



# **MapQuest Platform Flash CS3 Developer Guide**

**Version 5.3.0.1**

*July 16, 2008*

## Copyrights, Trademarks, and Legal Information

Copyright 2008 MapQuest, Inc. All rights reserved. The use of this documentation and the product with which it is provided are governed by the product license agreement and applicable intellectual property laws. No part of the product or this documentation may be reproduced in any form by any means without prior written authorization from MapQuest, Inc. MapQuest may revise this documentation from time to time without notice.

THIS DOCUMENTATION IS PROVIDED “AS IS” WITHOUT WARRANTY OF ANY KIND. IN NO EVENT SHALL MAPQUEST, ITS PARENT OR ITS LICENSORS BE LIABLE FOR INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY KIND ARISING FROM ANY ERROR IN THIS DOCUMENTATION, INCLUDING WITHOUT LIMITATION ANY LOSS OR INTERRUPTION OF BUSINESS, PROFITS, USE, OR DATA.

The downloading, exporting, or reexporting of MapQuest products or any underlying information or technology must be in full compliance with all United States and other applicable laws and regulations. Any provision of a MapQuest product or documentation to the U.S. government is with restricted rights as described in the license agreement for that product.

MapQuest, MapQuest.com, the MapQuest logo, Advantage, Site Advantage, Advantage API, and Advantage Enterprise are trademarks of MapQuest, Inc. or its parent AOL LLC. Other product and brand names are trademarks of their respective owners.

Companies, names, and data used in examples are fictitious unless otherwise noted. MapQuest may have patents or pending patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. The furnishing of this document does not give you license to these patents, trademarks, copyrights, or other intellectual property except as expressly provided in any written license agreement from MapQuest. This document and the software described in it are copyrighted, with all rights reserved.

# CONTENTS

---

1	Introduction .....	1
1.1	Assumptions .....	1
1.2	Additional References .....	1
2	MapQuest Platform API 5.3.0 Libraries .....	3
	How to Obtain the MapQuest Platform Flash CS3 Libraries .....	3
	How to Install the MapQuest Platform Flash CS3 Libraries .....	3
3	Mapping .....	5
	How to Generate a Simple Map .....	5
	How to Add Controls to a Map .....	5
	How to Center a Map .....	6
	How to Add Points of Interest (POIs) to a Map .....	8
	How to Use Custom POI Icons .....	10
4	Geocoding, Routing, and Searching .....	13
4.1	Geocoding .....	13
	How to Retrieve Geocode Data .....	13
	How to Add a POI to the Geocoded Address .....	16
	How to Calculate the Route .....	19
	How to Display the Route .....	23
	How to Display Search Results on a Map .....	27
5	Using Remote Collections .....	33
	How to Add and Remove RemoteCollections to and from a TileMap .....	33
5.1	Updating RemoteCollections .....	34
5.2	Using RemoteCollection Events .....	34
6	Using Multiple Collections .....	35
	How to Add or Remove Multiple ShapeCollections To or From a TileMap .....	35
	How to Update ShapeCollections .....	36
	How to Use ShapeCollection Decluttering .....	36
6.1	ShapeCollection Properties .....	36
6.2	Using TileMap ShapeCollection Methods .....	37
6.3	Default Shape Collection Methods .....	38
7	Using Traffic .....	39
7.1	Adding Traffic to Your Maps .....	39
	How to Create the Traffic Object .....	39
	How to Access the Traffic Flow Image Overlay .....	40
	How to Access the Traffic Market POIs .....	41
	How to Access the Traffic Incident POIs .....	42

	How to Use the Assorted Traffic Functionality .....	43
7.2	Traffic Events .....	44
7.3	Traffic Control Button .....	44

# 1

## Introduction

Welcome to MapQuest Platform Edition 5.3.0 for Flash CS3! The goal of this document is to familiarize developers with the basics of the Flash API and TileMap Toolkit.

### 1.1 Assumptions

This document assumes the developer has a basic knowledge of Flash CS3, has a copy of Flash CS3, and has an understanding of ActionScript 3. A trial version of Flash CS3 is available from Adobe's Web site.

### 1.2 Additional References

Other available resources include:

- [Flash API Reference](#)
- [XML reference Guide](#)
- [Flash API code samples contained in this guide](#)
- [MapQuest Platform API Forums](#)



## MapQuest Platform API 5.3.0 Libraries



### How to Obtain the MapQuest Platform Flash CS3 Libraries

- 1) Go to [http://developer.mapquest.com/content/downloads/53/MQFLAPI\\_5\\_3\\_0.zip](http://developer.mapquest.com/content/downloads/53/MQFLAPI_5_3_0.zip).
- 2) Extract the libraries.

Take note of where the libraries are extracted.



### How to Install the MapQuest Platform Flash CS3 Libraries

The Flash CS3 libraries are packaged into an **.mxp** file for installation using the Adobe Extension manager. The Adobe Extension manager can be downloaded for free from Adobe's web site:

<http://www.adobe.com>

- 1) Double-click the **.mxp** file containing the libraries and the Adobe Extension Manager will install the components.
- 2) Within the Adobe Flash CS3 IDE, use the **Window | Components** menu options to enable the **Components** panel.



## Mapping



### How to Generate a Simple Map

#### Note

---

In an effort to keep the sample simple, no scaling functionality has been applied to this sample.

---

See page [HERE](#).

- 1) Open the Flash CS3 IDE.
- 2) From the menu bar, select **File | New | Flash File (ActionScript 3.0)**.
- 3) From the **Components** toolbox, click and drag the **Tilemap** component to the stage.
- 4) Click the **Parameters** tab relating to the **Tilemap** component, and enter the values shown below.
  - **key:** *Enter a valid MapQuest Platform API key*
  - **X:** 0
  - **Y:** 0
- 5) From the menu bar, select **File | Save** and save the file as **SimpleMap**.
- 6) From the menu bar, select **File | Export | Export Movie...** and save the file as **SimpleMap**.
- 7) Run the application by browsing to the exported movie and double-clicking the **SimpleMap.swf** file.



### How to Add Controls to a Map

This sample details how to add controls to a map to allow the map to be manipulated by users.

#### Note

---

In an effort to keep the sample simple, no scaling functionality has been applied to this sample.

---

There are four different map controls available in the MapQuest Flash CS3 API:

- **ZoomControl:** a control consisting of a "+" and "-" symbol used to increase or decrease the current zoom level of a map.
- **PanControl:** a control consisting of four directional arrows (N, S, E, W) used to pan a map in the specified direction.
- **ViewControl:** a control consisting of three buttons (Street Map, Aerial Image, Hybrid Map) used to set the map type.



## How to Add Controls to a Map (cont'd.)

- **LargeZoomControl**: a control consisting of a combination of a **ZoomControl**, **PanControl** and a **ZoombarControl**.

The Flash CS3 IDE should be opened displaying the **SimpleMap.fla** file to perform the steps below. (See page [HERE](#).)

- 1) Save the **SimpleMap.fla** file as **ControlsMap.fla**.
- 2) Click the **Parameters** tab relating to the **Tilemap** component, and enter the values shown below.  
**InstanceName:** myMap
- 3) From the **Components** toolbox, click and drag the **LargeZoomControl** component to the stage.
- 4) Click the **Parameters** tab relating to the **LargeZoomControl** component, and enter the values shown below.
  - **map:** myMap
  - **X:** 5.3
  - **Y:** 62.4
- 5) From the **Components** toolbox, click and drag the **ViewControl** component to the stage.
- 6) Click the **Parameters** tab relating to the **ViewControl** component, and enter the values shown below.
  - **map:** myMap
  - **X:** 204.3
  - **Y:** 2.4
- 7) From the menu bar, select **File | Save**", and save the file as **ControlsMap**.
- 8) From the menu bar, select **File | Export | Export Movie...** and save the file as **ControlsMap**.
- 9) Run the application by browsing to the exported movie and double-clicking the **ControlsMap.swf** file.



## How to Center a Map

This sample details how to center a map programmatically.

The Flash CS3 IDE should be opened displaying the **ControlsMap.fla** file to perform the steps below. (See page [HERE](#).)

- 1) Save the **ControlsMap.fla** file as **CenterMap.fla**.
- 2) Click the **Parameters** tab relating to the **Tilemap** component, and enter the value shown below.  
**zoom:** 9
- 3) From the **Components** toolbox, click and drag a button to the bottom center stage.
- 4) Click the **Parameters** tab relating to the **Button** component, and enter the values shown below.
  - **InstanceName:** myButton
  - **Label:** Set Center
- 5) From the **File** menu select **New** and then **ActionScript File** from the resulting dialog box. Click **OK**.
- 6) Enter code below to import the required resources, assign and create the button event listener and handle resizing.



