# ImageMagick Media Filters

## ImageMagick Media Filters

1 ImageMagick Media Filters
- 1.1 Overview
- 1.2 Installation
- 1.3 DSpace Configuration
  - 1.3.1 Thumbnail Dimensions
  - 1.3.2 Conversion Utility Path
  - 1.3.3 Supported file formats
  - 1.3.4 Overwriting Existing Thumbnails
  - 1.3.5 Flatten
  - 1.3.6 ICC Profiles
  - 1.3.7 Override ImageMagick Default Density
- 1.4 Additional Customization
- 1.5 Possible Errors / Issues
  - 1.5.1 "convert.im6: not authorized" errors
  - 1.5.2 "convert-im6.q16: cache resources exhausted" errors

As of DSpace 7.6, the ImageMagick media filter also supports creating thumbnails of video (MP4) files, provided that "ffmpeg" is installed locally.  See instructions below.

### Overview

The ImageMagick Media Filters provide consistent, high quality thumbnails for image bitstreams, PDF bitstreams and video (MP4) bitstreams.

These filters require a separate software installation of the conversion utilities: ImageMagick, Ghostscript (to support PDF thumbnails) and/or ffmpeg (to support MP4 thumbnails).

The media filters use the library im4java to invoke the conversion utilities.  This library constructs a conversion command launches a sub-process to perform the generation of media files.

### Installation

Before ImageMagick Media Filters can be used, you must setup ImageMagick (and optionally Ghostscript) as follows:

1. Install ImageMagick on your server.  The installation process differs based on your operating system.  For example, on Debian/Ubuntu, it's similar to this:

   ```
   apt-get install imagemagick
   ```

2. If you wish to generate PDF thumbnails, install Ghostscript on your server. The installation process differs based on your operating system.  For example, on Debian/Ubuntu, it's similar to this:

   ```
   apt-get install ghostscript
   ```

3. (New in 7.6) If you wish to generate MP4 (video) thumbnails, install FFmpeg on your server. The installation process differs based on your operating system.  For example, on Debian/Ubuntu, it's similar to this:

   ```
   apt-get install ffmpeg
   ```

4. The ImageMagick, Ghostscript, and FFmpeg executables should be accessible from the same directory (e.g. `/usr/bin`)
   a. This directory MUST be defined in the `org.dspace.app.mediafilter.ImageMagickThumbnailFilter.ProcessStarter` configuration as describe below.

### DSpace Configuration

In the **filter.plugins** section of your **dspace.cfg** (or local.cfg) file, specify the ImageMagick media filters you wish to use.

**local.cfg**

```
# Make sure to always keep this plugin enabled if you want to support search within text documents
filter.plugin = Text Extractor

# NOTE: When "ImageMagick Image Thumbnail" is enabled, the default "JPEG Thumbnail" should NOT be enabled
filter.plugins = ImageMagick Image Thumbnail

# NOTE: When "ImageMagick PDF Thumbnail" is enabled, the default "PDFBox JPEG Thumbnail" should NOT be enabled
# Requires Ghostscript to also be installed
filter.plugins = ImageMagick PDF Thumbnail

# New in 7.6, this will generate thumbnails from video files.
# Requires ffmpeg to also be installed
filter.plugins = ImageMagick Video Thumbnail
```

This will activate the following settings which are already present in **dspace.cfg** (these do NOT need to be added, as they already exist)

```
plugin.named.org.dspace.app.mediafilter.FormatFilter = org.dspace.app.mediafilter.
ImageMagickImageThumbnailFilter = ImageMagick Image Thumbnail
plugin.named.org.dspace.app.mediafilter.FormatFilter = org.dspace.app.mediafilter.ImageMagickPdfThumbnailFilter
= ImageMagick PDF Thumbnail
plugin.named.org.dspace.app.mediafilter.FormatFilter = org.dspace.app.mediafilter.
ImageMagickVideoThumbnailFilter = ImageMagick Video Thumbnail
```

These media filters contain the several properties which can be configured.

