

Activity

`public class Activity`

`extends ContextThemeWrapper`

(<https://developer.android.com/reference/android/view/ContextThemeWrapper.html>) implements

`LayoutInflater.Factory2` ([https://developer.android.com/reference/android](https://developer.android.com/reference/android/view/LayoutInflater.Factory2.html)

[/view/LayoutInflater.Factory2.html](https://developer.android.com/reference/android/view/LayoutInflater.Factory2.html)), `Window.Callback` ([https://developer.android.com/reference](https://developer.android.com/reference/android/view/Window.Callback.html)

[/android/view/Window.Callback.html](https://developer.android.com/reference/android/view/Window.Callback.html)), `KeyEvent.Callback` ([https://developer.android.com/reference](https://developer.android.com/reference/android/view/KeyEvent.Callback.html)

[/android/view/KeyEvent.Callback.html](https://developer.android.com/reference/android/view/KeyEvent.Callback.html)), `View.OnCreateContextMenuListener`

(<https://developer.android.com/reference/android/view/View.OnCreateContextMenuListener.html>),

`ComponentCallbacks2` ([https://developer.android.com/reference/android/content](https://developer.android.com/reference/android/content/ComponentCallbacks2.html)

[/ComponentCallbacks2.html](https://developer.android.com/reference/android/content/ComponentCallbacks2.html))

`java.lang.Object` (<https://developer.android.com/reference/java/lang/Object.html>)

↳ `android.content.Context` (<https://developer.android.com/reference/android/content/Context.html>)

↳ `android.content.ContextWrapper` (<https://developer.android.com/reference/android/content/ContextWrapper.html>)

↳ `android.view.ContextThemeWrapper` (<https://developer.android.com/reference/android/view/ContextThemeWrapper.html>)

↳ `android.app.Activity`

▼ (#)Known Direct Subclasses

`AccountAuthenticatorActivity` ([https://developer.android.com/reference/android/accounts](https://developer.android.com/reference/android/accounts/AccountAuthenticatorActivity.html)

[/AccountAuthenticatorActivity.html](https://developer.android.com/reference/android/accounts/AccountAuthenticatorActivity.html)), `ActivityGroup` ([https://developer.android.com/reference/android](https://developer.android.com/reference/android/app/ActivityGroup.html)

[/app/ActivityGroup.html](https://developer.android.com/reference/android/app/ActivityGroup.html)), `AliasActivity` (<https://developer.android.com/reference/android/app/AliasActivity.html>),

`ExpandableListActivity` (<https://developer.android.com/reference/android/app/ExpandableListActivity.html>),

`FragmentActivity` (<https://developer.android.com/reference/android/support/v4/app/FragmentActivity.html>),

`ListActivity` (<https://developer.android.com/reference/android/app/ListActivity.html>), `NativeActivity`

(<https://developer.android.com/reference/android/app/NativeActivity.html>)

▼ (#)Known Indirect Subclasses

`ActionBarActivity` (<https://developer.android.com/reference/android/support/v7/app/ActionBarActivity.html>),

`AppCompatActivity` (<https://developer.android.com/reference/android/support/v7/app/AppCompatActivity.html>),

`LauncherActivity` (<https://developer.android.com/reference/android/app/LauncherActivity.html>), `PreferenceActivity`

(<https://developer.android.com/reference/android/preference/PreferenceActivity.html>), `TabActivity`

(<https://developer.android.com/reference/android/app/TabActivity.html>)

Added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

Summary: [Constants \(#constants\)](#) | [Inherited Constants \(#inhconstants\)](#) | [Fields \(#lfields\)](#) | [Ctors \(#pubctors\)](#) | [Methods \(#pubmethods\)](#) | [Protected Methods \(#promethods\)](#) | [Inherited Methods \(#inhmethods\)](#) | [\[Expand All\]](#) (#)

An activity is a single, focused thing that the user can do. Almost all activities interact with the user, so the Activity class takes care of creating a window for you in which you can place your UI with `setContentView(View)` ([https://developer.android.com/reference/android/app/Activity.html#setContentView\(android.view.View\)](https://developer.android.com/reference/android/app/Activity.html#setContentView(android.view.View))). While activities are often presented to the user as full-screen windows, they can also be used in other ways: as floating windows (via a theme with `windowIsFloating` (<https://developer.android.com/reference/android/R.attr.html#windowIsFloating>) set) or embedded inside of another activity (using `ActivityGroup` (<https://developer.android.com/reference/android/app/ActivityGroup.html>)). There are two methods almost all subclasses of Activity will implement:

- `onCreate(Bundle)` ([https://developer.android.com/reference/android/app/Activity.html#onCreate\(android.os.Bundle\)](https://developer.android.com/reference/android/app/Activity.html#onCreate(android.os.Bundle))) is where you initialize your activity. Most importantly, here you will usually call `setContentView(int)` ([https://developer.android.com/reference/android/app/Activity.html#setContentView\(int\)](https://developer.android.com/reference/android/app/Activity.html#setContentView(int))) with a layout resource defining your UI, and using `findViewById(int)` ([https://developer.android.com/reference/android/app/Activity.html#findViewById\(int\)](https://developer.android.com/reference/android/app/Activity.html#findViewById(int))) to retrieve the widgets in that UI that you need to interact with programmatically.
- `onPause()` ([https://developer.android.com/reference/android/app/Activity.html#onPause\(\)](https://developer.android.com/reference/android/app/Activity.html#onPause())) is where you deal with the user leaving your activity. Most importantly, any changes made by the user should at this point be committed (usually to the `ContentProvider` (<https://developer.android.com/reference/android/content/ContentProvider.html>) holding the data).

To be of use with `Context.startActivity()` ([https://developer.android.com/reference/android/content/Context.html#startActivity\(android.content.Intent\)](https://developer.android.com/reference/android/content/Context.html#startActivity(android.content.Intent))), all activity classes must have a corresponding `<activity>` (<https://developer.android.com/reference/android/R.styleable.html#AndroidManifestActivity>) declaration in their package's `AndroidManifest.xml`.

Topics covered here:

1. Fragments (#Fragments)
2. Activity Lifecycle (#ActivityLifecycle)
3. Configuration Changes (#ConfigurationChanges)
4. Starting Activities and Getting Results (#StartingActivities)
5. Saving Persistent State (#SavingPersistentState)

This site uses cookies to store your preferences for site-specific language options. OK

7. Process Lifecycle (#ProcessLifecycle)

Developer Guides

The Activity class is an important part of an application's overall lifecycle, and the way activities are launched and put together is a fundamental part of the platform's application model. For a detailed perspective on the structure of an Android application and how activities behave, please read the Application Fundamentals (<https://developer.android.com/guide/topics/fundamentals.html>) and Tasks and Back Stack (<https://developer.android.com/guide/components/tasks-and-back-stack.html>) developer guides.

You can also find a detailed discussion about how to create activities in the Activities (<https://developer.android.com/guide/components/activities.html>) developer guide.

Fragments

Starting with **HONEYCOMB** (https://developer.android.com/reference/android/os/Build.VERSION_CODES.html#HONEYCOMB), Activity implementations can make use of the **Fragment** (<https://developer.android.com/reference/android/app/Fragment.html>) class to better modularize their code, build more sophisticated user interfaces for larger screens, and help scale their application between small and large screens.

Activity Lifecycle

Activities in the system are managed as an *activity stack*. When a new activity is started, it is placed on the top of the stack and becomes the running activity – the previous activity always remains below it in the stack, and will not come to the foreground again until the new activity exits.

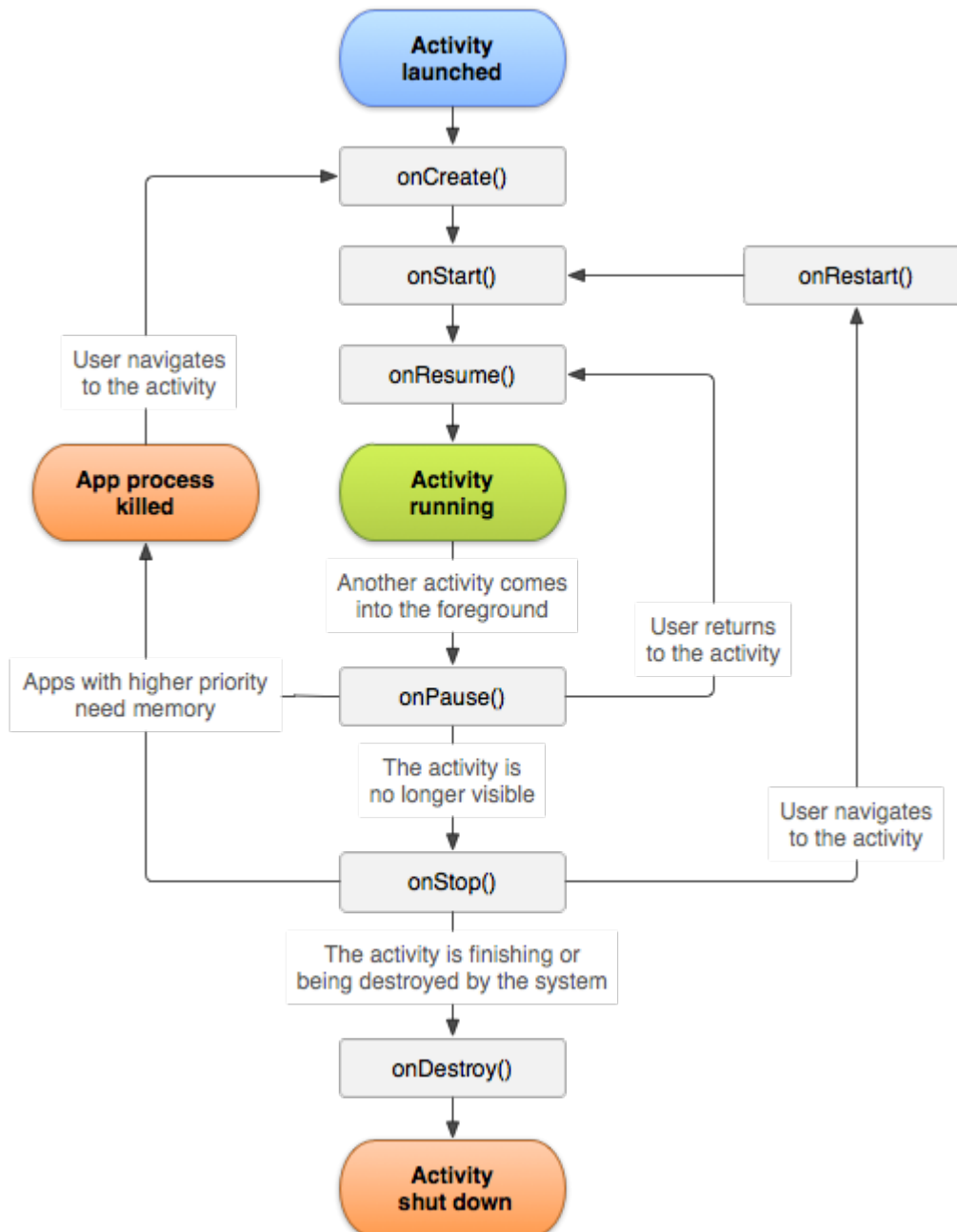
An activity has essentially four states:

- If an activity is in the foreground of the screen (at the top of the stack), it is *active* or *running*.
- If an activity has lost focus but is still visible (that is, a new non-full-sized or transparent activity has focus on top of your activity), it is *paused*. A paused activity is completely alive (it maintains all state and member information and remains attached to the window manager), but can be killed by the system in extreme low memory situations.
- If an activity is completely obscured by another activity, it is *stopped*. It still retains all state and member information, however, it is no longer visible to the user so its window is hidden and it will often be killed by the system when memory is needed elsewhere.

This site uses cookies to store your preferences for site-specific language options.

completely restarted and restored to its previous state.

The following diagram shows the important state paths of an Activity. The square rectangles represent callback methods you can implement to perform operations when the Activity moves between states. The colored ovals are major states the Activity can be in.



There are three key loops you may be interested in monitoring within your activity:

- The **entire lifetime** of an activity happens between the first call to `onCreate(Bundle)` ([https://developer.android.com/reference/android/app/Activity.html#onCreate\(android.os.Bundle\)\)](https://developer.android.com/reference/android/app/Activity.html#onCreate(android.os.Bundle)))

This site uses cookies to store your preferences for site-specific language options.

release all remaining resources in `onDestroy()`. For example, if it has a thread running in the background to download data from the network, it may create that thread in `onCreate()` and then stop the thread in `onDestroy()`.

- The **visible lifetime** of an activity happens between a call to `onStart()` ([https://developer.android.com/reference/android/app/Activity.html#onStart\(\)](https://developer.android.com/reference/android/app/Activity.html#onStart())) until a corresponding call to `onStop()` ([https://developer.android.com/reference/android/app/Activity.html#onStop\(\)](https://developer.android.com/reference/android/app/Activity.html#onStop())). During this time the user can see the activity on-screen, though it may not be in the foreground and interacting with the user. Between these two methods you can maintain resources that are needed to show the activity to the user. For example, you can register a **BroadcastReceiver** (<https://developer.android.com/reference/android/content/BroadcastReceiver.html>) in `onStart()` to monitor for changes that impact your UI, and unregister it in `onStop()` when the user no longer sees what you are displaying. The `onStart()` and `onStop()` methods can be called multiple times, as the activity becomes visible and hidden to the user.
- The **foreground lifetime** of an activity happens between a call to `onResume()` ([https://developer.android.com/reference/android/app/Activity.html#onResume\(\)](https://developer.android.com/reference/android/app/Activity.html#onResume())) until a corresponding call to `onPause()` ([https://developer.android.com/reference/android/app/Activity.html#onPause\(\)](https://developer.android.com/reference/android/app/Activity.html#onPause())). During this time the activity is in front of all other activities and interacting with the user. An activity can frequently go between the resumed and paused states -- for example when the device goes to sleep, when an activity result is delivered, when a new intent is delivered -- so the code in these methods should be fairly lightweight.

The entire lifecycle of an activity is defined by the following Activity methods. All of these are hooks that you can override to do appropriate work when the activity changes state. All activities will implement `onCreate(Bundle)` ([https://developer.android.com/reference/android/app/Activity.html#onCreate\(android.os.Bundle\)](https://developer.android.com/reference/android/app/Activity.html#onCreate(android.os.Bundle))) to do their initial setup; many will also implement `onPause()` ([https://developer.android.com/reference/android/app/Activity.html#onPause\(\)](https://developer.android.com/reference/android/app/Activity.html#onPause())) to commit changes to data and otherwise prepare to stop interacting with the user. You should always call up to your superclass when implementing these methods.

```
public class Activity extends ApplicationContext {  
    protected void onCreate(Bundle savedInstanceState);  
  
    protected void onStart();  
  
    protected void onRestart();  
  
    protected void onResume();  
  
    protected void onPause();  
}
```

This site uses cookies to store your preferences for site-specific language options.

```
protected void onDestroy();
}
```

In general the movement through an activity's lifecycle looks like this:

Method	Description	Killable?
<code>onCreate()</code> (https://developer.android.com/reference/android/app/Activity.html#onCreate(android.os.Bundle))	Called when the activity is first created. This is where you should do all of your normal static set up: create views, bind data to lists, etc. This method also provides you with a Bundle containing the activity's previously frozen state, if there was one. Always followed by <code>onStart()</code> .	No
<code>onRestart()</code> (https://developer.android.com/reference/android/app/Activity.html#onRestart())	Called after your activity has been stopped, prior to it being started again. Always followed by <code>onStart()</code>	No
<code>onStart()</code> (https://developer.android.com/reference/android/app/Activity.html#onStart())	Called when the activity is becoming visible to the user. Followed by <code>onResume()</code> if the activity comes to the foreground, or <code>onStop()</code> if it becomes hidden.	No
<code>onResume()</code> (https://developer.android.com/reference/android/app/Activity.html#onResume())	Called when the activity will start interacting with the user. At this point your activity is at the top of the activity stack, with user input going to it. Always followed by <code>onPause()</code> .	No
<code>onPause()</code> (https://developer.android.com/reference/android/app/Activity.html#onPause())	Called when the system is about to start resuming a previous	Pre-HONEYCOMB (https://dev

This site uses cookies to store your preferences for site-specific language options.

Method	Description	Killable?
	<p>persistent data, stop animations and other things that may be consuming CPU, etc.</p> <p>Implementations of this method must be very quick because the next activity will not be resumed until this method returns.</p> <p>Followed by either <code>onResume()</code> if the activity returns back to the front, or <code>onStop()</code> if it becomes invisible to the user.</p>	
<code>onStop()</code> https://developer.android.com/reference/android/app/Activity.html#onStop()	<p>Called when the activity is no longer visible to the user, because another activity has been resumed and is covering this one. This may happen either because a new activity is being started, an existing one is being brought in front of this one, or this one is being destroyed.</p> <p>Followed by either <code>onRestart()</code> if this activity is coming back to interact with the user, or <code>onDestroy()</code> if this activity is going away.</p>	Yes
<code>onDestroy()</code> (https://developer.android.com/reference/android/app/Activity.html#onDestroy())	<p>The final call you receive before your activity is destroyed. This can happen either because the activity is finishing (someone called <code>finish()</code> (https://developer.android.com/reference/android/app/Activity.html#finish()) on it, or because the system is</p>	Yes

This site uses cookies to store your preferences for site-specific language options.

OK

Method	Description	Killable?
	space. You can distinguish between these two scenarios with the <code>isFinishing()</code> (https://developer.android.com/reference/android/app/Activity.html#isFinishing()) method.	

Note the "Killable" column in the above table -- for those methods that are marked as being killable, after that method returns the process hosting the activity may be killed by the system *at any time* without another line of its code being executed. Because of this, you should use the `onPause()` ([https://developer.android.com/reference/android/app/Activity.html#onPause\(\)](https://developer.android.com/reference/android/app/Activity.html#onPause())) method to write any persistent data (such as user edits) to storage. In addition, the method `onSaveInstanceState(Bundle)` ([https://developer.android.com/reference/android/app/Activity.html#onSaveInstanceState\(android.os.Bundle\)](https://developer.android.com/reference/android/app/Activity.html#onSaveInstanceState(android.os.Bundle))) is called before placing the activity in such a background state, allowing you to save away any dynamic instance state in your activity into the given Bundle, to be later received in `onCreate(Bundle)` ([https://developer.android.com/reference/android/app/Activity.html#onCreate\(android.os.Bundle\)](https://developer.android.com/reference/android/app/Activity.html#onCreate(android.os.Bundle))) if the activity needs to be re-created. See the Process Lifecycle (#ProcessLifecycle) section for more information on how the lifecycle of a process is tied to the activities it is hosting. Note that it is important to save persistent data in `onPause()` ([https://developer.android.com/reference/android/app/Activity.html#onPause\(\)](https://developer.android.com/reference/android/app/Activity.html#onPause())) instead of `onSaveInstanceState(Bundle)` ([https://developer.android.com/reference/android/app/Activity.html#onSaveInstanceState\(android.os.Bundle\)](https://developer.android.com/reference/android/app/Activity.html#onSaveInstanceState(android.os.Bundle))) because the latter is not part of the lifecycle callbacks, so will not be called in every situation as described in its documentation.

Be aware that these semantics will change slightly between applications targeting platforms starting with `HONEYCOMB` (https://developer.android.com/reference/android/os/Build.VERSION_CODES.html#HONEYCOMB) vs. those targeting prior platforms. Starting with Honeycomb, an application is not in the killable state until its `onStop()` ([https://developer.android.com/reference/android/app/Activity.html#onStop\(\)](https://developer.android.com/reference/android/app/Activity.html#onStop())) has returned. This impacts when `onSaveInstanceState(Bundle)` ([https://developer.android.com/reference/android/app/Activity.html#onSaveInstanceState\(android.os.Bundle\)](https://developer.android.com/reference/android/app/Activity.html#onSaveInstanceState(android.os.Bundle))) may be called (it may be safely called after `onPause()` ([https://developer.android.com/reference/android/app/Activity.html#onPause\(\)](https://developer.android.com/reference/android/app/Activity.html#onPause())) and allows an application to safely wait until `onStop()` ([https://developer.android.com/reference/android/app/Activity.html#onStop\(\)](https://developer.android.com/reference/android/app/Activity.html#onStop())) to save persistent state.

This site uses cookies to store your preferences for site-specific language options.

activity is in the killable state, for example, between after `onPause()` to the start of `onResume()`.

Configuration Changes

If the configuration of the device (as defined by the `Resources.Configuration` (<https://developer.android.com/reference/android/content/res/Configuration.html>) class) changes, then anything displaying a user interface will need to update to match that configuration. Because Activity is the primary mechanism for interacting with the user, it includes special support for handling configuration changes.

Unless you specify otherwise, a configuration change (such as a change in screen orientation, language, input devices, etc) will cause your current activity to be *destroyed*, going through the normal activity lifecycle process of `onPause()` ([https://developer.android.com/reference/android/app/Activity.html#onPause\(\)](https://developer.android.com/reference/android/app/Activity.html#onPause())), `onStop()` ([https://developer.android.com/reference/android/app/Activity.html#onStop\(\)](https://developer.android.com/reference/android/app/Activity.html#onStop())), and `onDestroy()` ([https://developer.android.com/reference/android/app/Activity.html#onDestroy\(\)](https://developer.android.com/reference/android/app/Activity.html#onDestroy())) as appropriate. If the activity had been in the foreground or visible to the user, once `onDestroy()` ([https://developer.android.com/reference/android/app/Activity.html#onDestroy\(\)](https://developer.android.com/reference/android/app/Activity.html#onDestroy())) is called in that instance then a new instance of the activity will be created, with whatever `savedInstanceState` the previous instance had generated from `onSaveInstanceState(Bundle)` ([https://developer.android.com/reference/android/app/Activity.html#onSaveInstanceState\(android.os.Bundle\)](https://developer.android.com/reference/android/app/Activity.html#onSaveInstanceState(android.os.Bundle))).

This is done because any application resource, including layout files, can change based on any configuration value. Thus the only safe way to handle a configuration change is to re-retrieve all resources, including layouts, drawables, and strings. Because activities must already know how to save their state and re-create themselves from that state, this is a convenient way to have an activity restart itself with a new configuration.

In some special cases, you may want to bypass restarting of your activity based on one or more types of configuration changes. This is done with the `android:configChanges` (<https://developer.android.com/reference/android/R.attr.html#configChanges>) attribute in its manifest. For any types of configuration changes you say that you handle there, you will receive a call to your current activity's `onConfigurationChanged(Configuration)` ([https://developer.android.com/reference/android/app/Activity.html#onConfigurationChanged\(android.content.res.Configuration\)](https://developer.android.com/reference/android/app/Activity.html#onConfigurationChanged(android.content.res.Configuration))) method instead of being restarted. If a configuration change involves any that you do not handle, however, the activity will still be restarted and `onConfigurationChanged(Configuration)` ([https://developer.android.com/reference/android/app/Activity.html#onConfigurationChanged\(android.content.res.Configuration\)](https://developer.android.com/reference/android/app/Activity.html#onConfigurationChanged(android.content.res.Configuration))) will not be called.

This site uses cookies to store your preferences for site-specific language options.

OK

The `startActivity(Intent)` ([https://developer.android.com/reference/android/app/Activity.html#startActivity\(android.content.Intent\)](https://developer.android.com/reference/android/app/Activity.html#startActivity(android.content.Intent))) method is used to start a new activity, which will be placed at the top of the activity stack. It takes a single argument, an `Intent` (<https://developer.android.com/reference/android/content/Intent.html>), which describes the activity to be executed.

Sometimes you want to get a result back from an activity when it ends. For example, you may start an activity that lets the user pick a person in a list of contacts; when it ends, it returns the person that was selected. To do this, you call the `startActivityForResult(Intent, int)` ([https://developer.android.com/reference/android/app/Activity.html#startActivityForResult\(android.content.Intent, int\)](https://developer.android.com/reference/android/app/Activity.html#startActivityForResult(android.content.Intent, int))) version with a second integer parameter identifying the call. The result will come back through your `onActivityResult(int, int, Intent)` ([https://developer.android.com/reference/android/app/Activity.html#onActivityResult\(int, int, android.content.Intent\)](https://developer.android.com/reference/android/app/Activity.html#onActivityResult(int, int, android.content.Intent))) method.

When an activity exits, it can call `setResult(int)` ([https://developer.android.com/reference/android/app/Activity.html#setResult\(int\)](https://developer.android.com/reference/android/app/Activity.html#setResult(int))) to return data back to its parent. It must always supply a result code, which can be the standard results `RESULT_CANCELED`, `RESULT_OK`, or any custom values starting at `RESULT_FIRST_USER`. In addition, it can optionally return back an `Intent` containing any additional data it wants. All of this information appears back on the parent's `Activity.onActivityResult()`, along with the integer identifier it originally supplied.

If a child activity fails for any reason (such as crashing), the parent activity will receive a result with the code `RESULT_CANCELED`.

```
public class MyActivity extends Activity {
    ...

    static final int PICK_CONTACT_REQUEST = 0;

    public boolean onKeyDown(int keyCode, KeyEvent event) {
        if (keyCode == KeyEvent.KEYCODE_DPAD_CENTER) {
            // When the user center presses, let them pick a contact.
            startActivityForResult(
                new Intent(Intent.ACTION_PICK,
                    new Uri("content://contacts")),
                PICK_CONTACT_REQUEST);
            return true;
        }
        return false;
    }

    protected void onActivityResult(int requestCode, int resultCode,
```

This site uses cookies to store your preferences for site-specific language options.

```
        if (resultCode == RESULT_OK) {  
            // A contact was picked. Here we will just display it  
            // to the user.  
            startActivity(new Intent(Intent.ACTION_VIEW, data));  
        }  
    }  
}
```

Saving Persistent State

There are generally two kinds of persistent state than an activity will deal with: shared document-like data (typically stored in a SQLite database using a content provider (<https://developer.android.com/reference/android/content/ContentProvider.html>)) and internal state such as user preferences.

For content provider data, we suggest that activities use a "edit in place" user model. That is, any edits a user makes are effectively made immediately without requiring an additional confirmation step. Supporting this model is generally a simple matter of following two rules:

- When creating a new document, the backing database entry or file for it is created immediately. For example, if the user chooses to write a new e-mail, a new entry for that e-mail is created as soon as they start entering data, so that if they go to any other activity after that point this e-mail will now appear in the list of drafts.
- When an activity's `onPause()` method is called, it should commit to the backing content provider or file any changes the user has made. This ensures that those changes will be seen by any other activity that is about to run. You will probably want to commit your data even more aggressively at key times during your activity's lifecycle: for example before starting a new activity, before finishing your own activity, when the user switches between input fields, etc.

This model is designed to prevent data loss when a user is navigating between activities, and allows the system to safely kill an activity (because system resources are needed somewhere else) at any time after it has been paused. Note this implies that the user pressing BACK from your activity does *not* mean "cancel" -- it means to leave the activity with its current contents saved away. Canceling edits in an activity must be provided through some other mechanism, such as an explicit "revert" or "undo" option.

See the content package (<https://developer.android.com/reference/android/content/ContentProvider.html>) for more information about content providers. These are a key aspect of how different activities invoke and

This site uses cookies to store your preferences for site-specific language options.

OK

activity. This can be used, for example, to remember the user's preferred initial display in a calendar (day view or week view) or the user's default home page in a web browser.

Activity persistent state is managed with the method `getPreferences(int)`

([https://developer.android.com/reference/android/app/Activity.html#getPreferences\(int\)](https://developer.android.com/reference/android/app/Activity.html#getPreferences(int))), allowing you to retrieve and modify a set of name/value pairs associated with the activity. To use preferences that are shared across multiple application components (activities, receivers, services, providers), you can use the underlying `Context.getSharedPreferences()` ([https://developer.android.com/reference/android/content/Context.html#getSharedPreferences\(java.lang.String, int\)](https://developer.android.com/reference/android/content/Context.html#getSharedPreferences(java.lang.String, int))) method to retrieve a preferences object stored under a specific name. (Note that it is not possible to share settings data across application packages -- for that you will need a content provider.)

Here is an excerpt from a calendar activity that stores the user's preferred view mode in its persistent settings:

```
public class CalendarActivity extends Activity {
    ...

    static final int DAY_VIEW_MODE = 0;
    static final int WEEK_VIEW_MODE = 1;

    private SharedPreferences mPrefs;
    private int mCurViewMode;

    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        SharedPreferences mPrefs = getSharedPreferences();
        mCurViewMode = mPrefs.getInt("view_mode", DAY_VIEW_MODE);
    }

    protected void onPause() {
        super.onPause();

        SharedPreferences.Editor ed = mPrefs.edit();
        ed.putInt("view_mode", mCurViewMode);
        ed.commit();
    }
}
```

Permissions

The ability to start a particular Activity can be enforced when it is declared in its manifest's

This site uses cookies to store your preferences for site-specific language options.

corresponding `<uses-permission>` (<https://developer.android.com/reference/android/R.styleable.html#AndroidManifestUsesPermission>) element in their own manifest to be able to start that activity.

When starting an Activity you can set `Intent.FLAG_GRANT_READ_URI_PERMISSION` (https://developer.android.com/reference/android/content/Intent.html#FLAG_GRANT_READ_URI_PERMISSION) and/or `Intent.FLAG_GRANT_WRITE_URI_PERMISSION` (https://developer.android.com/reference/android/content/Intent.html#FLAG_GRANT_WRITE_URI_PERMISSION) on the Intent. This will grant the Activity access to the specific URIs in the Intent. Access will remain until the Activity has finished (it will remain across the hosting process being killed and other temporary destruction). As of **GINGERBREAD** (https://developer.android.com/reference/android/os/Build.VERSION_CODES.html#GINGERBREAD), if the Activity was already created and a new Intent is being delivered to `onNewIntent(Intent)` ([https://developer.android.com/reference/android/app/Activity.html#onNewIntent\(android.content.Intent\)](https://developer.android.com/reference/android/app/Activity.html#onNewIntent(android.content.Intent))), any newly granted URI permissions will be added to the existing ones it holds.

See the Security and Permissions (<https://developer.android.com/guide/topics/security/security.html>) document for more information on permissions and security in general.

Process Lifecycle

The Android system attempts to keep application process around for as long as possible, but eventually will need to remove old processes when memory runs low. As described in Activity Lifecycle (#ActivityLifecycle), the decision about which process to remove is intimately tied to the state of the user's interaction with it. In general, there are four states a process can be in based on the activities running in it, listed here in order of importance. The system will kill less important processes (the last ones) before it resorts to killing more important processes (the first ones).

1. The **foreground activity** (the activity at the top of the screen that the user is currently interacting with) is considered the most important. Its process will only be killed as a last resort, if it uses more memory than is available on the device. Generally at this point the device has reached a memory paging state, so this is required in order to keep the user interface responsive.
2. A **visible activity** (an activity that is visible to the user but not in the foreground, such as one sitting behind a foreground dialog) is considered extremely important and will not be killed unless that is required to keep the foreground activity running.
3. A **background activity** (an activity that is not visible to the user and has been paused) is no longer critical, so the system may safely kill its process to reclaim memory for other foreground or visible processes. If its process needs to be killed when the user navigates back to the activity.

This site uses cookies to store your preferences for site-specific language options.

savedInstanceState it had previously supplied in `onSaveInstanceState(Bundle)`

(<https://developer.android.com/reference/android>

[/app/Activity.html#onSaveInstanceState\(android.os.Bundle\)](#)) so that it can restart itself in the same state as the user last left it.

4. An **empty process** is one hosting no activities or other application components (such as `Service` (<https://developer.android.com/reference/android/app/Service.html>) or `BroadcastReceiver` (<https://developer.android.com/reference/android/content/BroadcastReceiver.html>) classes). These are killed very quickly by the system as memory becomes low. For this reason, any background operation you do outside of an activity must be executed in the context of an activity `BroadcastReceiver` or `Service` to ensure that the system knows it needs to keep your process around.

Sometimes an Activity may need to do a long-running operation that exists independently of the activity lifecycle itself. An example may be a camera application that allows you to upload a picture to a web site. The upload may take a long time, and the application should allow the user to leave the application while it is executing. To accomplish this, your Activity should start a `Service` (<https://developer.android.com/reference/android/app/Service.html>) in which the upload takes place. This allows the system to properly prioritize your process (considering it to be more important than other non-visible applications) for the duration of the upload, independent of whether the original activity is paused, stopped, or finished.

Summary

Constants	
int	<p><code>DEFAULT_KEYS_DIALER</code> (https://developer.android.com/reference/android/app/Activity.html#DEFAULT_KEYS_DIALER)</p> <p>Use with <code>setDefaultKeyMode(int)</code> (https://developer.android.com/reference/android/app/Activity.html#setDefaultKeyMode(int)) to launch the dialer during default key handling.</p>
int	<p><code>DEFAULT_KEYS_DISABLE</code> (https://developer.android.com/reference/android/app/Activity.html#DEFAULT_KEYS_DISABLE)</p> <p>Use with <code>setDefaultKeyMode(int)</code> (https://developer.android.com/reference/android</p>

This site uses cookies to store your preferences for site-specific language options.

<code>int</code>	<p>DEFAULT_KEYS_SEARCH_GLOBAL (https://developer.android.com/reference/android/app/Activity.html#DEFAULT_KEYS_SEARCH_GLOBAL)</p> <p>Use with <code>setDefaultKeyMode(int)</code> (https://developer.android.com/reference/android/app/Activity.html#setDefaultKeyMode(int)) to specify that unhandled keystrokes will start a global search (typically web search, but some platforms may define alternate methods for global search)</p> <p>See <code>android.app.SearchManager</code> (https://developer.android.com/reference/android/app/SearchManager.html) for more details.</p>
<code>int</code>	<p>DEFAULT_KEYS_SEARCH_LOCAL (https://developer.android.com/reference/android/app/Activity.html#DEFAULT_KEYS_SEARCH_LOCAL)</p> <p>Use with <code>setDefaultKeyMode(int)</code> (https://developer.android.com/reference/android/app/Activity.html#setDefaultKeyMode(int)) to specify that unhandled keystrokes will start an application-defined search.</p>
<code>int</code>	<p>DEFAULT_KEYS_SHORTCUT (https://developer.android.com/reference/android/app/Activity.html#DEFAULT_KEYS_SHORTCUT)</p> <p>Use with <code>setDefaultKeyMode(int)</code> (https://developer.android.com/reference/android/app/Activity.html#setDefaultKeyMode(int)) to execute a menu shortcut in default key handling.</p>
<code>int</code>	<p>RESULT_CANCELED (https://developer.android.com/reference/android/app/Activity.html#RESULT_CANCELED)</p> <p>Standard activity result: operation canceled.</p>
<code>int</code>	<p>RESULT_FIRST_USER (https://developer.android.com/reference/android/app/Activity.html#RESULT_FIRST_USER)</p> <p>Start of user-defined activity results.</p>
<code>int</code>	<p>RESULT_OK (https://developer.android.com/reference/android/app/Activity.html#RESULT_OK)</p> <p>Standard activity result: operation succeeded.</p>

Inherited constants

- ▼ (#)From class `android.content.Context` (<https://developer.android.com/reference/android/content/Context.html>)
- ▼ (#)From interface `android.content.ComponentCallbacks2` (<https://developer.android.com/reference/android/content/ComponentCallbacks2.html>)

This site uses cookies to store your preferences for site-specific language options.

OK

protected static final int[]	FOCUSED_STATE_SET (https://developer.android.com/reference/android/app/Activity.html#FOCUSED_STATE_SET)
---------------------------------------	--

Public constructors

Activity ([https://developer.android.com/reference/android/app/Activity.html#Activity\(\)](https://developer.android.com/reference/android/app/Activity.html#Activity()))()

Public methods

void	addContentView (https://developer.android.com/reference/android/app/Activity.html#addContentView(android.view.View, android.view.ViewGroup.LayoutParams)) Add an additional content view to the activity.
void	closeContextMenu (https://developer.android.com/reference/android/app/Activity.html#closeContextMenu())() Programmatically closes the most recently opened context menu.
void	closeOptionsMenu (https://developer.android.com/reference/android/app/Activity.html#closeOptionsMenu())() Programmatically closes the options menu.
PendingIntent (https://developer.android.com/reference/android/app/PendingIntent.html)	createPendingResult (https://developer.android.com/reference/android/app/Activity.html#createPendingResult(int, android.content.Intent)) Create a new PendingIntent object which you can hand to other components back to your onActivityResult(int, int, Intent) (http://android.app/Activity.html#onActivityResult(int, int, android.content.Intent))
final void	dismissDialog (https://developer.android.com/reference/android/app/Activity.html#dismissDialog(int)(int id)) <i>This method was deprecated in API level 13. Use the new DialogFragment (https://developer.android.com/reference/android/app/DialogFragment) instead; this is also available on older platforms through the AlertDialog (https://developer.android.com/reference/android/app/AlertDialog).</i>
final void	dismissKeyboardShortcutsHelper (https://developer.android.com/reference/android/app/Activity.html#dismissKeyboardShortcutsHelper())()

This site uses cookies to store your preferences for site-specific language options.

OK

boolean	<code>dispatchGenericMotionEvent</code> (https://developer.android.com/app/Activity.html#dispatchGenericMotionEvent(android.view.MotionEvent) (https://developer.android.com/reference/android/view/MotionEvent) Called to process generic motion events.
boolean	<code>dispatchKeyEvent</code> ((K (https://developer.android.com/reference/android/view/KeyEvent) Called to process key events.
boolean	<code>dispatchKeyShortcutEvent</code> ((https://developer.android.com/reference/android/view/KeyEvent) Called to process a key shortcut event.
boolean	<code>dispatchPopulateAccessibilityEvent</code> ((AccessibilityEvent (https://developer.android.com/refere / AccessibilityEvent.html) event) Called to process population of <code>AccessibilityEvent</code> (https://android/view/accessibility/AccessibilityEvent.html)S.
boolean	<code>dispatchTouchEvent</code> ((https://developer.android.com/reference/android/view/MotionEvent) Called to process touch screen events.
boolean	<code>dispatchTrackballEvent</code> ((https://developer.android.com/reference/android/view/MotionEvent) Called to process trackball events.
void	<code>dump</code> ((https://developer.android.com/reference/java/lang/String.html) (https://developer.android.com/reference/java/io/FileDescriptc (https://developer.android.com/reference/java/io/PrintWriter.h (https://developer.android.com/reference/java/lang/String.html) Print the Activity's state into the given stream.
void	<code>enterPictureInPictureMode</code> (https://developer.android.com/reference/android/app/Activity#enterPictureInPictureMode())

This site uses cookies to store your preferences for site-specific language options.

OK

View (https://developer.android.com/reference/android/view/View.html)	findViewById ((int id)) Finds a view that was identified by the id attribute from the XML. onCreate(Bundle) (android.os.Bundle)).
void	finish ((int requestCode)) Call this when your activity is done and should be closed.
void	finishActivity ((int requestCode)) Force finish another activity that you had previously started with startActivityForResult(Intent, int) (Intent)).
void	finishActivityFromChild ((android.app.Activity)) This is called when a child activity of this one calls its finish() method.
void	finishAffinity ((int requestCode)) Finish this activity as well as all activities immediately below same affinity.
void	finishAfterTransition ((int requestCode)) Reverses the Activity Scene entry Transition and triggers the onPostResume() method.
void	finishAndRemoveTask ((int requestCode)) Call this when your activity is done and should be closed and removed as a part of finishing the root activity of the task.
void	finishFromChild ((android.app.Activity)) This is called when a child activity of this one calls its finish() method.
ActionBar	getActionBar ((int requestCode))

This site uses cookies to store your preferences for site-specific language options.

