



# EXERCISE 2

## Segmentation for Stand Delineation

### Introduction

Traditional methods of hand digitizing polygons (e.g., for creating stand layers) from imagery can be tedious and yield inconsistent and unrepeatable results. Users can use the segmentation algorithm in eCognition to automate this process, enabling users to quickly and objectively derive segments from digital imagery. The goal of this exercise is to present an overview of this process and provide you with an in-depth understanding of how to control your segmentation output.

### Objectives

- Learn how to use eCognition to generate stand-level segments from imagery
- Learn the intricacies of the segmentation algorithm and its parameters

### Required Data

- NAIP image file: **mttaylor\_naip\_10m.img** (3 bands, NIR, R, G, 10 m resolution)
- Derived vegetation index file: **mttaylor\_ndvi\_10m.img** (Normalized Difference Vegetation Index)
- Tri-shade image file: **mttaylor\_trishade\_10m.img** (fully-illuminated hillshade)
- Developer project file: **mttaylor\_stand\_delin.dpr** (created in Exercise 1 and should reference all of the above data layers)

### Prerequisites

- You have properly installed and licensed eCognition
- You have completed Exercise 1
- eCognition and the **mttaylor\_stand\_delin.dpr** file are open
- You know how to take a screen shot
- You have Microsoft PowerPoint





## Table of Contents

Part 1: Create Segmentation Parent and Child Process.....	3
Part 2: Explore the Scale Parameter for Segmentation .....	6
Part 3: Explore Image Layer Weight Parameter for Segmentation .....	7
Part 4: Explore Shape & Compactness Parameters for Segmentation .....	10
Part 5: Export Results.....	15

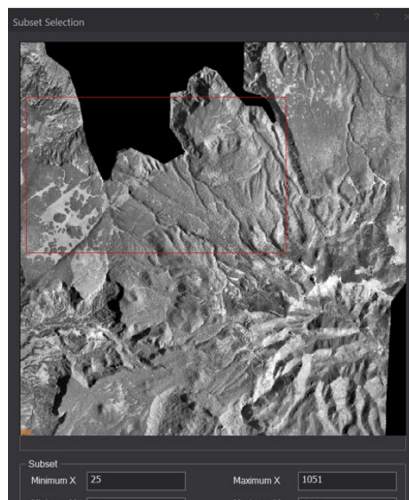


## Part 1: Create Segmentation Parent and Child Process

You will jump right in and create the segmentation process within the project you created in Exercise 1.

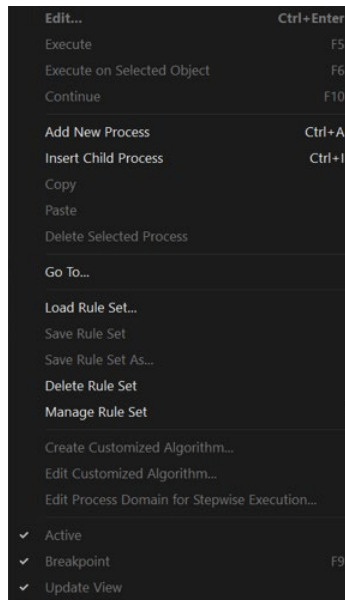
### A. Open eCognition & project

1. If necessary, launch eCognition and Open the project that you created in Exercise 1, **mttaylor\_stand\_delin.dpr**.
2. If you want, double check the project information, the No Data values, the Subset and the Alias names.
  - i. Make sure the No Data Values have been set to 0 (zero) for the NIR, Red, and Green layers; the Subset matches that of the following graphic; and that the Alias names have been set to "NIR", "Red", "Green", "NDVI", etc. If you do not specify what the No Data Values are, the black regions will be included in all subsequent processing and will yield unneeded data and result in longer processing times.



### B. Add Segmentation Algorithm

1. To begin, **right click** inside the **Process Tree** window and take a moment to read the various options from the drop down menu (see following graphic). The primary options we will use are **Add New Process** and **Insert Child**.
  - i. Additionally, make note of the grayed out options **Edit...** and **Execute**. After you create a process, you will be able to access both of these options.



2. From the **Process Tree** menu, select the **Add New Process** option. The **Edit Process** dialog opens. This dialog window is where you will spend a lot of time throughout the development of any eCognition process, creating ‘parent’ and ‘child processes’, adjusting parameters, setting thresholds and much more.

