

## Adding Traffic Flow & Incidents to a Map

Traffic can be added to the map by displaying markets, flow and incidents.

### **To declare a new MQA.Traffic object**

```
myTraffic = new MQA.Traffic(myMap);
```

myMap is an MQA.Tilemap object

## Market Data

Traffic market data indicators are icons that will display on the map at zoom levels 2-6, showing that traffic is available for that area.

Valid JSON traffic market data elements - city, latitude, longitude, and state

### **MQA.Traffic.addMarkets();**

This function will add the traffic market data to the map.

Example: `myTraffic.addMarkets();`

### **MQA.Traffic.removeMarkets();**

This function will remove the traffic market data from the map.

Example: `myTraffic.removeMarkets();`

### **MQA.Traffic.setValue("marketTitleCallback",callback)**

This function sets the title of the rollover window that displays when the mouse is moved over the market data icon.

Example:

```
myTraffic.setValue("marketTitleCallback",MarketCallbackTitle);

function MarketCallbackTitle(data)
{
    //sets the rollover window to display the city & //state
    for the market
    return data.city + ', ' + data.state;
}
```

### **MQA.Traffic.setValue("marketContentCallback",callback)**

This function sets the content of the infowindow that displays when the market icon is clicked.

Example: `myTraffic.setValue("marketContentCallback",MarketCallbackContent);`

```
function MarketCallbackContent(data)
{
    //will set the contents of the market data indicator
    //infowindow with a link to zoom in to the street
    //level to view traffic
    var s;
```

```

        s = 'Click <a href="#" ' ;
        s += 'onclick=" myMap.setCenter(new MQA.LatLng( ' ;
        s += data.latitude + ',' + data.longitude;
        s += '), 7)">here</a> to zoom in.';
        return s;
    }

```

## **Flow Data**

Traffic flow is the image that is displayed on the map indicating whether or not the flow of a particular road is good (green), slow (yellow) or stopped (red).

### **MQA.Traffic.addFlow()**

This function will add the flow image to the map. This image will be displayed at zoom levels 6-16.

Example: `myTraffic.addFlow();`

### **MQA.Traffic.removeFlow()**

This function will remove the flow image from the map.

Example: `myTraffic.removeFlow();`

### **MQA.Traffic.setValue("flowOpacity",opacity)**

This function will set the opacity of the traffic flow image on the map.

opacity - float value from 0 to 1

Example: `myTraffic.setValue("flowOpacity",.7);`

### **MQA.Traffic.getValue("flowOpacity")**

This function will return the opacity of the traffic flow image.

Example: `var s = myTraffic.getValue("flowOpacity");`

## **Incident Data**

Incidents are displayed on the map for construction, traffic incidents, and events.

Valid JSON traffic incident data elements - description, endTime, incidentType, key, latitude, longitude, severity, startTime, and title

### **MQA.Traffic.addIncidents()**

This function will add all available incidents to the map.

Example: `myTraffic.addIncidents();`

### **MQA.Traffic.removeIncidents()**

This function will remove all incidents from the map.

Example: `myTraffic.removeIncidents();`

### **MQA.Traffic.setValue("incidentTypeFilter",incidentTypes)**

incidentTypes - array of incident types. Valid types are:  
MQA.TRAFFIC\_INCIDENTS, MQA.TRAFFIC\_CONSTRUCTION,  
MQA.TRAFFIC\_EVENTS

Example: 

```
var myIncidents = new Array();
myIncidents.push(MQA.TRAFFIC_INCIDENTS);
myTraffic.setValue("incidentTypeFilter",myIncidents);
```

### **MQA.Traffic.setValue("incidentTitleCallback",callback)**

This function sets the title of the rollover window that displays when the mouse is moved over the traffic incident icon.

Example:

```
myTraffic.setValue("incidentTitleCallback",IncidentCallbackTitle);

function IncidentCallbackTitle(data)
{
    //sets the rollover window to display the incident type &
    //title
    return data.incidentType + ', ' + data.title;
}
```

### **MQA.Traffic.setValue("incidentContentCallback",callback)**

This function sets the contents of the infowindow that is displayed when a traffic icon is clicked.

Example:

```
myTraffic.setValue("incidentContentCallback",IncidentCallbackContent);

function IncidentCallbackContent(data)
{
    //displays the description of the incident
    var s;
    s = data.description;
    return s;
}
```

## **Other Functions**

### **MQA.Traffic.isAvailable()**

This function will return true/false, indicating if traffic is available.

Example: 

```
myTraffic.isAvailable();
```

### **MQA.Traffic.setTimeoutDuration(m integer)**

This function sets the timeout duration in milliseconds before failing the request. Default value is 10 seconds.

m - timeout in milliseconds

Example: `myTraffic.setTimeoutDuration(200);`

**MQA.Traffic.getTimeoutDuration()**

This function will return the timeout duration in milliseconds. Default value is 10 seconds.

Example: `var timeout = myTraffic.getTimeoutDuration();`

**MQA.Traffic.setMaxInfoWindowWidth(w integer)**

This function sets the width of the infowindow that displays for a traffic incident. Default value is 285 pixels.

w - maximum width in pixels

Example: `myTraffic.setMaxInfoWindowWidth(350);`

**MQA.Traffic.refresh()**

This function makes a new call to the traffic service thus 'updating' the map with the latest information.

Example: `myTraffic.refresh();`

## **Events**

Traffic events are handled the same as other events within the TileMap Toolkit. Utilize the `MQA.EventManager` to add/remove events.

flowadded  
flowremoved  
marketsadded  
marketsremoved  
marketttimeout  
incidentsadded  
incidentsremoved  
incidenttimeout

The following are the event names that will be returned:

MQA.Traffic.flowadded  
MQA.Traffic.flowremoved  
MQA.Traffic.marketsadded  
MQA.Traffic.marketsremoved  
MQA.Traffic.marketttimeout  
MQA.Traffic.incidentsadded  
MQA.Traffic.incidentsremoved  
MQA.Traffic.incidenttimeout