ArcGIS API for JavaScript: Dynamic Vector Symbology

Anne Fitz, Kristian Ekenes, and Russell Roberts

https://arcg.is/1STzGK0

2021 ESRI
DEVELOPER SUMMIT

Agenda

- What are CIM Symbols?
- Create a symbol from scratch
- Creating a web style
- Data-driven CIM

What are CIM Symbols?

Anne Fitz

CIM Symbols

High quality, scalable

Scaled vector symbol

Scaled image

Symbol layers

Primitive Overrides

Dynamically update attributes of an individual symbol layer using Arcade

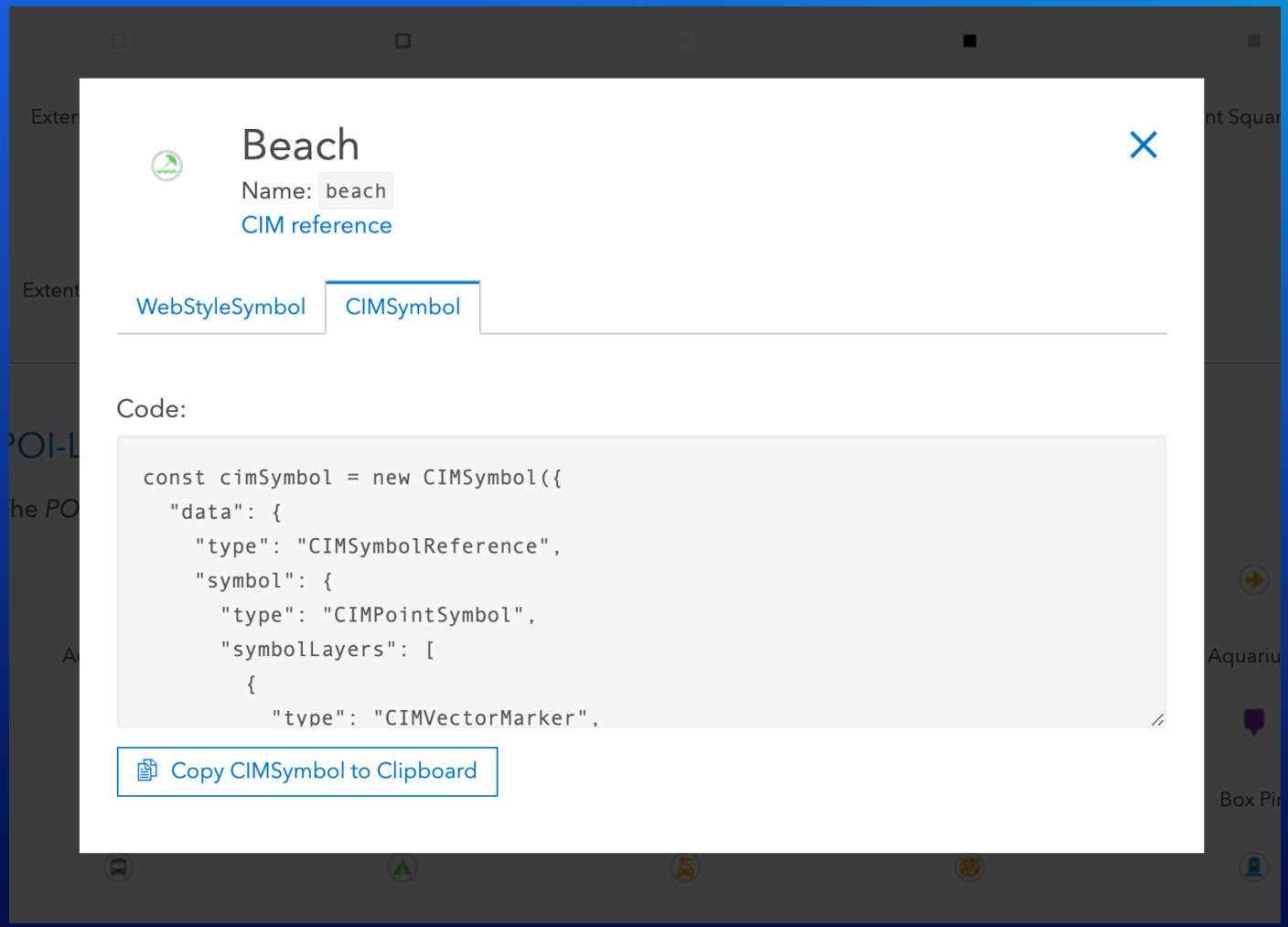
How is it used in the ArcGIS API for JavaScript?

```
// require(["esri/symbols/CIMSymbol"], function(CIMSymbol)
const cimSymbol = new CIMSymbol({
   data: {
      type: "CIMSymbolReference",
      symbol: {
         type: "CIMLineSymbol", // CIMPointSymbol or CIMPolygonSymbol
         symbolLayers: [{ ... }]
      },
      primitiveOverrides: [{ ... }]
   }
});
```

follows the cim-spec

WebStyleSymbols

- Predefined 2D CIM symbols that you can conveniently reference in your apps
- Esri WebStyleSymbols (2D) guide page
- Create your own web style symbols

