Table of Contents

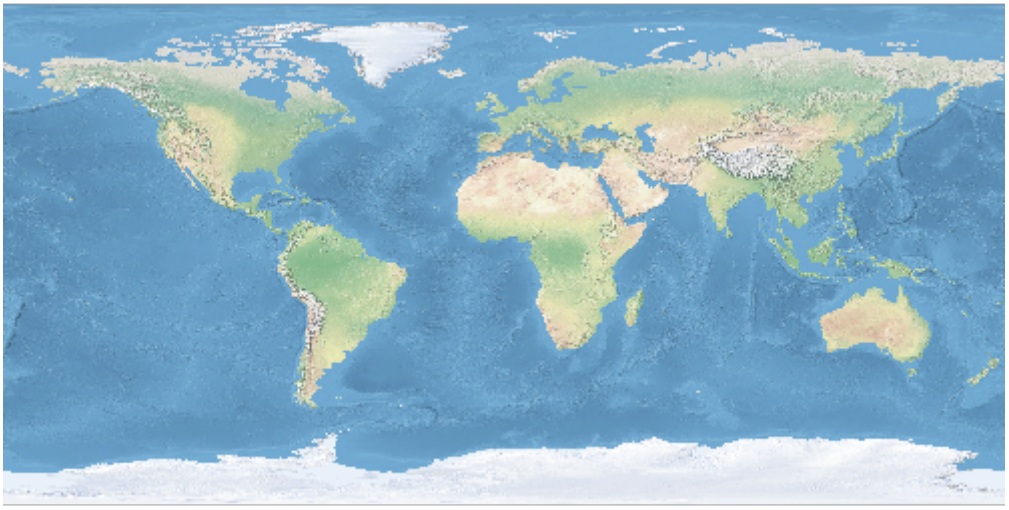
Raster Recipes

The Raster classes are in the **geoscript.layer** package.

Raster Properties

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")
```



Get the Raster's Bounds.

```
Bounds bounds = raster.bounds
println "Bounds: ${bounds}"
```

```
Bounds: (-179.99999999999997,-89.99999999998205,179.99999999996405,90.0,EPSG:4326)
```

Get the Raster's Projection.

```
Projection projection = raster.proj
println "Projection: ${projection}"
```

```
Projection: EPSG:4326
```

Get the Raster's Size.

```
List size = raster.size
println "Size: ${size[0]}x${size[1]}"
```

```
Size: 800x400
```

Get the Raster's number of columns and rows.

```
int cols = raster.cols
int rows = raster.rows
println "Columns: ${cols} Rows: ${rows}"
```

```
Columns: 800 Rows: 400
```

Get the Raster's Bands.

```
List<Band> bands = raster.bands
println "Bands:"
bands.each { Band band ->
    println "  ${band}"
}
```

```
Band:
  RED_BAND
  GREEN_BAND
  BLUE_BAND
```

Get the Raster's block size.

```
List blockSize = raster.blockSize
println "Block size: ${blockSize[0]}x${blockSize[1]}"
```

```
Block size: 800x8
```

Get the Raster's pixel size.

```
List pixelSize = raster.pixelSize
println "Pixel size: ${pixelSize[0]}x${pixelSize[1]}"
```

```
Pixel size: 0.44999999999995505x0.4499999999999551
```

Get more information about a Raster's Bounds.

```
File file = new File("src/main/resources/earth.tif")
Format format = Format.getFormat(file)
Raster raster = format.read("earth")
List<Band> bands = raster.bands
bands.each { Band band ->
    println "${band}"
    println "  Min = ${band.min}"
    println "  Max = ${band.max}"
    println "  No Data = ${band.noData}"
    println "  Is No Data = ${band.isNoData(12.45)}"
    println "  Unit = ${band.unit}"
    println "  Scale = ${band.scale}"
    println "  Offset = ${band.offset}"
    println "  Type = ${band.type}"
}
```

```
RED_BAND
  Min = 0.0
  Max = 255.0
  No Data = [0.0]
  Is No Data = false
  Unit = null
  Scale = 1.0
  Offset = 0.0
  Type = byte

GREEN_BAND
  Min = 0.0
  Max = 255.0
  No Data = [0.0]
  Is No Data = false
  Unit = null
  Scale = 1.0
  Offset = 0.0
  Type = byte

BLUE_BAND
  Min = 0.0
  Max = 255.0
  No Data = [0.0]
  Is No Data = false
  Unit = null
  Scale = 1.0
  Offset = 0.0
  Type = byte
```

Get the minimum and maximum values from a Raster for each band

```
Map extrema = raster.extrema
println "Min values: ${extrema.min} Max values: ${extrema.max}"
```

```
Min value: [56.0, 84.0, 91.0] Max value: [255.0, 255.0, 255.0]
```

Get a Point at the given pixel location.

```
Point point = raster.getPoint(7,9)
println "Geographic location at pixel 7,9 is ${point}"
```

```
Geographic location at pixel 7,9 is POINT (-176.625 85.7249984741211)
```

Get a Pixel location at the given Point.

```
List pixel = raster.getPixel(new Point(-176.625, 85.72499))
println "Pixel coordinates at POINT (-176.625 85.7249984741211) is ${pixel[0]},
${pixel[1]}"
```

```
Pixel coordinates at POINT (-176.625 85.7249984741211) is 7.0, 9.0
```

Determine whether the Raster covers the given Point.

