Chapter 9: Customize!
Navigating with Tabs on a Tablet App
Objectives

In this chapter, you learn to:

• Create an Android tablet project using a tab layout
• Code an XML layout with a TabHost control
• Display a TabWidget and FrameLayout within a TabHost
• Customize a GridView XML layout
• Develop a user interface that displays images in a GridView control
• Extend a TabActivity class
Objectives (continued)

- Display multiple classes as content within a tab layout
- Customize the ImageAdapter class for a GridView layout
- Open an Android Web browser in a tablet
- Customize a tab specification with TabSpec
- Add a TabSpec to a TabHost
Navigating with Tabs

• Tab interfaces have replaced traditional drop-down menus
• Tabs are used to:
  – Show different topics within a single window
  – Simplify Layout
  – Eliminate the need for additional instructions

Figures 9-1, 9-2, and 9-3 Bike and Barge Android tablet app, Second Tab displays tour information, Third tab opens a Web site
Navigating with Tabs (continued)

• Steps to complete the App:
  1. Create a TabHost within the XML code that includes the TabWidget and FrameLayout.
  2. Extend the TabActivity class.
  3. Add the tab and GridView images needed for the project.
  4. Create three different XML layouts for each of the tabs.
      a. Create the XML layout for the first tab to display a GridView control.
      b. Create the XML layout for the second tab to display TextView information.
      c. Create the XML layout for the third tab to display a Web site in a browser.
  5. Create three different Activities; one for each tab.
      a. Code the first tab to display a GridView control.
      b. Code the second tab to display TextView information.
      c. Code the third tab to display a Web site in a browser.
  6. Code the Main Activity to specify the TabSpec and launch the tabs.
Creating a Tab Layout for a Tablet

– An Android SDK provides a TabHost control so you can wrap multiple views in a single window
– The TabHost contains two distinct parts:
  • A TabWidget for displaying the Tabs
  • A FrameLayout for displaying the Tab content
– Each tab launches a separate Activity class
– All tabs are launched within a single Activity class
Creating a Tab Layout for a Tablet (continued)

Figure 9-4 Anatomy of a TabHost control
The TabHost Layout

- TabHost requires a TabWidget and a FrameLayout within the XML layout code
- Use the default LinearLayout to position the TabWidget and the FrameLayout vertically
- TabHost, TabWidget, and FrameLayout must have ids assigned in the code

<table>
<thead>
<tr>
<th>Control</th>
<th>Android id</th>
</tr>
</thead>
<tbody>
<tr>
<td>TabHost</td>
<td>android:id=&quot;@+id/tabhost&quot;</td>
</tr>
<tr>
<td>TabWidget</td>
<td>android:id=&quot;@android:id/tabs&quot;</td>
</tr>
<tr>
<td>FrameLayout</td>
<td>android:id=&quot;@android:id/tabcontent&quot;</td>
</tr>
</tbody>
</table>

Table 9-1 Tab control Android id

Figure 9-6 TabHost XML code
The TabHost Layout (continued)

Figure 9-8 TabWidget and FrameLayout within the TabHost
The TabHost Layout (continued)

Figure 9-9 Graphical layout of TabHost
The TabHost Layout  (continued)

• Extending the TabActivity Class
  – A class named TabActivity makes it simple to display tabs within the app

![Main.java code snippet]

Figure 9-10 Main extends TabActivity
The TabHost Layout (continued)

• Adding the Tab and GridView Images
  – Drag image files to the drawable-hdpi folder until a plus sign pointer appears
  – Tab icons should be simple, flat shapes rather than 3D images
    • Tab icons are typically sized at 72 X 72 pixels
• Creating a GridView XML Layout for the First Tab
  – A GridView control is similar to the Gallery control
  – Displays objects in a two-dimensional, scrollable grid
  • You specify:
    – The number of columns
    – The width of each column
    – The space between columns
The TabHost Layout (continued)

Figure 9-13 Custom GridView layout
The TabHost Layout (continued)

Figure 9-14 First TextView control

Figure 9-15 Second TextView control
The TabHost Layout (continued)

Figure 9-16 Complete code for tab2.xml
The TabHost Layout (continued)

- Creating the XML Layout for the Third Tab

Figure 9-17  tab3.xml
Coding the GridView Activity for the First Tab

- The First Tab of the GridView starts by displaying images
- Other two tabs available in the TabHost control
- This app requires four Java code Activity files
  - Main.Java, Tab1.Java, Tab2.java, Tab3.java
Coding the GridView Activity for the First Tab (continued)

Figure 9-18 Tab1.java class opens the tab1.xml layout

```java
package net.androidbootcamp.bikeandbarge;
import android.app.Activity;

public class Tab1 extends Activity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        // TODO Auto-generated method stub
        super.onCreate(savedInstanceState);
        setContentView(R.layout.tab1);
    }
}
```

Tab1 class is created

Figure 9-19 GridView control is instantiated

```java
    GridView g = (GridView) findViewById(R.id.photos);
```

Tab1.xml layout is displayed

Instance of GridView
Coding the GridView Activity for the First Tab (continued)