/app/ActionBar.html)	
final Application (https://developer.android.com/reference/android/app/Application.html) 	getApplication (https://developer.android.com/reference/android/app/Activity.html#getApplication())() Return the application that owns this activity.
ComponentName (https://developer.android.com/reference/android/content/ComponentName.html) 	getCallingActivity (https://developer.android.com/reference/android/app/Activity.html#getCallingActivity())() Return the name of the activity that invoked this activity.
String (https://developer.android.com/reference/java/lang/String.html) 	getCallingPackage (https://developer.android.com/reference/android/app/Activity.html#getCallingPackage())() Return the name of the package that invoked this activity.
int	getChangingConfigurations (https://developer.android.com/reference/android/app/Activity.html#getChangingConfigurations())() If this activity is being destroyed because it can not handle a changed (and thus its onConfigurationChanged(Configuration) method is <i>not</i> being called), then you can use this method to occurred while in the process of being destroyed.
ComponentName (https://developer.android.com/reference/android/content/ComponentName.html) 	getComponentName (https://developer.android.com/reference/android/app/Activity.html#getComponentName())() Returns complete component name of this activity.
Scene (https://developer.android.com/reference/android/transition/Scene.html) 	getContentScene (https://developer.android.com/reference/android/app/Activity.html#getContentScene())() Retrieve the Scene (https://developer.android.com/reference/android/transition/Scene.html) representing this window's current content.
TransitionManager (https://developer.android.com/reference/android/transition/TransitionManager.html) 	getContentTransitionManager (https://developer.android.com/reference/android/app/Activity.html#getContentTransitionManager())() Retrieve the TransitionManager (https://developer.android.com/reference/android/transition/TransitionManager.html) responsible for default transitions in
View (https://developer.android.com/reference/android/view/View.html) 	getCurrentFocus (https://developer.android.com/reference/android/app/Activity.html#getCurrentFocus())() Calls getCurrentFocus() (https://developer.android.com/reference/android/view/View.html#getCurrentFocus())

This site uses cookies to store your preferences for site-specific language options.

FragmentManager https://developer.android.com/reference/android/app/FragmentManager.html	getFragmentManager (https://developer.android.com/reference/android/app/Activity.html#getFragmentManager())() Return the FragmentManager for interacting with fragments in this activity.
Intent https://developer.android.com/reference/android/content/Intent.html	getIntent (https://developer.android.com/reference/android/app/Activity.html#getIntent())() Return the intent that started this activity.
Object https://developer.android.com/reference/java/lang/Object.html	getLastNonConfigurationInstance (https://developer.android.com/reference/android/app/Activity.html#getLastNonConfigurationInstance())() Retrieve the non-configuration instance data that was previously stored by onRetainNonConfigurationInstance() (https://developer.android.com/reference/android/app/Activity.html#onRetainNonConfigurationInstance()).
LayoutInflater https://developer.android.com/reference/android/view/LayoutInflater.html	getLayoutInflater (https://developer.android.com/reference/android/app/Activity.html#getLayoutInflater())() Convenience for calling getLayoutInflater() (https://developer.android.com/reference/android/view/Window.html#getLayoutInflater()).
LoaderManager https://developer.android.com/reference/android/app/LoaderManager.html	getLoaderManager (https://developer.android.com/reference/android/app/Activity.html#getLoaderManager())() Return the LoaderManager for this activity, creating it if needed.
String https://developer.android.com/reference/java/lang/String.html	getLocalClassName (https://developer.android.com/reference/android/app/Activity.html#getLocalClassName())() Returns class name for this activity with the package prefix removed.
final MediaController https://developer.android.com/reference/android/media/session/MediaController.html	getMediaController (https://developer.android.com/reference/android/app/Activity.html#getMediaController())() Gets the controller which should be receiving media key and volume events from the foreground.
MenuInflater https://developer.android.com/reference/android/view/MenuInflater.html	getMenuInflater (https://developer.android.com/reference/android/app/Activity.html#getMenuInflater())() Returns a MenuInflater (https://developer.android.com/reference/android/view/MenuInflater.html) with this context.
final Activity https://developer.android.com/reference/android/app/Activity.html	getParent (https://developer.android.com/reference/android/view/View.html#getParent())() Return the parent activity if this view is an embedded child.

This site uses cookies to store your preferences for site-specific language and theme options.

OK

Intent https://developer.android.com/reference/android/content/Intent.html	getParentActivityIntent (https://developer.android.com/reference/android/app/Activity.html#getParentActivityIntent())() Obtain an Intent (https://developer.android.com/reference/android/app/Activity.html#obtainParentActivityIntent()) launch an explicit target activity specified by this activity's log
SharedPreferences https://developer.android.com/reference/android/content/SharedPreferences.html	getPreferences ((int mode)) Retrieve a SharedPreferences (https://developer.android.com/reference/android/content/SharedPreferences.html) object for accessing preferences that
Uri https://developer.android.com/reference/android/net/Uri.html	getReferrer (https://developer.android.com/reference/android/app/Activity.html#getReferrer()) Return information about who launched this activity.
int	getRequestedOrientation (https://developer.android.com/reference/android/app/Activity.html#getRequestedOrientation())() Return the current requested orientation of the activity.
final SearchEvent https://developer.android.com/reference/android/view/SearchEvent.html	getSearchEvent (https://developer.android.com/reference/android/app/Activity.html#getSearchEvent())() During the onSearchRequested() callbacks, this function will return the SearchEvent object that exists.
Object https://developer.android.com/reference/java/lang/Object.html	getSystemService ((String name)) Return the handle to a system-level service by name.
int	getTaskId (https://developer.android.com/reference/android/app/Activity.html#getTaskId()) Return the identifier of the task this activity is in.
final CharSequence https://developer.android.com/reference/java/lang/CharSequence.html	getTitle (https://developer.android.com/reference/android/app/Activity.html#getTitle()) Return the title of the activity.
final int	getTitleColor (https://developer.android.com/reference/android/app/Activity.html#getTitleColor())() Return the title color of the activity.
VoiceInteractor https://developer.android.com/reference/android/app/Activity.html#voiceInteractor	getVoiceInteractor (https://developer.android.com/reference/android/app/Activity.html#getVoiceInteractor())() Return the voice interactor of the activity.

This site uses cookies to store your preferences for site-specific language options.

OK

<code>final int</code>	<code>getVolumeControlStream</code> (https://developer.android.com/reference/android/app/Activity.html#getVolumeControlStream())() Gets the suggested audio stream whose volume should be controlled by the volume controls.
Window (https://developer.android.com/reference/android/view/Window.html)	<code>getWindow</code> (https://developer.android.com/reference/android/app/Activity.html#getWindow())() Retrieve the current Window (https://developer.android.com/reference/android/view/Window.html) of this activity.
WindowManager (https://developer.android.com/reference/android/view/WindowManager.html)	<code>getWindowManager</code> (https://developer.android.com/reference/android/app/Activity.html#getWindowManager())() Retrieve the window manager for showing custom windows.
<code>boolean</code>	<code>hasWindowFocus</code> (https://developer.android.com/reference/android/app/Activity.html#hasWindowFocus())() Returns true if this activity's <i>main</i> window currently has window focus.
<code>void</code>	<code>invalidateOptionsMenu</code> (https://developer.android.com/reference/android/app/Activity.html#invalidateOptionsMenu())() Declare that the options menu has changed, so should be re-created.
<code>boolean</code>	<code>isChangingConfigurations</code> (https://developer.android.com/reference/android/app/Activity.html#isChangingConfigurations())() Check to see whether this activity is in the process of being destroyed to make way for a new configuration.
<code>final boolean</code>	<code>isChild</code> (https://developer.android.com/reference/android/app/Activity.html#isChild())() Is this activity embedded inside of another activity?
<code>boolean</code>	<code>isDestroyed</code> (https://developer.android.com/reference/android/app/Activity.html#isDestroyed())() Returns true if the final <code>onDestroy()</code> (https://developer.android.com/reference/android/app/Activity.html#onDestroy()) call has been made on the Activity.
<code>boolean</code>	<code>isFinishing</code> (https://developer.android.com/reference/android/app/Activity.html#isFinishing())() Check to see whether this activity is in the process of finishing. <code>finish()</code> (https://developer.android.com/reference/android/app/Activity.html#finish()) has requested that it finished.
<code>boolean</code>	<code>isImmersive</code> (https://developer.android.com/reference/android/app/Activity.html#isImmersive())() Bit indicating that this activity is "immersive" and should not be dismissed by the system.

This site uses cookies to store your preferences for site-specific language and theme options.

OK

boolean	<p><code>isInMultiWindowMode</code> (https://developer.android.com/reference/android/app/Activity.html#isInMultiWindowMode())()</p> <p>Returns true if the activity is currently in multi-window mode.</p>
boolean	<p><code>isInPictureInPictureMode</code> (https://developer.android.com/reference/android/app/Activity.html#isInPictureInPictureMode())()</p> <p>Returns true if the activity is currently in picture-in-picture mode.</p>
boolean	<p><code>isLocalVoiceInteractionSupported</code> (https://developer.android.com/reference/android/app/Activity.html#isLocalVoiceInteractionSupported())()</p> <p>Queries whether the currently enabled voice interaction service interactor for use by the activity.</p>
boolean	<p><code>isTaskRoot</code> (https://developer.android.com/reference/android/app/Activity.html#isTaskRoot())()</p> <p>Return whether this activity is the root of a task.</p>
boolean	<p><code>isVoiceInteraction</code> (https://developer.android.com/reference/android/app/Activity.html#isVoiceInteraction())()</p> <p>Check whether this activity is running as part of a voice interaction.</p>
boolean	<p><code>isVoiceInteractionRoot</code> (https://developer.android.com/reference/android/app/Activity.html#isVoiceInteractionRoot())()</p> <p>Like <code>isVoiceInteraction()</code> (https://developer.android.com/reference/android/app/Activity.html#isVoiceInteraction()), but only returns true if the activity is the root of a voice interaction.</p>
<p>final Cursor</p> <p>(https://developer.android.com/reference/android/database/Cursor.html)</p>	<p><code>managedQuery</code> (https://developer.android.com/reference/android/app/Activity.html#managedQuery(android.net.Uri, java.lang.String, java.lang.String[], java.lang.String[]))(Uri (https://developer.android.com/reference/android/net/Uri.html) uri, String[] (https://developer.android.com/reference/java/lang/String.html) projection, String (https://developer.android.com/reference/java/lang/String.html) selection, String (https://developer.android.com/reference/java/lang/String.html) sortOrder)</p> <p><i>This method was deprecated in API level 11. Use CursorLoader instead.</i></p>
boolean	<p><code>moveTaskToBack</code> (https://developer.android.com/reference/android/app/Activity.html#moveTaskToBack(boolean))(boolean nonRoot)</p> <p>Move the task containing this activity to the back of the activity stack.</p>
boolean	<p><code>navigateUpTo</code> (https://developer.android.com/reference/android/app/Activity.html#navigateUpTo(android.content.Intent))(Intent intent)</p>

This site uses cookies to store your preferences for site-specific language options.

boolean	<p><code>navigateUpToFromChild</code> (https://developer.android.com/reference/app/Activity.html#navigateUpToFromChild(android.app.Activity, (https://developer.android.com/reference/android/app/Activity. (https://developer.android.com/reference/android/content/Inter</p> <p>This is called when a child activity of this one calls its <code>navigateUpToFromChild</code> method.</p> <p>(https://developer.android.com/reference/android/app/Activity.html#navigateUpTo(android.content.Intent)) method</p>
void	<p><code>onActionModeFinished</code> (https://developer.android.com/reference/app/Activity.html#onActionModeFinished(android.view.ActionMode (https://developer.android.com/reference/android/view/ActionMode</p> <p>Notifies the activity that an action mode has finished.</p>
void	<p><code>onActionModeStarted</code> (https://developer.android.com/reference/app/Activity.html#onActionModeStarted(android.view.ActionMode (https://developer.android.com/reference/android/view/ActionMode</p> <p>Notifies the Activity that an action mode has been started.</p>
void	<p><code>onActivityResult</code> (https://developer.android.com/reference/app/Activity.html#onActivityResult(int, android.content.Intent (https://developer.android.com/reference/android/content/Inter</p> <p>Called when an activity you launched with an activity transition returns, giving you the resultCode and any data returned by the activity.</p>
void	<p><code>onAttachFragment</code> (https://developer.android.com/reference/app/Activity.html#onAttachFragment(android.app.Fragment)) (Fragment (https://developer.android.com/reference/android/app/Fragment.html) fragment)</p> <p>Called when a Fragment is being attached to this activity, immediately after <code>Fragment.onCreate()</code> and before <code>Fragment.onStart()</code>.</p> <p><code>onAttachFragment</code> (https://developer.android.com/reference/app/Fragment.html#onAttach(android.app.Activity)) method and <code>onAttachFragment</code> (https://developer.android.com/reference/android/app/Fragment.</p>
void	<p><code>onAttachedToWindow</code> (https://developer.android.com/reference/app/Activity.html#onAttachedToWindow())</p> <p>Called when the main window associated with the activity has been attached to the activity's window manager.</p>
void	<p><code>onBackPressed</code> (https://developer.android.com/reference/app/Activity.html#onBackPressed())</p> <p>Called when the activity has detected the user's press of the back key.</p>

This site uses cookies to store your preferences for site-specific language and theme options.

OK

	https://developer.android.com/reference/android/content/res/C Called by the system when the device configuration changes
void	onContentChanged (https://developer.android.com/reference/app/Activity.html#onContentChanged())() This hook is called whenever the content view of the screen c Window.setContentView (https://developer.android.com/ref Window.addContentView (https://developer.android.com/ref https://developer.android.com/ref
boolean	onContextItemSelected (https://developer.android.com/ref https://developer.android.com/reference/app/Activity.html#onContextItemSelected(android.view.MenuItem) This hook is called whenever an item in a context menu is sel
void	onContextMenuClosed (https://developer.android.com/refere https://developer.android.com/reference/app/Activity.html#onContextMenuClosed(android.view.Menu))(Menu) This hook is called whenever the context menu is being close menu with the back/menu button, or when an item is selecte
void	onCreate (https://developer.android.com/reference/android https://developer.android.com/reference/app/Activity.html#onCreate(android.os.Bundle, android.os.Pers Bundle) PersistableBundle (https://developer.android.com/reference https://developer.android.com/reference persistentState) Same as onCreate(android.os.Bundle) (https://developer https://developer.android.com/reference/app/Activity.html#onCreate(android.os.Bundle)) but called for attribute persistableMode (https://developer.android.com/re https://developer.android.com/re /R.attr.html#persistableMode) set to persistAcrossReboots .
void	onCreateContextMenu (https://developer.android.com/refere https://developer.android.com/reference/app/Activity.html#onCreateContextMenu(android.view.ContextMer android.view.ContextMenu.ContextMenuInfo))(ContextMenu (htt https://develop /android/view/ContextMenu.html) menu, View (https://develop https://develop /view/View.html) v, ContextMenu.ContextMenuInfo (https https://developer.android.com/reference/android/view/ContextMenu.ContextMenuInfo.html) menuInfo) Called when a context menu for the view is about to be show
CharSequence	onCreateDescription (https://developer.android.com/refere

This site uses cookies to store your preferences for site-specific language options.

[/reference/java/lang](#) Generate a new description for this activity.

`onCreateNavigateUpTaskStack` ([https://developer.android.com/app/Activity.html#onCreateNavigateUpTaskStack\(android.app.TaskStackBuilder\)](https://developer.android.com/app/Activity.html#onCreateNavigateUpTaskStack(android.app.TaskStackBuilder))) (<https://developer.android.com/reference/android/app/TaskStackBuilder>)

Define the synthetic task stack that will be generated during launch.

```
onCreateOptionsMenu (https://developer.android.com/refere  
/app/Activity.html#onCreateOptionsMenu\(android.view.Menu\))(Me  
/reference/android/view/Menu.html) menu)
```

Initialize the contents of the Activity's standard options menu

```
onCreatePanelMenu (https://developer.android.com/reference/app/Activity.html#onCreatePanelMenu\(int, android.view.Menu\))(  
(https://developer.android.com/reference/android/view/Menu.htm  
Default implementation of onCreatePanelMenu(int, Menu)  
/reference/android/view/Window.Callback.html#onCreatePanelMenu
```

`onCreatePanelView` ([\(https://developer.android.com/reference/app/Activity.html#onCreatePanelView\(int\)\)](https://developer.android.com/reference/app/Activity.html#onCreatePanelView(int)))(`int featureId`)

Default implementation of `onCreatePanelView(int)` ([\(https://android/view/Window.Callback.html#onCreatePanelView\(int\)\)](https://android/view/Window.Callback.html#onCreatePanelView(int))) for

```
onCreateThumbnail (https://developer.android.com/reference/android/app/Activity.html#onCreateThumbnail\(android.graphics.Bitmap,  
android.graphics.Canvas\))  
(https://developer.android.com/reference/android/graphics/Bitmap)  
(https://developer.android.com/reference/android/graphics/Canvas)
```

Generate a new thumbnail for this activity.

```
onCreateView (\(https://developer.android.com/reference/android/app/Activity.html#onCreateView\(android.view.View, java.lang.String, android.util.AttributeSet\)\))(View (https://developer.android.com/reference/android.view.View), String (https://developer.android.com/reference/java/lang/String), Context (https://developer.android.com/reference/android/content/Context), AttributeSet (https://developer.android.com/reference/android/util/AttributeSet)) used when the Activity is first created. The standard implementation of onCreateView(View, String, Context, AttributeSet) is provided by the Activity class. The standard implementation of onCreateView(View, String, Context, AttributeSet) is provided by the Activity class. The standard implementation of onCreateView(View, String, Context, AttributeSet) is provided by the Activity class.
```

ngu OK

View (https://developer.android.com/reference/android/view/View.html)	<code>onCreateView</code> (https://developer.android.com/reference/android/app/Activity.html#onCreateView(java.lang.String, android.content.res.Resources, android.util.AttributeSet))(String (https://developer.android.com/reference/android/os/Bundle) name, Context (https://developer.android.com/reference/android/content/Context) AttributeSet (https://developer.android.com/reference/android/util/AttributeSet) Standard implementation of <code>onCreateView(String, Context, AttributeSet)</code> (https://developer.android.com/reference/android/view/LayoutInflater.Factory.html#onCreateView(java.lang.String, android.content.res.Resources, android.util.AttributeSet)) used when inflating with the <code>LayoutInflater</code> . <code>getSystemService(Class)</code> (<a href="https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>)">https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>)).
void	<code>onDetachedFromWindow</code> (https://developer.android.com/reference/android/app/Activity.html#onDetachedFromWindow())() Called when the main window associated with the activity has been destroyed.
void	<code>onEnterAnimationComplete</code> (https://developer.android.com/reference/android/app/Activity.html#onEnterAnimationComplete())() Activities cannot draw during the period that their windows are animating.
boolean	<code>onGenericMotionEvent</code> (https://developer.android.com/reference/android/app/Activity.html#onGenericMotionEvent(android.view.MotionEvent))(android.view.MotionEvent (https://developer.android.com/reference/android/view/MotionEvent) event) Called when a generic motion event was not handled by any other view.
boolean	<code>onKeyDown</code> (https://developer.android.com/reference/android/app/Activity.html#onKeyDown(int, android.view.KeyEvent))(int keyCode, KeyEvent (https://developer.android.com/reference/android/view/KeyEvent) event) Called when a key was pressed down and not handled by any other view.
boolean	<code>onKeyLongPress</code> (https://developer.android.com/reference/android/app/Activity.html#onKeyLongPress(int, android.view.KeyEvent))(int keyCode, KeyEvent (https://developer.android.com/reference/android/view/KeyEvent) event) Default implementation of <code>KeyEvent.Callback.onKeyLongPress</code> (https://developer.android.com/reference/android/view/KeyEvent.Callback.html#onKeyLongPress(int, android.view.KeyEvent)) returns false (doesn't handle the event).
boolean	<code>onKeyMultiple</code> (https://developer.android.com/reference/android/app/Activity.html#onKeyMultiple(int, int, android.view.KeyEvent))(int keyCode, int repeatCount, KeyEvent (https://developer.android.com/reference/android/view/KeyEvent) event) Called when a key was pressed down and not handled by any other view.

This site uses cookies to store your preferences for site-specific language and theme options.

OK

	Default implementation of <code>KeyEvent.Callback.onKeyMultiple()</code> always returns false (doesn't handle the event).
boolean	<code>onKeyShortcut</code> (https://developer.android.com/reference/android.app.Activity.html#onKeyShortcut(int, android.view.KeyEvent)) (https://developer.android.com/reference/android.view.KeyEvent) Called when a key shortcut event is not handled by any of the
boolean	<code>onKeyUp</code> (https://developer.android.com/reference/android.app.Activity.html#onKeyUp(int, android.view.KeyEvent)) (https://developer.android.com/reference/android.view.KeyEvent) event) Called when a key was released and not handled by any of the
void	<code>onLocalVoiceInteractionStarted</code> (https://developer.android.com/reference/android.app.Activity.html#onLocalVoiceInteractionStarted()) () Callback to indicate that <code>startLocalVoiceInteraction(Bundle)</code> (https://developer.android.com/reference/android/app/Activity.html#startLocalVoiceInteraction(Bundle)) voice interaction session being started.
void	<code>onLocalVoiceInteractionStopped</code> (https://developer.android.com/reference/android.app.Activity.html#onLocalVoiceInteractionStopped()) () Callback to indicate that the local voice interaction has stopped through a call to <code>stopLocalVoiceInteraction()</code> (https://developer.android.com/reference/android/app/Activity.html#stopLocalVoiceInteraction()) or because it was interrupted.
void	<code>onLowMemory</code> (https://developer.android.com/reference/android.app.Activity.html#onLowMemory()) This is called when the overall system is running low on memory. Applications should trim their memory usage.
boolean	<code>onMenuItemSelected</code> (https://developer.android.com/reference/android.app.Activity.html#onMenuItemSelected(int, android.view.MenuItem)) (https://developer.android.com/reference/android.view/MenuItem) Default implementation of <code>onMenuItemSelected(int, MenuItem)</code> (https://developer.android.com/reference/android/view/Window.Callback.html#onMenuItemSelected(int, android.view.MenuItem)) activities.
boolean	<code>onMenuOpened</code> (https://developer.android.com/reference/android.app.Activity.html#onMenuOpened(int, android.view.Menu)) (https://developer.android.com/reference/android.view/Menu) (https://developer.android.com/reference/android/view/Menu) menu) Called when a panel's menu is opened by the user.

This site uses cookies to store your preferences for site-specific language options.

OK

	Called by the system when the activity changes from fullscreen to portrait or visa-versa.
boolean	<code>onNavigateUp</code> (https://developer.android.com/reference/android/app/Activity#onNavigateUp) This method is called whenever the user chooses to navigate up the activity hierarchy from the action bar.
boolean	<code>onNavigateUpFromChild</code> (https://developer.android.com/reference/android/app/Activity#onNavigateUpFromChild(android.app.Activity)) This is called when a child activity of this one attempts to navigate up.
boolean	<code>onOptionsItemSelected</code> (https://developer.android.com/reference/android/view/MenuItem#onOptionsItemSelected) This hook is called whenever an item in your options menu is selected.
void	<code>onOptionsMenuClosed</code> (https://developer.android.com/reference/android/view/Menu#onOptionsMenuClosed(android.view.Menu)(MenuItem)) This hook is called whenever the options menu is being closed, either by the user with the back/menu button, or when an item is selected.
void	<code>onPanelClosed</code> (https://developer.android.com/reference/android/view/Menu#onPanelClosed(int, android.view.Menu)(int)) Default implementation of <code>onPanelClosed(int, Menu)</code> (https://developer.android.com/reference/android/view/Window.Callback#onPanelClosed(int, android.view.Menu))
void	<code>onPictureInPictureModeChanged</code> (https://developer.android.com/reference/android/app/Activity#onPictureInPictureModeChanged(boolean)) Called by the system when the activity changes to and from picture-in-picture mode.
void	<code>onPostCreate</code> (https://developer.android.com/reference/android/os/PersistableBundle) This is the same as <code>onPostCreate(Bundle)</code> (https://developer.android.com/reference/android/os/PersistableBundle#onPostCreate(Bundle)) but is called when the activity is created with the attribute <code>persistableMode</code> (https://developer.android.com/reference/android/os/PersistableBundle)

This site uses cookies to store your preferences for site-specific language and theme options.

void	<code>onPrepareNavigateUpTaskStack</code> (https://developer.android.com/reference/android.app.TaskStackBuilder) Prepare the synthetic task stack that will be generated during
boolean	<code>onPrepareOptionsMenu</code> (https://developer.android.com/reference/android/view/Menu.html) (Menu) Prepare the Screen's standard options menu to be displayed.
boolean	<code>onPreparePanel</code> (https://developer.android.com/reference/android/view/View.html) (Menu) Default implementation of <code>onPreparePanel(int, View, Menu)</code> for activities.
void	<code>onProvideAssistContent</code> (https://developer.android.com/reference/android/app/Activity.html) This is called when the user is requesting an assist, to provide current activity.
void	<code>onProvideAssistData</code> (https://developer.android.com/reference/android/os/Bundle.html) (Bundle) This is called when the user is requesting an assist, to build a Intent in the context of the current application.
void	<a href="https://developer.android.com/reference/android.view.KeyboardShortcutGroup.html#onProvideKeyboardShortcuts(java.util.List<KeyboardShortcutGroup>, android.view.Menu, int)"><code>onProvideKeyboardShortcuts</code> (https://developer.android.com/reference/android/view/KeyboardShortcutGroup.html) (Menu , KeyboardShortcutGroup , int) Called when Keyboard Shortcuts are requested for the current
Uri	<code>onProvideReferrer</code> (https://developer.android.com/reference/android/app/Activity.html)

This site uses cookies to store your preferences for site-specific language options.

void	<p><code>onRequestPermissionsResult</code> (https://developer.android.com/app/Activity.html#onRequestPermissionsResult(int, java.lang.String[])) (https://developer.android.com/reference/java/lang/grantResults)</p> <p>Callback for the result from requesting permissions.</p>
void	<p><code>onRestoreInstanceState</code> (https://developer.android.com/app/Activity.html#onRestoreInstanceState(android.os.Bundle, android.os.Parcelable.PersistableBundle)) (https://developer.android.com/reference/android/os/PersistableBundle)</p> <p>This is the same as <code>onRestoreInstanceState(Bundle)</code> (https://developer.android.com/app/Activity.html#onRestoreInstanceState(android.os.Bundle, android.os.Parcelable.PersistableBundle)) created with the attribute <code>persistableMode</code> (https://developer.android.com/R.attr.html#persistableMode) set to <code>persistAcrossReboots</code>.</p>
Object (https://developer.android.com/reference/java/lang/Object.html)	<p><code>onRetainNonConfigurationInstance</code> (https://developer.android.com/app/Activity.html#onRetainNonConfigurationInstance())()</p> <p>Called by the system, as part of destroying an activity due to configuration change, to return an object that can be retained for the new instance. Called only if the activity was created with the attribute <code>persistableMode</code> (https://developer.android.com/R.attr.html#persistableMode) set to <code>persistAcrossReboots</code>.</p>
void	<p><code>onSaveInstanceState</code> (https://developer.android.com/app/Activity.html#onSaveInstanceState(android.os.Bundle, android.os.Parcelable.PersistableBundle)) (https://developer.android.com/reference/android/os/PersistableBundle)</p> <p>This is the same as <code>onSaveInstanceState(Bundle)</code> (https://developer.android.com/app/Activity.html#onSaveInstanceState(android.os.Bundle, android.os.Parcelable.PersistableBundle)) created with the attribute <code>persistableMode</code> (https://developer.android.com/R.attr.html#persistableMode) set to <code>persistAcrossReboots</code>.</p>
boolean	<p><code>onSearchRequested</code> (https://developer.android.com/app/Activity.html#onSearchRequested(android.view.SearchEvent)) (https://developer.android.com/reference/android/view/SearchEvent)</p> <p>This hook is called when the user signals the desire to start a search.</p>
boolean	<p><code>onSearchRequested</code> (https://developer.android.com/app/Activity.html#onSearchRequested())()</p> <p>Called when the user signals the desire to start a search.</p>
void	<p><code>onStateNotSaved</code> (https://developer.android.com/app/Activity.html#onStateNotSaved())()</p>

This site uses cookies to store your preferences for site-specific language options.

OK

	/app/Activity.html#onResume() is coming up, prior to other pre onNewIntent(Intent) (https://developer.android.com/refere /app/Activity.html#onNewIntent(android.content.Intent)) and on Intent) (https://developer.android.com/reference/android/app int, android.content.Intent)).
boolean	onTouchEvent (https://developer.android.com/reference/andrc /app/Activity.html#onTouchEvent(android.view.MotionEvent))(MO (https://developer.android.com/reference/android/view/MotionEv Called when a touch screen event was not handled by any of
boolean	onTrackballEvent (https://developer.android.com/reference/ /app/Activity.html#onTrackballEvent(android.view.MotionEvent)) (https://developer.android.com/reference/android/view/MotionEv Called when the trackball was moved and not handled by any
void	onTrimMemory (https://developer.android.com/reference/andrc /app/Activity.html#onTrimMemory(int))(int level) Called when the operating system has determined that it is a unneeded memory from its process.
void	onUserInteraction (https://developer.android.com/reference/ /app/Activity.html#onUserInteraction())() Called whenever a key, touch, or trackball event is dispatched
void	onVisibleBehindCanceled (https://developer.android.com/r /app/Activity.html#onVisibleBehindCanceled())() Called when a translucent activity over this activity is becomi launched.
void	onWindowAttributesChanged (https://developer.android.cor /app/Activity.html#onWindowAttributesChanged(android.view.Winc (WindowManager.LayoutParams (https://developer.android. /view/WindowManager.LayoutParams.html) params) This is called whenever the current window attributes change
void	onWindowFocusChanged (https://developer.android.com/refer /app/Activity.html#onWindowFocusChanged(boolean))(boolean h Called when the current Window (https://developer.android.cc of the activity gains or loses focus.
ActionMode	onWindowStartingActionMode (https://developer.android.c

This site uses cookies to store your preferences for site-specific langu

/view/ActionMode.html)	/view/ActionMode.Callback.html) <code>callback, int type</code>) Called when an action mode is being started for this window.
ActionMode (https://developer.android.com/reference/android/view/ActionMode.html)	<code>onWindowStartingActionMode</code> (https://developer.android.com/reference/android/view/Activity.html#onWindowStartingActionMode(android.view.ActionMode.Callback (https://developer.android.com/reference/android/view/ActionMode.Callback.html) <code>callback</code>) Give the Activity a chance to control the UI for an action mode.
void	<code>openContextMenu</code> (https://developer.android.com/reference/android/view/Activity.html#openContextMenu(android.view.View)(View.OnClickListener (https://developer.android.com/reference/android/view/View.html) <code>view</code>) Programmatically opens the context menu for a particular view.
void	<code>openOptionsMenu</code> (https://developer.android.com/reference/android/view/Activity.html#openOptionsMenu())() Programmatically opens the options menu.
void	<code>overridePendingTransition</code> (https://developer.android.com/reference/android/view/Activity.html#overridePendingTransition(int,int)(int,int))(<code>int enterTransitionId</code> , <code>int returnTransitionId</code>) Call immediately after one of the flavors of <code>startActivity()</code> (https://developer.android.com/reference/android/view/Activity.html#startActivity(android.content.Intent)) or <code>startActivityForResult()</code> (https://developer.android.com/reference/android/view/Activity.html#startActivityForResult(android.content.Intent,int)) to specify the transition animation to perform next.
void	<code>postponeEnterTransition</code> (https://developer.android.com/reference/android/view/Activity.html#postponeEnterTransition())() Postpone the entering activity transition when Activity was started with <code>makeSceneTransitionAnimation(Activity, android.util.Pair<android.view.View, java.lang.String>...)</code> (https://developer.android.com/reference/android/util/Pair.html) or <code>makeSceneTransitionAnimation(ActivityOptions, android.util.Pair<android.view.View, java.lang.String>...)</code> (https://developer.android.com/reference/android/view/ActivityOptions.html#makeSceneTransitionAnimation(android.view.ActivityOptions,android.util.Pair(android.view.View,java.lang.String)...)").
void	<code>recreate</code> (https://developer.android.com/reference/android/view/Activity.html#recreate()) Cause this Activity to be recreated with a new instance.
void	<code>registerForContextMenu</code> (https://developer.android.com/reference/android/view/Activity.html#registerForContextMenu(android.view.View)(View.OnClickListener (https://developer.android.com/reference/android/view/View.html) <code>view</code>) Registers a context menu to be shown for the given view (<code>mu</code>

This site uses cookies to store your preferences for site-specific language options.

OK

boolean	<p><code>releaseInstance</code> (https://developer.android.com/reference/android/app/Activity.html#releaseInstance())()</p> <p>Ask that the local app instance of this activity be released to</p>
final void	<p><code>removeDialog</code> ((int id))</p> <p><i>This method was deprecated in API level 13. Use the new <code>DialogFragment</code> (https://developer.android.com/reference/android/app/DialogFragment) instead; this is also available on older platforms through the <code>AlertDialog</code> (https://developer.android.com/reference/android/app/AlertDialog) class.</i></p>
void	<p><code>reportFullyDrawn</code> (())</p> <p>Report to the system that your app is now fully drawn, purely for performance monitoring purposes (this does not impact the visible behavior of the activity).</p>
<p><code>DragAndDropPermissions</code></p> <p>(https://developer.android.com/reference/android/view/DragAndDropPermissions.html)</p>	<p><code>requestDragAndDropPermissions</code> ((android.view.View view))</p> <p>Request <code>DragAndDropPermissions</code> (https://developer.android.com/reference/android/view/DragAndDropPermissions.html) object bound to this activity to be able to request permissions for content URIs associated with the <code>DragEvent</code> (https://developer.android.com/reference/android/view/DragEvent.html).</p>
final void	<p><code>requestPermissions</code> ((java.lang.String[] permissions, int requestCode))</p> <p>Requests permissions to be granted to this application.</p>
final void	<p><code>requestShowKeyboardShortcuts</code> (())</p> <p>Request the Keyboard Shortcuts screen to show up.</p>
boolean	<p><code>requestVisibleBehind</code> ((boolean visibleBehind))(boolean visibleBehind)</p> <p>Activities that want to remain visible behind a translucent activity should call this method anytime between the start of <code>onResume()</code> (<code>onResume()</code>) and the return from <code>onPause()</code> (<code>onPause()</code>).</p>
final boolean	<p><code>requestWindowFeature</code> ((int featureId))</p>

This site uses cookies to store your preferences for site-specific language and theme options.

<code>final void</code>	<code>runOnUiThread</code> ((https://developer.android.com/reference/android/app/Activity.html#runOnUiThread(java.lang.Runnable))) (<code>Runnable</code> (https://developer.android.com/reference/java/lang/Runnable.html) <code>action</code>) Runs the specified action on the UI thread.
<code>void</code>	<code>setActionBar</code> ((https://developer.android.com/reference/android/app/Activity.html#setActionBar(android.widget.Toolbar))) (<code>Toolbar</code> (https://developer.android.com/reference/android/widget/Toolbar.html) <code>toolbar</code>) Set a Toolbar (https://developer.android.com/reference/android.support.design.widget.Toolbar) on the activity window.
<code>void</code>	<code>setContentTransitionManager</code> ((https://developer.android.com/reference/android/app/Activity.html#setContentTransitionManager(android.transition.TransitionManager)) (<code>TransitionManager</code> (https://developer.android.com/reference/android/transition/TransitionManager.html) <code>tm</code>) Set the TransitionManager (https://developer.android.com/reference/android/transition/TransitionManager) to use for default transitions in this view.
<code>void</code>	<code>setContentView</code> ((https://developer.android.com/reference/android/app/Activity.html#setContentView(android.view.View)) (<code>View</code> (https://developer.android.com/reference/android/view/View.html) <code>view</code>) Set the activity content to an explicit view.
<code>void</code>	<code>setContentView</code> ((https://developer.android.com/reference/android/app/Activity.html#setContentView(int)) (<code>int</code> <code>layoutResID</code>) Set the activity content to an explicit view.
<code>void</code>	<code>setContentView</code> ((https://developer.android.com/reference/android/app/Activity.html#setContentView(int)) (<code>int</code> <code>layoutResID</code>) Set the activity content from a layout resource.
<code>final void</code>	<code>setDefaultKeyMode</code> ((https://developer.android.com/reference/android/app/Activity.html#setDefaultKeyMode(int)) (<code>int</code> <code>mode</code>) Select the default key handling for this activity.
<code>void</code>	<code>setEnterSharedElementCallback</code> ((https://developer.android.com/reference/android/app/Activity.html#setEnterSharedElementCallback(android.app.Activity.OnSharedElementCallback)) (<code>OnSharedElementCallback</code> (https://developer.android.com/reference/android/app/Activity.OnSharedElementCallback) <code>callback</code>) Set the callback for the shared element transition.

This site uses cookies to store your preferences for site-specific language options.

OK

	<p>(https://developer.android.com/reference/android/app/ActivityOptions.html#makeSceneTransitionAnimation(android.app.Activity, java.lang.String)) was used to start an Activity, <i>callback</i> will be called on the <i>launched</i> Activity.</p>
void	<p><code>setExitSharedElementCallback</code> (https://developer.android.com/reference/android/app/Activity.html#setExitSharedElementCallback(android.app.Activity, SharedElementCallback) (https://developer.android.com/reference/android/app/SharedElementCallback.html) <i>callback</i>)</p> <p>When <code>makeSceneTransitionAnimation(Activity, android.app.ActivityOptions)</code> (https://developer.android.com/reference/android/app/ActivityOptions.html#makeSceneTransitionAnimation(android.app.Activity, java.lang.String)) was used to start an Activity, <i>callback</i> will be called on the <i>launching</i> Activity.</p>
final void	<p><code>setFeatureDrawable</code> (https://developer.android.com/reference/android/view/Window.html#setFeatureDrawable(int, android.graphics.drawable.Drawable) (https://developer.android.com/reference/android/graphics/drawable/Drawable) (https://developer.android.com/reference/android/graphics/drawable/Drawable)</p> <p>Convenience for calling <code>setFeatureDrawable(int, Drawable)</code> (https://developer.android.com/reference/android/view/Window.html#setFeatureDrawable(int, android.graphics.drawable.Drawable))</p>
final void	<p><code>setFeatureDrawableAlpha</code> (https://developer.android.com/reference/android/view/Window.html#setFeatureDrawableAlpha(int, int) (https://developer.android.com/reference/android/view/Window.html#setFeatureDrawableAlpha(int, int)) (https://developer.android.com/reference/android/view/Window.html#setFeatureDrawableAlpha(int, int))</p> <p>Convenience for calling <code>setFeatureDrawableAlpha(int, int)</code> (https://developer.android.com/reference/android/view/Window.html#setFeatureDrawableAlpha(int, int))</p>
final void	<p><code>setFeatureDrawableResource</code> (https://developer.android.com/reference/android/view/Window.html#setFeatureDrawableResource(int, int) (https://developer.android.com/reference/android/view/Window.html#setFeatureDrawableResource(int, int)) (https://developer.android.com/reference/android/view/Window.html#setFeatureDrawableResource(int, int))</p> <p>Convenience for calling <code>setFeatureDrawableResource(int, int)</code> (https://developer.android.com/reference/android/view/Window.html#setFeatureDrawableResource(int, int))</p>
final void	<p><code>setFeatureDrawableUri</code> (https://developer.android.com/reference/android/view/Window.html#setFeatureDrawableUri(int, android.net.Uri) (https://developer.android.com/reference/android/net/Uri.html) (https://developer.android.com/reference/android/net/Uri.html)</p> <p>Convenience for calling <code>setFeatureDrawableUri(int, Uri)</code> (https://developer.android.com/reference/android/view/Window.html#setFeatureDrawableUri(int, android.net.Uri))</p>
void	<p><code>setFinishOnTouchOutside</code> (https://developer.android.com/reference/android/app/Activity.html#setFinishOnTouchOutside(boolean)) (https://developer.android.com/reference/android/app/Activity.html#setFinishOnTouchOutside(boolean)) (https://developer.android.com/reference/android/app/Activity.html#setFinishOnTouchOutside(boolean))</p> <p>Set whether this activity is finished when touched outside its window.</p>

This site uses cookies to store your preferences for site-specific language options.

void	setImmersive (https://developer.android.com/reference/android/app/Activity.html#setImmersive(boolean))(boolean i) Adjust the current immersive mode setting.
void	setIntent (https://developer.android.com/reference/android/app/Activity.html#setIntent(android.content.Intent))(Intent / reference/android/content/Intent.html) newIntent) Change the intent returned by <code>getIntent()</code> (https://developer.android.com/reference/android/app/Activity.html#getIntent()).
final void	setMediaController (https://developer.android.com/reference/android/app/Activity.html#setMediaController(android.media.session.MediaController) (https://developer.android.com/reference/android/media/session/MediaController.html) to send media keys and volume changes.
final void	setProgress (https://developer.android.com/reference/android/app/Activity.html#setProgress(int))(int progress) <i>This method was deprecated in API level 24. No longer supported.</i>
final void	setProgressBarIndeterminate (https://developer.android.com/reference/android/app/Activity.html#setProgressBarIndeterminate(boolean))(boolean) <i>This method was deprecated in API level 24. No longer supported.</i>
final void	setProgressBarIndeterminateVisibility (https://developer.android.com/reference/android/app/Activity.html#setProgressBarIndeterminateVisibility(boolean))(boolean) <i>This method was deprecated in API level 24. No longer supported.</i>
final void	setProgressBarVisibility (https://developer.android.com/reference/android/app/Activity.html#setProgressBarVisibility(boolean))(boolean) <i>This method was deprecated in API level 24. No longer supported.</i>
void	setRequestedOrientation (https://developer.android.com/reference/android/app/Activity.html#setRequestedOrientation(int))(int requestedOrientation) Change the desired orientation of this activity.
final void	setResult (https://developer.android.com/reference/android/app/Activity.html#setResult(int,Intent))(int resultCode, Intent (https://developer.android.com/reference/android/content/Intent.html) data) Call this to set the result that your activity will return to its caller.
final void	setResult (https://developer.android.com/reference/android/app/Activity.html#setResult(int,Intent))(int resultCode, Intent (https://developer.android.com/reference/android/content/Intent.html) data)

This site uses cookies to store your preferences for site-specific language options.

