

Image Analysis for ArcGIS FAQs¹



Fall 2004

Remote Sensing Applications Center
<http://fsweb.rsac.fs.fed.us>

How Do I Perform a Data Rescale?

Use Spatial Analyst to perform data rescaling in ArcMap. Rescaling raster data (e.g., 8-bit, 16-bit, etc.) is often necessary to make datasets compatible with one another for further processing, or to accurately reflect the type of data being analyzed (e.g., DEMs in the western US typically require a 16-bit data range). This document provides an example of how to rescale a 32-bit floating point image to an 8-bit unsigned integer format using the Raster Calculator in Spatial Analyst.

What You Will Need and Other Assumptions

- An unsigned integer 32-bit raster image
- This example will only work for a 32-bit floating point to 8-bit unsigned integer conversion—other conversions using the Raster Calculator will vary.

Overview of Steps

1. Review image statistics (e.g., range, mean, etc.) and data properties for your image.
2. Develop the conversion expression from 32-bit to 8-bit using the Raster Calculator.
3. Inspect the results.

Step-by-Step Example

1. Start ArcMap from your Desktop, or on the Windows Taskbar click **Start | Programs | ArcGIS | ArcMap**. Ensure the Spatial Analyst extension is visible. If not, from ArcMap's main menu select: 1) **Tools | Extensions** and enable **Spatial Analyst**; and 2) **View | Toolbars** and enable **Spatial Analyst**.
2. Click the **Add Data** button, navigate to your 32-bit image, and add it to the Table of Contents in ArcMap.
3. **Right-click** on the image filename in the Table of Contents, select **Properties**, and then click the **Source** tab. Review the image properties (particularly the image stats) and ensure that the data type is indeed a 32-bit floating point. Write down the **Minimum** and **Maximum** values from the Statistics section (for this example, assume the Minimum = -0.43 and the Maximum = 0.75). Select **OK** to dismiss the dialog.
4. From the Spatial Analyst toolbar, select **Spatial Analyst | Raster Calculator**.
5. From the Raster Calculator, your 32-bit image should be listed in the **Layers** section. Ensure you have access to the advanced functions by selecting the **Arrow** icon on the bottom of the dialog.
6. First, type what you want your new output image to be named, followed by an equals sign (**8bitimage =**) in the formula area, and select the **Int** button (for integer). Your expression should now look like this: **8bitimage = Int()**. *Note: ensure that you leave spaces between operators.*

¹ Produced by the USDA Forest Service RSAC (<http://fsweb.rsac.fs.fed.us>). A Forest Service version of Image Analysis for ArcGIS was used to develop this reference document. No warranty is made as to completeness or accuracy.

7. Place your cursor inside the parentheses. **Double-click** your 32-bit image from the layers section, followed by clicking on the **add sign** button. Fill in the rest of the expression to make it look like this: $8bitimage = \text{Int}([\text{32bitimage.img}] + 0.43^a) * 255 / (0.75 + 0.43^b)$. *Note: the ArcInfo Help (found by selecting **Start | Programs | ArcGIS | ArcInfo Workstation | ArcDoc**) is the best source for help with developing expressions using the Raster Calculator.*

^a Input the additive inverse of your Minimum value in the expression here (e.g., the additive inverse of -0.43 equals 0.43).

^b Input the Maximum value minus the Minimum value here (e.g., $0.75 - (-0.43) = 0.75 + 0.43$).

8. Select the **Evaluate** button. *Note: when you evaluate an expression using the raster calculator, the result is produced in a GRID format even if the input image is an Imagine (.img) file. If you wish to convert from a GRID back to an .img file, use **ArcCatalog**, navigate to and **Right-click** on your final GRID, select **Export | Raster to a Different Format**, and change the **Save As Type** to **ERDAS Imagine**.*

9. Inspect the results.

Note: it is very important to identify the data type (floating point or unsigned integer) for the conversion process. For instance, in our example, if we were converting an unsigned integer 32-bit image to an unsigned integer 8-bit image the formula used would be different.