```
boolean containsPoint = raster.contains(new Point(-180, -90))
println "Does raster cover point? ${containsPoint}"
```

```
Does raster cover point? true
```

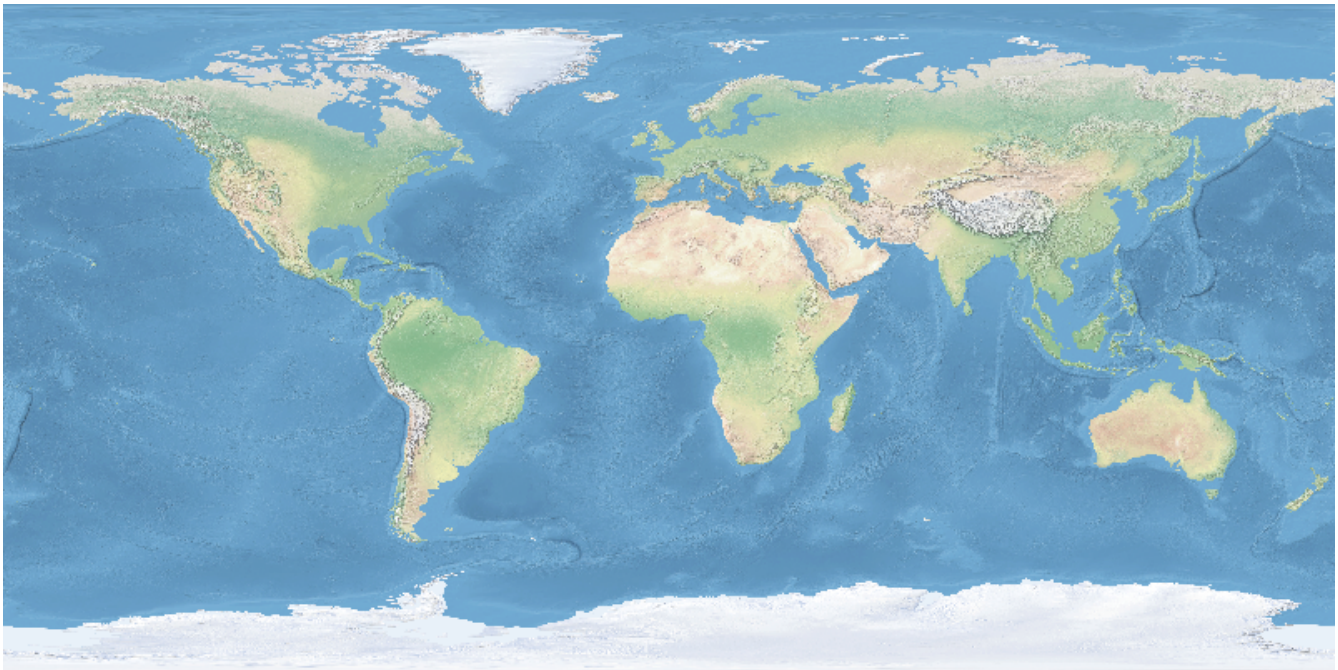
Determine whether the Raster covers the given Pixel.

```
boolean containsPixel = raster.contains(500,600)
println "Does raster cover pixel? ${containsPixel}"
```

```
Does raster cover pixel? false
```

Get a RenderedImage from the Raster

```
RenderedImage image = raster.image
```



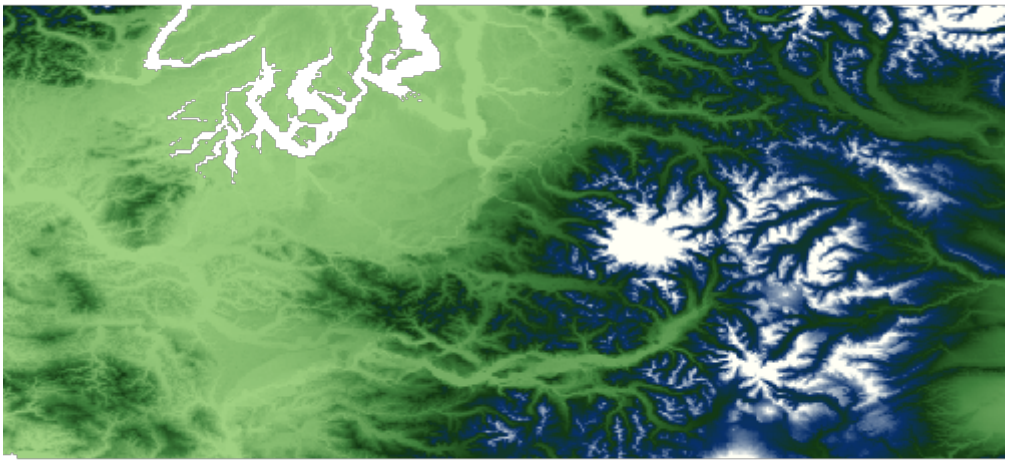
Dispose of the Raster when you are done

```
raster.dispose()
```

Raster Values

Get values from a Raster

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



Get values from a Raster with a Point.

```
double elevation = raster.getValue(new Point(-121.799927,46.867703))
println elevation
```

```
3069.0
```

Get values from a Raster with a Pixel Location.

```
List pixel = [100,200]
elevation = raster.getValue(pixel)
println elevation
```

```
288.0
```

Get neighboring values from a Raster with a Point Location.