OK

<code>final void</code>	<code>setSecondaryProgress</code> ((https://developer.android.com/reference/app/Activity.html#setSecondaryProgress(int)))(<code>int secondaryProgress</code>) <i>This method was deprecated in API level 24. No longer supported.</i>
<code>void</code>	<code>setTaskDescription</code> ((https://developer.android.com/reference/app/Activity.html#setTaskDescription(android.app.ActivityManager.TaskDescription))(<code>ActivityManager.TaskDescription taskDescription</code>) Sets information describing the task with this activity for presentation to the UI.
<code>void</code>	<code>setTheme</code> ((https://developer.android.com/reference/android/app/Activity.html#setTheme(int)))(<code>int themeResId</code>) Set the base theme for this context.
<code>void</code>	<code>setTitle</code> ((https://developer.android.com/reference/android/app/Activity.html#setTitle(java.lang.CharSequence)))(<code>CharSequence title</code>) Change the title associated with this activity.
<code>void</code>	<code>setTitle</code> ((https://developer.android.com/reference/android/app/Activity.html#setTitle(int)))(<code>int titleId</code>) Change the title associated with this activity.
<code>void</code>	<code>setTitleColor</code> ((https://developer.android.com/reference/android/app/Activity.html#setTitleColor(int)))(<code>int textColor</code>) <i>This method was deprecated in API level 21. Use action bar styling instead.</i>
<code>void</code>	<code>setVisible</code> ((https://developer.android.com/reference/android/app/Activity.html#setVisible(boolean)))(<code>boolean visible</code>) Control whether this activity's main window is visible.
<code>final void</code>	<code>setVolumeControlStream</code> ((https://developer.android.com/reference/android/app/Activity.html#setVolumeControlStream(int)))(<code>int streamType</code>) Suggests an audio stream whose volume should be changed.
<code>void</code>	<code>setVrModeEnabled</code> ((https://developer.android.com/reference/android/app/Activity.html#setVrModeEnabled(boolean, android.content.ComponentName))(<code>boolean enabled</code> , <code>ComponentName requestedComponent</code>) Enable or disable virtual reality (VR) mode for this Activity.

This site uses cookies to store your preferences for site-specific language options.

OK

	https://developer.android.com/reference/java/lang/String.html Gets whether you should show UI with rationale for requestin
boolean	shouldUpRecreateTask (https://developer.android.com/refer/app/Activity.html#shouldUpRecreateTask(android.content.Intent (https://developer.android.com/reference/android/content/Inter Returns true if the app should recreate the task when navigat targetIntent.
boolean	showAssist (https://developer.android.com/reference/android/app/Activity.html#showAssist(android.os.Bundle))(Bundle (https://developer.android.com/reference/android/os/Bundle.html) args) Ask to have the current assistant shown to the user.
final boolean	showDialog (https://developer.android.com/reference/android/os.Bundle))(int id, Bundle (https://developer.andr/os/Bundle.html) args) <i>This method was deprecated in API level 13. Use the new Dia</i> (https://developer.android.com/reference/android/app/DialogFra FragmentManager (https://developer.android.com/reference/a instead; this is also available on older platforms through the A
final void	showDialog (https://developer.android.com/reference/android/app/Activity.html#showDialog(int))(int id) <i>This method was deprecated in API level 13. Use the new Dia</i> (https://developer.android.com/reference/android/app/DialogFra FragmentManager (https://developer.android.com/reference/a instead; this is also available on older platforms through the A
void	showLockTaskEscapeMessage (https://developer.android.com/refer/app/Activity.html#showLockTaskEscapeMessage())() Shows the user the system defined message for telling the u
ActionMode (https://developer.android.com/reference/android/view/ActionMode.html)	startActionMode (https://developer.android.com/reference/a/app/Activity.html#startActionMode(android.view.ActionMode.Call (https://developer.android.com/reference/android/view/ActionMc type) Start an action mode of the given type.
ActionMode (https://developer.android.com/reference/a/app/Activity.html#startActionMode(android.view.ActionMode.Call)	startActionMode (https://developer.android.com/reference/a/app/Activity.html#startActionMode(android.view.ActionMode.Call

This site uses cookies to store your preferences for site-specific langu options.

	<code>/android/view/ActionMode.html#TYPE_PRIMARY).</code>
<code>void</code>	<code>startActivities</code> (https://developer.android.com/reference/android/app/Activity.html#startActivities(android.content.Intent[], android.os.Bundle)) Launch a new activity.
<code>void</code>	<code>startActivities</code> (https://developer.android.com/reference/android/app/Activity.html#startActivities(android.content.Intent[])) Same as <code>startActivities(Intent[], Bundle)</code> (https://developer.android.com/reference/android/app/Activity.html#startActivities(android.content.Intent[], android.os.Bundle)) specified.
<code>void</code>	<code>startActivity</code> (https://developer.android.com/reference/android/app/Activity.html#startActivity(android.content.Intent)) (Intent) Same as <code>startActivity(Intent, Bundle)</code> (https://developer.android.com/reference/android/app/Activity.html#startActivity(android.content.Intent, android.os.Bundle)) specified.
<code>void</code>	<code>startActivity</code> (https://developer.android.com/reference/android/app/Activity.html#startActivity(android.content.Intent, android.os.Bundle)) Launch a new activity.
<code>void</code>	<code>startActivityForResult</code> (https://developer.android.com/reference/android/app/Activity.html#startActivityForResult(android.content.Intent, int)) Same as calling <code>startActivityForResult(Intent, int, Bundle)</code> (https://developer.android.com/reference/android/app/Activity.html#startActivityForResult(android.content.Intent, int, android.os.Bundle)) options.
<code>void</code>	<code>startActivityForResult</code> (https://developer.android.com/reference/android/app/Activity.html#startActivityForResult(android.content.Intent, int, android.os.Bundle)) Launch an activity for which you would like a result when it finishes.

This site uses cookies to store your preferences for site-specific language options.

void	<p><code>startActivityFromChild</code> ((https://developer.android.com/re/app/Activity.html#startActivityFromChild(android.app.Activity,int)))(<code>Activity</code> (https://developer.android.com/reference/andri <code>Intent</code> (https://developer.android.com/reference/android/cont <code>requestCode</code>))</p> <p>Same as calling <code>startActivityFromChild(Activity, Intent</code> ((https://developer.android.com/reference/android/app/Activity.html#startActivityFromChild(android.app.Activity,android.os.Bundle))) with no options.</p>
void	<p><code>startActivityFromChild</code> ((https://developer.android.com/re/app/Activity.html#startActivityFromChild(android.app.Activity,android.os.Bundle)))(<code>Activity</code> (https://developer.android.com/child, <code>Intent</code> (https://developer.android.com/reference/andri <code>requestCode</code>, <code>Bundle</code> (https://developer.android.com/r <code>options</code>))</p> <p>This is called when a child activity of this one calls its <code>start/</code> ((https://developer.android.com/reference/android/app/Activity.html#startActivity(android.content.Intent))) OR <code>S</code> <code>int</code>) (https://developer.android.com/reference/android/app/Activity.html#startActivityForResult(android.content.Inte</p>
void	<p><code>startActivityFromFragment</code> ((https://developer.android.com/re/app/Activity.html#startActivityFromFragment(android.app.Fragment,android.os.Bundle)))(<code>Fragment</code> (https://developer.android.com/fragment, <code>Intent</code> (https://developer.android.com/reference/andri <code>requestCode</code>, <code>Bundle</code> (https://developer.android.com/r <code>options</code>))</p> <p>This is called when a <code>Fragment</code> in this activity calls its <code>start/</code> ((https://developer.android.com/reference/android/app/Fragment.html#startActivity(android.content.Intent))) OR <code>S</code> <code>int</code>) (https://developer.android.com/reference/android/app/Fragment.html#startActivityForResult(android.content.Inte</p>
void	<p><code>startActivityFromFragment</code> ((https://developer.android.com/re/app/Activity.html#startActivityFromFragment(android.app.Fragment,int)))(<code>Fragment</code> (https://developer.android.com/reference/andri <code>Intent</code> (https://developer.android.com/reference/android/cont <code>requestCode</code>))</p>

This site uses cookies to store your preferences for site-specific langu options.

	<pre> https://developer.android.com/reference/android /app/Activity.html#startActivityFromFragment(android.app.Fragm android.os.Bundle)) with no options. </pre>
boolean	<pre> startActivityIfNeeded (https://developer.android.com/refi /app/Activity.html#startActivityIfNeeded(android.content.Inter (https://developer.android.com/reference/android/content/Inter Bundle (https://developer.android.com/reference/android/os/B A special variation to launch an activity only if a new activity i Intent. </pre>
boolean	<pre> startActivityIfNeeded (https://developer.android.com/refi /app/Activity.html#startActivityIfNeeded(android.content.Inter (https://developer.android.com/reference/android/content/Inter Same as calling startActivityIfNeeded(Intent, int, B /reference/android/app/Activity.html#startActivityIfNeeded(anc android.os.Bundle)) with no options. </pre>
void	<pre> startIntentSender (https://developer.android.com/reference /app/Activity.html#startIntentSender(android.content.IntentSer int))(IntentSender (https://developer.android.com/reference intent, Intent (https://developer.android.com/reference/an fillInIntent, int flagsMask, int flagsValues, int Same as calling startIntentSender(IntentSender, Inte (https://developer.android.com/reference/android /app/Activity.html#startIntentSender(android.content.IntentSer int, android.os.Bundle)) with no options. </pre>
void	<pre> startIntentSender (https://developer.android.com/reference /app/Activity.html#startIntentSender(android.content.IntentSer int, android.os.Bundle))(IntentSender (https://developer.ai /IntentSender.html) intent, Intent (https://developer.andr /Intent.html) fillInIntent, int flagsMask, int flags (https://developer.android.com/reference/android/os/Bundle.htm Like startActivity(Intent, Bundle) (https://developer.a /app/Activity.html#startActivity(android.content.Intent, andrc IntentSender to start; see startIntentSenderForResult(int, int, Bundle) (https://developer.android.com/reference /app/Activity.html#startIntentSenderForResult(android.content. </pre>

ngu OK

[startIntentSenderFromChild](https://developer.android.com/reference/android/app/IntentSenderFromChild) (<https://developer.android.com/reference/android/app/IntentSenderFromChild>)

OK

	<p><code>int, android.content.Intent, int, int, int))(Activity (http://android/app/Activity.html) child, IntentSender (https://content/IntentSender.html) intent, int requestCode, Ir/reference/android/content/Intent.html) fillInIntent, int int extraFlags)</code></p> <p>Same as calling <code>startIntentSenderFromChild(Activity, int, int, int, Bundle)</code> (https://developer.android.com/re/app/Activity.html#startIntentSenderFromChild(android.app.Acti</p> <p><code>int, android.content.Intent, int, int, int, android.os.Bundle)</code></p>
<code>void</code>	<p><code>startLocalVoiceInteraction</code> (https://developer.android.c/app/Activity.html#startLocalVoiceInteraction(android.os.Bundl</p> <p>(https://developer.android.com/reference/android/os/Bundle.htm</p> <p>Starts a local voice interaction session.</p>
<code>void</code>	<p><code>startLockTask</code> (https://developer.android.com/reference/andi/app/Activity.html#startLockTask())()</p> <p>Request to put this Activity in a mode where the user is locke</p>
<code>void</code>	<p><code>startManagingCursor</code> (https://developer.android.com/refere/app/Activity.html#startManagingCursor(android.database.Cursor</p> <p>(https://developer.android.com/reference/android/database/Curs</p> <p><i>This method was deprecated in API level 11. Use the new Cur:</i></p> <p>(https://developer.android.com/reference/android/content/Cursc</p> <p><code>LoaderManager</code> (https://developer.android.com/reference/andi</p> <p><i>this is also available on older platforms through the Android co</i></p>
<code>boolean</code>	<p><code>startNextMatchingActivity</code> (https://developer.android.com/app/Activity.html#startNextMatchingActivity(android.content.I</p> <p>(https://developer.android.com/reference/android/content/Inter</p> <p>(https://developer.android.com/reference/android/os/Bundle.htm</p> <p>Special version of starting an activity, for use when you are re</p>
<code>boolean</code>	<p><code>startNextMatchingActivity</code> (https://developer.android.com/app/Activity.html#startNextMatchingActivity(android.content.I</p> <p>(https://developer.android.com/reference/android/content/Inter</p> <p>Same as calling <code>startNextMatchingActivity(Intent, Bu</code></p> <p>/reference/android/app/Activity.html#startNextMatchingActivity</p> <p><code>android.os.Bundle))</code> with no options.</p>
<code>void</code>	<p><code>startPostponedEnterTransition</code> (https://developer.android.com/app/Activity.html#startPostponedEnterTransition(android.os.Bundle))</p>

This site uses cookies to store your preferences for site-specific language options.

	<p>Begin postponed transitions after <code>postponeEnterTransition()</code></p> <p><code>postponeEnterTransition()</code></p>
<code>void</code>	<p><code>startSearch</code> (https://developer.android.com/reference/android/app/Activity.html#startSearch(java.lang.String, boolean, android.support.design.widget.SearchView))</p> <p><code>selectInitialQuery</code>, <code>Bundle</code> (https://developer.android.com/reference/android/os/Bundle.html)</p> <p><code>appSearchData</code>, <code>boolean globalSearch</code>)</p> <p>This hook is called to launch the search UI.</p>
<code>void</code>	<p><code>stopLocalVoiceInteraction</code> (<code>stopLocalVoiceInteraction()</code>)</p> <p>Request to terminate the current voice interaction that was previously started by <code>startLocalVoiceInteraction(Bundle)</code> (<code>startLocalVoiceInteraction(Bundle)</code>)</p>
<code>void</code>	<p><code>stopLockTask</code> (<code>stopLockTask()</code>)</p> <p>Allow the user to switch away from the current task.</p>
<code>void</code>	<p><code>stopManagingCursor</code> (<code>stopManagingCursor(android.database.Cursor)</code>)</p> <p>(https://developer.android.com/reference/android/database/Cursor.html)</p> <p><i>This method was deprecated in API level 11. Use the new <code>CursorLoader</code> (https://developer.android.com/reference/android/content/CursorLoader.html) instead.</i></p> <p><code>LoaderManager</code> (https://developer.android.com/reference/android/content/LoaderManager.html)</p> <p><i>this is also available on older platforms through the <code>AndroidManifest</code> attribute <code>android.support.design.widget.SearchView</code></i></p>
<code>void</code>	<p><code>takeKeyEvents</code> (<code>takeKeyEvents(boolean)</code>)</p> <p><code>(boolean get)</code></p> <p>Request that key events come to this activity.</p>
<code>void</code>	<p><code>triggerSearch</code> (<code>triggerSearch(java.lang.String, android.os.Bundle)</code>)</p> <p>(https://developer.android.com/reference/java/lang/String.html)</p> <p>(https://developer.android.com/reference/android/os/Bundle.html)</p> <p>Similar to <code>startSearch(String, boolean, Bundle, boolean)</code> (<code>startSearch(String, boolean, Bundle, boolean)</code>), but actually fires off the search query after invoking <code>startSearch</code>.</p>
<code>void</code>	<p><code>unregisterForContextMenu</code> (<code>unregisterForContextMenu(android.view.View)</code>)</p>

This site uses cookies to store your preferences for site-specific language options.

OK

Protected methods	
void	<p><code>onActivityResult</code> (https://developer.android.com/reference/android/app/Activity.html#onActivityResult(int, int, android.content.Intent) <code>requestCode</code>, <code>int resultCode</code>, <code>Intent</code> (https://developer.android.com/reference/android/content/Intent.html) <code>data</code>)</p> <p>Called when an activity you launched exits, giving you the requestCode you started it with, the resultCode it returned, and any additional data</p>
void	<p><code>onApplyThemeResource</code> (https://developer.android.com/reference/android/content/res/Resources.Theme.html) <code>theme</code>, <code>int resid</code>, <code>boolean first</code>)</p> <p>Called by <code>setTheme(int)</code> (https://developer.android.com/reference/android/view/ContextThemeWrapper.html#setTheme(int)) and <code>getTheme()</code> (https://developer.android.com/reference/android/view/ContextThemeWrapper.html#getTheme()) to apply a theme resource to the current Theme object.</p>
void	<p><code>onChildTitleChanged</code> (https://developer.android.com/reference/android/app/Activity.html#onChildTitleChanged(android.app.Activity, java.lang.CharSequence)) (<code>Activity</code> (https://developer.android.com/reference/android/app/Activity.html) <code>childActivity</code>, <code>CharSequence title</code>)</p>
void	<p><code>onCreate</code> (https://developer.android.com/reference/android/app/Activity.html#onCreate(android.os.Bundle)) (<code>Bundle</code> (https://developer.android.com/reference/android/os/Bundle.html) <code>savedInstanceState</code>)</p> <p>Called when the activity is starting.</p>
Dialog (https://developer.android.com/reference/android/app/Dialog.html)	<p><code>onCreateDialog</code> (https://developer.android.com/reference/android/app/Activity.html#onCreateDialog(int)) (<code>int id</code>)</p> <p><i>This method was deprecated in API level 8. Old no-arguments version <code>onCreateDialog(int, Bundle)</code> (https://developer.android.com/reference/android/app/Activity.html#onCreateDialog(int, android.os.Bundle))</i></p>
Dialog	<p><code>onCreateDialog</code> (https://developer.android.com/reference/android/app/Activity.html#onCreateDialog(int, android.os.Bundle)) (<code>int id</code>, <code>Bundle savedInstanceState</code>)</p>

This site uses cookies to store your preferences for site-specific language options.

<code>/app/Dialog.html)</code>	<p><code>args()</code></p> <p><i>This method was deprecated in API level 13. Use the new <code>DialogFragment</code> (https://developer.android.com/reference/android/app/DialogFragment) class with <code>FragmentManager</code> (https://developer.android.com/reference/android/app/FragmentManager.html) instead; this is also available on platforms through the Android compatibility package.</i></p>
<code>void</code>	<p><code>onDestroy</code> (https://developer.android.com/reference/android/app/Activity.html#onDestroy())()</p> <p>Perform any final cleanup before an activity is destroyed.</p>
<code>void</code>	<p><code>onNewIntent</code> (https://developer.android.com/reference/android/app/Activity.html#onNewIntent(android.content.Intent))(<code>Intent</code> (https://developer.android.com/reference/android/content/Intent.html) <code>intent</code>)</p> <p>This is called for activities that set <code>launchMode</code> to "singleTop" in the manifest, or if a client used the <code>FLAG_ACTIVITY_SINGLE_TOP</code> (https://developer.android.com/reference/android/content/Intent.html#FLAG_ACTIVITY_SINGLE_TOP) flag when calling <code>startActivity(Intent)</code> (https://developer.android.com/reference/android/app/Activity.html#startActivity(android.content.Intent)).</p>
<code>void</code>	<p><code>onPause</code> (https://developer.android.com/reference/android/app/Activity.html#onPause())()</p> <p>Called as part of the activity lifecycle when an activity is going into the background, but has not (yet) been killed.</p>
<code>void</code>	<p><code>onPostCreate</code> (https://developer.android.com/reference/android/app/Activity.html#onPostCreate(android.os.Bundle))(<code>Bundle</code> (https://developer.android.com/reference/android/os/Bundle.html) <code>savedInstanceState</code>)</p> <p>Called when activity start-up is complete (after <code>onStart()</code> (https://developer.android.com/reference/android/app/Activity.html#onStart()) and <code>onRestoreInstanceState(Bundle)</code> (https://developer.android.com/reference/android/app/Activity.html#onRestoreInstanceState(android.os.Bundle)) have been called).</p>
<code>void</code>	<p><code>onPostResume</code> (https://developer.android.com/reference/android/app/Activity.html#onPostResume())()</p>

This site uses cookies to store your preferences for site-specific language options.

OK

	<code>/app/Activity.html#onResume()</code> has been called).
<code>void</code>	<p><code>onPrepareDialog</code> ((https://developer.android.com/reference/android/app/Activity.html#onPrepareDialog(int, android.app.Dialog, android.os.Bundle)))(<code>int id</code>, <code>Dialog</code> (https://developer.android.com/reference/android/app/Dialog.html) <code>dialog</code>, <code>Bundle</code> (https://developer.android.com/reference/android/os/Bundle.html)) ;</p> <p><i>This method was deprecated in API level 13. Use the new <code>DialogFragment</code> (https://developer.android.com/reference/android/app/DialogFragment) class with <code>FragmentManager</code> (https://developer.android.com/reference/android/app/FragmentManager) instead; this is also available on platforms through the Android compatibility package.</i></p>
<code>void</code>	<p><code>onPrepareDialog</code> ((https://developer.android.com/reference/android/app/Activity.html#onPrepareDialog(int, android.app.Dialog)))(<code>int id</code>, <code>Dialog</code> (https://developer.android.com/reference/android/app/Dialog.html) <code>dialog</code>)</p> <p><i>This method was deprecated in API level 8. Old no-arguments version <code>onPrepareDialog(int, Dialog, Bundle)</code> ((https://developer.android.com/reference/android/app/Activity.html#onPrepareDialog(int, android.app.Dialog, android.os.Bundle))).</i></p>
<code>void</code>	<p><code>onRestart</code> ((https://developer.android.com/reference/android/app/Activity.html#onRestart()))()</p> <p>Called after <code>onStop()</code> ((https://developer.android.com/reference/android/app/Activity.html#onStop())) when the current activity is being re-created to the user (the user has navigated back to it).</p>
<code>void</code>	<p><code>onRestoreInstanceState</code> ((https://developer.android.com/reference/android/app/Activity.html#onRestoreInstanceState(android.os.Bundle)))(<code>Bundle</code> (https://developer.android.com/reference/android/os/Bundle.html) <code>savedInstanceState</code>)</p> <p>This method is called after <code>onStart()</code> ((https://developer.android.com/reference/android/app/Activity.html#onStart())) when the activity is re-initialized from a previously saved state, given here in <i>savedInstanceState</i>.</p>
<code>void</code>	<p><code>onResume</code> ((https://developer.android.com/reference/android/app/Activity.html#onResume()))()</p> <p>Called after <code>onRestoreInstanceState(Bundle)</code></p>

This site uses cookies to store your preferences for site-specific language options.

OK

	<p>(https://developer.android.com/reference/android/app/Activity.html#onRestart()), or <code>onPause()</code> (https://developer.android.com/reference/android/app/Activity.html#onPause()), for your activity to start interacting user.</p>
<code>void</code>	<p><code>onSaveInstanceState</code> (https://developer.android.com/reference/android/app/Activity.html#onSaveInstanceState(android.os.Bundle)) (<code>Bundle</code> (https://developer.android.com/reference/android/os/Bundle.html) <code>outState</code>)</p> <p>Called to retrieve per-instance state from an activity before being that the state can be restored in <code>onCreate(Bundle)</code></p> <p>(https://developer.android.com/reference/android/app/Activity.html#onCreate(android.os.Bundle)) OR <code>onRestoreInstanceState(Bundle)</code> (https://developer.android.com/reference/android/app/Activity.html#onRestoreInstanceState(android.os.Bundle)) (the (https://developer.android.com/reference/android/os/Bundle.html) <code>p</code> by this method will be passed to both).</p>
<code>void</code>	<p><code>onStart</code> (https://developer.android.com/reference/android/app/Activity.html#onStart())()</p> <p>Called after <code>onCreate(Bundle)</code> (https://developer.android.com/reference/android/app/Activity.html#onCreate(android.os.Bundle)) — or after <code>onRestart()</code> (https://developer.android.com/reference/android/app/Activity.html#onRestart()) when the activity had been stopped now again being displayed to the user.</p>
<code>void</code>	<p><code>onStop</code> (https://developer.android.com/reference/android/app/Activity.html#onStop())()</p> <p>Called when you are no longer visible to the user.</p>
<code>void</code>	<p><code>onTitleChanged</code> (https://developer.android.com/reference/android/app/Activity.html#onTitleChanged(java.lang.CharSequence,int)) (<code>CharSequence</code> (https://developer.android.com/reference/java/CharSequence.html) <code>title</code>, <code>int</code> <code>color</code>)</p>
<code>void</code>	<p><code>onUserLeaveHint</code> (https://developer.android.com/reference/android/app/Activity.html#onUserLeaveHint())()</p> <p>Called as part of the activity lifecycle when an activity is about to</p>

This site uses cookies to store your preferences for site-specific language options.

Inherited methods

▼ (#)From class `android.view.ContextThemeWrapper` (<https://developer.android.com/reference/android/view/ContextThemeWrapper.html>)

▼ (#)From class `android.content.ContextWrapper` (<https://developer.android.com/reference/android/content/ContextWrapper.html>)

▼ (#)From class `android.content.Context` (<https://developer.android.com/reference/android/content/Context.html>)

▼ (#)From class `java.lang.Object` (<https://developer.android.com/reference/java/lang/Object.html>)

▼ (#)From interface `android.view.LayoutInflater.Factory2` (<https://developer.android.com/reference/android/view/LayoutInflater.Factory2.html>)

▼ (#)From interface `android.view.Window.Callback` (<https://developer.android.com/reference/android/view/Window.Callback.html>)

▼ (#)From interface `android.view.KeyEvent.Callback` (<https://developer.android.com/reference/android/view/KeyEvent.Callback.html>)

▼ (#)From interface `android.view.View.OnCreateContextMenuListener` (<https://developer.android.com/reference/android/view/View.OnCreateContextMenuListener.html>)

▼ (#)From interface `android.content.ComponentCallbacks2` (<https://developer.android.com/reference/android/content/ComponentCallbacks2.html>)

▼ (#)From interface `android.view.LayoutInflater.Factory` (<https://developer.android.com/reference/android/view/LayoutInflater.Factory.html>)

▼ (#)From interface `android.content.ComponentCallbacks` (<https://developer.android.com/reference/android/content/ComponentCallbacks.html>)

This site uses cookies to store your preferences for site-specific language options.

DEFAULT_KEYS_DIALER

Added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

`int DEFAULT_KEYS_DIALER`

Use with `setDefaultKeyMode(int)` ([https://developer.android.com/reference/android/app/Activity.html#setDefaultKeyMode\(int\)](https://developer.android.com/reference/android/app/Activity.html#setDefaultKeyMode(int))) to launch the dialer during default key handling.

See also:

`setDefaultKeyMode(int)` ([https://developer.android.com/reference/android/app/Activity.html#setDefaultKeyMode\(int\)](https://developer.android.com/reference/android/app/Activity.html#setDefaultKeyMode(int)))

Constant Value: 1 (0x00000001)

DEFAULT_KEYS_DISABLE

Added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

`int DEFAULT_KEYS_DISABLE`

Use with `setDefaultKeyMode(int)` ([https://developer.android.com/reference/android/app/Activity.html#setDefaultKeyMode\(int\)](https://developer.android.com/reference/android/app/Activity.html#setDefaultKeyMode(int))) to turn off default handling of keys.

See also:

`setDefaultKeyMode(int)` ([https://developer.android.com/reference/android/app/Activity.html#setDefaultKeyMode\(int\)](https://developer.android.com/reference/android/app/Activity.html#setDefaultKeyMode(int)))

Constant Value: 0 (0x00000000)

DEFAULT_KEYS_SEARCH_GLOBAL

Added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

`int DEFAULT_KEYS_SEARCH_GLOBAL`

Use with `setDefaultKeyMode(int)` ([https://developer.android.com/reference/android/app/Activity.html#setDefaultKeyMode\(int\)](https://developer.android.com/reference/android/app/Activity.html#setDefaultKeyMode(int))) to specify that unhandled keystrokes will start a global search (typically web search, but some platforms may define alternate methods for global search)

See `android.app.SearchManager` (<https://developer.android.com/reference/android/app/SearchManager>)

This site uses cookies to store your preferences for site-specific language options.



See also:

`setDefaultKeyMode(int)` ([https://developer.android.com/reference/android/app/Activity.html#setDefaultKeyMode\(int\)](https://developer.android.com/reference/android/app/Activity.html#setDefaultKeyMode(int)))

Constant Value: 4 (0x00000004)

DEFAULT_KEYS_SEARCH_LOCAL

Added in API level 11 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

`int DEFAULT_KEYS_SEARCH_LOCAL`

Use with `setDefaultKeyMode(int)` ([https://developer.android.com/reference/android/app/Activity.html#setDefaultKeyMode\(int\)](https://developer.android.com/reference/android/app/Activity.html#setDefaultKeyMode(int))) to specify that unhandled keystrokes will start an application-defined search. (If the application or activity does not actually define a search, the the keys will be ignored.)

See `android.app.SearchManager` (<https://developer.android.com/reference/android/app/SearchManager.html>) for more details.

See also:

`setDefaultKeyMode(int)` ([https://developer.android.com/reference/android/app/Activity.html#setDefaultKeyMode\(int\)](https://developer.android.com/reference/android/app/Activity.html#setDefaultKeyMode(int)))

Constant Value: 3 (0x00000003)

DEFAULT_KEYS_SHORTCUT

Added in API level 7 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

`int DEFAULT_KEYS_SHORTCUT`

Use with `setDefaultKeyMode(int)` ([https://developer.android.com/reference/android/app/Activity.html#setDefaultKeyMode\(int\)](https://developer.android.com/reference/android/app/Activity.html#setDefaultKeyMode(int))) to execute a menu shortcut in default key handling.

That is, the user does not need to hold down the menu key to execute menu shortcuts.

See also:

`setDefaultKeyMode(int)` (<https://developer.android.com/reference/android>)

This site uses cookies to store your preferences for site-specific language options.

OK

RESULT_CANCELED

Added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
int RESULT_CANCELED
```

Standard activity result: operation canceled.

Constant Value: 0 (0x00000000)

RESULT_FIRST_USER

Added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
int RESULT_FIRST_USER
```

Start of user-defined activity results.

Constant Value: 1 (0x00000001)

RESULT_OK

Added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
int RESULT_OK
```

Standard activity result: operation succeeded.

Constant Value: -1 (0xffffffff)

Fields

FOCUSED_STATE_SET

Added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
int[] FOCUSED_STATE_SET
```

This site uses cookies to store your preferences for site-specific language options.

Activity

Added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

Activity ()

Public methods

addContentView

Added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
void addContentView (View (https://developer.android.com/reference/android/view/View.html) view, ViewGroup.LayoutParams (https://developer.android.com/reference/android/view
```

Add an additional content view to the activity. Added after any existing ones in the activity -- existing views are NOT removed.

Parameters	
<code>view</code>	<code>View</code> : The desired content to display.
<code>params</code>	<code>ViewGroup.LayoutParams</code> : Layout parameters for the view.

closeContextMenu

Added in API level 3 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
void closeContextMenu ()
```

Programmatically closes the most recently opened context menu, if showing.

closeOptionsMenu

Added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
void closeOptionsMenu ()
```

Programmatically closes the options menu. If the options menu is already closed, this method does nothing.

This site uses cookies to store your preferences for site-specific language options.

OK

createPendingResult

Added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
PendingIntent (https://developer.android.com/reference/android/app/PendingIntent.html) createPendingIntent (https://developer.android.com/reference/android/content/Intent.html) data int flags)
```

Create a new PendingIntent object which you can hand to others for them to use to send result data back to your `onActivityResult(int, int, Intent)` ([https://developer.android.com/reference/android/app/Activity.html#onActivityResult\(int, int, android.content.Intent\)](https://developer.android.com/reference/android/app/Activity.html#onActivityResult(int, int, android.content.Intent))) callback. The created object will be either one-shot (becoming invalid after a result is sent back) or multiple (allowing any number of results to be sent through it).

Parameters	
requestCode	int: Private request code for the sender that will be associated with the result data when it is returned. The sender can not modify this value, allowing you to identify incoming results.
data	Intent: Default data to supply in the result, which may be modified by the sender.
flags	int: May be PendingIntent.FLAG_ONE_SHOT (https://developer.android.com/reference/android/app/PendingIntent.html#FLAG_ONE_SHOT), PendingIntent.FLAG_NO_CREATE (https://developer.android.com/reference/android/app/PendingIntent.html#FLAG_NO_CREATE), PendingIntent.FLAG_CANCEL_CURRENT (https://developer.android.com/reference/android/app/PendingIntent.html#FLAG_CANCEL_CURRENT), PendingIntent.FLAG_UPDATE_CURRENT (https://developer.android.com/reference/android/app/PendingIntent.html#FLAG_UPDATE_CURRENT), or any of the flags as supported by Intent.fillIn() (https://developer.android.com/reference/android/content/Intent.html#fillIn(android.content.Intent, int)) to control which unspecified parts of the intent that can be supplied when the actual send happens.

Returns	
PendingIntent (https://developer.android.com/reference/android/app/PendingIntent.html)	Returns an existing or new PendingIntent matching the given parameters. May return null only if PendingIntent.FLAG_NO_CREATE (https://developer.android.com/reference/android/app/PendingIntent.html#FLAG_NO_CREATE) has been supplied.

PendingIntent (<https://developer.android.com/reference/android/app/PendingIntent.html>)

dismissDialog Added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
void dismissDialog (int id)
```

This method was deprecated in API level 13.

Use the new [DialogFragment](https://developer.android.com/reference/android/app/DialogFragment.html) (<https://developer.android.com/reference/android/app/DialogFragment.html>) class with [FragmentManager](https://developer.android.com/reference/android/app/FragmentManager.html) (<https://developer.android.com/reference/android/app/FragmentManager.html>) instead; this is also available on older platforms through the Android compatibility package.

Dismiss a dialog that was previously shown via `showDialog(int)` ([https://developer.android.com/reference/android/app/Activity.html#showDialog\(int\)](https://developer.android.com/reference/android/app/Activity.html#showDialog(int))).

Parameters	
id	int: The id of the managed dialog.

Throws	
IllegalArgumentException (https://developer.android.com/reference/java/lang/IllegalArgumentException.html)	if the id was not previously shown via <code>showDialog(int)</code> (https://developer.android.com/reference/android/app/Activity.html#showDialog(int)).

See also:

- `onCreateDialog(int, Bundle)` ([https://developer.android.com/reference/android/app/Activity.html#onCreateDialog\(int, android.os.Bundle\)](https://developer.android.com/reference/android/app/Activity.html#onCreateDialog(int, android.os.Bundle)))
- `onPrepareDialog(int, Dialog, Bundle)` ([https://developer.android.com/reference/android/app/Activity.html#onPrepareDialog\(int, android.app.Dialog, android.os.Bundle\)](https://developer.android.com/reference/android/app/Activity.html#onPrepareDialog(int, android.app.Dialog, android.os.Bundle)))
- `showDialog(int)` ([https://developer.android.com/reference/android/app/Activity.html#showDialog\(int\)](https://developer.android.com/reference/android/app/Activity.html#showDialog(int)))
- `removeDialog(int)` ([https://developer.android.com/reference/android/app/Activity.html#removeDialog\(int\)](https://developer.android.com/reference/android/app/Activity.html#removeDialog(int)))

This site uses cookies to store your preferences for site-specific language options. OK

dismissKeyboardShortcutsHelper

Added in API level 24 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
void dismissKeyboardShortcutsHelper ()
```

Dismiss the Keyboard Shortcuts screen.

dispatchGenericMotionEvent

Added in API level 12 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
boolean dispatchGenericMotionEvent (MotionEvent (https://developer.android.com/reference/a
```

Called to process generic motion events. You can override this to intercept all generic motion events before they are dispatched to the window. Be sure to call this implementation for generic motion events that should be handled normally.

Parameters

ev	MotionEvent : The generic motion event.
----	--

Returns

boolean	boolean Return true if this event was consumed.
---------	---

dispatchKeyEvent

Added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
boolean dispatchKeyEvent (KeyEvent (https://developer.android.com/reference/android/view/KeyE
```

Called to process key events. You can override this to intercept all key events before they are dispatched to the window. Be sure to call this implementation for key events that should be handled normally.

Parameters

event	KeyEvent : The key event.
-------	----------------------------------

Returns

boolean	boolean Return true if this event was consumed.
---------	---

This site uses cookies to store your preferences for site-specific language options.

OK

dispatchKeyShortcutEvent

Added in API level 11 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
boolean dispatchKeyShortcutEvent (KeyEvent (https://developer.android.com/reference/android
```

Called to process a key shortcut event. You can override this to intercept all key shortcut events before they are dispatched to the window. Be sure to call this implementation for key shortcut events that should be handled normally.

Parameters

event	KeyEvent : The key shortcut event.
-------	---

Returns

boolean	True if this event was consumed.
---------	----------------------------------

dispatchPopulateAccessibilityEvent

Added in API level 11 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
boolean dispatchPopulateAccessibilityEvent (AccessibilityEvent (https://developer.andr
```

Called to process population of **AccessibilityEvent** (<https://developer.android.com/reference/android/view/accessibility/AccessibilityEvent.html>)S.

Parameters

event	AccessibilityEvent : The event.
-------	--

Returns

boolean	boolean Return true if event population was completed.
---------	--

dispatchTouchEvent

Added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
boolean dispatchTouchEvent (MotionEvent (https://developer.android.com/reference/android/vie
```

Called to process touch screen events. You can override this to intercept all touch screen events before they are dispatched to the window. Be sure to call this implementation for touch screen

This site uses cookies to store your preferences for site-specific language options.

OK

ev	MotionEvent : The touch screen event.
----	--

Returns

boolean	boolean Return true if this event was consumed.
----------------	---

dispatchTrackballEvent

Added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
boolean dispatchTrackballEvent (MotionEvent (https://developer.android.com/reference/android
```

Called to process trackball events. You can override this to intercept all trackball events before they are dispatched to the window. Be sure to call this implementation for trackball events that should be handled normally.

Parameters

ev	MotionEvent : The trackball event.
----	---

Returns

boolean	boolean Return true if this event was consumed.
----------------	---

dump

Added in API level 11 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
void dump (String (https://developer.android.com/reference/java/lang/String.html) prefix,
           FileDescriptor (https://developer.android.com/reference/java/io/FileDescriptor
           PrintWriter (https://developer.android.com/reference/java/io/PrintWriter.html)
           String[] (https://developer.android.com/reference/java/lang/String.html) args)
```

Print the Activity's state into the given stream. This gets invoked if you run "adb shell dumpsys activity <activity_component_name>".

Parameters

prefix	String : Desired prefix to prepend at each line of output.
--------	---

fd	FileDescriptor : The raw file descriptor that the dump is being sent to.
----	---

args	String[] The PrintWriter will be passed to dumpSystemState. This will be
------	--

This site uses cookies to store your preferences for site-specific language options.

OK

args**String:** additional arguments to the dump request.

enterPictureInPictureMode

Added in API level 24 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
void enterPictureInPictureMode ()
```

Puts the activity in picture-in-picture mode.