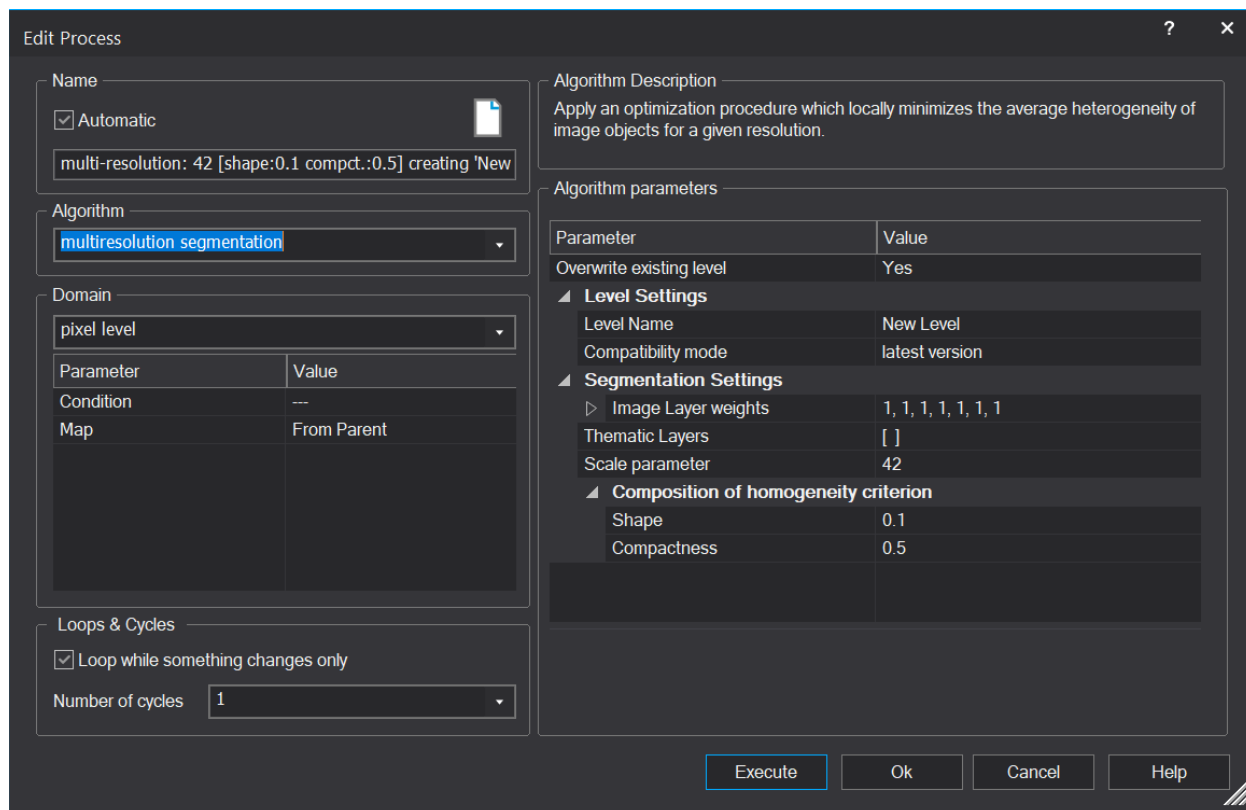
**Note:** For Parent processes that you insert to organize your rulesets into blocks, we recommend you give them an intuitive name.

For Child processes that execute an algorithm you always need to leave the **Automatic** (typically “do”) name option. This is necessary to avoid creating bugs or infinite loops in the rule set, which might be hard to find. You may receive a warning message the first time you set a Parent process (that you are using as an organizational placeholder) a name to other than “do”; read the message and click **Yes**.

3. This first process is going to be a parent process that we use to organize our rules, so in the **Name** section where it says “do” change it to say “**Create Segments**” and leave the Algorithm set to **execute child processes**.
4. Leave the rest of the settings at their defaults and click **Ok** to exit the **Edit Process** dialog.
5. A warning message may pop up, check the box “**Don’t ask me again**” and then click **Yes**.
6. Now that you have a Parent process created, let’s create a **Child Process**. **Right click** on the **Create Segments** rule (in the Process Tree window) and select **Insert Child Process**.
7. The **Edit Process** dialog opens (see following graphic)—this is a new dialog for the new **Child Process**. Leave the **Name** option set to **Automatic**.
8. From the **Algorithm** drop down menu, find the **multiresolution segmentation** option and select it.

**Note:** If you begin typing the name of the algorithm you would like, the software automatically fills in the options that match. You can use your down arrows to select these - a shortcut compared to scanning and selecting from the full list of options.