```
Map neighborsOfPoint = raster.getNeighbors(new Point(-176.625, 85.72499), 0)
println "Values neighboring POINT (-176.625 85.7249984741211) = ${neighborsOfPoint}"
```

```
Values neighboring POINT (-176.625 85.7249984741211) = [nw:103.0, n:104.0, ne:109.0,
e:109.0, se:111.0, s:110.0, sw:110.0, w:108.0]
```

Get neighboring values from a Raster with a Pixel Location.

```
Map neighborsOfPixel = raster.getNeighbors([7,9], 0)
println "Values neighboring pixel 7,9 = ${neighborsOfPixel}"
```

```
Values neighboring pixel 7,9 = [nw:103.0, n:104.0, ne:109.0, e:109.0, se:111.0,
s:110.0, sw:110.0, w:108.0]
```

Get values from a Raster for a range of pixels in a list of lists.

```
int x = 10
int y = 8
int w = 5
int h = 6
int band = 0
List values = raster.getValues(x, y, w, h, band, false)
println values
```

```
[[1032, 1186, 1340, 1435, 1301], [1143, 1143, 1193, 1224, 1313], [942, 938, 966, 982,
1129], [746, 835, 912, 949, 1028], [723, 948, 1130, 1244, 1211], [673, 890, 1100,
1133, 1024]]
```

Get values from a Raster for a range of pixels in a flat list.

```
List flatValues = raster.getValues(x, y, w, h, band, true)
println flatValues
```

```
[1032, 1186, 1340, 1435, 1301, 1143, 1143, 1193, 1224, 1313, 942, 938, 966, 982, 1129,
746, 835, 912, 949, 1028, 723, 948, 1130, 1244, 1211, 673, 890, 1100, 1133, 1024]
```

Get values from a Raster for a range of pixels as a pretty printed string.

```
String valuesAsString = raster.getValuesAsString(x, y, w, h, band, prettyPrint: true)
println valuesAsString
```

```
-----------------------------------------------
| 1,032.0 | 1,186.0 | 1,340.0 | 1,435.0 | 1,301.0 |
-----------------------------------------------
| 1,143.0 | 1,143.0 | 1,193.0 | 1,224.0 | 1,313.0 |
-----------------------------------------------
|  942.0 |  938.0 |  966.0 |  982.0 | 1,129.0 |
-----------------------------------------------
|  746.0 |  835.0 |  912.0 |  949.0 | 1,028.0 |
-----------------------------------------------
|  723.0 |  948.0 | 1,130.0 | 1,244.0 | 1,211.0 |
-----------------------------------------------
|  673.0 |  890.0 | 1,100.0 | 1,133.0 | 1,024.0 |
-----------------------------------------------
```

Iterate over the cells in a Raster.

```
raster.eachCell(bounds: [0,0,5,5]) { double value, double pixelX, double pixelY ->
    println "${pixelX},${pixelY} = ${value}"
}
```

```
0.0,0.0 = 1061.0
1.0,0.0 = 996.0
2.0,0.0 = 945.0
3.0,0.0 = 960.0
4.0,0.0 = 904.0
0.0,1.0 = 1167.0
1.0,1.0 = 1149.0
2.0,1.0 = 1085.0
3.0,1.0 = 966.0
4.0,1.0 = 862.0
0.0,2.0 = 1112.0
1.0,2.0 = 998.0
2.0,2.0 = 882.0
3.0,2.0 = 775.0
4.0,2.0 = 700.0
0.0,3.0 = 990.0
1.0,3.0 = 850.0
2.0,3.0 = 715.0
3.0,3.0 = 638.0
4.0,3.0 = 654.0
0.0,4.0 = 833.0
1.0,4.0 = 706.0
2.0,4.0 = 611.0
3.0,4.0 = 681.0
4.0,4.0 = 841.0
```

Iterate over a window of cells in a Raster.

```
raster.eachWindow (bounds: [0,0,8,8], window: [2,2]) { Number[][] windowsValues,
double pixelX, double pixelY ->
    println "${pixelX},${pixelY} = ${windowsValues}"
}
```

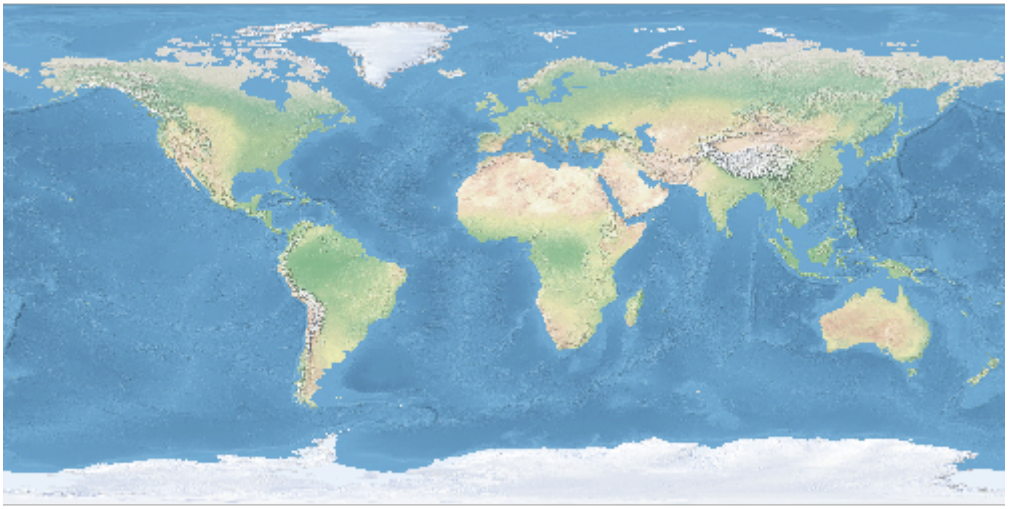
```
0.0,0.0 = [[1061, 996], [1167, 1149]]
1.0,0.0 = [[996, 945], [1149, 1085]]
2.0,0.0 = [[945, 960], [1085, 966]]
3.0,0.0 = [[960, 904], [966, 862]]
4.0,0.0 = [[904, 727], [862, 696]]
5.0,0.0 = [[727, 744], [696, 748]]
6.0,0.0 = [[744, 934], [748, 900]]
7.0,0.0 = [[934, 1099], [900, 1042]]
0.0,1.0 = [[1167, 1149], [1112, 998]]
1.0,1.0 = [[1149, 1085], [998, 882]]
2.0,1.0 = [[1085, 966], [882, 775]]
3.0,1.0 = [[966, 862], [775, 700]]
4.0,1.0 = [[862, 696], [700, 661]]
5.0,1.0 = [[696, 748], [661, 818]]
6.0,1.0 = [[748, 900], [818, 995]]
```

