

Notifications

Several types of situations may arise that require you to notify the user about an event that occurs in your application. Some events require the user to respond and others do not. For example:

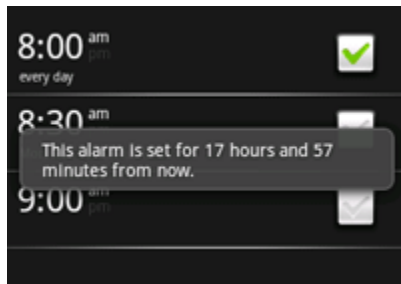
- When an event such as saving a file is complete, a small message should appear to confirm that the save was successful.
- If the application is running in the background and needs the user's attention, the application should create a notification that allows the user to respond at his or her convenience.
- If the application is performing work that the user must wait for (such as loading a file), the application should show a hovering progress wheel or bar.

Each of these notification tasks can be achieved using a different technique:

- A [Toast Notification](#), for brief messages that come from the background.
- A [Status Bar Notification](#), for persistent reminders that come from the background and request the user's response.
- A [Dialog Notification](#), for Activity-related notifications.

This document summarizes each of these techniques for notifying the user and includes links to full documentation.

Toast Notification

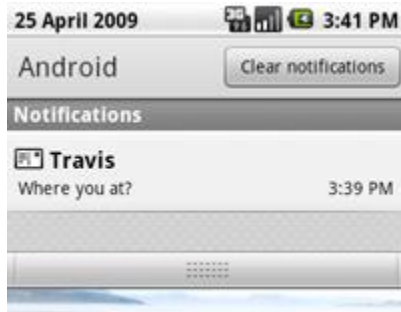


A toast notification is a message that pops up on the surface of the window. It only fills the amount of space required for the message and the user's current activity remains visible and interactive. The notification automatically fades in and out, and does not accept interaction events. Because a toast can be created from a background [Service](#), it appears even if the application isn't visible.

A toast is best for short text messages, such as "File saved," when you're fairly certain the user is paying attention to the screen. A toast can not accept user interaction events; if you'd like the user to respond and take action, consider using a [Status Bar Notification](#) instead.

For more information, refer to [Toast Notifications](#).

Status Bar Notification

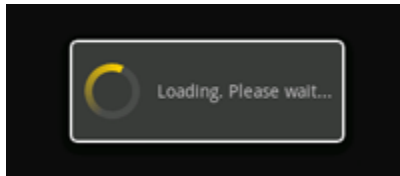


A status bar notification adds an icon to the system's status bar (with an optional ticker-text message) and an expanded message in the "Notifications" window. When the user selects the expanded message, Android fires an [Intent](#) that is defined by the notification (usually to launch an [Activity](#)). You can also configure the notification to alert the user with a sound, a vibration, and flashing lights on the device.

This kind of notification is ideal when your application is working in a background [Service](#) and needs to notify the user about an event. If you need to alert the user about an event that occurs while your Activity is still in focus, consider using a [Dialog Notification](#) instead.

For more information, refer to [Status Bar Notifications](#).

Dialog Notification



A dialog is usually a small window that appears in front of the current Activity. The underlying Activity loses focus and the dialog accepts all user interaction. Dialogs are normally used for notifications and short activities that directly relate to the application in progress.

You should use a dialog when you need to show a progress bar or a short message that requires confirmation from the user (such as an alert with "OK" and "Cancel" buttons). You can also use dialogs as integral components in your application's UI and for other purposes besides notifications. For a complete discussion on all the available types of dialogs, including its uses for notifications, refer to [Dialogs](#).