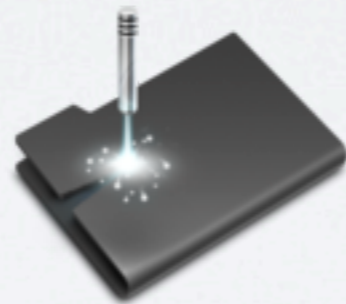


# VisiCut

## An Application Genre for Lasercutting in Personal Fabrication

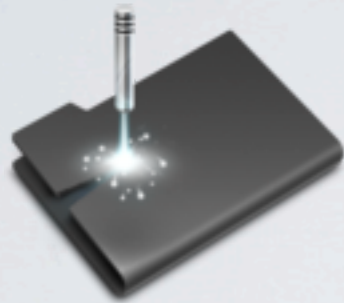
*(Bachelor's Thesis Talk)*



Thomas Oster, RWTH

Advisor: Dipl.-Inform. René Bohne

# OVERVIEW



VisiCut

LibLaserCut

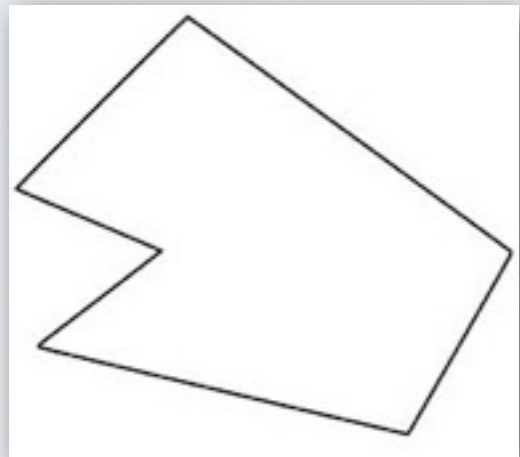
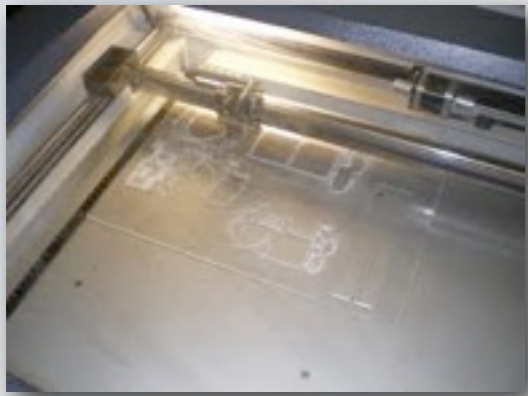
Epilog Driver

...other Drivers



# LASER MODES

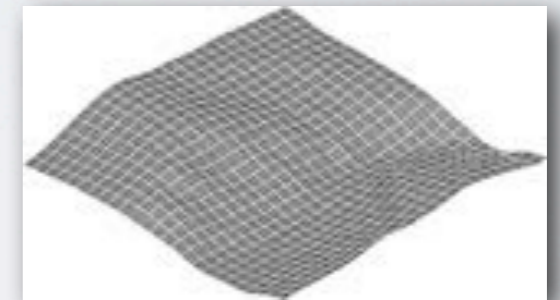
Vector



Raster



3D Raster



# IMPLEMENT A NEW DRIVER

## The LaserCutter Interface

```
package com.t_oster.liblaserCut;  
  
import java.util.List;  
  
public interface LaserCutter  
{  
  
    public List<Integer> getResolutions();  
  
    public double getBedWidth();  
  
    public double getBedHeight();  
  
    public List<String> getSettingAttributes();  
  
    public String getSettingValue(String attribute);  
  
    public void setSettingValue(String attribute, String value);  
  
    public void sendJob(LaserJob job) throws IllegalJobException;  
  
}
```

