Employing Fancy Widgets and Containers

The widgets and containers covered so far are not only found in many GUI toolkits (in one form or fashion), but also are widely used in building GUI applications—whether web-based, desktop, or mobile. The widgets and containers described in this chapter are a little less widely used, though you will likely find many to be quite useful.

Pick and Choose

With limited-input devices like phones, having widgets and dialogs that are aware of the type of stuff someone is supposed to be entering is very helpful. These elements minimize keystrokes and screen taps, as well as reduce the chance of making some sort of error (e.g., entering a letter somewhere only numbers are expected).

As shown in Chapter 5, EditText has content-aware flavors for entering numbers and text. Android also supports widgets (DatePicker and TimePicker) and dialogs (DatePickerDialog and TimePickerDialog) for helping users enter dates and times.

DatePicker and DatePickerDialog allow you to set the starting date for the selection, in the form of a year, month, and day of month value. Note that the month runs from 0 for January through 11 for December. Each lets you provide a callback object (OnDateChangedListener or OnDateSetListener) where you are informed of a new date selected by the user. It is up to you to store that date someplace, particularly if you are using the dialog, since there is no other way for you to get at the chosen date later.

Similarly, TimePicker and TimePickerDialog let you set the initial time the user can adjust, in the form of an hour (0 through 23) and a minute (0 through 59). You can indicate if the selection should be in 12-hour mode with an AM/PM toggle or in 24-hour mode (what in the United States is thought of as “military time” and in the rest of the world as “the way times are supposed to be”). You can also provide a callback object (OnTimeChangedListener or OnTimeSetListener) to be notified of when the user has chosen a new time, which is supplied to you in the form of an hour and minute.
For example, from the Fancy/Chrono sample project, here’s a trivial layout containing a label and two buttons, which will pop up the dialog flavors of the date and time pickers:

```xml
<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/dateAndTime"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content" />
    <Button android:id="@+id/dateBtn"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="Set the Date" />
    <Button android:id="@+id/timeBtn"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="Set the Time" />
</LinearLayout>
```

The more interesting stuff comes in the Java source:

```java
public class ChronoDemo extends Activity {
    DateFormat fmtDateAndTime=DateFormat.getDateTimeInstance();
    TextView dateAndTimeLabel;
    Calendar dateAndTime=Calendar.getInstance();
    DatePickerDialog.OnDateSetListener d=new DatePickerDialog.OnDateSetListener() {
        public void onDateSet(DatePicker view, int year, int monthOfYear,
            int dayOfMonth) {
            dateAndTime.set(Calendar.YEAR, year);
            dateAndTime.set(Calendar.MONTH, monthOfYear);
            dateAndTime.set(Calendar.DAY_OF_MONTH, dayOfMonth);
            updateLabel();
        }
    };
    TimePickerDialog.OnTimeSetListener t=new TimePickerDialog.OnTimeSetListener() {
        public void onTimeSet(TimePicker view, int hourOfDay,
            int minute) {
            dateAndTime.set(Calendar.HOUR_OF_DAY, hourOfDay);
            dateAndTime.set(Calendar.MINUTE, minute);
            updateLabel();
        }
    };
    @Override
    public void onCreate(Bundle icicle) {
        super.onCreate(icicle);
        setContentView(R.layout.main);
        Button btn=(Button)findViewById(R.id.dateBtn);
    }
}
```