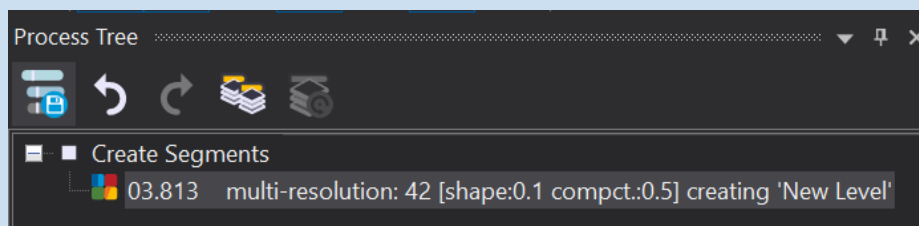
9. Click **Execute** to run your new Child Process with the defaults.



10. After the segmentation has completed, take a moment and review the resulting segments, paying special attention to their shape (the detailed nature of their boundaries), their compactness (the linearity of them) and how it follows the features within the image; we will learn how to control these by adjusting segmentation settings.

**Note:** In the Process Tree panel, you may have noticed that the name of the child process has automatically filled with information that reflects the current algorithm parameters (default parameters for now).

It also displays the time it took to complete the process, 4.243 seconds in the example below. This information is invaluable when troubleshooting rule sets that take too long to run on your full image set. You can use these times to determine which process is taking the longest to run (or not completing).



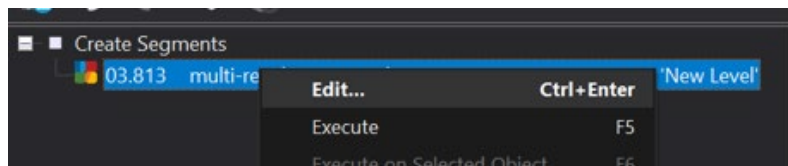
In Parts 2 and 3, you will change the multiresolution parameters to adjust the segments until you have a set that delineates polygons representing vegetated stands.

## Part 2: Explore the Scale Parameter for Segmentation

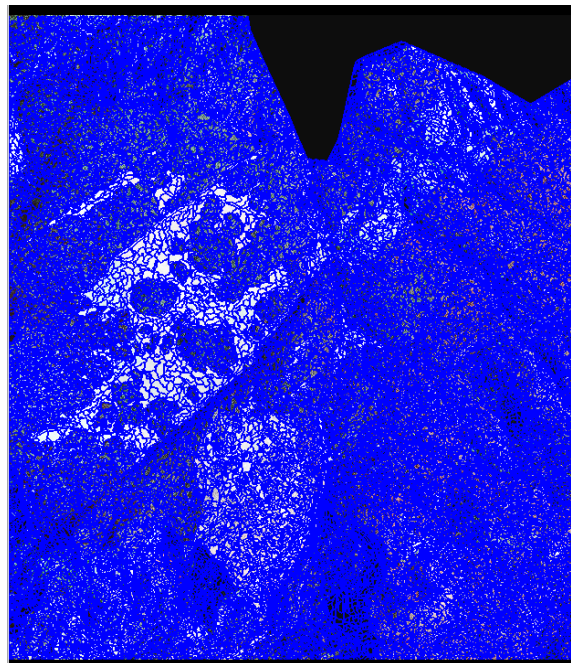
In Part 1, you used the default parameters for generating segments. Now, you will change the Scale parameter which controls the size of segments to create segments which represent stand boundaries.

### A. Change Scale Parameter from 10 to 5

1. Edit your multiresolution segmentation process by right clicking in the Process Tree window and select **Edit...** to open the Edit Process dialog (Hint: Right click on **10 [shape:0.1 compct.:0.5] creating 'New Level'**, similar to following graphic).



2. Locate the **Scale** parameter and change it to 5 and click **Execute** to run the segmentation. Your results should look similar to the following graphic (the zoom factor in the image below is set to 100%).



3. Visually evaluate the new segments, noting their shape, size and relationship with apparent stand boundaries (Hint: Pan and zoom to explore the imagery and segments).

**Question:** Do you think this value for Scale is appropriate for stand delineation?

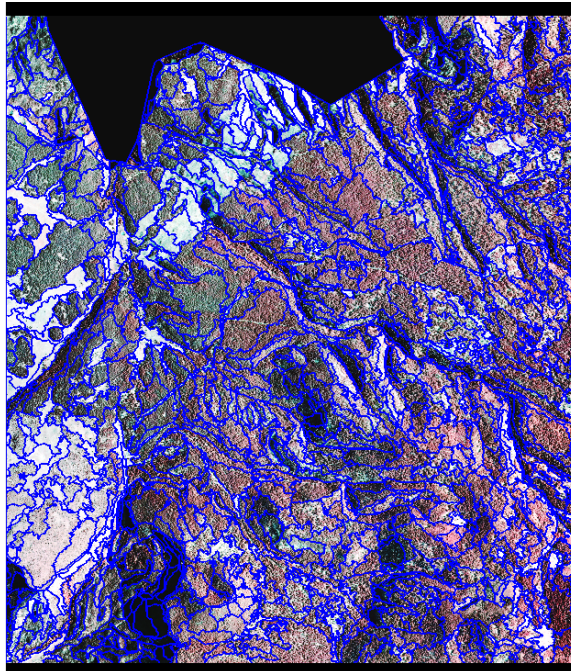
### B. Change Scale Parameter to 25

1. Next, let's change the Scale parameter to something larger and more appropriate for vegetation stand delineation. **Right click** on the **child process** and select **Edit...**