# IMPLEMENT A NEW DRIVER

The driver interprets the LaserJob

## **LaserJob**

Name  
Resolution  
Dimensions  
VectorPart  
RasterPart  
3D RasterPart

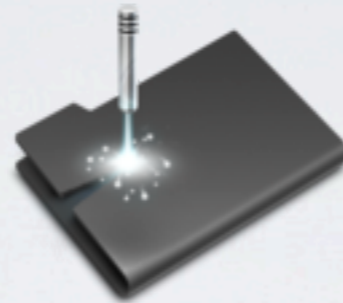
## **VectorPart**

LineTo  
MoveTo  
SetSpeed  
SetPower  
SetFrequency

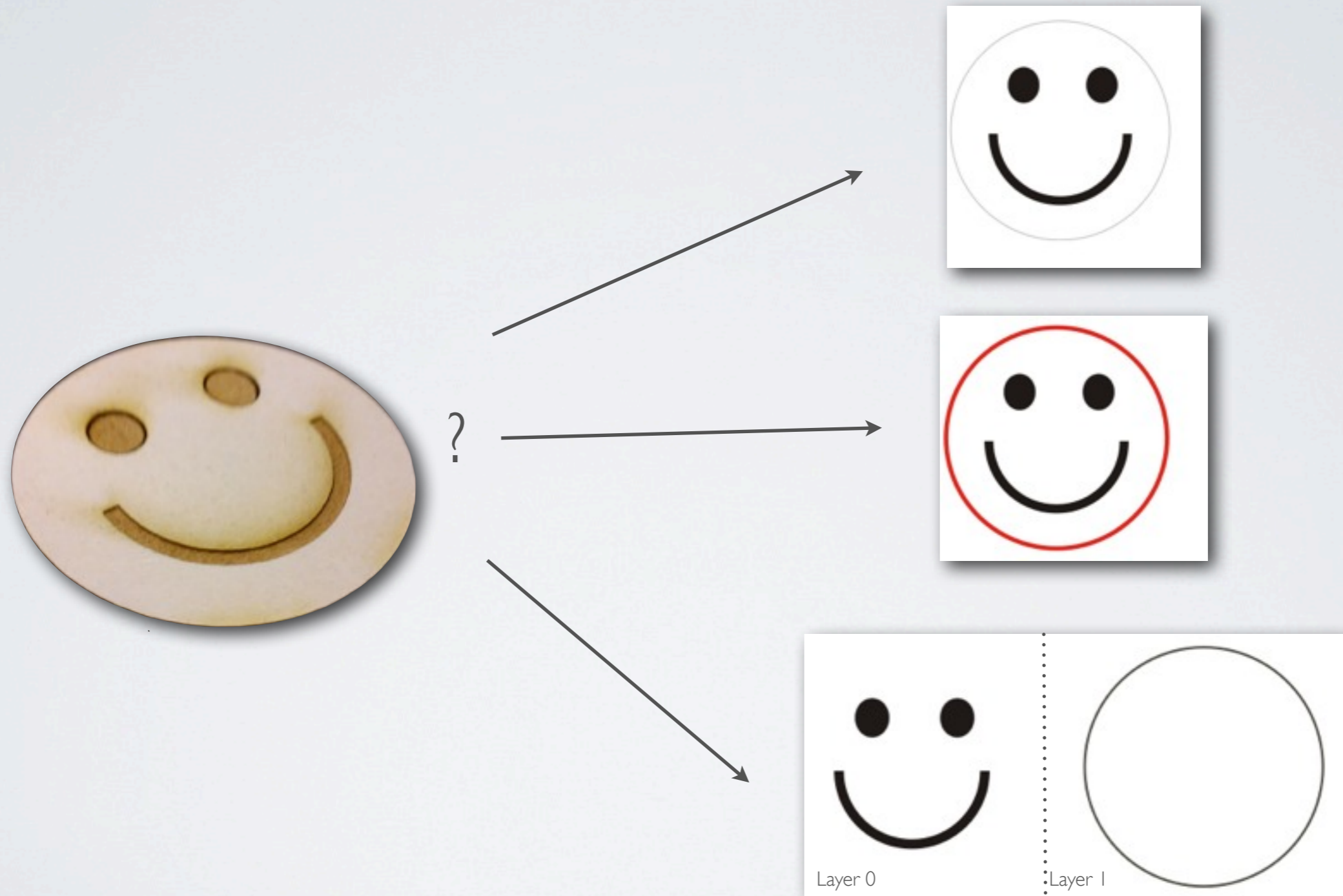
## **(3D) RasterPart**

Offset  
RasterData  
Power  
Speed

# VISICUT

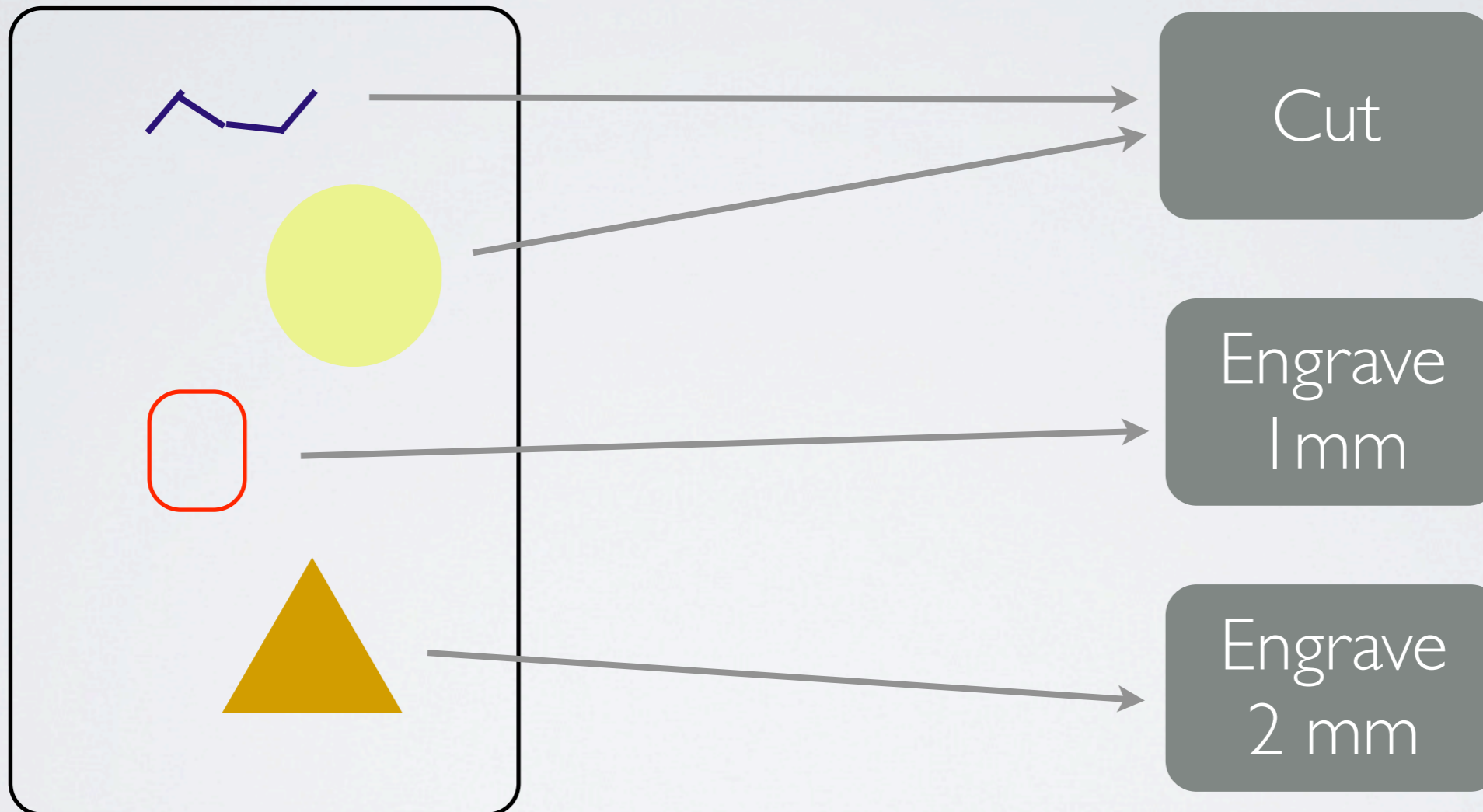


# MODELING

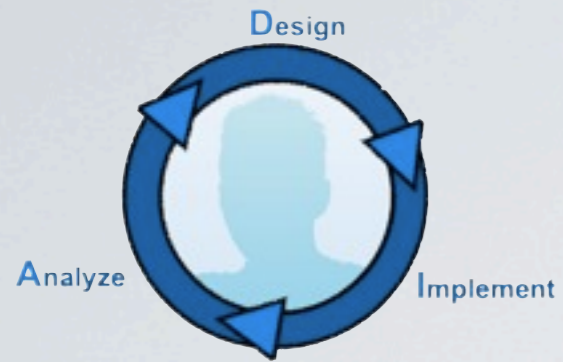


# MAPPING

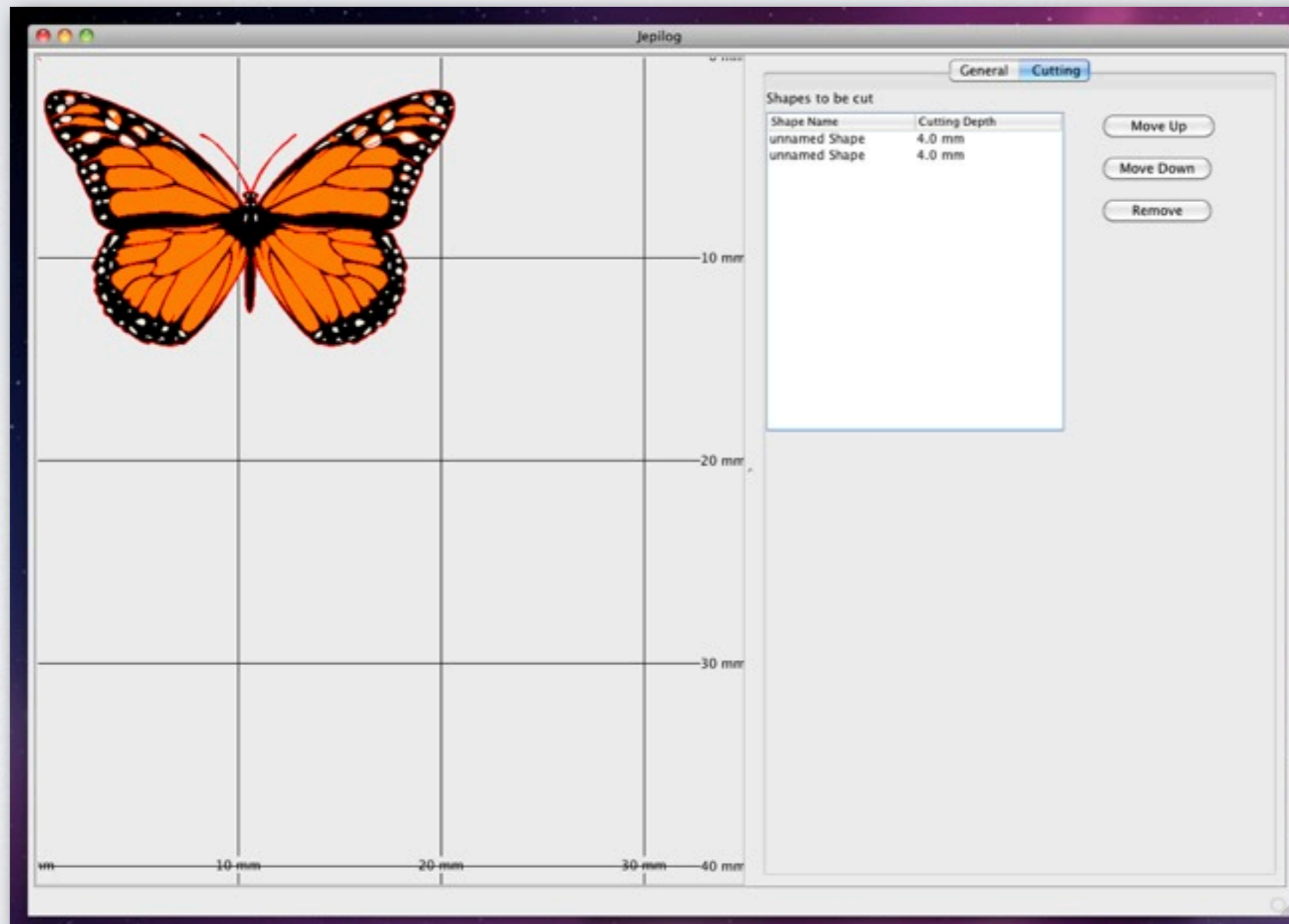
File

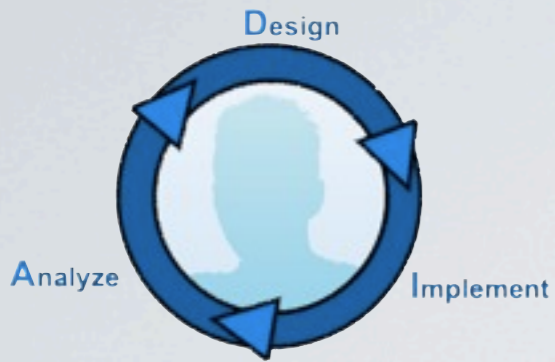






# PROTOTYPE I





# PROTOTYPE 2

**Available Filters**

- Everything
  - WHERE Stroke Width
  - WHERE Stroke Color
  - WHERE Fill Color
  - WHERE Type
  - WHERE Group
    - IS Layer 1
      - AND Stroke Width
      - AND Stroke Color
        - IS none
        - IS NOT none
          - AND Type
            - IS [red square]
            - IS NOT [red square]
          - AND Fill Color
          - AND Type
        - IS NOT Layer 1
        - IS Lamellen
        - IS NOT Lamellen
        - IS Layer Engrave
        - IS NOT Layer Engrave
        - IS Layer Cut
        - IS NOT Layer Cut
        - IS Layer Cut 2

**Matching Elements**

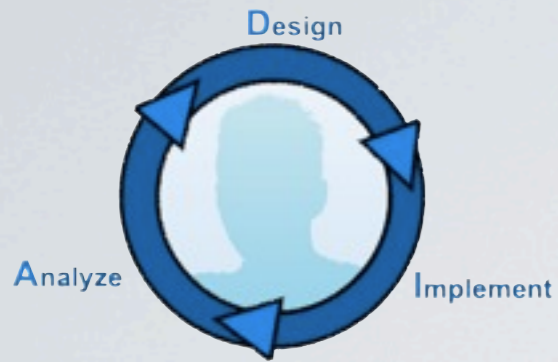
**Map to Profile**

- Cut Line
- Engraved Line
- Engrave Drawing
- Engrave Photo
- Engrave 3D

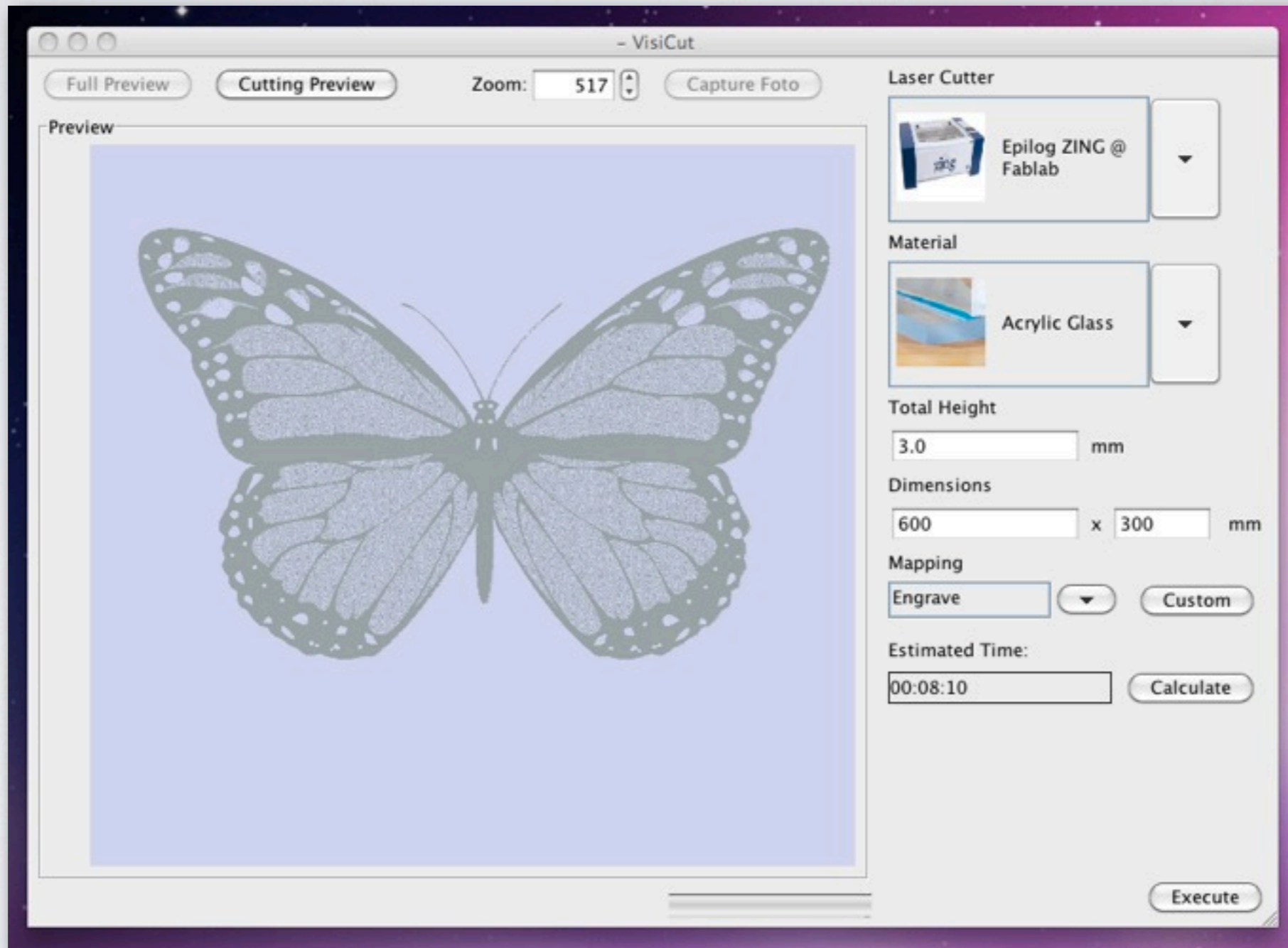
**Mappings:**  Preview Mode

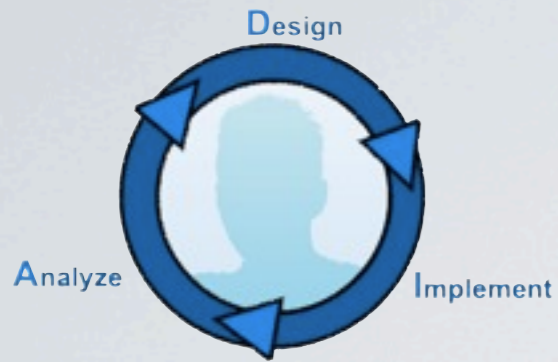
Filters	Profile	Use Outline
Stroke Width IS 0.00762 AND Stroke Color IS [red square] AND Type IS Line	Engrave Drawing	<input type="checkbox"/>
Fill Color IS [orange square] AND Type IS NOT Path	Cut Line	<input type="checkbox"/>
Group IS Layer Cut	Cut Line	<input type="checkbox"/>

