

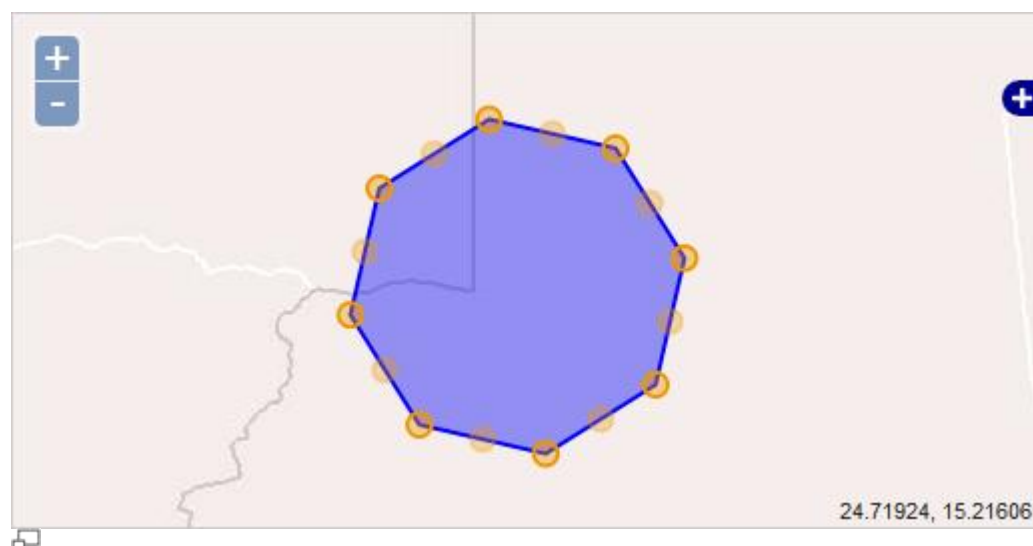
# Настройка стиля отображения вершин и виртуальных узлов редактируемого объекта в OpenLayers

[Обсудить в форуме](#) Комментариев — 0

Эта страница опубликована в основном списке статей сайта по адресу <http://gis-lab.info/qa/openlayers-modify-style.html>

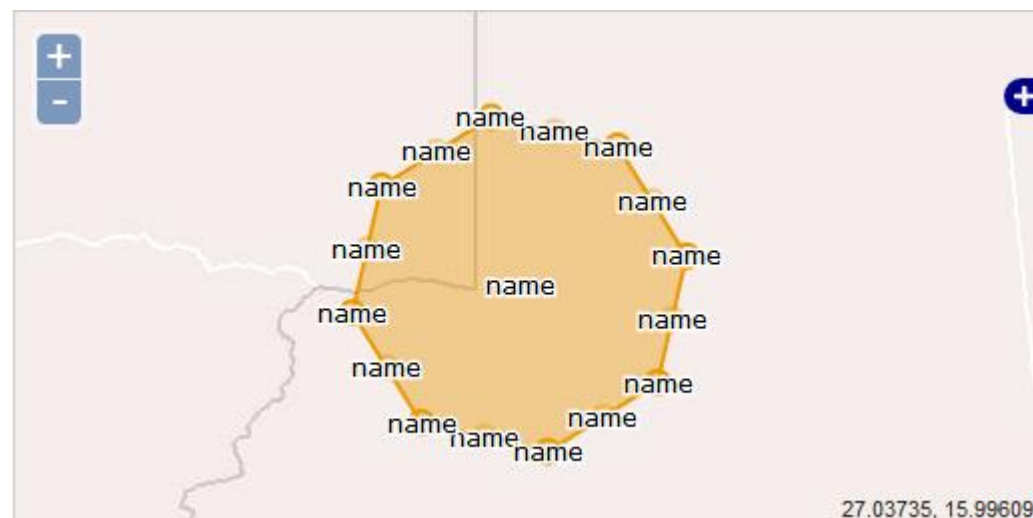
В статье рассмотрен вопрос решения проблемы, связанной с устранением дублирующих подписей объектов в режиме редактирования.