## How to Center a Map (cont'd.)

```
package {

    import flash.display.Sprite;
    import flash.display.Stage;
    import flash.display.StageAlign;
    import flash.display.StageScaleMode;
    import flash.events.Event;
    import flash.events.MouseEvent;
    import com.mapquest.*;
    import com.mapquest.tilemap.*;

    public class CenterMap extends Sprite {

        /**
         * Constructor
         */
        public function CenterMap():void {

            // Set Stage Properties
            this.stage.scaleMode = StageScaleMode.NO_SCALE;
            this.stage.align=StageAlign.TOP_LEFT;

            // Handle Resize event
            this.stage.addEventListener(Event.RESIZE,
                this.onStageResize);

            //create an even listener for the button click mouse event
            myButton.addEventListener(MouseEvent.CLICK, onButtonClick);

            // Set Initial size
            onStageResize(null);

        }

        /**
         * Handle Button Click
         */
        private function onButtonClick(event:MouseEvent):void {

            var newCenter:PointLL = new PointLL(40.0446, -76.4131);
            myMap.setCenter(newCenter);

        }

        /**
         * Handle Map Sizing
         */
        private function onStageResize(evt:Event):void {

            myMap.setSize(new Size(this.stage.stageWidth, this.stage.stage
                Height - 50));

        }

    }

}
```



## How to Center a Map (cont'd.)

```
myButton.x = (this.stage.stageWidth / 2) - (myButton.width/2);  
myButton.y = myMap.height + 10;  
  
}  
  
} // End Class  
  
} // End Package
```

- 7) Save the file containing the code as **CenterMap.as** in the same folder as the **.fla** file.
- 8) Select the tab containing the **CenterMap.fla** file.
- 9) Select **File | Publish Settings**.
- 10) Select the **Flash** tab.
- 11) Select **Settings...**
- 12) Enter **CenterMap** in the **Document Class** field and click **OK**.
- 13) From the menu bar, select **File | Export | Export Movie...**
- 14) Run the application by browsing to the exported movie and double-clicking the **CenterMap.swf** file.
- 15) Select **Set Center** to see the map re-center on the **LatLng** data specified in the code.



## How to Add Points of Interest (POIs) to a Map

This sample will build on the previous sample. In addition to re-centering the map when the button is clicked, a POI for the city of New York will be added to the map.

The Flash CS3 IDE should be opened displaying the **CenterMap.fla** file to perform the steps below.

### Note

---

A variety of options are available for adding POIs to your map, including setting the zoom levels.

### Zoom Levels

The **Poi** object now has functions to handle at which zoom level the items will appear on the map, including the following:

- `Poi.setMinZoomLevel (zoomlevel);`  
Sets the minimum zoom level at which the POI will be displayed.

#### Example:

```
myPoi.setMinZoomLevel (6);
```

- `Poi.setMaxZoomLevel (zoomlevel);`  
Sets the maximum zoom level at which the POI will be displayed.

#### Example:

```
myPoi.setMaxZoomLevel (9);
```

Valid zoom levels can be set between 1 and 16.

---



## How to Add Points of Interest (POIs) to a Map (cont'd.)

Page can be seen [HERE](#).

- 1) Save the **CenterMap.fla** file as **AddPOI.fla**.
- 2) Change the label on the button from **Set Center** to **New York**.
- 3) Save the file containing the class code as **AddPOI.as**.
- 4) Select the tab containing the **AddPOI.fla** file.
- 5) From the file menu select **Publish Settings**, click on the **Flash** tab and then the **Settings...** button
- 6) Enter **AddPOI** in the **Document Class** field and click **OK**.
- 7) Add the import directive for the **Poi** class to the class code in the section with the other import statements.
- 8) The code in the **onButtonClick** function should be replaced with the code below. This code will do the following:
  - Create **PointLL** object to be used as the POI location and to center the map.
  - Create display text to be associated with the POI.
    - **infoTitle** displays on mouse rollover
    - **infoContent** displays when clicking on the POI
    - Other methods are available, such as **setLabel**, **setLabelVisible**, and **setRolloverEnabled**
  - Assign a key to the POI.
  - Add the POI to the map using the **addPoi()** function.

The ActionScript code below represents the changes to the previous example.

```
//import the required classes
import com.mapquest.tilemap.pois.*;
/**
 * Handle Button Click
 */
private function onButtonClick(event:MouseEvent):void {

    // Create lat/lng point for NYC
    var nycLL:PointLL = new PointLL(40.720409,-73.994637);

    // Create Poi
    var myPoi:Poi = new Poi(nycLL);
    myPoi.setInfoTitle("New York, New York");
    myPoi.setInfoContent("The city that never sleeps");
    myPoi.setKey("A1");

    // Add Poi to Map.
    myMap.addPoi(myPoi);
    myMap.setCenter(nycLL);
}
```

- 9) In the **AddPOI.as** file, rename the class definition and the constructor from **CenterMap** to **AddPOI**.
- 10) Select the **AddPOI.fla** tab.
- 11) From the menu bar, select **File | Export | Export Movie...**



## How to Add Points of Interest (POIs) to a Map (cont'd.)

- 12) Run the application by browsing to the exported movie and double-clicking the **AddPOI.swf** file.
- 13) Click **New York** button to see the map re-center on the "Big Apple".



## How to Use Custom POI Icons

This sample will build on the previous sample. The POI created for New York will utilize a custom icon rather than the default icon. (Page can be seen [HERE](#).)

- 1) Save the **AddPOI.fla** file as **CustomIcon.fla**.
- 2) Change the label on the button from **New York** to **Custom Icon**.
- 3) Save the file containing the class code as **CustomIcon.as**.
- 4) Click the **CustomIcon.fla** tab.
- 5) Select **File | Publish Settings**. Select the **Flash** tab and then the **Settings...** button.
- 6) Enter **CustomIcon** in the document class field and click **OK**.
- 7) Create a private variable, inside the class definition, to hold the NYC POI.
- 8) The code below replaces the constructor and the `onButtonClick` code from the previous example, and does the following:
  - In the constructor, a POI for NYC is created and the map is centered on NYC. This POI will have a default icon
  - In the `onButtonClick` event handler, a new `ImageMapIcon` is created and it is pointed to a URL for the image
  - The POI for NYC is then associated with the new icon

```
private var nycPoi:Poi;

/**
 * Constructor
 */
public function CustomIcon():void {

    // Set Stage Properties
    this.stage.scaleMode = StageScaleMode.NO_SCALE;
    this.stage.align=StageAlign.TOP_LEFT;

    // Handle Map Sizing
    this.stage.addEventListener(Event.RESIZE, this.onStageResize);

    // Create lat/lng point for NYC
    var nycLL:PointLL = new PointLL(40.720409,-73.994637);

    // Create Poi
    nycPoi= new Poi(nycLL);
    nycPoi.setInfoTitle("New York, New York");
    nycPoi.setInfoContent("The city that never sleeps,
        That's why it looks that way in the morning");
    nycPoi.setKey("A1");
    myMap.addPoi(nycPoi);
```



## How to Use Custom POI Icons (cont'd.)

```

myMap.setCenter(nycLL);

// Handle Resize event
this.stage.addEventListener(Event.RESIZE, this.onStageResize);

// Handle Button Click
myButton.addEventListener(MouseEvent.CLICK, onButtonClick);

// Set Initial size
onStageResize(null);
}

/**
 * Handle Button Click
 */
private function onButtonClick(event:MouseEvent):void {

    var nycIcon:ImageMapIcon = new ImageMapIcon();
    nycIcon.setImageURL("http://www.mqdemo.com/dww_demo/apple.jpg",
        32, 32);
    nycPoi.setIcon(nycIcon);
}

```

### Note

Instead of setting a specific icon, you can now pass in an image URL. For example:

```

// Embed the Icon in the Class
[Embed (source="http://www.yourserver.com/images/
    BigApple.gif")]
var NyIcon:Class;

// Create new icon object
var _myIcon:MapIcon = new MapIcon();
_myIcon.setImage(new NyIcon(), 32, 32);

// Associate icon with POI
_myPoi.setIcon(_myIcon);

// Add the Poi to the map.
_map.addPoi(_myPoi);

```

- 9) Select **File | Export | Export Movie...**
- 10) Run the application by browsing to the exported movie and double-clicking the **CustomIcon.swf** file.
- 11) Click **Custom Icon** to see the icon change from the default icon to the image you selected.



