1. Rendering

# Visual

## Rendering

Rendering in WPF is supported via the class Visual. UIElement extends Visual and FrameworkElement extends UIElement meaning all WPF framework elements can be rendered. As Visual is an abstract class we need to use a subclass in order to render to the screen. The DrawingVisual provides support for geometries and also provides support for simple hit testing. The following code shows how to setup a DrawingVisual that will render a red rectangle

In order to be rendered onto screen Visual needs to be added as the visual child of at least one **UIElement**. As Window ultimately is a subclass of UIElement we can render our DrawingVisual by adding it to the visual tree of the window.

The complete code for a window that renders a single red rectangle is then as follows

 Hierarchy

Depth-first rendering

Hit Testing

# FrameworkElement

## Layout

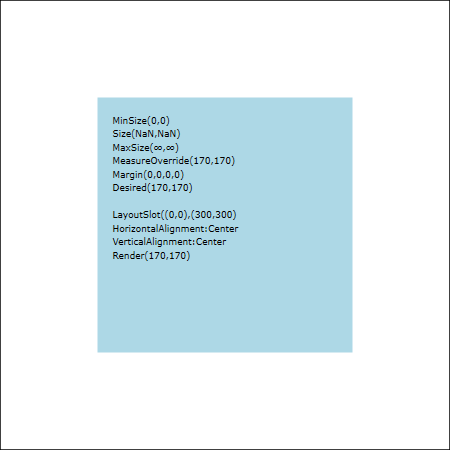
The first thing we will look at is how a single atomic element tries to size itself. A component tries to remain small enough to render its content. This process happens as an interaction between a parent element and its child. A parent will invoke the Measure method on its child FrameworkElement. The FrameworkElement is defined as a non-virtual method in the UIElement baseclass. Measure in turn invokes the MeasureOverride method enabling the subclass to specify its content size. The Measure method then returns a size determined by

* The value returned by measure override
* The min height/width
* The height/width

In this section we will make sure of a FrameworkElement I created which paints diagnostic text showing the values of the layout inputs and outputs as well as painting its given layout slot. Using this class in a simple Window coded as follows



We see the following



And the complete code for this LayoutSlotElement is shown for completeness below.

