**Multiple Raster Clip**

**Title** Multiple Raster Clip

**Summary**

This tool clips multiple rasters to a feature class, creates aspect, contour, hillshades, and slope rasters, and calculates min, max, mean, and standard deviation for the elevation value.

**Usage**

This tool allows you to extract a portion of multiple raster datasets based on a feature class extent.

You also have the ability to create aspect, contour, hillshades, and slope rasters based on the clipped rasters elevation information.

The minimum, maximum, mean, and standard deviation will be calculated and exported to the output table.

**Syntax**

rasterclip (Input\_Workspace, Clip\_Feature, Output\_Workspace, Output\_Table, {Aspect\_Contour\_Hillshade\_Slope})

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Explanation** | **Data Type** |
| Input\_Workspace | **Dialog Reference**  The folder in which the input raster datasets to be clipped.  There is no python reference for this parameter. | Workspace |
| Clip\_Feature | **Dialog Reference**  The feature used to clip the input raster datasets.  There is no python reference for this parameter. | Feature Class |
| Output\_Workspace | **Dialog Reference**  The folder in which clipped raster datasets will be created.  There is no python reference for this parameter. | Workspace |
| Output\_Table | **Dialog Reference**  The table in which raster properties of the clipped raster will be exported to. The properties to be obtained from the input raster includes: minimum, maximum, mean, and standard deviation of all cells in the input raster.  There is no python reference for this parameter. | Table |
| Aspect\_Contour\_Hillshade\_Slope (Optional) | **Dialog Reference**  Derive aspect, contour lines, hill shades, and slopes from raster surface to quantify digital elevation model's terrain landform.  There is no python reference for this parameter. | Boolean |

**Code Samples**

**Standalone Python Script**

There is no description for this code sample.

\_\_author\_\_ = 'Li Xi'

import arcpy

import os

arcpy.CheckOutExtension("spatial")

arcpy.CheckOutExtension("3D")

#Set parameters

workspace = arcpy.GetParameterAsText(0)

clipfc = arcpy.GetParameterAsText(1)

outspace = arcpy.GetParameterAsText(2)

outtable = arcpy.GetParameterAsText(3)

surface\_tiff = arcpy.GetParameterAsText(4)

arcpy.env.workspace = workspace

arcpy.env.overwriteOutput = True

tifflist = arcpy.ListRasters()

cnt = len(tifflist)

arcpy.env.mask = clipfc

#Clip multiple rasters

arcpy.SetProgressor("step","Clipping Rasters to Geodatabase...", 0, cnt, 1)

i = 1

for tiff in tifflist:

arcpy.SetProgressorLabel("Clipping " + str(i) + "/" + str(cnt) + ": " + tiff + "...")

desc = arcpy.Describe(tiff)

out\_tiff = os.path.join(outspace, desc.name)

arcpy.env.snapraster = tiff

out\_ras = arcpy.sa.ExtractByMask(tiff,clipfc)

out\_ras = arcpy.sa.ApplyEnvironment(out\_ras)

out\_ras.save(out\_tiff)

#Calculate raster properties and get output

elev\_min = arcpy.GetRasterProperties\_management(out\_tiff,"MINIMUM")

elev\_max = arcpy.GetRasterProperties\_management(out\_tiff,"MAXIMUM")

elev\_mean = arcpy.GetRasterProperties\_management(out\_tiff,"MEAN")

elev\_sd = arcpy.GetRasterProperties\_management(out\_tiff,"STD")

elevmin = elev\_min.getOutput(0)

elevmax = elev\_max.getOutput(0)

elevmean = elev\_mean.getOutput(0)

elevsd = elev\_sd.getOutput(0)

#use cursor to add sequential numbers to OID field of output table

x = 0

field1 = ["OID"]

Cursor = arcpy.da.UpdateCursor(outttable,field1)

for row in Cursor:

row.field1 = x

x += 1

Cursor.updateRow(row)

del Cursor

#insert values into statistic table

fields = ["Minimum","Maximum","Mean","SD"]

cursor = arcpy.da.InsertCursor(outtable,fields)

cursor.insertRow((elevmin,elevmax,elevmean,elevsd))

del cursor

#quantify landform terrain with aspect, contour, hillshades, and slope.

ras\_aspect = "aspect" + ".tif"

ras\_contour = "contour" + ".tif"

ras\_shades = "shades" + ".tif"

ras\_slope = "slope" + ".tif"

if surface\_tiff.lower() == "true":

arcpy.SetProgressorLabel("Creating aspect & contour & hillshades & slope...")

arcpy.env.workspace = outspace

raslist = arcpy.ListRasters()

for ras in raslist:

arcpy.Aspect\_3d(ras,ras\_aspect)

arcpy.Contour\_3d(ras,ras\_contour,200)

arcpy.HillShade\_3d(ras,ras\_shades)

arcpy.Slope\_3d(ras,ras\_slope)

arcpy.SetProgressorPosition()

arcpy.CheckInExtension("spatial")

arcpy.CheckInExtension("3D")

**Tags**

Raster clip, raster surface properties, aspect, contour, hillshades, slope.

**Credits**

There are no credits for this item.

**Use limitations**

There are no access and use limitations for this item.

*You are currently using the Item Description metadata style. Change your metadata style in the Options dialog box to see additional metadata content.*