- **Using a setAdapter with an ImageAdapter**
  - `setAdapter` provides a data model for the layout
  - `ImageAdapter` binds the images to the GridView control

```java
package net.androidbootcamp.bikeandbarge;

import android.app.Activity;

public class Tab1 extends Activity {

  @Override
  protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.tab1);
    GridView g = (GridView) findViewById(R.id.photos);
    g.setAdapter(new ImageAdapter(this));
  }
}
```

Figure 9-20 Instance of the ImageAdapter class
Coding the GridView Activity for the First Tab (continued)

Figure 9-21 ImageAdapter class

Auto-generated ImageAdapter class
Coding the GridView Activity for the First Tab  (continued)

• Customizing the ImageAdapter class
  – The ImageAdapter class provides information to set up the GridView display
  – **Context** is used to load and access resources for the app

```java
package net.androidbootcamp.bikeandbarg;  // Net.ABC is the package name.

import android.app.Activity;

public class Tab1 extends Activity {
  @Override
  protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.tab1);
    GridView g = (GridView) findViewById(R.id.photos);
    g.setAdapter(new ImageAdapter(this));
  }

  public class ImageAdapter extends BaseAdapter {
    private Context context;
    // the context variable is initialized
    public ImageAdapter(Tab1 tab1) {
      // TODO Auto-generated constructor stub
    }
  }
}
```

Figure 9-22 Context variable
Coding the GridView Activity for the First Tab  (continued)

• **Coding the getView Method**
  – The `getView()` method uses Context to create a new `ImageView` instance that is responsible for temporarily holding each image in the GridView control.

![Code Snippet]

Figure 9-26 The getView method creates an instance of `ImageView`
Coding the GridView Activity for the First Tab  
(continued)

Figure 9-27 Complete code for Tab1.java
Coding the Second Tab Java File

Figure 9-28 Complete code for Tab2.java

```java
package net.androidbootcamp.bikeandbarge;

import android.app.Activity;

public class Tab2 extends Activity {
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.tab2);
    }
}
```
Coding the Third Tab Java File to Display a Web Site

Figure 9-30 Complete code for Tab3.java

```java
package net.androidbootcamp.bikeandbarge;
import android.app.Activity;

public class Tab3 extends Activity {
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.tab3);
        startActivity(new Intent(Intent.ACTION_VIEW, Uri.parse("http://bikebarge.com/")));
    }
}
```
Coding the TabHost

• Each tab must have a **TabSpec**, which specifies how the tab should appear
• The instance of the TabSpec calls the **setIndicator**, which sets the tab button caption and supplies an icon image
• The **setContent** method then indicates what is displayed in the tab content area
Coding the TabHost (continued)

Figure 9-31 First tab is displayed with the Photos contents

Figure 9-32 Second tab displays the Tour contents
Coding the TabHost (continued)

Figure 9-33 Third tab displays the Web Site contents
Coding the TabHost (continued)

- Adding the TabSpec to the TabHost

```java
package net.androidbootcamp.bikeandcharge;

import android.app.TabActivity;

public class Main extends TabActivity {
    // First tab
    TabSpec photospec = tabHost.newTabSpec("Photos");
    photospec.setIndicator("Photos", getResources().getDrawable(R.drawable.tab1));
    Intent photosIntent = new Intent(this, Tab1.class);
    photospec.setContent(photosIntent);
    // Second tab
    TabSpec tourspec = tabHost.newTabSpec("Tour");
    tourspec.setIndicator("Tour", getResources().getDrawable(R.drawable.tab2));
    Intent tourIntent = new Intent(this, Tab2.class);
    tourspec.setContent(tourIntent);
    // Third tab
    TabSpec webspec = tabHost.newTabSpec("Web Site");
    webspec.setIndicator("Web Site", getResources().getDrawable(R.drawable.tab3));
    Intent webIntent = new Intent(this, Tab3.class);
    webspec.setContent(webIntent);
    // Add TabSpec to TabHost
    tabHost.addTab(photospec);
    tabHost.addTab(tourspec);
    tabHost.addTab(webspec);
}
```

Figure 9-34 Complete code for Main.java
Updating the Android Manifest File

- Add references to all the tabs in the app

Figure 9-35 Activities added to the Android Manifest
Running and Testing the Application

• Click Run on the menu bar
• Select Android Application from the dialog box
• Save all the files when prompted
• Unlock the emulator (if necessary)
• Test each tab to ensure it works as intended
Summary

- Use a TabHost control to create a tabbed interface similar to one used in a browser. The TabWidget displays the tabs, including labels and icons, and the FrameLayout displays the tab content.
- Specify the layout of a tabbed interface in the XML code. LinearLayout positions the controls vertically (FrameLayout below the TabWidget).
- Design a tabbed control in the main Java file by extending the TabActivity class instead of the Activity.
- A GridView control displays objects in a two-dimensional, scrollable grid.
Summary (continued)

• A method to instantiates the custom ImageAdapter class
• ImageAdapter controls the layout of the GridView and connects the data sources from an array for display.
• Context is used to load and access resources for the application.
• To display the tabbed interface, you must fully code the Main.java class that extends TabActivity.
• Each tab in a tabbed interface must have a TabSpec statement, which creates an instance, that details tab content.

• The instance calls the setIndicator method to set the tab button caption and supplies an icon image. The setContent method indicates what is displayed in the tab content area.

• TabSpec must be added to the instance of the TabHost control and every tab change closes the previously opened tab and opens the selected tab.