Buttons: +, -, Edit, Cancel, OK

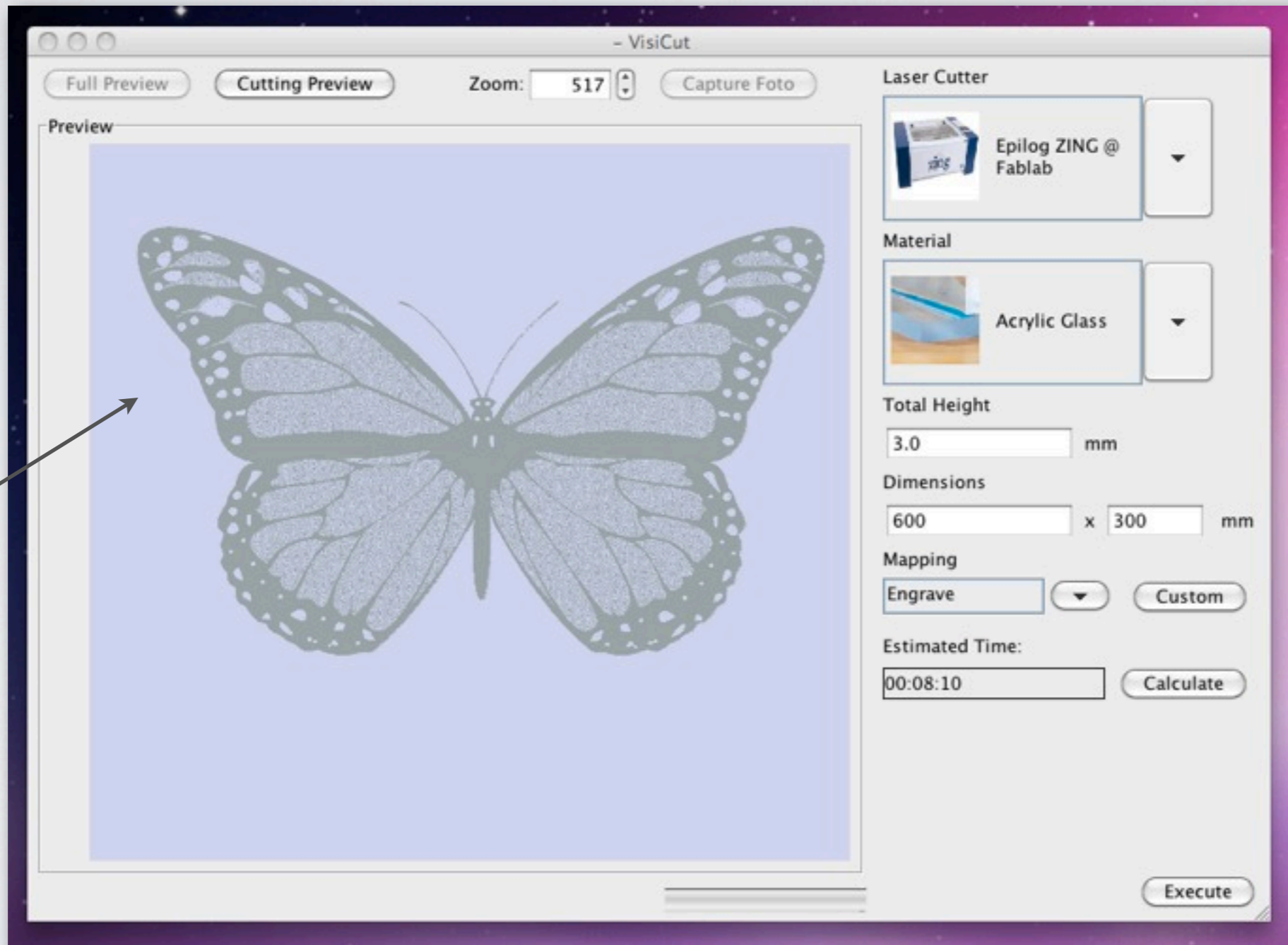


# FINAL VERSION



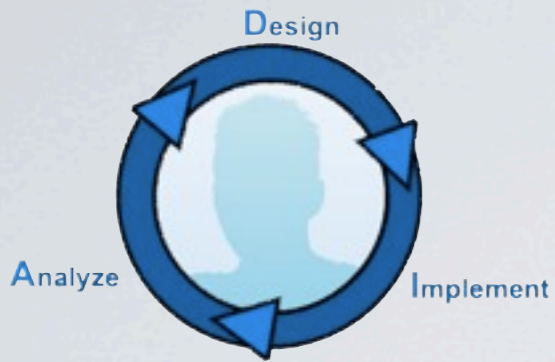


# FINAL VERSION

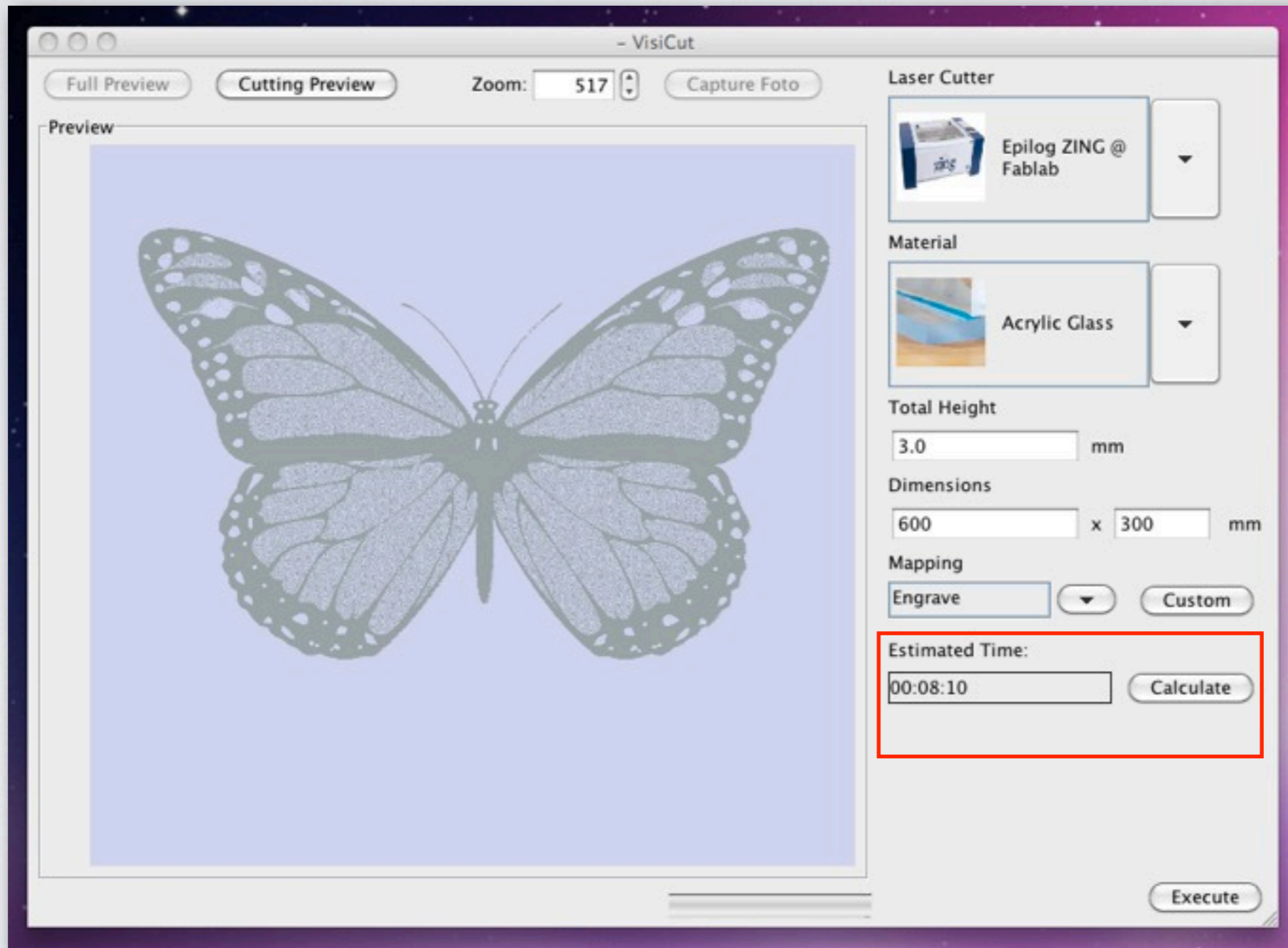


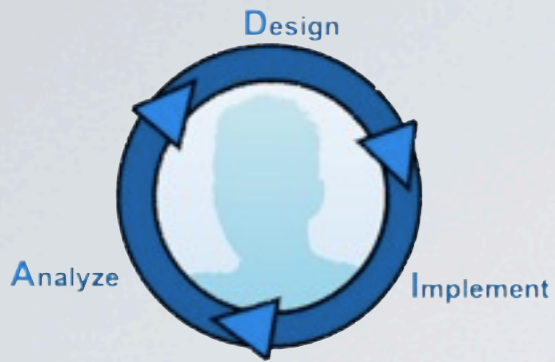
Drag & Drop





# FINAL VERSION





# FINAL VERSION

Full Preview Cutting Preview Zoom: 517 Capt

Preview

Laser Cutter

Epilog ZING @ Fablab

Please select

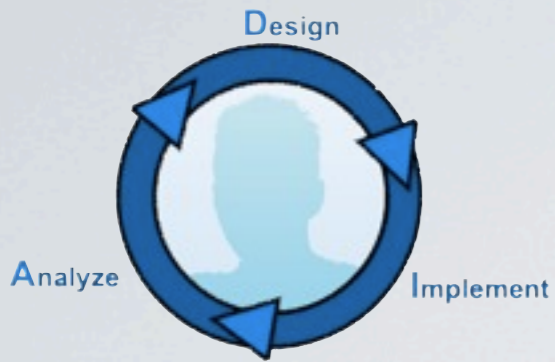
Epilog ZING @ Fablab

