

# 10

## Android Jelly Bean

Android 4.1, also known as Android Jelly Bean, was released by Google in July 2012.

The topics covered in this chapter are as follows:

- Android Jelly Bean
- New APIs in Android Jelly Bean

### What is Android Jelly Bean

Android Jelly Bean is an optimized version of Android in which performance improvements are made and touch latency is decreased. This version is a minor update for developers. However, there are important updates for Android users. The important changes for the users are listed as follows:

- Better performance and low touch latency with vsync timing, triple buffering, and synchronizing touch.
- **Google Now** is a new feature introduced with Android Jelly Bean. It gives information about your current time. For instance, it gives information about traffic, it gives live scores about your favorite team, and it gives information about the weather conditions. It also suggests bars and restaurants that are near you.
- Search experience is increased with knowledge graph. This means search results are shown in a richer way.
- Android Beam experience is increased.
- Provides a better and smarter keyboard.
- Viewing photos is faster and easier with Android Jelly Bean.

- Personalizing the home screen become easier and widgets can resize themselves.
- Expandable and more interactive notifications.
- Offline voice typing.

Although it is a minor update for developers, there are some important changes for developers, too. In the following sections, we will see those important changes.

## Expandable and richer notifications

Ability to block notifications for a specific application is introduced for users with Jelly Bean. Thus, developers should use notifications properly and shouldn't disturb the user.

### Style

The notification style could be set with the `setStyle()` method. There are three options for the style:

- **Notification.BigPictureStyle:** This style is for the notifications that contain large images
- **Notification.BigTextStyle:** This style is for notifications that contain a lot of text
- **Notification.InboxStyle:** This style is for notifications that contain a list of strings

### Actions

Actions could be added to notifications with the `addAction()` method. Actions are shown as a button.

### Priorities

The priority of notifications could be set with the `setPriority()` method. There are five priorities that can be set - `PRIORITY_MAX`, `PRIORITY_HIGH`, `PRIORITY_DEFAULT`, `PRIORITY_LOW`, and `PRIORITY_MIN`.

## Live wallpaper preview

Using the `ACTION_CHANGE_LIVE_WALLPAPER` intent, users can preview the wallpaper without leaving their application.

---

## New permissions

New permissions are introduced for the Android applications, listed as follows:

- `READ_EXTERNAL_STORAGE`: This permission is for the read access to external storage
- `READ_USER_DICTIONARY`: This permission is for the read access to the user dictionary
- `READ_CALL_LOG`: This permission is for the read access to system logs of incoming and outgoing calls
- `WRITE_CALL_LOG`: This permission is for the write access to system logs
- `WRITE_USER_DICTIONARY`: This permission is for the write access to the user dictionary

## New device feature

A new device feature has been introduced with Android Jelly Bean called `FEATURE_TELEVISION`. If your application requires a TV interface, you can request this requirement with the `<uses-feature>` element in the `AndroidManifest.xml` file as in the following code snippet:

```
<uses-feature android:name="android.hardware.type.television"
              android:required="true" />
```

## New flags for system UI element's visibility

New flags that control the visibility of system UI elements with `setSystemUiVisibility()` method are introduced with Android Jelly Bean. These flags are as follows:

- `SYSTEM_UI_FLAG_FULLSCREEN`
- `SYSTEM_UI_FLAG_LAYOUT_FULLSCREEN`
- `SYSTEM_UI_FLAG_LAYOUT_HIDE_NAVIGATION`
- `SYSTEM_UI_FLAG_LAYOUT_STABLE`

## Connectivity

- **Android Beam:** It is now possible to send large payloads with Android Beam using the `setBeamPushUri()` method and the `NfcAdapter.CreateBeamUriCallback` callback. Bluetooth is used for transferring such payloads.
- **Wi-Fi Direct Service Discovery:** With this feature, a device can search for nearby Wi-Fi Direct devices and can find the services that the nearby Wi-Fi Direct device offers before connecting to them. New methods are added to the `WifiP2pManager` class such as `addLocalService()`, `addServiceRequest()`, and `discoverServices()`.
- **Wi-Fi Network Service Discovery:** With this feature, a device can search nearby devices over the Wi-Fi network and can find the services that the nearby devices offer before connecting to them.

## App stack navigation

With this feature, you can define the parent activity of an activity. In order to define the parent activity, you should declare it with `android:parentActivityName` in the `<activity>` element. Parent activity is used when the user presses the Up button. The system opens the appropriate activity with this information. This feature is also used in scenarios like users entering their application from a notification. In this case, there is no back stack. The system can create a synthetic back stack using the `onPrepareNavigateUpTaskStack()` method of the `Activity` class. This method takes the `TaskStackBuilder` object as a parameter, which is introduced with Android Jelly Bean.

## Multimedia

The `MediaCodec` class is added to APIs with Android Jelly Bean. This class is used to access a low level media codec.

## Summary

In this chapter, you learned the new features of Android Jelly Bean for users. Then, you learned some of the new APIs introduced with Android Jelly Bean, as well as a few packages that are added with Android Jelly Bean. Besides this, we also explored some new methods, flags, and constants which are also added to classes.