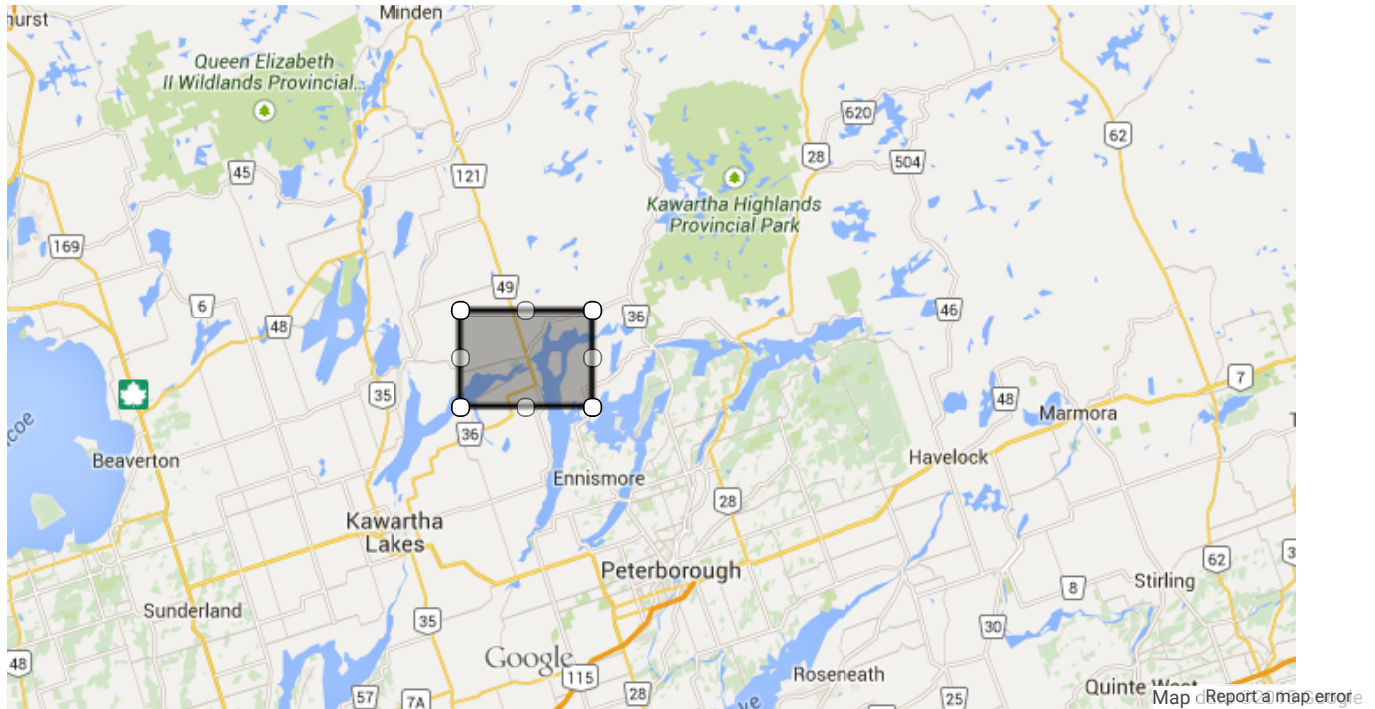


Rectangle Events



View this example [full screen](#).

JavaScript

JavaScript + HTML

```
// This example adds a user-editable rectangle to the map.  
// When the user changes the bounds of the rectangle,  
// an info window pops up displaying the new bounds.  
  
var rectangle;  
var map;  
var infoWindow;  
  
function initialize() {  
  var mapOptions = {  
    center: new google.maps.LatLng(44.5452, -78.5389),  
    zoom: 9  
  };  
  map = new google.maps.Map(document.getElementById('map-canvas'),  
    mapOptions);  
  
  var bounds = new google.maps.LatLngBounds(  
    new google.maps.LatLng(44.490, -78.649),  
    new google.maps.LatLng(44.599, -78.443)  
  );  
}
```

```
// Define the rectangle and set its editable property to true.
rectangle = new google.maps.Rectangle({
  bounds: bounds,
  editable: true,
  draggable: true
});

rectangle.setMap(map);

// Add an event listener on the rectangle.
google.maps.event.addListener(rectangle, 'bounds_changed', showNewRect);

// Define an info window on the map.
infoWindow = new google.maps.InfoWindow();
}
// Show the new coordinates for the rectangle in an info window.

/** @this {google.maps.Rectangle} */
function showNewRect(event) {
  var ne = rectangle.getBounds().getNorthEast();
  var sw = rectangle.getBounds().getSouthWest();

  var contentString = '<b>Rectangle moved.</b><br>' +
    'New north-east corner: ' + ne.lat() + ', ' + ne.lng() + '<br>' +
    'New south-west corner: ' + sw.lat() + ', ' + sw.lng();

  // Set the info window's content and position.
  infoWindow.setContent(contentString);
  infoWindow.setPosition(ne);

  infoWindow.open(map);
}

google.maps.event.addDomListener(window, 'load', initialize);
```

Except as otherwise noted, the content of this page is licensed under the [Creative Commons Attribution 3.0 License](#), and code samples are licensed under the [Apache 2.0 License](#). For details, see our [Site Policies](#).

Last updated March 17, 2015.