Trotec SP 1500 !@ Fablab  
Material not supported

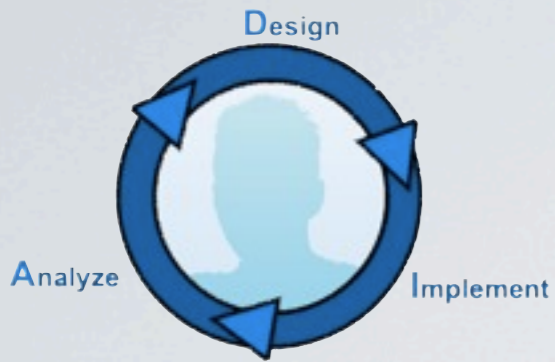
Dimensions

00:08:10 Calculate

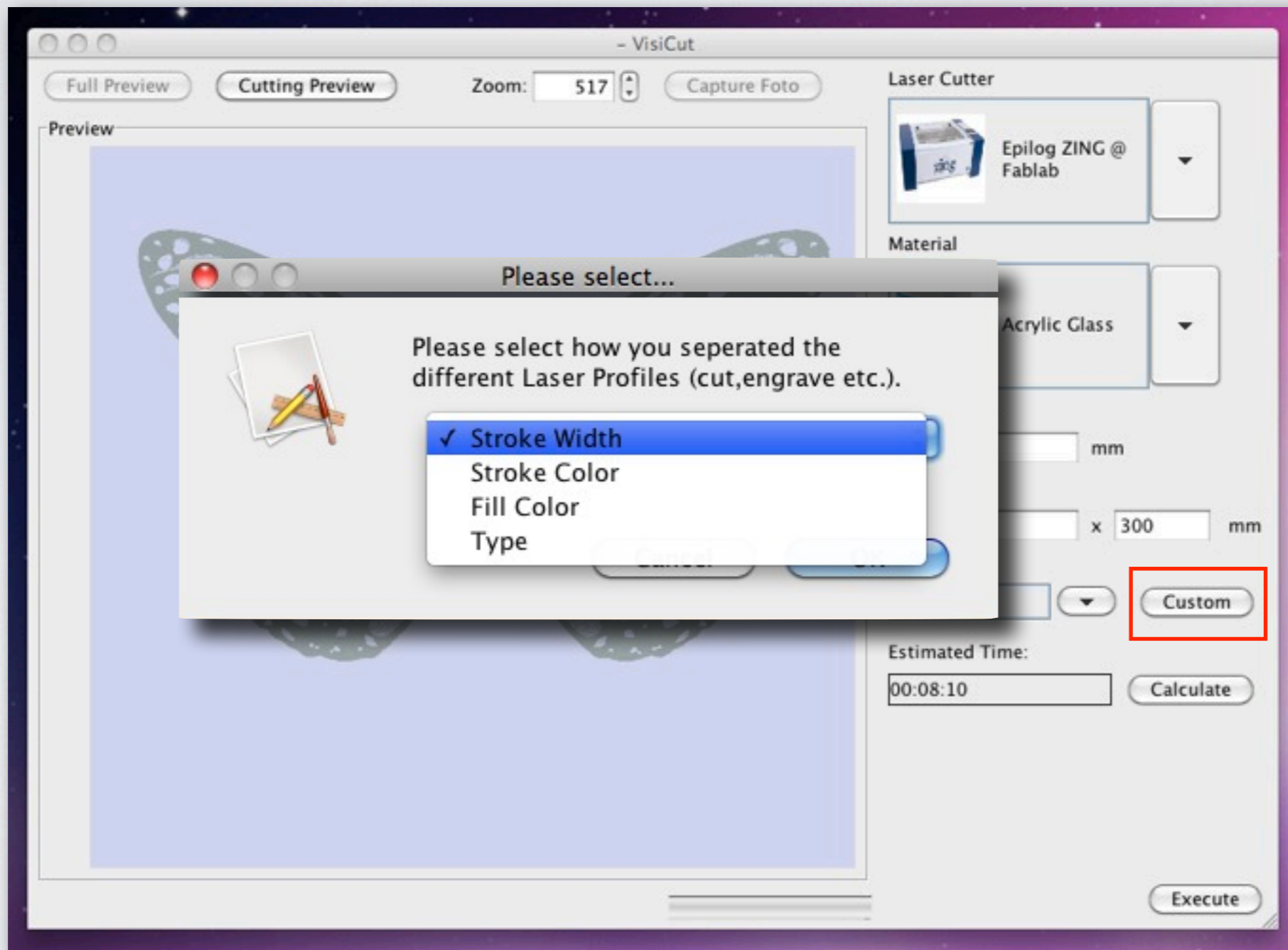
Execute



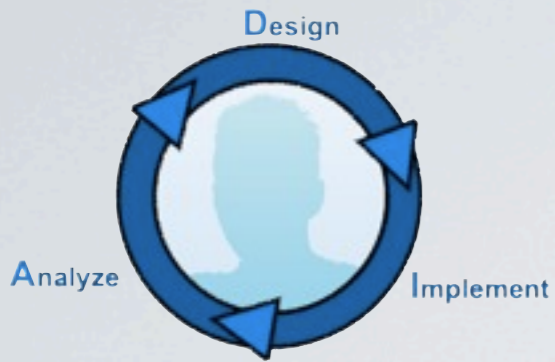
# FINAL VERSION



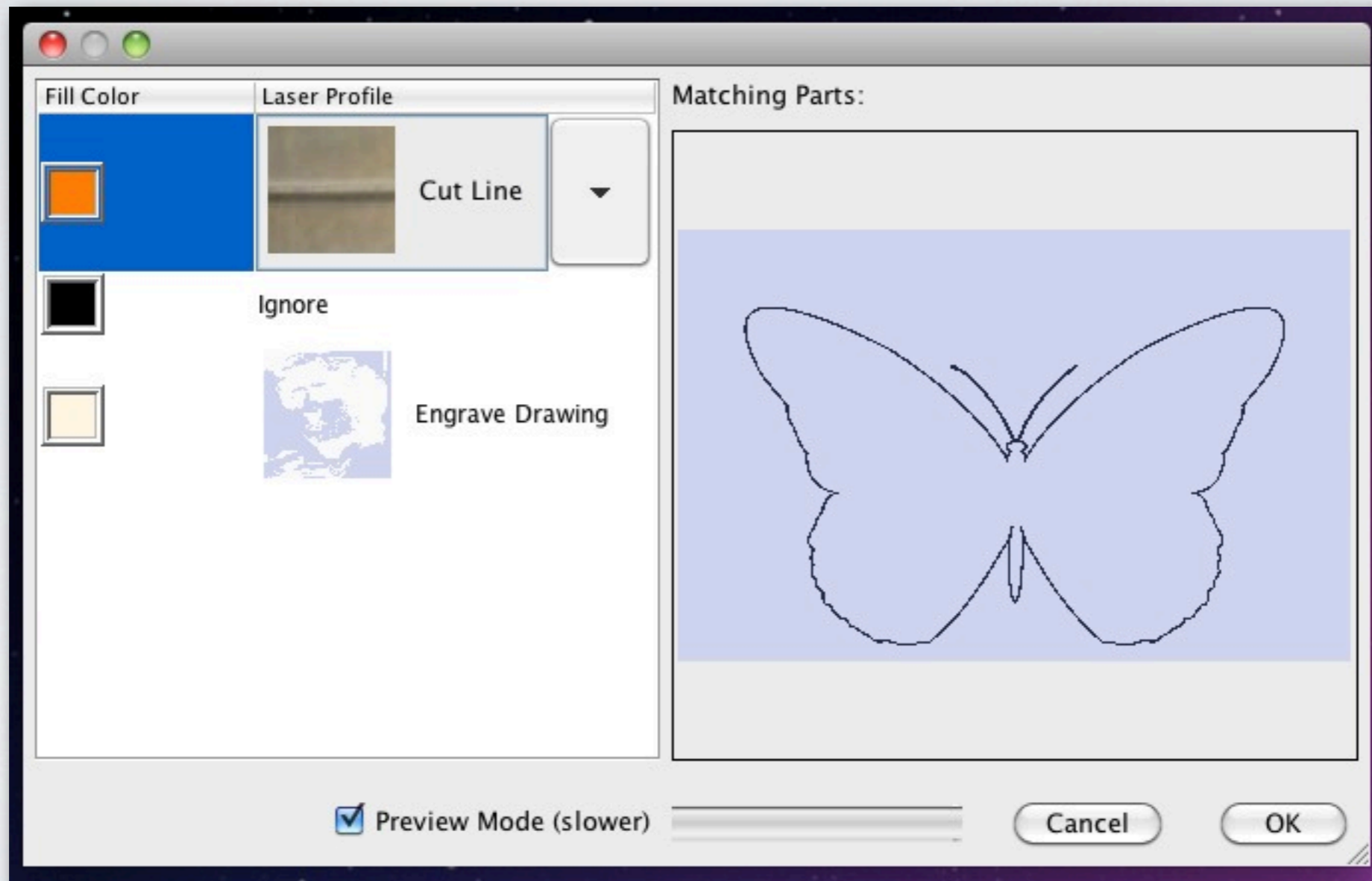
# FINAL VERSION



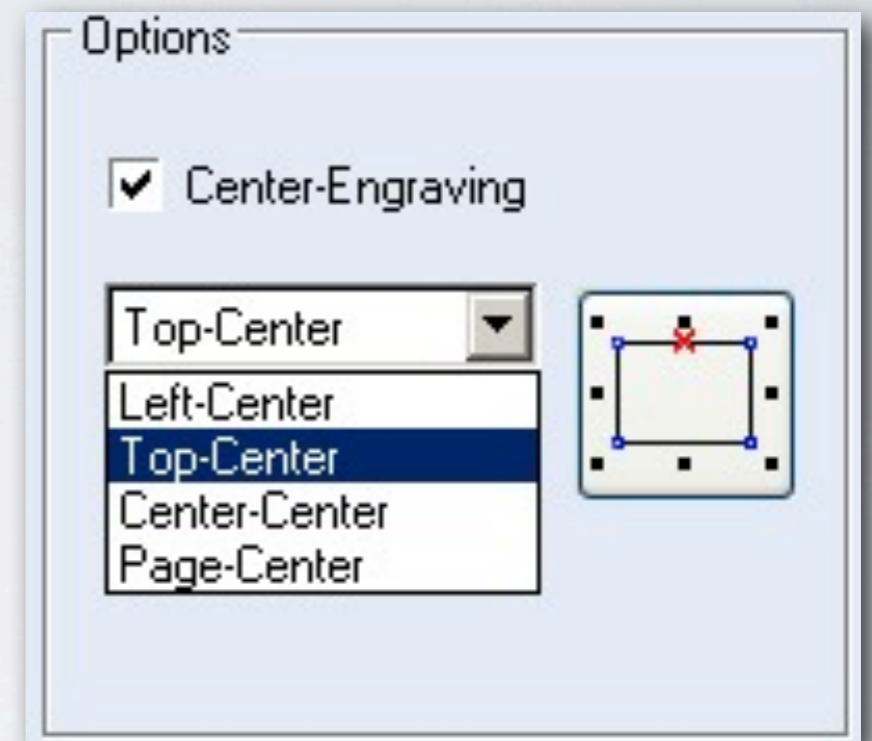




# FINAL VERSION

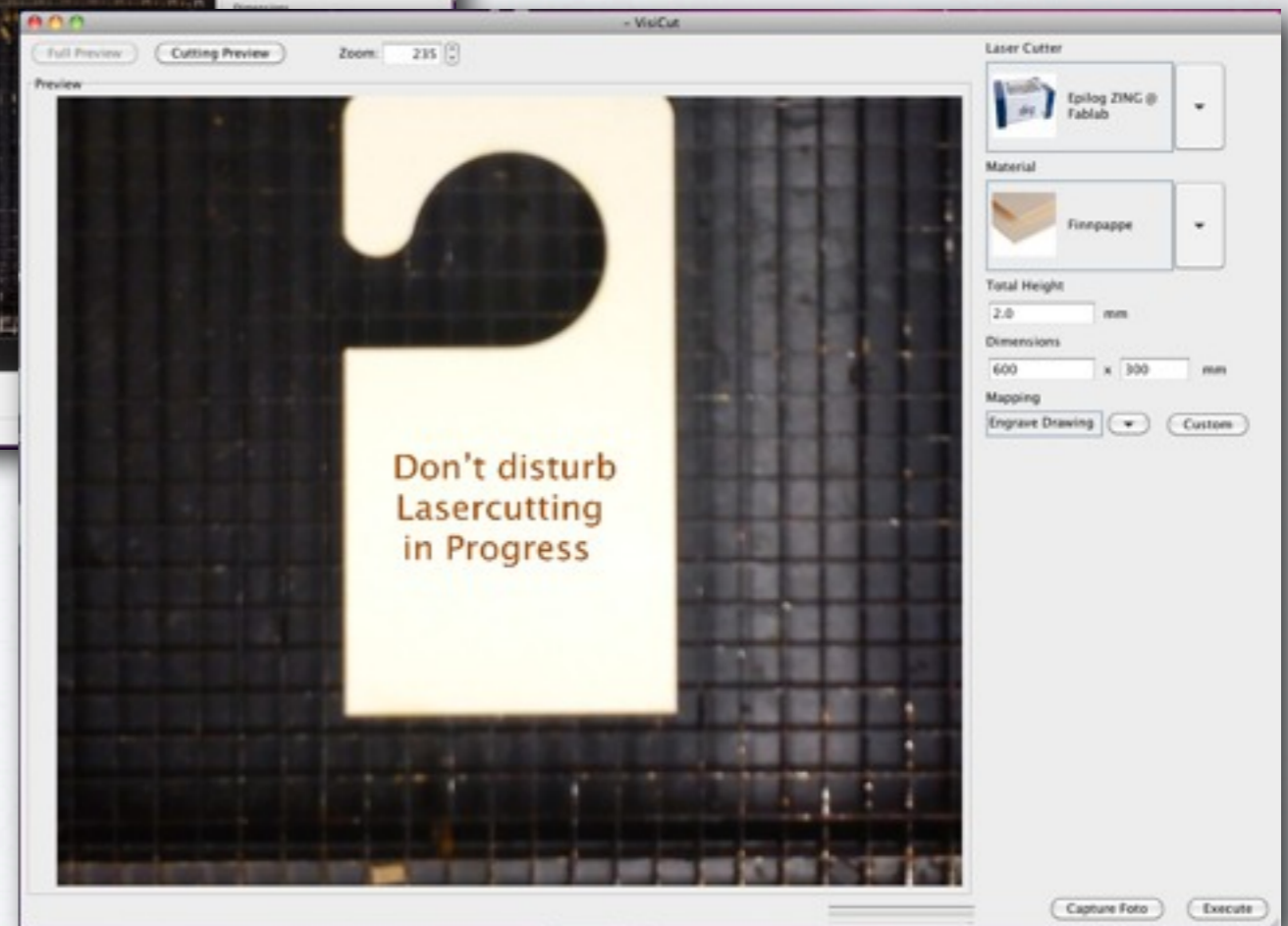
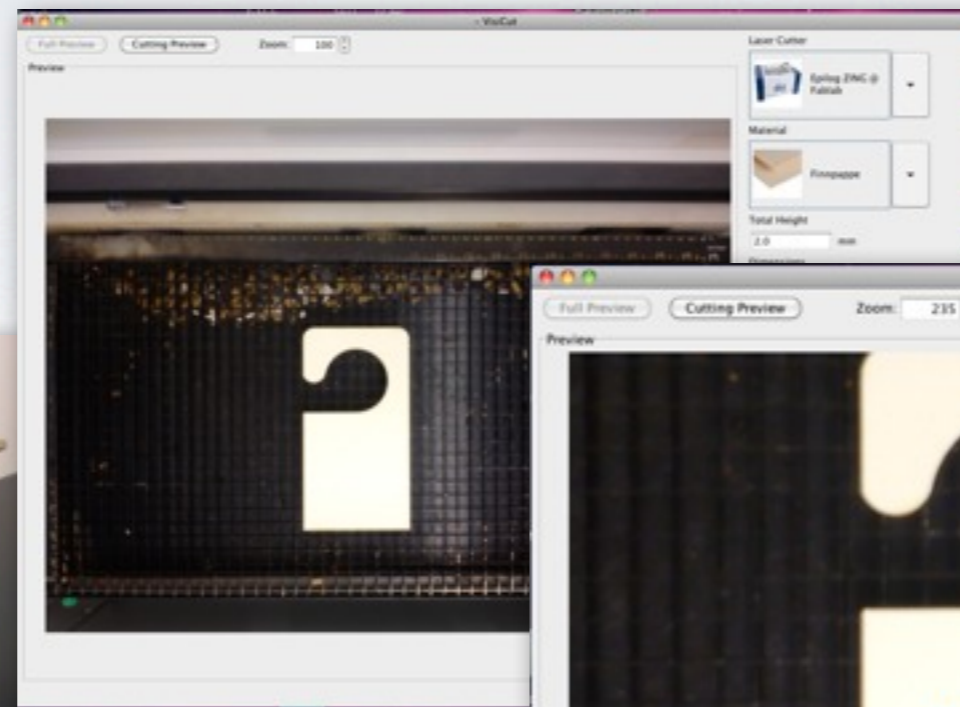