## Geocoding, Routing, and Searching

### 4.1 Geocoding

This sample utilizes the MapQuest Geocode server to retrieve geocode data for an address.



#### How to Retrieve Geocode Data

Page can be seen [HERE](#).

- 1) Open the Flash CS3 IDE.
- 2) Select **File | New | Flash File (ActionScript 3.0)**.
- 3) From the **Components** toolbox, click and drag the **Advantage API** component to the stage.
- 4) From the **Components** toolbox, click and drag a **label** to the stage, and set its parameters as shown below.  
**Text:** City:
- 5) From the **Components** toolbox, click and drag a **TextInput** to the stage, and set its parameters as shown below.  
**Instance Name:** txtCity
- 6) Repeat [Step 4](#)) and [Step 5](#)) for state, zip and country.
- 7) From the **Components** toolbox, click and drag a Button to the stage.
- 8) Select the **Parameters** tab relating to the Button component, and enter the values shown below.
  - **InstanceName:** myButton
  - **Label:** Geocode Address
- 9) From the **Components** toolbox, click and drag a **TextArea** to the stage, and enter the values shown below.  
**Instance Name:** myTextArea
- 10) Save the newly created flash file as **Geocode . fla**.
- 11) Select **File | New** and then **ActionScript File** from the resulting dialog box.
- 12) Click **OK**.
- 13) Enter the code below into the opened window.



## How to Retrieve Geocode Data (cont'd.)

```
package {

    //import the required classes
    import com.mapquest.*;
    import flash.display.Sprite;
    import flash.events.MouseEvent;

    public class Geocode extends Sprite {

        //create and set required variable to execute the geocode process
        private var serverName:String = "geocode.access.mapquest.com";
        private var serverPath:String = "mq";
        private var serverPort:int = 80;
        private var clientId:String = "[Your Client ID]";
        private var password:String = "[Your Password]";
        /**
         * Constructor
         */
        public function Geocode():void {

            // Add listener for button click
            myButton.addEventListener(MouseEvent.CLICK, onButtonClick);

        }

        /**
         * Handle Button Click
         */
        private function onButtonClick(event:MouseEvent):void {

            var address:Address = new Address();
            address.setCity(txtCity.text);
            address.setState(txtState.text);
            address.setPostalCode(txtZip.text);
            address.setCountry(txtCountry.text);

            // Setup server object for call.
            var exec:Exec = new Exec(serverName, serverPath, serverPort);
            exec.setClientId(clientId);
            exec.setPassword(password);

            // Add a listener to be called when geocode completes
            exec.addEventListener(Exec.TRANS_TYPE_GEOCODE, onGeocode);

            // Add a listener to the exec object.
            // To handle the Geocode Error.
            exec.addEventListener(Exec.EVENT_DO_TRANSACTION_ERROR,
                onExecError);
            //call the geocode function of the exec object
            // and give it the address object to be geocoded
            exec.geocode(address);
        }
    }
}
```



## How to Retrieve Geocode Data (cont'd.)

```

    }

    /**
     * Handle successful call to geocode service
     */
    private function onGeocode(evt:ExecResultEvent):void {

        //retrive the location collection from the event object.
        var lc:LocationCollection = LocationCollection(evt.resultData);

        //get the first geocode result out of the collection
        var geoAddr:GeoAddress = GeoAddress(lc.get(0));

        //build a nice string to show in the TextArea.
        var str:String = "Lat: "+geoAddr.getLatLng().lat+"
            Lng: "+geoAddr.getLatLng().lng+" Resultcode:
            "+geoAddr.getResultCode()+"\n";
        myTextArea.text="Geocode Results\n";
        myTextArea.text += str;

    }

    /**
     * Geocode Error - Handle errors.
     */
    private function onExecError(e:ExecFaultEvent):void {

        trace("Error retrieving data from the server.
            \n\nDid you set your Client Id and/or Password?",
            "Server Communication Error:");

    }

} // End Class

} // End Package

```

- 14) Replace the instances of **Your Client ID** and **Your Password** with the your account information.
- 15) Save the file containing the above code as **Geocode.as** in the same folder as the **.fla** file.
- 16) Select the **Geocode.fla** tab.
- 17) Select **File | Publish Settings**.
- 18) Select the **Flash** tab and then **Settings...**
- 19) Enter **Geocode** in the document class field and click **OK**.
- 20) Select **File | Export | Export Movie...**
- 21) Run the application by browsing to the exported movie and double-clicking the **Goecode.swf** file.
- 22) Enter an address and click the **Geocode Address**.



## How to Add a POI to the Geocoded Address

This sample will build on the previous sample. This sample utilizes the MapQuest Geocode servers to retrieve geocode data for an address and places a POI on a map. (Page can be seen [HERE](#).)

- 1) Save the **Geocode.fla** file as **GeocodeAndMap.fla**.
- 2) Save the **Geocode.as** file as **GeocodeAndMap.as**.
- 3) From the **Components** toolbox, click and drag a **TileMap** to the stage.
- 4) Select the **Parameters** tab relating to the **TileMap** component, and enter the values shown below.
  - **InstanceName:** myMap
  - **mapType:** map
  - **key:** ***Your User Key***
  - **zoomLevel:** 4
- 5) Save the newly created flash file as **GeocodeAndMap.fla**.
- 6) Enter the code below **GeocodeAndMap.as** file, overwriting the existing code.

```
package {

    //import the required classes
    import flash.display.Sprite;
    import flash.display.StageScaleMode;
    import flash.display.StageAlign;
    import flash.events.MouseEvent;
    import com.mapquest.*;

    public class BatchGeocode extends Sprite {
        //create and set required variable to execute
        the geocode process
        private var serverName:String = "geocode.access.mapquest.com";
        private var serverPath:String = "mq";
        private var serverPort:int = 80;
        private var clientId:String = "[Your Client ID]";
        private var password:String = "[Your Password]";

        /**
         * Constructor
         */
        public function BatchGeocode():void {

            // Set Stage Properties
            this.stage.scaleMode = StageScaleMode.NO_SCALE;
            this.stage.align=StageAlign.TOP_LEFT;

            // Add listener for button click
            myButton.addEventListener(MouseEvent.CLICK, onButtonClick);
        }

        /**
         * Handle button click
         */
        public function onButtonClick(evt:MouseEvent):void {
```



## How to Add a POI to the Geocoded Address (cont'd.)

```
// Create an empty location collection to hold
    the addresses that
var lc:LocationCollection = new LocationCollection();

// Create an empty address object.
var address1:Address = new Address();

//populate the address object with the
    information from the form
address1.setStreet(this.txtStreet1.text);
address1.setCity(this.txtCity1.text);
address1.setState(this.txtState1.text);
address1.setPostalCode(this.txtZip1.text);
address1.setCountry(this.txtCountry1.text);

// Add the address to the location collection created above
lc.add(address1);

//populate the address object with the information
    from the form
var address2:Address = new Address();
address2.setStreet(this.txtStreet2.text);
address2.setCity(this.txtCity2.text);
address2.setState(this.txtState2.text);
address2.setPostalCode(this.txtZip2.text);
address2.setCountry(this.txtCountry2.text);

// Add the address to the location collection created above
lc.add(address2);

// Setup server object for call.
var exec:Exec = new Exec(serverName, serverPath,
    serverPort);
exec.setClientId(clientId);
exec.setPassword(password);

// Add a listener to be called when batch
    geocode completes
exec.addEventListener(Exec.TRANS_TYPE_BATCH_GEOCODE,
    onBatchGeocode);

// Add a listener to the exec object. To handle
    the Geocode response - Error.
exec.addEventListener(Exec.EVENT_DO_TRANSACTION_ERROR,
    onExecError);

// call the batch geocode function of the exec
    object. Passing it the location collection
exec.batchGeocode(lc);
```



## How to Add a POI to the Geocoded Address (cont'd.)

```
    }

    /**
     * Handle successful call to batchGeocode service
     */
    private function onBatchGeocode(evt:ExecResultEvent):void {
        var str:String = "";

        // Get the locationCollectionCollection
        var lcc:LocationCollectionCollection =
            LocationCollectionCollection
            (ExecResultEvent(evt).resultData);

        // Loop through all the locationCollections in
        // my collectionCollection
        for (var i:int=0; i < lcc.getSize(); i++){
            // Get the 1st for each location collection.
            var geoAddr:GeoAddress = LocationCollection
                (lcc.getAt(i)).getAt(0);

            // Build a nice string to display the data
            str += "Lat: "+geoAddr.getLatLng().lat+ " Lng: " +
                geoAddr.getLatLng().lng+" Resultcode:
                "+geoAddr.getResultCode() + "\n";
        }

        // update results.
        this.txtResults.text += str;
    }

    /**
     * BatchGeocode Error - Handle errors.
     */
    private function onExecError(e:ExecFaultEvent):void {
        trace("Error retrieving data from the server.
        \n\nDid you set your Client Id and/or Password?",
            "Server Communication Error:");
    }

} // End Class

} // End Package
```

- 7) Enter your **Client ID** and **Password** in the appropriate places in the **GeocodeAndMap.as** file.
- 8) Select the **GeocodeAndMap.fla** tab.
- 9) Select **File | Publish Settings**. Select the **Flash** tab and then click **Settings...**
- 10) Enter **GeocodeAndMap** in the **document class** field and click **OK**.
- 11) Select **File | Export | Export Movie...**
- 12) Run the application by browsing to the exported movie and double-clicking the **GeocodeAndMap.swf** file.