2. Change the **Scale** parameter to **25** this time and click the **Execute** button to generate the new segments.

**Note:** The **Edit Process** dialog has a setting “**Overwrites existing level**”. By default, this is set to Yes, which means that each time you Execute this segmentation process it will overwrite the previous output, if it exists. For our purposes and most situations in the future, this is a great feature facilitating an iterative approach to developing your eCognition processes.

3. **Evaluate** the new segments. Your output should look similar to the following graphic.



Next, you will explore some other parameters for segmentation, but keep the Scale parameter set to 25 units.

## Part 3: Explore Image Layer Weight Parameter for Segmentation

One component influencing segmentation is the layer weighting. You will explore this next.

In Parts 4 and 5 you will continue to vary segmentation parameters and explore the resulting segmentation output. The purpose of this is to help you gain a complete understanding of how to control the output of your segmentation results.

### A. Adjust Image Layer Weights

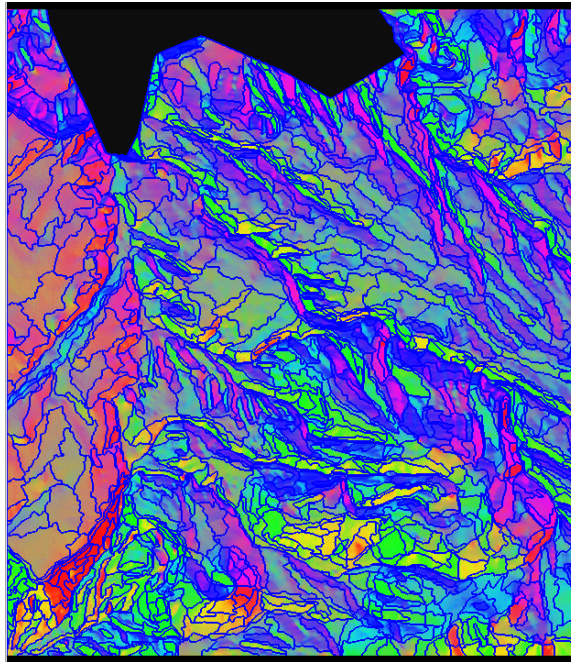
1. Right click the multiresolution segmentation child process and choose **Edit** to open the **Edit Process** dialog.

2. Under the section **Algorithm parameters**, locate and expand the section titled **Image Layer weights** (see following graphic).

Parameter	Value
Compatibility mode	latest version
▲ Segmentation Settings	
▲ Image Layer weights	1, 1, 1, 1, 1, 1, 1
Green	1
NDVI	1
NIR	1
Red	1
ts1	1
ts2	1
ts3	1
Thematic Layers	[ ]
Scale parameter	25
▲ Composition of homogeneity criterion	

**Note:** In the West, vegetation patterns are heavily influenced by topography. Therefore, it can be useful to use topography in the segmentation process (in this case, the tri-shade layers).