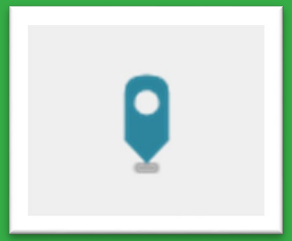


CIMPointSymbol

CIMPictureMarker

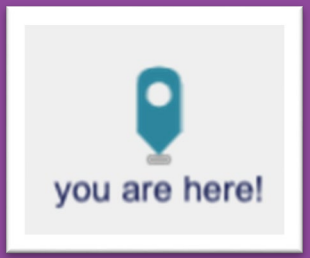
- Created from raster image file

CIMVectorMarker



- Represents vector graphics
- Constructed from MarkerGraphics

CIMTextSymbol

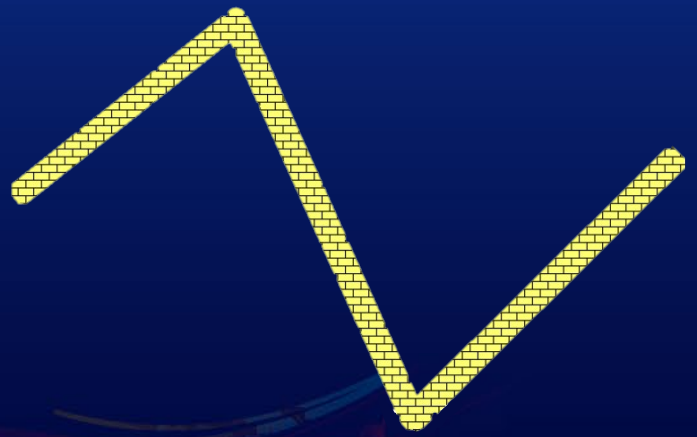


- Supported as a marker graphic on a CIMVectorMarker symbol layer

CIMLineSymbol

CIMSolidStroke

CIMPictureStroke



- Geometric Effects