See also:

[supportsPictureInPicture](https://developer.android.com/reference/android/R.attr.html#supportsPictureInPicture) (<https://developer.android.com/reference/android/R.attr.html#supportsPictureInPicture>)

findViewById

Added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
View (https://developer.android.com/reference/android/view/View.html) findViewById (int id)
```

Finds a view that was identified by the id attribute from the XML that was processed in

[onCreate\(Bundle\)](https://developer.android.com/reference/android/app/Activity.html#onCreate(android.os.Bundle)) ([https://developer.android.com/reference/android/app/Activity.html#onCreate\(android.os.Bundle\)](https://developer.android.com/reference/android/app/Activity.html#onCreate(android.os.Bundle))).

Parameters

id	int
-----------	------------

Returns

View (https://developer.android.com/reference/android/view/View.html)	The view if found or null otherwise.
--	--------------------------------------

finish

Added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

This site uses cookies to store your preferences for site-specific language options.

Call this when your activity is done and should be closed. The `ActivityResult` is propagated back to whoever launched you via `onActivityResult()`.

finishActivity

Added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
void finishActivity (int requestCode)
```

Force finish another activity that you had previously started with

`startActivityForResult(Intent, int)` ([https://developer.android.com/reference/android/app/Activity.html#startActivityForResult\(android.content.Intent, int\)](https://developer.android.com/reference/android/app/Activity.html#startActivityForResult(android.content.Intent, int))).

Parameters

<code>requestCode</code>	<code>int</code> : The request code of the activity that you had given to <code>startActivityForResult()</code> . If there are multiple activities started with this request code, they will all be finished.
--------------------------	---

finishActivityFromChild

Added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
void finishActivityFromChild (Activity (https://developer.android.com/reference/android/app/  
int requestCode)
```

This is called when a child activity of this one calls its `finishActivity()`.

Parameters

<code>child</code>	<code>Activity</code> : The activity making the call.
<code>requestCode</code>	<code>int</code> : Request code that had been used to start the activity.

finishAffinity

Added in API level 16 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
void finishAffinity ()
```

Finish this activity as well as all activities immediately below it in the current task that have the

This site uses cookies to store your preferences for site-specific language options.

to switch out of the current task and in to its own task. In this case, if the user has navigated down into any other activities of the second application, all of those should be removed from the original task as part of the task switch.

Note that this finish does *not* allow you to deliver results to the previous activity, and an exception will be thrown if you are trying to do so.

finishAfterTransition

Added in API level 21 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
void finishAfterTransition ()
```

Reverses the Activity Scene entry Transition and triggers the calling Activity to reverse its exit Transition. When the exit Transition completes, `finish()` ([https://developer.android.com/reference/android/app/Activity.html#finish\(\)](https://developer.android.com/reference/android/app/Activity.html#finish())) is called. If no entry Transition was used, `finish()` is called immediately and the Activity exit Transition is run.

See also:

```
makeSceneTransitionAnimation(Activity, android.util.Pair[])  
(https://developer.android.com/reference/android/app/ActivityOptions.html#makeSceneTransitionAnimation\(android.app.Activity, android.util.Pair<android.view.View, java.lang.String>...\))
```

finishAndRemoveTask

Added in API level 21 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
void finishAndRemoveTask ()
```

Call this when your activity is done and should be closed and the task should be completely removed as a part of finishing the root activity of the task.

finishFromChild

Added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
void finishFromChild (Activity (https://developer.android.com/reference/android/app/Activity.h
```

This site uses cookies to store your preferences for site-specific language options.

OK

`finish()` on this activity (the parent), finishing the entire group.

Parameters	
<code>child</code>	Activity: The activity making the call.

See also:

`finish()` ([https://developer.android.com/reference/android/app/Activity.html#finish\(\)](https://developer.android.com/reference/android/app/Activity.html#finish()))

getActionBar

Added in API level 11 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

`ActionBar` (<https://developer.android.com/reference/android/app/ActionBar.html>) `getActionBar ()`

Retrieve a reference to this activity's ActionBar.

Returns	
<code>ActionBar</code> (https://developer.android.com/reference/android/app/ActionBar.html)	The Activity's ActionBar, or null if it does not have one.

getApplication

Added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

`Application` (<https://developer.android.com/reference/android/app/Application.html>) `getApplication`

Return the application that owns this activity.

Returns	
<code>Application</code> (https://developer.android.com/reference/android/app/Application.html)	

This site uses cookies to store your preferences for site-specific language options.

getCallingActivity

Added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

`ComponentName` (<https://developer.android.com/reference/android/content/ComponentName.html>) `getCa`

Return the name of the activity that invoked this activity. This is who the data in `setResult()` ([https://developer.android.com/reference/android/app/Activity.html#setResult\(int\)](https://developer.android.com/reference/android/app/Activity.html#setResult(int))) will be sent to. You can use this information to validate that the recipient is allowed to receive the data.

Note: if the calling activity is not expecting a result (that is it did not use the `startActivityForResult(Intent, int)` ([https://developer.android.com/reference/android/app/Activity.html#startActivityForResult\(android.content.Intent, int\)](https://developer.android.com/reference/android/app/Activity.html#startActivityForResult(android.content.Intent, int))) form that includes a request code), then the calling package will be null.

Returns

`ComponentName`

(<https://developer.android.com/reference/android/content/ComponentName.html>)

The `ComponentName` of the activity that will receive your reply, or null if none.

getCallingPackage

Added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

`String` (<https://developer.android.com/reference/java/lang/String.html>) `getCallingPackage()`

Return the name of the package that invoked this activity. This is who the data in `setResult()` ([https://developer.android.com/reference/android/app/Activity.html#setResult\(int\)](https://developer.android.com/reference/android/app/Activity.html#setResult(int))) will be sent to. You can use this information to validate that the recipient is allowed to receive the data.

Note: if the calling activity is not expecting a result (that is it did not use the `startActivityForResult(Intent, int)` ([https://developer.android.com/reference/android/app/Activity.html#startActivityForResult\(android.content.Intent, int\)](https://developer.android.com/reference/android/app/Activity.html#startActivityForResult(android.content.Intent, int))) form that includes a request code), then the calling package will be null.

Note: prior to `JELLY_BEAN_MR2` (https://developer.android.com/reference/android/os/Build.VERSION_CODES.html#JELLY_BEAN_MR2), the result from this method was unstable. If the process hosting the calling package was no longer running, it would return null instead of the proper package name. You can use `getCallingActivity()` ([https://developer.android.com/reference/android/app/Activity.html#getCallingActivity\(\)](https://developer.android.com/reference/android/app/Activity.html#getCallingActivity())) and retrieve the package name from

This site uses cookies to store your preferences for site-specific language options.

Returns	
String (https://developer.android.com/reference/java/lang/String.html) 	The package of the activity that will receive your reply, or null if none.

getChangingConfigurations

Added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
int getChangingConfigurations ()
```

If this activity is being destroyed because it can not handle a configuration parameter being changed (and thus its `onConfigurationChanged(Configuration)` ([https://developer.android.com/reference/android/app/Activity.html#onConfigurationChanged\(android.content.res.Configuration\)](https://developer.android.com/reference/android/app/Activity.html#onConfigurationChanged(android.content.res.Configuration))) method is *not* being called), then you can use this method to discover the set of changes that have occurred while in the process of being destroyed. Note that there is no guarantee that these will be accurate (other changes could have happened at any time), so you should only use this as an optimization hint.

Returns	
int	Returns a bit field of the configuration parameters that are changing, as defined by the Configuration (https://developer.android.com/reference/android/content/res/Configuration.html) class.

getComponentName

Added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
ComponentName (https://developer.android.com/reference/android/content/ComponentName.html) getCon
```

Returns complete component name of this activity.

Returns	
ComponentName (https://developer.android.com/reference/android/content 	Returns the complete component name for this activity

This site uses cookies to store your preferences for site-specific language options.

OK

getContentScene

Added in API level 21 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

`Scene` (<https://developer.android.com/reference/android/transition/Scene.html>) `getContentScene ()`

Retrieve the `Scene` (<https://developer.android.com/reference/android/transition/Scene.html>) representing this window's current content. Requires `FEATURE_CONTENT_TRANSITIONS` (https://developer.android.com/reference/android/view/Window.html#FEATURE_CONTENT_TRANSITIONS).

This method will return null if the current content is not represented by a Scene.

Returns	
<code>Scene</code> (https://developer.android.com/reference/android/transition/Scene.html)	Current Scene being shown or null

getContentTransitionManager

Added in API level 21 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

`TransitionManager` (<https://developer.android.com/reference/android/transition/TransitionManager.html>)

Retrieve the `TransitionManager` (<https://developer.android.com/reference/android/transition/TransitionManager.html>) responsible for default transitions in this window. Requires `FEATURE_CONTENT_TRANSITIONS` (https://developer.android.com/reference/android/view/Window.html#FEATURE_CONTENT_TRANSITIONS).

This method will return non-null after content has been initialized (e.g. by using `setContentView(View)` ([https://developer.android.com/reference/android/app/Activity.html#setContentView\(android.view.View\)](https://developer.android.com/reference/android/app/Activity.html#setContentView(android.view.View))) if `FEATURE_CONTENT_TRANSITIONS` (https://developer.android.com/reference/android/view/Window.html#FEATURE_CONTENT_TRANSITIONS) has been granted.

Returns	
<code>TransitionManager</code> (https://developer.android.com/reference/android/transition/TransitionManager.html)	This window's content TransitionManager or null if none is set.

This site uses cookies to store your preferences for site-specific language options.

OK

getCurrentFocus

Added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

`View` (<https://developer.android.com/reference/android/view/View.html>) `getCurrentFocus()`

Calls `getCurrentFocus()` ([https://developer.android.com/reference/android/view/Window.html#getCurrentFocus\(\)](https://developer.android.com/reference/android/view/Window.html#getCurrentFocus())) on the `Window` of this `Activity` to return the currently focused view.

Returns

`View`

(<https://developer.android.com/reference/android/view/View.html>)

View The current View with focus or null.

See also:

`getWindow()` ([https://developer.android.com/reference/android/app/Activity.html#getWindow\(\)](https://developer.android.com/reference/android/app/Activity.html#getWindow()))

`getCurrentFocus()` ([https://developer.android.com/reference/android/view/Window.html#getCurrentFocus\(\)](https://developer.android.com/reference/android/view/Window.html#getCurrentFocus()))

getFragmentManager

Added in API level 11 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

`FragmentManager` (<https://developer.android.com/reference/android/app/FragmentManager.html>) `getFr`

Return the `FragmentManager` for interacting with fragments associated with this activity.

Returns

`FragmentManager`

(<https://developer.android.com/reference/android/app/FragmentManager.html>)

getIntent

Added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

This site uses cookies to store your preferences for site-specific language options.

OK

Returns	
Intent (https://developer.android.com/reference/android/content/Intent.html)	

getLastNonConfigurationInstance

Added in API level <https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>

Object (<https://developer.android.com/reference/java/lang/Object.html>) `getLastNonConfiguration`

Retrieve the non-configuration instance data that was previously returned by

`onRetainNonConfigurationInstance()` ([https://developer.android.com/reference/android/app/Activity.html#onRetainNonConfigurationInstance\(\)](https://developer.android.com/reference/android/app/Activity.html#onRetainNonConfigurationInstance())). This will be available from the initial `onCreate(Bundle)` ([https://developer.android.com/reference/android/app/Activity.html#onCreate\(android.os.Bundle\)](https://developer.android.com/reference/android/app/Activity.html#onCreate(android.os.Bundle))) and `onStart()` ([https://developer.android.com/reference/android/app/Activity.html#onStart\(\)](https://developer.android.com/reference/android/app/Activity.html#onStart())) calls to the new instance, allowing you to extract any useful dynamic state from the previous instance.

Note that the data you retrieve here should *only* be used as an optimization for handling configuration changes. You should always be able to handle getting a null pointer back, and an activity must still be able to restore itself to its previous state (through the normal `onSaveInstanceState(Bundle)` ([https://developer.android.com/reference/android/app/Activity.html#onSaveInstanceState\(android.os.Bundle\)](https://developer.android.com/reference/android/app/Activity.html#onSaveInstanceState(android.os.Bundle))) mechanism) even if this function returns null.

Note: For most cases you should use the **Fragment** (<https://developer.android.com/reference/android/app/Fragment.html>) API `setRetainInstance(boolean)` ([https://developer.android.com/reference/android/app/Fragment.html#setRetainInstance\(boolean\)](https://developer.android.com/reference/android/app/Fragment.html#setRetainInstance(boolean))) instead; this is also available on older platforms through the Android support libraries.

Returns	
Object (https://developer.android.com/reference/java/lang/Object.html)	the object previously returned by <code>onRetainNonConfigurationInstance()</code> (https://developer.android.com/reference/android/app/Activity.html#onRetainNonConfigurationInstance())

This site uses cookies to store your preferences for site-specific language options.

OK

getLayoutInflater Added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

`LayoutInflater` (<https://developer.android.com/reference/android/view/LayoutInflater.html>) `getLayoutInflater()`

Convenience for calling `getLayoutInflater()` ([https://developer.android.com/reference/android/view/Window.html#getLayoutInflater\(\)](https://developer.android.com/reference/android/view/Window.html#getLayoutInflater())).

Returns	
<code>LayoutInflater</code> (https://developer.android.com/reference/android/view/LayoutInflater.html)	

getLoaderManager Added in API level 11 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

`LoaderManager` (<https://developer.android.com/reference/android/app/LoaderManager.html>) `getLoaderManager()`

Return the `LoaderManager` for this activity, creating it if needed.

Returns	
<code>LoaderManager</code> (https://developer.android.com/reference/android/app/LoaderManager.html)	

getLocalClassName Added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

`String` (<https://developer.android.com/reference/java/lang/String.html>) `getLocalClassName()`

Returns class name for this activity with the package prefix removed. This is the default name used to read and write settings.

Returns	
<code>String</code>	The local class name.

This site uses cookies to store your preferences for site-specific language options.

OK

`/String.html)`

getMediaController

Added in API level 21 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)`MediaController` (<https://developer.android.com/reference/android/media/session/MediaController.html>)

Gets the controller which should be receiving media key and volume events while this activity is in the foreground.

Returns

`MediaController`

(<https://developer.android.com/reference/android/media/session/MediaController.html>)

The controller which should receive events.

See also:

`setMediaController(android.media.session.MediaController)`

([https://developer.android.com/reference/android](https://developer.android.com/reference/android/app/Activity.html#setMediaController(android.media.session.MediaController))

[/app/Activity.html#setMediaController\(android.media.session.MediaController\)](https://developer.android.com/reference/android/app/Activity.html#setMediaController(android.media.session.MediaController)))

getMenuInflater

Added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)`MenuInflater` (<https://developer.android.com/reference/android/view/MenuInflater.html>) `getMenuInf`

Returns a `MenuInflater` (<https://developer.android.com/reference/android/view/MenuInflater.html>) with this context.

Returns

`MenuInflater`

(<https://developer.android.com/reference/android/view/MenuInflater.html>)

This site uses cookies to store your preferences for site-specific language options.

getParent

Added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

`Activity` (<https://developer.android.com/reference/android/app/Activity.html>) `getParent ()`

Return the parent activity if this view is an embedded child.

Returns	
<code>Activity</code> (https://developer.android.com/reference/android/app/Activity.html)	

getParentActivityIntent

Added in API level 16 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

`Intent` (<https://developer.android.com/reference/android/content/Intent.html>) `getParentActivityIntent()`

Obtain an `Intent` (<https://developer.android.com/reference/android/content/Intent.html>) that will launch an explicit target activity specified by this activity's logical parent. The logical parent is named in the application's manifest by the `parentActivityName` (<https://developer.android.com/reference/android/R.attr.html#parentActivityName>) attribute. Activity subclasses may override this method to modify the Intent returned by `super.getParentActivityIntent()` or to implement a different mechanism of retrieving the parent intent entirely.

Returns	
<code>Intent</code> (https://developer.android.com/reference/android/content/Intent.html)	a new Intent targeting the defined parent of this activity or null if there is no valid parent.

getPreferences

Added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

`SharedPreferences` (<https://developer.android.com/reference/android/content/SharedPreferences.html>)

Retrieve a `SharedPreferences` (<https://developer.android.com/reference/android/content/SharedPreferences.html>)

This site uses cookies to store your preferences for site-specific language options.

OK

`/reference/android/content/ContextWrapper.html#getSharedPreferences(java.lang.String, int))` method by passing in this activity's class name as the preferences name.

Parameters	
<code>mode</code>	<code>int</code> : Operating mode. Use <code>MODE_PRIVATE</code> (https://developer.android.com/reference/android/content/Context.html#MODE_PRIVATE) for the default operation.
Returns	
<code>SharedPreferences</code> (https://developer.android.com/reference/android/content/SharedPreferences.html)	Returns the single <code>SharedPreferences</code> instance that can be used to retrieve and modify the preference values.

getReferrer

Added in API level 22 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

`Uri` (<https://developer.android.com/reference/android/net/Uri.html>) `getReferrer ()`

Return information about who launched this activity. If the launching Intent contains an `Intent.EXTRA_REFERRER` (https://developer.android.com/reference/android/content/Intent.html#EXTRA_REFERRER), that will be returned as-is; otherwise, if known, an `android-app:` (https://developer.android.com/reference/android/content/Intent.html#URI_ANDROID_APP_SCHEME) referrer URI containing the package name that started the Intent will be returned. This may return null if no referrer can be identified -- it is neither explicitly specified, nor is it known which application package was involved.

If called while inside the handling of `onNewIntent(Intent)` ([https://developer.android.com/reference/android/app/Activity.html#onNewIntent\(android.content.Intent\)](https://developer.android.com/reference/android/app/Activity.html#onNewIntent(android.content.Intent))), this function will return the referrer that submitted that new intent to the activity. Otherwise, it always returns the referrer of the original Intent.

Note that this is *not* a security feature -- you can not trust the referrer information, applications can spoof it.

Returns	
<code>Uri</code> (https://developer.android.com)	

This site uses cookies to store your preferences for site-specific langu options.

OK

getRequestedOrientation

Added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
int getRequestedOrientation ()
```

Return the current requested orientation of the activity. This will either be the orientation requested in its component's manifest, or the last requested orientation given to

[setRequestedOrientation\(int\)](https://developer.android.com/reference/android/app/Activity.html#setRequestedOrientation(int)) ([https://developer.android.com/reference/android/app/Activity.html#setRequestedOrientation\(int\)](https://developer.android.com/reference/android/app/Activity.html#setRequestedOrientation(int))).

Returns

<code>int</code>	Returns an orientation constant as used in <code>ActivityInfo.screenOrientation</code> (https://developer.android.com/reference/android/content/pm/ActivityInfo.html#screenOrientation).
------------------	---

getSearchEvent

Added in API level 23 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

`SearchEvent` (<https://developer.android.com/reference/android/view/SearchEvent.html>) `getSearchEvent`

During the `onSearchRequested()` callbacks, this function will return the `SearchEvent` (<https://developer.android.com/reference/android/view/SearchEvent.html>) that triggered the callback, if it exists.

Returns

<code>SearchEvent</code> (https://developer.android.com/reference/android/view/SearchEvent.html)	<code>SearchEvent</code> The <code>SearchEvent</code> that triggered the <code>onSearchRequested()</code> (https://developer.android.com/reference/android/app/Activity.html#onSearchRequested()) callback.
---	---

getSystemService

Added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

`Object` (<https://developer.android.com/reference/java/lang/Object.html>) `getSystemService` (`String`)

Return the handle to a system-level service by name. The class of the returned object varies by the requested name. Currently available names are:

This site uses cookies to store your preferences for site-specific language options.

OK

`/Context.html#WINDOW_SERVICE`) ("window")

The top-level window manager in which you can place custom windows. The returned object is a `WindowManager` (<https://developer.android.com/reference/android/view/WindowManager.html>).

`LAYOUT_INFLATER_SERVICE` (https://developer.android.com/reference/android/content/Context.html#LAYOUT_INFLATER_SERVICE) ("layout_inflater")

A `LayoutInflater` (<https://developer.android.com/reference/android/view/LayoutInflater.html>) for inflating layout resources in this context.

`ACTIVITY_SERVICE` (https://developer.android.com/reference/android/content/Context.html#ACTIVITY_SERVICE) ("activity")

A `ActivityManager` (<https://developer.android.com/reference/android/app/ActivityManager.html>) for interacting with the global activity state of the system.

`POWER_SERVICE` (https://developer.android.com/reference/android/content/Context.html#POWER_SERVICE) ("power")

A `PowerManager` (<https://developer.android.com/reference/android/os/PowerManager.html>) for controlling power management.

`ALARM_SERVICE` (https://developer.android.com/reference/android/content/Context.html#ALARM_SERVICE) ("alarm")

A `AlarmManager` (<https://developer.android.com/reference/android/app/AlarmManager.html>) for receiving intents at the time of your choosing.

`NOTIFICATION_SERVICE` (https://developer.android.com/reference/android/content/Context.html#NOTIFICATION_SERVICE) ("notification")

A `NotificationManager` (<https://developer.android.com/reference/android/app/NotificationManager.html>) for informing the user of background events.

`KEYGUARD_SERVICE` (https://developer.android.com/reference/android/content/Context.html#KEYGUARD_SERVICE) ("keyguard")

A `KeyguardManager` (<https://developer.android.com/reference/android/app/KeyguardManager.html>) for controlling keyguard.

This site uses cookies to store your preferences for site-specific language options.

`/Context.html#LOCATION_SERVICE)` ("location")

A **LocationManager** (<https://developer.android.com/reference/android/location/LocationManager.html>) for controlling location (e.g., GPS) updates.

`SEARCH_SERVICE` (https://developer.android.com/reference/android/content/Context.html#SEARCH_SERVICE) ("search")

A **SearchManager** (<https://developer.android.com/reference/android/app/SearchManager.html>) for handling search.

`VIBRATOR_SERVICE` (https://developer.android.com/reference/android/content/Context.html#VIBRATOR_SERVICE) ("vibrator")

A **Vibrator** (<https://developer.android.com/reference/android/os/Vibrator.html>) for interacting with the vibrator hardware.

`CONNECTIVITY_SERVICE` (https://developer.android.com/reference/android/content/Context.html#CONNECTIVITY_SERVICE) ("connection")

A **ConnectivityManager** (<https://developer.android.com/reference/android/net/ConnectivityManager.html>) for handling management of network connections.

`WIFI_SERVICE` (https://developer.android.com/reference/android/content/Context.html#WIFI_SERVICE) ("wifi")

A **WifiManager** (<https://developer.android.com/reference/android/net/wifi/WifiManager.html>) for management of Wi-Fi connectivity. On releases before NYC, it should only be obtained from an application context, and not from any other derived context to avoid memory leaks within the calling process.

`WIFI_P2P_SERVICE` (https://developer.android.com/reference/android/content/Context.html#WIFI_P2P_SERVICE) ("wifip2p")

A **WifiP2pManager** (<https://developer.android.com/reference/android/net/wifi/p2p/WifiP2pManager.html>) for management of Wi-Fi Direct connectivity.

`INPUT_METHOD_SERVICE` (https://developer.android.com/reference/android/content/Context.html#INPUT_METHOD_SERVICE) ("input_method")

This site uses cookies to store your preferences for site-specific language options.

UI_MODE_SERVICE (https://developer.android.com/reference/android/content/Context.html#UI_MODE_SERVICE) ("uimode")

An **UiModeManager** (<https://developer.android.com/reference/android/app/UiModeManager.html>) for controlling UI modes.

DOWNLOAD_SERVICE (https://developer.android.com/reference/android/content/Context.html#DOWNLOAD_SERVICE) ("download")

A **DownloadManager** (<https://developer.android.com/reference/android/app/DownloadManager.html>) for requesting HTTP downloads

BATTERY_SERVICE (https://developer.android.com/reference/android/content/Context.html#BATTERY_SERVICE) ("batterymanager")

A **BatteryManager** (<https://developer.android.com/reference/android/os/BatteryManager.html>) for managing battery state

JOB_SCHEDULER_SERVICE (https://developer.android.com/reference/android/content/Context.html#JOB_SCHEDULER_SERVICE) ("taskmanager")

A **JobScheduler** (<https://developer.android.com/reference/android/app/job/JobScheduler.html>) for managing scheduled tasks

NETWORK_STATS_SERVICE (https://developer.android.com/reference/android/content/Context.html#NETWORK_STATS_SERVICE) ("netstats")

A **NetworkStatsManager** (<https://developer.android.com/reference/android/app/usage/NetworkStatsManager.html>) for querying network usage statistics.

HARDWARE_PROPERTIES_SERVICE (https://developer.android.com/reference/android/content/Context.html#HARDWARE_PROPERTIES_SERVICE) ("hardware_properties")

A **HardwarePropertiesManager** (<https://developer.android.com/reference/android/os/HardwarePropertiesManager.html>) for accessing hardware properties.

Note: System services obtained via this API may be closely associated with the Context in which they are obtained from. In general, do not share the service objects between various different contexts (Activities, Applications, Services, Providers, etc.)

Parameters

This site uses cookies to store your preferences for site-specific language options.

OK

Returns

Object

(<https://developer.android.com/reference/java/lang/Object.html>)

The service or null if the name does not exist.

getTaskId

Added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
int getTaskId ()
```

Return the identifier of the task this activity is in. This identifier will remain the same for the lifetime of the activity.

Returns

int

Task identifier, an opaque integer.

getTitle

Added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
CharSequence (https://developer.android.com/reference/java/lang/CharSequence.html) getTitle ()
```

Returns

CharSequence

(<https://developer.android.com/reference/java/lang/CharSequence.html>)

getTitleColor

Added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
int getTitleColor ()
```

This site uses cookies to store your preferences for site-specific language options.

OK

getVoiceInteractor

Added in API level 23 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

`VoiceInteractor` (<https://developer.android.com/reference/android/app/VoiceInteractor.html>) `getVo`.

Retrieve the active `VoiceInteractor` (<https://developer.android.com/reference/android/app/VoiceInteractor.html>) that the user is going through to interact with this activity.

Returns

`VoiceInteractor`
(<https://developer.android.com/reference/android/app/VoiceInteractor.html>)

getVolumeControlStream

Added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

`int` `getVolumeControlStream` ()

Gets the suggested audio stream whose volume should be changed by the hardware volume controls.

Returns

<code>int</code>	The suggested audio stream type whose volume should be changed by the hardware volume controls.
------------------	---

See also:

`setVolumeControlStream(int)` ([https://developer.android.com/reference/android/app/Activity.html#setVolumeControlStream\(int\)](https://developer.android.com/reference/android/app/Activity.html#setVolumeControlStream(int)))

getWindow

Added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

`Window` (<https://developer.android.com/reference/android/view/Window.html>) `getWindow` ()

Retrieve the current `Window` (<https://developer.android.com/reference/android/view/Window.html>) for the activity. This can be used to directly access parts of the Window API that are not available

This site uses cookies to store your preferences for site-specific language options.

OK

Returns	
Window (https://developer.android.com/reference/android/view/Window.html) 	Window The current window, or null if the activity is not visual.

getWindowManager

Added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

`WindowManager` (<https://developer.android.com/reference/android/view/WindowManager.html>) `getWindowManager()`

Retrieve the window manager for showing custom windows.

Returns	
WindowManager (https://developer.android.com/reference/android/view/WindowManager.html) 	

hasWindowFocus

Added in API level 3 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

`boolean` `hasWindowFocus()`

Returns true if this activity's *main* window currently has window focus. Note that this is not the same as the view itself having focus.

Returns	
boolean	True if this activity's main window currently has window focus.

See also:

`onWindowAttributesChanged(android.view.WindowManager.LayoutParams)`
[\(https://developer.android.com/reference/android/app/Activity.html#onWindowAttributesChanged\(android.view.WindowManager.LayoutParams\)\)](https://developer.android.com/reference/android/app/Activity.html#onWindowAttributesChanged(android.view.WindowManager.LayoutParams))

This site uses cookies to store your preferences for site-specific language options.

OK

invalidateOptionsMenu

Added in API level 11 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
void invalidateOptionsMenu ()
```

Declare that the options menu has changed, so should be recreated. The

`onCreateOptionsMenu(Menu)` ([https://developer.android.com/reference/android/app/Activity.html#onCreateOptionsMenu\(android.view.Menu\)](https://developer.android.com/reference/android/app/Activity.html#onCreateOptionsMenu(android.view.Menu))) method will be called the next time it needs to be displayed.

isChangingConfigurations

Added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
boolean isChangingConfigurations ()
```

Check to see whether this activity is in the process of being destroyed in order to be recreated with a new configuration. This is often used in `onStop()` ([https://developer.android.com/reference/android/app/Activity.html#onStop\(\)](https://developer.android.com/reference/android/app/Activity.html#onStop())) to determine whether the state needs to be cleaned up or will be passed on to the next instance of the activity via `onRetainNonConfigurationInstance()` ([https://developer.android.com/reference/android/app/Activity.html#onRetainNonConfigurationInstance\(\)](https://developer.android.com/reference/android/app/Activity.html#onRetainNonConfigurationInstance())).

Returns

`boolean`

If the activity is being torn down in order to be recreated with a new configuration, returns true; else returns false.

isChild

Added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
boolean isChild ()
```

Is this activity embedded inside of another activity?

Returns

`boolean`

This site uses cookies to store your preferences for site-specific language options.

OK


```
boolean isDestroyed ()
```

Returns true if the final `onDestroy()` ([https://developer.android.com/reference/android/app/Activity.html#onDestroy\(\)](https://developer.android.com/reference/android/app/Activity.html#onDestroy())) call has been made on the Activity, so this instance is now dead.

Returns	
boolean	

isFinishing

Added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
boolean isFinishing ()
```

Check to see whether this activity is in the process of finishing, either because you called `finish()` ([https://developer.android.com/reference/android/app/Activity.html#finish\(\)](https://developer.android.com/reference/android/app/Activity.html#finish())) on it or someone else has requested that it finished. This is often used in `onPause()` ([https://developer.android.com/reference/android/app/Activity.html#onPause\(\)](https://developer.android.com/reference/android/app/Activity.html#onPause())) to determine whether the activity is simply pausing or completely finishing.

Returns	
boolean	If the activity is finishing, returns true; else returns false.

See also:

`finish()` ([https://developer.android.com/reference/android/app/Activity.html#finish\(\)](https://developer.android.com/reference/android/app/Activity.html#finish()))

isImmersive

Added in API level 18 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
boolean isImmersive ()
```

Bit indicating that this activity is "immersive" and should not be interrupted by notifications if possible. This value is initially set by the manifest property `android:immersive` but may be changed at runtime by `setImmersive(boolean)` ([https://developer.android.com/reference/android/app/Activity.html#setImmersive\(boolean\)](https://developer.android.com/reference/android/app/Activity.html#setImmersive(boolean))).

Returns	

This site uses cookies to store your preferences for site-specific language options.

OK

See also:

`setImmersive(boolean)` ([https://developer.android.com/reference/android/app/Activity.html#setImmersive\(boolean\)](https://developer.android.com/reference/android/app/Activity.html#setImmersive(boolean)))

`FLAG_IMMERSIVE` (https://developer.android.com/reference/android/content/pm/ActivityInfo.html#FLAG_IMMERSIVE)

isInMultiWindowMode

Added in API level 24 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
boolean isInMultiWindowMode ()
```

Returns true if the activity is currently in multi-window mode.

Returns	
<code>boolean</code>	True if the activity is in multi-window mode.

See also:

`resizeableActivity` (<https://developer.android.com/reference/android/R.attr.html#resizeableActivity>)

isInPictureInPictureMode

Added in API level 24 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
boolean isInPictureInPictureMode ()
```

Returns true if the activity is currently in picture-in-picture mode.

Returns	
<code>boolean</code>	True if the activity is in picture-in-picture mode.

See also:

`supportsPictureInPicture` (<https://developer.android.com/reference/android/R.attr.html#supportsPictureInPicture>)

This site uses cookies to store your preferences for site-specific language options. OK

isLocalVoiceInteractionSupported

Added in API level 24 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
boolean isLocalVoiceInteractionSupported ()
```

Queries whether the currently enabled voice interaction service supports returning a voice interactor for use by the activity. This is valid only for the duration of the activity.

Returns	
boolean	whether the current voice interaction service supports local voice interaction

isTaskRoot

Added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
boolean isTaskRoot ()
```

Return whether this activity is the root of a task. The root is the first activity in a task.

Returns	
boolean	True if this is the root activity, else false.

isVoiceInteraction

Added in API level 23 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
boolean isVoiceInteraction ()
```

Check whether this activity is running as part of a voice interaction with the user. If true, it should perform its interaction with the user through the **VoiceInteractor** (<https://developer.android.com/reference/android/app/VoiceInteractor.html>) returned by **getVoiceInteractor()** ([https://developer.android.com/reference/android/app/Activity.html#getVoiceInteractor\(\)](https://developer.android.com/reference/android/app/Activity.html#getVoiceInteractor())).

Returns	
boolean	

isVoiceInteractionRoot

This site uses cookies to store your preferences for site-specific language options.

OK

Like `isVoiceInteraction()` ([https://developer.android.com/reference/android/app/Activity.html#isVoiceInteraction\(\)](https://developer.android.com/reference/android/app/Activity.html#isVoiceInteraction())), but only returns true if this is also the root of a voice interaction. That is, returns true if this activity was directly started by the voice interaction service as the initiation of a voice interaction. Otherwise, for example if it was started by another activity while under voice interaction, returns false.

Returns	
<code>boolean</code>	

managedQuery Added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

`Cursor` (<https://developer.android.com/reference/android/database/Cursor.html>) `managedQuery` (`Uri` `String[]` (<https://developer.android.com/reference/java/lang/String.html>) `projection` `String` (<https://developer.android.com/reference/java/lang/String.html>) `selection` `String[]` (<https://developer.android.com/reference/java/lang/String.html>) `selectionArgs` `String` (<https://developer.android.com/reference/java/lang/String.html>) `sortOrder`

This method was deprecated in API level 11.

Use `CursorLoader` (<https://developer.android.com/reference/android/content/CursorLoader.html>) instead.

Wrapper around `query(android.net.Uri, String[], String, String[], String)` ([https://developer.android.com/reference/android/content/ContentResolver.html#query\(android.net.Uri, java.lang.String\[\], java.lang.String, java.lang.String\[\], java.lang.String\)](https://developer.android.com/reference/android/content/ContentResolver.html#query(android.net.Uri, java.lang.String[], java.lang.String, java.lang.String[], java.lang.String))) that gives the resulting `Cursor` (<https://developer.android.com/reference/android/database/Cursor.html>) to call `startManagingCursor(Cursor)` ([https://developer.android.com/reference/android/app/Activity.html#startManagingCursor\(android.database.Cursor\)](https://developer.android.com/reference/android/app/Activity.html#startManagingCursor(android.database.Cursor))) so that the activity will manage its lifecycle for you. *If you are targeting HONEYCOMB* (https://developer.android.com/reference/android/os/Build.VERSION_CODES.html#HONEYCOMB) *or later, consider instead using LoaderManager* (<https://developer.android.com/reference/android/app/LoaderManager.html>) *instead, available via* `getLoaderManager()` ([https://developer.android.com/reference/android/app/Activity.html#getLoaderManager\(\)](https://developer.android.com/reference/android/app/Activity.html#getLoaderManager())).

Warning: Do not call `close()` ([https://developer.android.com/reference/android/database/Cursor.html#close\(\)](https://developer.android.com/reference/android/database/Cursor.html#close())) on a cursor obtained using this method, because the activity will do that for you at the appropriate time. However, if you call `stopManagingCursor(Cursor)` ([https://developer.android.com/reference/android/app/Activity.html#stopManagingCursor\(android.database.Cursor\)](https://developer.android.com/reference/android/app/Activity.html#stopManagingCursor(android.database.Cursor)))

This site uses cookies to store your preferences for site-specific language options.

[https://developer.android.com/reference/android/database/Cursor.html#close\(\)](https://developer.android.com/reference/android/database/Cursor.html#close())).

Parameters	
<code>uri</code>	Uri : The URI of the content provider to query.
<code>projection</code>	String : List of columns to return.
<code>selection</code>	String : SQL WHERE clause.
<code>selectionArgs</code>	String : The arguments to selection, if any ?s are present
<code>sortOrder</code>	String : SQL ORDER BY clause.

Returns	
Cursor https://developer.android.com/reference/android/database/Cursor.html	The Cursor that was returned by <code>query()</code> .

See also:

`query(android.net.Uri, String[], String, String[], String)`

[https://developer.android.com/reference/android/content/ContentResolver.html#query\(android.net.Uri,java.lang.String\[\], java.lang.String, java.lang.String\[\], java.lang.String\)\)](https://developer.android.com/reference/android/content/ContentResolver.html#query(android.net.Uri,java.lang.String[],java.lang.String,java.lang.String[],java.lang.String))

`startManagingCursor(Cursor)` [https://developer.android.com/reference/android/app/Activity.html#startManagingCursor\(android.database.Cursor\)](https://developer.android.com/reference/android/app/Activity.html#startManagingCursor(android.database.Cursor))

moveTaskToBack

Added in API level 1 <https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>

`boolean moveTaskToBack (boolean nonRoot)`

Move the task containing this activity to the back of the activity stack. The activity's order within the task is unchanged.

Parameters	
<code>nonRoot</code>	boolean : If false then this only works if the activity is the root of a task; if true it will work for any activity in a task.

This site uses cookies to store your preferences for site-specific language options.

<code>boolean</code>	If the task was moved (or it was already at the back) true is returned, else false.
----------------------	---

navigateUpTo

Added in API level 16 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
boolean navigateUpTo (Intent (https://developer.android.com/reference/android/content/Intent.html)
```

Navigate from this activity to the activity specified by `upIntent`, finishing this activity in the process. If the activity indicated by `upIntent` already exists in the task's history, this activity and all others before the indicated activity in the history stack will be finished.

If the indicated activity does not appear in the history stack, this will finish each activity in this task until the root activity of the task is reached, resulting in an "in-app home" behavior. This can be useful in apps with a complex navigation hierarchy when an activity may be reached by a path not passing through a canonical parent activity.

This method should be used when performing up navigation from within the same task as the destination. If up navigation should cross tasks in some cases, see

`shouldUpRecreateTask(Intent) (https://developer.android.com/reference/android/app/Activity.html#shouldUpRecreateTask\(android.content.Intent\)\).`

Parameters

<code>upIntent</code>	Intent: An intent representing the target destination for up navigation
-----------------------	--

Returns

<code>boolean</code>	true if up navigation successfully reached the activity indicated by <code>upIntent</code> and <code>upIntent</code> was delivered to it. false if an instance of the indicated activity could not be found and this activity was simply finished normally.
----------------------	---

navigateUpToFromChild

Added in API level 16 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
boolean navigateUpToFromChild (Activity (https://developer.android.com/reference/android/app/Activity.html) Intent (https://developer.android.com/reference/android/content/Intent.html) upIntent)
```

This is called when a child activity of this one calls its `navigateUpTo(Intent)`

This site uses cookies to store your preferences for site-specific language options.

OK

calls `navigateUpTo(upIntent)` on this activity (the parent).

Parameters	
<code>child</code>	Activity: The activity making the call.
<code>upIntent</code>	Intent: An intent representing the target destination for up navigation

Returns	
<code>boolean</code>	true if up navigation successfully reached the activity indicated by <code>upIntent</code> and <code>upIntent</code> was delivered to it. false if an instance of the indicated activity could not be found and this activity was simply finished normally.

onActionModeFinished

Added in API level 11 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
void onActionModeFinished (ActionMode (https://developer.android.com/reference/android/view/
```

Notifies the activity that an action mode has finished. Activity subclasses overriding this method should call the superclass implementation.

Parameters	
<code>mode</code>	ActionMode: The action mode that just finished.

onActionModeStarted

Added in API level 11 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
void onActionModeStarted (ActionMode (https://developer.android.com/reference/android/view/Ac
```

Notifies the Activity that an action mode has been started. Activity subclasses overriding this method should call the superclass implementation.

Parameters	
<code>mode</code>	ActionMode: The new action mode.

This site uses cookies to store your preferences for site-specific language options.

```
void onActivityReenter (int resultCode,
                        Intent (https://developer.android.com/reference/android/content/Intent.html) data)
```

Called when an activity you launched with an activity transition exposes this Activity through a returning activity transition, giving you the resultCode and any additional data from it. This method will only be called if the activity set a result code other than `RESULT_CANCELED` (https://developer.android.com/reference/android/app/Activity.html#RESULT_CANCELED) and it supports activity transitions with `FEATURE_ACTIVITY_TRANSITIONS` (https://developer.android.com/reference/android/view/Window.html#FEATURE_ACTIVITY_TRANSITIONS).

The purpose of this function is to let the called Activity send a hint about its state so that this underlying Activity can prepare to be exposed. A call to this method does not guarantee that the called Activity has or will be exiting soon. It only indicates that it will expose this Activity's Window and it has some data to pass to prepare it.

Parameters	
<code>resultCode</code>	<code>int</code> : The integer result code returned by the child activity through its <code>setResult()</code> .
<code>data</code>	<code>Intent</code> : An Intent, which can return result data to the caller (various data can be attached to Intent "extras").

onAttachFragment

Added in API level 11 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
void onAttachFragment (Fragment (https://developer.android.com/reference/android/app/Fragment.
```

Called when a Fragment is being attached to this activity, immediately after the call to its `Fragment.onAttach()` ([https://developer.android.com/reference/android/app/Fragment.html#onAttach\(android.app.Activity\)](https://developer.android.com/reference/android/app/Fragment.html#onAttach(android.app.Activity))) method and before `Fragment.onCreate()` ([https://developer.android.com/reference/android/app/Fragment.html#onCreate\(android.os.Bundle\)](https://developer.android.com/reference/android/app/Fragment.html#onCreate(android.os.Bundle))).

