
The Monte Media Handbook

Version 0.3

Werner Randelshofer, 2012-04-01



© 2012 Werner Randelshofer.
This work is licensed under a Creative Commons Attribution 3.0 Unported License.

Contents

Introduction to Monte Media.....	3
License.....	4
Getting Started.....	5
Still Images.....	6
Audio.....	6
Movies.....	6
Reading video frames from a movie file.....	6
Writing video frames into a movie file.....	6
Using Monte Media codecs with JMF.....	7
References.....	8

Introduction to Monte Media

Monte Media is a Java library for processing media data. Supported media formats include still images, video, audio and meta-data.

License

Monte Media is Copyright 2011-2012 © Werner Randelshofer.
All Rights Reserved.

Monte Media can be used for free under the terms of Creative Commons Attribution 3.0 [1].

You are free:

- to Share — to copy, distribute and transmit the work
- to Remix — to adapt the work
- to make commercial use of the work

Under the following conditions:

- Attribution — You must attribute the work in the manner specified by the author or licensor (but not in any way that suggests that they endorse you or your use of the work).

Development and bug fixes of Monte Media are subject to fees.
Unless you do them on your own, of course.

Getting Started

Download the latest release of Monte Media from
<http://www.randelshofer.ch/monte>

You should get a folder which contains the following files and folders:

- monte.jar
- JavaDoc
- monte-src.zip

Essentially, all you need is the file monte.jar. Put it into the class-path of your application, and you are done.

The JavaDoc folder provides an API documentation.

The monte-src.zip file contains a NetBeans project with all the source code. You will need it only if you want to do changes in the library.

Still Images

Audio

Movies

Reading video frames from a movie file

To read video frames from a movie file, you have to perform the following steps:

- Get a `MovieReader` from the Registry.
- Determine which track contains video media.
- Get a Codec from the Registry.
- In a while-loop: read the media data into a `Buffer`, and decode the `Buffer` into a `BufferedImage`.

The example code below shows how to read an array of `BufferedImages` from a movie file.

```
BufferedImage[] readMovie(File file) throws IOException {
    ArrayList<BufferedImage> frames=new ArrayList<BufferedImage> ();

    MovieReader in = Registry.getInstance().getReader(file);

    Format format = new Format(DataClassKey, BufferedImage.class);

    int track = in.findTrack(0, new Format(MediaTypeKey,MediaType.VIDEO));
    Codec codec=Registry.getInstance().getCodec(in.getFormat(track), format);

    try {
        Buffer inbuf = new Buffer();
        Buffer codecbuf = new Buffer();
        do {
            in.read(track, inbuf);
```

```
        codec.process(inbuf, codecbuf);
        if (!codecbuf.isFlag(BufferFlag.DISCARD)) {
            frames.add(Images.cloneImage((BufferedImage)codecbuf.data));
        }

    } while (!inbuf.isFlag(BufferFlag.END_OF_MEDIA));
} finally {
    in.close();
}

return frames.toArray(new BufferedImage[frames.size()]);
}
```

Writing video frames into a movie file

To write video frames from into movie file, you have to perform the following steps:

- Get a `MovieWriter` from the Registry.
- Add a track with the desired video format to the `MovieWriter`.
- Create a Buffer
- In a for-loop: encode a `BufferedImage` into the Buffer, and write it into the `MovieWriter`.

The example code below shows how to write an array of `BufferedImages` into a movie file.

```
void writeMovie(File file, BufferedImage[] frames) throws IOException {
    MovieWriter out = Registry.getInstance().getWriter(file);

    Format format = new Format(MediaTypeKey, MediaType.VIDEO, //
        EncodingKey, ENCODING_AVI_MJPEG,
        FrameRateKey, new Rational(30, 1), //
        WidthKey, frames[0].getWidth(), //
        HeightKey, frames[0].getHeight(), //
        DepthKey, 24
    );

    int track = out.addTrack(format);

    try {
        Buffer buf = new Buffer();
        buf.format = new Format(DataClassKey, BufferedImage.class);
        buf.sampleDuration = format.get(FrameRateKey).inverse();
        for (int i = 0; i < frames.length; i++) {
            buf.data = frames[i];
            out.write(track, buf);
        }
    } finally {
        out.close();
    }
}
```


Using Monte Media codecs with JMF

Monte Media currently provides the following codes for JMF:

- `org.monte.media.jmf.codec.video.TSCCCodec`
Supports encoding and decoding of video frames with the Techsmith Screen Capture Codec (TSCC)
- `org.monte.media.jmf.codec.video.ZMBVDecoder`
Supports decoding of video frames with the DosBox Capture Codec (“Zip Motion Block Video” ZMBV)

To use one of these codecs with JMF, include `monte.jar` in the class path, and then register the desired codec with the JMF Registry Editor as shown in Figure 1.

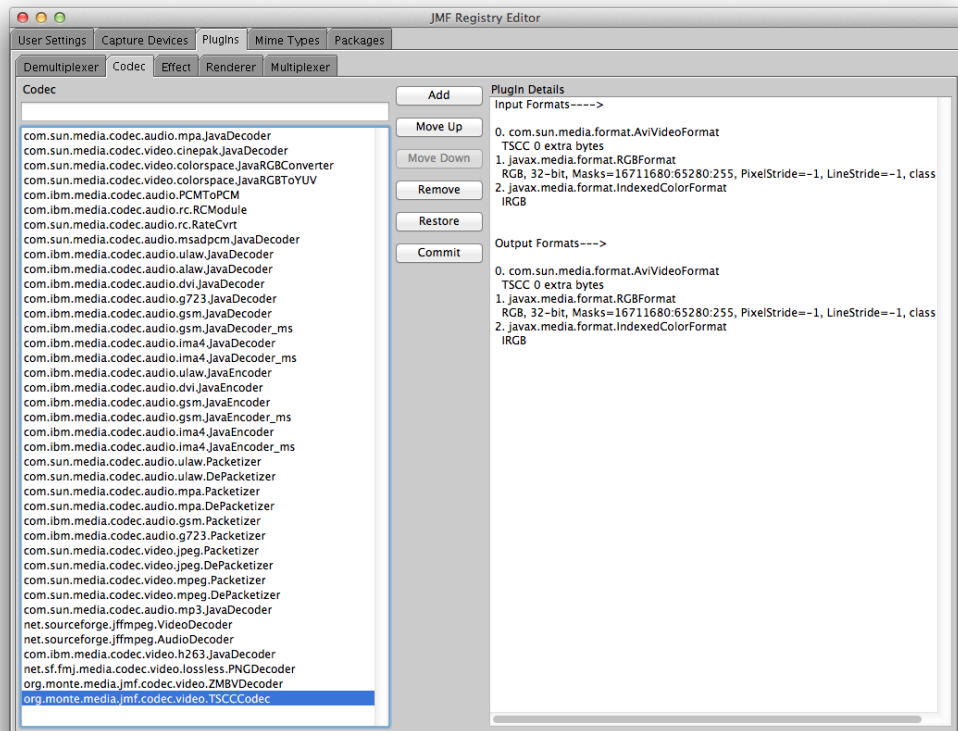


Figure 1: Registering a new codec with the JMF Registry Editor

References

- [1] Creative Commons Attribution 3.0
<http://creativecommons.org/licenses/by/3.0/>
- [2] Java Media Framework API (JMF)
<http://www.oracle.com/technetwork/java/javase/tech/index-jsp-140239.html>