```
7.0,1.0 = [[900, 1042], [995, 1125]]
0.0,2.0 = [[1112, 998], [990, 850]]
1.0,2.0 = [[998, 882], [850, 715]]
2.0,2.0 = [[882, 775], [715, 638]]
3.0,2.0 = [[775, 700], [638, 654]]
4.0,2.0 = [[700, 661], [654, 826]]
5.0,2.0 = [[661, 818], [826, 945]]
6.0,2.0 = [[818, 995], [945, 922]]
7.0,2.0 = [[995, 1125], [922, 1078]]
0.0,3.0 = [[990, 850], [833, 706]]
1.0,3.0 = [[850, 715], [706, 611]]
2.0,3.0 = [[715, 638], [611, 681]]
3.0,3.0 = [[638, 654], [681, 841]]
4.0,3.0 = [[654, 826], [841, 949]]
5.0,3.0 = [[826, 945], [949, 1084]]
6.0,3.0 = [[945, 922], [1084, 1054]]
7.0,3.0 = [[922, 1078], [1054, 1093]]
0.0,4.0 = [[833, 706], [652, 618]]
1.0,4.0 = [[706, 611], [618, 548]]
2.0,4.0 = [[611, 681], [548, 631]]
3.0,4.0 = [[681, 841], [631, 694]]
4.0,4.0 = [[841, 949], [694, 877]]
5.0,4.0 = [[949, 1084], [877, 1018]]
6.0,4.0 = [[1084, 1054], [1018, 1142]]
7.0,4.0 = [[1054, 1093], [1142, 1172]]
0.0,5.0 = [[652, 618], [631, 579]]
1.0,5.0 = [[618, 548], [579, 506]]
2.0,5.0 = [[548, 631], [506, 483]]
3.0,5.0 = [[631, 694], [483, 556]]
4.0,5.0 = [[694, 877], [556, 686]]
5.0,5.0 = [[877, 1018], [686, 825]]
6.0,5.0 = [[1018, 1142], [825, 1004]]
7.0,5.0 = [[1142, 1172], [1004, 1053]]
0.0,6.0 = [[631, 579], [772, 766]]
1.0,6.0 = [[579, 506], [766, 627]]
2.0,6.0 = [[506, 483], [627, 529]]
3.0,6.0 = [[483, 556], [529, 473]]
4.0,6.0 = [[556, 686], [473, 556]]
5.0,6.0 = [[686, 825], [556, 669]]
6.0,6.0 = [[825, 1004], [669, 810]]
7.0,6.0 = [[1004, 1053], [810, 917]]
0.0,7.0 = [[772, 766], [900, 880]]
1.0,7.0 = [[766, 627], [880, 778]]
2.0,7.0 = [[627, 529], [778, 671]]
3.0,7.0 = [[529, 473], [671, 540]]
4.0,7.0 = [[473, 556], [540, 483]]
5.0,7.0 = [[556, 669], [483, 536]]
6.0,7.0 = [[669, 810], [536, 641]]
```

Set values on a 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")

arcGridRaster.eachCell {double value, double x, double y ->
    double newValue = value + 100
    arcGridRaster.setValue([x as int, y as int], newValue)
}

File arcGridAddFile = new File("target/earth_100.asc")
ArcGrid arcGridAdd = new ArcGrid(arcGridAddFile)
arcGridAdd.write(arcGridRaster)
Raster arcGridRasterAdd = arcGridAdd.read("earth_100")

List pixels = [
    [92, 298],
    [393.0, 343.0],
    [795.0, 399.0]
]
pixels.each { List pixel ->
    println "Original: ${raster.getValue(pixel)} New:
${arcGridRasterAdd.getValue(pixel)}"
}
```

```
Original: 97.0 New: 197.0
Original: 96.0 New: 196.0
Original: 237.0 New: 337.0
```



Raster Processing

Crop

Crop a Raster with a Bounds

```
File file = new File("src/main/resources/earth.tif")
Format format = Format.getFormat(file)
Raster raster = format.read("earth")
Raster croppedRaster = raster.crop(new Bounds(-160.927734,6.751896,-34.716797
,57.279043, "EPSG:4326"))
```

