

# Determining and Monitoring the Connectivity Status

Some of the most common uses for repeating alarms and background services is to schedule regular updates of application data from Internet resources, cache data, or execute long running downloads. But if you aren't connected to the Internet, or the connection is too slow to complete your download, why both waking the device to schedule the update at all?

You can use the [ConnectivityManager](#) ([//reference/android/net/ConnectivityManager.html](#)) to check that you're actually connected to the Internet, and if so, what type of connection is in place.

## Determine if You Have an Internet Connection

There's no need to schedule an update based on an Internet resource if you aren't connected to the Internet. The following snippet shows how to use the [ConnectivityManager](#) ([//reference/android/net/ConnectivityManager.html](#)) to query the active network and determine if it has Internet connectivity.

```
ConnectivityManager cm =  
    (ConnectivityManager) context.getSystemService(Context.CONNECTIVITY_SERVICE);  
  
NetworkInfo activeNetwork = cm.getActiveNetworkInfo();  
boolean isConnected = activeNetwork != null &&  
    activeNetwork.isConnectedOrConnecting();
```

### THIS LESSON TEACHES YOU TO

1. [Determine if you Have an Internet Connection](#)
2. [Determine the Type of your Internet Connection](#)
3. [Monitor for Changes in Connectivity](#)

### YOU SHOULD ALSO READ

- [Intents and Intent Filters](#)

## Determine the Type of your Internet Connection

It's also possible to determine the type of Internet connection currently available.

Device connectivity can be provided by mobile data, WiMAX, Wi-Fi, and ethernet connections. By querying the type of the active network, as shown below, you can alter your refresh rate based on the bandwidth available.

```
boolean isWiFi = activeNetwork.getType() == ConnectivityManager.TYPE_WIFI;
```

Mobile data costs tend to be significantly higher than Wi-Fi, so in most cases, your app's update rate should be lower when on mobile connections. Similarly, downloads of significant size should be suspended until you have a Wi-Fi connection.

Having disabled your updates, it's important that you listen for changes in connectivity in order to resume them once an Internet connection has been established.

## Monitor for Changes in Connectivity

The [ConnectivityManager](#) ([//reference/android/net/ConnectivityManager.html](#)) broadcasts the [CONNECTIVITY\\_ACTION](#) ([//reference/android/net/ConnectivityManager.html#CONNECTIVITY\\_ACTION](#)) ("android.net.conn.CONNECTIVITY\_CHANGE") action whenever the connectivity details have changed.

You can register a broadcast receiver in your manifest to listen for these changes and resume (or suspend) your background updates accordingly.

```
<action android:name="android.net.conn.CONNECTIVITY_CHANGE"/>
```

Changes to a device's connectivity can be very frequent—this broadcast is triggered every time you move between mobile data and Wi-Fi. As a result, it's good practice to monitor this broadcast only when you've previously suspended updates or downloads in order to resume them. It's generally sufficient to simply check for Internet connectivity before beginning an update and, should there be none, suspend further updates until connectivity is restored.

This technique requires toggling broadcast receivers you've declared in the manifest, which is described in the next lesson.