  - Dashes

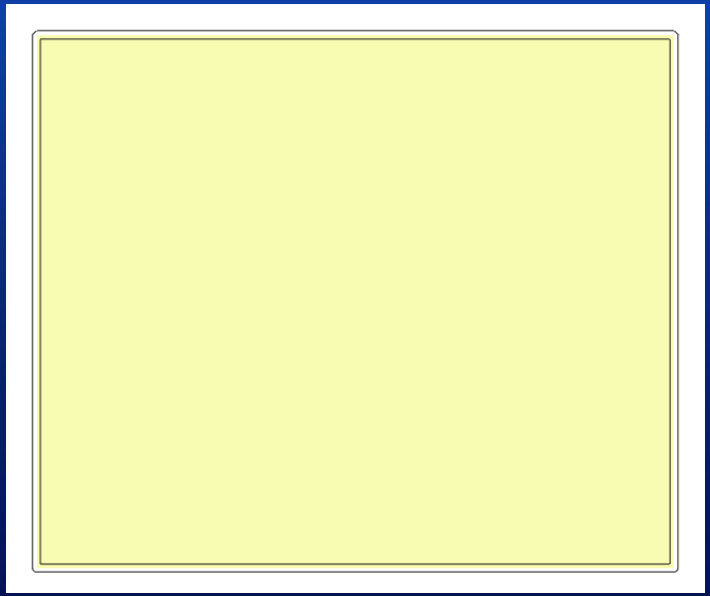
  - Buffer

  - Arrows

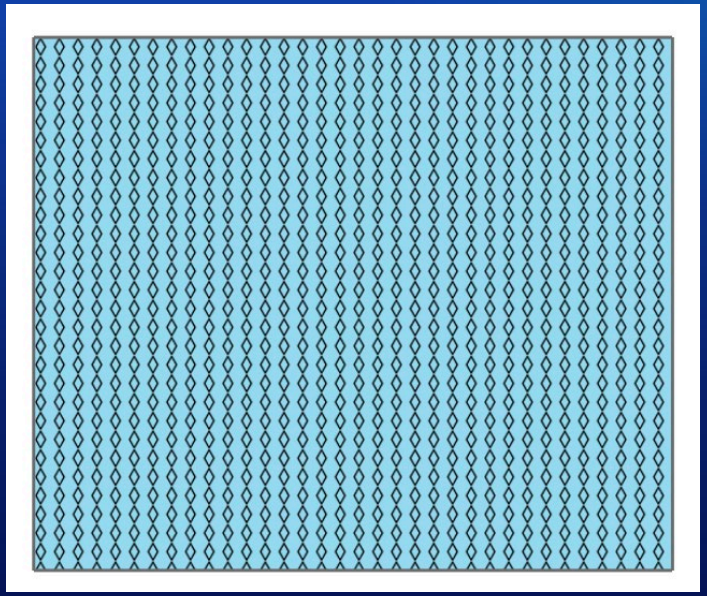
- Marker placement

CIMPolygonSymbol

CIMSolidFill



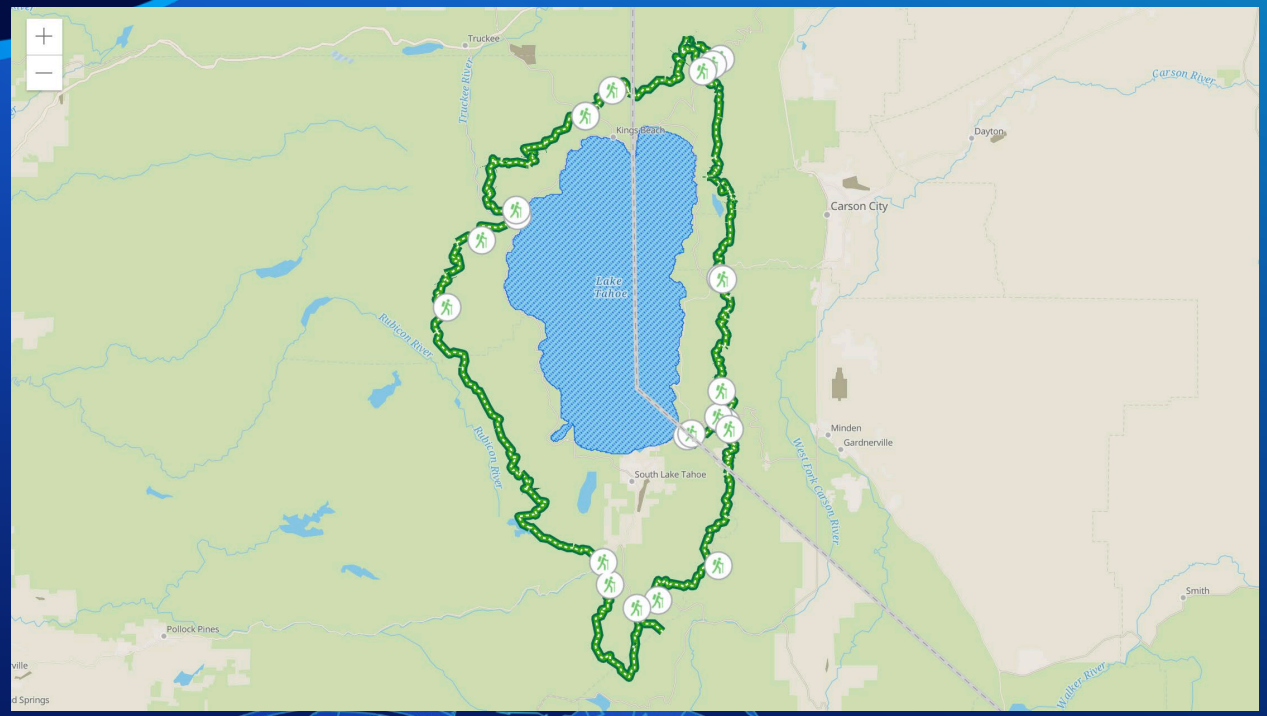
CIMHatchFill



CIMPictureFill

Create a symbol from scratch

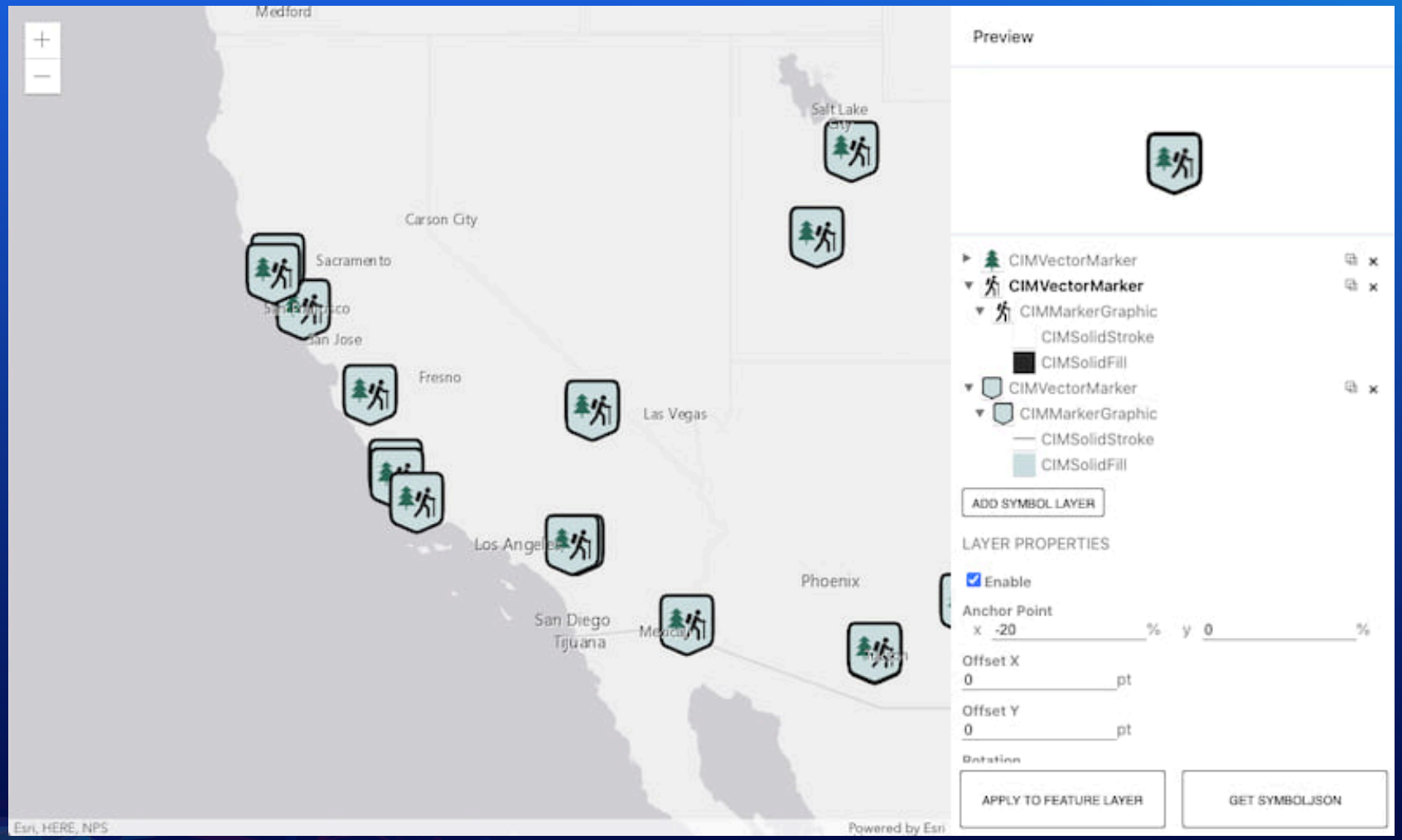
Anne Fitz



CIM Starter App

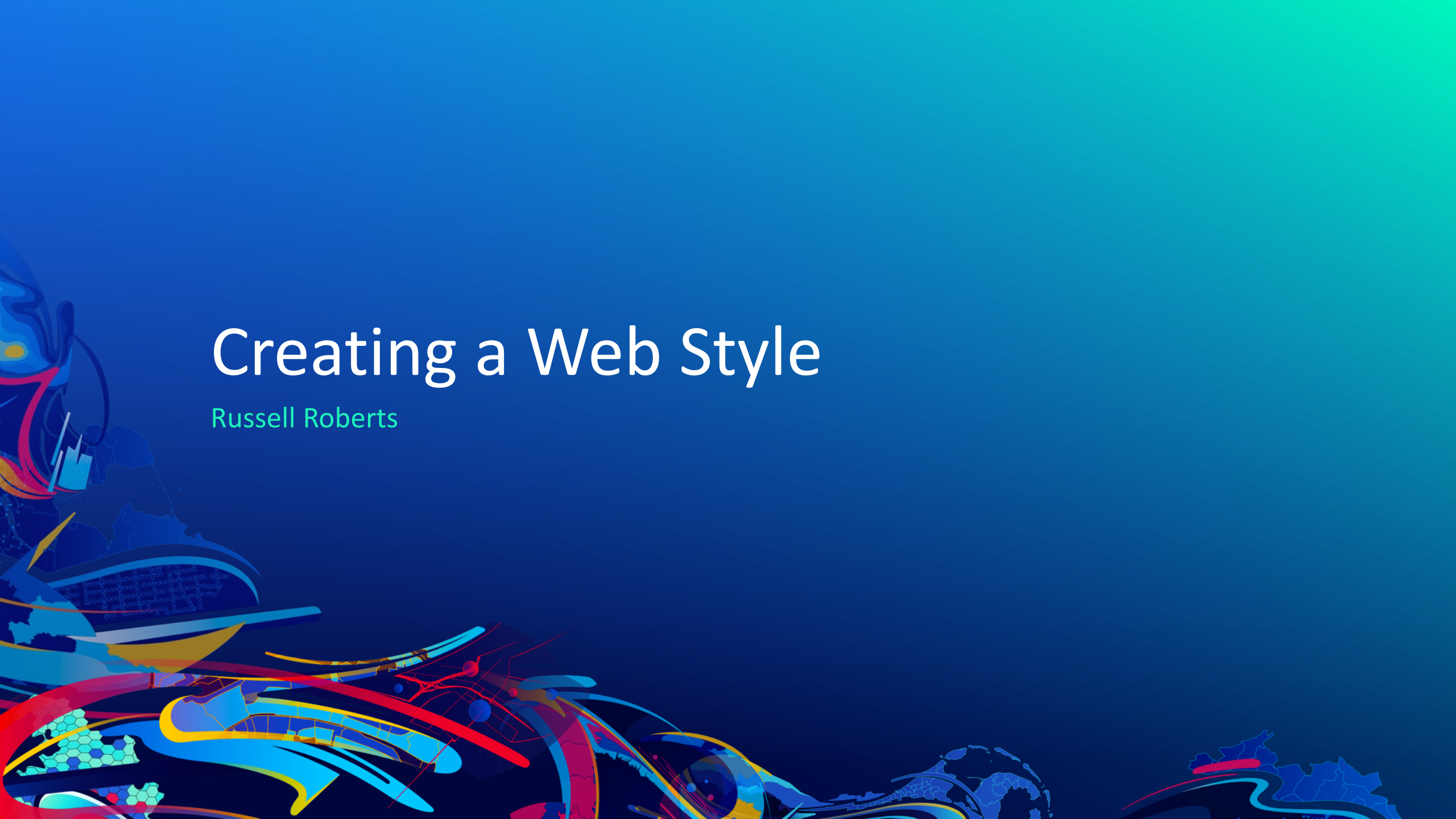
Anne Fitz

CIM Symbol Builder



