MKMapViewDelegate Protocol Reference

User Experience

2010-05-11

Every effort has been made to ensure that the information in this document is accurate. Apple is not responsible for typographical errors.

Apple Inc.
1 Infinite Loop
Cupertino, CA 95014
408-996-1010

Simultaneously published in the United States and Canada.

Even though Apple has reviewed this document, APPLE MAKES NO WARRANTY OR REPRESENTATION, EITHER EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT, ITS QUALITY, ACCURACY, MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE. AS A RESULT, THIS DOCUMENT IS PROVIDED "AS IS," AND YOU, THE READER, ARE ASSUMING THE ENTIRE RISK AS TO ITS QUALITY AND ACCURACY.

IN NO EVENT WILL APPLE BE LIABLE FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES RESULTING FROM ANY DEFECT OR INACCURACY IN THIS DOCUMENT, even if advised of the possibility of such damages.

THE WARRANTY AND REMEDIES SET FORTH ABOVE ARE EXCLUSIVE AND IN LIEU OF ALL OTHERS, ORAL OR WRITTEN, EXPRESS OR IMPLIED. No Apple dealer, agent, or employee is authorized to make any modification, extension, or addition to this warranty.

Some states do not allow the exclusion or limitation of implied warranties or liability for incidental or consequential damages, so the above limitation or exclusion may not apply to you. This warranty gives you specific legal rights, and you may also have other rights which vary from state to state.

Contents

MKMapViewDelegate Protocol Reference 5

Overview 5
Tasks 5
- Responding to Map Position Changes 5
- Loading the Map Data 5
- Tracking the User Location 6
- Managing Annotation Views 6
- Dragging an Annotation View 6
- Selecting Annotation Views 6
- Managing Overlay Views 6

Instance Methods 7
- mapView:annotationView:calloutAccessoryControlTapped: 7
- mapView:annotationView:didChangeDragState:fromOldState: 7
- mapView:didAddAnnotationViews: 8
- mapView:didAddOverlayViews: 8
- mapView:didDeselectAnnotationView: 9
- mapView:didFailToLocateUserWithError: 9
- mapView:didSelectAnnotationView: 9
- mapView:didUpdateUserLocation: 10
- mapView:regionDidChangeAnimated: 10
- mapView:regionWillChangeAnimated: 11
- mapView:viewForAnnotation: 11
- mapView:viewForOverlay: 12
- mapViewDidFailLoadingMap:withError: 12
- mapViewDidFinishLoadingMap: 13
- mapViewDidStopLocatingUser: 13
- mapViewWillStartLoadingMap: 13
- mapViewWillStartLocatingUser: 14

Document Revision History 15

MKMapViewDelegate Protocol Reference

Conforms to	NSObject
Framework	/System/Library/Frameworks/MapKit.framework
Availability	Available in iOS 3.0 and later.
Declared in	MKMapView.h

Overview

The `MKMapViewDelegate` protocol defines a set of optional methods that you can use to receive map-related update messages. Because many map operations require the `MKMapView` class to load data asynchronously, the map view calls these methods to notify your application when specific operations complete. The map view also uses these methods to request annotation and overlay views and to manage interactions with those views.

Important: The MapKit framework uses Google services to provide map data. Use of this protocol and the associated interfaces binds you to the Google Maps/Google Earth API terms of service. You can find these terms of service at http://code.google.com/apis/maps/iphone/terms.html.

Tasks

Responding to Map Position Changes

`– mapView:regionWillChangeAnimated:` (page 11)
Tells the delegate that the region displayed by the map view is about to change.

`– mapView:regionDidChangeAnimated:` (page 10)
Tells the delegate that the region displayed by the map view just changed.

Loading the Map Data

`– mapViewWillStartLoadingMap:` (page 13)
Tells the delegate that the specified map view is about to retrieve some map data.

`– mapViewDidFinishLoadingMap:` (page 13)
Tells the delegate that the specified map view successfully loaded the needed map data.

