



## Create an Application to Manage Test Location Providers

In this example, you'll create a new project to set up the emulator to simplify testing other location-based applications. Running this project will ensure that the GPS provider is active and updating regularly.

1. Create a new Android project, TestProviderController, which includes a TestProviderController Activity.

```
package com.paad.testprovidercontroller;
import java.util.List;
import android.app.Activity;
import android.content.Context;
import android.location.Criteria;
import android.location.Location;
import android.location.LocationManager;
import android.location.LocationListener;
import android.location.LocationProvider;
import android.os.Bundle;
import android.widget.TextView;
public class TestProviderController extends Activity {
    @Override
    public void onCreate(Bundle icle) {
        super.onCreate(icle);
        setContentView(R.layout.main);
    }
}
```

2. Add an instance variable to store a reference to the LocationManager, then get that reference to it from within the onCreate method. Add stubs for creating a new test provider and to enable the GPS provider for testing.

```
LocationManager locationManager;
@Override
public void onCreate(Bundle icle) {
    super.onCreate(icle);
    setContentView(R.layout.main);
    String location_context = Context.LOCATION_SERVICE;
    locationManager = (LocationManager) getSystemService(location_context);
    testProviders();
}
public void testProviders() {}
```

3. Add a FINE\_LOCATION permission to test the providers.

```
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
```

4. Update the testProviders method to check the enabled status of each provider and return the last known location; also request periodic updates for each provider to force Android to start updating the locations for other applications. The methods used here are presented without comment; you'll learn more about how to use each of them in the remainder of this chapter.

```
public void testProviders() {
    TextView tv = (TextView)findViewById(R.id.myTextView);
    StringBuilder sb = new StringBuilder("Enabled Providers:");
    List<String> providers = locationManager.getProviders(true);
    for (String provider : providers) {
        locationManager.requestLocationUpdates(provider, 1000, 0,
        new LocationListener() {
            public void onLocationChanged(Location location) {}
            public void onProviderDisabled(String provider){}

            public void onProviderEnabled(String provider){}
            public void onStatusChanged(String provider, int status,
            Bundle extras){}
        });
        sb.append("\n").append(provider).append(": ");
        Location location = locationManager.getLastKnownLocation(provider);
        if (location != null) {
            double lat = location.getLatitude();
            double lng = location.getLongitude();
```

```

sb.append(lat).append(", ").append(lng);
} else {
sb.append("No Location");
}
}
tv.setText(sb);
}

```

5. The final step before you run the application is to update the main.xml layout resource to add an ID for the text label you're updating in Step 4.

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
android:orientation="vertical"
android:layout_width="fill_parent"
android:layout_height="fill_parent">
<TextView
android:id="@+id/myTextView"
android:layout_width="fill_parent"
android:layout_height="wrap_content"
android:text="@string/hello"
/>
</LinearLayout>

```

6. Run your application, and it should appear as shown in Figure 7-2.

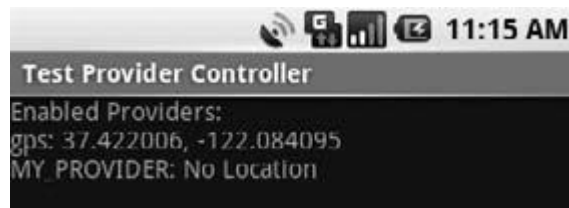


Figure 7-2

7. Android will now update the last known position for any applications using location-based services. You can update the current location using the techniques described in the previous section.

The test provider controller application you just wrote needs to be restarted to reflect any changes in the current location. Below in this chapter, you'll learn how to request updates based on the elapsed time and distance traveled.