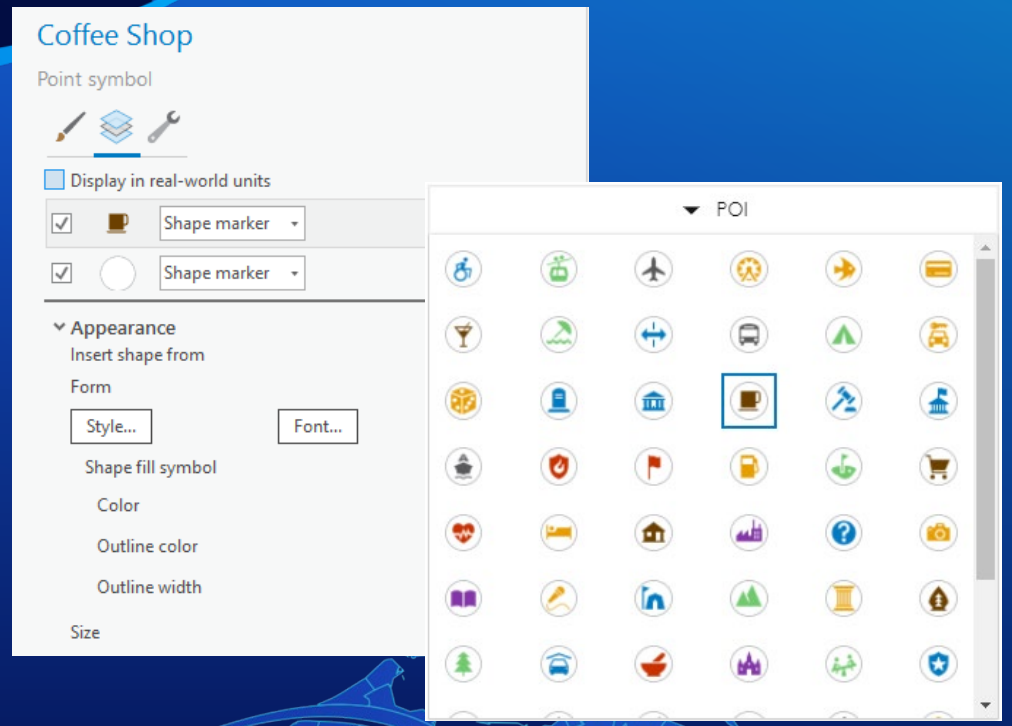
Creating a Web Style

Russell Roberts



How to create a Web Style

- Share web styles with ArcGIS Pro 2.6 and up
- Currently only points supported
- For 2D Web Styles publish picture or shape marker
- Vector symbols can be .svg or .emf format with shape marker
- Use them in web maps and 4.x web apps, runtime 100.x apps and Pro.
  - Map Viewer Beta, Experience Builder, Dashboard Beta and more