– `mapViewDidFailLoadingMap:withError:` (page 12)
    Tells the delegate that the specified view was unable to load the map data.

Tracking the User Location

– `mapViewWillStartLocatingUser:` (page 14)
    Tells the delegate that the map view will start tracking the user's position.

– `mapViewDidStopLocatingUser:` (page 13)
    Tells the delegate that the map view stopped tracking the user's location.

– `mapView:didUpdateUserLocation:` (page 10)
    Tells the delegate that the location of the user was updated.

– `mapView:didFailToLocateUserWithError:` (page 9)
    Tells the delegate that an attempt to locate the user's position failed.

Managing Annotation Views

– `mapView:viewForAnnotation:` (page 11)
    Returns the view associated with the specified annotation object.

– `mapView:didAddAnnotationViews:` (page 8)
    Tells the delegate that one or more annotation views were added to the map.

– `mapView:annotationView:calloutAccessoryControlTapped:` (page 7)
    Tells the delegate that the user tapped one of the annotation view's accessory buttons.

Dragging an Annotation View

– `mapView:annotationView:didChangeDragState:fromOldState:` (page 7)
    Tells the delegate that the drag state of one of its annotation views changed.

Selecting Annotation Views

– `mapView:didSelectAnnotationView:` (page 9)
    Tells the delegate that one of its annotation views was selected.

– `mapView:didDeselectAnnotationView:` (page 9)
    Tells the delegate that one of its annotation views was deselected.

Managing Overlay Views

– `mapView:viewForOverlay:` (page 12)
    Asks the delegate for the overlay view to use when displaying the specified overlay object.

– `mapView:didAddOverlayViews:` (page 8)
    Tells the delegate that one or more overlay views were added to the map.

Instance Methods

mapView:annotationView:calloutAccessoryControlTapped:

Tells the delegate that the user tapped one of the annotation view's accessory buttons.

```
- (void)mapView:(MKMapView *)mapView  annotationView:(MKAnnotationView *)view
    calloutAccessoryControlTapped:(UIControl *)control
```

Parameters

mapView

The map view containing the specified annotation view.

view

The annotation view whose button was tapped.

control

The control that was tapped.

Discussion

Accessory views contain custom content and are positioned on either side of the annotation title text. If a view you specify is a descendant of the `UIControl` class, the map view calls this method as a convenience whenever the user taps your view. You can use this method to respond to taps and perform any actions associated with that control. For example, if your control displayed additional information about the annotation, you could use this method to present a modal panel with that information.

If your custom accessory views are not descendants of the `UIControl` class, the map view does not call this method.

Availability

Available in iOS 3.0 and later.

Declared In

`MKMapView.h`

mapView:annotationView:didChangeDragState:fromOldState:

Tells the delegate that the drag state of one of its annotation views changed.

```
- (void)mapView:(MKMapView *)mapView annotationView:(MKAnnotationView
    *)annotationView didChangeDragState:(MKAnnotationViewDragState)newState
       fromOldState:(MKAnnotationViewDragState)oldState
```

Parameters

mapView

The map view containing the annotation view.

annotationView

The annotation view whose drag state changed.

newState

The new drag state of the annotation view.

oldState

The previous drag state of the annotation view.

Discussion

The drag state typically changes in response to user interactions with the annotation view. However, the annotation view itself is responsible for changing that state as well.

Availability

Available in iOS 4.0 and later.

Declared In

`MKMapView.h`

mapView:didAddAnnotationViews:

Tells the delegate that one or more annotation views were added to the map.

```
- (void)mapView:(MKMapView *)mapView   didAddAnnotationViews:(NSArray *)views
```

Parameters

mapView

The map view that added the annotation views.

views

An array of `MKAnnotationView` objects representing the views that were added.

Discussion

By the time this method is called, the specified views are already added to the map.

Availability

Available in iOS 3.0 and later.

Declared In

`MKMapView.h`

mapView:didAddOverlayViews:

Tells the delegate that one or more overlay views were added to the map.

```
- (void)mapView:(MKMapView *)mapView didAddOverlayViews:(NSArray *)overlayViews
```

Parameters

mapView

The map view that added the overlay views.

overlayViews

An array of `MKOverlayView` objects representing the views that were added.