Project

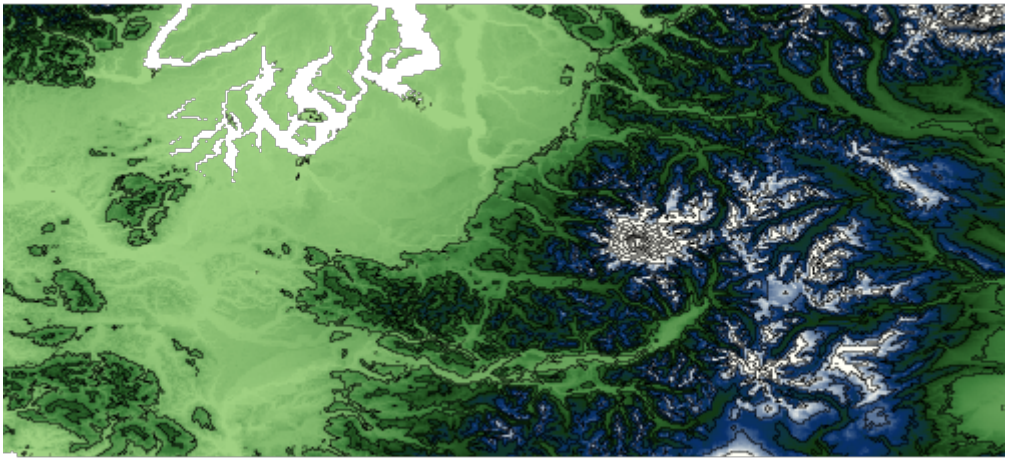
Reproject a Raster to another Projection

```
File file = new File("src/main/resources/earth.tif")
Format format = Format.getFormat(file)
Raster raster = format.read("earth")
Projection projection = new Projection("EPSG:3857")
Raster projectedRaster = raster.crop(projection.geoBounds).reproject(projection)
```

Contours

Create vector contours from a Raster

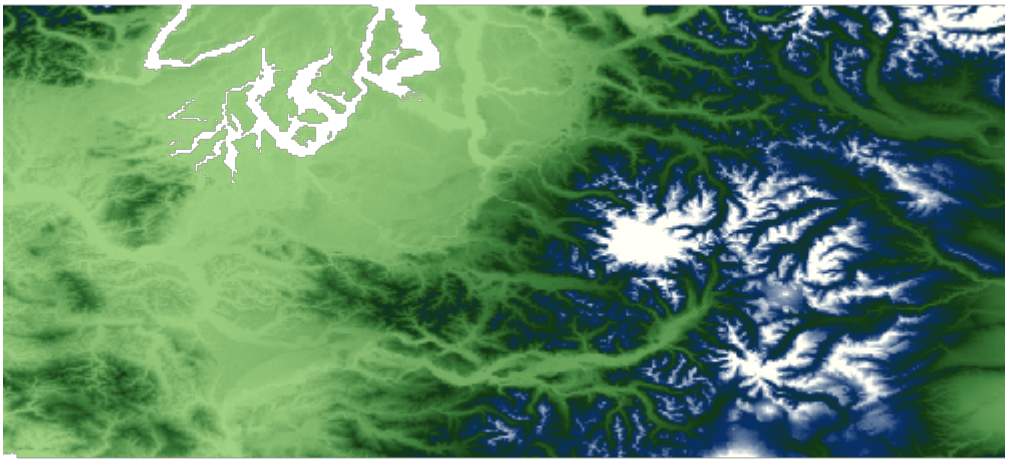
```
File file = new File("src/main/resources/pc.tif")
Format format = Format.getFormat(file)
Raster raster = format.read("pc")
int band = 0
int interval = 300
boolean simplify = true
boolean smooth = true
Layer contours = raster.contours(band, interval, simplify, smooth)
```



Stylize

Stylize a Raster by baking in a style to create a new Raster

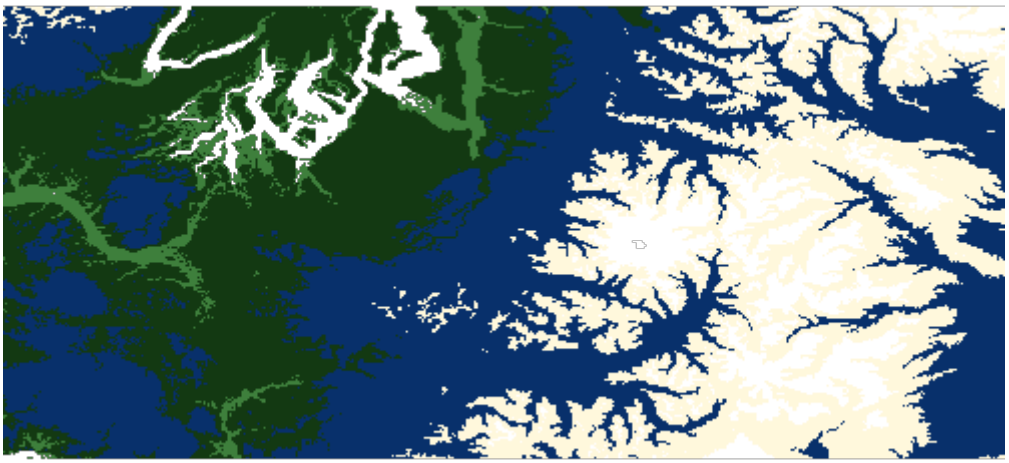
```
File file = new File("src/main/resources/pc.tif")
Format format = Format.getFormat(file)
Raster raster = format.read("pc")
Raster stylizedRaster = raster.stylize(new ColorMap([
        [color: "#9fd182", quantity:25],
        [color: "#3e7f3c", quantity:470],
        [color: "#133912", quantity:920],
        [color: "#08306b", quantity:1370],
        [color: "#fffff5", quantity:1820],
]))
```