## How to Add a POI to the Geocoded Address (cont'd.)

- 13) Enter an address and click **Geocode Address**.



## How to Calculate the Route

This sample will utilize the previous sample and the MapQuest Geocoding and Routing servers to calculate a route between two addresses. (Page can be seen [HERE](#).)

- 1) Save the **BatchGeocode.fla** file as **DoRoute.fla**.
- 2) Save the **BatchGeocode.as** file as **DoRoute.as**.
- 3) Select the **Parameters** tab relating to the Button component, and enter the values shown below.
  - **InstanceName:** myButton
  - **Label:** Calculate Route
- 4) From the **Components** toolbox, click and drag a **DataGrid** to the stage, and give it an instance name of myDataGrid.
- 5) Replace the code in the **DoRoute.as** file with the code shown below.

```
package {

    import fl.controls.DataGrid;
    import flash.display.*;
    import flash.events.MouseEvent;
    import com.mapquest.*;

    public class DoRoute extends Sprite {

        // Server for geocoding.
        private var geoCodeServerName:String =
            "geocode.access.mapquest.com";
        private var geoCodeServerPath:String = "mq";
        private var geoCodeServerPort:int = 80;

        // Server for routing.
        private var routeServerName:String =
            "route.access.mapquest.com";
        private var routeServerPath:String = "mq";
        private var routeServerPort:int = 80;

        // Client Id and password
        private var clientId:String = "[Your Client ID]";
        private var password:String = "[Your Password]";

        // Exec server object
        private var exec:Exec;

        /**
         * Constructor
         */
        public function DoRoute():void {
            // Set Stage Properties
```



## How to Calculate the Route (cont'd.)

```
this.stage.scaleMode = StageScaleMode.NO_SCALE;
this.stage.align=StageAlign.TOP_LEFT;

// Add Listener for button.
myButton.addEventListener(MouseEvent.CLICK,
    onClick);
}

/**
 * Handle button click
 */
public function onClick(e:MouseEvent):void {

    // Create an empty location collection to
    hold the addresses.
    var lc:LocationCollection = new LocationCollection();

    // Create an empty address object.
    var address1:Address = new Address();

    //populate the address object with the information
    from the form
    address1.setStreet(this.txtStreet1.text);
    address1.setCity(this.txtCity1.text);
    address1.setState(this.txtState1.text);
    address1.setPostalCode(this.txtZip1.text);
    address1.setCountry(this.txtCountry1.text);

    // Add the address to the location collection
    created above
    lc.add(address1);

    //populate the address object with the information
    from the form
    var address2:Address = new Address();
    address2.setStreet(this.txtStreet2.text);
    address2.setCity(this.txtCity2.text);
    address2.setState(this.txtState2.text);
    address2.setPostalCode(this.txtZip2.text);
    address2.setCountry(this.txtCountry2.text);

    // Add the address to the location collection created above
    lc.add(address2);

    // Set up connection for geocode server
    exec = new Exec(geoCodeServerName, geoCodeServerPath,
        geoCodeServerPort);
    exec.setClientId(clientId);
    exec.setPassword(password);

    // Add a listener to be called when batch geocode completes
```



## How to Calculate the Route (cont'd.)

```

exec.addListener(Exec.TRANS_TYPE_BATCH_GEOCODE,
    onBatchGeocode);

// Add a listener to the exec object. To handle
// the Geocode response - Error.
exec.addListener(Exec.EVENT_DO_TRANSACTION_ERROR,
    onBatchGeocodeError);

// Geocode all addresses involved.
exec.batchGeocode(lc);
}

/**
 * Handle successful response to BatchGeocode service call
 */
private function onBatchGeocode(evt:ExecResultEvent):void {
    var lcc:LocationCollectionCollection =
        LocationCollectionCollection(evt.resultData);
    var lc:LocationCollection = new LocationCollection();

    // Extract the geocoded address from the collection
    // of location collections
    for (var i:int=0; i < lcc.getSize(); i++) {
        lc.add(GeoAddress(LocationCollection(lcc.get(i)).get(0)));
    }

    /*
    Take the geocoded addresses and calculate the route.
    */
    // Set up connection for route server
    exec = new Exec(routeServerName, routeServerPath,
        routeServerPort);
    exec.setClientId(clientId);
    exec.setPassword(password);

    // Add Listeners for route calculation
    exec.addListener(Exec.TRANS_TYPE_DOROUTE, onDoRoute);
    exec.addListener(Exec.EVENT_DO_TRANSACTION_ERROR,
        onRouteCalcError);

    // Create an empty RouteOptions object;
    var routeOptions:RouteOptions = new RouteOptions();

    // Make the call
    exec.doRoute(lc, routeOptions, null);
}

/**
 * Handle successful response to DoRoute service call
 */
private function onDoRoute(evt:ExecResultEvent):void {

```



## How to Calculate the Route (cont'd.)

```
var rr:RouteResults = RouteResults(evt.resultData);
var treks:TrekRouteCollection = rr.getTrekRoutes();
var maneuvers:ManeuverCollection =
    TrekRoute(treks.get(0)).getManeuvers();

// Populate grid with results
myDataGrid.rowCount = maneuvers.getSize();
myDataGrid.columns = ["Route Instructions", "Distance
    (miles)", "Time (mins)"];
var os:String=new String();
for (var i:int = 0; i < maneuvers.getSize(); i++) {
var maneuver:Maneuver = Maneuver(maneuvers.get(i));
var d:String= (Math.round(maneuver.getDistance()
    * 10) / 10 ).toString();
var t:String= (Math.round((maneuver.getTime()/60)
    * 10) / 10 ).toString();
var ro:Object = { "Route Instructions":maneuver.
    getNarrative(),"Distance (miles)":d,"Time (mins)":t };
myDataGrid.addItem(ro);
}
}

/**
 * BatchGeocode Error - Handle errors.
 */
private function onBatchGeocodeError(e:ExecFaultEvent):void {
    trace("Error retrieving data from the server.
    \n\nDid you set your Client Id and/or Password?",
    "Server Communication Error:");
}

/**
 * Route Calculation Error - Handle errors.
 */
private function onRouteCalcError(e:ExecFaultEvent):void {
    trace("Error retrieving data from the server", "Error:");
}

} // End Class

} // End Package
```

- 6) Replace the instances of **Your ClientID** and **Your Password**, in the **DoRoute.as** file, with the proper information from your account.
- 7) Save the file containing the above code as **DoRoute.as** in the same folder as the **.fla** file.
- 8) Select the **DoRoute.fla** tab.
- 9) Select **File | Publish Settings**. Select the **Flash** tab and then click **Settings...**
- 10) Enter **DoRoute** in the document class field and click **OK**.
- 11) Select **File | Export | Export Movie...**



## How to Calculate the Route (cont'd.)

- 12) Run the application by browsing to the exported movie and double-clicking the **DoRoute.swf** file.
- 13) Enter addresses and click **Calculate Route**.



## How to Display the Route

This sample will utilize the previous sample and the MapQuest Geocoding, Routing and Mapping servers to calculate a route between two addresses and display it as an overlay on a map. (Page can be seen [HERE](#).)