Discussion

By the time this method is called, the specified views are already added to the map.

Availability

Available in iOS 4.0 and later.

Declared In

`MKMapView.h`

mapView:didDeselectAnnotationView:

Tells the delegate that one of its annotation views was deselected.

```
- (void)mapView:(MKMapView *)mapView didDeselectAnnotationView:(MKAnnotationView
    *)view
```

Parameters

mapView

The map view containing the annotation view.

view

The annotation view that was deselected.

Discussion

You can use this method to track changes in the selection state of annotation views.

Availability

Available in iOS 4.0 and later.

Declared In

`MKMapView.h`

mapView:didFailToLocateUserWithError:

Tells the delegate that an attempt to locate the user’s position failed.

```
- (void)mapView:(MKMapView *)mapView didFailToLocateUserWithError:(NSError *)error
```

Parameters

mapView

The map view that is tracking the user’s location.

error

An error object containing the reason why location tracking failed.

Availability

Available in iOS 4.0 and later.

Declared In

`MKMapView.h`

mapView:didSelectAnnotationView:

Tells the delegate that one of its annotation views was selected.

```
- (void)mapView:(MKMapView *)mapView didSelectAnnotationView:(MKAnnotationView
    *)view
```

Parameters

mapView

The map view containing the annotation view.

view

The annotation view that was selected.

Discussion
You can use this method to track changes in the selection state of annotation views.

Availability
Available in iOS 4.0 and later.

Declared In
`MKMapView.h`

mapView:didUpdateUserLocation:

Tells the delegate that the location of the user was updated.

```
- (void)mapView:(MKMapView *)mapView didUpdateUserLocation:(MKUserLocation
    *)userLocation
```

Parameters

`mapView`

The map view that is tracking the user's location.

`userLocation`

The location object representing the user's latest location.

Discussion
While the `showsUserLocation` property is set to `YES`, this method is called whenever a new location update is received by the map view.

Availability
Available in iOS 4.0 and later.

Declared In
`MKMapView.h`

mapView:regionDidChangeAnimated:

Tells the delegate that the region displayed by the map view just changed.

```
- (void)mapView:(MKMapView *)mapView   regionDidChangeAnimated:(BOOL)animated
```

Parameters

`mapView`

The map view whose visible region changed.

`animated`

If `YES`, the change to the new region was animated.

Discussion
This method is called whenever the currently displayed map region changes. During scrolling, this method may be called many times to report updates to the map position. Therefore, your implementation of this method should be as lightweight as possible to avoid affecting scrolling performance.

Availability
Available in iOS 3.0 and later.

Declared In
`MKMapView.h`

mapView:regionWillChangeAnimated:

Tells the delegate that the region displayed by the map view is about to change.

```
- (void)mapView:(MKMapView *)mapView  regionWillChangeAnimated:(BOOL)animated
```

Parameters

mapView
The map view whose visible region is about to change.

animated
If `YES`, the change to the new region will be animated. If `NO`, the change will be made immediately.

Discussion
This method is called whenever the currently displayed map region changes. During scrolling, this method may be called many times to report updates to the map position. Therefore, your implementation of this method should be as lightweight as possible to avoid affecting scrolling performance.

Availability
Available in iOS 3.0 and later.

Declared In
`MKMapView.h`

mapView:viewForAnnotation:

Returns the view associated with the specified annotation object.

```
- (MKAnnotationView *)mapView:(MKMapView *)mapView  viewForAnnotation:(id
   <MKAnnotation>)annotation
```

Parameters

mapView
The map view that requested the annotation view.

annotation
The object representing the annotation that is about to be displayed. In addition to your custom annotations, this object could be an `MKUserLocation` object representing the user's current location.

Return Value
The annotation view to display for the specified annotation or `nil` if you want to display a standard annotation view.