# THE POSITIONING PROBLEM

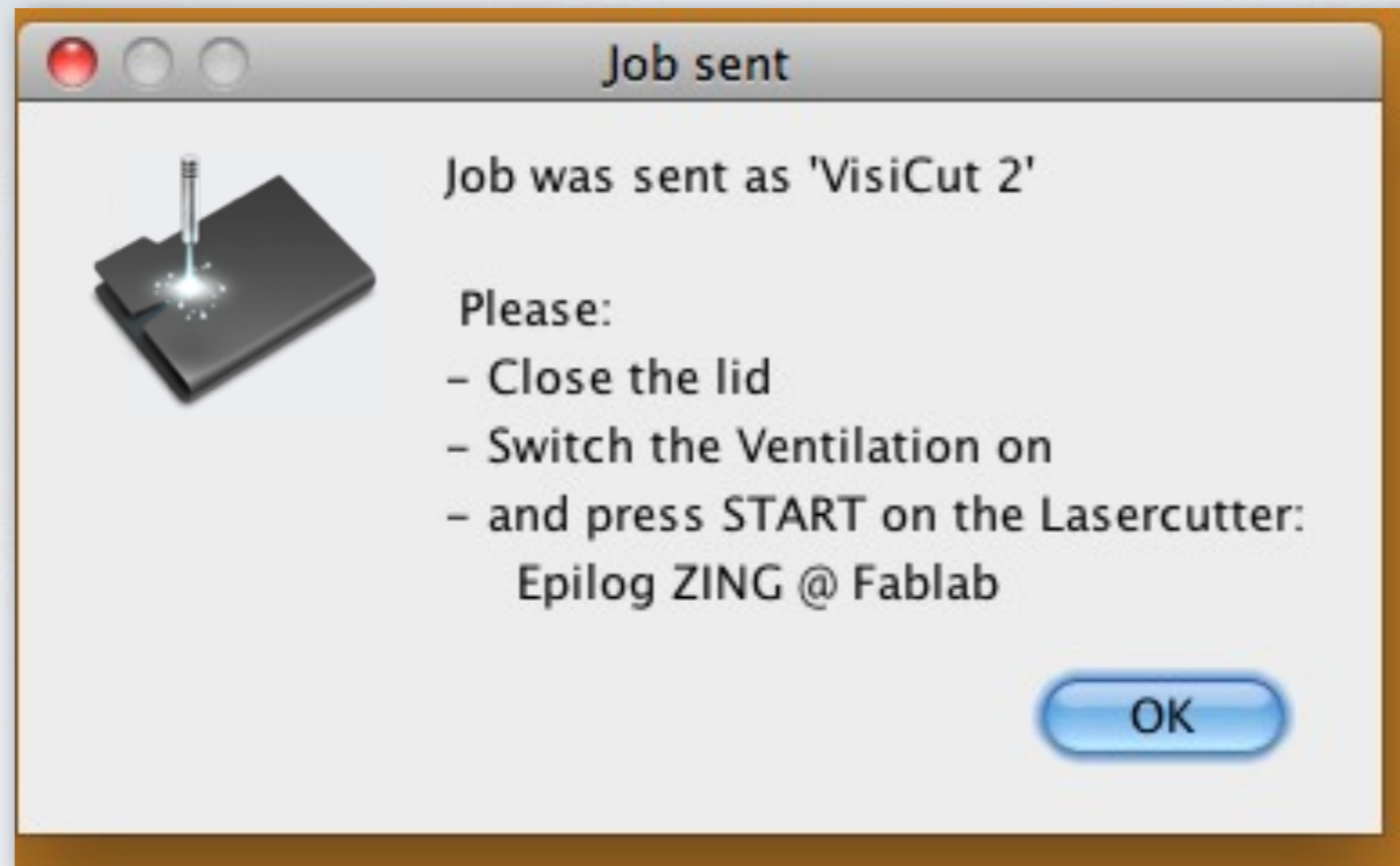


# THE CAMERA

Lasercutting à la WYSWIG

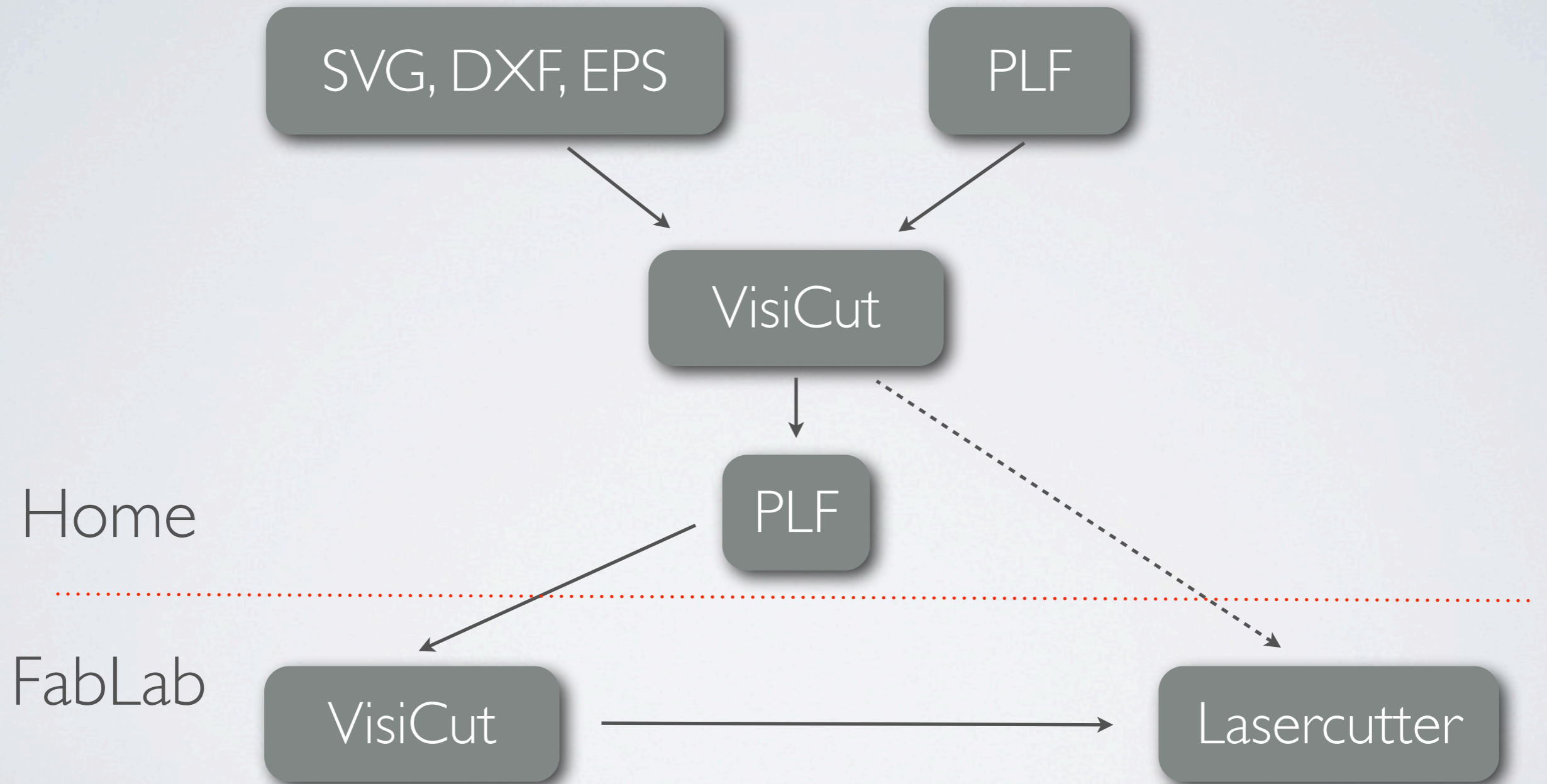


# AS EASY AS IT SOUNDS

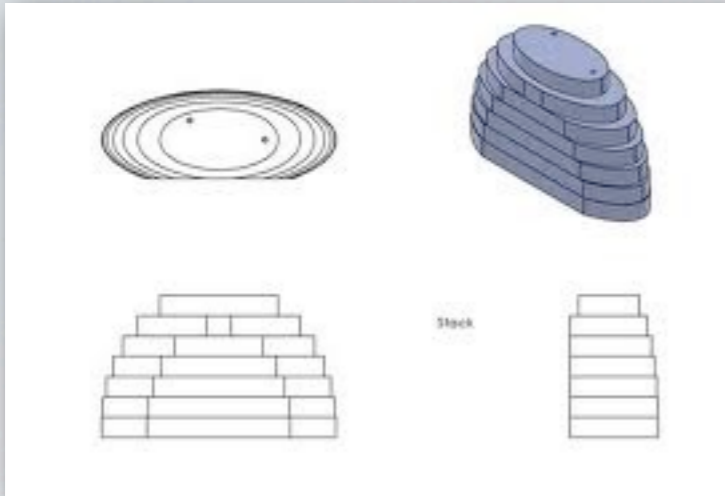


- no focus setting
- no head moving

# VISICUT - WORKFLOW



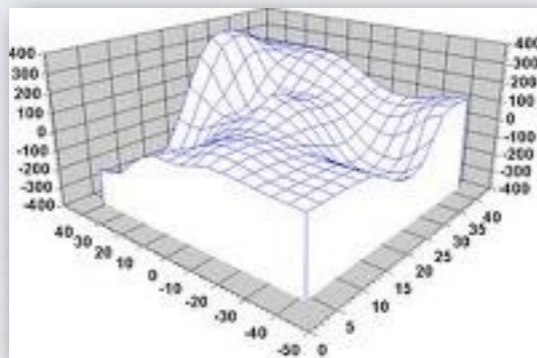
# FUTURE WORK



3D Slices Preview



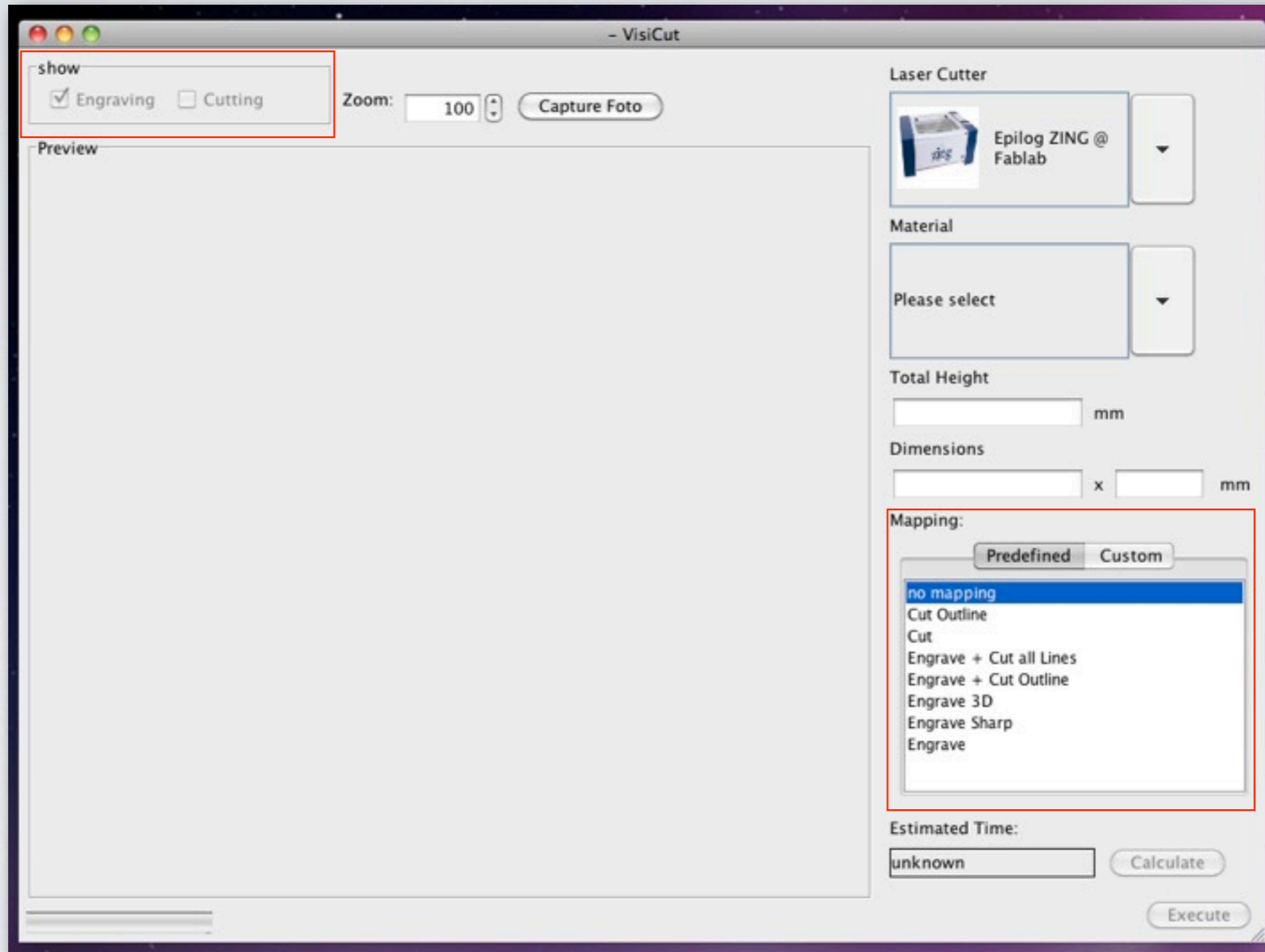
3D Rendering



Lasercut non planar surface

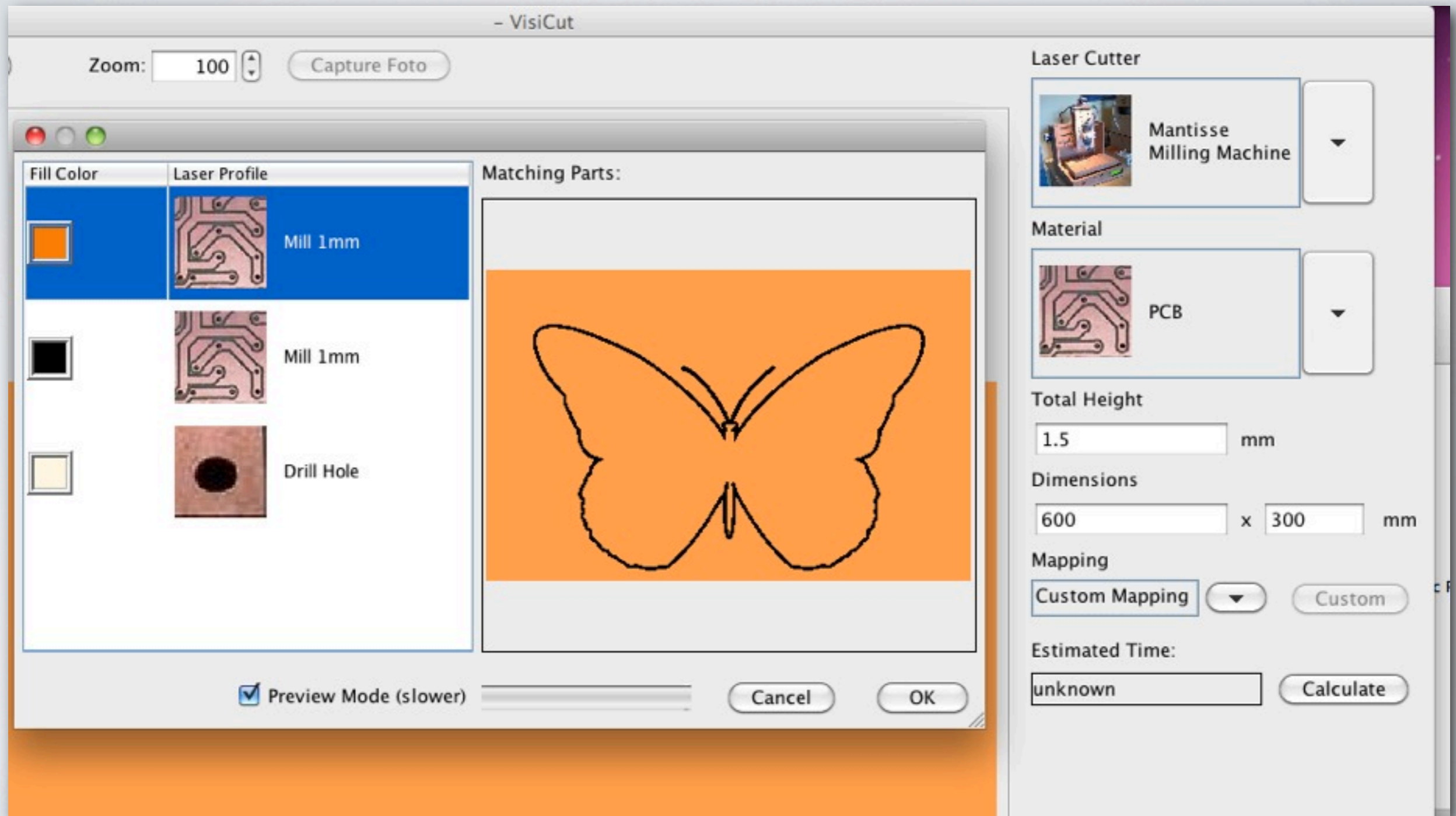


# FINETUNING





# A G-CODE DRIVER



# INPUT FILE FORMATS

```
public interface GraphicObject
{
    public Rectangle2D getBoundingBox();