Parameters	
<code>fragment</code>	<code>Fragment</code>

onAttachedToWindow

This site uses cookies to store your preferences for site-specific language options.

OK

Called when the main window associated with the activity has been attached to the window manager. See `View.onAttachedToWindow()` ([https://developer.android.com/reference/android/view/View.html#onAttachedToWindow\(\)](https://developer.android.com/reference/android/view/View.html#onAttachedToWindow())) for more information.

See also:

`onAttachedToWindow()` ([https://developer.android.com/reference/android/view/View.html#onAttachedToWindow\(\)](https://developer.android.com/reference/android/view/View.html#onAttachedToWindow()))

onBackPressed Added in API level 5 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

`void onBackPressed ()`

Called when the activity has detected the user's press of the back key. The default implementation simply finishes the current activity, but you can override this to do whatever you want.

onConfigurationChanged Added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

`void onConfigurationChanged (Configuration (https://developer.android.com/reference/android`

Called by the system when the device configuration changes while your activity is running. Note that this will *only* be called if you have selected configurations you would like to handle with the `configChanges` (<https://developer.android.com/reference/android/R.attr.html#configChanges>) attribute in your manifest. If any configuration change occurs that is not selected to be reported by that attribute, then instead of reporting it the system will stop and restart the activity (to have it launched with the new configuration).

At the time that this function has been called, your Resources object will have been updated to return resource values matching the new configuration.

Parameters	
<code>newConfig</code>	<code>Configuration</code> : The new device configuration.

```
void onContentChanged ()
```

This hook is called whenever the content view of the screen changes (due to a call to `Window setContentView` ([\).](https://developer.android.com/reference/android/view/Window.html#setContentView(android.view.View, android.view.ViewGroup.LayoutParams)) OR <code>Window addContentView</code> (<a href=)

onContextItemSelected Added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
boolean onContextItemSelected (MenuItem (https://developer.android.com/reference/android/view/MenuItem.html#getMenuInfo\(\))
```

This hook is called whenever an item in a context menu is selected. The default implementation simply returns false to have the normal processing happen (calling the item's Runnable or sending a message to its Handler as appropriate). You can use this method for any items for which you would like to do processing without those other facilities.

Use `getMenuInfo()` ([https://developer.android.com/reference/android/view/MenuItem.html#getMenuInfo\(\)](https://developer.android.com/reference/android/view/MenuItem.html#getMenuInfo())) to get extra information set by the View that added this menu item.

Derived classes should call through to the base class for it to perform the default menu handling.

Parameters	
<code>item</code>	<code>MenuItem</code> : The context menu item that was selected.
Returns	
<code>boolean</code>	boolean Return false to allow normal context menu processing to proceed, true to consume it here.

onContextMenuClosed Added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
void onContextMenuClosed (Menu (https://developer.android.com/reference/android/view/Menu.html
```

This hook is called whenever the context menu is being closed (either by the user canceling the

This site uses cookies to store your preferences for site-specific language options.

OK

Parameters	
menu	Menu: The context menu that is being closed.

onCreate

Added in API level 21 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
void onCreate (Bundle (https://developer.android.com/reference/android/os/Bundle.html) savedInstanceState:
    PersistableBundle (https://developer.android.com/reference/android/os/PersistableBundle.html))
```

Same as `onCreate(android.os.Bundle)` ([https://developer.android.com/reference/android/app/Activity.html#onCreate\(android.os.Bundle\)](https://developer.android.com/reference/android/app/Activity.html#onCreate(android.os.Bundle))) but called for those activities created with the attribute `persistableMode` (<https://developer.android.com/reference/android/R.attr.html#persistableMode>) set to `persistAcrossReboots`.

Parameters	
savedInstanceState	Bundle: if the activity is being re-initialized after previously being shut down then this Bundle contains the data it most recently supplied in <code>onSaveInstanceState(Bundle)</code> (https://developer.android.com/reference/android/app/Activity.html#onSaveInstanceState(android.os.Bundle)). Note: Otherwise it is null.
persistentState	PersistableBundle: if the activity is being re-initialized after previously being shut down or powered off then this Bundle contains the data it most recently supplied to <code>outPersistentState</code> in <code>onSaveInstanceState(Bundle)</code> (https://developer.android.com/reference/android/app/Activity.html#onSaveInstanceState(android.os.Bundle)). Note: Otherwise it is null.

See also:

- `onCreate(android.os.Bundle)` ([https://developer.android.com/reference/android/app/Activity.html#onCreate\(android.os.Bundle\)](https://developer.android.com/reference/android/app/Activity.html#onCreate(android.os.Bundle)))
- `onStart()` ([https://developer.android.com/reference/android/app/Activity.html#onStart\(\)](https://developer.android.com/reference/android/app/Activity.html#onStart()))
- `onSaveInstanceState(Bundle)` ([https://developer.android.com/reference/android/app/Activity.html#onSaveInstanceState\(Bundle\)](https://developer.android.com/reference/android/app/Activity.html#onSaveInstanceState(Bundle)))

```
/app/Activity.html#onRestoreInstanceState(android.os.Bundle))

onPostCreate(Bundle) (https://developer.android.com/reference/android
/app/Activity.html#onPostCreate(android.os.Bundle))
```

onCreateContextMenu

Added in API level 1 (https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

```
void onCreateContextMenu (ContextMenu (https://developer.android.com/reference/android/view/View (https://developer.android.com/reference/android/view/View.html) v,
ContextMenu.ContextMenuInfo (https://developer.android.com/reference/android...
```

Called when a context menu for the **view** is about to be shown. Unlike **onCreateOptionsMenu(Menu)** (https://developer.android.com/reference/android/app/Activity.html#onCreateOptionsMenu(android.view.Menu)), this will be called every time the context menu is about to be shown and should be populated for the view (or item inside the view for **AdapterView** (https://developer.android.com/reference/android/widget/AdapterView.html) subclasses, this can be found in the **menuInfo**)).

Use **onContextItemSelected(android.view.MenuItem)** (https://developer.android.com/reference/android/app/Activity.html#onContextItemSelected(android.view.MenuItem)) to know when an item has been selected.

It is not safe to hold onto the context menu after this method returns.

Parameters	
menu	ContextMenu : The context menu that is being built
v	View : The view for which the context menu is being built
menuInfo	ContextMenu.ContextMenuInfo : Extra information about the item for which the context menu should be shown. This information will vary depending on the class of v.

onCreateDescription

Added in API level 1 (https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

```
CharSequence (https://developer.android.com/reference/java/lang/CharSequence.html) onCreateDesc
```

This site uses cookies to store your preferences for site-specific language options.

OK

The default implementation returns null, which will cause you to inherit the description from the previous activity. If all activities return null, generally the label of the top activity will be used as the description.

Returns	
CharSequence https://developer.android.com/reference/java/lang/CharSequence.html	A description of what the user is doing. It should be short and sweet (only a few words).

See also:

onCreateThumbnail(Bitmap, Canvas) ([https://developer.android.com/reference/android/app/Activity.html#onCreateThumbnail\(android.graphics.Bitmap, android.graphics.Canvas\)](https://developer.android.com/reference/android/app/Activity.html#onCreateThumbnail(android.graphics.Bitmap, android.graphics.Canvas)))

onSaveInstanceState(Bundle) ([https://developer.android.com/reference/android/app/Activity.html#onSaveInstanceState\(android.os.Bundle\)](https://developer.android.com/reference/android/app/Activity.html#onSaveInstanceState(android.os.Bundle)))

onPause() ([https://developer.android.com/reference/android/app/Activity.html#onPause\(\)](https://developer.android.com/reference/android/app/Activity.html#onPause()))

onCreateNavigateUpTaskStack

Added in API level 11 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

void onCreateNavigateUpTaskStack (TaskStackBuilder (<https://developer.android.com/referen>

Define the synthetic task stack that will be generated during Up navigation from a different task.

The default implementation of this method adds the parent chain of this activity as specified in the manifest to the supplied **TaskStackBuilder** (<https://developer.android.com/reference/android/app/TaskStackBuilder.html>). Applications may choose to override this method to construct the desired task stack in a different way.

This method will be invoked by the default implementation of **onNavigateUp()**

([https://developer.android.com/reference/android/app/Activity.html#onNavigateUp\(\)](https://developer.android.com/reference/android/app/Activity.html#onNavigateUp())) if

shouldUpRecreateTask(Intent) ([https://developer.android.com/reference/android](https://developer.android.com/reference/android/app/Activity.html#shouldUpRecreateTask(android.content.Intent))

shouldUpRecreateTask(android.content.Intent)) returns true when supplied with the intent returned by **getParentActivityIntent()** ([https://developer.android.com/reference/android/app/Activity.html#getParentActivityIntent\(\)](https://developer.android.com/reference/android/app/Activity.html#getParentActivityIntent())).

Applications that wish to supply extra Intent parameters to the parent stack defined by the manifest

This site uses cookies to store your preferences for site-specific language options.

```
/app/Activity.html#onPrepareNavigateUpTaskStack(android.app.TaskStackBuilder)).
```

Parameters	
<code>builder</code>	TaskStackBuilder : An empty TaskStackBuilder - the application should add intents representing the desired task stack

onCreateOptionsMenu

boolean onCreateOptionsMenu (Menu [Menu](https://developer.android.com/reference/android/view/Menu) (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>))

Initialize the contents of the Activity's standard options menu. You should place your menu items in to **menu**.

This is only called once, the first time the options menu is displayed. To update the menu every time it is displayed, see **onPrepareOptionsMenu(Menu)** ([https://developer.android.com/reference/android/app/Activity.html#onPrepareOptionsMenu\(android.view.Menu\)](https://developer.android.com/reference/android/app/Activity.html#onPrepareOptionsMenu(android.view.Menu))).

The default implementation populates the menu with standard system menu items. These are placed in the **CATEGORY_SYSTEM** (https://developer.android.com/reference/android/view/Menu.html#CATEGORY_SYSTEM) group so that they will be correctly ordered with application-defined menu items. Deriving classes should always call through to the base implementation.

You can safely hold on to **menu** (and any items created from it), making modifications to it as desired, until the next time **onCreateOptionsMenu()** is called.

When you add items to the menu, you can implement the Activity's **onOptionsItemSelected(MenuItem)** ([https://developer.android.com/reference/android/app/Activity.html#onOptionsItemSelected\(android.view.MenuItem\)](https://developer.android.com/reference/android/app/Activity.html#onOptionsItemSelected(android.view.MenuItem))) method to handle them there.

Parameters	
<code>menu</code>	Menu : The options menu in which you place your items.

Returns	
<code>boolean</code>	You must return true for the menu to be displayed; if you return false it will not be shown.

See also:

This site uses cookies to store your preferences for site-specific language options.



```
/app/Activity.html#onPrepareOptionsMenu(android.view.Menu))
```

```
onOptionsItemSelected(MenuItem) (https://developer.android.com/reference/android  
/app/Activity.html#onOptionsItemSelected(android.view.MenuItem))
```

onCreatePanelMenu

Added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
boolean onCreatePanelMenu (int featureId,  
Menu (https://developer.android.com/reference/android/view/Menu.html) menu)
```

Default implementation of `onCreatePanelMenu(int, Menu)` ([https://developer.android.com/reference/android/view/Window.Callback.html#onCreatePanelMenu\(int, android.view.Menu\)](https://developer.android.com/reference/android/view/Window.Callback.html#onCreatePanelMenu(int, android.view.Menu))) for activities. This calls through to the new `onCreateOptionsMenu(Menu)` ([https://developer.android.com/reference/android/app/Activity.html#onCreateOptionsMenu\(android.view.Menu\)](https://developer.android.com/reference/android/app/Activity.html#onCreateOptionsMenu(android.view.Menu))) method for the `FEATURE_OPTIONS_PANEL` (https://developer.android.com/reference/android/view/Window.html#FEATURE_OPTIONS_PANEL) panel, so that subclasses of Activity don't need to deal with feature codes.

Parameters	
<code>featureId</code>	<code>int</code> : The panel being created.
<code>menu</code>	<code>Menu</code> : The menu inside the panel.
Returns	
<code>boolean</code>	boolean You must return true for the panel to be displayed; if you return false it will not be shown.

onCreatePanelView

Added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
View (https://developer.android.com/reference/android/view/View.html) onCreatePanelView (int f
```

Default implementation of `onCreatePanelView(int)` ([https://developer.android.com/reference/android/view/Window.Callback.html#onCreatePanelView\(int\)](https://developer.android.com/reference/android/view/Window.Callback.html#onCreatePanelView(int))) for activities. This simply returns null so that all panel sub-windows will have the default menu behavior.

This site uses cookies to store your preferences for site-specific language options.

OK

<code>featureId</code>	<code>int</code> : Which panel is being created.
Returns	
<code>View</code> (https://developer.android.com/reference/android/view/View.html)	view The top-level view to place in the panel.

onCreateThumbnail

Added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

`boolean onCreateThumbnail (Bitmap (https://developer.android.com/reference/android/graphics/Bitmap.html) Canvas (https://developer.android.com/reference/android/graphics/Canvas.html) Context)`

Generate a new thumbnail for this activity. This method is called before pausing the activity, and should draw into **outBitmap** the imagery for the desired thumbnail in the dimensions of that bitmap. It can use the given **canvas**, which is configured to draw into the bitmap, for rendering if desired.

The default implementation returns fails and does not draw a thumbnail; this will result in the platform creating its own thumbnail if needed.

Parameters	
<code>outBitmap</code>	<code>Bitmap</code> : The bitmap to contain the thumbnail.
<code>canvas</code>	<code>Canvas</code> : Can be used to render into the bitmap.
Returns	
<code>boolean</code>	Return true if you have drawn into the bitmap; otherwise after you return it will be filled with a default thumbnail.

See also:

- `onCreateDescription()` ([https://developer.android.com/reference/android/app/Activity.html#onCreateDescription\(\)](https://developer.android.com/reference/android/app/Activity.html#onCreateDescription()))
- `onSaveInstanceState(Bundle)` ([https://developer.android.com/reference/android/app/Activity.html#onSaveInstanceState\(android.os.Bundle\)](https://developer.android.com/reference/android/app/Activity.html#onSaveInstanceState(android.os.Bundle)))

onCreateView

Added in API level 11 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

`View` (<https://developer.android.com/reference/android/view/View.html>) `onCreateView` (`View` (<https://developer.android.com/reference/android/view/View.html>) `String` (<https://developer.android.com/reference/java/lang/String.html>) `name`, `Context` (<https://developer.android.com/reference/android/content/Context.html>) (`AttributeSet` (<https://developer.android.com/reference/android/util/AttributeSet>

Standard implementation of `onCreateView(View, String, Context, AttributeSet)`

([https://developer.android.com/reference/android/view/LayoutInflater.Factory2.html#onCreateView\(android.view.View, java.lang.String, android.content.Context, android.util.AttributeSet\)](https://developer.android.com/reference/android/view/LayoutInflater.Factory2.html#onCreateView(android.view.View, java.lang.String, android.content.Context, android.util.AttributeSet))) used when inflating with the `LayoutInflater` returned by `getSystemService(Class)` ([https://developer.android.com/reference/android/content/Context.html#getSystemService\(java.lang.Class<T>\)](https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>))). This implementation handles tags to embed fragments inside of the activity.

Parameters	
parent	View: The parent that the created view will be placed in; <i>note that this may be null</i> .
name	String: Tag name to be inflated.
context	Context: The context the view is being created in.
attrs	AttributeSet: Inflation attributes as specified in XML file.

Returns	
<code>View</code> (https://developer.android.com/reference/android/view/View.html)	View Newly created view. Return null for the default behavior.

See also:

`createView(String, String, AttributeSet)` ([https://developer.android.com/reference/android/view/LayoutInflater.html#createView\(java.lang.String, java.lang.String, android.util.AttributeSet\)](https://developer.android.com/reference/android/view/LayoutInflater.html#createView(java.lang.String, java.lang.String, android.util.AttributeSet)))

`getLayoutInflater()` ([https://developer.android.com/reference/android/view/Window.html#getLayoutInflater\(\)](https://developer.android.com/reference/android/view/Window.html#getLayoutInflater()))

onCreateView

Added in API level 11 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

This site uses cookies to store your preferences for site-specific language options.

OK

Context (<https://developer.android.com/reference/android/content/Context.html>) (
AttributeSet (<https://developer.android.com/reference/android/util/AttributeSet>

Standard implementation of `onCreateView(String, Context, AttributeSet)`
([https://developer.android.com/reference/android/view/LayoutInflater.Factory.html#onCreateView\(java.lang.String, android.content.Context, android.util.AttributeSet\)](https://developer.android.com/reference/android/view/LayoutInflater.Factory.html#onCreateView(java.lang.String, android.content.Context, android.util.AttributeSet)))) used when inflating with the `LayoutInflater` returned by `getSystemService(Class)` ([https://developer.android.com/reference/android/content/Context.html#getSystemService\(java.lang.Class<T>\)](https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>)))). This implementation does nothing and is for `pre-HONEYCOMB` (https://developer.android.com/reference/android/os/Build.VERSION_CODES.html#HONEYCOMB) apps. Newer apps should use `onCreateView(View, String, Context, AttributeSet)` ([https://developer.android.com/reference/android/app/Activity.html#onCreateView\(android.view.View, java.lang.String, android.content.Context, android.util.AttributeSet\)](https://developer.android.com/reference/android/app/Activity.html#onCreateView(android.view.View, java.lang.String, android.content.Context, android.util.AttributeSet)))).

Parameters	
name	String: Tag name to be inflated.
context	Context: The context the view is being created in.
attrs	AttributeSet: Inflation attributes as specified in XML file.

Returns	
View (https://developer.android.com/reference/android/view/View.html)	View Newly created view. Return null for the default behavior.

See also:

`createView(String, String, AttributeSet)` ([https://developer.android.com/reference/android/view/LayoutInflater.html#createView\(java.lang.String, java.lang.String, android.util.AttributeSet\)](https://developer.android.com/reference/android/view/LayoutInflater.html#createView(java.lang.String, java.lang.String, android.util.AttributeSet))))
`getLayoutInflater()` ([https://developer.android.com/reference/android/view/Window.html#getLayoutInflater\(\)](https://developer.android.com/reference/android/view/Window.html#getLayoutInflater()))

onDetachedFromWindow Added in API level 8 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

This site uses cookies to store your preferences for site-specific language options.

OK

Called when the main window associated with the activity has been detached from the window manager. See `View.onDetachedFromWindow()` ([https://developer.android.com/reference/android/view/View.html#onDetachedFromWindow\(\)](https://developer.android.com/reference/android/view/View.html#onDetachedFromWindow())) for more information.

See also:

`onDetachedFromWindow()` ([https://developer.android.com/reference/android/view/View.html#onDetachedFromWindow\(\)](https://developer.android.com/reference/android/view/View.html#onDetachedFromWindow()))

onEnterAnimationComplete

Added in API level 12 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
void onEnterAnimationComplete ()
```

Activities cannot draw during the period that their windows are animating in. In order to know when it is safe to begin drawing they can override this method which will be called when the entering animation has completed.

onGenericMotionEvent

Added in API level 12 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
boolean onGenericMotionEvent (MotionEvent (https://developer.android.com/reference/android/view/MotionEvent))
```

Called when a generic motion event was not handled by any of the views inside of the activity.

Generic motion events describe joystick movements, mouse hovers, track pad touches, scroll wheel movements and other input events. The `source` ([https://developer.android.com/reference/android/view/MotionEvent.html#getSource\(\)](https://developer.android.com/reference/android/view/MotionEvent.html#getSource())) of the motion event specifies the class of input that was received. Implementations of this method must examine the bits in the source before processing the event. The following code example shows how this is done.

Generic motion events with source class `SOURCE_CLASS_POINTER` (https://developer.android.com/reference/android/view/InputDevice.html#SOURCE_CLASS_POINTER) are delivered to the view under the pointer. All other generic motion events are delivered to the focused view.

See `onGenericMotionEvent(MotionEvent)` ([https://developer.android.com/reference/android/view/View.html#onGenericMotionEvent\(android.view.MotionEvent\)](https://developer.android.com/reference/android/view/View.html#onGenericMotionEvent(android.view.MotionEvent))) for an example of how to handle this event.

This site uses cookies to store your preferences for site-specific language options.

OK

event	MotionEvent : The generic motion event being processed.
-------	--

Returns

boolean	Return true if you have consumed the event, false if you haven't. The default implementation always returns false.
---------	--

onKeyDown

Added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
boolean onKeyDown (int keyCode,
                    KeyEvent (https://developer.android.com/reference/android/view/KeyEvent.html) e)
```

Called when a key was pressed down and not handled by any of the views inside of the activity. So, for example, key presses while the cursor is inside a `TextView` will not trigger the event (unless it is a navigation to another object) because `TextView` handles its own key presses.

If the focused view didn't want this event, this method is called.

The default implementation takes care of `KEYCODE_BACK` (https://developer.android.com/reference/android/view/KeyEvent.html#KEYCODE_BACK) by calling `onBackPressed()` ([https://developer.android.com/reference/android/app/Activity.html#onBackPressed\(\)](https://developer.android.com/reference/android/app/Activity.html#onBackPressed())), though the behavior varies based on the application compatibility mode: for `ECLAIR` (https://developer.android.com/reference/android/os/Build.VERSION_CODES.html#ECLAIR) or later applications, it will set up the dispatch to call `onKeyUp(int, KeyEvent)` ([https://developer.android.com/reference/android/app/Activity.html#onKeyUp\(int, android.view.KeyEvent\)](https://developer.android.com/reference/android/app/Activity.html#onKeyUp(int, android.view.KeyEvent))) where the action will be performed; for earlier applications, it will perform the action immediately in on-down, as those versions of the platform behaved.

Other additional default key handling may be performed if configured with `setDefaultKeyMode(int)` ([https://developer.android.com/reference/android/app/Activity.html#setDefaultKeyMode\(int\)](https://developer.android.com/reference/android/app/Activity.html#setDefaultKeyMode(int))).

Parameters

keyCode	int : The value in <code>event.getKeyCode()</code> .
event	KeyEvent : Description of the key event.

Returns

This site uses cookies to store your preferences for site-specific language options.

OK

boolean	Return true to prevent this event from being propagated further, or false to indicate that you have not handled this event and it should continue to be propagated.
----------------	---

See also:

onKeyUp(int, KeyEvent) ([https://developer.android.com/reference/android/app/Activity.html#onKeyUp\(int, android.view.KeyEvent\)](https://developer.android.com/reference/android/app/Activity.html#onKeyUp(int, android.view.KeyEvent)))

KeyEvent (<https://developer.android.com/reference/android/view/KeyEvent.html>)

onKeyLongPress

Added in API level 5 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
boolean onKeyLongPress (int keyCode,
                        KeyEvent (https://developer.android.com/reference/android/view/KeyEvent.html) e
```

Default implementation of **KeyEvent.Callback.onKeyLongPress()** ([https://developer.android.com/reference/android/view/KeyEvent.Callback.html#onKeyLongPress\(int, android.view.KeyEvent\)](https://developer.android.com/reference/android/view/KeyEvent.Callback.html#onKeyLongPress(int, android.view.KeyEvent))): **always** returns false (doesn't handle the event).

Parameters	
keyCode	int : The value in event.getKeyCode().
event	KeyEvent : Description of the key event.
Returns	
boolean	If you handled the event, return true. If you want to allow the event to be handled by the next receiver, return false.

onKeyMultiple

Added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
boolean onKeyMultiple (int keyCode,
                      int repeatCount,
                      KeyEvent (https://developer.android.com/reference/android/view/KeyEvent.html) e
```

Default implementation of **KeyEvent.Callback.onKeyMultiple()** ([https://developer.android.com/reference/android/view/KeyEvent.Callback.html#onKeyMultiple\(int, int, android.view.KeyEvent\)](https://developer.android.com/reference/android/view/KeyEvent.Callback.html#onKeyMultiple(int, int, android.view.KeyEvent))):

This site uses cookies to store your preferences for site-specific language options.

OK

Parameters	
<code>keyCode</code>	<code>int</code> : The value in <code>event.getKeyCode()</code> .
<code>repeatCount</code>	<code>int</code> : Number of pairs as returned by <code>event.getRepeatCount()</code> .
<code>event</code>	<code>KeyEvent</code> : Description of the key event.

Returns	
<code>boolean</code>	If you handled the event, return <code>true</code> . If you want to allow the event to be handled by the next receiver, return <code>false</code> .

onKeyShortcut Added in API level 11 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
boolean onKeyShortcut (int keyCode,
                      KeyEvent (https://developer.android.com/reference/android/view/KeyEvent.html) e
```

Called when a key shortcut event is not handled by any of the views in the Activity. Override this method to implement global key shortcuts for the Activity. Key shortcuts can also be implemented by setting the `shortcut` ([https://developer.android.com/reference/android/view/MenuItem.html#setShortcut\(char, char\)](https://developer.android.com/reference/android/view/MenuItem.html#setShortcut(char, char))) property of menu items.

Parameters	
<code>keyCode</code>	<code>int</code> : The value in <code>event.getKeyCode()</code> .
<code>event</code>	<code>KeyEvent</code> : Description of the key event.

Returns	
<code>boolean</code>	True if the key shortcut was handled.

onKeyUp Added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
boolean onKeyUp (int keyCode,
                KeyEvent (https://developer.android.com/reference/android/view/KeyEvent.html) e
```

~~Called when a key was released and not handled by any of the views inside of the activity. See for~~

This site uses cookies to store your preferences for site-specific language options.

The default implementation handles `KEYCODE_BACK` to stop the activity and go back.

Parameters	
<code>keyCode</code>	<code>int</code> : The value in <code>event.getKeyCode()</code> .
<code>event</code>	<code>KeyEvent</code> : Description of the key event.
Returns	
<code>boolean</code>	Return <code>true</code> to prevent this event from being propagated further, or <code>false</code> to indicate that you have not handled this event and it should continue to be propagated.

See also:

`onKeyDown(int, KeyEvent)` ([https://developer.android.com/reference/android/app/Activity.html#onKeyDown\(int, android.view.KeyEvent\)](https://developer.android.com/reference/android/app/Activity.html#onKeyDown(int, android.view.KeyEvent)))

`KeyEvent` (<https://developer.android.com/reference/android/view/KeyEvent.html>)

onLocalVoiceInteractionStarted

Added in API level 24 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
void onLocalVoiceInteractionStarted ()
```

Callback to indicate that `startLocalVoiceInteraction(Bundle)` ([https://developer.android.com/reference/android/app/Activity.html#startLocalVoiceInteraction\(android.os.Bundle\)](https://developer.android.com/reference/android/app/Activity.html#startLocalVoiceInteraction(android.os.Bundle))) has resulted in a voice interaction session being started. You can now retrieve a voice interactor using `getVoiceInteractor()` ([https://developer.android.com/reference/android/app/Activity.html#getVoiceInteractor\(\)](https://developer.android.com/reference/android/app/Activity.html#getVoiceInteractor())).

onLocalVoiceInteractionStopped

Added in API level 24 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
void onLocalVoiceInteractionStopped ()
```

Callback to indicate that the local voice interaction has stopped either because it was requested through a call to `stopLocalVoiceInteraction()` (<https://developer.android.com/reference/android...>)

This site uses cookies to store your preferences for site-specific language options.

OK

`/app/VoiceInteractor.html`) is no longer valid after this.

onLowMemory

Added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
void onLowMemory ()
```

This is called when the overall system is running low on memory, and actively running processes should trim their memory usage. While the exact point at which this will be called is not defined, generally it will happen when all background process have been killed. That is, before reaching the point of killing processes hosting service and foreground UI that we would like to avoid killing.

You should implement this method to release any caches or other unnecessary resources you may be holding on to. The system will perform a garbage collection for you after returning from this method.

Preferably, you should implement `onTrimMemory(int)` ([https://developer.android.com/reference/android/content/ComponentCallbacks2.html#onTrimMemory\(int\)](https://developer.android.com/reference/android/content/ComponentCallbacks2.html#onTrimMemory(int))) from `ComponentCallbacks2` (<https://developer.android.com/reference/android/content/ComponentCallbacks2.html>) to incrementally unload your resources based on various levels of memory demands. That API is available for API level 14 and higher, so you should only use this `onLowMemory()` ([https://developer.android.com/reference/android/content/ComponentCallbacks.html#onLowMemory\(\)](https://developer.android.com/reference/android/content/ComponentCallbacks.html#onLowMemory())) method as a fallback for older versions, which can be treated the same as `onTrimMemory(int)` ([https://developer.android.com/reference/android/content/ComponentCallbacks2.html#onTrimMemory\(int\)](https://developer.android.com/reference/android/content/ComponentCallbacks2.html#onTrimMemory(int))) with the `TRIM_MEMORY_COMPLETE` (https://developer.android.com/reference/android/content/ComponentCallbacks2.html#TRIM_MEMORY_COMPLETE) level.

onMenuItemSelected

Added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
boolean onMenuItemSelected (int featureId,
                             MenuItem (https://developer.android.com/reference/android/view/MenuItem.html) i
```

Default implementation of `onMenuItemSelected(int, MenuItem)` ([https://developer.android.com/reference/android/view/Window.Callback.html#onMenuItemSelected\(int, android.view.MenuItem\)](https://developer.android.com/reference/android/view/Window.Callback.html#onMenuItemSelected(int, android.view.MenuItem))) for activities. This calls through to the new `onOptionsItemSelected(MenuItem)` (<https://developer.android.com/reference/android>

This site uses cookies to store your preferences for site-specific language options.

OK

`/view/Window.html#FEATURE_OPTIONS_PANEL`) panel, so that subclasses of Activity don't need to deal with feature codes.

Parameters	
<code>featureId</code>	<code>int</code> : The panel that the menu is in.
<code>item</code>	<code>MenuItem</code> : The menu item that was selected.

Returns	
<code>boolean</code>	boolean Return true to finish processing of selection, or false to perform the normal menu handling (calling its Runnable or sending a Message to its target Handler).

onMenuOpened

Added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
boolean onMenuOpened (int featureId,  
                      Menu (https://developer.android.com/reference/android/view/Menu.html) menu)
```

Called when a panel's menu is opened by the user. This may also be called when the menu is changing from one type to another (for example, from the icon menu to the expanded menu).

Parameters	
<code>featureId</code>	<code>int</code> : The panel that the menu is in.
<code>menu</code>	<code>Menu</code> : The menu that is opened.

Returns	
<code>boolean</code>	The default implementation returns true.

onMultiWindowModeChanged

Added in API level 24 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
void onMultiWindowModeChanged (boolean isInMultiWindowMode)
```

Called by the system when the activity changes from fullscreen mode to multi-window mode and visa-versa.

This site uses cookies to store your preferences for site-specific language options.

OK

<code>isInMultiWindowMode</code>	<code>boolean</code> : True if the activity is in multi-window mode.
----------------------------------	--

See also:

`resizeableActivity` (<https://developer.android.com/reference/android/R.attr.html#resizeableActivity>)

onNavigateUp

Added in API level 16 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

`boolean onNavigateUp ()`

This method is called whenever the user chooses to navigate Up within your application's activity hierarchy from the action bar.

If the attribute `parentActivityName` (<https://developer.android.com/reference/android/R.attr.html#parentActivityName>) was specified in the manifest for this activity or an activity-alias to it, default Up navigation will be handled automatically. If any activity along the parent chain requires extra Intent arguments, the Activity subclass should override the method `onPrepareNavigateUpTaskStack(TaskStackBuilder)` ([https://developer.android.com/reference/android/app/Activity.html#onPrepareNavigateUpTaskStack\(android.app.TaskStackBuilder\)](https://developer.android.com/reference/android/app/Activity.html#onPrepareNavigateUpTaskStack(android.app.TaskStackBuilder))) to supply those arguments.

See [Tasks and Back Stack](https://developer.android.com/guide/components/tasks-and-back-stack.html) (<https://developer.android.com/guide/components/tasks-and-back-stack.html>) from the developer guide and [Navigation](https://developer.android.com/design/patterns/navigation.html) (<https://developer.android.com/design/patterns/navigation.html>) from the design guide for more information about navigating within your app.

See the `TaskStackBuilder` (<https://developer.android.com/reference/android/app/TaskStackBuilder.html>) class and the Activity methods `getParentActivityIntent()` ([https://developer.android.com/reference/android/app/Activity.html#getParentActivityIntent\(\)](https://developer.android.com/reference/android/app/Activity.html#getParentActivityIntent())), `shouldUpRecreateTask(Intent)` ([https://developer.android.com/reference/android/app/Activity.html#shouldUpRecreateTask\(android.content.Intent\)](https://developer.android.com/reference/android/app/Activity.html#shouldUpRecreateTask(android.content.Intent))), and `navigateUpTo(Intent)` ([https://developer.android.com/reference/android/app/Activity.html#navigateUpTo\(android.content.Intent\)](https://developer.android.com/reference/android/app/Activity.html#navigateUpTo(android.content.Intent))) for help implementing custom Up navigation. The AppNavigation sample application in the Android SDK is also available for reference.

Returns

This site uses cookies to store your preferences for site-specific language options.

OK

onNavigateUpFromChild

Added in API level 16 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

`boolean onNavigateUpFromChild (Activity` (<https://developer.android.com/reference/android/app>

This is called when a child activity of this one attempts to navigate up. The default implementation simply calls `onNavigateUp()` on this activity (the parent).

Parameters	
<code>child</code>	Activity: The activity making the call.
Returns	
<code>boolean</code>	

onOptionsItemSelected

Added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

`boolean onOptionsItemSelected (MenuItem` (<https://developer.android.com/reference/android/view>

This hook is called whenever an item in your options menu is selected. The default implementation simply returns false to have the normal processing happen (calling the item's Runnable or sending a message to its Handler as appropriate). You can use this method for any items for which you would like to do processing without those other facilities.

Derived classes should call through to the base class for it to perform the default menu handling.

Parameters	
<code>item</code>	MenuItem: The menu item that was selected.
Returns	
<code>boolean</code>	boolean Return false to allow normal menu processing to proceed, true to consume it here.

See also:

`onCreateOptionsMenu(Menu)` ([https://developer.android.com/reference/android/app/Activity.html#onCreateOptionsMenu\(android.view.Menu\)](https://developer.android.com/reference/android/app/Activity.html#onCreateOptionsMenu(android.view.Menu)))

This site uses cookies to store your preferences for site-specific language options.

OK

onOptionsItemSelected

Added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
void onOptionsItemSelected (Menu (https://developer.android.com/reference/android/view/Menu.html) menu)
```

This hook is called whenever the options menu is being closed (either by the user canceling the menu with the back/menu button, or when an item is selected).

Parameters

menu	Menu: The options menu as last shown or first initialized by <code>onCreateOptionsMenu()</code> .
-------------	--

onPanelClosed

Added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
void onPanelClosed (int featureId,  
                    Menu (https://developer.android.com/reference/android/view/Menu.html) menu)
```

Default implementation of `onPanelClosed(int, Menu)` ([https://developer.android.com/reference/android/view/Window.Callback.html#onPanelClosed\(int, android.view.Menu\)](https://developer.android.com/reference/android/view/Window.Callback.html#onPanelClosed(int, android.view.Menu))) for activities. This calls through to `onOptionsItemSelected(Menu)` ([https://developer.android.com/reference/android/app/Activity.html#onOptionsItemSelected\(android.view.Menu\)](https://developer.android.com/reference/android/app/Activity.html#onOptionsItemSelected(android.view.Menu))) method for the `FEATURE_OPTIONS_PANEL` (https://developer.android.com/reference/android/view/Window.html#FEATURE_OPTIONS_PANEL) panel, so that subclasses of Activity don't need to deal with feature codes. For context menus (`FEATURE_CONTEXT_MENU` (https://developer.android.com/reference/android/view/Window.html#FEATURE_CONTEXT_MENU)), the `onContextMenuClosed(Menu)` ([https://developer.android.com/reference/android/app/Activity.html#onContextMenuClosed\(android.view.Menu\)](https://developer.android.com/reference/android/app/Activity.html#onContextMenuClosed(android.view.Menu))) will be called.

Parameters

featureId	int: The panel that is being displayed.
menu	Menu: If <code>onCreatePanelView()</code> returned null, this is the Menu being displayed in the panel.

onPictureInPictureModeChanged

Added in API level 24 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
void onPictureInPictureModeChanged (boolean isInPictureInPictureMode)
```

This site uses cookies to store your preferences for site-specific language options.

OK

Parameters	
<code>isInPictureInPictureMode</code>	<code>boolean</code> : True if the activity is in picture-in-picture mode.

See also:

`supportsPictureInPicture` (<https://developer.android.com/reference/android/R.attr.html#supportsPictureInPicture>)

onPostCreate Added in API level 21 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

`void onPostCreate (Bundle (https://developer.android.com/reference/android/os/Bundle.html) savedInstanceState (https://developer.android.com/reference/android/os/PersistableBundle`

This is the same as `onPostCreate(Bundle)` ([https://developer.android.com/reference/android/app/Activity.html#onPostCreate\(android.os.Bundle\)](https://developer.android.com/reference/android/app/Activity.html#onPostCreate(android.os.Bundle))) but is called for activities created with the attribute `persistableMode` (<https://developer.android.com/reference/android/R.attr.html#persistableMode>) set to `persistAcrossReboots`.

Parameters	
<code>savedInstanceState</code>	<code>Bundle</code> : The data most recently supplied in <code>onSaveInstanceState(Bundle)</code> (https://developer.android.com/reference/android/app/Activity.html#onSaveInstanceState(android.os.Bundle))
<code>persistentState</code>	<code>PersistableBundle</code> : The data coming from the <code>PersistableBundle</code> first saved in <code>onSaveInstanceState(Bundle, PersistableBundle)</code> (https://developer.android.com/reference/android/app/Activity.html#onSaveInstanceState(android.os.Bundle, android.os.PersistableBundle)).

See also:

`onCreate(Bundle)` ([https://developer.android.com/reference/android/app/Activity.html#onCreate\(android.os.Bundle\)](https://developer.android.com/reference/android/app/Activity.html#onCreate(android.os.Bundle)))

```
void onPrepareNavigateUpTaskStack (TaskStackBuilder (https://developer.android.com/refere
```

Prepare the synthetic task stack that will be generated during Up navigation from a different task.

This method receives the `TaskStackBuilder` (<https://developer.android.com/reference/android/app/TaskStackBuilder.html>) with the constructed series of Intents as generated by `onCreateNavigateUpTaskStack(TaskStackBuilder)` ([https://developer.android.com/reference/android/app/Activity.html#onCreateNavigateUpTaskStack\(android.app.TaskStackBuilder\)](https://developer.android.com/reference/android/app/Activity.html#onCreateNavigateUpTaskStack(android.app.TaskStackBuilder))). If any extra data should be added to these intents before launching the new task, the application should override this method and add that data here.

Parameters	
<code>builder</code>	TaskStackBuilder: A TaskStackBuilder that has been populated with Intents by <code>onCreateNavigateUpTaskStack</code> .

onPrepareOptionsMenu

Added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
boolean onPrepareOptionsMenu (Menu (https://developer.android.com/reference/android/view/Menu
```

Prepare the Screen's standard options menu to be displayed. This is called right before the menu is shown, every time it is shown. You can use this method to efficiently enable/disable items or otherwise dynamically modify the contents.

The default implementation updates the system menu items based on the activity's state. Deriving classes should always call through to the base class implementation.

Parameters	
<code>menu</code>	Menu: The options menu as last shown or first initialized by <code>onCreateOptionsMenu()</code> .

Returns	
<code>boolean</code>	You must return true for the menu to be displayed; if you return false it will not be shown.

See also:

`onCreateOptionsMenu(Menu)` ([https://developer.android.com/reference/android/app/Activity.html#onCreateOptionsMenu\(android.view.Menu\)](https://developer.android.com/reference/android/app/Activity.html#onCreateOptionsMenu(android.view.Menu)))

This site uses cookies to store your preferences for site-specific language options.