Creating a WebStyle

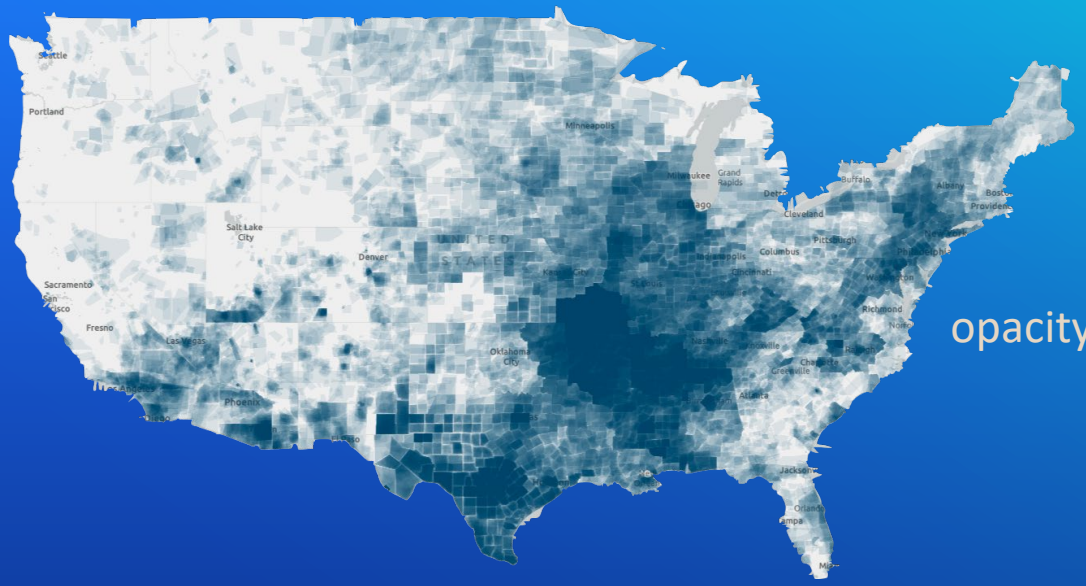
Russell Roberts

Data-driven CIM

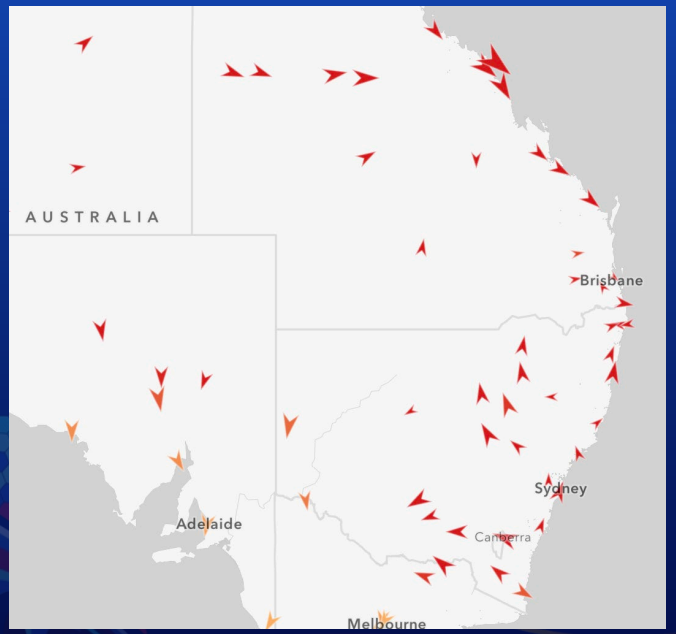
Kristian Ekenes

Data-driven mapping

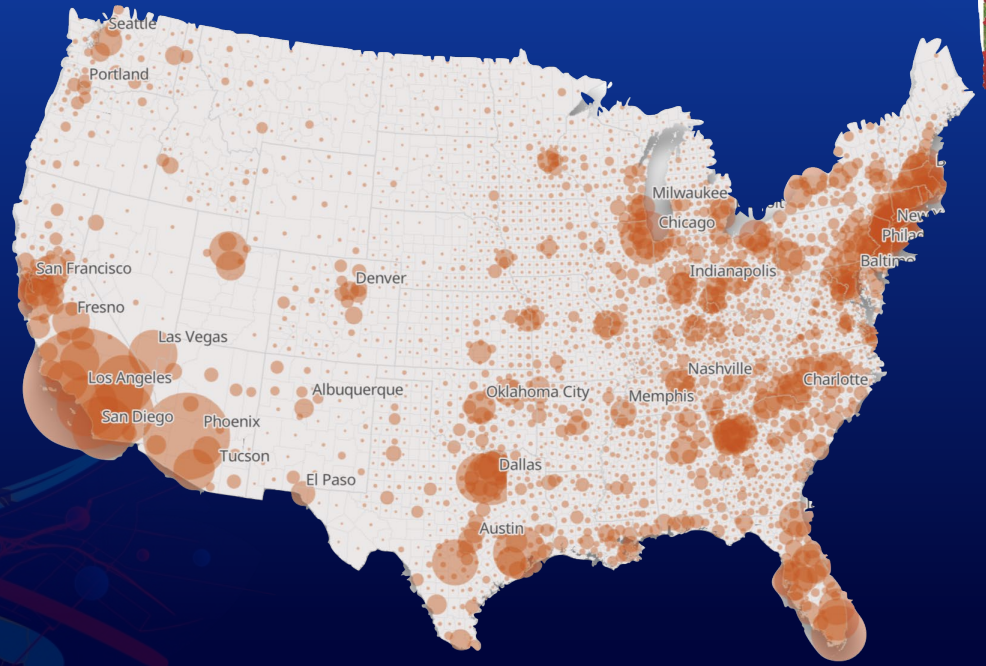
- Override a symbol's visual variable with a data value (either a field value, or an Arcade expression).