    /**
     * Returns a list of attribute values for the given
     * Attribute.
     * @param name
     * @return
     */
    public List<Object> getAttributeValues(String name);
    /**
     * Returns a List of Attributes where the Object
     * has values set
     * @return
     */
    public List<String> getAttributes();

    /**
     * Renders the Object on the given Graphics2D
     * @param g
     */
    public void render(Graphics2D g);

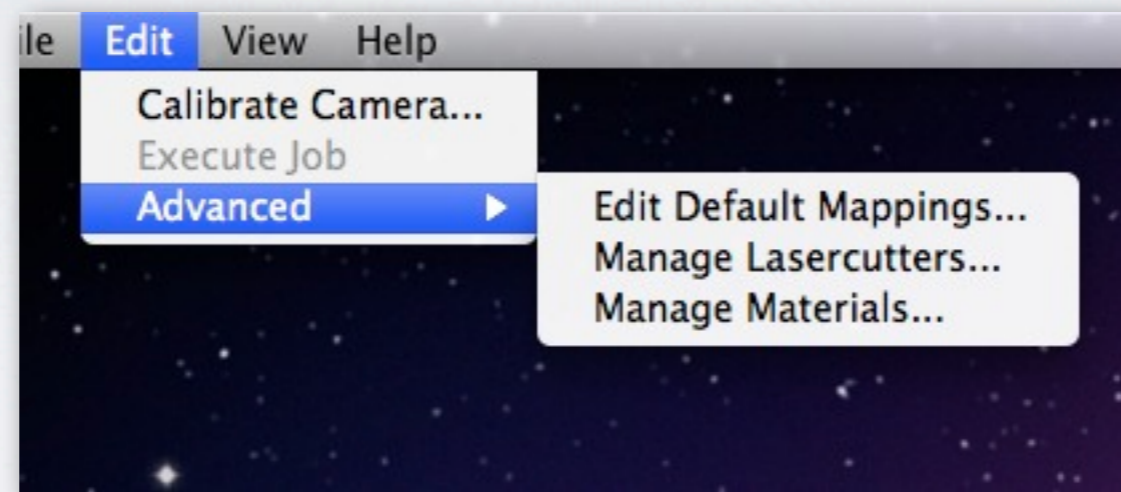
    /**
     * Returns a Shape representing the Object
     * @return
     */
    public abstract Shape getShape();
}
```

