

# im4java

## Table of contents

1 Introduction.....	2
2 License.....	2
3 Description.....	2
4 Features.....	2
5 Download.....	3
6 Maven-Repository.....	3
7 Mailing Lists.....	3
8 Projects using im4java.....	3

**Note:**

The current version of im4java is 1.3.1

## 1. Introduction

im4java is a pure-java interface to the ImageMagick commandline. For details about the project, please visit the [main project page](#).

## 2. License

im4java is licensed under the LGPL (see file COPYING.LIB in the downloaded packages). Since im4java is a wrapper library, you should also make sure that you comply to the individual license-statements of all the tools you use indirectly through im4java.

## 3. Description

im4java is (to my knowledge) the second java interface to ImageMagick. It is not meant as a replacement for [JMagick](#), but as an addition.

JMagick is a thin JNI layer above the ImageMagick C-API. im4java in contrast just generates the commandline for the ImageMagick commands and passes the generated line to the selected IM-command (using the `java.lang.ProcessBuilder.start()`-method).

Disadvantages of the approach: you are limited to the capabilities of the IM commands. With JMagick, you have access to the low-level interface of IM and therefore you have a very detailed control of the processing of images. And you have better performance.

Advantages of im4java: the interface of the IM commandline is quite stable, so your java program (and the im4java-library) will work across many versions of IM. im4java also provides a better OO interface (the "language" of the IM-commandline with it's postfix-operation notation translates very easily into OO-notation). And most important: you can use im4java everywhere JMagick can't be used because of the [JNI hazard](#) (e.g. java application servers).

## 4. Features

- supports most ImageMagick commands
- reuse a commandline multiple times for different set of images
- IM options and operators translate into similar method-names, e.g.

```
.resize(1024, null)      -resize 1024          -> .resize(1024) or
                        -resize 1024x768       -> .resize(1024,768)
                        -resize x768           -> .resize(null,768)
-sigmoidal-contrast 20 -> .sigmoidalContrast(20.)
-append                 -> .append()
```

```
+append                                -> .p_append( )
```

- you can pipe input to and from ImageMagick commands
- BufferedImage support, i.e you can use BufferedImages as input to IM commands and pipe output of IM commands into a BufferedImage
- asynchronous execution of commands possible
- parallel processing of multiple input-images
- the architecture supports arbitrary commandline-tools
- supports GraphicsMagick
- supports jpegtran
- supports ufraw
- supports exiftool
- supports dcraw

## 5. Download

Source and binary packages are available for download. See the [download area](#) of the im4java-project. The current version is **1.3.1**.

You can find a comprehensive list of changes in the [release notes](#) and all the details in the file *ChangeLog*.

## 6. Maven-Repository

Thanks to the effort of users, the im4java-library is also available from [maven](#). An alternative site is [here](#) (kudos to the people at cedarsoft). Note that maven is not supported by the im4java-project due to lack of expertise.

The necessary pom-entry for the alternative repository is:

```
<repositories>
  <repository>
    <id>cedarsoft-thirdparty</id>
    <url>http://maven.cedarsoft.com/content/repositories/thirdparty/</url>
  </repository>
</repositories>
```

## 7. Mailing Lists

Currently (due to low traffic and due to the same topic), the JMagick and the im4java projects share the same mailing-lists. Please go to the project page of [JMagick](#) and register to the jmagick-users mailing list.

## 8. Projects using im4java

- <http://natbraille.free.fr>

