

# 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
GeoTIFF .....	3
WorldImage .....	4
ArcGrid .....	5
NetCDF .....	5
MrSID .....	6
GTopo30 .....	6

# 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
```

## GeoTIFF

### Read

Read a GeoTIFF Raster from a File

```
File file = new File("src/main/resources/earth.tif")
GeoTIFF geotiff = new GeoTIFF(file)
Raster raster = geotiff.read("earth")
```



### Write

Write a GeoTIFF Raster to an ArcGrid Raster

```
File file = new File("src/main/resources/earth.tif")
GeoTIFF geotiff = new GeoTIFF(file)
Raster raster = geotiff.read("earth")

File arcGridFile = new File("target/earth.asc")
ArcGrid arcGrid = new ArcGrid(arcGridFile)
arcGrid.write(raster)
Raster arcGridRaster = arcGrid.read("earth")
```



## WorldImage

### Read

Read a WorldImage Raster from a File

```
File file = new File("src/main/resources/earth.png")
WorldImage worldImage = new WorldImage(file)
Raster raster = worldImage.read("earth")
```



## ArcGrid

### Read

Read an ArcGrid Raster from a File

```
File file = new File("src/main/resources/raster.asc")
ArcGrid arcGrid = new ArcGrid(file)
Raster raster = arcGrid.read("raster")
```

Read a Grass ArcGrid Raster from a File

```
File file = new File("src/main/resources/grass.arx")
ArcGrid arcGrid = new ArcGrid(file)
Raster raster = arcGrid.read("grass")
```

## NetCDF

### Read

Read a NetCDF Raster from a File

```
File file = new File("src/main/resources/03-N02.nc")
NetCDF netCDF = new NetCDF(file)
Raster raster = netCDF.read("N02")
```

# MrSID

## Read

Read a MrSID Raster from a File

```
File file = new File("src/main/resources/ortho.sid")
MrSID mrsid = new MrSID(file)
Raster raster = mrsid.read("ortho")
```

# GTopo30

## Read

Read a GTopo30 Raster from a File

```
File file = new File("src/main/resources/topo.dem")
GTopo30 gtopo30 = new GTopo30(file)
Raster raster = gtopo30.read("topo")
```