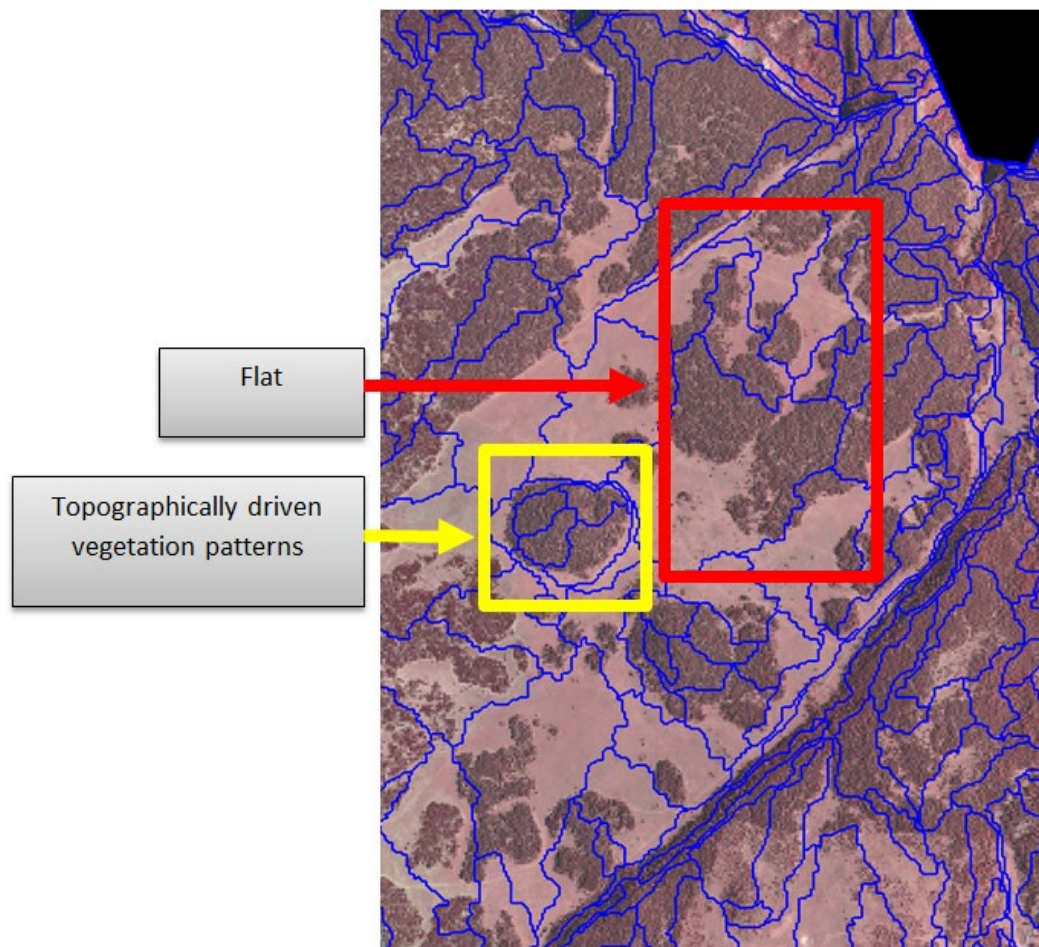
3. Change the weights to use only the tri-shade topography layers. Set them to 1, and change all the others to 0 (zero). Then **Execute** the process.
  - i. green = 0
  - ii. NDVI = 0
  - iii. NIR = 0
  - iv. red = 0
  - v. ts1 = 1
  - vi. ts2 = 1
  - vii. ts3 = 1
4. Once the process finishes, use the knowledge you gained from Exercise 1 to **change the view** to display the tri-shade layers in a 3-band mix (Hint: Open **View Setting** to display ts1, ts2 and ts3 in RGB).
5. **Pan** and **zoom** to evaluate your segments for shape, size and for adequacy. Your view should look similar to the following graphic—note the segments tend to follow the topographic features.



**Question:** How could you use the **Image Layer Weighting** to your advantage when creating segments for vegetation stand delineation purposes?

**Answer:** The NDVI layer strongly characterizes variation in vegetation. Giving this layer more weight in the segmentation process will result in segments which better represent areas of homogenous vegetation (e.g., stands).

6. Next, **switch** your view from the tri-shade to a **Color-Infrared** view (NIR, red, green) to evaluate the segments against the imagery—see following graphic.



**Note:** In areas where vegetation pattern is topographically driven, the segments will follow the vegetation pattern more closely than in flat areas where the role of topography on the vegetation pattern is diminished.

## Part 4: Explore Shape & Compactness Parameters for Segmentation

We will now explore how the **Shape** and **Compactness** parameters influence the segmentation process. We will continue to develop a segmentation process to delineate stands.

### A. Set Shape and Compactness

Now we need to create the other 6 stratification classes using two separate conditional statements.

1. **Right click** the child process (i.e., **multi-resolution: 25[shape:0.1 compact.:0.5]**) and select **Edit** from the menu items (or double click on the child process).

2. In the **Edit Process** dialog you will now change the **Image Layer weights** to something more appropriate for vegetation delineation:

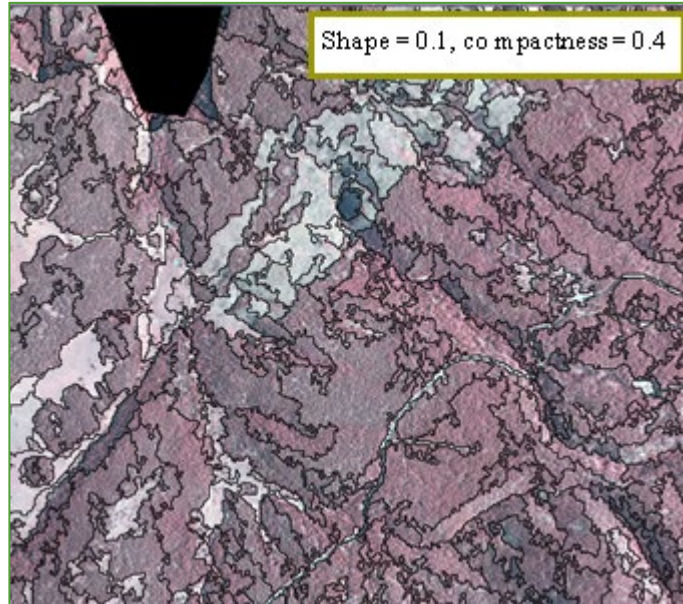
- i. Green = 1
- ii. NDVI = 2
- iii. NIR = 1
- iv. Red = 1
- v. ts1 = .3
- vi. ts2 = .3
- vii. ts3 = .3

**Image Layer Weights** are relative values. By assigning NDVI a weight of 2, we are giving NDVI twice as much influence as green, NIR and red on the segmentation process. Similarly, the value of 0.3 given to ts1, ts2 and ts3 indicates that these layers should be given approximately one third the influence of green, NIR, etc.

3. Next you will begin playing around with the **Shape** and **Compactness** parameters by iteratively testing different combinations of parameter values—this is where the more “subtle” changes between segment outputs begin to unfold. Using the table below as a guide, you will **create segment outputs** for each of the combinations between the Shape and Compactness parameters (Leave the Layer weights and the Scale parameter fixed). In the first round, set the **Shape to 0.1** and the **Compactness to 0.1** and click **Execute**.

		Shape		
		0.1	0.3	0.9
Compactness	0.1			
	0.4			
	1.0			

4. Make sure your view is set to the **Color-Infrared 3 band view** (RGB = NIR, Red, Green) and visually assess the segments for their shape, compactness and how they follow (or don't follow) the apparent stand boundaries in the image.
5. **Zoom** into the image so that you can more closely inspect the segments (see following graphic). You want to zoom in enough so that as you change the parameters of the segmentation process, you will be able to see enough of the detail and also enough of the “scope of the land” to really assess the segments.












## B. Set up PowerPoint to Capture Screen Shots for Comparing Segments

1. Take a screen shot of the zoomed in view by making sure eCognition window is the active software (single click inside it to make sure it is).
2. Launch **Microsoft PowerPoint** from the **Start** menu.
3. Change the Page Size to accommodate a whole series of these screen shots, by clicking on the **Design** tab, then on the **Page Setup** button.
  - i. Set the **Width** to **48** inches and the **Height** to **36** inches and click **OK**.
  - ii. It **might not be apparent** that the page has changed size because the dimensions are similar to the default **Page setup**.
  - iii. Delete any text or title boxes on the page.

**Note:** There is also a Power Point template stored in the Course\_Data folder that you can use to paste your images into (along with an answer key if you want to check your work).

4. Then **right click** and select **Paste**. If the Paste button is grayed out, you may need to go back to eCognition and take another screenshot and try pasting again.
5. **Place** this first screenshot in the **upper left hand corner** of the page and resize it if necessary. See following graphic for an idea of what your document should look like when you are done.

		Shape		
		0.1	0.3	0.9
Compactness	0.1			
	0.4			
	1.0			

6. **Add** text boxes and fill them in with the information you see in the preceding graphic—**Shape** and the values, and **Compactness** and the values.
7. Lastly, **save** your PowerPoint presentation to a **location of your choice** with an intuitive name—and **keep it Open**. This document may help you in the future, to understand or explain how the shape and compactness parameter influence the segmentation process.

### C. Adjust Compactness, Keeping Shape Constant

1. Now, go back to eCognition and **right-click** the child process and choose **Edit** (or just double-click on it).
2. Adjust the **Compactness** parameter to 0.4 while keeping the **Shape** parameter at 0.1.
3. Click **Execute** to create segments with the new parameter. The process will run, but you will not likely see drastic differences between this and the previous set of segments.

**Note:** *Shape* is inversely proportional to **Color**. So when **Shape** is given a value of 0.1, color (behind the scenes) is given a value (weight) of 0.9 and increases the influence of spectral information on the resultant segments. **Compactness** is inversely proportional to **smoothness**. One more thing to remember, **Compactness** is a modifier of **Shape**. This means that if **Shape** is set to a low value (e.g. 0.1) then the **Compactness** setting will have less influence (no matter how high the Compactness setting is) on the segmentation. You will observe this phenomenon in the process we have asked you to complete.