OK

onPreparePanel Added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
boolean onPreparePanel (int featureId,
                        View (https://developer.android.com/reference/android/view/View.html) view,
                        Menu (https://developer.android.com/reference/android/view/Menu.html) menu)
```

Default implementation of `onPreparePanel(int, View, Menu)` ([https://developer.android.com/reference/android/view/Window.Callback.html#onPreparePanel\(int, android.view.View, android.view.Menu\)](https://developer.android.com/reference/android/view/Window.Callback.html#onPreparePanel(int, android.view.View, android.view.Menu))) for activities. This calls through to the new `onPrepareOptionsMenu(Menu)` ([https://developer.android.com/reference/android/app/Activity.html#onPrepareOptionsMenu\(android.view.Menu\)](https://developer.android.com/reference/android/app/Activity.html#onPrepareOptionsMenu(android.view.Menu))) method for the `FEATURE_OPTIONS_PANEL` (https://developer.android.com/reference/android/view/Window.html#FEATURE_OPTIONS_PANEL) panel, so that subclasses of Activity don't need to deal with feature codes.

Parameters	
<code>featureId</code>	<code>int</code> : The panel that is being displayed.
<code>view</code>	<code>View</code> : The View that was returned by <code>onCreatePanelView()</code> .
<code>menu</code>	<code>Menu</code> : If <code>onCreatePanelView()</code> returned null, this is the Menu being displayed in the panel.

Returns	
<code>boolean</code>	<code>boolean</code> You must return true for the panel to be displayed; if you return false it will not be shown.

onProvideAssistContent Added in API level 23 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
void onProvideAssistContent (AssistContent (https://developer.android.com/reference/android...))
```

This is called when the user is requesting an assist, to provide references to content related to the current activity. Before being called, the `outContent` Intent is filled with the base Intent of the activity (the Intent returned by `getIntent()` ([https://developer.android.com/reference/android/app/Activity.html#getIntent\(\)](https://developer.android.com/reference/android/app/Activity.html#getIntent()))). The Intent's extras are stripped of any types that are not valid for `PersistableBundle` (<https://developer.android.com/reference/android/os/PersistableBundle.html>) or non-framework Parcelables, and the flags `FLAG_GRANT_WRITE_URI_PERMISSION` (https://developer.android.com/reference/android/content/Intent.html#FLAG_GRANT_WRITE_URI_PERMISSION)

This site uses cookies to store your preferences for site-specific language options.

OK

Custom implementation may adjust the content intent to better reflect the top-level context of the activity, and fill in its ClipData with additional content of interest that the user is currently viewing. For example, an image gallery application that has launched in to an activity allowing the user to swipe through pictures should modify the intent to reference the current image they are looking it; such an application when showing a list of pictures should add a ClipData that has references to all of the pictures currently visible on screen.

Parameters	
<code>outContent</code>	<code>AssistContent</code> : The assist content to return.

onProvideAssistData

Added in API level 18 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
void onProvideAssistData (Bundle (https://developer.android.com/reference/android/os/Bundle.ht
```

This is called when the user is requesting an assist, to build a full `ACTION_ASSIST` (https://developer.android.com/reference/android/content/Intent.html#ACTION_ASSIST) Intent with all of the context of the current application. You can override this method to place into the bundle anything you would like to appear in the `EXTRA_ASSIST_CONTEXT` (https://developer.android.com/reference/android/content/Intent.html#EXTRA_ASSIST_CONTEXT) part of the assist Intent.

This function will be called after any global assist callbacks that had been registered with `Application.registerOnProvideAssistDataListener` ([https://developer.android.com/reference/android/app/Application.html#registerOnProvideAssistDataListener\(android.app.Application.OnProvideAssistDataListener\)](https://developer.android.com/reference/android/app/Application.html#registerOnProvideAssistDataListener(android.app.Application.OnProvideAssistDataListener))).

Parameters	
<code>data</code>	<code>Bundle</code>

onProvideKeyboardShortcuts

```
void onProvideKeyboardShortcuts (List (https://developer.android.com/reference/java/util/List.html) menu,
                                Menu (https://developer.android.com/reference/android/view/Menu.html) menu,
                                int deviceId)
```

This site uses cookies to store your preferences for site-specific language options.

OK

Parameters	
<code>data</code>	List : The data list to populate with shortcuts.
<code>menu</code>	Menu : The current menu, which may be null.
<code>deviceId</code>	int : The id for the connected device the shortcuts should be provided for.

onProvideReferrer

Added in API level 23 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

`Uri` (<https://developer.android.com/reference/android/net/Uri.html>) `onProvideReferrer` ()

Override to generate the desired referrer for the content currently being shown by the app. The default implementation returns null, meaning the referrer will simply be the android-app: of the package name of this activity. Return a non-null Uri to have that supplied as the **EXTRA_REFERRER** (https://developer.android.com/reference/android/content/Intent.html#EXTRA_REFERRER) of any activities started from it.

Returns	
<code>Uri</code> (https://developer.android.com/reference/android/net/Uri.html)	

onRequestPermissionsResult

Added in API level 23 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
void onRequestPermissionsResult (int requestCode,
                                String[] (https://developer.android.com/reference/java/lang/String.html) permissions,
                                int[] grantResults)
```

Callback for the result from requesting permissions. This method is invoked for every call on `requestPermissions(String[], int)` ([https://developer.android.com/reference/android/app/Activity.html#requestPermissions\(java.lang.String\[\], int\)](https://developer.android.com/reference/android/app/Activity.html#requestPermissions(java.lang.String[], int))).

Note: It is possible that the permissions request interaction with the user is interrupted. In this case you will receive empty permissions and results arrays which should be treated as a cancellation.

This site uses cookies to store your preferences for site-specific language options.

OK

<code>requestCode</code>	<code>int</code> : The request code passed in <code>requestPermissions(String[], int)</code> (https://developer.android.com/reference/android/app/Activity.html#requestPermissions(java.lang.String[], int)).
<code>permissions</code>	<code>String</code> : The requested permissions. Never null.
<code>grantResults</code>	<code>int</code> : The grant results for the corresponding permissions which is either <code>PERMISSION_GRANTED</code> (https://developer.android.com/reference/android/content/pm/PackageManager.html#PERMISSION_GRANTED) or <code>PERMISSION_DENIED</code> (https://developer.android.com/reference/android/content/pm/PackageManager.html#PERMISSION_DENIED). Never null.

See also:

`requestPermissions(String[], int)` ([https://developer.android.com/reference/android/app/Activity.html#requestPermissions\(java.lang.String\[\], int\)](https://developer.android.com/reference/android/app/Activity.html#requestPermissions(java.lang.String[], int)))

onRestoreInstanceState

Added in API level 21 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
void onRestoreInstanceState (Bundle (https://developer.android.com/reference/android/os/Bundle)
    PersistableBundle (https://developer.android.com/reference/android/os/PersistableBundle))
```

This is the same as `onRestoreInstanceState(Bundle)` ([https://developer.android.com/reference/android/app/Activity.html#onRestoreInstanceState\(android.os.Bundle\)](https://developer.android.com/reference/android/app/Activity.html#onRestoreInstanceState(android.os.Bundle))) but is called for activities created with the attribute `persistableMode` (<https://developer.android.com/reference/android/R.attr.html#persistableMode>) set to `persistAcrossReboots`. The `PersistableBundle` (<https://developer.android.com/reference/android/os/PersistableBundle.html>) passed came from the restored `PersistableBundle` first saved in `onSaveInstanceState(Bundle, PersistableBundle)` ([https://developer.android.com/reference/android/app/Activity.html#onSaveInstanceState\(android.os.Bundle, android.os.PersistableBundle\)](https://developer.android.com/reference/android/app/Activity.html#onSaveInstanceState(android.os.Bundle, android.os.PersistableBundle))).

This method is called between `onStart()` ([https://developer.android.com/reference/android/app/Activity.html#onStart\(\)](https://developer.android.com/reference/android/app/Activity.html#onStart())) and `onPostCreate(Bundle)` ([https://developer.android.com/reference/android/app/Activity.html#onPostCreate\(android.os.Bundle\)](https://developer.android.com/reference/android/app/Activity.html#onPostCreate(android.os.Bundle))).

If this method is called `onRestoreInstanceState(Bundle)` ([https://developer.android.com/reference/android/app/Activity.html#onRestoreInstanceState\(android.os.Bundle\)](https://developer.android.com/reference/android/app/Activity.html#onRestoreInstanceState(android.os.Bundle))) will not be called.

Parameters

This site uses cookies to store your preferences for site-specific language options.

OK

<code>savedInstanceState</code>	Bundle : the data most recently supplied in <code>onSaveInstanceState(Bundle)</code> (https://developer.android.com/reference/android/app/Activity.html#onSaveInstanceState(android.os.Bundle)).
<code>persistentState</code>	PersistableBundle : the data most recently supplied in <code>onSaveInstanceState(Bundle)</code> (https://developer.android.com/reference/android/app/Activity.html#onSaveInstanceState(android.os.Bundle)).

See also:

`onRestoreInstanceState(Bundle)` ([https://developer.android.com/reference/android/app/Activity.html#onRestoreInstanceState\(android.os.Bundle\)](https://developer.android.com/reference/android/app/Activity.html#onRestoreInstanceState(android.os.Bundle)))

`onCreate(Bundle)` ([https://developer.android.com/reference/android/app/Activity.html#onCreate\(android.os.Bundle\)](https://developer.android.com/reference/android/app/Activity.html#onCreate(android.os.Bundle)))

`onPostCreate(Bundle)` ([https://developer.android.com/reference/android/app/Activity.html#onPostCreate\(android.os.Bundle\)](https://developer.android.com/reference/android/app/Activity.html#onPostCreate(android.os.Bundle)))

`onResume()` ([https://developer.android.com/reference/android/app/Activity.html#onResume\(\)](https://developer.android.com/reference/android/app/Activity.html#onResume()))

`onSaveInstanceState(Bundle)` ([https://developer.android.com/reference/android/app/Activity.html#onSaveInstanceState\(android.os.Bundle\)](https://developer.android.com/reference/android/app/Activity.html#onSaveInstanceState(android.os.Bundle)))

onRetainNonConfigurationInstance

Added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

`Object` (<https://developer.android.com/reference/java/lang/Object.html>) `onRetainNonConfigurationInstance()`

Called by the system, as part of destroying an activity due to a configuration change, when it is known that a new instance will immediately be created for the new configuration. You can return any object you like here, including the activity instance itself, which can later be retrieved by calling `getLastNonConfigurationInstance()` ([https://developer.android.com/reference/android/app/Activity.html#getLastNonConfigurationInstance\(\)](https://developer.android.com/reference/android/app/Activity.html#getLastNonConfigurationInstance())) in the new activity instance. *If you are targeting HONEYCOMB* (https://developer.android.com/reference/android/os/Build.VERSION_CODES.html#HONEYCOMB) *or later, consider instead using a* `Fragment` (<https://developer.android.com/reference/android/app/Fragment.html>) *with* `Fragment.setRetainInstance(boolean)` ([https://developer.android.com/reference/android/app/Fragment.html#setRetainInstance\(boolean\)](https://developer.android.com/reference/android/app/Fragment.html#setRetainInstance(boolean))).

This site uses cookies to store your preferences for site-specific language options.

OK

called, a number of guarantees will be made to help optimize configuration switching:

- The function will be called between `onStop()` ([https://developer.android.com/reference/android/app/Activity.html#onStop\(\)](https://developer.android.com/reference/android/app/Activity.html#onStop())) and `onDestroy()` ([https://developer.android.com/reference/android/app/Activity.html#onDestroy\(\)](https://developer.android.com/reference/android/app/Activity.html#onDestroy())).
- A new instance of the activity will *always* be immediately created after this one's `onDestroy()` ([https://developer.android.com/reference/android/app/Activity.html#onDestroy\(\)](https://developer.android.com/reference/android/app/Activity.html#onDestroy())) is called. In particular, *no* messages will be dispatched during this time (when the returned object does not have an activity to be associated with).
- The object you return here will *always* be available from the `getLastNonConfigurationInstance()` ([https://developer.android.com/reference/android/app/Activity.html#getLastNonConfigurationInstance\(\)](https://developer.android.com/reference/android/app/Activity.html#getLastNonConfigurationInstance())) method of the following activity instance as described there.

These guarantees are designed so that an activity can use this API to propagate extensive state from the old to new activity instance, from loaded bitmaps, to network connections, to evenly actively running threads. Note that you should *not* propagate any data that may change based on the configuration, including any data loaded from resources such as strings, layouts, or drawables.

The guarantee of no message handling during the switch to the next activity simplifies use with active objects. For example if your retained state is an `AsyncTask` (<https://developer.android.com/reference/android/os/AsyncTask.html>) you are guaranteed that its call back functions (like `onPostExecute(Result)` ([https://developer.android.com/reference/android/os/AsyncTask.html#onPostExecute\(Result\)](https://developer.android.com/reference/android/os/AsyncTask.html#onPostExecute(Result)))) will not be called from the call here until you execute the next instance's `onCreate(Bundle)` ([https://developer.android.com/reference/android/app/Activity.html#onCreate\(android.os.Bundle\)](https://developer.android.com/reference/android/app/Activity.html#onCreate(android.os.Bundle))). (Note however that there is of course no such guarantee for `doInBackground(Params...)` ([https://developer.android.com/reference/android/os/AsyncTask.html#doInBackground\(Params...\)](https://developer.android.com/reference/android/os/AsyncTask.html#doInBackground(Params...))) since that is running in a separate thread.)

Note: For most cases you should use the `Fragment` (<https://developer.android.com/reference/android/app/Fragment.html>) API `setRetainInstance(boolean)` ([https://developer.android.com/reference/android/app/Fragment.html#setRetainInstance\(boolean\)](https://developer.android.com/reference/android/app/Fragment.html#setRetainInstance(boolean))) instead; this is also available on older platforms through the Android support libraries.

Returns	
<code>Object</code> (https://developer.android.com/reference/java/lang	any Object holding the desired state to propagate to the next activity instance

This site uses cookies to store your preferences for site-specific language options.

OK

onSaveInstanceState

Added in API level 21 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
void onSaveInstanceState (Bundle (https://developer.android.com/reference/android/os/Bundle.html)
    PersistableBundle (https://developer.android.com/reference/android/os/PersistableBundle.html))
```

This is the same as `onSaveInstanceState(Bundle)` ([https://developer.android.com/reference/android/app/Activity.html#onSaveInstanceState\(android.os.Bundle\)](https://developer.android.com/reference/android/app/Activity.html#onSaveInstanceState(android.os.Bundle))) but is called for activities created with the attribute `persistableMode` (<https://developer.android.com/reference/android/R.attr.html#persistableMode>) set to `persistAcrossReboots`. The `PersistableBundle` (<https://developer.android.com/reference/android/os/PersistableBundle.html>) passed in will be saved and presented in `onCreate(Bundle, PersistableBundle)` ([https://developer.android.com/reference/android/app/Activity.html#onCreate\(android.os.Bundle, android.os.PersistableBundle\)](https://developer.android.com/reference/android/app/Activity.html#onCreate(android.os.Bundle, android.os.PersistableBundle))) the first time that this activity is restarted following the next device reboot.

Parameters	
<code>outState</code>	<code>Bundle</code> : Bundle in which to place your saved state.
<code>outPersistentState</code>	<code>PersistableBundle</code> : State which will be saved across reboots.

See also:

`onSaveInstanceState(Bundle)` ([https://developer.android.com/reference/android/app/Activity.html#onSaveInstanceState\(android.os.Bundle\)](https://developer.android.com/reference/android/app/Activity.html#onSaveInstanceState(android.os.Bundle)))

`onCreate(Bundle)` ([https://developer.android.com/reference/android/app/Activity.html#onCreate\(android.os.Bundle\)](https://developer.android.com/reference/android/app/Activity.html#onCreate(android.os.Bundle)))

`onRestoreInstanceState(Bundle, PersistableBundle)` ([https://developer.android.com/reference/android/app/Activity.html#onRestoreInstanceState\(android.os.Bundle, android.os.PersistableBundle\)](https://developer.android.com/reference/android/app/Activity.html#onRestoreInstanceState(android.os.Bundle, android.os.PersistableBundle)))

`onPause()` ([https://developer.android.com/reference/android/app/Activity.html#onPause\(\)](https://developer.android.com/reference/android/app/Activity.html#onPause()))

onSearchRequested

Added in API level 23 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
boolean onSearchRequested (SearchEvent (https://developer.android.com/reference/android/view/SearchEvent.html))
```

This hook is called when the user signals the desire to start a search.

You can use this function as a simple way to launch the search UI, in response to a menu item,

This site uses cookies to store your preferences for site-specific language options.

`/reference/android/app/Activity.html#startSearch(java.lang.String, boolean, android.os.Bundle, boolean))`, which launches search for the current activity as specified in its manifest, see **SearchManager** (<https://developer.android.com/reference/android/app/SearchManager.html>).

You can override this function to force global search, e.g. in response to a dedicated search key, or to block search entirely (by simply returning false).

Note: when running in a **UI_MODE_TYPE_TELEVISION** (https://developer.android.com/reference/android/content/res/Configuration.html#UI_MODE_TYPE_TELEVISION), the default implementation changes to simply return false and you must supply your own custom implementation if you want to support search.

Parameters	
<code>searchEvent</code>	SearchEvent : The SearchEvent (https://developer.android.com/reference/android/view/SearchEvent.html) that signaled this search.
Returns	
<code>boolean</code>	Returns <code>true</code> if search launched, and <code>false</code> if the activity does not respond to search. The default implementation always returns <code>true</code> , except when in UI_MODE_TYPE_TELEVISION (https://developer.android.com/reference/android/content/res/Configuration.html#UI_MODE_TYPE_TELEVISION) mode where it returns <code>false</code> .

See also:

SearchManager (<https://developer.android.com/reference/android/app/SearchManager.html>)

onSearchRequested

Added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

`boolean onSearchRequested ()`

Called when the user signals the desire to start a search.

Returns	
<code>boolean</code>	true if search launched, false if activity refuses (blocks)

See also:

This site uses cookies to store your preferences for site-specific language options.

OK

```
/app/Activity.html#onSearchRequested(android.view.SearchEvent))
```

onStateNotSaved

Added in API level 23 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
void onStateNotSaved ()
```

Called when an `onResume()` ([https://developer.android.com/reference/android/app/Activity.html#onResume\(\)](https://developer.android.com/reference/android/app/Activity.html#onResume())) is coming up, prior to other pre-resume callbacks such as `onNewIntent(Intent)` ([https://developer.android.com/reference/android/app/Activity.html#onNewIntent\(android.content.Intent\)](https://developer.android.com/reference/android/app/Activity.html#onNewIntent(android.content.Intent))) and `onActivityResult(int, int, Intent)` ([https://developer.android.com/reference/android/app/Activity.html#onActivityResult\(int, int, android.content.Intent\)](https://developer.android.com/reference/android/app/Activity.html#onActivityResult(int, int, android.content.Intent))). This is primarily intended to give the activity a hint that its state is no longer saved -- it will generally be called after `onSaveInstanceState(Bundle)` ([https://developer.android.com/reference/android/app/Activity.html#onSaveInstanceState\(android.os.Bundle\)](https://developer.android.com/reference/android/app/Activity.html#onSaveInstanceState(android.os.Bundle))) and prior to the activity being resumed/started again.

onTouchEvent

Added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
boolean onTouchEvent (MotionEvent (https://developer.android.com/reference/android/view/MotionEvent))
```

Called when a touch screen event was not handled by any of the views under it. This is most useful to process touch events that happen outside of your window bounds, where there is no view to receive it.

Parameters

<code>event</code>	<code>MotionEvent</code> : The touch screen event being processed.
--------------------	--

Returns

<code>boolean</code>	Return true if you have consumed the event, false if you haven't. The default implementation always returns false.
----------------------	--

This site uses cookies to store your preferences for site-specific language options.

```
boolean onTrackballEvent (MotionEvent (https://developer.android.com/reference/android.view.View#onTrackballEvent(MotionEvent))
```

Called when the trackball was moved and not handled by any of the views inside of the activity. So, for example, if the trackball moves while focus is on a button, you will receive a call here because buttons do not normally do anything with trackball events. The call here happens *before* trackball movements are converted to DPAD key events, which then get sent back to the view hierarchy, and will be processed at the point for things like focus navigation.

Parameters

event	MotionEvent : The trackball event being processed.
--------------	---

Returns

boolean	Return true if you have consumed the event, false if you haven't. The default implementation always returns false.
----------------	--

onTrimMemory Added in API level 14 (https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

```
void onTrimMemory (int level)
```

Called when the operating system has determined that it is a good time for a process to trim unneeded memory from its process. This will happen for example when it goes in the background and there is not enough memory to keep as many background processes running as desired. You should never compare to exact values of the level, since new intermediate values may be added -- you will typically want to compare if the value is greater or equal to a level you are interested in.

To retrieve the processes current trim level at any point, you can use

```
ActivityManager.getMyMemoryState(RunningAppProcessInfo) (https://developer.android.com/reference/android/app/ActivityManager.html#getMyMemoryState(android.app.ActivityManager.RunningAppProcessInfo)).
```

Parameters

level	int : The context of the trim, giving a hint of the amount of trimming the application may like to perform. May be TRIM_MEMORY_COMPLETE (https://developer.android.com/reference/android/content/ComponentCallbacks2.html#TRIM_MEMORY_COMPLETE), TRIM_MEMORY_MODERATE (https://developer.android.com/reference/android/content/ComponentCallbacks2.html#TRIM_MEMORY_MODERATE), TRIM_MEMORY_BACKGROUND
--------------	---

This site uses cookies to store your preferences for site-specific language options.


```
(https://developer.android.com/reference/android/content/ComponentCallbacks2.html#TRIM\_MEMORY\_UI\_HIDDEN), TRIM_MEMORY_RUNNING_CRITICAL
(https://developer.android.com/reference/android/content/ComponentCallbacks2.html#TRIM\_MEMORY\_RUNNING\_CRITICAL), TRIM_MEMORY_RUNNING_LOW
(https://developer.android.com/reference/android/content/ComponentCallbacks2.html#TRIM\_MEMORY\_RUNNING\_LOW), OR
TRIM_MEMORY_RUNNING_MODERATE (https://developer.android.com/reference/android/content/ComponentCallbacks2.html#TRIM\_MEMORY\_RUNNING\_MODERATE).
```

onUserInteraction

Added in API level 3 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
void onUserInteraction ()
```

Called whenever a key, touch, or trackball event is dispatched to the activity. Implement this method if you wish to know that the user has interacted with the device in some way while your activity is running. This callback and `onUserLeaveHint()` ([https://developer.android.com/reference/android/app/Activity.html#onUserLeaveHint\(\)](https://developer.android.com/reference/android/app/Activity.html#onUserLeaveHint())) are intended to help activities manage status bar notifications intelligently; specifically, for helping activities determine the proper time to cancel a notification.

All calls to your activity's `onUserLeaveHint()` ([https://developer.android.com/reference/android/app/Activity.html#onUserLeaveHint\(\)](https://developer.android.com/reference/android/app/Activity.html#onUserLeaveHint())) callback will be accompanied by calls to `onUserInteraction()` ([https://developer.android.com/reference/android/app/Activity.html#onUserInteraction\(\)](https://developer.android.com/reference/android/app/Activity.html#onUserInteraction())). This ensures that your activity will be told of relevant user activity such as pulling down the notification pane and touching an item there.

Note that this callback will be invoked for the touch down action that begins a touch gesture, but may not be invoked for the touch-moved and touch-up actions that follow.

See also:

`onUserLeaveHint()` ([https://developer.android.com/reference/android/app/Activity.html#onUserLeaveHint\(\)](https://developer.android.com/reference/android/app/Activity.html#onUserLeaveHint()))

onVisibleBehindCanceled

Added in API level 20 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

This site uses cookies to store your preferences for site-specific language options.

OK

launched. Activities that override this method must call `super.onVisibleBehindCanceled()` or a `SuperNotCalledException` will be thrown.

When this method is called the activity has 500 msec to release any resources it may be using while visible in the background. If the activity has not returned from this method in 500 msec the system will destroy the activity and kill the process in order to recover the resources for another process. Otherwise `onStop()` ([https://developer.android.com/reference/android/app/Activity.html#onStop\(\)](https://developer.android.com/reference/android/app/Activity.html#onStop())) will be called following return.

See also:

`requestVisibleBehind(boolean)` ([https://developer.android.com/reference/android/app/Activity.html#requestVisibleBehind\(boolean\)](https://developer.android.com/reference/android/app/Activity.html#requestVisibleBehind(boolean)))

onWindowAttributesChanged

Added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
void onWindowAttributesChanged (WindowManager.LayoutParams (https://developer.android.c
```

This is called whenever the current window attributes change.

Parameters	
<code>params</code>	<code>WindowManager.LayoutParams</code>

onWindowFocusChanged

Added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
void onWindowFocusChanged (boolean hasFocus)
```

Called when the current `Window` (<https://developer.android.com/reference/android/view/Window.html>) of the activity gains or loses focus. This is the best indicator of whether this activity is visible to the user. The default implementation clears the key tracking state, so should always be called.

Note that this provides information about global focus state, which is managed independently of activity lifecycles. As such, while focus changes will generally have some relation to lifecycle changes (an activity that is stopped will not generally get window focus), you should not rely on any particular order between the callbacks here and those in the other lifecycle methods such as `onResume()` ([https://developer.android.com/reference/android/app/Activity.html#onResume\(\)](https://developer.android.com/reference/android/app/Activity.html#onResume()))

This site uses cookies to store your preferences for site-specific language options.

dialogs or popups that take input focus, in which case the activity itself will not have focus when the other windows have it. Likewise, the system may display system-level windows (such as the status bar notification panel or a system alert) which will temporarily take window input focus without pausing the foreground activity.

Parameters	
<code>hasFocus</code>	<code>boolean</code> : Whether the window of this activity has focus.

See also:

`hasWindowFocus()` ([https://developer.android.com/reference/android/app/Activity.html#hasWindowFocus\(\)](https://developer.android.com/reference/android/app/Activity.html#hasWindowFocus()))

`onResume()` ([https://developer.android.com/reference/android/app/Activity.html#onResume\(\)](https://developer.android.com/reference/android/app/Activity.html#onResume()))

`onWindowFocusChanged(boolean)` ([https://developer.android.com/reference/android/view/View.html#onWindowFocusChanged\(boolean\)](https://developer.android.com/reference/android/view/View.html#onWindowFocusChanged(boolean)))

onWindowStartingActionMode

Added in API level 23 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

`ActionMode` (<https://developer.android.com/reference/android/view/ActionMode.html>) `onWindowStart:`
`int type`)

Called when an action mode is being started for this window. Gives the callback an opportunity to handle the action mode in its own unique and beautiful way. If this method returns null the system can choose a way to present the mode or choose not to start the mode at all.

Parameters	
<code>callback</code>	<code>ActionMode.Callback</code> : Callback to control the lifecycle of this action mode
<code>type</code>	<code>int</code> : One of <code>TYPE_PRIMARY</code> (https://developer.android.com/reference/android/view/ActionMode.html#TYPE_PRIMARY) or <code>TYPE_FLOATING</code> (https://developer.android.com/reference/android/view/ActionMode.html#TYPE_FLOATING).

Returns	
<code>ActionMode</code>	The <code>ActionMode</code> that was started, or null if the system should present it

This site uses cookies to store your preferences for site-specific language options.

OK

/view/ActionMode.html)

onWindowStartingActionMode

Added in API level 11 (https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

ActionMode (https://developer.android.com/reference/android/view/ActionMode.html) onWindowStart:

Give the Activity a chance to control the UI for an action mode requested by the system.

Note: If you are looking for a notification callback that an action mode has been started for this activity, see `onActionModeStarted(ActionMode)` (https://developer.android.com/reference/android/app/Activity.html#onActionModeStarted(android.view.ActionMode)).

Parameters	
<code>callback</code>	<code>ActionMode.Callback</code> : The callback that should control the new action mode
Returns	
<code>ActionMode</code> (https://developer.android.com/reference/android/view/ActionMode.html)	The new action mode, or <code>null</code> if the activity does not want to provide special handling for this action mode. (It will be handled by the system.)

openContextMenu

Added in API level 1 (https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

void openContextMenu (View (https://developer.android.com/reference/android/view/View.html) vi

Programmatically opens the context menu for a particular `view`. The `view` should have been added via `registerForContextMenu(View)` (https://developer.android.com/reference/android/app/Activity.html#registerForContextMenu(android.view.View)).

Parameters	
<code>view</code>	<code>View</code> : The view to show the context menu for.

This site uses cookies to store your preferences for site-specific langu options.

OK

```
void openOptionsMenu ()
```

Programmatically opens the options menu. If the options menu is already open, this method does nothing.

overridePendingTransition

Added in API level 5 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
void overridePendingTransition (int enterAnim,
                               int exitAnim)
```

Call immediately after one of the flavors of `startActivity(Intent)` ([https://developer.android.com/reference/android/app/Activity.html#startActivity\(android.content.Intent\)](https://developer.android.com/reference/android/app/Activity.html#startActivity(android.content.Intent))) or `finish()` ([https://developer.android.com/reference/android/app/Activity.html#finish\(\)](https://developer.android.com/reference/android/app/Activity.html#finish())) to specify an explicit transition animation to perform next.

As of **JELLY_BEAN** (https://developer.android.com/reference/android/os/Build.VERSION_CODES.html#JELLY_BEAN) an alternative to using this with starting activities is to supply the desired animation information through a `ActivityOptions` (<https://developer.android.com/reference/android/app/ActivityOptions.html>) bundle to `startActivity(Intent, Bundle)` ([https://developer.android.com/reference/android/app/Activity.html#startActivity\(android.content.Intent, android.os.Bundle\)](https://developer.android.com/reference/android/app/Activity.html#startActivity(android.content.Intent, android.os.Bundle))) or a related function. This allows you to specify a custom animation even when starting an activity from outside the context of the current top activity.

Parameters	
<code>enterAnim</code>	<code>int</code> : A resource ID of the animation resource to use for the incoming activity. Use 0 for no animation.
<code>exitAnim</code>	<code>int</code> : A resource ID of the animation resource to use for the outgoing activity. Use 0 for no animation.

postponeEnterTransition

Added in API level 21 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
void postponeEnterTransition ()
```

Postpone the entering activity transition when Activity was started with

This site uses cookies to store your preferences for site-specific language options.

```
/app/ActivityOptions.html#makeSceneTransitionAnimation(android.app.Activity,
android.util.Pair<android.view.View, java.lang.String>...)).
```

This method gives the Activity the ability to delay starting the entering and shared element transitions until all data is loaded. Until then, the Activity won't draw into its window, leaving the window transparent. This may also cause the returning animation to be delayed until data is ready. This method should be called in `onCreate(android.os.Bundle)` ([https://developer.android.com/reference/android/app/Activity.html#onCreate\(android.os.Bundle\)](https://developer.android.com/reference/android/app/Activity.html#onCreate(android.os.Bundle))) or in `onActivityResult(int, android.content.Intent)` ([https://developer.android.com/reference/android/app/Activity.html#onActivityResult\(int, android.content.Intent\)](https://developer.android.com/reference/android/app/Activity.html#onActivityResult(int, android.content.Intent))).

`startPostponedEnterTransition()` ([https://developer.android.com/reference/android/app/Activity.html#startPostponedEnterTransition\(\)](https://developer.android.com/reference/android/app/Activity.html#startPostponedEnterTransition())) must be called to allow the Activity to start the transitions. If the Activity did not use `makeSceneTransitionAnimation(Activity, android.util.Pair[])` ([https://developer.android.com/reference/android/app/ActivityOptions.html#makeSceneTransitionAnimation\(android.app.Activity, android.util.Pair<android.view.View, java.lang.String>...\)\)](https://developer.android.com/reference/android/app/ActivityOptions.html#makeSceneTransitionAnimation(android.app.Activity, android.util.Pair<android.view.View, java.lang.String>...))), then this method does nothing.

recreate

Added in API level 11 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
void recreate ()
```

Cause this Activity to be recreated with a new instance. This results in essentially the same flow as when the Activity is created due to a configuration change -- the current instance will go through its lifecycle to `onDestroy()` ([https://developer.android.com/reference/android/app/Activity.html#onDestroy\(\)](https://developer.android.com/reference/android/app/Activity.html#onDestroy())) and a new instance then created after it.

registerForContextMenu

Added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
void registerForContextMenu (View (https://developer.android.com/reference/android/view/View.html))
```

Registers a context menu to be shown for the given view (multiple views can show the context menu). This method will set the `View.OnCreateContextMenuListener` (<https://developer.android.com/reference/android/view/View.OnCreateContextMenuListener.html>) on the view to this activity, so `onCreateContextMenu(ContextMenu, View, ContextMenuInfo)`

This site uses cookies to store your preferences for site-specific language options.

OK

`android.view.ContextMenu.ContextMenuInfo))` will be called when it is time to show the context menu.

Parameters

<code>view</code>	View: The view that should show a context menu.
-------------------	--

See also:

`unregisterForContextMenu(View)` ([https://developer.android.com/reference/android/app/Activity.html#unregisterForContextMenu\(android.view.View\)](https://developer.android.com/reference/android/app/Activity.html#unregisterForContextMenu(android.view.View)))

releaseInstance

Added in API level 21 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

`boolean releaseInstance ()`

Ask that the local app instance of this activity be released to free up its memory. This is asking for the activity to be destroyed, but does **not** finish the activity -- a new instance of the activity will later be re-created if needed due to the user navigating back to it.

Returns

<code>boolean</code>	Returns true if the activity was in a state that it has started the process of destroying its current instance; returns false if for any reason this could not be done: it is currently visible to the user, it is already being destroyed, it is being finished, it hasn't yet saved its state, etc.
----------------------	---

removeDialog

Added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

`void removeDialog (int id)`

This method was deprecated in API level 13.

Use the new `DialogFragment` (<https://developer.android.com/reference/android/app/DialogFragment.html>) class with `FragmentManager` (<https://developer.android.com/reference/android/app/FragmentManager.html>) instead; this is also available on older platforms through the Android compatibility package.

~~Remove any internal references to a dialog managed by this Activity. If the dialog is showing, it will~~

This site uses cookies to store your preferences for site-specific language options.

This can be useful if you know that you will never show a dialog again and want to avoid the overhead of saving and restoring it in the future.

As of **GINGERBREAD** (https://developer.android.com/reference/android/os/Build.VERSION_CODES.html#GINGERBREAD), this function will not throw an exception if you try to remove an ID that does not currently have an associated dialog.

Parameters	
id	int : The id of the managed dialog.

See also:

onCreateDialog(int, Bundle) ([https://developer.android.com/reference/android/app/Activity.html#onCreateDialog\(int, android.os.Bundle\)](https://developer.android.com/reference/android/app/Activity.html#onCreateDialog(int, android.os.Bundle)))

onPrepareDialog(int, Dialog, Bundle) ([https://developer.android.com/reference/android/app/Activity.html#onPrepareDialog\(int, android.app.Dialog, android.os.Bundle\)](https://developer.android.com/reference/android/app/Activity.html#onPrepareDialog(int, android.app.Dialog, android.os.Bundle)))

showDialog(int) ([https://developer.android.com/reference/android/app/Activity.html#showDialog\(int\)](https://developer.android.com/reference/android/app/Activity.html#showDialog(int)))

dismissDialog(int) ([https://developer.android.com/reference/android/app/Activity.html#dismissDialog\(int\)](https://developer.android.com/reference/android/app/Activity.html#dismissDialog(int)))

reportFullyDrawn Added in API level 19 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
void reportFullyDrawn ()
```

Report to the system that your app is now fully drawn, purely for diagnostic purposes (calling it does not impact the visible behavior of the activity). This is only used to help instrument application launch times, so that the app can report when it is fully in a usable state; without this, the only thing the system itself can determine is the point at which the activity's window is *first* drawn and displayed. To participate in app launch time measurement, you should always call this method after first launch (when **onCreate(android.os.Bundle)** ([https://developer.android.com/reference/android/app/Activity.html#onCreate\(android.os.Bundle\)](https://developer.android.com/reference/android/app/Activity.html#onCreate(android.os.Bundle))) is called), at the point where you have entirely drawn your UI and populated with all of the significant data. You can safely call this method any time after first launch as well, in which case it will simply be ignored.

This site uses cookies to store your preferences for site-specific language options.

OK

requestDragAndDropPermissions

Added in API level 24 (https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

DragAndDropPermissions (https://developer.android.com/reference/android/view/DragAndDropPermissi

Create DragAndDropPermissions (https://developer.android.com/reference/android/view/DragAndDropPermissions.html) object bound to this activity and controlling the access permissions for content URIs associated with the DragEvent (https://developer.android.com/reference/android/view/DragEvent.html).

Parameters	
event	DragEvent: Drag event
Returns	
DragAndDropPermissions (https://developer.android.com/reference/android/view/DragAndDropPermissions.html)	The DragAndDropPermissions (https://developer.android.com/reference/android/view/DragAndDropPermissions.html) object used to control access to the content URIs. Null if no content URIs are associated with the event or if permissions could not be granted.

requestPermissions

Added in API level 23 (https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

```
void requestPermissions (String[] (https://developer.android.com/reference/java/lang/String.h
int requestCode)
```

Requests permissions to be granted to this application. These permissions must be requested in your manifest, they should not be granted to your app, and they should have protection level #PROTECTION_DANGEROUS dangerous (https://developer.android.com/reference/android/content/pm/PermissionInfo.html), regardless whether they are declared by the platform or a third-party app.

Normal permissions PROTECTION_NORMAL (https://developer.android.com/reference/android/content/pm/PermissionInfo.html#PROTECTION_NORMAL) are granted at install time if requested in the manifest.

Signature permissions PROTECTION_SIGNATURE (https://developer.android.com/reference/android/content/pm/PermissionInfo.html#PROTECTION_SIGNATURE) are granted at install time if requested in the manifest and the signature of your app matches the signature of the app declaring the permissions.

If your app does not have the requested permissions the user will be presented with UI for accepting

This site uses cookies to store your preferences for site-specific language options.

`/reference/android/app/Activity.html#onRequestPermissionsResult(int, java.lang.String[], int[]))`
reporting whether the permissions were granted or not.

Note that requesting a permission does not guarantee it will be granted and your app should be able to run without having this permission.

This method may start an activity allowing the user to choose which permissions to grant and which to reject. Hence, you should be prepared that your activity may be paused and resumed. Further, granting some permissions may require a restart of you application. In such a case, the system will recreate the activity stack before delivering the result to `onRequestPermissionsResult(int, String[], int[])` ([https://developer.android.com/reference/android/app/Activity.html#onRequestPermissionsResult\(int, java.lang.String\[\], int\[\]\)](https://developer.android.com/reference/android/app/Activity.html#onRequestPermissionsResult(int, java.lang.String[], int[]))).

When checking whether you have a permission you should use `checkSelfPermission(String)` ([https://developer.android.com/reference/android/content/ContextWrapper.html#checkSelfPermission\(java.lang.String\)](https://developer.android.com/reference/android/content/ContextWrapper.html#checkSelfPermission(java.lang.String))).

Calling this API for permissions already granted to your app would show UI to the user to decide whether the app can still hold these permissions. This can be useful if the way your app uses data guarded by the permissions changes significantly.

You cannot request a permission if your activity sets `noHistory` (https://developer.android.com/reference/android/R.styleable.html#AndroidManifestActivity_noHistory) to `true` because in this case the activity would not receive result callbacks including `onRequestPermissionsResult(int, String[], int[])` ([https://developer.android.com/reference/android/app/Activity.html#onRequestPermissionsResult\(int, java.lang.String\[\], int\[\]\)](https://developer.android.com/reference/android/app/Activity.html#onRequestPermissionsResult(int, java.lang.String[], int[]))).

The `RuntimePermissions` (<http://developer.android.com/samples/RuntimePermissions/index.html>) sample app demonstrates how to use this method to request permissions at run time.

Parameters	
<code>permissions</code>	<code>String</code> : The requested permissions. Must me non-null and not empty.
<code>requestCode</code>	<code>int</code> : Application specific request code to match with a result reported to <code>onRequestPermissionsResult(int, String[], int[])</code> (https://developer.android.com/reference/android/app/Activity.html#onRequestPermissionsResult(int, java.lang.String[], int[])). Should be <code>>= 0</code> .

See also:

This site uses cookies to store your preferences for site-specific langu options.

OK

`checkSelfPermission(String)` ([https://developer.android.com/reference/android/content/ContextWrapper.html#checkSelfPermission\(java.lang.String\)](https://developer.android.com/reference/android/content/ContextWrapper.html#checkSelfPermission(java.lang.String)))

`shouldShowRequestPermissionRationale(String)` ([https://developer.android.com/reference/android/app/Activity.html#shouldShowRequestPermissionRationale\(java.lang.String\)](https://developer.android.com/reference/android/app/Activity.html#shouldShowRequestPermissionRationale(java.lang.String)))

requestShowKeyboardShortcuts

Added in API level 24 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
void requestShowKeyboardShortcuts ()
```

Request the Keyboard Shortcuts screen to show up. This will trigger

`onProvideKeyboardShortcuts(List, Menu, int)` ([https://developer.android.com/reference/android/app/Activity.html#onProvideKeyboardShortcuts\(java.util.List<android.view.KeyboardShortcutGroup>, android.view.Menu, int\)](https://developer.android.com/reference/android/app/Activity.html#onProvideKeyboardShortcuts(java.util.List<android.view.KeyboardShortcutGroup>, android.view.Menu, int))) to retrieve the shortcuts for the foreground activity.

requestVisibleBehind

Added in API level 21 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
boolean requestVisibleBehind (boolean visible)
```

Activities that want to remain visible behind a translucent activity above them must call this method anytime between the start of `onResume()` ([https://developer.android.com/reference/android/app/Activity.html#onResume\(\)](https://developer.android.com/reference/android/app/Activity.html#onResume())) and the return from `onPause()` ([https://developer.android.com/reference/android/app/Activity.html#onPause\(\)](https://developer.android.com/reference/android/app/Activity.html#onPause())). If this call is successful then the activity will remain visible after `onPause()` ([https://developer.android.com/reference/android/app/Activity.html#onPause\(\)](https://developer.android.com/reference/android/app/Activity.html#onPause())) is called, and is allowed to continue playing media in the background.

The actions of this call are reset each time that this activity is brought to the front. That is, every time `onResume()` ([https://developer.android.com/reference/android/app/Activity.html#onResume\(\)](https://developer.android.com/reference/android/app/Activity.html#onResume())) is called the activity will be assumed to not have requested visible behind. Therefore, if you want this activity to continue to be visible in the background you must call this method again.

Only fullscreen opaque activities may make this call. I.e. this call is a nop for dialog and translucent activities.

Under all circumstances, the activity must stop playing and release resources prior to or within a call

This site uses cookies to store your preferences for site-specific language options.

OK

False will be returned any time this method is called between the return of `onPause` and the next call to `onResume`.

Parameters	
<code>visible</code>	<code>boolean</code> : true to notify the system that the activity wishes to be visible behind other translucent activities, false to indicate otherwise. Resources must be released when passing false to this method.

Returns	
<code>boolean</code>	the resulting visibility state. If true the activity will remain visible beyond <code>onPause()</code> (https://developer.android.com/reference/android/app/Activity.html#onPause()) if the next activity is translucent or not fullscreen. If false then the activity may not count on being visible behind other translucent activities, and must stop any media playback and release resources. Returning false may occur in lieu of a call to <code>onVisibleBehindCanceled()</code> (https://developer.android.com/reference/android/app/Activity.html#onVisibleBehindCanceled()) so the return value must be checked.

See also:

`onVisibleBehindCanceled()` ([https://developer.android.com/reference/android/app/Activity.html#onVisibleBehindCanceled\(\)](https://developer.android.com/reference/android/app/Activity.html#onVisibleBehindCanceled()))

requestWindowFeature

Added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
boolean requestWindowFeature (int featureId)
```

Enable extended window features. This is a convenience for calling

`getWindow().requestFeature()` ([https://developer.android.com/reference/android/view/Window.html#requestFeature\(int\)](https://developer.android.com/reference/android/view/Window.html#requestFeature(int))).

