
MediaVorus Documentation

Release 0.1

Romain Neutron

August 26, 2014

1	Example	3
2	Goals	5
3	API	7
3.1	Guesser	7
3.2	Medias	7
3.3	Media serialization	9
3.4	Silex Service Provider	10

MediaVorus is a small PHP library wich provide a set of tools to extract every technical information from multimedia files.

Example

```
<?php

use MediaVorus\MediaVorus;
use MediaVorus\MediaVorus;
use MediaVorus\Media\Image;

$mediavorus = MediaVorus::create();
$Media = $mediavorus->guess('tests/files/CanonRaw.CR2');

if($Media instanceof Image)
{
    echo sprintf("Image was taken with a %f shutter speed", $Media->getShutterSpeed());
}
```

Goals

MediaVorus is a small PHP library wich provide a set of tools to deal with multimedia files.

The aim is to provide an abstract layer between the program and the multimedia file.

First, the need is to analyze multimedia files and get their properties.

In a very next future, we would add metadata mapper to handle various configurations.

3.1 Guesser

```
<?php

use MediaVorus\MediaVorus;

$Media = MediaVorus::guess('tests/files/ExifTool.jpg');

//Returns a \Doctrine\Common\Collections\ArrayCollection of Medias
$MediaCollection = MediaVorus::inspectDirectory($dir, $recursive);
```

3.2 Medias

3.2.1 Media\DefaultMedia

Default Media is the Default container. This object provides GPS informations :

```
<?php

use MediaVorus\MediaVorus;
use MediaVorus\Media;

$Media = MediaVorus::guess('somefile.smf');

if($Media instanceof Media\DefaultMedia)
{
    /**
     * Returns the longitude as a described above
     */
    $Media->getLongitude();
    /**
     * Returns the longitude as a described above
     */
    $Media->getLatitude();
    /**
     * Returns Longitude reference (West or East) and equals to one of the
     * Media\DefaultMedia::GPSREF_LONGITUDE_* constants
     */
    $Media->getLongitudeRef();
}
```

```
/**
 * Returns Latitude reference (North or South) and equals to one of the
 * Media\DefaultMedia::GPSREF_LATITUDE_* constants
 */
$Media->getLatitudeRef();
}
```

3.2.2 Media\Image

Media Image extends the default media. It has much more methods and provides the following informations :

```
<?php
```

```
use MediaVorus\MediaVorus;
use MediaVorus\Media;

$Media = MediaVorus::guess('tests/files/ExifTool.jpg');

if($Media instanceof Media\Image)
{
    /**
     * It extends the DefaultMedia
     */
    assert($Media instanceof Media\DefaultMedia);
    /**
     * Returns the width (int)
     */
    $Media->getWidth();
    /**
     * Returns the height (int)
     */
    $Media->getHeight();
    /**
     * Returns the number of channels (int)
     */
    $Media->getChannels();
    /**
     * Returns the focal length (string), not parsed
     */
    $Media->getFocalLength();
    /**
     * Returns the color depth in bits (int)
     */
    $Media->getColorDepth();
    /**
     * Returns the camera model name (string)
     */
    $Media->getCameraModel();
    /**
     * Returns true if the flash has been fired (bool)
     */
    $Media->getFlashFired();
    /**
     * Returns the aperture (string), not parsed
     */
    $Media->getAperture();
    /**
```

```

    * Returns the shutter speed (string), not parsed
    */
    $Media->getShutterSpeed();
    /**
    * Returns the orientation (string), one of the Media\Image::ORIENTATION_*
    */
    $Media->getOrientation();
    /**
    * Returns the date when the photos has been taken (string), not parsed
    */
    $Media->getCreationDate();
    /**
    * Returns the hyperfocal distance (string), not parsed
    */
    $Media->getHyperfocalDistance();
    /**
    * Returns the ISO value (int)
    */
    $Media->getISO();
    /**
    * Returns the light value (string), not parsed
    */
    $Media->getLightValue();
}

```

3.3 Media serialization

All medias are serializable with [JMS Serializer](#). For example :

```
<?php
```

```

use Doctrine\Common\Annotations\AnnotationRegistry;
use JMS\Serializer\SerializerBuilder;
use MediaVorus\MediaVorus;

AnnotationRegistry::registerAutoloadNamespace(
    'JMS\Serializer\Annotation', __DIR__ . '/vendor/jms/serializer/src'
);

$serializer = SerializerBuilder::create()
    ->setCacheDir(__DIR__ . '/cache')
    ->build();

$mediavorus = MediaVorus::create();
print($serializer->serialize($mediavorus->guess('image.jpg'), 'json'));

```

would result in the following output :

```

{
    "type": "Image",
    "raw_image": false,
    "multiple_layers": false,
    "width": 3264,
    "height": 2448,
    "channels": 3,
    "focal_length": 4.28,

```

```
"color_depth": 8,
"camera_model": "iPhone 4S",
"flash_fired": false,
"aperture": 2.4,
"shutter_speed": 0.05,
"orientation": 90,
"creation_date": "2012:03:16 16:29:09",
"hyperfocal_distance": 2.0773522348635,
"ISO": 400,
"light_value": 4.847996906555,
"color_space": "RGB"
}
```

3.4 Silex Service Provider

MediaVorus comes bundled with its Silex Service Provider. As MediaVorus relies on [PHP-Exiftool](#) and [FFProbe](#), you'll have to register both bundles to use it :

```
<?php

use FFMpeg\FFMpegServiceProvider;
use MediaVorus\MediaVorusServiceProvider;
use PHPExiftool\PHPExiftoolServiceProvider;
use Silex\Application;

$app = new Application();

$app->register(new MediaVorusServiceProvider());
$app->register(new PHPExiftoolServiceProvider());
$app->register(new FFMpegServiceProvider());

// you will now have access to $app['mediavorus']
$video = $app['mediavorus']->guess('/path/to/video/file');
```

3.4.1 API

3.4.2 Use Cases