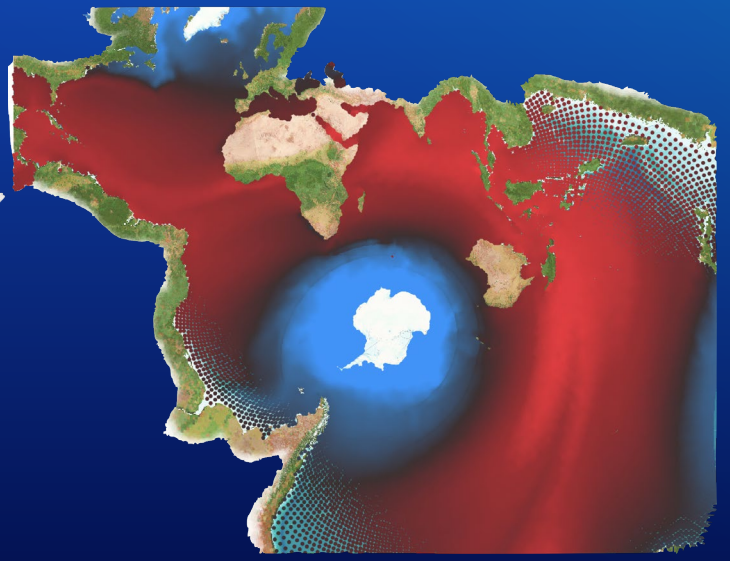


rotation



size

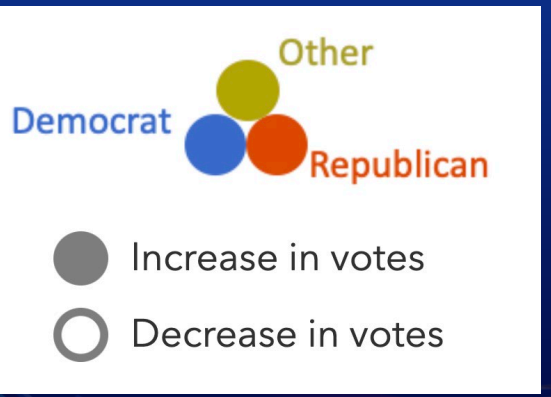


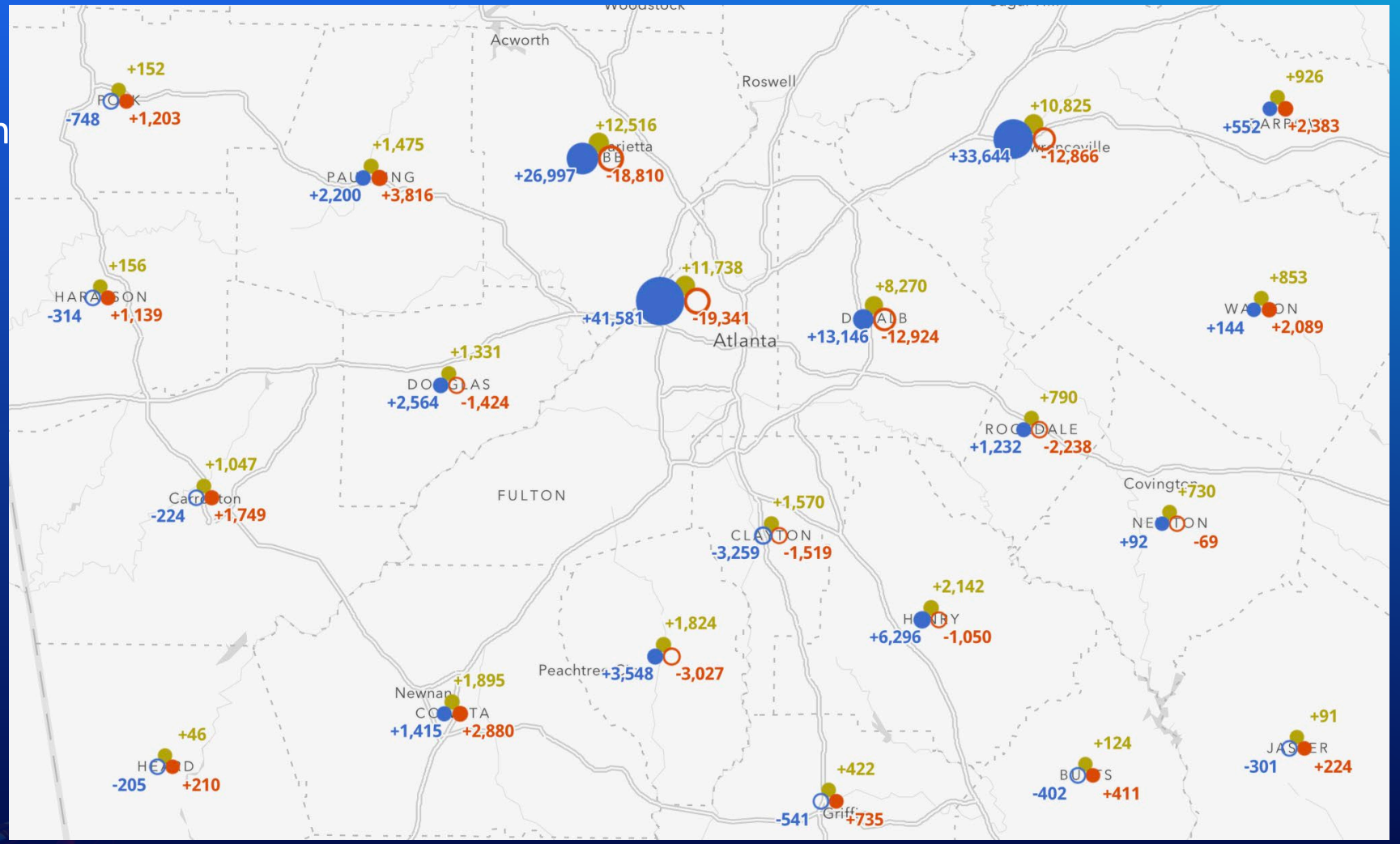


color

Multi-part data-driven symbols

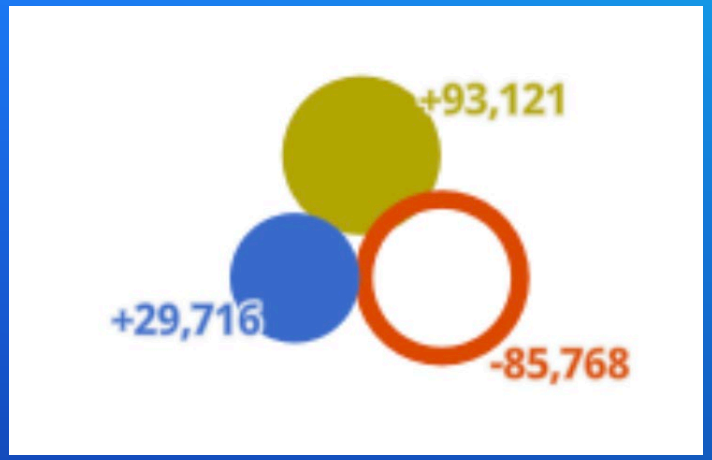
- Multiple symbol layers.
- The visual properties of each can be controlled by data values, so you can create more complex, multi-part, data-driven visualizations.
- More complex usually isn't a good thing, but it can be!





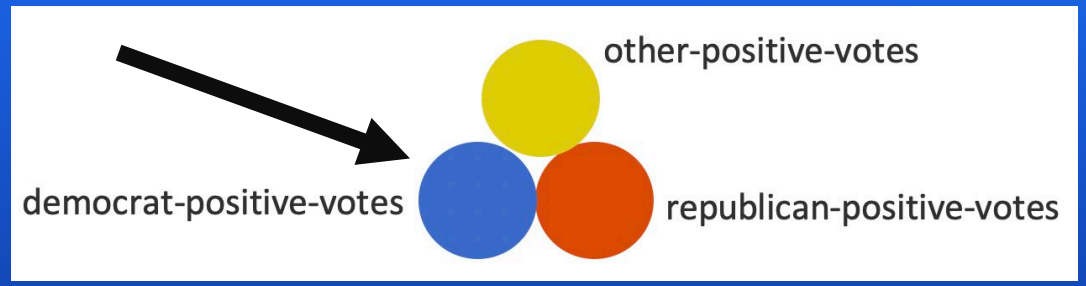
Primitive Overrides

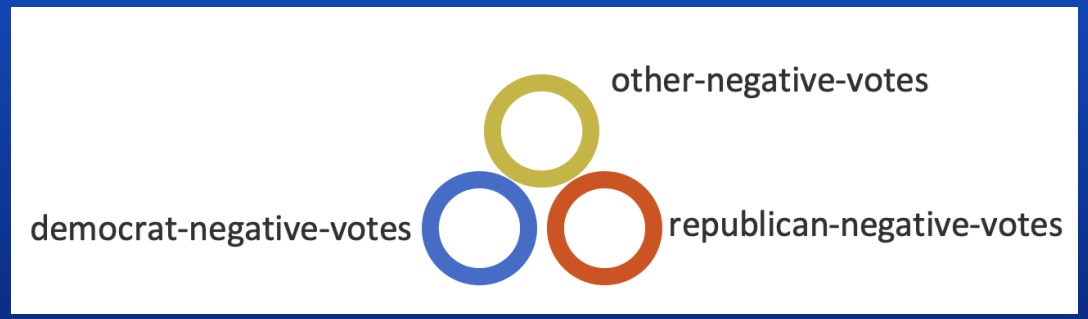
- A property of CIMSymbol.
- Arcade expressions that dynamically return the value of a primitive's visual property.
- e.g. a primitive override targeting size returns a new size value for a targeted symbol layer (primitive).
- Typically used to change a symbol property based on a data value or expression.




```
export const stateChangeRenderer = new SimpleRenderer({
  symbol: new CIMSymbol({
    data: {
      type: `CIMSymbolReference`,
      symbol: {
        type: `CIMPointSymbol`,
        symbolLayers: [⋯
        ]
      },
      primitiveOverrides: [⋯
      ]
    }
  })
});
```

Primitive Overrides






```
const symbolLayer = {
  type: "CIMVectorMarker",
  enable: true,
  primitiveName: "democrat-positive-votes",
  frame: { xmin: 0.0, ymin: 0.0, xmax: 17.0, ymax: 17.0 },
  anchorPoint: { x: 0.5, y: 0 },
  anchorPointUnits: "Relative",
```

```
  markerGraphics: [
    {
      type: "CIMMarkerGraphic",
      geometry: {
        rings: [
          [
            [8.5, 0.2],
            [7.06, 0.33],
            [5.66, 0.7],
            [4.35, 1.31],
            // additional vertices
            // to complete the circle
          ]
        ]
      },
      symbol: {
        type: "CIMPolygonSymbol",
        symbolLayers: [
          {
            type: "CIMSolidFill",
            enable: true,
            color: [60, 108, 204, 255]
          }
        ]
      }
    }
  ],
  scaleSymbolsProportionally: true,
  respectFrame: true
};
```