# PORTABLE LASER FORMAT

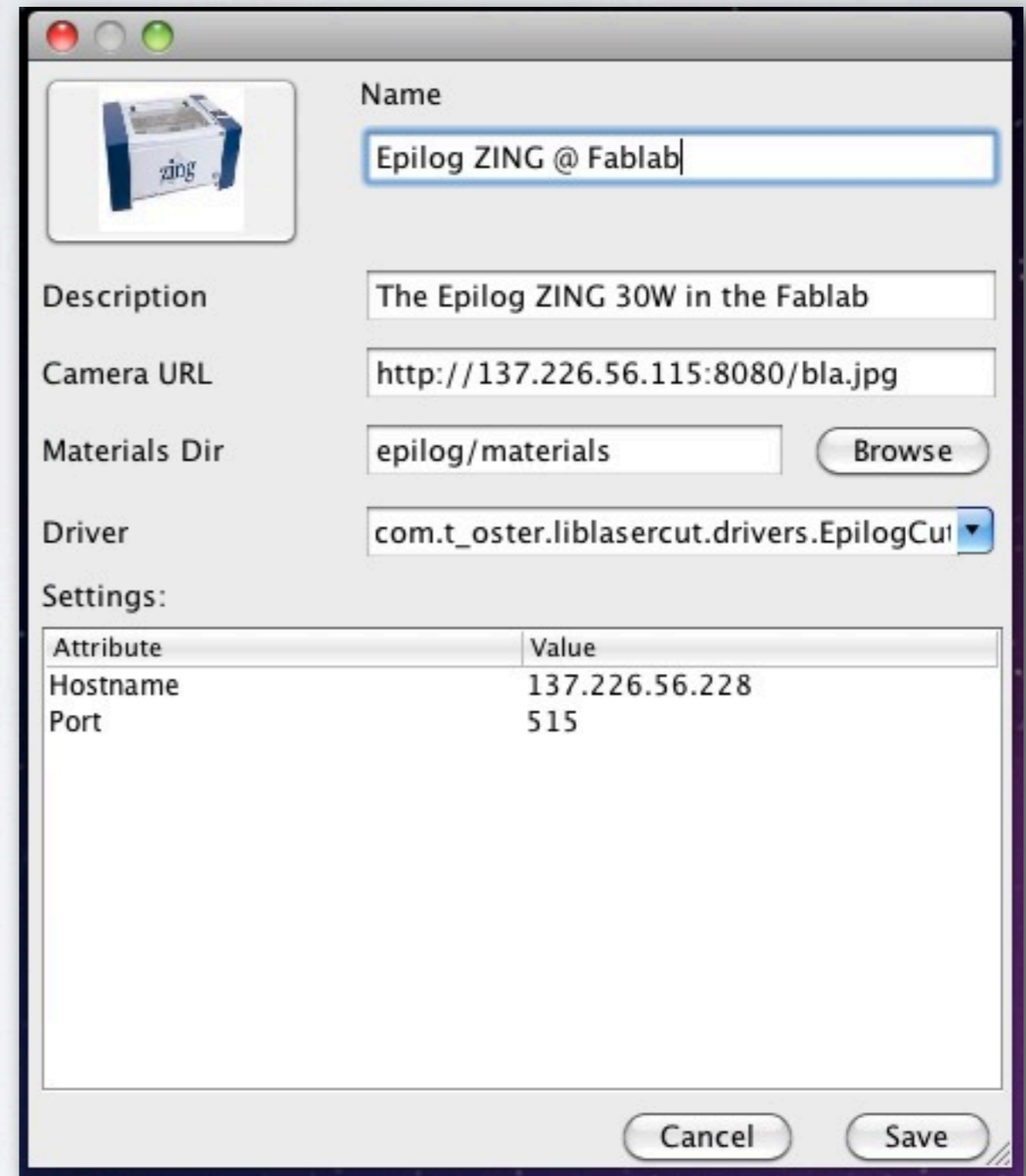
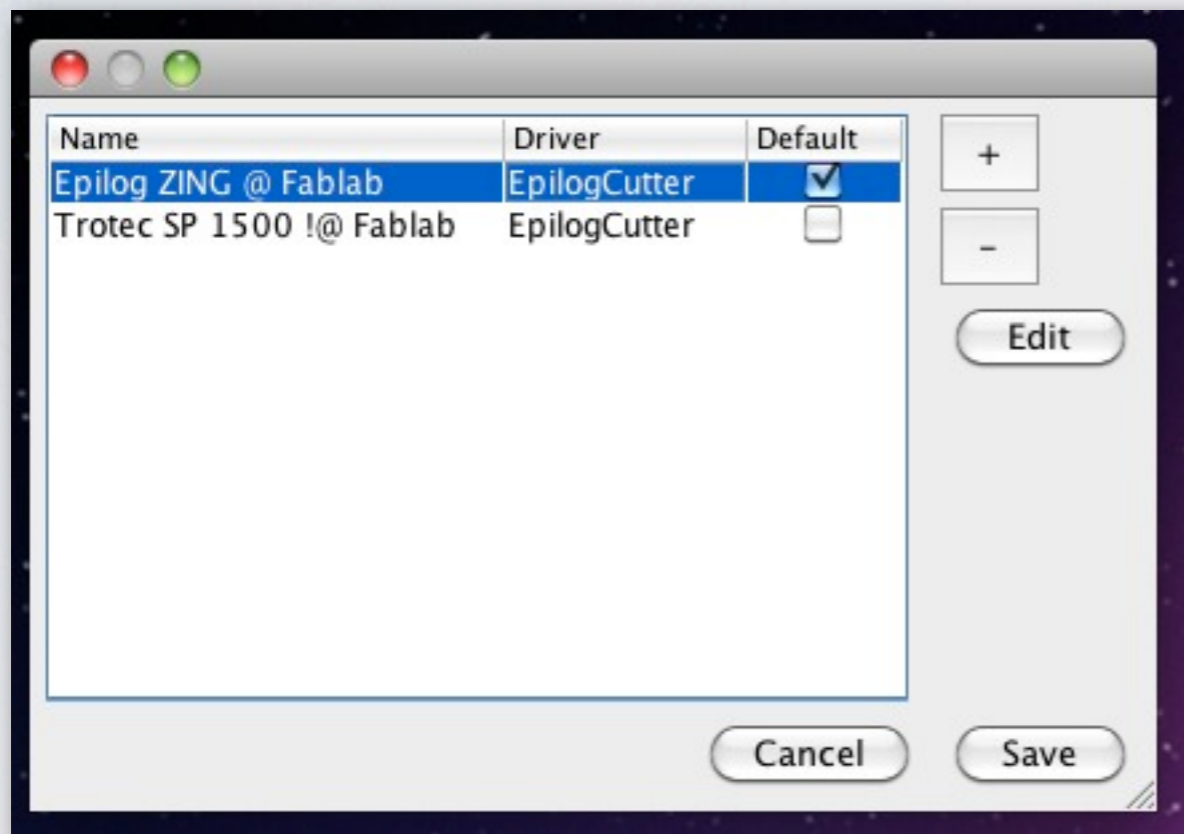
## ZIP Container

- Input File (Original Format)
- Mapping (XML)
- Transformation (XML)

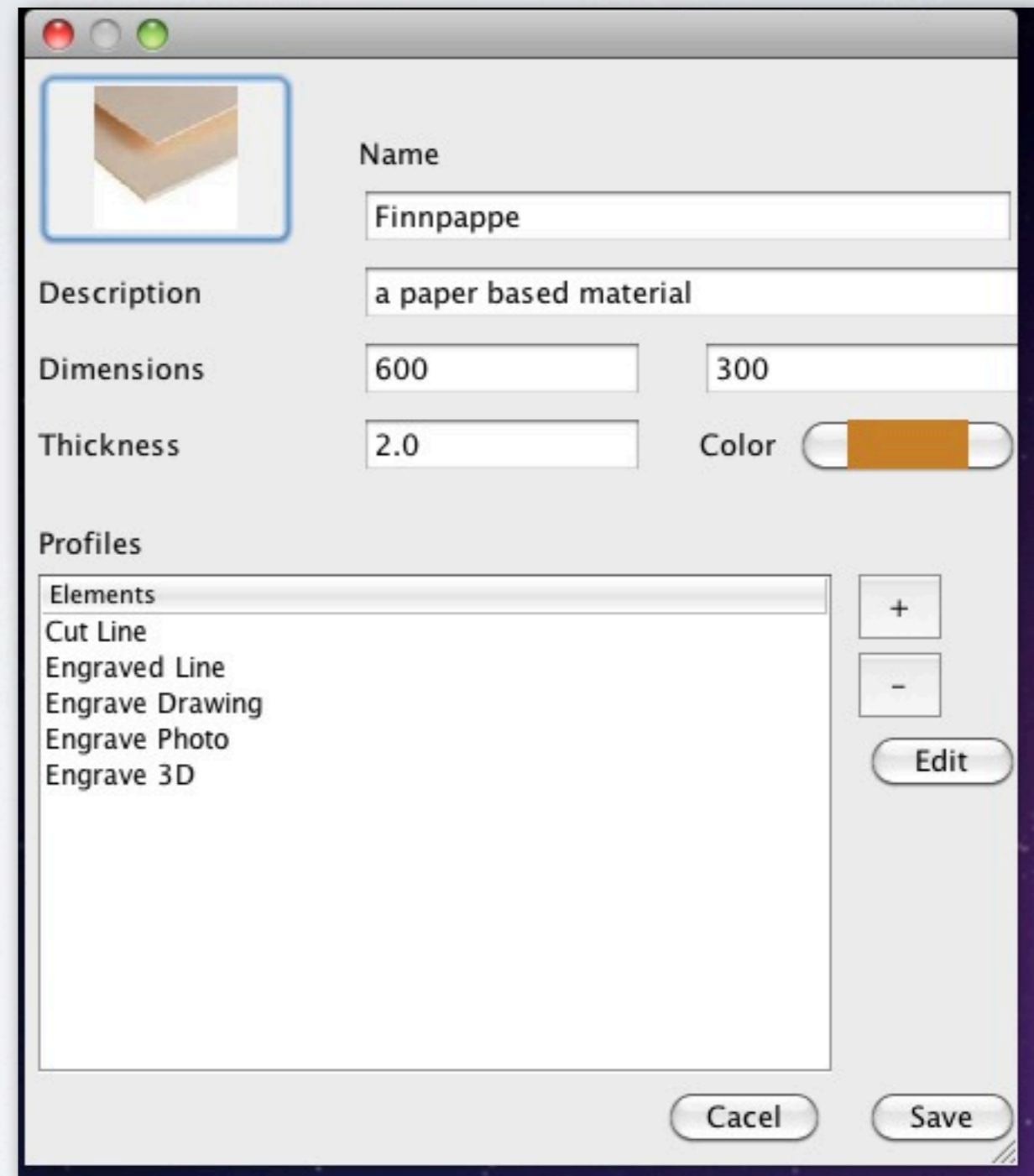
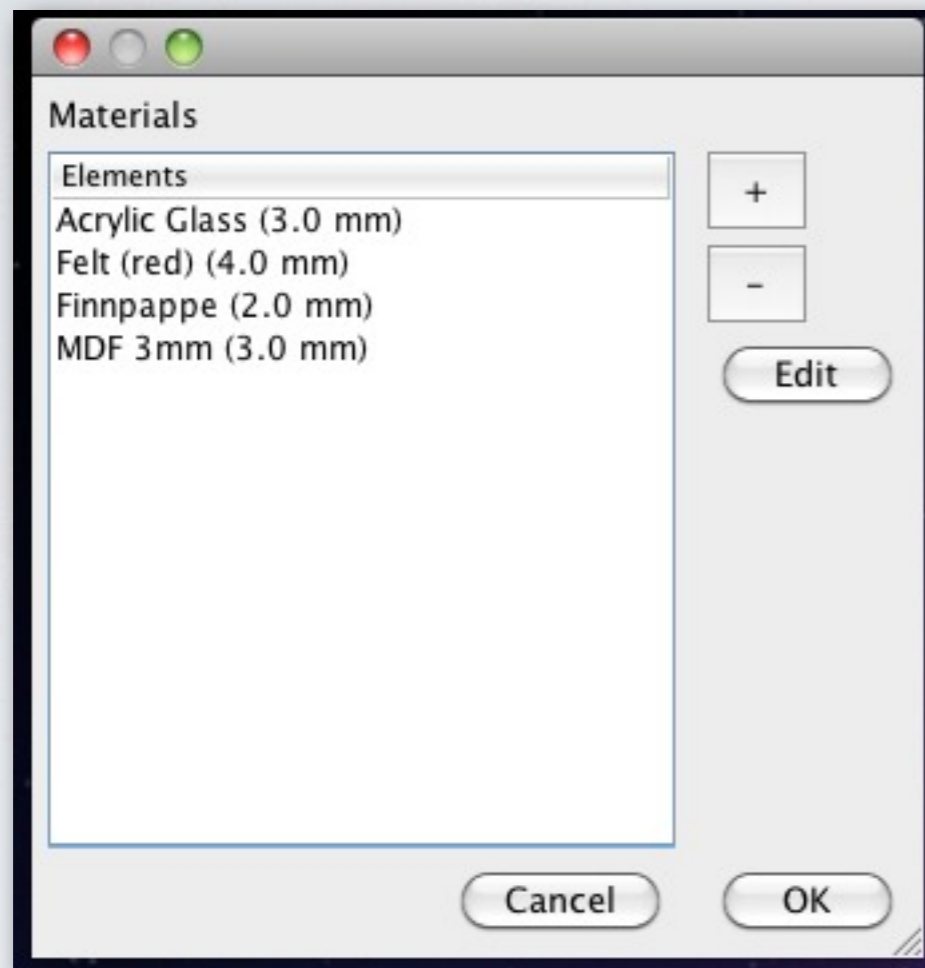
# ADVANCED OPTIONS



