

# Table of Contents

Format Recipes .....	1
Get a Format .....	1
Get Names .....	1
Read a Raster .....	1
Write a Raster.....	2
Check for a Raster .....	2

# Format Recipes

The Format classes are in the [geoscript.layer](#) package.

A Format is a collection of Rasters.

## Get a Format

Get a Format from a File

```
File file = new File("src/main/resources/earth.tif")
Format format = Format.getFormat(file)
println format.name
```

GeoTIFF

## Get Names

Get names of the Rasters in a Format. Some Formats can contain more than one Raster.

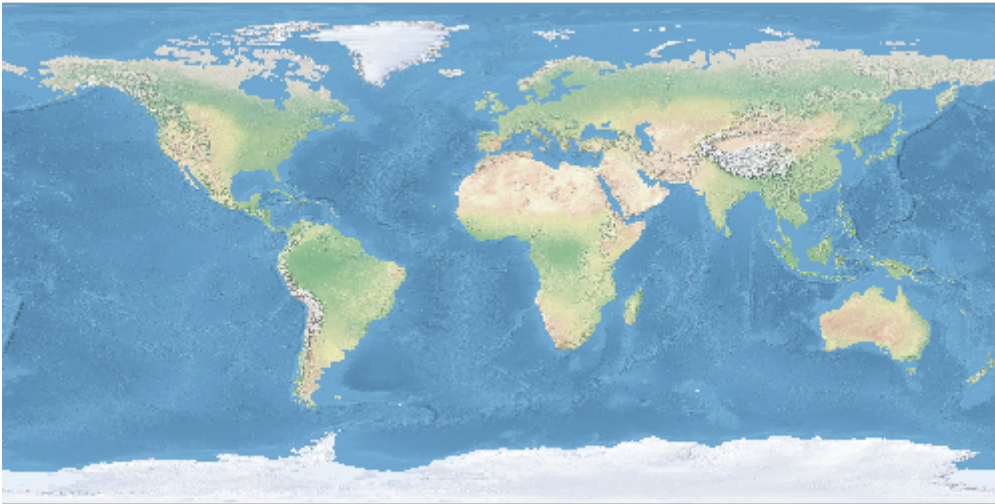
```
File file = new File("src/main/resources/earth.tif")
Format format = Format.getFormat(file)
List<String> names = format.names
names.each { String name ->
    println name
}
```

earth

## Read a Raster

Read a Raster from a File

```
File file = new File("src/main/resources/earth.tif")
Format format = Format.getFormat(file)
Raster raster = format.read("earth")
```

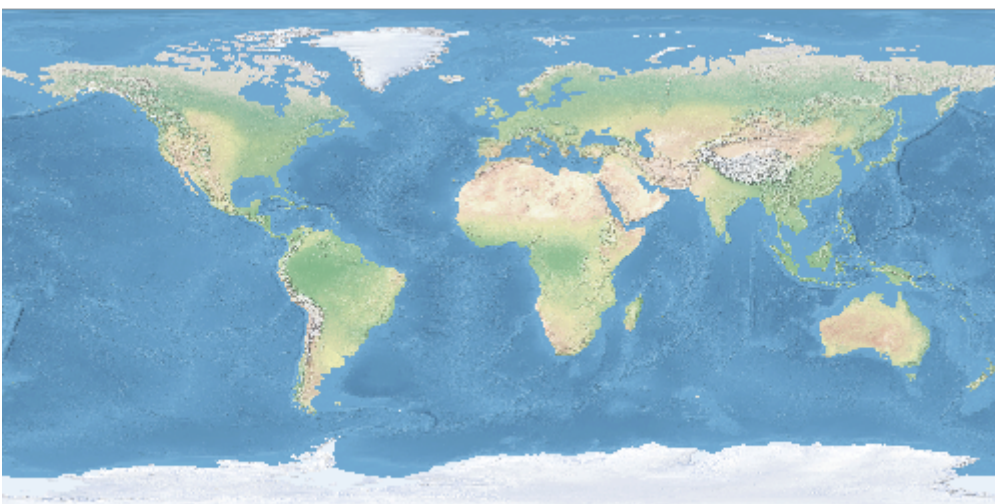


## Write a Raster

Write a Raster to a File

```
File file = new File("src/main/resources/earth.tif")
Format format = Format.getFormat(file)
Raster raster = format.read("earth")

File outFile = new File("target/earth.png")
Format outFormat = Format.getFormat(outFile)
outFormat.write(raster)
Raster outRaster = outFormat.read("earth")
```



## Check for a Raster

Check to see if the Format has a Raster

```
File file = new File("src/main/resources/earth.tif")
Format format = Format.getFormat(file)

boolean hasEarth = format.has("earth")
println "Has raster named earth? ${hasEarth}"

boolean hasWorld = format.has("world")
println "Has raster named world? ${hasWorld}"
```

```
Has raster named earth? true
Has raster named world? false
```