Primitive Overrides

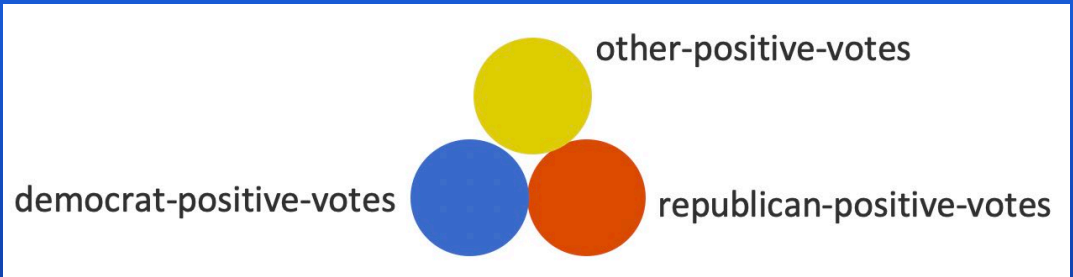
```
export const stateChangeRenderer = new SimpleRenderer({
  symbol: new CIMSymbol({
    data: {
      type: `CIMSymbolReference`,
      symbol: {
        type: `CIMPointSymbol`,
        symbolLayers: [...
        ],
      },
      primitiveOverrides: [...
      ]
    }
  })
});
```

```
createCircleSymbolLayer({
  primitiveName: `other-positive-votes`,
  offsetX: 0,
  offsetY: 10,
  color: oColorCIM,
  donutEnabled: false
}),
createCircleSymbolLayer({
  primitiveName: `other-negative-votes`,
  offsetX: 0,
  offsetY: 10,
  color: oColorCIM,
  donutEnabled: true
})
```

```
createCircleSymbolLayer({
  primitiveName: `democrat-positive-votes`,
  offsetX: -10,
  offsetY: 0,
  color: dColorCIM,
  donutEnabled: false
})
createCircleSymbolLayer({
  primitiveName: `democrat-negative-votes`,
  offsetX: -10,
  offsetY: 0,
  color: dColorCIM,
  donutEnabled: true
})
```

```
createCircleSymbolLayer({
  primitiveName: `republican-positive-votes`,
  offsetX: 10,
  offsetY: 0,
  color: rColorCIM,
  donutEnabled: false
}),
createCircleSymbolLayer({
  primitiveName: `republican-negative-votes`,
  offsetX: 10,
  offsetY: 0,
  color: rColorCIM,
  donutEnabled: true
}),
```

Primitive Overrides




```
var votesNext = $feature.PRS_DEM_16;
var votesPrevious = $feature.PRS_DEM_12;
var change = votesNext - votesPrevious;
var value = IIF( change > 0, change, 0);

// assign sizes to data values
// interpolate between stops
var sizeFactor = When(
  value >= 500000, 40,
  value >= 100000, 30 + ((40-30)/(500000-100000)) * (value - 100000)),
  value >= 50000, 5 + ((30-5)/(100000-50000)) * (value - 1)),
  value >= 10000, 1 + ((5-1)/(50000-10000)) * (value - 0.5)),
  value > 0, 0.5 + ((1-0.5)/(10000) * value),
  0
);
```

```
const democratPositiveOverride2 = {
  type: "CIMPrimitiveOverride",
  primitiveName: "democrat-positive-votes",
  propertyName: "Size",
  valueExpressionInfo: {
    type: "CIMExpressionInfo",
    title: "Increase in Democrat votes",
    expression: `
      var votesNext = $feature.PRS_DEM_16;
      var votesPrevious = $feature.PRS_DEM_12;
      var change = votesNext - votesPrevious;
      var value = IIF( change > 0, change, 0);

      // assign sizes to data values
      // interpolate between stops
      var sizeFactor = When(
        value >= 500000, 40,
        value >= 100000, 30 + ((40-30)/(500000-100000)) * (value - 100000)),
        value >= 50000, 5 + ((30-5)/(100000-50000)) * (value - 1)),
        value >= 10000, 1 + ((5-1)/(50000-10000)) * (value - 0.5)),
        value > 0, 0.5 + ((1-0.5)/(10000) * value),
        0
      );

      // adjust size based on scale
      // symbols grow as you zoom in
      // symbols shrink as you zoom out
      var scaleFactorBase = ( ${referenceScale} / $view.scale );
      var scaleFactor = When(
        scaleFactorBase >= 1, 1,  // 1
        scaleFactorBase >= 0.5, scaleFactorBase * 1,  // 0.6
        scaleFactorBase >= 0.25, scaleFactorBase * 1,  // 0.45
        scaleFactorBase >= 0.125, scaleFactorBase * 1,  // 0.3125
        scaleFactorBase * 1  // 0.1875
      );

      return sizeFactor * scaleFactor;
    `,
    returnType: "Default"
  }
}
```

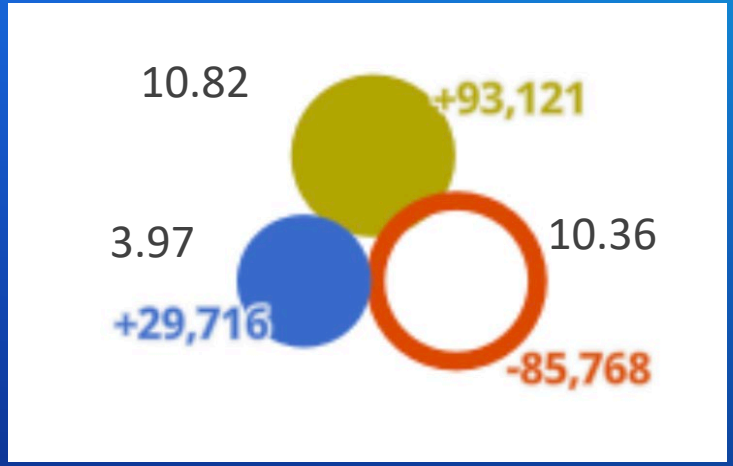
Primitive Overrides

```
var votesNext = $feature.PRS_DEM_16;
var votesPrevious = $feature.PRS_DEM_12;
var change = votesNext - votesPrevious;
var value = IIF( change > 0, change, 0);

// assign sizes to data values
// interpolate between stops
var sizeFactor = When(
  value >= 500000, 40,
  value >= 100000, 30 + ((40-30)/(500000-100000)) * (value - 100000)),
  value >= 50000, 5 + ((30-5)/(100000-50000)) * (value - 1)),
  value >= 10000, 1 + ((5-1)/(50000-10000)) * (value - 0.5)),
  value > 0, 0.5 + ((1-0.5)/(10000) * value),
  0
);
```

```
var change = votesNext - votesPrevious;
var value = IIF( change < 0, Abs(change), 0);
```

Computed sizes



All other symbol layers are assigned a size of zero, and therefore aren’t drawn.

Please provide your feedback for this session by clicking on the session survey link directly below the video.

Copyright © 2021 Esri. All rights reserved.