Discussion
Rather than create a new view each time this method is called, you should use the `dequeueReusableAnnotationViewWithIdentifier:` method of the `MKMapView` class to see if an existing annotation view of the desired type already exists. If one does exist, you should update the view to reflect the attributes of the specified annotation and return it. If a view of the appropriate type does not exist, you should create one, configure it with the needed annotation data, and return it.

If the object in the annotation *parameter* is an instance of the MKUserLocation class, you can provide a custom view to denote the user's location. To display the user's location using the default system view, return nil.

If you do not implement this method, or if you return nil from your implementation for annotations other than the user location annotation, the map view uses a standard pin annotation view.

Availability
Available in iOS 3.0 and later.

Declared In
MKMapView.h

mapView:viewForOverlay:

Asks the delegate for the overlay view to use when displaying the specified overlay object.

```
- (MKOverlayView *)mapView:(MKMapView *)mapView viewForOverlay:(id
    <MKOverlay>)overlay
```

Parameters

mapView
The map view that requested the overlay view.

overlay
The object representing the overlay that is about to be displayed.

Return Value
The view to use when presenting the specified overlay on the map. If you return nil, no view is displayed for the specified overlay object.

Discussion
If you support the presentation of overlays, you must implement this method and provide the views for your overlay objects.

Availability
Available in iOS 4.0 and later.

Declared In
MKMapView.h

mapViewDidFailLoadingMap:withError:

Tells the delegate that the specified view was unable to load the map data.

```
- (void)mapViewDidFailLoadingMap:(MKMapView *)mapView  withError:(NSError *)error
```

Parameters

mapView
The map view that started the load operation.

error
The reason that the map data could not be loaded.

Discussion
This method might be called in situations where the device does not have access to the network or is unable to load the map data for some reason. It may also be called if a request for additional map tiles comes in while a previous request for tiles is still pending. You can use this message to notify the user that the map data is unavailable.

Availability
Available in iOS 3.0 and later.

Declared In
`MKMapView.h`

mapViewDidFinishLoadingMap:

Tells the delegate that the specified map view successfully loaded the needed map data.

```
- (void)mapViewDidFinishLoadingMap:(MKMapView *)mapView
```

Parameters

`mapView`
The map view that started the load operation.

Discussion
This method is called when all map tiles in the currently visible area have been loaded.

Availability
Available in iOS 3.0 and later.

Declared In
`MKMapView.h`

mapViewDidStopLocatingUser:

Tells the delegate that the map view stopped tracking the user’s location.

```
- (void)mapViewDidStopLocatingUser:(MKMapView *)mapView
```

Parameters

`mapView`
The map view that stopped tracking the user’s location.

Discussion
This method is called when the value of the `showsUserLocation` property changes to `NO`.

Availability
Available in iOS 4.0 and later.

Declared In
`MKMapView.h`

mapViewWillStartLoadingMap:

Tells the delegate that the specified map view is about to retrieve some map data.

```
- (void)mapViewWillStartLoadingMap:(MKMapView *)mapView
```

Parameters

mapView

The map view that began loading the data.

Discussion

This method is called whenever a new group of map tiles need to be downloaded from the server. This typically occurs whenever you expose portions of the map by panning or zooming the content. You can use this method to mark the time that it takes for the map view to load the data.

Availability

Available in iOS 3.0 and later.

Declared In

`MKMapView.h`

mapViewWillStartLocatingUser:

Tells the delegate that the map view will start tracking the user's position.

```
- (void)mapViewWillStartLocatingUser:(MKMapView *)mapView
```

Parameters

mapView

The map view that is tracking the user's location.

Discussion

This method is called when the value of the `showsUserLocation` property changes to `YES`.

Availability

Available in iOS 4.0 and later.

Declared In

`MKMapView.h`

Document Revision History

This table describes the changes to *MKMapViewDelegate Protocol Reference*.

Date	Notes
2010-05-11	Added delegate methods introduced in iOS 4.0.
2009-08-19	Corrected the description of the mapView:viewForAnnotation: method.
2009-05-12	New document that describes the methods for responding to map changes.