4. As you did in the previous step, **take a screenshot** of the output and **paste** it in PowerPoint.
5. **Resize it** and **place it below** the previous output. You are building the table above in graphic form with screenshots of the resulting segments from the **Shape** and **Compactness** parameters.
6. Repeat steps 1-5 for the remaining **Compactness** value (**1.0**).

### D. Change Shape to 0.3 and Adjust Compactness Values

1. In this step, you will change the **Shape** parameter to **0.3** and then repeat the changes to the **Compactness** parameter following the table. Open the Child Process for editing and change the **Shape** parameter from **0.1** to **0.3**, and change the **Compactness** back to **0.1**.
2. Click **Execute** to generate your new segments.
3. **Repeat** the steps for taking a screenshot and pasting it into PowerPoint for this second column of the table—the screenshots from this step will make up the **second column** of your “graphic” version of the table in PowerPoint.
4. Make sure to **change** the **Compactness** parameter for each of the values and generate new segments.

## E. Change Shape to 0.9 and Adjust Compactness Values

1. In this step, you will change the **Shape** parameter to **0.9** and then repeat the changes to the **Compactness** parameter following the table. **Open** the Child Process for editing and change the **Shape** parameter from **0.3** to **0.9**, and change the **Compactness** back to **0.1**.
2. Click **Execute** to generate your new segments.
3. **Repeat** the steps for taking a screenshot and pasting it into PowerPoint for this third column of the table—the screenshots from this step will make up the **third column** of your “graphic” version of the table in PowerPoint.
4. Make sure to **change** the **Compactness** parameter for each of the values and generate new segments until you have created segments for every pairing of **Shape** and **Compactness** in the table.

### *What differences do you see in the output with different shape and compactness values?*

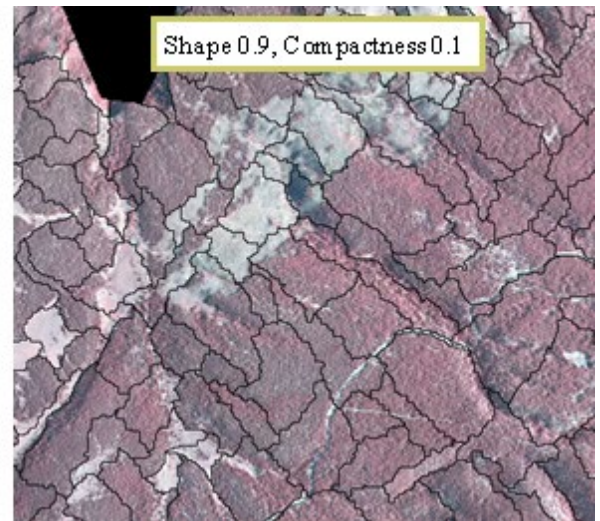
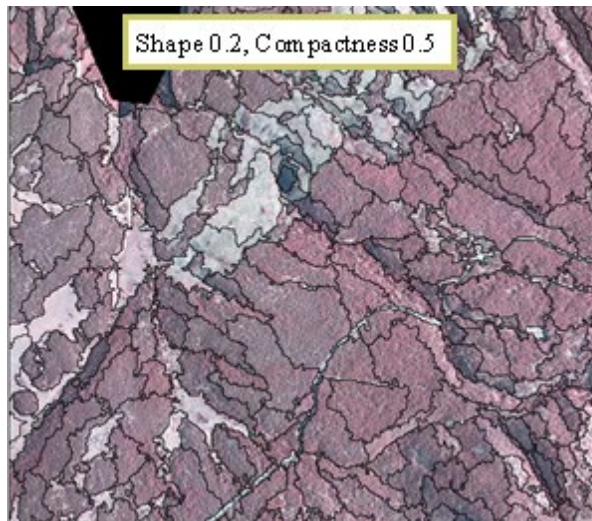
*You should notice significant differences in the segments between a **Compactness** value of 0.1 and 1.0 when the **Shape** value is 0.9.*

*You should also notice that when the value of **Shape** is 0.9, the segments do not follow the spectral patterns of the landscape. **Do you know why?** Recall that the complement of **Shape** is **Color**. So when **Shape** is given a weighting of 0.9 then color is only given a weight of 0.1 (not much weight for the spectral component of the segmentation process).*

## F. Final Shape and Compactness Settings

1. **Create** one last set of segments using the parameters listed below. You will use this last output as a **final result for exporting** in the next part of this exercise.
  - i. Based on our own exploration of this dataset, we decided on this set of Parameters for our final segments:
    - (a) Scale: 25
    - (b) Shape: 0.2
    - (c) Compactness: 0.5
2. You may notice that with a Shape of 0.2 that the boundaries of the segments are a bit “squirrely”. At first, this may concern you. You could change the Shape to 0.3 to mitigate for some of this squirrelyness, however, the segments don’t follow the subtle spectral patterns as well when Shape is 0.3. We have come up with a post-GIS processing workflow to help

smooth out the segments for a more hand-drawn appearance, **this is why we have suggested a Shape of 0.2** (see following graphic).