- 1) Save the **DoRoute.fla** file as **OverlayRouteMap.fla**.
- 2) Select the **DataGrid** component in the **OverlayRouteMap.fla** file and delete it.
- 3) Drag a **Tilemap** component onto the stage.
- 4) Select the **Parameters** tab relating to the **TileMap** component and enter the values shown below.
  - **InstanceName:** myMap
  - **mapType:** map
  - **key:** ***Your User Key***
- 5) Save the **DoRoute.as** file as **OverlayRouteMap.as**.
- 6) Replace the code in the **OverlayRouteMap.as** file with the code below.

```
package {

    //import the required classes
    import flash.display.Stage;
    import flash.display.StageAlign;
    import flash.display.StageScaleMode;
    import fl.controls.DataGrid;
    import flash.display.*;
    import flash.events.MouseEvent;
    import com.mapquest.*;
    import com.mapquest.tilemap.overlays.*;

    public class OverlayRouteMap extends Sprite {
        // Server for geocoding.
        private var geoCodeServerName:String =
            "geocode.access.mapquest.com";
        private var geoCodeServerPath:String = "mq";
        private var geoCodeServerPort:int = 80;

        // Server for Routing
        private var routeServerName:String =
            "route.access.mapquest.com";
        private var routeServerPath:String = "mq";
        private var routeServerPort:int = 80;

        // Client Id and password
        private var clientId:String = "[Your Client ID]";
        private var password:String = "[Your Password]";

        // Exec server object
```



## How to Display the Route (cont'd.)

```
private var exec:Exec;

/**
 * Constructor
 */
public function OverlayRouteMap():void {
    // Set Stage Properties
    this.stage.scaleMode = StageScaleMode.NO_SCALE;
    this.stage.align=StageAlign.TOP_LEFT;

    // Add Listener for button.
    myButton.addEventListener(MouseEvent.CLICK, onButtonClick);
}

/**
 * Handle button click
 */
public function onButtonClick(e:MouseEvent):void {
    // Create an empty location collection to hold
    // the addresses that
    var lc:LocationCollection = new LocationCollection();

    // Create an empty address object.
    var address1:Address = new Address();

    //populate the address object with the information
    // from the form
    address1.setStreet(this.txtStreet1.text);
    address1.setCity(this.txtCity1.text);
    address1.setState(this.txtState1.text);
    address1.setPostalCode(this.txtZip1.text);
    address1.setCountry(this.txtCountry1.text);

    // Add the address to the location collection
    // created above
    lc.add(address1);

    //populate the address object with the information
    // from the form
    var address2:Address = new Address();
    address2.setStreet(this.txtStreet2.text);
    address2.setCity(this.txtCity2.text);
    address2.setState(this.txtState2.text);
    address2.setPostalCode(this.txtZip2.text);
    address2.setCountry(this.txtCountry2.text);

    // Add the address to the location collection
    // created above
    lc.add(address2);

    // Set up connection for geocode server
```



## How to Display the Route (cont'd.)

```

exec = new Exec(geoCodeServerName,
    geoCodeServerPath, geoCodeServerPort);
exec.setClientId(clientId);
exec.setPassword(password);

// Add a listener to be called when batch
// geocode completes
exec.addListener(Exec.TRANS_TYPE_BATCH_GEOCODE,
    onBatchGeocode);

// Add a listener to the exec object. To handle
// the Geocode response - Error.
exec.addListener(Exec.EVENT_DO_TRANSACTION_ERROR,
    onBatchGeocodeError);

// Geocode all addresses involved.
exec.batchGeocode(lc);
}

/**
 * Handle successful response to BatchGeocode service call
 */
private function onBatchGeocode(evt:ExecResultEvent):void {
    //Get the location collection collection from the event.
    var lcc:LocationCollectionCollection =
        LocationCollectionCollection(evt.resultData);

    // Create a LocationCollection for geocoded addresses.
    var lc:LocationCollection = new LocationCollection();

    // Get the geocoded addresses from the location
    // collections
    for (var i:int=0; i < lcc.getSize(); i++) {
        lc.add(GeoAddress(LocationCollection
            (lcc.get(i)).get(0)));
    }

    // Create an exec object this time pointing the
    // routing server.
    var exec:Exec = new Exec(routeServerName,
        routeServerPath, routeServerPort);
    exec.setClientId(clientId);
    exec.setPassword(password);

    // Create an empty route options object
    var routeOptions:RouteOptions = new RouteOptions();

    // Set the maxShapePoints property to 10000.
    // Just to be safe.
    routeOptions.setMaxShapePointsPerManeuver(10000);

```



## How to Display the Route (cont'd.)

```
// Add listeners to receive the results of this call.
exec.addListener(Exec.TRANS_TYPE_DOROUTE, onDoRoute);
exec.addListener(Exec.EVENT_DO_TRANSACTION_ERROR,
    onRouteCalcError);

// Make the call
exec.doRoute(lc, routeOptions, null);
}

/**
 * Handle successful response to DoRoute service call
 */
private function onDoRoute(evt:ExecResultEvent):void {
    // Extract the RouteResults object from the event data.
    var rr:RouteResults = RouteResults(evt.resultData);

    // Get the shape points from the routeResults object
    var shapes:IPointLLCollection = rr.getShapePoints();

    // Get rid of previous overlays
    myMap.removeAllOverlays();

    // If a route has been calculated then create an
    // overlay and place it on the map
    if (shapes.getSize() != 0 ) {
        // Create a line overlay and set it's display
        // properties
        var lo:LineOverlay = new LineOverlay();
        lo.setBorderWidth(10);
        lo.setColor(0x0000ff);
        lo.setColorAlpha(.4);
        lo.setShapePoints(shapes);

        // Add the line overlay to the map
        myMap.addOverlay(lo);

        // set zoom to the center of the route.
        myMap.zoomToRect(new RectLL(shapes.getPointLL(0),
            rr.getShapePoints().getPointLL(shapes.getSize()-1)));
    }
}

/**
 * BatchGeocode Error - Handle errors.
 */
private function onBatchGeocodeError
(e:ExecFaultEvent):void {
    trace("Error retrieving data from the server.
        \n\nDid you set your Client Id and/or Password?",
        "Server Communication Error:");
}
```



## How to Display the Route (cont'd.)

```

    }

    /**
     * Handle error getting route
     */
    private function onRouteCalcError(e:ExecFaultEvent):void {
        trace("Error retrieving data from the server", "Error:");
    }

    } // End Class

} // End Package

```

- 7) Replace the **Client ID** and **Password** fields in the code above with the appropriate values from your account.
- 8) Select the tab containing the **OverlayRouteMap.fla** file.
- 9) Select **File | Publish Settings**. Select the **Flash** tab and then click **Settings...**
- 10) Enter **OverlayRouteMap** in the `document class` field and click **OK**.
- 11) Select **File | Export | Export Movie...**
- 12) Run the application by browsing to the exported movie and double-clicking the **OverlayRouteMap.swf** file.
- 13) Enter addresses and click **Calculate Route**.  
Batch Geocode page can be seen [HERE](#).



## How to Display Search Results on a Map

This sample will utilize the Geocoding sample (from earlier in this guide), and the MapQuest Geocoding, Spatial Searching, and Mapping servers to display search results on a map. (Page can be seen [HERE](#).)

- 1) Open the **Geocode.as** file and save it as **SearchAndMap.as**.
- 2) Open the **Geocode.fla** file and save it as **SearchAndMap.fla**.
- 3) Select the `text` area on the stage and delete it.
- 4) From the **Components** toolbox, click and drag a **label** to the stage, and assign the parameters shown below.  
**Text:** Radius:
- 5) From the **Components** toolbox, click and drag a **TextInput** to the stage, and assign the parameters shown below.  
**InstanceName:** txtRadius
- 6) From the **Components** toolbox, click and drag a **label** to the stage, and assign the parameters shown below.  
**Text:** Distance Units:
- 7) From the **Components** toolbox, click and drag a **ComboBox** to the stage, and assign the parameters shown below.  
**InstanceName:** distanceUnitsCombo
- 8) From the **Components** toolbox, click and drag a **label** to the stage, and assign the parameters shown below.  
**Text:** Max Results:
- 9) From the **Components** toolbox, click and drag a **TextInput** to the stage, and assign the parameters shown below.



## How to Display Search Results on a Map (cont'd.)

**InstanceName:** txtMaxResults

10) Click **Geocode Address** on the stage and change its label to the value shown below.

**Label:** Search

11) From the **Components** toolbox, click and drag a **TileMap** to the stage.

12) Select the **Parameters** tab relating to the **TileMap** component, and enter the values shown below.