Reclassify

Reclassify a Raster

```
File file = new File("src/main/resources/pc.tif")
Format format = Format.getFormat(file)
Raster raster = format.read("pc")
Raster reclassifiedRaster = raster.reclassify([
        [min:0,    max:0,    value: 1],
        [min:0,    max:50,   value: 2],
        [min:50,   max:200,  value: 3],
        [min:200,  max:1000, value: 4],
        [min:1000, max:1500, value: 5],
        [min:1500, max:4000, value: 6]
])
```

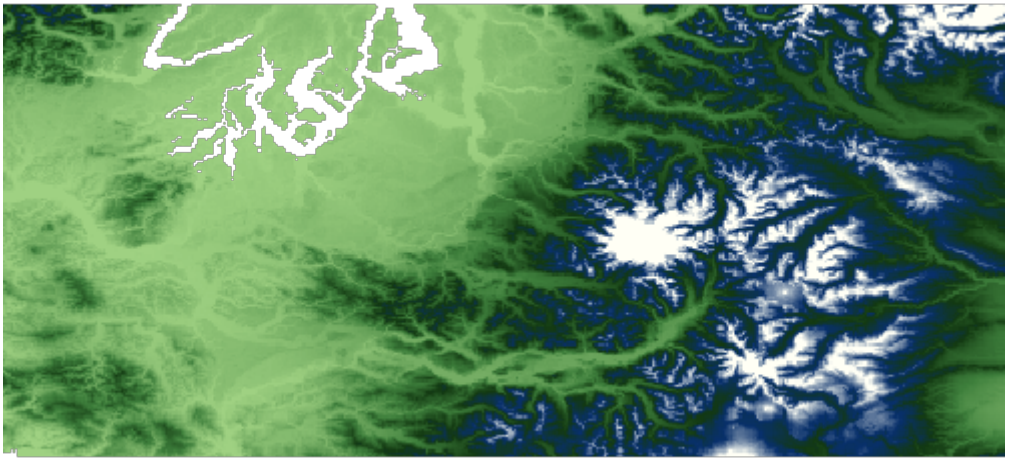


Scale

Scale a Raster

```
File file = new File("src/main/resources/pc.tif")
Format format = Format.getFormat(file)
Raster raster = format.read("pc")
println "Original Raster Size = ${raster.size[0]}x${raster.size[1]}"

Raster scaledRaster = raster.scale(0.5, 0.5)
println "Scaled Raster Size = ${scaledRaster.size[0]}x${scaledRaster.size[1]}"
```

```
Original Raster Size = 800x400
Scaled Raster Size = 400x200
```



Invert

Invert the values of a Raster

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

Exponent

Calculate the exponent of the values of a Raster

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

Log

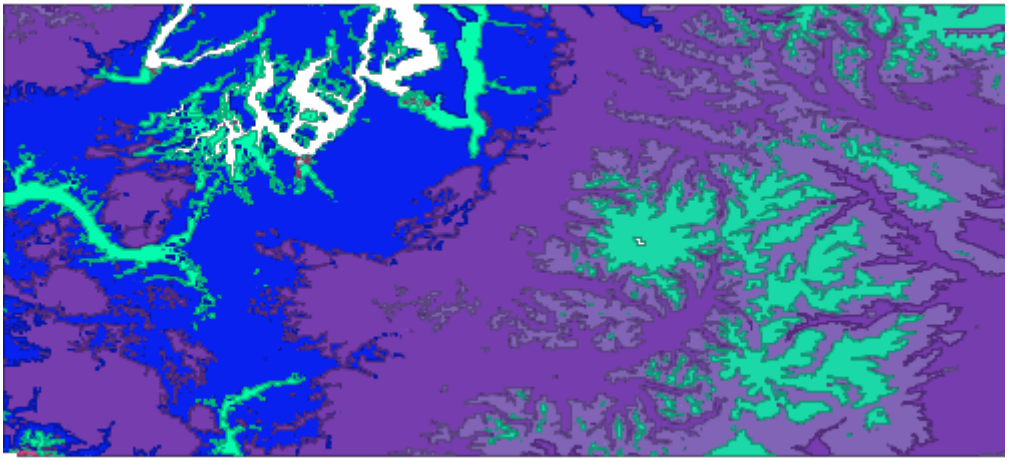
Calculate the log of the values of a Raster

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

Vectorize

Create a Polygon Layer from a Raster

```
File file = new File("src/main/resources/pc.tif")
Format format = Format.getFormat(file)
Raster raster = format.read("pc")
Raster reclassifiedRaster = raster.reclassify([
        [min:0,    max:0,    value: 1],
        [min:0,    max:50,   value: 2],
        [min:50,   max:200,  value: 3],
        [min:200,  max:1000, value: 4],
        [min:1000, max:1500, value: 5],
        [min:1500, max:4000, value: 6]
])
Layer layer = reclassifiedRaster.polygonLayer
```