Parameters	
<code>featureId</code>	<code>int</code> : The desired feature as defined in <code>Window</code> (https://developer.android.com/reference/android/view/Window.html).

Returns	

This site uses cookies to store your preferences for site-specific language options.

OK

See also:

`requestFeature(int)` ([https://developer.android.com/reference/android/view/Window.html#requestFeature\(int\)](https://developer.android.com/reference/android/view/Window.html#requestFeature(int)))

runOnUiThread

Added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

`void runOnUiThread (Runnable` (<https://developer.android.com/reference/java/lang/Runnable.html>)

Runs the specified action on the UI thread. If the current thread is the UI thread, then the action is executed immediately. If the current thread is not the UI thread, the action is posted to the event queue of the UI thread.

Parameters	
<code>action</code>	<code>Runnable</code> : the action to run on the UI thread

setActionBar

Added in API level 21 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

`void setActionBar (Toolbar` (<https://developer.android.com/reference/android/widget/Toolbar.html>)

Set a `Toolbar` (<https://developer.android.com/reference/android/widget/Toolbar.html>) to act as the `ActionBar` (<https://developer.android.com/reference/android/app/ActionBar.html>) for this Activity window.

When set to a non-null value the `getActionBar()` ([https://developer.android.com/reference/android/app/Activity.html#getActionBar\(\)](https://developer.android.com/reference/android/app/Activity.html#getActionBar())) method will return an `ActionBar` (<https://developer.android.com/reference/android/app/ActionBar.html>) object that can be used to control the given toolbar as if it were a traditional window decor action bar. The toolbar's menu will be populated with the Activity's options menu and the navigation button will be wired through the standard `home` (<https://developer.android.com/reference/android/R.id.html#home>) menu select action.

In order to use a Toolbar within the Activity's window content the application must not request the window feature `FEATURE_ACTION_BAR` (https://developer.android.com/reference/android/view/Window.html#FEATURE_ACTION_BAR).

Parameters

This site uses cookies to store your preferences for site-specific language options.

OK

setContentTransitionManager

Added in API level 21 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
void setContentTransitionManager (TransitionManager (https://developer.android.com/refere
```

Set the [TransitionManager](https://developer.android.com/reference/android/transition/TransitionManager.html) (<https://developer.android.com/reference/android/transition/TransitionManager.html>) to use for default transitions in this window. Requires [FEATURE_CONTENT_TRANSITIONS](https://developer.android.com/reference/android/view/Window.html#FEATURE_CONTENT_TRANSITIONS) (https://developer.android.com/reference/android/view/Window.html#FEATURE_CONTENT_TRANSITIONS).

Parameters

tm	TransitionManager : The TransitionManager to use for scene changes.
----	---

setContentView

Added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
void setContentView (View (https://developer.android.com/reference/android/view/View.html) view, ViewGroup.LayoutParams (https://developer.android.com/reference/android/view
```

Set the activity content to an explicit view. This view is placed directly into the activity's view hierarchy. It can itself be a complex view hierarchy.

Parameters

view	View : The desired content to display.
params	ViewGroup.LayoutParams : Layout parameters for the view.

See also:

[setContentView\(android.view.View\)](https://developer.android.com/reference/android/app/Activity.html#setContentView(android.view.View)) ([https://developer.android.com/reference/android/app/Activity.html#setContentView\(android.view.View\)](https://developer.android.com/reference/android/app/Activity.html#setContentView(android.view.View)))

[setContentView\(int\)](https://developer.android.com/reference/android/app/Activity.html#setContentView(int)) ([https://developer.android.com/reference/android/app/Activity.html#setContentView\(int\)](https://developer.android.com/reference/android/app/Activity.html#setContentView(int)))

setContentView

Added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
void setContentView (View (https://developer.android.com/reference/android/view/View.html) view)
```

This site uses cookies to store your preferences for site-specific language options.

OK

of the specified view are ignored. Both the width and the height of the view are set by default to `MATCH_PARENT` (https://developer.android.com/reference/android/view/ViewGroup.LayoutParams.html#MATCH_PARENT). To use your own layout parameters, invoke `setContentView(android.view.View, android.view.ViewGroup.LayoutParams)` ([https://developer.android.com/reference/android/app/Activity.html#setContentView\(android.view.View, android.view.ViewGroup.LayoutParams\)](https://developer.android.com/reference/android/app/Activity.html#setContentView(android.view.View, android.view.ViewGroup.LayoutParams))) instead.

Parameters	
<code>view</code>	<code>View</code> : The desired content to display.

See also:

`setContentView(int)` ([https://developer.android.com/reference/android/app/Activity.html#setContentView\(int\)](https://developer.android.com/reference/android/app/Activity.html#setContentView(int)))

`setContentView(android.view.View, android.view.ViewGroup.LayoutParams)` ([https://developer.android.com/reference/android/app/Activity.html#setContentView\(android.view.View, android.view.ViewGroup.LayoutParams\)](https://developer.android.com/reference/android/app/Activity.html#setContentView(android.view.View, android.view.ViewGroup.LayoutParams)))

setContentView

Added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
void setContentView (int layoutResID)
```

Set the activity content from a layout resource. The resource will be inflated, adding all top-level views to the activity.

Parameters	
<code>layoutResID</code>	<code>int</code> : Resource ID to be inflated.

See also:

`setContentView(android.view.View)` ([https://developer.android.com/reference/android/app/Activity.html#setContentView\(android.view.View\)](https://developer.android.com/reference/android/app/Activity.html#setContentView(android.view.View)))

`setContentView(android.view.View, android.view.ViewGroup.LayoutParams)` ([https://developer.android.com/reference/android/app/Activity.html#setContentView\(android.view.View, android.view.ViewGroup.LayoutParams\)](https://developer.android.com/reference/android/app/Activity.html#setContentView(android.view.View, android.view.ViewGroup.LayoutParams)))

This site uses cookies to store your preferences for site-specific language options. OK

setDefaultKeyMode

Added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
void setDefaultKeyMode (int mode)
```

Select the default key handling for this activity. This controls what will happen to key events that are not otherwise handled. The default mode ([DEFAULT_KEYS_DISABLE](https://developer.android.com/reference/android/app/Activity.html#DEFAULT_KEYS_DISABLE) (https://developer.android.com/reference/android/app/Activity.html#DEFAULT_KEYS_DISABLE)) will simply drop them on the floor. Other modes allow you to launch the dialer ([DEFAULT_KEYS_DIALER](https://developer.android.com/reference/android/app/Activity.html#DEFAULT_KEYS_DIALER) (https://developer.android.com/reference/android/app/Activity.html#DEFAULT_KEYS_DIALER)), execute a shortcut in your options menu without requiring the menu key be held down ([DEFAULT_KEYS_SHORTCUT](https://developer.android.com/reference/android/app/Activity.html#DEFAULT_KEYS_SHORTCUT) (https://developer.android.com/reference/android/app/Activity.html#DEFAULT_KEYS_SHORTCUT)), or launch a search ([DEFAULT_KEYS_SEARCH_LOCAL](https://developer.android.com/reference/android/app/Activity.html#DEFAULT_KEYS_SEARCH_LOCAL) (https://developer.android.com/reference/android/app/Activity.html#DEFAULT_KEYS_SEARCH_LOCAL) and [DEFAULT_KEYS_SEARCH_GLOBAL](https://developer.android.com/reference/android/app/Activity.html#DEFAULT_KEYS_SEARCH_GLOBAL) (https://developer.android.com/reference/android/app/Activity.html#DEFAULT_KEYS_SEARCH_GLOBAL)).

Note that the mode selected here does not impact the default handling of system keys, such as the "back" and "menu" keys, and your activity and its views always get a first chance to receive and handle all application keys.

Parameters	
mode	int: The desired default key mode constant.

See also:

[DEFAULT_KEYS_DISABLE](https://developer.android.com/reference/android/app/Activity.html#DEFAULT_KEYS_DISABLE) (https://developer.android.com/reference/android/app/Activity.html#DEFAULT_KEYS_DISABLE)

[DEFAULT_KEYS_DIALER](https://developer.android.com/reference/android/app/Activity.html#DEFAULT_KEYS_DIALER) (https://developer.android.com/reference/android/app/Activity.html#DEFAULT_KEYS_DIALER)

[DEFAULT_KEYS_SHORTCUT](https://developer.android.com/reference/android/app/Activity.html#DEFAULT_KEYS_SHORTCUT) (https://developer.android.com/reference/android/app/Activity.html#DEFAULT_KEYS_SHORTCUT)

[DEFAULT_KEYS_SEARCH_LOCAL](https://developer.android.com/reference/android/app/Activity.html#DEFAULT_KEYS_SEARCH_LOCAL) (https://developer.android.com/reference/android/app/Activity.html#DEFAULT_KEYS_SEARCH_LOCAL)

[DEFAULT_KEYS_SEARCH_GLOBAL](https://developer.android.com/reference/android/app/Activity.html#DEFAULT_KEYS_SEARCH_GLOBAL) (https://developer.android.com/reference/android/app/Activity.html#DEFAULT_KEYS_SEARCH_GLOBAL)

[onKeyDown\(int, KeyEvent\)](https://developer.android.com/reference/android/app/Activity.html#onKeyDown(int,%20KeyEvent)) ([https://developer.android.com/reference/android/app/Activity.html#onKeyDown\(int, android.view.KeyEvent\)](https://developer.android.com/reference/android/app/Activity.html#onKeyDown(int,%20KeyEvent)))

This site uses cookies to store your preferences for site-specific language options.

OK

setEnterSharedElementCallback

Added in API level 21 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
void setEnterSharedElementCallback (SharedElementCallback (https://developer.android.com/
```

When `makeSceneTransitionAnimation(Activity, android.view.View, String)`

(<https://developer.android.com/reference/android>

[/app/ActivityOptions.html#makeSceneTransitionAnimation\(android.app.Activity, android.view.View, java.lang.String\)\)](/app/ActivityOptions.html#makeSceneTransitionAnimation(android.app.Activity, android.view.View, java.lang.String))) was used to start an Activity, *callback* will be called to handle shared elements on

the *launched* Activity. This requires `FEATURE_ACTIVITY_TRANSITIONS`

(https://developer.android.com/reference/android/view/Window.html#FEATURE_ACTIVITY_TRANSITIONS).

Parameters

<code>callback</code>	<code>SharedElementCallback</code> : Used to manipulate shared element transitions on the launched Activity.
-----------------------	--

setExitSharedElementCallback

Added in API level 21 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
void setExitSharedElementCallback (SharedElementCallback (https://developer.android.com/
```

When `makeSceneTransitionAnimation(Activity, android.view.View, String)`

(<https://developer.android.com/reference/android>

[/app/ActivityOptions.html#makeSceneTransitionAnimation\(android.app.Activity, android.view.View, java.lang.String\)\)](/app/ActivityOptions.html#makeSceneTransitionAnimation(android.app.Activity, android.view.View, java.lang.String))) was used to start an Activity, *callback* will be called to handle shared elements on

the *launching* Activity. Most calls will only come when returning from the started Activity. This

requires `FEATURE_ACTIVITY_TRANSITIONS` (<https://developer.android.com/reference/android>

/view/Window.html#FEATURE_ACTIVITY_TRANSITIONS).

Parameters

<code>callback</code>	<code>SharedElementCallback</code> : Used to manipulate shared element transitions on the launching Activity.
-----------------------	---

setFeatureDrawable

Added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
void setFeatureDrawable (int featureId,
```

This site uses cookies to store your preferences for site-specific language options.

OK

Convenience for calling `setFeatureDrawable(int, Drawable)` ([https://developer.android.com/reference/android/view/Window.html#setFeatureDrawable\(int, android.graphics.drawable.Drawable\)](https://developer.android.com/reference/android/view/Window.html#setFeatureDrawable(int, android.graphics.drawable.Drawable))).

Parameters	
<code>featureId</code>	<code>int</code>
<code>drawable</code>	<code>Drawable</code>

setFeatureDrawableAlpha

Added in API level 23 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
void setFeatureDrawableAlpha (int featureId,  
                             int alpha)
```

Convenience for calling `setFeatureDrawableAlpha(int, int)` ([https://developer.android.com/reference/android/view/Window.html#setFeatureDrawableAlpha\(int, int\)](https://developer.android.com/reference/android/view/Window.html#setFeatureDrawableAlpha(int, int))).

Parameters	
<code>featureId</code>	<code>int</code>
<code>alpha</code>	<code>int</code>

setFeatureDrawableResource

Added in API level 23 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
void setFeatureDrawableResource (int featureId,  
                                int resId)
```

Convenience for calling `setFeatureDrawableResource(int, int)` ([https://developer.android.com/reference/android/view/Window.html#setFeatureDrawableResource\(int, int\)](https://developer.android.com/reference/android/view/Window.html#setFeatureDrawableResource(int, int))).

Parameters	
<code>featureId</code>	<code>int</code>
<code>resId</code>	<code>int</code>

This site uses cookies to store your preferences for site-specific language options.

OK

```
void setFeatureDrawableUri (int featureId,
                           Uri (https://developer.android.com/reference/android/net/Uri.html) uri)
```

Convenience for calling `setFeatureDrawableUri(int, Uri)` ([https://developer.android.com/reference/android/view/Window.html#setFeatureDrawableUri\(int, android.net.Uri\)](https://developer.android.com/reference/android/view/Window.html#setFeatureDrawableUri(int, android.net.Uri))).

Parameters	
<code>featureId</code>	<code>int</code>
<code>uri</code>	<code>Uri</code>

setFinishOnTouchOutside Added in API level 3 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
void setFinishOnTouchOutside (boolean finish)
```

Sets whether this activity is finished when touched outside its window's bounds.

Parameters	
<code>finish</code>	<code>boolean</code>

setImmersive Added in API level 18 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
void setImmersive (boolean i)
```

Adjust the current immersive mode setting. Note that changing this value will have no effect on the activity's `ActivityInfo` (<https://developer.android.com/reference/android/content/pm/ActivityInfo.html>) structure; that is, if `android:immersive` is set to `true` in the application's manifest entry for this activity, the `ActivityInfo.flags` (<https://developer.android.com/reference/android/content/pm/ActivityInfo.html#flags>) member will always have its `FLAG_IMMERSIVE` (https://developer.android.com/reference/android/content/pm/ActivityInfo.html#FLAG_IMMERSIVE) bit set.

Parameters	
<code>i</code>	<code>boolean</code>

See also:

This site uses cookies to store your preferences for site-specific language options.



`FLAG_IMMERSIVE` (https://developer.android.com/reference/android/content/pm/ActivityInfo.html#FLAG_IMMERSIVE)

setIntent

Added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
void setIntent (Intent (https://developer.android.com/reference/android/content/Intent.html) newIntent)
```

Change the intent returned by `getIntent()` ([https://developer.android.com/reference/android/app/Activity.html#getIntent\(\)](https://developer.android.com/reference/android/app/Activity.html#getIntent())). This holds a reference to the given intent; it does not copy it. Often used in conjunction with `onNewIntent(Intent)` ([https://developer.android.com/reference/android/app/Activity.html#onNewIntent\(android.content.Intent\)](https://developer.android.com/reference/android/app/Activity.html#onNewIntent(android.content.Intent))).

Parameters

<code>newIntent</code>	<code>Intent</code> : The new Intent object to return from <code>getIntent</code>
------------------------	---

See also:

`getIntent()` ([https://developer.android.com/reference/android/app/Activity.html#getIntent\(\)](https://developer.android.com/reference/android/app/Activity.html#getIntent()))

`onNewIntent(Intent)` ([https://developer.android.com/reference/android/app/Activity.html#onNewIntent\(android.content.Intent\)](https://developer.android.com/reference/android/app/Activity.html#onNewIntent(android.content.Intent)))

setMediaController

Added in API level 21 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
void setMediaController (MediaController (https://developer.android.com/reference/android/media/session/MediaController.html) mediaController)
```

Sets a `MediaController` (<https://developer.android.com/reference/android/media/session/MediaController.html>) to send media keys and volume changes to.

The controller will be tied to the window of this Activity. Media key and volume events which are received while the Activity is in the foreground will be forwarded to the controller and used to invoke transport controls or adjust the volume. This may be used instead of or in addition to `setVolumeControlStream(int)` ([https://developer.android.com/reference/android/app/Activity.html#setVolumeControlStream\(int\)](https://developer.android.com/reference/android/app/Activity.html#setVolumeControlStream(int))) to affect a specific session instead of a specific stream.

This site uses cookies to store your preferences for site-specific language options.

OK

default use null as the controller.

Parameters	
<code>controller</code>	MediaController : The controller for the session which should receive media keys and volume changes.

setProgress

Added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
void setProgress (int progress)
```

This method was deprecated in API level 24.

No longer supported starting in API 21.

Sets the progress for the progress bars in the title.

In order for the progress bar to be shown, the feature must be requested via

`requestWindowFeature(int)` ([https://developer.android.com/reference/android/app/Activity.html#requestWindowFeature\(int\)](https://developer.android.com/reference/android/app/Activity.html#requestWindowFeature(int))).

Parameters	
<code>progress</code>	int : The progress for the progress bar. Valid ranges are from 0 to 10000 (both inclusive). If 10000 is given, the progress bar will be completely filled and will fade out.

setProgressBarIndeterminate

Added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
void setProgressBarIndeterminate (boolean indeterminate)
```

This method was deprecated in API level 24.

No longer supported starting in API 21.

Sets whether the horizontal progress bar in the title should be indeterminate (the circular is always indeterminate).

In order for the progress bar to be shown, the feature must be requested via

This site uses cookies to store your preferences for site-specific language options.

OK

Parameters

<code>indeterminate</code>	<code>boolean</code> : Whether the horizontal progress bar should be indeterminate.
----------------------------	---

setProgressBarIndeterminateVisibility

Added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
void setProgressBarIndeterminateVisibility (boolean visible)
```

This method was deprecated in API level 24.

No longer supported starting in API 21.

Sets the visibility of the indeterminate progress bar in the title.

In order for the progress bar to be shown, the feature must be requested via

`requestWindowFeature(int)` ([https://developer.android.com/reference/android/app/Activity.html#requestWindowFeature\(int\)](https://developer.android.com/reference/android/app/Activity.html#requestWindowFeature(int))).

Parameters

<code>visible</code>	<code>boolean</code> : Whether to show the progress bars in the title.
----------------------	--

setProgressBarVisibility

Added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
void setProgressBarVisibility (boolean visible)
```

This method was deprecated in API level 24.

No longer supported starting in API 21.

Sets the visibility of the progress bar in the title.

In order for the progress bar to be shown, the feature must be requested via

`requestWindowFeature(int)` ([https://developer.android.com/reference/android/app/Activity.html#requestWindowFeature\(int\)](https://developer.android.com/reference/android/app/Activity.html#requestWindowFeature(int))).

Parameters

<code>visible</code>	<code>boolean</code> : Whether to show the progress bars in the title.
----------------------	--

This site uses cookies to store your preferences for site-specific language options.

OK

setRequestedOrientation

Added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
void setRequestedOrientation (int requestedOrientation)
```

Change the desired orientation of this activity. If the activity is currently in the foreground or otherwise impacting the screen orientation, the screen will immediately be changed (possibly causing the activity to be restarted). Otherwise, this will be used the next time the activity is visible.

Parameters	
<code>requestedOrientation</code>	<code>int</code> : An orientation constant as used in <code>ActivityInfo.screenOrientation</code> (https://developer.android.com/reference/android/content/pm/ActivityInfo.html#screenOrientation).

setResult

Added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
void setResult (int resultCode,  
                Intent (https://developer.android.com/reference/android/content/Intent.html) data)
```

Call this to set the result that your activity will return to its caller.

As of **GINGERBREAD** (https://developer.android.com/reference/android/os/Build.VERSION_CODES.html#GINGERBREAD), the Intent you supply here can have `Intent.FLAG_GRANT_READ_URI_PERMISSION` (https://developer.android.com/reference/android/content/Intent.html#FLAG_GRANT_READ_URI_PERMISSION) and/or `Intent.FLAG_GRANT_WRITE_URI_PERMISSION` (https://developer.android.com/reference/android/content/Intent.html#FLAG_GRANT_WRITE_URI_PERMISSION) set. This will grant the Activity receiving the result access to the specific URIs in the Intent. Access will remain until the Activity has finished (it will remain across the hosting process being killed and other temporary destruction) and will be added to any existing set of URI permissions it already holds.

Parameters	
<code>resultCode</code>	<code>int</code> : The result code to propagate back to the originating activity, often <code>RESULT_CANCELED</code> or <code>RESULT_OK</code>
<code>data</code>	<code>Intent</code> : The data to propagate back to the originating activity.

See also

This site uses cookies to store your preferences for site-specific language options.

OK

[/app/Activity.html#RESULT_CANCELED](#))

[RESULT_OK](#) (https://developer.android.com/reference/android/app/Activity.html#RESULT_OK)

[RESULT_FIRST_USER](#) (https://developer.android.com/reference/android/app/Activity.html#RESULT_FIRST_USER)

[setResult\(int\)](#) ([https://developer.android.com/reference/android/app/Activity.html#setResult\(int\)](https://developer.android.com/reference/android/app/Activity.html#setResult(int)))

setResult

Added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
void setResult (int resultCode)
```

Call this to set the result that your activity will return to its caller.

Parameters	
resultCode	int : The result code to propagate back to the originating activity, often <code>RESULT_CANCELED</code> or <code>RESULT_OK</code>

See also:

[RESULT_CANCELED](#) (https://developer.android.com/reference/android/app/Activity.html#RESULT_CANCELED)

[RESULT_OK](#) (https://developer.android.com/reference/android/app/Activity.html#RESULT_OK)

[RESULT_FIRST_USER](#) (https://developer.android.com/reference/android/app/Activity.html#RESULT_FIRST_USER)

[setResult\(int, Intent\)](#) ([https://developer.android.com/reference/android/app/Activity.html#setResult\(int, android.content.Intent\)](https://developer.android.com/reference/android/app/Activity.html#setResult(int, android.content.Intent)))

setSecondaryProgress

Added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
void setSecondaryProgress (int secondaryProgress)
```

This method was deprecated in API level 24.

No longer supported. Detection is API 24.

This site uses cookies to store your preferences for site-specific language options.

OK

primary progress (set via `setProgress(int)` ([https://developer.android.com/reference/android/app/Activity.html#setProgress\(int\)](https://developer.android.com/reference/android/app/Activity.html#setProgress(int))) and the background. It can be ideal for media scenarios such as showing the buffering progress while the default progress shows the play progress.

In order for the progress bar to be shown, the feature must be requested via `requestWindowFeature(int)` ([https://developer.android.com/reference/android/app/Activity.html#requestWindowFeature\(int\)](https://developer.android.com/reference/android/app/Activity.html#requestWindowFeature(int))).

Parameters	
<code>secondaryProgress</code>	<code>int</code> : The secondary progress for the progress bar. Valid ranges are from 0 to 10000 (both inclusive).

setTaskDescription

Added in API level 21 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
void setTaskDescription (ActivityManager.TaskDescription (https://developer.android.com/
```

Sets information describing the task with this activity for presentation inside the Recents System UI. When `getRecentTasks(int, int)` ([https://developer.android.com/reference/android/app/ActivityManager.html#getRecentTasks\(int, int\)](https://developer.android.com/reference/android/app/ActivityManager.html#getRecentTasks(int, int))) is called, the activities of each task are traversed in order from the topmost activity to the bottommost. The traversal continues for each property until a suitable value is found. For each task the taskDescription will be returned in `ActivityManager.TaskDescription` (<https://developer.android.com/reference/android/app/ActivityManager.TaskDescription.html>).

Parameters	
<code>taskDescription</code>	<code>ActivityManager.TaskDescription</code> : The TaskDescription properties that describe the task with this activity

See also:

`getRecentTasks(int, int)` ([https://developer.android.com/reference/android/app/ActivityManager.html#getRecentTasks\(int, int\)](https://developer.android.com/reference/android/app/ActivityManager.html#getRecentTasks(int, int)))

`ActivityManager.TaskDescription` (<https://developer.android.com/reference/android/app/ActivityManager.TaskDescription.html>)

This site uses cookies to store your preferences for site-specific language options. OK

setTheme

Added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
void setTheme (int resid)
```

Set the base theme for this context. Note that this should be called before any views are instantiated in the Context (for example before calling `setContentView(View)`

([https://developer.android.com/reference/android/app/Activity.html#setContentView\(android.view.View\)](https://developer.android.com/reference/android/app/Activity.html#setContentView(android.view.View))) or `inflate(int, ViewGroup)` ([https://developer.android.com/reference/android/view/LayoutInflater.html#inflate\(int, android.view.ViewGroup\)](https://developer.android.com/reference/android/view/LayoutInflater.html#inflate(int, android.view.ViewGroup)))).

Parameters

<code>resid</code>	<code>int</code> : The style resource describing the theme.
--------------------	---

setTitle

Added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
void setTitle (CharSequence (https://developer.android.com/reference/java/lang/CharSequence.htm
```

Change the title associated with this activity. If this is a top-level activity, the title for its window will change. If it is an embedded activity, the parent can do whatever it wants with it.

Parameters

<code>title</code>	<code>CharSequence</code>
--------------------	---------------------------

setTitle

Added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
void setTitle (int titleId)
```

Change the title associated with this activity. If this is a top-level activity, the title for its window will change. If it is an embedded activity, the parent can do whatever it wants with it.

Parameters

<code>titleId</code>	<code>int</code>
----------------------	------------------

This site uses cookies to store your preferences for site-specific language options. OK

setTitleColor

Added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
void setTitleColor (int textColor)
```

This method was deprecated in API level 21.

Use action bar styles instead.

Change the color of the title associated with this activity.

This method is deprecated starting in API Level 11 and replaced by action bar styles. For information on styling the Action Bar, read the Action Bar (<https://developer.android.com/guide/topics/ui/actionbar.html>) developer guide.

Parameters

textColor	int
-----------	-----

setVisible

Added in API level 3 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
void setVisible (boolean visible)
```

Control whether this activity's main window is visible. This is intended only for the special case of an activity that is not going to show a UI itself, but can't just finish prior to `onResume()` because it needs to wait for a service binding or such. Setting this to false allows you to prevent your UI from being shown during that time.

The default value for this is taken from the `windowNoDisplay` (<https://developer.android.com/reference/android/R.attr.html#windowNoDisplay>) attribute of the activity's theme.

Parameters

visible	boolean
---------	---------

setVolumeControlStream

Added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
void setVolumeControlStream (int streamType)
```

This site uses cookies to store your preferences for site-specific language options.

received while the Activity is in the foreground will affect this stream.

It is not guaranteed that the hardware volume controls will always change this stream's volume (for example, if a call is in progress, its stream's volume may be changed instead). To reset back to the default, use `USE_DEFAULT_STREAM_TYPE` (https://developer.android.com/reference/android/media/AudioManager.html#USE_DEFAULT_STREAM_TYPE).

Parameters	
<code>streamType</code>	<code>int</code> : The type of the audio stream whose volume should be changed by the hardware volume controls.

setVrModeEnabled

Added in API level 24 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
void setVrModeEnabled (boolean enabled,
                      ComponentName (https://developer.android.com/reference/android/content/ComponentName))
```

Enable or disable virtual reality (VR) mode for this Activity.

VR mode is a hint to Android system to switch to a mode optimized for VR applications while this Activity has user focus.

It is recommended that applications additionally declare `enableVrMode` (<https://developer.android.com/reference/android/R.attr.html#enableVrMode>) in their manifest to allow for smooth activity transitions when switching between VR activities.

If the requested `VrListenerService` (<https://developer.android.com/reference/android/service/vr/VrListenerService.html>) component is not available, VR mode will not be started. Developers can handle this case as follows:

```
String servicePackage = "com.whatever.app";
String serviceClass = "com.whatever.app.MyVrListenerService";

// Name of the component of the VrListenerService to start.
ComponentName serviceComponent = new ComponentName(servicePackage, serviceClass);

try {
    setVrModeEnabled(true, myComponentName);
} catch (PackageManager.NameNotFoundException e) {
    List<ApplicationInfo> installed = getPackageManager().getInstalledApplications();
    boolean isInstalled = false;
```

This site uses cookies to store your preferences for site-specific language options.

OK

```
        isInstalled = true;
        break;
    }
}
if (isInstalled) {
    // Package is installed, but not enabled in Settings. Let user enable.
    startActivity(new Intent(Settings.ACTION_VR_LISTENER_SETTINGS));
} else {
    // Package is not installed. Send an intent to download this.
    sentIntentToLaunchAppStore(servicePackage);
}
}
```

Parameters

enabled

boolean: true to enable this mode.

requestedComponent

ComponentName: the name of the component to use as a `VrListenerService` (<https://developer.android.com/reference/android/service/vr/VrListenerService.html>) while VR mode is enabled.

Throws

`PackageManager.NameNotFoundException` (<https://developer.android.com/reference/android/content/pm/PackageManager.NameNotFoundException.html>)

if the given component to run as a `VrListenerService` (<https://developer.android.com/reference/android/service/vr/VrListenerService.html>) is not installed, or has not been enabled in user settings.

See also:

`FEATURE_VR_MODE` (https://developer.android.com/reference/android/content/pm/PackageManager.html#FEATURE_VR_MODE)

`FEATURE_VR_MODE_HIGH_PERFORMANCE` (https://developer.android.com/reference/android/content/pm/PackageManager.html#FEATURE_VR_MODE_HIGH_PERFORMANCE)

`VrListenerService` (<https://developer.android.com/reference/android/service/vr/VrListenerService.html>)

`ACTION_VR_LISTENER_SETTINGS` (https://developer.android.com/reference/android/provider/Settings.html#ACTION_VR_LISTENER_SETTINGS)

This site uses cookies to store your preferences for site-specific language options.

OK

shouldShowRequestPermissionRationale

Added in API level 23 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

`boolean shouldShowRequestPermissionRationale (String` (<https://developer.android.com/refer>

Gets whether you should show UI with rationale for requesting a permission. You should do this only if you do not have the permission and the context in which the permission is requested does not clearly communicate to the user what would be the benefit from granting this permission.

For example, if you write a camera app, requesting the camera permission would be expected by the user and no rationale for why it is requested is needed. If however, the app needs location for tagging photos then a non-tech savvy user may wonder how location is related to taking photos. In this case you may choose to show UI with rationale of requesting this permission.

Parameters	
<code>permission</code>	<code>String</code> : A permission your app wants to request.
Returns	
<code>boolean</code>	Whether you can show permission rationale UI.

See also:

`checkSelfPermission(String)` ([https://developer.android.com/reference/android/content/ContextWrapper.html#checkSelfPermission\(java.lang.String\)](https://developer.android.com/reference/android/content/ContextWrapper.html#checkSelfPermission(java.lang.String)))

`requestPermissions(String[], int)` ([https://developer.android.com/reference/android/app/Activity.html#requestPermissions\(java.lang.String\[\], int\)](https://developer.android.com/reference/android/app/Activity.html#requestPermissions(java.lang.String[], int)))

`onRequestPermissionsResult(int, String[], int[])` ([https://developer.android.com/reference/android/app/Activity.html#onRequestPermissionsResult\(int, java.lang.String\[\], int\[\]\)](https://developer.android.com/reference/android/app/Activity.html#onRequestPermissionsResult(int, java.lang.String[], int[])))

shouldUpRecreateTask

Added in API level 16 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

`boolean shouldUpRecreateTask (Intent` (<https://developer.android.com/reference/android/content>

Returns true if the app should recreate the task when navigating 'up' from this activity by using `targetIntent`.

If this method returns false the app can trivially call `navigateUpTo(Intent)`

<https://developer.android.com/reference/android...>

This site uses cookies to store your preferences for site-specific language options.

OK

using `TaskStackBuilder` (<https://developer.android.com/reference/android/app/TaskStackBuilder.html>) or another similar mechanism to perform up navigation.

Parameters	
<code>targetIntent</code>	<code>Intent</code> : An intent representing the target destination for up navigation
Returns	
<code>boolean</code>	true if navigating up should recreate a new task stack, false if the same task should be used for the destination

showAssist

Added in API level 23 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
boolean showAssist (Bundle (https://developer.android.com/reference/android/os/Bundle.html) args)
```

Ask to have the current assistant shown to the user. This only works if the calling activity is the current foreground activity. It is the same as calling `VoiceInteractionService.showSession` ([https://developer.android.com/reference/android/service/voice/VoiceInteractionService.html#showSession\(android.os.Bundle, int\)](https://developer.android.com/reference/android/service/voice/VoiceInteractionService.html#showSession(android.os.Bundle, int))) and requesting all of the possible context. The receiver will always see `SHOW_SOURCE_APPLICATION` (https://developer.android.com/reference/android/service/voice/VoiceInteractionSession.html#SHOW_SOURCE_APPLICATION) set.

Parameters	
<code>args</code>	<code>Bundle</code>
Returns	
<code>boolean</code>	Returns true if the assistant was successfully invoked, else false. For example false will be returned if the caller is not the current top activity.

showDialog

Added in API level 8 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
boolean showDialog (int id,
                    Bundle (https://developer.android.com/reference/android/os/Bundle.html) args)
```

This site uses cookies to store your preferences for site-specific language options.

OK

`/app/DialogFragment.html`) class with `FragmentManager` (<https://developer.android.com/reference/android/app/FragmentManager.html>) instead; this is also available on older platforms through the Android compatibility package.

Show a dialog managed by this activity. A call to `onCreateDialog(int, Bundle)` ([https://developer.android.com/reference/android/app/Activity.html#onCreateDialog\(int, android.os.Bundle\)](https://developer.android.com/reference/android/app/Activity.html#onCreateDialog(int, android.os.Bundle))) will be made with the same id the first time this is called for a given id. From thereafter, the dialog will be automatically saved and restored. *If you are targeting HONEYCOMB* (https://developer.android.com/reference/android/os/Build.VERSION_CODES.html#HONEYCOMB) *or later, consider instead using a DialogFragment* (<https://developer.android.com/reference/android/app/DialogFragment.html>) *instead*.

Each time a dialog is shown, `onPrepareDialog(int, Dialog, Bundle)` ([https://developer.android.com/reference/android/app/Activity.html#onPrepareDialog\(int, android.app.Dialog, android.os.Bundle\)](https://developer.android.com/reference/android/app/Activity.html#onPrepareDialog(int, android.app.Dialog, android.os.Bundle))) will be made to provide an opportunity to do any timely preparation.

Parameters	
id	int: The id of the managed dialog.
args	Bundle: Arguments to pass through to the dialog. These will be saved and restored for you. Note that if the dialog is already created, <code>onCreateDialog(int, Bundle)</code> (https://developer.android.com/reference/android/app/Activity.html#onCreateDialog(int, android.os.Bundle)) will not be called with the new arguments but <code>onPrepareDialog(int, Dialog, Bundle)</code> (https://developer.android.com/reference/android/app/Activity.html#onPrepareDialog(int, android.app.Dialog, android.os.Bundle)) will be. If you need to rebuild the dialog, call <code>removeDialog(int)</code> (https://developer.android.com/reference/android/app/Activity.html#removeDialog(int)) first.

Returns	
boolean	Returns true if the Dialog was created; false is returned if it is not created because <code>onCreateDialog(int, Bundle)</code> (https://developer.android.com/reference/android/app/Activity.html#onCreateDialog(int, android.os.Bundle)) returns false.

See also:

[Dialog](https://developer.android.com/reference/android/app/Dialog.html) (<https://developer.android.com/reference/android/app/Dialog.html>)

`onPrepareDialog(int, Dialog, Bundle)` ([https://developer.android.com/reference/android/app/Activity.html#onPrepareDialog\(int, android.app.Dialog, android.os.Bundle\)](https://developer.android.com/reference/android/app/Activity.html#onPrepareDialog(int, android.app.Dialog, android.os.Bundle)))

`dismissDialog(int)` ([https://developer.android.com/reference/android/app/Activity.html#dismissDialog\(int\)](https://developer.android.com/reference/android/app/Activity.html#dismissDialog(int)))

`removeDialog(int)` ([https://developer.android.com/reference/android/app/Activity.html#removeDialog\(int\)](https://developer.android.com/reference/android/app/Activity.html#removeDialog(int)))

showDialog

Added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
void showDialog (int id)
```

This method was deprecated in API level 13.

Use the new `DialogFragment` (<https://developer.android.com/reference/android/app/DialogFragment.html>) class with `FragmentManager` (<https://developer.android.com/reference/android/app/FragmentManager.html>) instead; this is also available on older platforms through the Android compatibility package.

Simple version of `showDialog(int, Bundle)` ([https://developer.android.com/reference/android/app/Activity.html#showDialog\(int, android.os.Bundle\)](https://developer.android.com/reference/android/app/Activity.html#showDialog(int, android.os.Bundle))) that does not take any arguments. Simply calls `showDialog(int, Bundle)` ([https://developer.android.com/reference/android/app/Activity.html#showDialog\(int, android.os.Bundle\)](https://developer.android.com/reference/android/app/Activity.html#showDialog(int, android.os.Bundle))) with null arguments.

Parameters

id	int

showLockTaskEscapeMessage

Added in API level 23 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
void showLockTaskEscapeMessage ()
```

Shows the user the system defined message for telling the user how to exit lock task mode. The task containing this activity must be in lock task mode at the time of this call for the message to be displayed.

