
file:/dcs/dap/CS25610/JMF/Movies/welcome1.avi
file:/dcs/dap/CS25610/JMF/Movies/welcome2.avi
file:/dcs/dap/CS25610/JMF/Movies/welcome3.avi
file:/dcs/dap/CS25610/JMF/Movies/weezer.avi
file:/dcs/dap/CS25610/JMF/Movies/robroy.avi
file:/dcs/dap/CS25610/JMF/Movies/goodtime.avi
```

I suggest you use Browser as an easy way to see the effect. I was quite impressed, especially when you think the version of the JMF you are using on the Intel/Solaris workstations is all written in Pure Java !

## 4 Effort Allocated to the Worksheet

You may not complete all of this worksheet in your two hour practical this week. You should expect to spend some “own time” work to complete the worksheet.

## 5 Assessment of this Worksheet

These worksheets for the modules CS25610 and CSM5610 are NOT assessed.

The worksheets and the demonstrated practicals are provided as part of the educational offering of the module. The examination questions for these modules may be based on material covered in practicals in addition to lecture material and background reading drawn to your attention by the lecturers or by links on the course materials web site.