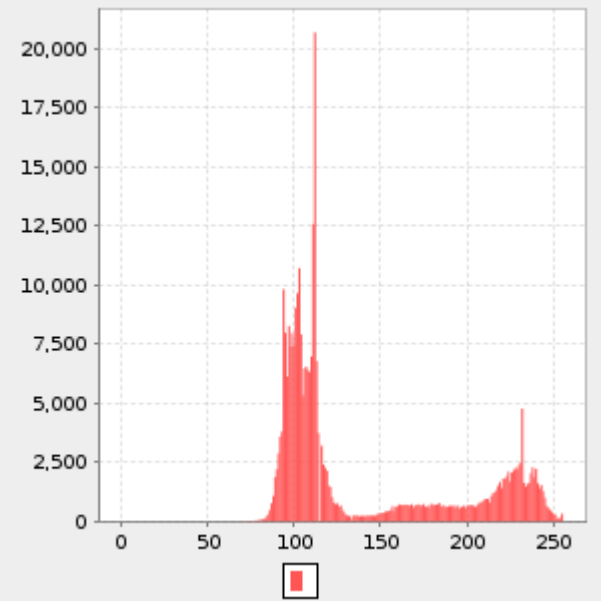
Create a Point Layer from a Raster

```
File file = new File("src/main/resources/pc.tif")
Format format = Format.getFormat(file)
Raster raster = format.read("pc").crop(new Bounds(-121.878548,46.808402,-121.636505
,46.896097, "EPSG:4326"))
Layer layer = raster.pointLayer
```

Get histogram of the Raster

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

Histogram histogram = raster.getHistogram()
println "# of bands = ${histogram.numberOfBands}"
println "# Counts = ${histogram.counts().size()}"
println "# Bins = ${histogram.bins().size()}"
println "Count 25 = ${histogram.count(25)}"
println "Bin 45 = ${histogram.bin(45)}"

Chart chart = Bar.xy(histogram.counts(0).withIndex().collect {int count, int index ->
[index, count]})
```

```
# of bands = 3
# Counts = 256
# Bins = 256
Count 25 = 0
Bin 45 = [45.0, 46.0]
```



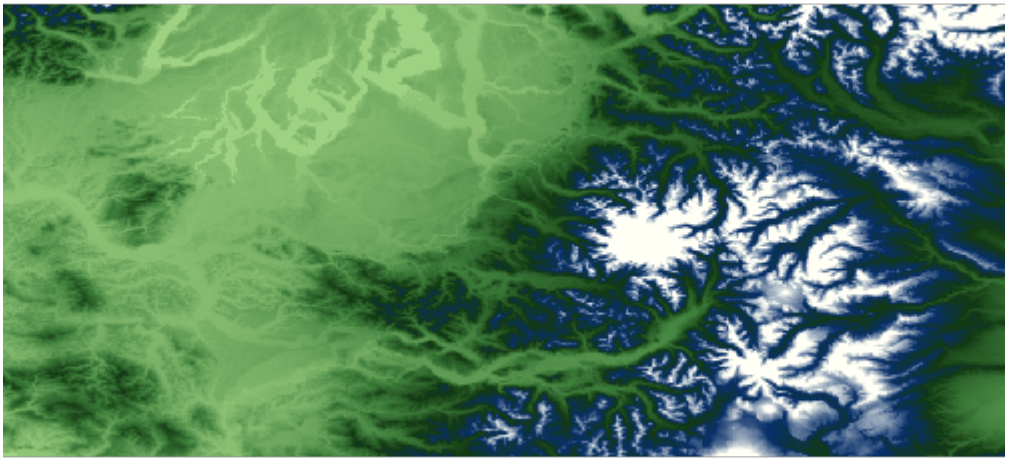
Raster Algebra

Add

Add a constant value to a Raster

```
File file = new File("src/main/resources/pc.tif")
Format format = Format.getFormat(file)
Raster raster = format.read("pc")
double elevation1 = raster.getValue(new Point(-121.799927,46.867703))
println elevation1

Raster higherRaster = raster.add(100.00)
double elevation2 = higherRaster.getValue(new Point(-121.799927,46.867703))
println elevation2
```

```
3069.0
3169.0
```

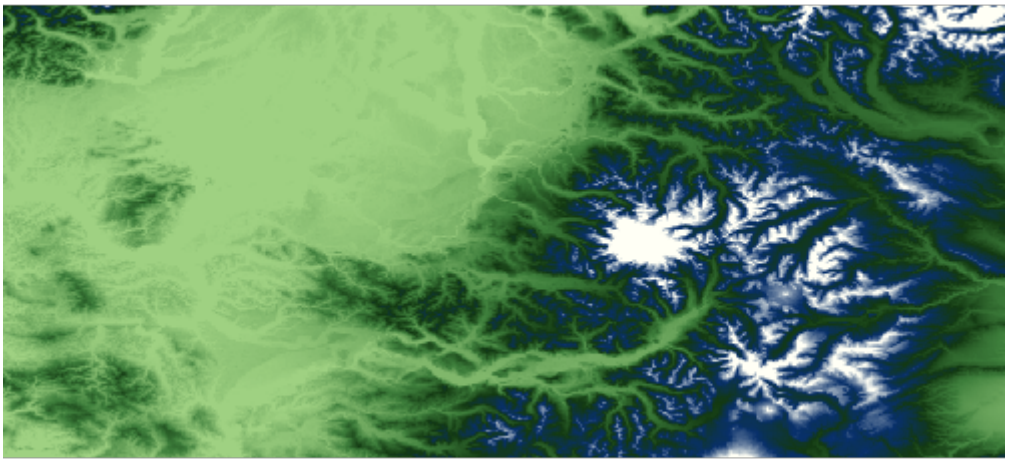


Subtract

Subtract a constant value from a Raster