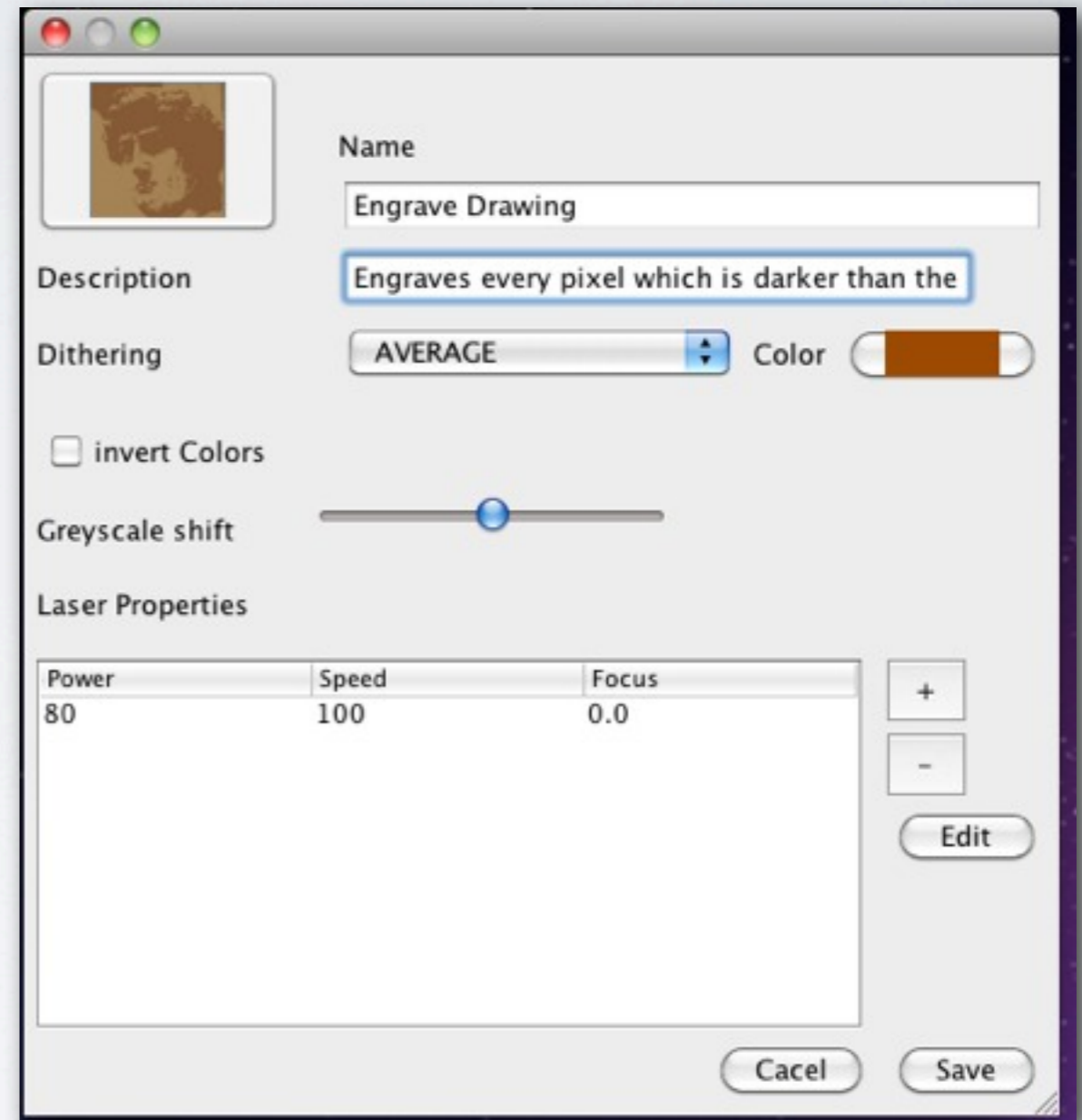
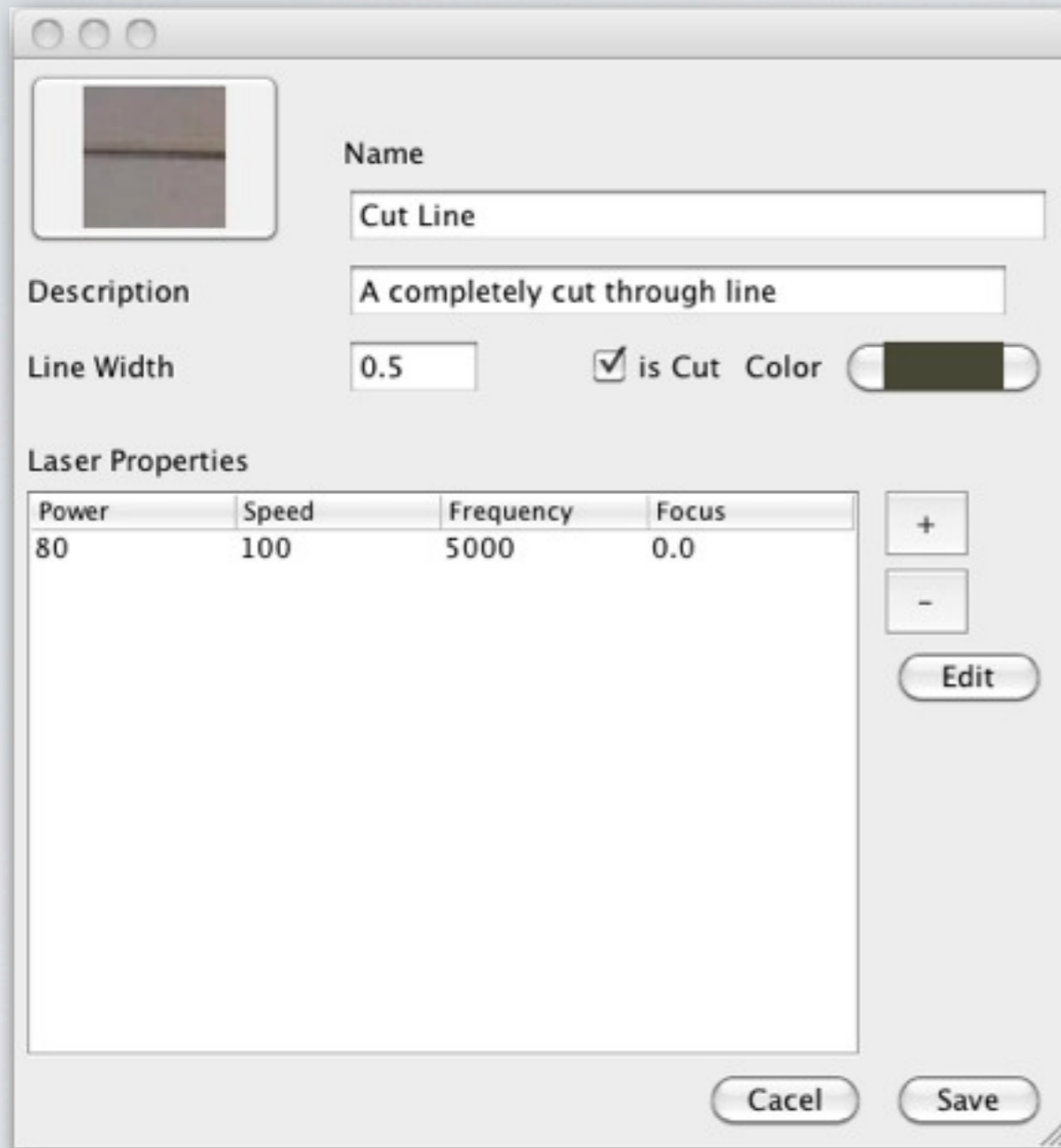
# MANAGER LASERCUTTERS



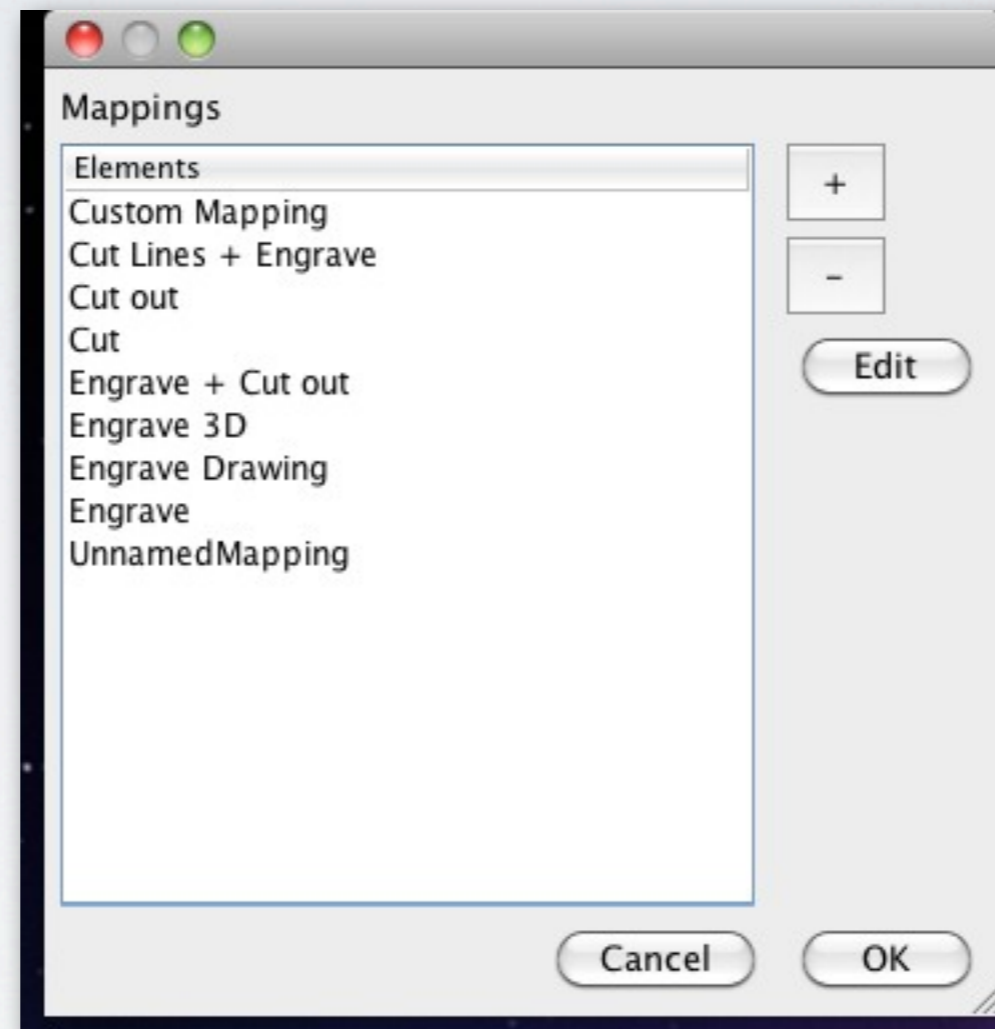
# MANAGE MATERIALS



# LASER PROFILES



# DEFAULT MAPPINGS





# INPUT FILE FORMATS

```
public interface GraphicObject
{
    public Rectangle2D getBoundingBox();
    /**
     * Returns a list of attribute values for the given
     * Attribute.
     * @param name
     * @return
     */
    public List<Object> getAttributeValues(String name);
    /**
     * Returns a List of Attributes where the Object
     * has values set
     * @return
     */
    public List<String> getAttributes();

    /**
     * Renders the Object on the given Graphics2D
     * @param g
     */
    public void render(Graphics2D g);

    /**
     * Returns a Shape representing the Object
     * @return
     */
    public abstract Shape getShape();
}
```

# INPUT FILE FORMATS

```
public interface GraphicObject
{
    public Rectangle2D getBoundingBox();
    /**
     * Returns a list of attribute values for the given
     * Attribute.
     * @param name
     * @return
     */
    public List<Object> getAttributeValues(String name);
    /**
     * Returns a List of Attributes where the Object
     * has values set
     * @return
     */
    public List<String> getAttributes();

    /**
     * Renders the Object on the given Graphics2D
     * @param g
     */
    public void render(Graphics2D g);

    /**
     * Returns a Shape representing the Object
     * @return
     */
    public abstract Shape getShape();
}
```