



# Degrafa

# Degrafa

## Declarative Graphics Framework

John Mason  
[mason@fusionlink.com](mailto:mason@fusionlink.com)



## What is Degrafa?

- A declarative graphical framework for Flex and AIR
- Open source project - [www.degrafa.com](http://www.degrafa.com)
- Developers:
  - Jason Hawryluk
  - Juan Sanchez
  - Andy McIntosh
  - Ben Stucki
  - Pavan Podila
  - Jim Cheng
  - Sean Chatman
  - Greg Dove
  - Tom Gonzales
- Under the MIT open source license



## Goals of Degrafa

- Create dynamic graphical assets within Flex / AIR
- Provide access to the Drawing API through MXML
- Tag based, eliminates the need for complex ActionScript

```
graphics.beginFill(#666666, .5);
```

- or -

```
<fill>
```

```
  <SolidFill color="#666" alpha=".5"/>
```

```
</fill>
```

- Reduce the need for Static External Assets  
 Embed(source='image.png');
- Advance stuff: data binding, CSS and SVG



## How to install Degrafa

- Download the latest Degrafa.swc from <http://code.google.com/p/degrafa>
- Add the Degrafa.swc file into the your project's library path
  - Consider using the RSL feature
- Add the XML namespace  
`xmlns:degrafa="http://www.degrafa.com/2007"`
- You should then be able to use the Degrafa markup language  
`<degrafa:Surface left="0" top="0">`



Two ways to start creating graphics

- `<Surface../>` tag

```
<degrafa:Surface>  
  <degrafa:GeometryGroup>  
    <!-- Add Degrafa Shapes here -->  
  </degrafa:GeometryGroup>  
</degrafa:Surface>
```

- `graphicsTarget="{[...]}"`

- to draw to a Flex Component like Canvas, Button, etc.

```
<degrafa:GeometryComposition  
  graphicsTarget="{[myButton]}">  
  <!-- Add Degrafa Shapes here -->  
</degrafa:GeometryComposition>
```



## Objects

-You have several shapes in Degrafa that you can use: Circle, Line, Polygon, RegularRectangle, AdvancedRectangle, QuadracticBezier, etc.

Pre-composed library objects, create your own and reference when you need them



## Fills & Strokes

- Fills: solid, gradients, bitmaps, blends

```
<degrafa:SolidFill color="#eee" alpha=".6"/>
```

- Strokes: solid, gradients

```
<degrafa:SolidStroke color="#333" alpha=".8"/>
```



Groups: <GeometryGroup..>

- group geometry objects

Geometry Composition: <GeometryComposition..>

- compose objects outside of <Surface>
- apply using graphicsTarget="{[...]}"
- reuse objects as new MXML shapes

Compose shapes within other shapes

```
<Circle>
```

```
  <Polygon/>
```

```
</Circle>
```





## Colors

- You can use short-hand notation, #C50 versus #CC5500
- you can also use color names like “red”, “black”, etc.
- RGB & CMYK supported

#0F0

0,255,0

63,0,100,0



## Advance CSS Support

- Border control
- Multiple background images
- Gradients
- Blend modes
- Shorthand declarations

## Dynamic Skins

- Use an MXML file for each skin state
- Or use one MXML file for all skin states
  
- Repeaters
- Derivatives
- Cloning



## Scalable Vector Graphics (SVG)

- Export Illustrator artwork as SVG and use path data in Degrafa
- A SVG-to-Degrafa converter is coming
  - converts SVG data to Degrafa markup
  - AIR application



## Recommended Links

- [degrafa.com](http://degrafa.com)
- [scalenine.com](http://scalenine.com)

## - **SVG to Degrafa demo**

-

**[www.brightcove.tv/title.jsp?title=1130068155&channel=1130055106](http://www.brightcove.tv/title.jsp?title=1130068155&channel=1130055106)**

## InsideRIA Article

- **[www.insideria.com/2008/05/an-introduction-to-degrafa-1.html](http://www.insideria.com/2008/05/an-introduction-to-degrafa-1.html)**