

Support story: Deterministic rendering order in SLDs

Posted on 04/10/14 by [Benjamin Trigona-Harany](#).

As with [our first support story](#), this one comes from MassGIS, the official state agency assigned to the collection, storage, and dissemination of geographic data. When a layer has overlapping lines or polygons, we often want to control which ones are rendered on top and which ones are rendered below. The classic example is when we have a roads layer that contains overpasses that we want to appear above regular roads at ground level. MassGIS was dealing with a land-use layer which required one type of polygon to be rendered with a solid color and other types to be rendered above with specific patterns.



The Aral Sea Example

We can simulate this problem by taking a layer that has both current and historical extents of bodies of water. In particular we'll look at the Aral Sea, which shrunk considerably in the twentieth century. We'll want to style the historical extents using hatching and then overlay the current lake in a plain blue style. What we don't want is to have the historical bounds drawn over the modern lake.

You can test on your own GeoServer install with the [data and SLDs in this ZIP](#).

SLD Rules

Our first attempt at creating an SLD to draw the two polygons in the correct order is to use two rule statements: the first will match our historical layer (where attribute `type` is equal to `historical`) and the second will match the modern layer (where the attribute `type` is equal to `modern`). We expect this to draw the historical lakes first and then the modern lakes on top:

```
<sld:StyledLayerDescriptor version="1.0.0" xmlns:sld="http://www.opengis.net/sld">
  <sld:NamedLayer>
    <sld:Name>Lakes</sld:Name>
    <UserStyle xmlns="http://www.opengis.net/sld">
      <FeatureTypeStyle>
        <Rule>
          <Name>Historical area</Name>
          ...
        </Rule>
        <Rule>
          <Name>Modern area</Name>
          ...
        </Rule>
      </FeatureTypeStyle>
    </UserStyle>
  </sld:NamedLayer>
</sld:StyledLayerDescriptor>
```

Unfortunately, this gives us the opposite effect: the historical area has been rendered *above* the modern area.



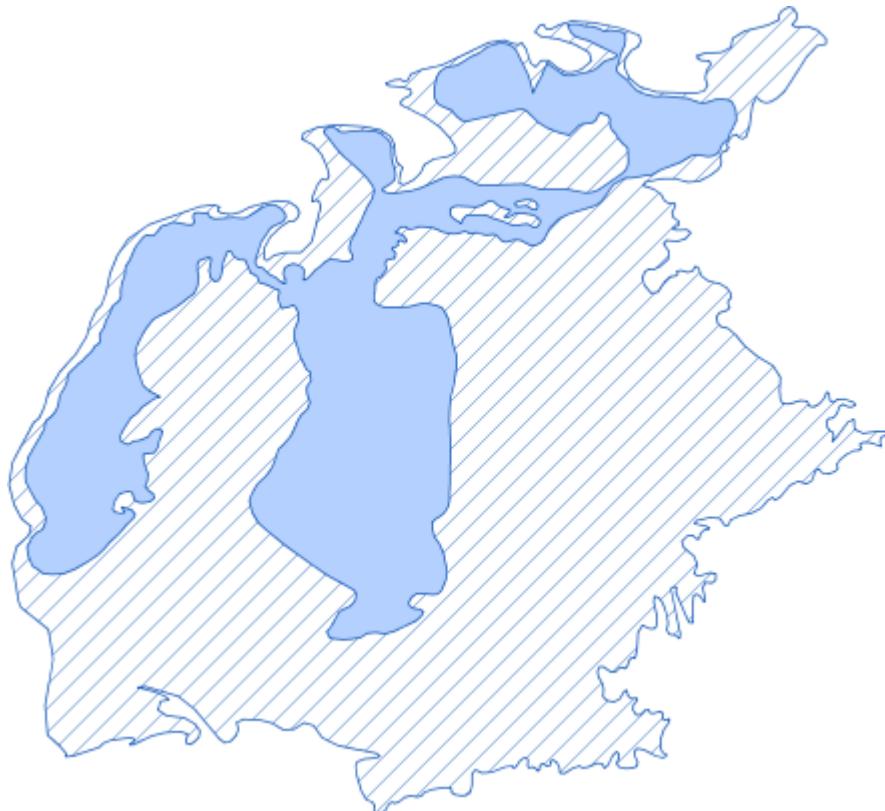
Since this ordering of the rules didn't work, we can try switching them around in the SLD, putting the rule for modern lakes first and historical lakes second. If you test this in GeoServer, however, you will see that the image that we get back is not affected.

Since reordering rules in an SLD does not affect the output image, we had to provide another approach.

SLD FeatureTypeStyle

The answer to the problem is to use the rules in separate `<FeatureTypeStyle>` sections of our SLD document:

```
<sld:StyledLayerDescriptor version="1.0.0" xmlns:sld="http://www.opengis.net/sld">
  <sld:NamedLayer>
    <sld:Name>Lakes</sld:Name>
    <UserStyle xmlns="http://www.opengis.net/sld">
      <FeatureTypeStyle>
        <Rule>
          <Name>Historical area</Name>
          ...
        </Rule>
      </FeatureTypeStyle>
      <FeatureTypeStyle>
        <Rule>
          <Name>Modern area</Name>
          ...
        </Rule>
      </FeatureTypeStyle>
    </UserStyle>
  </sld:NamedLayer>
</sld:StyledLayerDescriptor>
```



Why does this work?

Our first approach did not work since rules inside a single `FeatureTypeStyle` are applied to features as they arrive from the data source. GeoServer was drawing the modern Aral Sea first and the historical Aral Sea second simply because that is the order in which they were stored in the shapefile.

Even if the order had been correct, it would have been dangerous to rely on it for styling since that order could change at any time.

To reliably control the rendering order, therefore, we need the multiple `FeatureTypeStyle` blocks. GeoServer will draw each of these independently and then merge them in the defined order as the last step.

Benjamin Trigona-Harany leads our global support team from our offices in Victoria, BC. Interested in support or training for your enterprise? [Contact us](#) to learn more.