```
File file = new File("src/main/resources/pc.tif")
Format format = Format.getFormat(file)
Raster raster = format.read("pc")
double elevation1 = raster.getValue(new Point(-121.799927,46.867703))
println elevation1

Raster lowerRaster = raster.minus(50.00)
double elevation2 = lowerRaster.getValue(new Point(-121.799927,46.867703))
println elevation2
```

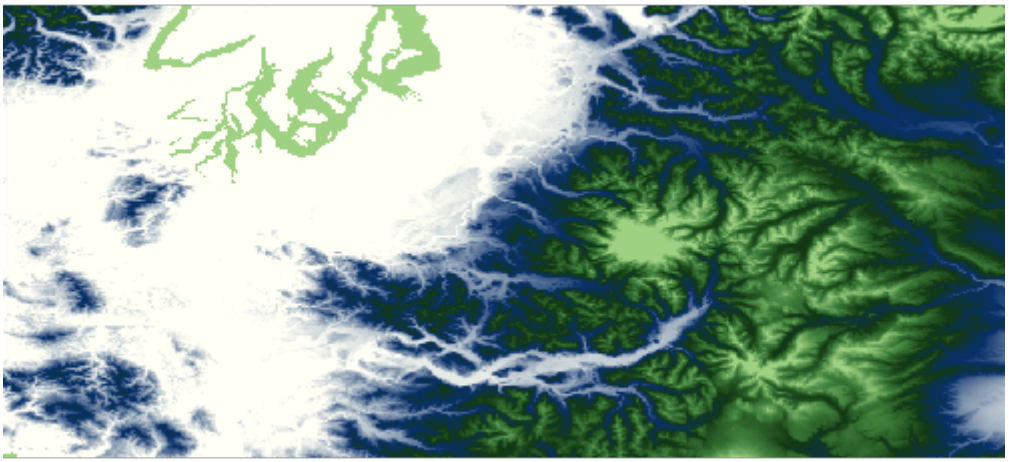
```
3069.0
3019.0
```



Subtract the Raster from a constant value

```
File file = new File("src/main/resources/pc.tif")
Format format = Format.getFormat(file)
Raster raster = format.read("pc")
double elevation1 = raster.getValue(new Point(-121.799927,46.867703))
println elevation1

Raster lowerRaster = raster.minusFrom(2000.0)
double elevation2 = lowerRaster.getValue(new Point(-121.799927,46.867703))
println elevation2
```

```
3069.0
-1069.0
```

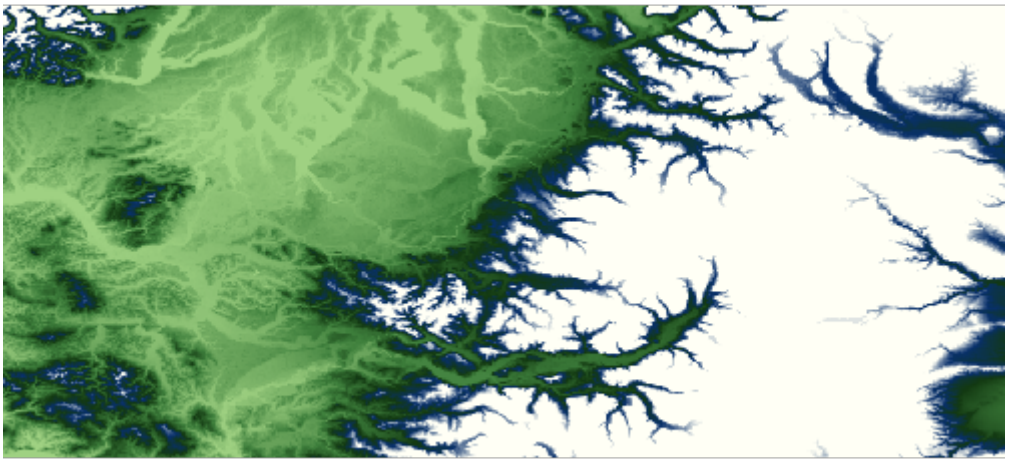


Multiply

Multiply a constant value against a Raster

```
File file = new File("src/main/resources/pc.tif")
Format format = Format.getFormat(file)
Raster raster = format.read("pc")
double elevation1 = raster.getValue(new Point(-121.799927,46.867703))
println elevation1

Raster higherRaster = raster.multiply(2.0)
double elevation2 = higherRaster.getValue(new Point(-121.799927,46.867703))
println elevation2
```

```
3069.0
6138.0
```



Divide

Divide a constant value against a Raster

```
File file = new File("src/main/resources/pc.tif")
Format format = Format.getFormat(file)
Raster raster = format.read("pc")
double elevation1 = raster.getValue(new Point(-121.799927,46.867703))
println elevation1

Raster lowerRaster = raster.divide(2.0)
double elevation2 = lowerRaster.getValue(new Point(-121.799927,46.867703))
println elevation2
```

```
3069.0
1534.5
```