Resolving Intent Filter Collision

In the previous section, you learned that the `<intent-filter>` element defines how your activity can be invoked by another activity. What happens if another activity (in either the same or a separate application) has the same filter name? For example, suppose your application has another activity named `Activity3`, with the following entry in the `AndroidManifest.xml` file:

```xml
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="net.learn2develop.UsingIntent"
    android:versionCode="1"
    android:versionName="1.0">
    <application>
        <activity android:icon="@drawable/ic_launcher" android:label="@string/app_name">
            <intent-filter>
                <action android:name="android.intent.action.MAIN"/>
                <category android:name="android.intent.category.LAUNCHER"/>
            </intent-filter>
        </activity>
        <activity android:label="Second Activity" android:name=".SecondActivity">
            <intent-filter>
                <action android:name="net.learn2develop.SecondActivity" />
                <category android:name="android.intent.category.DEFAULT" />
            </intent-filter>
        </activity>
        <activity android:label="Third Activity" android:name=".ThirdActivity">
            <intent-filter>
                <action android:name="net.learn2develop.SecondActivity" />
                <category android:name="android.intent.category.DEFAULT" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```

If you call the `startActivity()` method with the following intent, then the Android OS will display a selection of activities, as shown in Figure 2-12:

```java
startActivity(new Intent("net.learn2develop.SecondActivity"));
```
If you check the “Use by default for this action” item and then select an activity, then the next time the intent “net.learn2develop.SecondActivity” is called again, it will launch the previous activity that you have selected.

To clear this default, go to the Settings application in Android and select Apps ➪ Manage applications, and then select the application name (see Figure 2-13). When the details of the application are shown, scroll down to the bottom and click the Clear defaults button.

**Returning Results from an Intent**

The `startActivity()` method invokes another activity but does not return a result to the current activity. For example, you may have an activity that prompts the user for user name and password. The information entered by the user in that activity needs to be passed back to the calling activity for further processing. If you need to pass data back from an activity, you should instead use the `startActivityForResult()` method. The following Try It Out demonstrates this.

**TRY IT OUT  Obtaining a Result from an Activity**

1. Using the same project from the previous section, add the following statements in bold to the `secondactivity.xml` file:

```xml
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical">
    <TextView android:text="Obtain Result from Activity" />
</LinearLayout>
```

www.it-ebooks.info
CHAPTER 2  ACTIVITIES, FRAGMENTS, AND INTENTS

android:orientation="vertical">

<TextView
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="This is the Second Activity!"/>

<TextView
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="Please enter your name"/>

<EditText
    android:id="@+id/txt_username"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content" />

<Button
    android:id="@+id/btn_OK"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:onClick="onClick"/>

</LinearLayout>

2. Add the following statements in bold to SecondActivity.java:

```
package net.learn2develop.UsingIntent;

import android.app.Activity;
import android.content.Intent;
import android.net.Uri;
import android.os.Bundle;
import android.view.View;
import android.widget.EditText;

public class SecondActivity extends Activity{
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.secondactivity);
    }

    public void onClick(View view) {
        Intent data = new Intent();

        //---get the EditText view---
        EditText txt_username = (EditText) findViewById(R.id.txt_username);

        //---set the data to pass back---
        data.setData(Uri.parse("www.it-ebooks.info");
    }
}
```
3. Add the following statements in bold to the UsingIntentActivity.java file:

```java
package net.learn2develop.UsingIntent;
import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Toast;

public class UsingIntentActivity extends Activity {
    int request_Code = 1;

    /** Called when the activity is first created. **/
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
    }

    public void onClick(View view) {
        //startActivity(new Intent("net.learn2develop.SecondActivity"));
        //or
        //startActivity(new Intent(this, SecondActivity.class));
        startActivityForResult(new Intent("net.learn2develop.SecondActivity"), request_Code);
    }

    public void onActivityResult(int requestCode, int resultCode, Intent data) {
        if (requestCode == request_Code) {
            if (resultCode == RESULT_OK) {
                Toast.makeText(this, data.getData().toString(), Toast.LENGTH_SHORT).show();
            }
        }
    }
}
```

4. Press F11 to debug the application on the Android emulator. When the first activity is loaded, click the button. SecondActivity will now be loaded. Enter your name (see Figure 2-14) and click the OK button. The first activity will display the name you have entered using the Toast class.
How It Works

To call an activity and wait for a result to be returned from it, you need to use the `startActivityForResult()` method, like this:

```java
startActivityForResult(new Intent("net.learn2develop.SecondActivity"), request_Code);
```

In addition to passing in an `Intent` object, you need to pass in a request code as well. The request code is simply an integer value that identifies an activity you are calling. This is needed because when an activity returns a value, you must have a way to identify it. For example, you may be calling multiple activities at the same time, and some activities may not return immediately (for example, waiting for a reply from a server). When an activity returns, you need this request code to determine which activity is actually returned.

**NOTE** If the request code is set to -1, then calling it using the `startActivityForResult()` method is equivalent to calling it using the `startActivity()` method. That is, no result will be returned.

In order for an activity to return a value to the calling activity, you use an `Intent` object to send data back via the `setData()` method:

```java
Intent data = new Intent();
```

```java
//---get the EditText view---
```
EditText txt_username = (EditText) findViewById(R.id.txt_username);

//---get the data to pass back---
data.setData(Uri.parse(txt_username.getText().toString()));
setResult(RESULT_OK, data);

//--closes the activity---
finish();

The setResult() method sets a result code (either RESULT_OK or RESULT_CANCELED) and the data (an Intent object) to be returned back to the calling activity. The finish() method closes the activity and returns control back to the calling activity.

In the calling activity, you need to implement the onActivityResult() method, which is called whenever an activity returns:

```java
public void onActivityResult(int requestCode, int resultCode, Intent data) {
    if (requestCode == request_Code) {
        if (resultCode == RESULT_OK) {
            Toast.makeText(this, data.getData().toString(), Toast.LENGTH_SHORT).show();
        }
    }
}
```

Here, you check for the appropriate request and result codes and display the result that is returned. The returned result is passed in via the data argument; and you obtain its details through the getData() method.

### Passing Data Using an Intent Object

Besides returning data from an activity, it is also common to pass data to an activity. For example, in the previous example you may want to set some default text in the EditText view before the activity is displayed. In this case, you can use the Intent object to pass the data to the target activity.

The following Try It Out shows you the various ways in which you can pass data between activities.

### TRY IT OUT | Passing Data to the Target Activity

1. Using Eclipse, create a new Android project and name it PassingData.
2. Add the following statements in bold to the main.xml file:

```xml
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
android:layout_width="fill_parent"
android:layout_height="fill_parent"

www.it-ebooks.info
```
3. Add a new XML file to the res/layout folder and name it secondactivity.xml. Populate it as follows:

```xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical">
    <TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="Welcome to Second Activity"/>
    <Button
        android:id="@+id/btn_MainActivity"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="Click to return to main activity"
        android:onClick="onClick"/>
</LinearLayout>
```

4. Add a new Class file to the package and name it SecondActivity. Populate the SecondActivity.java file as follows:

```java
package net.learn2develop.PassingData;
import android.app.Activity;
import android.content.Intent;
import android.net.Uri;
import android.os.Bundle;
import android.view.View;
import android.widget.Toast;

public class SecondActivity extends Activity {
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.secondactivity);

        //---get the data passed in using getStringExtra()---
        Toast.makeText(this, getIntent().getStringExtra("str1"), Toast.LENGTH_SHORT).show();

        //---get the data passed in using getIntExtra()---
    }
}
```
5. Add the following statements in bold to the AndroidManifest.xml file:

```xml
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="net.learn2develop.PassingData"
    android:versionCode="1"
    android:versionName="1.0">

