

# Getting the Heading and Course of a Device

Core Location supports two different ways to get direction-related information:

- Devices with a magnetometer can report the direction in which a device is pointing, also known as its *heading*.
- Devices with GPS hardware can report the direction in which a device is moving, also known as its *course*.

Heading and course information don't represent the same information. The heading of a device reflects the actual orientation of the device relative to true north or magnetic north. The course of the device represents the direction of travel and doesn't take into account the device orientation. Depending on your app, you might prefer one type of information over the other or use a combination of the two. For example, a navigation app might toggle between course and heading information depending on the user's current speed. At walking speeds, heading information would be more useful for orienting the user to the current environment, whereas in a car, course information provides the general direction of the car's movement.

## Adding a Requirement for Direction-Related Events

If your iOS app requires direction-related information in order to function properly, include the `UIRequiredDeviceCapabilities` key in the app's `Info.plist` file. This key contains an array of strings indicating the features that your app requires of the underlying iOS-based device. The App Store uses this information to prevent users from installing apps on a device without the minimum required hardware.

For direction-related events, you can associate two relevant strings with the `UIRequiredDeviceCapabilities` key:

- `magnetometer`—Include this string if your app requires heading information.
- `gps`—Include this string if your app requires course-related information.

**Important:** If your iOS app uses heading or course events but is able to operate successfully without them, don't include the corresponding string value with the `UIRequiredDeviceCapabilities` key.

In both cases, also include the `location-services` string in the array. For more information about the `UIRequiredDeviceCapabilities` key, see *Information Property List Key Reference*.

## Getting Heading–Related Events

Heading events are available to apps running on a device that contains a magnetometer. A magnetometer measures nearby magnetic fields emanating from the Earth and uses them to determine the precise orientation of the device. Although a magnetometer can be affected by local magnetic fields, such as those emanating from fixed magnets found in audio speakers, motors, and many other types of electronic devices, Core Location is smart enough to filter out fields that move with the device.

Heading values can be reported relative either to magnetic north or true north on the map. Magnetic north represents the point on the Earth’s surface from which the planet’s magnetic field emanates. This location is not the same as the North Pole, which represents true north. Depending on the location of the device, magnetic north may be good enough for many purposes, but the closer to the poles you get, the less useful this value becomes.

To get heading events:

1. Create a `CLLocationManager` object.
2. Determine whether heading events are available by calling the `headingAvailable` class method.
3. Assign a delegate to the location manager object.
4. If you want true north values, start location services.
5. Call the `startUpdatingHeading` method to begin the delivery of heading events.

Listing 3–1 shows a custom method that configures a location manager and starts the delivery of heading events. In this case, the object is a view controller that displays the current heading to the user. Because the view controller displays the true north heading value, it starts location updates in addition to heading updates.

### Listing 3–1 Initiating the delivery of heading events

```
- (void)startHeadingEvents {
    if (!self.locManager) {
        CLLocationManager* theManager = [[[CLLocationManager alloc] init]
        autorelease];

        // Retain the object in a property.
        self.locManager = theManager;
        locManager.delegate = self;
    }
}
```

Apple v. Maxell

```

    }

    // Start location services to get the true heading.
    locationManager.distanceFilter = 1000;
    locationManager.desiredAccuracy = kCLLocationAccuracyKilometer;
    [locationManager startUpdatingLocation];

    // Start heading updates.
    if ([CLLocationManager headingAvailable]) {
        locationManager.headingFilter = 5;
        [locationManager startUpdatingHeading];
    }
}

```

The object you assign to the delegate property must conform to the `CLLocationManagerDelegate` [protocol](#). When a new heading event arrives, the location manager object calls the `locationManager:didUpdateHeading:` method to deliver that event to your app. Upon receiving a new event, check the `headingAccuracy` property to ensure that the data you just received is valid, as shown in Listing 3–2. In addition, if you are using the true heading value, also check whether it contains a valid value before using it.

### Listing 3–2 Processing heading events

```

- (void)locationManager:(CLLocationManager *)manager didUpdateHeading:
(CLHeading *)newHeading {
    if (newHeading.headingAccuracy < 0)
        return;

    // Use the true heading if it is valid.
    CLLocationDirection theHeading = ((newHeading.trueHeading > 0) ?
        newHeading.trueHeading : newHeading.magneticHeading);

    self.currentHeading = theHeading;
    [self updateHeadingDisplays];
}

```

# Getting Course Information While the User Is Moving

Devices that include GPS hardware can generate information that represents the device's current course and speed. Course information indicates the direction in which the device is moving and doesn't necessarily reflect the orientation of the device itself. As a result, course information is primarily intended for apps that provide navigation information while the user is moving.

The actual course and speed information is returned to your app in the same `CLLocation` objects you use to get the user's position. When you start location updates, Core Location automatically provides course and speed information when it's available. The framework uses the incoming location events to compute the current direction of motion. For more information on how to start location updates, see [Getting the User's Location](#).

---

Copyright © 2016 Apple Inc. All Rights Reserved. Terms of Use | Privacy Policy | Updated: 2016-03-21