Контроль `OpenLayers.Control.ModifyFeature` предназначен для изменения геометрий векторных объектов. По умолчанию данный контроль работает в режиме **RESHAPE**, который позволяет изменять положение вершин редактируемого объекта. Выглядит это следующим образом:



Объект в режиме редактирования

Полупрозрачные круги - это так называемые виртуальные узлы, располагающиеся в центре рёбер и доступные для перетягивания. На первый взгляд всё хорошо, всё работает. А теперь давайте попробуем включить на редактирование какой-нибудь объект, в стиле отображения слоя которого используется подпись. В этом случае редактируемый объект приобретает вот такой вид и редактировать его становится практически невозможно:





## Дублирующиеся подписи объектов в режиме редактирования

Это происходит потому что и виртуальные узлы и сами вершины отрисовываются стилем самого слоя. Очевидным решением данной проблемы является определение отдельного стиля для отрисовки вершин и виртуальных узлов.

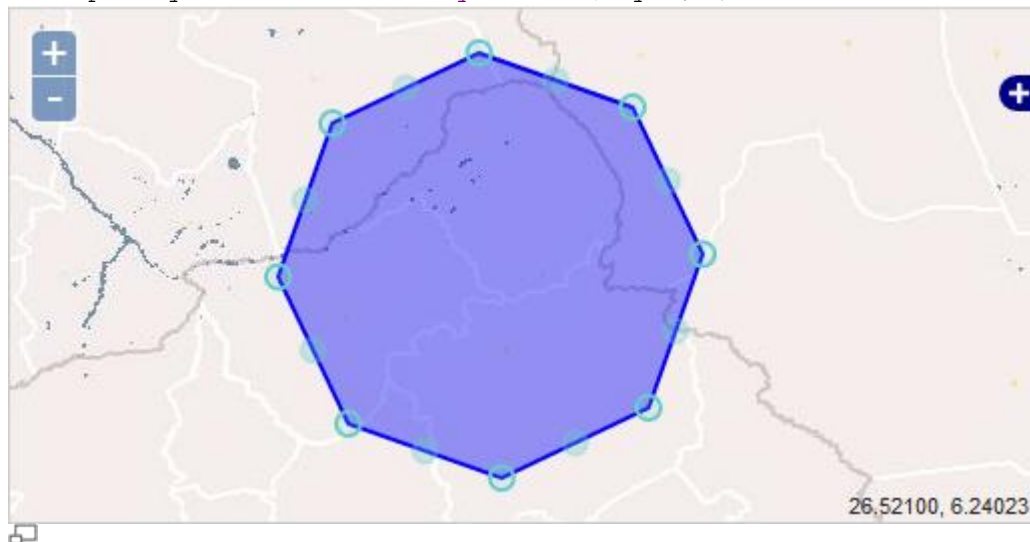
Посмотрим, что нам в этом направлении предлагает API OpenLayers. Согласно документации у `ModifyFeature` есть 2 свойства, позволяющих управлять внешним видом вершин и виртуальных узлов: [virtualStyle](#) и [vertexRenderIntent](#).

Фактически, `vertexRenderIntent` - это ключ, по которому доступно описание стиля (внутри карты стилей (`styleMap`) слоя), которым будут отрисовываться вершины. По умолчанию в OpenLayers векторный слой создаётся с картой стилей, содержащей описание 4-х предустановленных стилей:

```
layer.styleMap.styles["default"]
layer.styleMap.styles["delete"]
layer.styleMap.styles["select"]
layer.styleMap.styles["temporary"]
```

При создании слоя мы можем переопределить значение `styleMap`, указав только нужные стили и сконфигурировав их соответствующим образом (например, добавив вывод подписей). Предположим, наш слой содержит 4 вышеописанных предустановленных стиля, причём стиль "default" переопределён и содержит вывод подписей. Таким образом, чтобы при редактировании объекта его узлы не подписывались - их нужно отрисовывать любым из стилей, в описание которого отсутствует вывод подписей. Например, их можно отрисовать стилем "temporary", указав его при создании контрола:

```
new OpenLayers.Control.ModifyFeature(layer, {vertexRenderIntent: "temporary"});
```