### Thumbnail Dimensions

The following properties are used to define the dimensions of the generated thumbnails:

```
# maximum width and height of generated thumbnails
thumbnail.maxwidth  = 80
thumbnail.maxheight = 80
```

### Conversion Utility Path

The following property provides a path to the ImageMagick (convert), GhostScript (ghostscript), and ffmpeg utilities.

```
org.dspace.app.mediafilter.ImageMagickThumbnailFilter.ProcessStarter = /usr/bin
```

### Supported file formats

Each of these ImageMagick filters has its own list of configurable file formats.  The defaults are usually best, but may be updated if you have custom bitstream formats.  These settings already exist in your dspace.cfg.

```
filter.org.dspace.app.mediafilter.ImageMagickImageThumbnailFilter.inputFormats = BMP, GIF, PNG, JPG, TIFF,
JPEG, JPEG 2000
filter.org.dspace.app.mediafilter.ImageMagickPdfThumbnailFilter.inputFormats = Adobe PDF
filter.org.dspace.app.mediafilter.ImageMagickVideoThumbnailFilter.inputFormats = Video MP4
```

### Overwriting Existing Thumbnails

The ImageMagick media filters can differentiate thumbnails created by the DSpace default thumbnail generator and thumbnails that were manually uploaded by a user.  The media filter reads the bitstream description field to make this determination. A regular expression can be provided to define the set of thumbnails that should be overwritten by the ImageMagick thumbnail generator. Thumbnail descriptions matching this pattern will be overwritten even if the -f option is not passed to the filter media process.

```
org.dspace.app.mediafilter.ImageMagickThumbnailFilter.replaceRegex = ^Generated Thumbnail$
```

The ImageMagick media filter will use the bitstream description field to identify bitstreams that it has created using the following setting. Bitstreams containing this label will be overwritten only if the -f filter is applied.

```
org.dspace.app.mediafilter.ImageMagickThumbnailFilter.bitstreamDescription = IM Thumbnail
```

Thumbnail descriptions that do not match either of the patterns listed above are presumed to be manually uploaded thumbnails.  These thumbnails will not be replaced even if the -f option is passed to the filter media process.

### Flatten

DSpace uses the JPEG format for thumbnails. While JPEG doesn't support transparency, PDF, PNG and other formats do. As those formats are used as outgoing material in DSpace, DSpace has to care about transparency during the generation of the thumbnails. In combinations of specific versions of ImageMagick and Ghostscript it may occur that completely transparent areas will become black. As a solution ImageMagick recommends to flatten images extracted from PDFs before they are stored as JPEG.

Since DSpace 5.2 the ImageMagick media filter flattens thumbnails extracted from PDFs. If you run into problems caused by flattening of the extracted images, you can switch the flattening off by setting the following property in **dspace.cfg** to false:

```
org.dspace.app.mediafilter.ImageMagickThumbnailFilter.flatten = false
```

### ICC Profiles

PDFs optimized for physical printing often use the CMYK color space. On the web, however, the de facto color system is sRGB. By default, DSpace's ImageMagick-based thumbnailing system will create thumbnails that use the same color space as the source PDF. Most web browsers are not able to correctly display images that use the CMYK color space, which leads to images with visibly inaccurate colors.

If you are using Ghostscript version 9 or above, it is possible for DSpace to correctly convert images from CMYK to sRGB by providing it with appropriate ICC color profiles to use during thumbnail creation. Default ones are provided by most Ghostscript installations (version 9 or above). The following configuration options tell DSpace where those ICC profiles are located.

```
# org.dspace.app.mediafilter.ImageMagickThumbnailFilter.cmyk_profile = /usr/share/ghostscript/9.18/iccprofiles
/default_cmyk.icc
# org.dspace.app.mediafilter.ImageMagickThumbnailFilter.srgb_profile = /usr/share/ghostscript/9.18/iccprofiles
/default_rgb.icc
```

You may need to adjust those paths for your OS or the version of Ghostscript that you have.

Providing ICC profiles to ImageMagick is optional. If these configuration properties are unset, no profiles will be supplied to ImageMagick, and thumbnails produced from PDFs using the CMYK color space will also use CMYK. The transformation from CMYK to RGB is optional.

### Override ImageMagick Default Density

It is possible to override ImageMagick's default density of 72 DPI when creating PDF thumbnails. This increases the quality of resulting thumbnails at the expense of slightly longer execution times and higher memory usage. Any integer over 72 will help, but recommend 144 for a "2x" supersample. See the following configuration option in `dspace.cfg` :

```
# org.dspace.app.mediafilter.ImageMagickThumbnailFilter.density = 144
```

The effect is most notable on PDFs with a lot of text, gradients, or curved lines. See the pull request implementing this feature for more information and comparisons.

## Additional Customization

The ImageMagick conversion software provides a large number of conversion options.  Subclasses of these media filters could be written to take advantage of the additional conversion properties available in the software.

Note: The PDF thumbnail generator is hard-coded to generate a thumbnail from the first page of the PDF.