    <uses-sdk android:minSdkVersion="14" />

    <application>
        <activity>
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```

www.it-ebooks.info
6. Add the following statements in bold to the `PassingDataActivity.java` file:

```java
package net.learn2develop.PassingData;
import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Toast;
public class PassingDataActivity extends Activity {
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
    }
    public void onClick(View view) {
        Intent i = new Intent("net.learn2develop.PassingDataSecondActivity");
        //---use putExtra() to add new name/value pairs---
        i.putExtra("str1", "This is a string");
        i.putExtra("age1", 25);
        //---use a Bundle object to add new name/values---
        Bundle extras = new Bundle();
        extras.putString("str2", "This is another string");
        extras.putInt("age2", 35);
        //---attach the Bundle object to the Intent object---
        i.putExtras(extras);
        //---start the activity to get a result back---
        startActivityForResult(i, 1);
    }
    public void onActivityResult(int requestCode, int resultCode, Intent data) {
        //---check if the request code is 1---
        if (requestCode == 1) {
            //---if the result is OK---
        }
    }
}
```
if (resultCode == RESULT_OK) {

    //---get the result using getIntExtra()---
    Toast.makeText(this, data.getIntExtra("age3", 0),
    Toast.LENGTH_SHORT).show();

    //---get the result using getData()---
    Toast.makeText(this, data.getData().toString(),
    Toast.LENGTH_SHORT).show();
}
}

7. Press F11 to debug the application on the Android emulator. Click the button on each activity and observe the values displayed.

How It Works

While this application is not visually exciting, it does illustrate some important ways to pass data between activities.

First, you can use the putExtra() method of an Intent object to add a name/value pair:

    //---use putExtra() to add new name/value pairs---
    i.putExtra("str1", "This is a string");
    i.putExtra("age1", 25);

The preceding statements add two name/value pairs to the Intent object: one of type string and one of type integer.

Besides using the putExtra() method, you can also create a Bundle object and then attach it using the putExtras() method. Think of a Bundle object as a dictionary object — it contains a set of name/value pairs. The following statements create a Bundle object and then add two name/value pairs to it. It is then attached to the Intent object:

    //---use a Bundle object to add new name/values pairs---
    Bundle extras = new Bundle();
    extras.putString("str2", "This is another string");
    extras.putInt("age2", 35);

    //---attach the Bundle object to the Intent object---
    i.putExtras(extras);

On the second activity, to obtain the data sent using the Intent object, you first obtain the Intent object using the getIntent() method. Then, call its getStringExtra() method to get the string value set using the putExtra() method:

    //---get the data passed in using getStringExtra()---
    Toast.makeText(this, getIntent().getStringExtra("str1"),
    Toast.LENGTH_SHORT).show();
In this case, you have to call the appropriate method to extract the name/value pair based on the type of data set. For the integer value, use the `getIntExtra()` method (the second argument is the default value in case no value is stored in the specified name):

```java
//---get the data passed in using getIntExtra()---
Toast.makeText(this, Integer.toString(getIntent().getIntExtra("age1", 0)), Toast.LENGTH_SHORT).show();
```

To retrieve the Bundle object, use the `getExtras()` method:

```java
//---get the Bundle object passed in---
Bundle bundle = getIntent().getExtras();
```

To get the individual name/value pairs, use the appropriate method. For the string value, use the `getString()` method:

```java
//---get the data using the getString()---
Toast.makeText(this, bundle.getString("str2"), Toast.LENGTH_SHORT).show();
```

Likewise, use the `getInt()` method to retrieve an integer value:

```java
//---get the data using the getInt() method---
Toast.makeText(this, Integer.toString(bundle.getInt("age2")), Toast.LENGTH_SHORT).show();
```

Another way to pass data to an activity is to use the `setData()` method (as used in the previous section), like this:

```java
//---use the setData() method to return some value---
i.setData(Uri.parse("Something passed back to main activity"));
```

Usually, you use the `setData()` method to set the data on which an Intent object is going to operate (such as passing a URL to an Intent object so that it can invoke a web browser to view a web page; see the section “Calling Built-In Applications Using Intents” later in this chapter for more examples).

To retrieve the data set using the `setData()` method, use the `getData()` method (in this example data is an Intent object):

```java
//---get the result using getData()---
Toast.makeText(this, data.getData().toString(), Toast.LENGTH_SHORT).show();
```