

Multiple ShapeCollections

Multiple ShapeCollections can be added to MapQuest TileMaps, providing developers 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.

Adding/Removing Multiple ShapeCollections to/from a TileMap

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

Import the following package:

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

Declare as many ShapeCollection objects as needed, as shown below:

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

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

```
myDormsColl.add(myPOI);  
myDormsColl.add(myPolyOL);
```

Call the TileMap object's addShapeCollection() or removeShapeCollection() methods, passing the name of the ShapeCollection to be added or removed.

The following example adds both ShapeCollections to the map to be displayed.

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

The following example removes both ShapeCollections from the map.

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

Updating 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);
```

ShapeCollection Decluttering

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

The following line will subject any POIs contained in myAthleticColl to POI decluttering when it is invoked.

```
myAthleticColl.setDeclutter(true);
```

The following line will ignore any POIs contained in myAthleticColl when POI decluttering when is invoked.

```
myAthleticColl.setDeclutter(false);
```

If the ShapeCollection's setDeclutter() method is altered after it is displayed on the map, the TileMap's replaceShapeCollection() method must be utilized to make these changes evident to map users.

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

Accessing 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();
```

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.

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

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

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

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

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

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

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, as shown below.

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

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, as shown below.

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

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

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

TileMap ShapeCollection Methods

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

Replacing a shape collection 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);
```

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

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

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, as shown below.

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

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, as shown below.

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

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, as shown below.

```
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 upon the first collection it encounters with the specified name.

NOTE: For a complete list of available properties and methods pertaining to ShapeCollections, or for any of the classes mentioned above, please refer to the Developer's Guide available on the MapQuest Developer Network at <http://www.developer.mapquest.com>