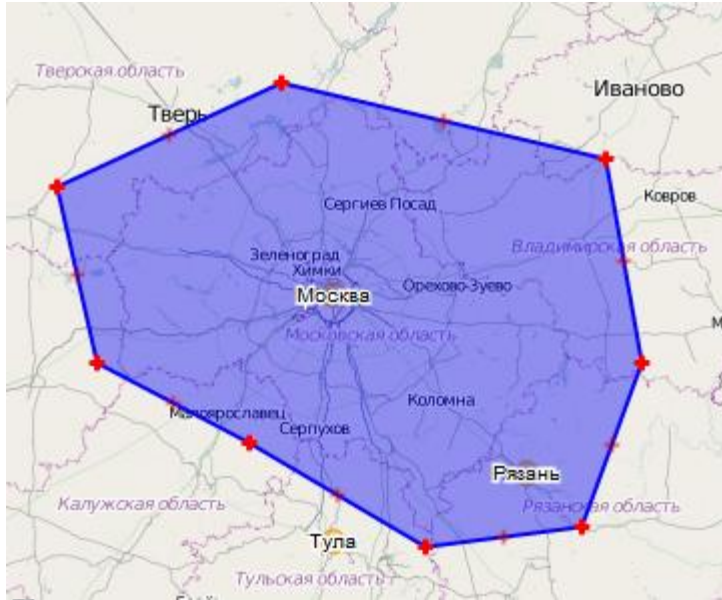
Объект в режиме редактирования, свойство `vertexRenderIntent` контрола `ModifyFeature` установлено в значение "temporary"

Если нас не устраивает стиль "temporary", мы можем использовать любой другой, указанный в `styleMap` слоя:

```
var vertexStyle = {
  strokeColor: "#ff0000",
  fillColor: "#ff0000",
  strokeOpacity: 1,
  strokeWidth: 2,
  pointRadius: 3,
  graphicName: "cross"
}

var styleMap = new OpenLayers.StyleMap({
  "default": OpenLayers.Feature.Vector.style['default'],
  "vertex": vertexStyle
}, {extendDefault: false});
```

```
var layer = new OpenLayers.Layer.Vector('features', {styleMap: styleMap});
map.addControl(new OpenLayers.Control.ModifyFeature(layer, {vertexRenderIntent:
"vertex"}));
```



Использование собственного vertexRenderIntent

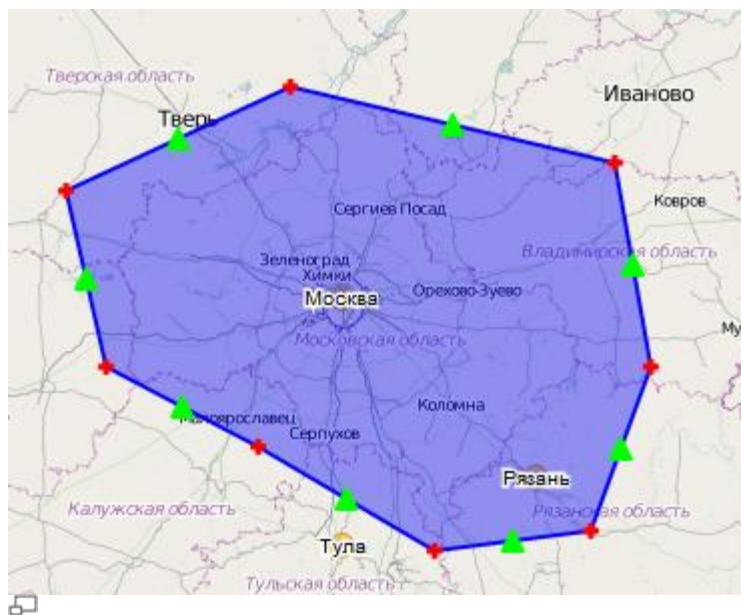
Опция [extendDefault](#) очень важна, если значение extendDefault равно true (по умолчанию), то во время рендеринга используемый стиль будет расширять стиль "default", если false - то использоваться как независимый стиль.

Как можно увидеть из рисунка виртуальные узлы отрисовываются в этом случае случае точно таким же стилем, что и вершины, только с другой прозрачностью заливки и обводки.

Если же мы хотим чтобы стиль отрисовки виртуальных узлов не базировался на стиле вершин, мы можем задать его отдельно:

```
var virtual = {
  strokeColor: "#00ff00",
  fillColor: "#00ff00",
  strokeOpacity: 1,
  strokeWidth: 2,
  pointRadius: 5,
  graphicName: "triangle"
};
```

```
map.addControl(new OpenLayers.Control.ModifyFeature(vectors, {vertexRenderIntent:
"vertex", virtualStyle: virtual}));
```



Совместное использование vertexRenderIntent и virtualStyle

Конечно, существуют и другие решения описанной проблемы, [например](#), однако предложенный выше подход видится более общим и поэтому правильным.

[Обсудить в форуме](#) Комментариев — 0

Последнее обновление: 2014-05-15 00:11

Дата создания: 29.06.2012

Автор(ы): [Денис Рыков](#)