## Possible Errors / Issues

### "convert.im6: not authorized" errors

On Ubuntu (possibly other OSes), you may see errors like these when attempting to generate PDF thumbnails:

```
ERROR filtering, skipping bitstream:
        Item Handle: 1234/5678
        Bundle Name: ORIGINAL
        File Size: 30406135
        Checksum: c1df4b3a4755e9bed956383b61fc5042 (MD5)
        Asset Store: 0
org.im4java.core.CommandException: org.im4java.core.CommandException: convert.im6: not authorized `/tmp/impdfthumb6294641076817830415.pdf' @ error/constitute.c/ReadImage/454.

OR

org.im4java.core.CommandException: convert-im6.q16: attempt to perform an operation not allowed by the security policy 'PDF' @ error/constitute.c/IsCoderAuthorized/421
```

These may be caused by a change in your ImageMagick policy configuration on your server.

In Ubuntu, the default "policy.xml" was recently updated to **exclude** all Ghostscript formats (including PDF, PS, etc).  See this ticket: https://bugs.launchpad.net/ubuntu/+source/imagemagick/+bug/1796563

- This exclusion was implemented to workaround a security vulnerability in Ghostscript reported here: https://www.kb.cert.org/vuls/id/332928
- According to that vulnerability report, this was patched in Ghostscript v9.24 (or above)

The newly added lines in the `/etc/ImageMagick/policy.xml` are these ones:

```
<!-- disable ghostscript format types -->
<policy domain="coder" rights="none" pattern="PS" />
<policy domain="coder" rights="none" pattern="EPS" />
<policy domain="coder" rights="none" pattern="PDF" />
<policy domain="coder" rights="none" pattern="XPS" />
```

To fix the error above requires you to re-enable ImageMagick to process Ghostscript format types. That can be done by simply commenting out those new "policy" lines in the configuration file (surround them with <!-- and --> to comment out)

Be aware that you MUST ensure you are running Ghostscript v9.24 or later to ensure that you are not at risk for the above security vulnerability in older versions of Ghostscript.

### "convert-im6.q16: cache resources exhausted" errors

On Debian, and possibly other OSes, you may see errors like these when attempting to generate video thumbnails (especially if video files are big):

```
File: video.mp4.jpg
ERROR filtering, skipping bitstream:
        Item Handle: 1234/5678
        Bundle Name: ORIGINAL
        File Size: 146761357
        Checksum: 735ceb1b6b249afc84a5bb1b87ae0c02 (MD5)
        Asset Store: 0
org.im4java.core.CommandException: convert-im6.q16: cache resources exhausted `/tmp/magick-64dziU-1nQJjQHZYu4_R1fFP4l9en5iL.pam' @ error/cache.c/OpenPixelCache/4095.
```

These may be caused by too conservative resource policies in your `policy.xml` file. As an example, default values are located at `/etc/ImageMagick-6/policiy.xml` in Debian 11 (Bullseye), and are:

**/etc/ImageMagick-6/policiy.xml**

```
<policymap>
  <!-- <policy domain="resource" name="temporary-path" value="/tmp"/> -->
  <policy domain="resource" name="memory" value="256MiB"/>
  <policy domain="resource" name="map" value="512MiB"/>
  <policy domain="resource" name="width" value="16KP"/>
  <policy domain="resource" name="height" value="16KP"/>
  <!-- <policy domain="resource" name="list-length" value="128"/> -->
  <policy domain="resource" name="area" value="128MP"/>
  <policy domain="resource" name="disk" value="1GiB"/>
```

To avoid the `cache resources exhausted` error, try increasing the resource limits policies. You may want to start by increasing the `memory` and `disk` policies (disk cache is used when the memory limit is reached). The actual values have to be adjusted depending on the size of your video bitstreams and the actual resources available in your installation. For example:

**/etc/ImageMagick-6/policiy.xml**

```
<policymap>
  <!-- <policy domain="resource" name="temporary-path" value="/tmp"/> -->
  <policy domain="resource" name="memory" value="4GiB"/> <!-- memory limit increased from 256MiB to 4GiB -->
  <policy domain="resource" name="map" value="512MiB"/>
  <policy domain="resource" name="width" value="16KP"/>
  <policy domain="resource" name="height" value="16KP"/>
  <!-- <policy domain="resource" name="list-length" value="128"/> -->
  <policy domain="resource" name="area" value="128MP"/>
  <policy domain="resource" name="disk" value="4GiB"/> <!-- disk limit increased from 1GiB to 4GiB -->
```

For a detailed description of the ImageMagick limits, see https://imagemagick.org/script/command-line-options.php#limit.

Once the limits are properly set, a successful execution of the filter should show a message similar to:

```
File: video.mp4.jpg
FILTERED: bitstream 12345678-abcd-efgh-ijkl-1234567890ab (item: 1234/5678) and created 'video.mp4.jpg'
```