- **InstanceName:** myMap
- **mapType:** map
- **key:** ***Your User Key***
- **zoomLevel:** 4

13) Replace the code in the **SearchAndMap.as** file with the code shown below.

```
package {

    import flash.display.Stage;
    import flash.display.StageAlign;
    import flash.display.StageScaleMode;
    import com.mapquest.*;
    import com.mapquest.tilemap.controls.LargeZoomControl;
    import com.mapquest.tilemap.pois.*;
    import fl.data.DataProvider;
    import flash.display.*;
    import flash.events.MouseEvent;

    public class SearchAndMap extends Sprite {
        // Server for geocoding.
        private var geocodeServerName:String =
            "geocode.access.mapquest.com";
        private var geocodeServerPort:int = 80;
        private var geocodeServerPath:String = "mq";

        // Server for search calculation.
        private var searchServerName:String =
            "spatial.access.mapquest.com";
        private var searchServerPort:int = 80;
        private var searchServerPath:String = "mq";

        // Client Id and password
        private var clientId:String = "[Your Client ID]";
        private var password:String = "[Your Password]";

        // Geocoded address
        private var geocodedAddress:GeoAddress;

        /**
         * Constructor
         */
        public function SearchAndMap():void {
            // Set Stage Properties
```



## How to Display Search Results on a Map (cont'd.)

```

this.stage.scaleMode = StageScaleMode.NO_SCALE;
this.stage.align=StageAlign.TOP_LEFT;

// Populate combo box
var distanceUnits:DataProvider = new DataProvider()
distanceUnits.addItem( {label:"Miles",
    data:Constants.DISTANCEUNITS_MILES} );
distanceUnits.addItem( {label:"Kilometers",
    data:Constants.DISTANCEUNITS_KILOMETERS} );
distanceUnitsCombo.dataProvider = distanceUnits;

// Add listener for button click
myButton.addEventListener(MouseEvent.CLICK, \
    onSearchClick);

// Add Zoom control to map
myMap.addControl( new LargeZoomControl());
}

/**
 * Call Search service when SearchButton is clicked.
 */
private function onSearchClick(e:MouseEvent):void {
    // Geocode address.
    var address:Address = new Address();

    // populate the address object with the
    information from the form
    //address.setStreet(this.txtStreet.text);
    address.setCity(this.txtCity.text);
    address.setState(this.txtState.text);
    address.setPostalCode(this.txtZip.text);
    address.setCountry(this.txtCountry.text);

    // Create an exec object for the geocode call
    var exec:Exec = new Exec(geocodeServerName,
        geocodeServerPath, geocodeServerPort);
    exec.setClientId(clientId);
    exec.setPassword(password);

    // Add Listeners for Geocode call
    exec.addEventListener(Exec.TRANS_TYPE_GEOCODE,
        onGeocodeAddress);
    exec.addEventListener(Exec.EVENT_DO_TRANSACTION_ERROR,
        onGeocodeError);

    //call the geocode function of the exec object -
    give it the address object to be geocoded
    exec.geocode(address);
}

```



## How to Display Search Results on a Map (cont'd.)

```
/**
 * Handle successful response to Geocode service call
 */
private function onGeocodeAddress(evt:ExecResultEvent):
void {
    //Get the 1st geocode result out of the collection
    var lc:LocationCollection = LocationCollection
        (evt.resultData);
    geocodedAddress = GeoAddress(lc.get(0));

    // Create an exec object for the Search Call
    var exec:Exec = new Exec(searchServerName,
        searchServerPath, searchServerPort);
    exec.setClientId(clientId);
    exec.setPassword(password);

    // Add listener for the search call.
    exec.addEventListener(Exec.TRANS_TYPE_SEARCH, onSearch);
    exec.addEventListener(Exec.EVENT_DO_TRANSACTION_ERROR,
        onSearchError);

    // Create objects required for search parameters
    var distanceUnits:DistanceUnits = new
        DistanceUnits(distanceUnitsCombo.selectedItem["data"]);
    var searchCriteria:RadiusSearchCriteria = new
        RadiusSearchCriteria();

    // Set the center of the search to the address entered.
    searchCriteria.setCenter(geocodedAddress.getLatLng());

    // Set Maximum matches
    searchCriteria.setMaxMatches(int(Number
        (txtMaxResults.text)));

    // Set the distance units value
    searchCriteria.setRadius(Number(txtRadius.text),
        distanceUnits);

    // Create a query object.
    var dbLayerQueryCollection:DBLayerQueryCollection =
        new DBLayerQueryCollection();
    var dbLayerQuery:DBLayerQuery = new DBLayerQuery();
    dbLayerQuery.setDBLayerName("MQA.test");
    dbLayerQueryCollection.add(dbLayerQuery);

    // call the search function of the exec object
    exec.search(searchCriteria, "", dbLayerQueryCollection);
}

/**
 * Handle successful response from the Search service call.
```



## How to Display Search Results on a Map (cont'd.)

```

*/
private function onSearch(evt:ExecResultEvent):void {
    // Clear All Pois
    myMap.removeAllPois();

    // Get the feature collection from the event.
    var fc:FeatureCollection = FeatureCollection
        (evt.resultData);

    // Get each feature (which is a poi) from the
    // collection and add it to the map.
    for ( var i:int = 0; i < fc.getSize(); i++ ) {
        var pointFeature:PointFeature =
            PointFeature(fc.get(i));

        // Create a POI object and set it's location.
        var poi:Poi = new Poi(pointFeature.getCenterLatLng());

        // Set rollover and popup text.
        poi.setInfoTitle(pointFeature.getName());
        poi.setInfoContent(pointFeature.getCenterLatLng().lat
            + ' by ' +
            pointFeature.getCenterLatLng().lng + '\n Distance: ' +
            pointFeature.getDistance());
        poi.setKey(pointFeature.getKey());

        // Add the Poi to the map.
        myMap.addPoi(poi);
    }

    //Add the search origin
    poi = new Poi(geocodedAddress.getLatLng());
    poi.setInfoTitle(geocodedAddress.getStreet() + ' '
        + geocodedAddress.getCity());
    poi.setInfoContent(geocodedAddress.getResultCode());
    poi.setKey('Origin');
    myMap.addPoi(poi);

    //Best-Fit the map to the results
    myMap.bestFit();
}

/**
 * Geocode Error - Handle error from geocode call.
 */
private function onGeocodeError(e:ExecFaultEvent):void {
    trace("Error retrieving data from the server.
        \n\nDid you set your Client Id and/or Password?",
        "Server Communication Error:");
}

```



## How to Display Search Results on a Map (cont'd.)

```
/**
 * Search Error - Handle error from search call.
 */
private function onSearchError(e:ExecFaultEvent):void {
    trace("Error performing search. \n\nCheck server names.");
}

} // End Class

} // End Package
```

- 14) Replace the **Client ID** and **Password** in the above code block with the proper information for your account.
- 15) Select the tab containing the **SearchAndMap.fla** file.
- 16) Select **File | Publish Settings**. Select the **Flash** tab and then click **Settings...**
- 17) Enter **SearchAndMap** in the **document class** field and click **OK**.
- 18) Select **File | Export | Export Movie...**
- 19) Run the application by browsing to the exported movie and double-clicking the **SearchAndMap.swf** file.
- 20) Enter addresses and click **Calculate Route**.

## Using Remote Collections

Remote Collections can be added to MapQuest TileMaps, providing you the ability to read external data feeds, including KML and GeoRSS, and display their shapes and information on MapQuest TileMaps with minimal effort.



### How to Add and Remove RemoteCollections to and from a TileMap

The example below creates a RemoteCollection from a KML file and displays its shapes and information on a MapQuest TileMap.

- 1) Import the following package:

```
import com.mapquest.tilemap.RemoteCollection;
import com.mapquest.tilemap.deserializers.*;
```

- 2) Declare as many RemoteCollection objects as needed:

```
private var myKMLColl:RemoteCollection;
```

- 3) Initialize the RemoteCollection, passing arguments for the URL to the KML file and the deserialization class to be used.

```
myKMLColl = new RemoteCollection("http://www.someserver.com/
somefile.kml", DeserializeKML);
```

#### Note

If the external feed is a GeoRSS file, the deserialization class passed to the constructor should be `DeserializeGeoRSS`.

- 4) Call the TileMap object's `addShapeCollection()`, passing the name of the RemoteCollection to be added.

```
myMap.addShapeCollection(myKMLColl);
```

Alternatively, call the TileMap object's `removeShapeCollection()` method, passing the name of the RemoteCollection to be removed.

```
myMap.removeShapeCollection(myKMLColl);
```

## 5.1 Updating RemoteCollections

One of the benefits of using RemoteCollections on TileMaps is the ability to contain “real time” information (for example, traffic, weather, etc.). Once a RemoteCollection has been created, it can be easily updated to include the most recent information from the data provider.

