

Working in the Background

Because of the limited screen size of most mobile devices, typically only one application is visible and active on a device at any given time. This offers a perfect environment for applications that run in the background without a User Interface — responding to events, polling for data, or updating Content Providers.

Android offers the `Service` class to create application components specifically to handle operations and functionality that should run silently, without a User Interface. Android accords Services a higher priority than inactive Activities, so they're less likely to be killed when the system requires resources. In fact, should the run time prematurely terminate a Service that's been started, it will be restarted as soon as sufficient resources are available. By using Services, you can ensure that your applications continue to run and respond to events, even when they're not in active use.

Services run without a dedicated GUI, but, like Activities and Broadcast Receivers, they still execute in the main thread of the application's process. To help keep your applications responsive, you'll learn to move time-consuming processes (like network lookups) into background threads.

Android offers several techniques for application components (particularly Services) to communicate with users without an Activity providing a direct User Interface. In this chapter, you'll learn how to use Notifications and Toasts to politely alert and update users, without interrupting the active application.

Toasts are a transient, non-modal Dialog-box mechanism used to display information to users without stealing focus from the active application. You'll learn to display Toasts from any application component to send unobtrusive on-screen messages to your users.

Where Toasts are silent and transient, *Notifications* represent a more robust mechanism for alerting users. For many users, when they're not actively using their mobile phones, they sit silent and unwatched in a pocket or on a desk until it rings, vibrates, or flashes. Should a user miss these alerts, status bar icons are used to indicate that an event has occurred. All of these attention-grabbing antics are available within Android as Notifications.

Alarms provide a mechanism for firing Intents at set times, outside the control of your application life cycle. You'll learn to use Alarms to start Services, open Activities, or broadcast Intents based on either the clock time or the time elapsed since device boot. An Alarm will fire even after its owner application has been closed, and can (if required) wake a device from sleep.