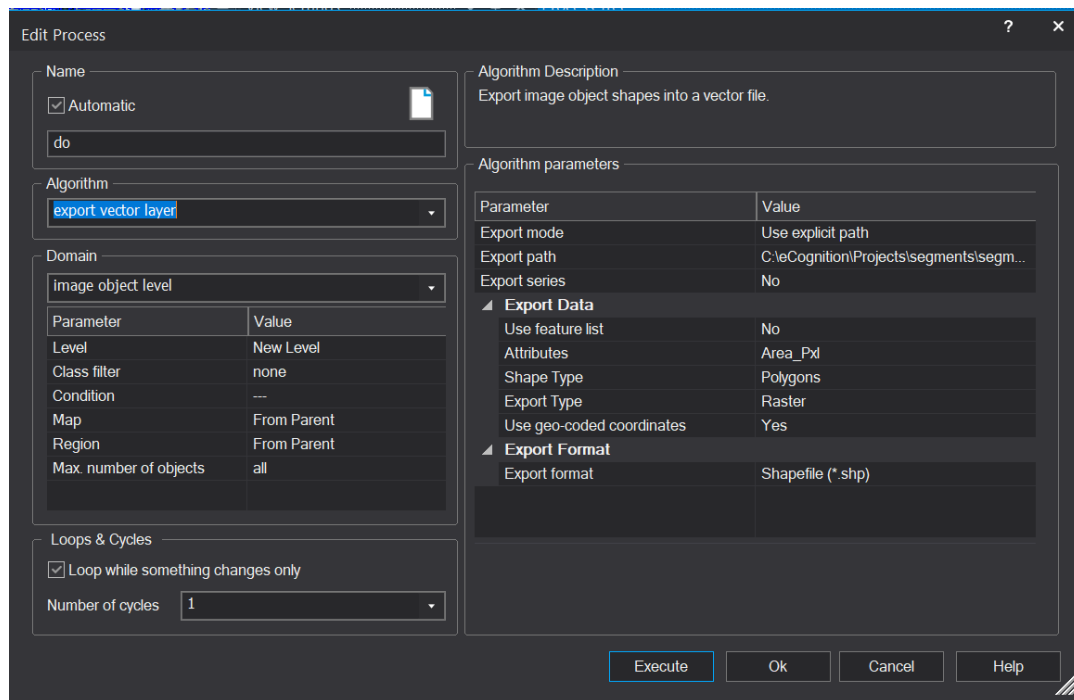


## Part 5: Export Results

Finally, we will learn how to export our results so that we can use the derived segments in other programs and analyses.

### A. Export Segments

1. In the **Process Tree**, right-click on the **Create segments** rule. Select **Add New Process** to create a new parent process.
2. In the **Name** field, type "**Export segments**" and press **OK**.
3. Right-click the newly-created parent process and select **Insert Child Process**
4. In the **Algorithm** drop-down list, scroll to the bottom of the bottom of the list and find all the **Export** options.
5. Locate and select the **Export Vector Layer** algorithm (see following graphic).



6. You can see there are many export options. We are exporting the segments as vectors so they could be used in other processes outside of eCognition.
  - i. Make sure the **Domain** is set to “**Image Object Level**” and the **Level** is set to “**New Level**”, or whatever level name you have chosen previously.
  - ii. Change the **Export Mode** to “**Use explicit path**” and define a specific path to export your segment shapefile to in **Export Path** (e.g., C:\eCognition\Projects\segments\segments.v1.shp). Add the file name as well (like you see in the path above).
  - iii. Add any **data** you would like to have included in the **Attribute Table** by clicking on the ellipsis (...) next to the **Attributes** parameter (e.g., Area of each segment in pixels: go to Object Features, then Geometry, Extent, and Area (Pxl)).
  - iv. Change the **Shape Type** to **Polygons**, and leave the default settings for **Export Type**, **Dimension**, and **Use geo-coded coordinates**.
  - v. Make sure the **Export Format** is set to **Shapefile (\*.shp)**.
  - vi. Lastly, press **Execute**.
7. Navigate to the path you have specified to ensure the file was successfully exported – feel free to add it to a GIS software, such as ArcMap to investigate the output.

**Congratulations!** You have successfully learned how to generate segments in eCognition and control the output size and shape using the algorithm parameters.