Assuming a myGeoRSSColl is displayed on the map, the line below will force the RemoteCollection to reconnect to its data provider and retrieve the latest information. Any changes in Shapes or their content will be immediately displayed on the TileMap, without additional developer effort. If the RemoteCollection is not displayed on the map when it is updated, the updated Shape information will be stored in the appropriate RemoteCollection object.

```
myGeoRSSColl.update();
```

## 5.2 Using RemoteCollection Events

MapQuest offers you the ability to attach Event Listeners to RemoteCollections to determine if the feed has completed loading or has failed to load.

### REMOTE\_COLLECTION\_LOADED

The following line adds an Event Listener to the myGeoRSSColl to determine if the RemoteCollection has finished loading

```
myGeoRSSColl.addEventListener(RemoteCollection.REMOTE_COLLECTION_LOADED, onLoaded);
```

#### onLoaded

The onLoaded listener would look as shown below.

```
private function onLoaded(event:Event):void {  
    // do necessary actions here  
}
```

### REMOTE\_COLLECTION\_FAILED

The following line adds an Even Listener to the myGeoRSSColl to determine if the RemoteCollection has failed to load.

```
myGeoRSSColl.addEventListener(RemoteCollection.REMOTE_COLLECTION_FAILED, onFailed);
```

#### onFailed

The onFailed listener would look as shown below.

```
private function onFailed(event:Event):void {  
    // do necessary actions here  
}
```

## Using Multiple Collections

Multiple `ShapeCollections` can be added to MapQuest TileMaps, providing you the ability to define specific `ShapeCollections` (containing POIs, Overlays, or a combination of the two), manipulate the `ShapeCollections`, and determine which of these collections are displayed on the map at any given time.



### How to Add or Remove Multiple `ShapeCollections` To or From a `TileMap`

The following example assumes the necessary Shapes (POIs and Overlays) have already been created.

- 1) Import the following package:

```
import com.mapquest.tilemap.ShapeCollection;
```

- 2) Declare as many `ShapeCollection` objects as needed, as shown below:

```
private var myDormsColl:ShapeCollection = new ShapeCollection();
private var myAdminColl:ShapeCollection = new ShapeCollection();
private var myAcademicColl:ShapeCollection = new ShapeCollection();
private var mySearchColl:ShapeCollection = new ShapeCollection();
```

- 3) Populate the `ShapeCollections`, using the `ShapeCollection` object's `add()` method, and passing in the Shape to be added, as shown below:

```
myDormsColl.add(myPOI);
myAdminColl.add(myPOI);
myAcademicColl.add(myPOI);
mySearchColl.add(myPOI);
```

- 4) Call the `TileMap` object's `addShapeCollection()` method, passing the name of the `ShapeCollection` to be added.

```
myMap.addShapeCollection(myDormsColl);
myMap.addShapeCollection(myAdminColl);
myMap.addShapeCollection(myAcademicColl);
myMap.addShapeCollection(mySearchColl);
```

Alternatively, call the `TileMap` object's `removeShapeCollection()` method, passing the name of the `ShapeCollection` to be removed.



## How to Add or Remove Multiple ShapeCollections To or From a TileMap (cont'd.)

```
myMap.removeShapeCollection(myDormsColl);  
myMap.removeShapeCollection(myAdminColl);  
myMap.removeShapeCollection(myAcademicColl);  
myMap.removeShapeCollection(mySearchColl);
```



## How to Update ShapeCollections

ShapeCollections can be updated (Shapes added or removed) whether displayed on a TileMap, or not. If a ShapeCollection is displayed on a map, and changes are made to it (Shapes added or removed), these changes will be visible instantly to map users.

Assuming the `myAthleticColl` is displayed on the map, the following two lines will append the specified Shapes to the `myAthleticColl` ShapeCollection, and make the newly added Shapes visible to map users.

```
myAthleticColl.add(myTennisOL);  
myAthleticColl.add(myPracticeOL);
```



## How to Use ShapeCollection Decluttering

You can determine whether or not POIs contained in ShapeCollections are subject to decluttering by setting the ShapeCollection object's `setDeclutter()` method.

- 1) Declutter any POIs contained in `myAthleticColl` using the following.

```
myAcademicColl.setDeclutter(true);
```

- 2) Use the following line to ignore any POIs contained in `myAthleticColl` when POI decluttering is invoked.

```
myAcademicColl.setDeclutter(false);
```

- 3) Make any changes visible to users if the ShapeCollection's `setDeclutter()` method is altered after it is displayed on the map by utilizing the TileMap's `replaceShapeCollection()` method.

```
myMap.replaceShapeCollection(myAcademicColl,myAcademicColl);
```

## 6.1 ShapeCollection Properties

When ShapeCollections are created, whether visible on the map or not, they have various properties and methods that can be accessed to adjust their appearance and behavior.

- Access to a ShapeCollection's properties and methods, when it is not displayed on a tile map, can be performed as shown below.

```
myDormsColl.setDeclutter(true);
```

- Access to a `ShapeCollection`'s properties and methods, when it is displayed on a `TileMap`, can be performed utilizing the `TileMap`'s `getShapeCollection()` method, and passing in the name of the desired `ShapeCollection`.

```
myMap.getShapeCollection(myDormsColl).getDeclutter();
```

## 6.2 Using `TileMap` `ShapeCollection` Methods

In addition to adding and removing `ShapeCollections`, the `TileMap` object supports other `ShapeCollection` functionality described below.

### Replacing a `ShapeCollection` with Another `ShapeCollection`

Replacing a `ShapeCollection` with another `ShapeCollection` can be performed utilizing the `TileMap`'s `replaceShapeCollection()` method, and passing in the name of the desired `ShapeCollection` to be replaced and the name of the `ShapeCollection` that should replace it.

In the example below, the `myDormsColl` `ShapeCollection` will be replaced on the map by the `myAdminColl` `ShapeCollection`.

```
myMap.replaceShapeCollection(myDormsColl,myAdminColl);
```

### Determining the Number of User-Defined `ShapeCollections`

To determine the number of user-defined `ShapeCollections` displayed on the map, the `TileMap`'s `getShapeCollectionCount()` function can be used.

```
intNumColl = myMap.getShapeCollectionCount();
```

### Determining the Names of User-Defined `ShapeCollections`

To determine the names of user-defined `ShapeCollections` displayed on the map, the `TileMap`'s `getShapeCollectionNames()` function can be used to return an array of `ShapeCollection` names on the map.

```
arrNames = myMap.getShapeCollectionNames();
```

### Accessing the User-Defined `ShapeCollections`

To access the user-defined `ShapeCollections` displayed on the map, the `TileMap`'s `getShapeCollections()` function can be used to return an array of `ShapeCollections` on the map.

```
arrColls = myMap.getShapeCollections();
```

### Accessing a Specific User-Defined `ShapeCollection`

To access a specific user-defined `ShapeCollection` displayed on the map, the `TileMap`'s `getShapeCollection()` method can be used, passing in the name of the desired `ShapeCollection`.

```
myMap.getShapeCollection(myAdminColl).setDeclutter(true);
```

#### Note

For any of the above functions or methods, if multiple `ShapeCollections` exist with the same name, the `TileMap` Toolkit will act on the first collection it encounters with the specified name.

## 6.3 Default Shape Collection Methods

When a Shape (POI or Overlay) is added to or removed from a TileMap using the TileMap's `addShape()` or `removeShape()` methods, that Shape is added to or removed from the TileMap's default ShapeCollection.

### Adding a Shape to Default ShapeCollection

To add a Shape to the default ShapeCollection, use the TileMap's `addShape()` method, passing in the Shape to be added.

```
myMap.addShape(myPOI);
```

### Removing a Shape from Default ShapeCollection

To remove a Shape from the default ShapeCollection, use the TileMap's `removeShape()` method, passing in the Shape to be removed.

```
myMap.removeShape(myPOI);
```

### Accessing Shapes in Default ShapeCollection

To access the Shapes in the default ShapeCollection, use the TileMap's `getShapes()` function to return a ShapeCollection.

```
myShapeColl = myMap.getShapes();
```

### Replacing the Shapes in Default ShapeCollection

To replace the Shapes in the default ShapeCollection, use the TileMap's `replaceShapes()` method, passing in the name of the ShapeCollection that should replace the map's default ShapeCollection.

```
myMap.replaceShapes(myShapeColl);
```

### Adding a ShapeCollection to Default ShapeCollection

To add a ShapeCollection to the map's default ShapeCollection, use the TileMap's `addShapes()` method, passing in the name of the ShapeCollection that should be appended to the map's default ShapeCollection.

```
myMap.addShapes(myShapeColl);
```

### Removing a ShapeCollection from Default ShapeCollection

To remove all Shapes from the map's default ShapeCollection, use the TileMap's `removeAllShapes()` method.

```
myMap.removeAllShapes();
```

## Using Traffic

Your map can now display traffic market indicators, flow data, and incident data. Additionally, a traffic control can be added to your map that enables you to add the same traffic functionality that you see on [www.mapquest.com](http://www.mapquest.com) to your maps.

