List of Articles
Avoiding Memory Leaks
Backward Compatibility
Can I Use this Intent?
Creating an Input Method
Drawable Mutations
Faster Screen Orientation Change
Future-Proofing Your Apps
Gestures
Introducing GLSurfaceView
Layout Tricks: Reusing
Layout Tricks: Efficiency
Layout Tricks: ViewStubs
Layout Tricks: Merging
ListView Backgrounds
Live Folders
Onscreen Input Methods
Painless Threading
Quick Search Box
Touch Mode
Tracking Memory Allocations
UI Framework Changes in Android 1.5
UI Framework Changes in Android 1.6
Updating the UI from a Timer
Using Text-to-Speech
Using WebViews
WikiNotes: Linkify your Text!
WikiNotes: Routing Intents

Updating the UI from a Timer

Background: While developing my first useful (though small) application for Android, which was a port of an existing utility I use when podcasting, I needed a way of updating a clock displayed on the UI at regular intervals, but in a lightweight and CPU efficient way.

Problem: In the original application I used java.util.Timer to update the clock, but that class is not such a good choice on Android. Using a Timer introduces a new thread into the application for a relatively minor reason. Thinking in terms of mobile applications often means re-considering choices that you might make differently for a desktop application with relatively richer resources at its disposal. We would like to find a more efficient way of updating that clock.

The Application: The original application is a Java Swing and SE application. It is like a