### 7.1 Adding Traffic to Your Maps

The traffic control itself has a lot of configurable parameters to customize the legend. However, each part of the traffic functionality is available separately and is completely customizable.

#### Traffic API Components

The Traffic API is composed of three main items:

- **Traffic Market Indicators:**  
Icons that will display on the map at zoom levels 2-6, showing that traffic is available for that area.
- **Traffic Flow Data:**  
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). Traffic flow will display from zoom levels 7-16.
- **Traffic Incident Data:**  
Incidents that are displayed on the map for construction, traffic incidents, and events.

The following procedure provides the basic information for getting traffic on your map.



#### How to Create the Traffic Object

Page can be seen [HERE](#).

- 1) Import the traffic class.

```
import com.mapquest.tilemap.Traffic;
```

- 2) Declare the traffic object.



## How to Create the Traffic Object (cont'd.)

```
private var myTraffic:Traffic = new Traffic(myMap.getTileMap());  
  
myTraffic = new Traffic(myMap.getTileMap());
```

The `myMap.getTileMap` returns an `MQTileMap` object.



## How to Access the Traffic Flow Image Overlay

- 1) Add the traffic flow image overlay.

```
myTraffic.addFlow();
```

Alternatively, remove the traffic flow image overlay

```
mytraffic.removeFlow();
```

- 2) Set the opacity of the traffic flow image overlay.

```
myTraffic.setFlowOpacity(0.5);
```

- 3) Get the opacity of the traffic flow image overlay.

```
var myFlowOpacity:Number = myTraffic.getFlowOpacity();
```

- 4) Add traffic flow event listeners.

```
myTraffic.addEventListener(TrafficEvent.FLOW_ADDED,onTrafficFlowAdded);
```

Alternatively, remove the traffic flow event listeners.

```
myTraffic.removeEventListener(TrafficEvent.FLOW_REMOVED,  
    onTrafficFlowRemoved);
```

- 5) Include the traffic flow event handler for traffic flow being added.

```
private function onTrafficFlowAdded(e:TrafficEvent):void {  
    trace("Traffic flow has been added.");  
}
```

Alternatively remove the traffic flow event handler for traffic flow being removed.

```
private function onTrafficFlowRemoved(e:TrafficEvent):void {  
    trace("Traffic flow has been removed.");  
}
```



## How to Access the Traffic Market POIs

- 1) Add the traffic market icons.

```
myTraffic.addMarkets();
```

- 2) Create the custom Market POI icon.

```
myMarketIcon = new ImageMapIcon();
myMarketIcon.setImageURL("image.gif",10,10);
myTraffic.setMarketIcon(myMarketIcon);
```

- 3) Get a reference to the Market POI icon.

```
myIcon:ImageMapIcon = ImageMapIcon(myTraffic.getMarketIcon());
```

- 4) Add Market event listeners.

```
myTraffic.addEventListener(TrafficEvent.MARKETS_ADDED,onMarketsAdded);
myTraffic.addEventListener(TrafficEvent.MARKETS_REMOVED,
    onMarketsRemoved);
myTraffic.addEventListener(TrafficEvent.MARKETS_TIMEOUT,
    onMarketsTimeout);
```

- 5) Add Market event handlers.

```
private function onMarketsAdded(e:TrafficEvent):void {
    trace("Markets have been added.");
}

private function onMarketsRemoved(e:TrafficEvent):void {
    trace("Markets have been removed.");
}
```

- 6) Add Market timeout event.

```
private function onMarketsTimeout(e:TrafficEvent):void {
    trace("Markets have timed-out.");
}
```

- 7) Access the individual Traffic Market POIs as they are created.

```
myTraffic.addEventListener(TrafficPoiEvent.MARKET_POI_CREATED,
    onMarketPoiCreated);
```

- 8) Add the listener for individual Market POI manipulation.



## How to Access the Traffic Market POIs (cont'd.)

```
private function onMarketPoiCreated(e:TrafficPoiEvent):void {
    e.poi.setInfoWindowTitleBackgroundColor(0x540000);
    e.poi.setInfoTitle("Rollover Text");
    e.poi.setInfoWindowTitleText("Title of the InfoWindow");

    //access the XML for the POI by using e.xml (if desired)
}
```



## How to Access the Traffic Incident POIs

- 1) Add the Traffic Market icons.

```
myTraffic.addIncidents();
```

- 2) Create the custom Incident POI icons by creating an array of ImageMapIcons that gets passed into the following function.

Five possible severities can be accessed:

- 0 & 1 – minimal impact
- 2 & 3 – moderate impact
- 4 – severe impact

### Note

---

A `traffic.setSeverityText` method exists that replaces the default InfoWindow Title text.

---

This method takes an array of strings to be used to replace minimal, moderate, and severe.

```
myTraffic.setIncidentIcons(myIncidentIconArr);
```

- 3) Get a reference to the Incident POI Icons Array, which returns an array of ImageMapIcons.

```
myIconArray:Array = Array(myTraffic.getIncidentIcons());
```

- 4) Add incident event listeners.

```
myTraffic.addEventListener(TrafficEvent.INCIDENTS_ADDED,
    onIncidentssAdded);
myTraffic.addEventListener(TrafficEvent.INCIDENTS_REMOVED,
    onIncidentsRemoved);
myTraffic.addEventListener(TrafficEvent.INCIDENTS_TIMEOUT,
    onIncidentsTimeout);
```

- 5) Add Incident event handlers.



## How to Access the Traffic Incident POIs (cont'd.)

```
private function onIncidentsAdded(e:TrafficEvent):void {
    trace("Incidentss have been added.");
}

private function onIncidentsRemoved(e:TrafficEvent):void {
    trace("Incidents have been removed.");
}

private function onIncidentsTimeout(e:TrafficEvent):void {
    trace("Incidents have timed-out.");
}
```

- 6) Set minimum and maximum zoom levels for Incident POI icons.

The default is min=7 and max=16.

```
myTraffic.setMinMarketZoomLevel(7);
myTraffic.setMaxMarketZoomLevel(16);
```

- 7) Get the minimum and maximum zoom levels for Incident POI icons.

```
myMin:int = myTraffic.getMinMarketZoomLevel();
myMax:int = myTraffic.getMaxMarketZoomLevel();
```



## How to Use the Assorted Traffic Functionality

- 1) Set timeout values for Market and Incident POIs.

```
myTraffic.setTimeoutDuration(10000);
```

- 2) Get timeout values for Market and Incident POIs in milliseconds.

```
myTO:int = myTraffic.getTimeoutDuration();
```

- 3) Set the maximum InfoWindow width for Market & Incident POIs in pixels.

```
myTraffic.setMaxInfoWindowWidth(100);
```

- 4) Get the maximum InfoWindow width for Market & Incident POIs in pixels.

```
myWidth:int = myTraffic.getMaxInfoWindowWidth();
```

## 7.2 Traffic Events

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

**Table 7–1 Traffic Events**

Event Name
<code>TrafficEvent.FLOW_ADDED</code>
<code>TrafficEvent.FLOW_REMOVED</code>
<code>TrafficEvent.MARKETS_ADDED</code>
<code>TrafficEvent.MARKETS_REMOVED</code>
<code>TrafficEvent.MARKETS_TIMEOUT</code>
<code>TrafficEvent.INCIDENTS_ADDED</code>
<code>TrafficEvent.INCIDENTS_REMOVED</code>
<code>TrafficEvent.INCIDENTS_TIMEOUT</code>

An example of this implementation would be:

```
myTraffic.addListener(TrafficEvent.FLOW_ADDED, onTrafficFlowAdded);

private function onTrafficFlowAdded(e:TrafficEvent):void
{
    Alert.show("Flow has been added");
}
```

## 7.3 Traffic Control Button

The Traffic Control automatically got included when you double-clicked the `.mxp` file containing the libraries and the Adobe Extension Manager installed the components. The traffic control will not appear on the stage, but will show up as an outline square. The control itself will appear in the `swf` at run time.

Page can be seen [HERE](#).

The code used to add the Traffic Control button is:

```
//add the traffic control to the map
myMap.addControl(new TrafficControl());
```