This site uses cookies to store your preferences for site-specific language options.

OK

startActionMode

Added in API level 23 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

ActionMode (<https://developer.android.com/reference/android/view/ActionMode.html>) startActionMode(
int type)

Start an action mode of the given type.

Parameters	
callback	ActionMode.Callback: Callback that will manage lifecycle events for this action mode
type	int: One of TYPE_PRIMARY (https://developer.android.com/reference/android/view/ActionMode.html#TYPE_PRIMARY) or TYPE_FLOATING (https://developer.android.com/reference/android/view/ActionMode.html#TYPE_FLOATING).
Returns	
ActionMode (https://developer.android.com/reference/android/view/ActionMode.html)	The ActionMode that was started, or null if it was canceled

See also:

ActionMode (<https://developer.android.com/reference/android/view/ActionMode.html>)

startActionMode

Added in API level 11 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

ActionMode (<https://developer.android.com/reference/android/view/ActionMode.html>) startActionMode()

Start an action mode of the default type TYPE_PRIMARY (https://developer.android.com/reference/android/view/ActionMode.html#TYPE_PRIMARY).

Parameters	
callback	ActionMode.Callback: Callback that will manage lifecycle events for this action mode

--

This site uses cookies to store your preferences for site-specific language options.

OK

<div>ActionMode</div> <div>(https://developer.android.com/reference/android/view/ActionMode.html)</div>	The ActionMode that was started, or null if it was canceled
--	---

See also:

ActionMode (<https://developer.android.com/reference/android/view/ActionMode.html>)

startActivities

Added in API level 16 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
void startActivities (Intent[] (https://developer.android.com/reference/android/content/Intent
Bundle (https://developer.android.com/reference/android/os/Bundle.html) option:
```

Launch a new activity. You will not receive any information about when the activity exits. This implementation overrides the base version, providing information about the activity performing the launch. Because of this additional information, the `FLAG_ACTIVITY_NEW_TASK` (https://developer.android.com/reference/android/content/Intent.html#FLAG_ACTIVITY_NEW_TASK) launch flag is not required; if not specified, the new activity will be added to the task of the caller.

This method throws `ActivityNotFoundException` (<https://developer.android.com/reference/android/content/ActivityNotFoundException.html>) if there was no Activity found to run the given Intent.

Parameters	
intents	Intent: The intents to start.
options	Bundle: Additional options for how the Activity should be started. See <code>Context.startActivity(Intent, Bundle)</code> (https://developer.android.com/reference/android/content/Context.html#startActivity(android.content.Intent, android.os.Bundle)) for more details.

Throws	
	android.content.ActivityNotFoundException

See also:

startActivities(Intent[]) ([https://developer.android.com/reference/android/app/Activity.html#startActivities\(android.content.Intent\[\]\)](https://developer.android.com/reference/android/app/Activity.html#startActivities(android.content.Intent[])))

This site uses cookies to store your preferences for site-specific language options.

OK

```
/app/Activity.html#startActivityForResult(android.content.Intent, int))
```

startActivities

Added in API level 11 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
void startActivities (Intent[] (https://developer.android.com/reference/android/content/Intent
```

Same as `startActivities(Intent[], Bundle)` ([https://developer.android.com/reference/android/app/Activity.html#startActivities\(android.content.Intent\[\], android.os.Bundle\)](https://developer.android.com/reference/android/app/Activity.html#startActivities(android.content.Intent[], android.os.Bundle))) with no options specified.

Parameters

<code>intents</code>	<code>Intent</code> : The intents to start.
----------------------	---

Throws

<code>android.content.ActivityNotFoundException</code>
--

See also:

`startActivities(Intent[], Bundle)` ([https://developer.android.com/reference/android/app/Activity.html#startActivities\(android.content.Intent\[\], android.os.Bundle\)](https://developer.android.com/reference/android/app/Activity.html#startActivities(android.content.Intent[], android.os.Bundle)))

`startActivityForResult(Intent, int)` ([https://developer.android.com/reference/android/app/Activity.html#startActivityForResult\(android.content.Intent, int\)](https://developer.android.com/reference/android/app/Activity.html#startActivityForResult(android.content.Intent, int)))

startActivity

Added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
void startActivity (Intent (https://developer.android.com/reference/android/content/Intent.html
```

Same as `startActivity(Intent, Bundle)` ([https://developer.android.com/reference/android/app/Activity.html#startActivity\(android.content.Intent, android.os.Bundle\)](https://developer.android.com/reference/android/app/Activity.html#startActivity(android.content.Intent, android.os.Bundle))) with no options specified.

Parameters

<code>intent</code>	<code>Intent</code> : The intent to start.
---------------------	--

This site uses cookies to store your preferences for site-specific language options.

OK

android.content.ActivityNotFoundException

See also:

`startActivity(Intent, Bundle)` ([https://developer.android.com/reference/android/app/Activity.html#startActivity\(android.content.Intent, android.os.Bundle\)\)](https://developer.android.com/reference/android/app/Activity.html#startActivity(android.content.Intent, android.os.Bundle))))

`startActivityForResult(Intent, int)` ([https://developer.android.com/reference/android/app/Activity.html#startActivityForResult\(android.content.Intent, int\)\)](https://developer.android.com/reference/android/app/Activity.html#startActivityForResult(android.content.Intent, int))))

startActivity

Added in API level 16 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
void startActivity (Intent (https://developer.android.com/reference/android/content/Intent.html  
Bundle (https://developer.android.com/reference/android/os/Bundle.html) option
```

Launch a new activity. You will not receive any information about when the activity exits. This implementation overrides the base version, providing information about the activity performing the launch. Because of this additional information, the `FLAG_ACTIVITY_NEW_TASK` (https://developer.android.com/reference/android/content/Intent.html#FLAG_ACTIVITY_NEW_TASK) launch flag is not required; if not specified, the new activity will be added to the task of the caller.

This method throws `ActivityNotFoundException` (<https://developer.android.com/reference/android/content/ActivityNotFoundException.html>) if there was no Activity found to run the given Intent.

Parameters

<code>intent</code>	<code>Intent</code> : The intent to start.
<code>options</code>	<code>Bundle</code> : Additional options for how the Activity should be started. See <code>Context.startActivity(Intent, Bundle)</code> (https://developer.android.com/reference/android/content/Context.html#startActivity(android.content.Intent, android.os.Bundle)) for more details.

Throws

<code>android.content.ActivityNotFoundException</code>
--

See also:

`startActivity(Intent)` (<https://developer.android.com/reference/android>

This site uses cookies to store your preferences for site-specific language options.

[/app/Activity.html#startActivityForResult\(android.content.Intent, int\)](#)

startActivityForResult

Available in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
void startActivityForResult (Intent (https://developer.android.com/reference/android/content/
    int requestCode)
```

Same as calling `startActivityForResult(Intent, int, Bundle)` ([https://developer.android.com/reference/android/app/Activity.html#startActivityForResult\(android.content.Intent, int, android.os.Bundle\)](https://developer.android.com/reference/android/app/Activity.html#startActivityForResult(android.content.Intent, int, android.os.Bundle))) with no options.

Parameters	
<code>intent</code>	Intent : The intent to start.
<code>requestCode</code>	int : If ≥ 0 , this code will be returned in <code>onActivityResult()</code> when the activity exits.

Throws	
	<code>android.content.ActivityNotFoundException</code>

See also:

`startActivity(Intent)` ([https://developer.android.com/reference/android/app/Activity.html#startActivity\(android.content.Intent\)](https://developer.android.com/reference/android/app/Activity.html#startActivity(android.content.Intent)))

startActivityForResult

Available in API level 16 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
void startActivityForResult (Intent (https://developer.android.com/reference/android/content/
    int requestCode,
    Bundle (https://developer.android.com/reference/android/os/Bundle.html) option:
```

Launch an activity for which you would like a result when it finished. When this activity exits, your `onActivityResult()` method will be called with the given `requestCode`. Using a negative `requestCode` is the same as calling `startActivity(Intent)` ([https://developer.android.com/reference/android/app/Activity.html#startActivity\(android.content.Intent\)](https://developer.android.com/reference/android/app/Activity.html#startActivity(android.content.Intent))) (the activity is not launched as a sub-activity).

This site uses cookies to store your preferences for site-specific language options.

OK

other protocols (such as `ACTION_MAIN` (https://developer.android.com/reference/android/content/Intent.html#ACTION_MAIN) or `ACTION_VIEW` (https://developer.android.com/reference/android/content/Intent.html#ACTION_VIEW)), you may not get the result when you expect. For example, if the activity you are launching uses the singleTask launch mode, it will not run in your task and thus you will immediately receive a cancel result.

As a special case, if you call `startActivityForResult()` with a `requestCode` ≥ 0 during the initial `onCreate(Bundle savedInstanceState)/onResume()` of your activity, then your window will not be displayed until a result is returned back from the started activity. This is to avoid visible flickering when redirecting to another activity.

This method throws `ActivityNotFoundException` (<https://developer.android.com/reference/android/content/ActivityNotFoundException.html>) if there was no Activity found to run the given Intent.

Parameters	
<code>intent</code>	<code>Intent</code> : The intent to start.
<code>requestCode</code>	<code>int</code> : If ≥ 0 , this code will be returned in <code>onActivityResult()</code> when the activity exits.
<code>options</code>	<code>Bundle</code> : Additional options for how the Activity should be started. See <code>Context.startActivity(Intent, Bundle)</code> (https://developer.android.com/reference/android/content/Context.html#startActivity(android.content.Intent, android.os.Bundle)) for more details.

Throws	
	<code>android.content.ActivityNotFoundException</code>

See also:

`startActivity(Intent)` ([https://developer.android.com/reference/android/app/Activity.html#startActivity\(android.content.Intent\)](https://developer.android.com/reference/android/app/Activity.html#startActivity(android.content.Intent)))

startActivityFromChild

Added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
void startActivityFromChild (Activity (https://developer.android.com/reference/android/app/Activity)
                             Intent (https://developer.android.com/reference/android/content/Intent.html) in
                             int requestCode)
```

This site uses cookies to store your preferences for site-specific language options.

OK

([https://developer.android.com/reference/android/app/Activity.html#startActivityFromChild\(android.app.Activity, android.content.Intent, int, android.os.Bundle\)\)](https://developer.android.com/reference/android/app/Activity.html#startActivityFromChild(android.app.Activity, android.content.Intent, int, android.os.Bundle))) with no options.

Parameters	
<code>child</code>	Activity: The activity making the call.
<code>intent</code>	Intent: The intent to start.
<code>requestCode</code>	int: Reply request code. < 0 if reply is not requested.

Throws	
	<code>android.content.ActivityNotFoundException</code>

See also:

`startActivity(Intent)` ([https://developer.android.com/reference/android/app/Activity.html#startActivity\(android.content.Intent\)](https://developer.android.com/reference/android/app/Activity.html#startActivity(android.content.Intent)))

`startActivityForResult(Intent, int)` ([https://developer.android.com/reference/android/app/Activity.html#startActivityForResult\(android.content.Intent, int\)](https://developer.android.com/reference/android/app/Activity.html#startActivityForResult(android.content.Intent, int)))

startActivityFromChild

Added in API level 16 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
void startActivityFromChild (Activity (https://developer.android.com/reference/android/app/Activity)
                             Intent (https://developer.android.com/reference/android/content/Intent.html) in
                             int requestCode,
                             Bundle (https://developer.android.com/reference/android/os/Bundle.html) option:
```

This is called when a child activity of this one calls its `startActivity(Intent)`

([https://developer.android.com/reference/android/app/Activity.html#startActivity\(android.content.Intent\)](https://developer.android.com/reference/android/app/Activity.html#startActivity(android.content.Intent))) or `startActivityForResult(Intent, int)` ([https://developer.android.com/reference/android/app/Activity.html#startActivityForResult\(android.content.Intent, int\)](https://developer.android.com/reference/android/app/Activity.html#startActivityForResult(android.content.Intent, int))) method.

This method throws `ActivityNotFoundException` (<https://developer.android.com/reference/android/content/ActivityNotFoundException.html>) if there was no Activity found to run the given Intent.

Parameters

This site uses cookies to store your preferences for site-specific language options.

OK

<code>intent</code>	Intent : The intent to start.
<code>requestCode</code>	int : Reply request code. < 0 if reply is not requested.
<code>options</code>	Bundle : Additional options for how the Activity should be started. See <code>Context.startActivity(Intent, Bundle)</code> (https://developer.android.com/reference/android/content/Context.html#startActivity(android.content.Intent, android.os.Bundle)) for more details.

Throws

`android.content.ActivityNotFoundException`

See also:

`startActivity(Intent)` ([https://developer.android.com/reference/android/app/Activity.html#startActivity\(android.content.Intent\)](https://developer.android.com/reference/android/app/Activity.html#startActivity(android.content.Intent)))

`startActivityForResult(Intent, int)` ([https://developer.android.com/reference/android/app/Activity.html#startActivityForResult\(android.content.Intent, int\)](https://developer.android.com/reference/android/app/Activity.html#startActivityForResult(android.content.Intent, int)))

startActivityFromFragment

Added in API level 15 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
void startActivityFromFragment (Fragment (https://developer.android.com/reference/android/app/Fragment.html)
    Intent (https://developer.android.com/reference/android/content/Intent.html) in
    int requestCode,
    Bundle (https://developer.android.com/reference/android/os/Bundle.html) option)
```

This is called when a Fragment in this activity calls its `startActivity(Intent)`

([https://developer.android.com/reference/android/app/Fragment.html#startActivity\(android.content.Intent\)](https://developer.android.com/reference/android/app/Fragment.html#startActivity(android.content.Intent))) or `startActivityForResult(Intent, int)` ([https://developer.android.com/reference/android/app/Fragment.html#startActivityForResult\(android.content.Intent, int\)](https://developer.android.com/reference/android/app/Fragment.html#startActivityForResult(android.content.Intent, int))) method.

This method throws `ActivityNotFoundException` (<https://developer.android.com/reference/android/content/ActivityNotFoundException.html>) if there was no Activity found to run the given Intent.

Parameters

<code>fragment</code>	Fragment : The fragment making the call.
-----------------------	---

This site uses cookies to store your preferences for site-specific language options.

OK

options

Bundle: Additional options for how the Activity should be started. See `Context.startActivity(Intent, Bundle)` ([https://developer.android.com/reference/android/content/Context.html#startActivity\(android.content.Intent, android.os.Bundle\)](https://developer.android.com/reference/android/content/Context.html#startActivity(android.content.Intent, android.os.Bundle))) for more details.

Throws

`android.content.ActivityNotFoundException`

See also:

`startActivity(Intent)` ([https://developer.android.com/reference/android/app/Fragment.html#startActivity\(android.content.Intent\)](https://developer.android.com/reference/android/app/Fragment.html#startActivity(android.content.Intent)))

`startActivityForResult(Intent, int)` ([https://developer.android.com/reference/android/app/Fragment.html#startActivityForResult\(android.content.Intent, int\)](https://developer.android.com/reference/android/app/Fragment.html#startActivityForResult(android.content.Intent, int)))

startActivityFromFragment

Added in API level 11 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
void startActivityFromFragment (Fragment (https://developer.android.com/reference/android/app/Fragment.html) Intent (https://developer.android.com/reference/android/content/Intent.html) in int requestCode)
```

Same as calling `startActivityFromFragment(Fragment, Intent, int, Bundle)`

([https://developer.android.com/reference/android/app/Activity.html#startActivityFromFragment\(android.app.Fragment, android.content.Intent, int, android.os.Bundle\)](https://developer.android.com/reference/android/app/Activity.html#startActivityFromFragment(android.app.Fragment, android.content.Intent, int, android.os.Bundle))) with no options.

Parameters

fragment	Fragment: The fragment making the call.
intent	Intent: The intent to start.
requestCode	int: Reply request code. < 0 if reply is not requested.

Throws

`android.content.ActivityNotFoundException`

See also:

This site uses cookies to store your preferences for site-specific language options.

OK

```

/app/Fragment.html#startActivity(android.content.Intent))

startActivityForResult(Intent, int) (https://developer.android.com/reference/android
/app/Fragment.html#startActivityForResult(android.content.Intent, int))

```

startActivityIfNeeded

Added in API level 16 (https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

```

boolean startActivityIfNeeded (Intent (https://developer.android.com/reference/android/content/Intent),
                                int requestCode,
                                Bundle (https://developer.android.com/reference/android/os/Bundle.html) options)

```

A special variation to launch an activity only if a new activity instance is needed to handle the given Intent. In other words, this is just like `startActivityForResult(Intent, int)`

(https://developer.android.com/reference/android/app/Activity.html#startActivityForResult(android.content.Intent, int)) except: if you are using the `FLAG_ACTIVITY_SINGLE_TOP` (https://developer.android.com/reference/android/content/Intent.html#FLAG_ACTIVITY_SINGLE_TOP) flag, or `singleTask` or `singleTop` `launchMode` (https://developer.android.com/reference/android/R.styleable.html#AndroidManifestActivity_launchMode), and the activity that handles *intent* is the same as your currently running activity, then a new instance is not needed. In this case, instead of the normal behavior of calling `onNewIntent(Intent)` (https://developer.android.com/reference/android/app/Activity.html#onNewIntent(android.content.Intent)) this function will return and you can handle the Intent yourself.

This function can only be called from a top-level activity; if it is called from a child activity, a runtime exception will be thrown.

Parameters	
<code>intent</code>	Intent : The intent to start.
<code>requestCode</code>	int : If ≥ 0 , this code will be returned in <code>onActivityResult()</code> when the activity exits, as described in <code>startActivityForResult(Intent, int)</code> (https://developer.android.com/reference/android/app/Activity.html#startActivityForResult(android.content.Intent, int)).
<code>options</code>	Bundle : Additional options for how the Activity should be started. See <code>Context.startActivity(Intent, Bundle)</code> (https://developer.android.com/reference/android/content/Context.html#startActivity(android.content.Intent, Bundle)).

This site uses cookies to store your preferences for site-specific language options.

OK

Returns	
<code>boolean</code>	If a new activity was launched then true is returned; otherwise false is returned and you must handle the Intent yourself.

See also:

`startActivity(Intent)` ([https://developer.android.com/reference/android/app/Activity.html#startActivity\(android.content.Intent\)](https://developer.android.com/reference/android/app/Activity.html#startActivity(android.content.Intent)))

`startActivityForResult(Intent, int)` ([https://developer.android.com/reference/android/app/Activity.html#startActivityForResult\(android.content.Intent, int\)](https://developer.android.com/reference/android/app/Activity.html#startActivityForResult(android.content.Intent, int)))

startActivityIfNeeded

Android API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
boolean startActivityIfNeeded (Intent intent, int requestCode)
```

Same as calling `startActivityIfNeeded(Intent, int, Bundle)` ([https://developer.android.com/reference/android/app/Activity.html#startActivityIfNeeded\(android.content.Intent, int, android.os.Bundle\)](https://developer.android.com/reference/android/app/Activity.html#startActivityIfNeeded(android.content.Intent, int, android.os.Bundle))) with no options.

Parameters	
<code>intent</code>	Intent: The intent to start.
<code>requestCode</code>	int: If ≥ 0 , this code will be returned in <code>onActivityResult()</code> when the activity exits, as described in <code>startActivityForResult(Intent, int)</code> (https://developer.android.com/reference/android/app/Activity.html#startActivityForResult(android.content.Intent, int)).

Returns	
<code>boolean</code>	If a new activity was launched then true is returned; otherwise false is returned and you must handle the Intent yourself.

See also:

`startActivity(Intent)` ([https://developer.android.com/reference/android/app/Activity.html#startActivity\(android.content.Intent\)](https://developer.android.com/reference/android/app/Activity.html#startActivity(android.content.Intent)))

This site uses cookies to store your preferences for site-specific language options.

OK

startIntentSender

Added in API level 5(<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
void startIntentSender (IntentSender (https://developer.android.com/reference/android/content/IntentSender) fi.  
    Intent (https://developer.android.com/reference/android/content/Intent.html) fi.  
    int flagsMask,  
    int flagsValues,  
    int extraFlags)
```

Same as calling `startIntentSender(IntentSender, Intent, int, int, int, Bundle)` ([https://developer.android.com/reference/android/app/Activity.html#startIntentSender\(android.content.IntentSender, android.content.Intent, int, int, int, android.os.Bundle\)](https://developer.android.com/reference/android/app/Activity.html#startIntentSender(android.content.IntentSender, android.content.Intent, int, int, int, android.os.Bundle))) with no options.

Parameters	
<code>intent</code>	<code>IntentSender</code> : The <code>IntentSender</code> to launch.
<code>fillInIntent</code>	<code>Intent</code> : If non-null, this will be provided as the intent parameter to <code>sendIntent(Context, int, Intent, IntentSender.OnFinished, Handler)</code> (https://developer.android.com/reference/android/content/IntentSender.html#sendIntent(android.content.Context, int, android.content.Intent, android.content.IntentSender.OnFinished, android.os.Handler)).
<code>flagsMask</code>	<code>int</code> : Intent flags in the original <code>IntentSender</code> that you would like to change.
<code>flagsValues</code>	<code>int</code> : Desired values for any bits set in <i>flagsMask</i>
<code>extraFlags</code>	<code>int</code> : Always set to 0.

Throws	
<code>IntentSender.SendIntentException</code> (https://developer.android.com/reference/android/content/IntentSender.SendIntentException.html)	

startIntentSender

Added in API level 16(<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
void startIntentSender (IntentSender (https://developer.android.com/reference/android/content/IntentSender) fi.  
    Intent (https://developer.android.com/reference/android/content/Intent.html) fi.
```

This site uses cookies to store your preferences for site-specific language options.

OK

`Bundle` (<https://developer.android.com/reference/android/os/Bundle.html>) `option`.

Like `startActivity(Intent, Bundle)` ([https://developer.android.com/reference/android/app/Activity.html#startActivity\(android.content.Intent, android.os.Bundle\)](https://developer.android.com/reference/android/app/Activity.html#startActivity(android.content.Intent, android.os.Bundle))), but taking a `IntentSender` to start; see `startIntentSenderForResult(IntentSender, int, Intent, int, int, int, Bundle)` ([https://developer.android.com/reference/android/app/Activity.html#startIntentSenderForResult\(android.content.IntentSender, int, android.content.Intent, int, int, int, android.os.Bundle\)](https://developer.android.com/reference/android/app/Activity.html#startIntentSenderForResult(android.content.IntentSender, int, android.content.Intent, int, int, int, android.os.Bundle))) for more information.

Parameters	
<code>intent</code>	<code>IntentSender</code> : The <code>IntentSender</code> to launch.
<code>fillInIntent</code>	<code>Intent</code> : If non-null, this will be provided as the intent parameter to <code>sendIntent(Context, int, Intent, IntentSender.OnFinished, Handler)</code> (https://developer.android.com/reference/android/content/IntentSender.html#sendIntent(android.content.Context, int, android.content.Intent, android.content.IntentSender.OnFinished, android.os.Handler)).
<code>flagsMask</code>	<code>int</code> : Intent flags in the original <code>IntentSender</code> that you would like to change.
<code>flagsValues</code>	<code>int</code> : Desired values for any bits set in <i>flagsMask</i>
<code>extraFlags</code>	<code>int</code> : Always set to 0.
<code>options</code>	<code>Bundle</code> : Additional options for how the Activity should be started. See <code>Context.startActivity(Intent, Bundle)</code> (https://developer.android.com/reference/android/content/Context.html#startActivity(android.content.Intent, android.os.Bundle)) for more details. If options have also been supplied by the <code>IntentSender</code> , options given here will override any that conflict with those given by the <code>IntentSender</code> .

Throws	
<code>IntentSender.SendIntentException</code> (https://developer.android.com/reference/android/content/IntentSender.SendIntentException.html)	

```
void startIntentSenderForResult (IntentSender (https://developer.android.com/reference/and
    int requestCode,
    Intent (https://developer.android.com/reference/android/content/Intent.html) fi
    int flagsMask,
    int flagsValues,
    int extraFlags)
```

Same as calling `startIntentSenderForResult(IntentSender, int, Intent, int, int, int, Bundle)` (https://developer.android.com/reference/android/app/Activity.html#startIntentSenderForResult(android.content.IntentSender, int, android.content.Intent, int, int, int, android.os.Bundle)) with no options.

Parameters	
<code>intent</code>	<code>IntentSender</code> : The IntentSender to launch.
<code>requestCode</code>	<code>int</code> : If ≥ 0 , this code will be returned in <code>onActivityResult()</code> when the activity exits.
<code>fillInIntent</code>	<code>Intent</code> : If non-null, this will be provided as the intent parameter to <code>sendIntent(Context, int, Intent, IntentSender.OnFinished, Handler)</code> (https://developer.android.com/reference/android/content/IntentSender.html#sendIntent(android.content.Context, int, android.content.Intent, android.content.IntentSender.OnFinished, android.os.Handler)).
<code>flagsMask</code>	<code>int</code> : Intent flags in the original IntentSender that you would like to change.
<code>flagsValues</code>	<code>int</code> : Desired values for any bits set in <i>flagsMask</i>
<code>extraFlags</code>	<code>int</code> : Always set to 0.

Throws	
<code>IntentSender.SendIntentException</code> (https://developer.android.com/reference/android/content/IntentSender.SendIntentException.html)	

startIntentSenderForResult

Started in API level 26 (https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

This site uses cookies to store your preferences for site-specific language options.

OK

```
int flagsMask,
int flagsValues,
int extraFlags,
Bundle (https://developer.android.com/reference/android/os/Bundle.html) option.
```

Like `startActivityForResult(Intent, int)` ([https://developer.android.com/reference/android/app/Activity.html#startActivityForResult\(android.content.Intent, int\)](https://developer.android.com/reference/android/app/Activity.html#startActivityForResult(android.content.Intent, int))), but allowing you to use a `IntentSender` to describe the activity to be started. If the `IntentSender` is for an activity, that activity will be started as if you had called the regular `startActivityForResult(Intent, int)` ([https://developer.android.com/reference/android/app/Activity.html#startActivityForResult\(android.content.Intent, int\)](https://developer.android.com/reference/android/app/Activity.html#startActivityForResult(android.content.Intent, int))) here; otherwise, its associated action will be executed (such as sending a broadcast) as if you had called `IntentSender.sendIntent` ([https://developer.android.com/reference/android/content/IntentSender.html#sendIntent\(android.content.Context, int, android.content.Intent, android.content.IntentSender.OnFinished, android.os.Handler\)](https://developer.android.com/reference/android/content/IntentSender.html#sendIntent(android.content.Context, int, android.content.Intent, android.content.IntentSender.OnFinished, android.os.Handler))) on it.

Parameters	
<code>intent</code>	<code>IntentSender</code> : The <code>IntentSender</code> to launch.
<code>requestCode</code>	<code>int</code> : If ≥ 0 , this code will be returned in <code>onActivityResult()</code> when the activity exits.
<code>fillInIntent</code>	<code>Intent</code> : If non-null, this will be provided as the intent parameter to <code>sendIntent(Context, int, Intent, IntentSender.OnFinished, Handler)</code> (https://developer.android.com/reference/android/content/IntentSender.html#sendIntent(android.content.Context, int, android.content.Intent, android.content.IntentSender.OnFinished, android.os.Handler)).
<code>flagsMask</code>	<code>int</code> : Intent flags in the original <code>IntentSender</code> that you would like to change.
<code>flagsValues</code>	<code>int</code> : Desired values for any bits set in <i>flagsMask</i>
<code>extraFlags</code>	<code>int</code> : Always set to 0.
<code>options</code>	<code>Bundle</code> : Additional options for how the Activity should be started. See <code>Context.startActivity(Intent, Bundle)</code> (https://developer.android.com/reference/android/content/Context.html#startActivity(android.content.Intent, android.os.Bundle)) for more details. If options have also been supplied by the <code>IntentSender</code> , options given here will override any that conflict with those given by the <code>IntentSender</code> .

This site uses cookies to store your preferences for site-specific language options.

IntentSender.SendIntentException

(<https://developer.android.com/reference/android/content/IntentSender.SendIntentException.html>)

startIntentSenderFromChild

Added in API level 16 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
void startIntentSenderFromChild (Activity (https://developer.android.com/reference/android/app/Activity.html);
    IntentSender (https://developer.android.com/reference/android/content/IntentSender.html);
    int requestCode,
    Intent (https://developer.android.com/reference/android/content/Intent.html) fillInIntent,
    int flagsMask,
    int flagsValues,
    int extraFlags,
    Bundle (https://developer.android.com/reference/android/os/Bundle.html) options);
```

Like `startActivityFromChild(Activity, Intent, int)` ([https://developer.android.com/reference/android/app/Activity.html#startActivityFromChild\(android.app.Activity, android.content.Intent, int\)](https://developer.android.com/reference/android/app/Activity.html#startActivityFromChild(android.app.Activity, android.content.Intent, int))), but taking a `IntentSender`; see `startIntentSenderForResult(IntentSender, int, Intent, int, int, int)` ([https://developer.android.com/reference/android/app/Activity.html#startIntentSenderForResult\(android.content.IntentSender, int, android.content.Intent, int, int, int\)](https://developer.android.com/reference/android/app/Activity.html#startIntentSenderForResult(android.content.IntentSender, int, android.content.Intent, int, int, int))) for more information.

Parameters	
child	Activity
intent	IntentSender
requestCode	int
fillInIntent	Intent
flagsMask	int
flagsValues	int
extraFlags	int
options	Bundle

IntentSender.SendIntentException

([https://developer.android.com/reference](https://developer.android.com/reference/android/content/IntentSender.SendIntentException.html)
[/android/content](https://developer.android.com/reference/android/content/IntentSender.SendIntentException.html)
[/IntentSender.SendIntentException.html](https://developer.android.com/reference/android/content/IntentSender.SendIntentException.html))

startIntentSenderFromChild

Added in API level 5 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
void startIntentSenderFromChild (Activity (https://developer.android.com/reference/android/  
    IntentSender (https://developer.android.com/reference/android/content/IntentSer  
    int requestCode,  
    Intent (https://developer.android.com/reference/android/content/Intent.html) fi  
    int flagsMask,  
    int flagsValues,  
    int extraFlags)
```

Same as calling `startIntentSenderFromChild(Activity, IntentSender, int, Intent, int, int, int, Bundle)` ([https://developer.android.com/reference/android/app/Activity.html#startIntentSenderFromChild\(android.app.Activity, android.content.IntentSender, int, android.content.Intent, int, int, int, android.os.Bundle\)](https://developer.android.com/reference/android/app/Activity.html#startIntentSenderFromChild(android.app.Activity, android.content.IntentSender, int, android.content.Intent, int, int, int, android.os.Bundle))) with no options.

Parameters	
child	Activity
intent	IntentSender
requestCode	int
fillInIntent	Intent
flagsMask	int
flagsValues	int
extraFlags	int

Throws	
IntentSender.SendIntentException	
(https://developer.android.com/reference /android/content /IntentSender.SendIntentException.html)	

startLocalVoiceInteraction

Added in API level 24 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
void startLocalVoiceInteraction (Bundle (https://developer.android.com/reference/android/os/
```

Starts a local voice interaction session. When ready, [onLocalVoiceInteractionStarted\(\)](https://developer.android.com/reference/android/app/Activity.html#onLocalVoiceInteractionStarted()) ([https://developer.android.com/reference/android/app/Activity.html#onLocalVoiceInteractionStarted\(\)](https://developer.android.com/reference/android/app/Activity.html#onLocalVoiceInteractionStarted())) is called. You can pass a bundle of private options to the registered voice interaction service.

Parameters	
<code>privateOptions</code>	Bundle: a Bundle of private arguments to the current voice interaction service

startLockTask

Added in API level 21 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
void startLockTask ()
```

Request to put this Activity in a mode where the user is locked to the current task. This will prevent the user from launching other apps, going to settings, or reaching the home screen. This does not include those apps whose `lockTaskMode` (<https://developer.android.com/reference/android/R.attr.html#lockTaskMode>) values permit launching while locked. If [isLockTaskPermitted\(String\)](https://developer.android.com/reference/android/app/admin/DevicePolicyManager.html#isLockTaskPermitted(java.lang.String)) ([https://developer.android.com/reference/android/app/admin/DevicePolicyManager.html#isLockTaskPermitted\(java.lang.String\)](https://developer.android.com/reference/android/app/admin/DevicePolicyManager.html#isLockTaskPermitted(java.lang.String))) returns true or `lockTaskMode=lockTaskModeAlways` for this component then the app will go directly into Lock Task mode. The user will not be able to exit this mode until [stopLockTask\(\)](https://developer.android.com/reference/android/app/Activity.html#stopLockTask()) ([https://developer.android.com/reference/android/app/Activity.html#stopLockTask\(\)](https://developer.android.com/reference/android/app/Activity.html#stopLockTask())) is called. If [isLockTaskPermitted\(String\)](https://developer.android.com/reference/android/app/admin/DevicePolicyManager.html#isLockTaskPermitted(java.lang.String)) ([https://developer.android.com/reference/android/app/admin/DevicePolicyManager.html#isLockTaskPermitted\(java.lang.String\)](https://developer.android.com/reference/android/app/admin/DevicePolicyManager.html#isLockTaskPermitted(java.lang.String))) returns false then the system will prompt the user with a dialog requesting permission to enter this mode. When entered through this method the user can exit at any time through an action described by the request dialog. Calling `stopLockTask` will also exit the mode.

See also:

`lockTaskMode` (<https://developer.android.com/reference/android/R.attr.html#lockTaskMode>)

This site uses cookies to store your preferences for site-specific language options.

OK

```
void startManagingCursor (Cursor (https://developer.android.com/reference/android/database/Cur
```

This method was deprecated in API level 11.

Use the new [CursorLoader](https://developer.android.com/reference/android/content/CursorLoader.html) (<https://developer.android.com/reference/android/content/CursorLoader.html>) class with [LoaderManager](https://developer.android.com/reference/android/app/LoaderManager.html) (<https://developer.android.com/reference/android/app/LoaderManager.html>) instead; this is also available on older platforms through the Android compatibility package.

This method allows the activity to take care of managing the given [Cursor](https://developer.android.com/reference/android/database/Cursor.html) (<https://developer.android.com/reference/android/database/Cursor.html>)'s lifecycle for you based on the activity's lifecycle. That is, when the activity is stopped it will automatically call [deactivate\(\)](https://developer.android.com/reference/android/database/Cursor.html#deactivate()) ([https://developer.android.com/reference/android/database/Cursor.html#deactivate\(\)](https://developer.android.com/reference/android/database/Cursor.html#deactivate())) on the given [Cursor](https://developer.android.com/reference/android/database/Cursor.html), and when it is later restarted it will call [requery\(\)](https://developer.android.com/reference/android/database/Cursor.html#requery()) ([https://developer.android.com/reference/android/database/Cursor.html#requery\(\)](https://developer.android.com/reference/android/database/Cursor.html#requery())) for you. When the activity is destroyed, all managed [Cursors](https://developer.android.com/reference/android/database/Cursor.html) will be closed automatically. *If you are targeting [HONEYCOMB](https://developer.android.com/reference/android/os/Build.VERSION_CODES.html#HONEYCOMB) (https://developer.android.com/reference/android/os/Build.VERSION_CODES.html#HONEYCOMB) or later, consider instead using [LoaderManager](https://developer.android.com/reference/android/app/LoaderManager.html) (<https://developer.android.com/reference/android/app/LoaderManager.html>) instead, available via [getLoaderManager\(\)](https://developer.android.com/reference/android/app/Activity.html#getLoaderManager()) ([https://developer.android.com/reference/android/app/Activity.html#getLoaderManager\(\)](https://developer.android.com/reference/android/app/Activity.html#getLoaderManager())).*

Warning: Do not call [close\(\)](https://developer.android.com/reference/android/database/Cursor.html#close()) ([https://developer.android.com/reference/android/database/Cursor.html#close\(\)](https://developer.android.com/reference/android/database/Cursor.html#close())) on cursor obtained from [managedQuery\(Uri, String\[\], String, String\[\], String\)](https://developer.android.com/reference/android/app/Activity.html#managedQuery(android.net.Uri, java.lang.String[], java.lang.String, java.lang.String[], java.lang.String)) ([https://developer.android.com/reference/android/app/Activity.html#managedQuery\(android.net.Uri, java.lang.String\[\], java.lang.String, java.lang.String\[\], java.lang.String\)](https://developer.android.com/reference/android/app/Activity.html#managedQuery(android.net.Uri, java.lang.String[], java.lang.String, java.lang.String[], java.lang.String))), because the activity will do that for you at the appropriate time. However, if you call [stopManagingCursor\(Cursor\)](https://developer.android.com/reference/android/app/Activity.html#stopManagingCursor(android.database.Cursor)) ([https://developer.android.com/reference/android/app/Activity.html#stopManagingCursor\(android.database.Cursor\)](https://developer.android.com/reference/android/app/Activity.html#stopManagingCursor(android.database.Cursor))) on a cursor from a managed query, the system *will not* automatically close the cursor and, in that case, you must call [close\(\)](https://developer.android.com/reference/android/database/Cursor.html#close()) ([https://developer.android.com/reference/android/database/Cursor.html#close\(\)](https://developer.android.com/reference/android/database/Cursor.html#close())).

Parameters

c	Cursor : The Cursor to be managed.
---	--

See also:

[managedQuery\(android.net.Uri, String\[\], String, String\[\], String\)](https://developer.android.com/reference/android/app/Activity.html#managedQuery(android.net.Uri, java.lang.String[], java.lang.String, java.lang.String[], java.lang.String)) ([https://developer.android.com/reference/android/app/Activity.html#managedQuery\(android.net.Uri, java.lang.String\[\], java.lang.String, java.lang.String\[\], java.lang.String\)](https://developer.android.com/reference/android/app/Activity.html#managedQuery(android.net.Uri, java.lang.String[], java.lang.String, java.lang.String[], java.lang.String)))

[stopManagingCursor\(Cursor\)](https://developer.android.com/reference/android/app/Activity.html#stopManagingCursor(android.database.Cursor)) ([https://developer.android.com/reference/android/app/Activity.html#stopManagingCursor\(android.database.Cursor\)](https://developer.android.com/reference/android/app/Activity.html#stopManagingCursor(android.database.Cursor)))

This site uses cookies to store your preferences for site-specific language options.

OK

startNextMatchingActivity

```
boolean startNextMatchingActivity (Intent (https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)
                                   Bundle (https://developer.android.com/reference/android/os/Bundle.html) option)
```

Special version of starting an activity, for use when you are replacing other activity components. You can use this to hand the Intent off to the next Activity that can handle it. You typically call this in `onCreate(Bundle)` (https://developer.android.com/reference/android/app/Activity.html#onCreate(android.os.Bundle)) with the Intent returned by `getIntent()` (https://developer.android.com/reference/android/app/Activity.html#getIntent()).

Parameters	
intent	Intent: The intent to dispatch to the next activity. For correct behavior, this must be the same as the Intent that started your own activity; the only changes you can make are to the extras inside of it.
options	Bundle: Additional options for how the Activity should be started. See Context.startActivity(Intent, Bundle) (https://developer.android.com/reference/android/content/Context.html#startActivity(android.content.Intent, android.os.Bundle)) for more details.
Returns	
boolean	Returns a boolean indicating whether there was another Activity to start: true if there was a next activity to start, false if there wasn't. In general, if true is returned you will then want to call finish() on yourself.

startNextMatchingActivity

Same as calling `startNextMatchingActivity(Intent, Bundle)` (https://developer.android.com/reference/android/app/Activity.html#startNextMatchingActivity(android.content.Intent, android.os.Bundle)) with no options.

Parameters	
intent	Intent: The intent to dispatch to the next activity. For correct behavior, this must be

This site uses cookies to store your preferences for site-specific language options.

OK

Returns	
<code>boolean</code>	Returns a boolean indicating whether there was another Activity to start: true if there was a next activity to start, false if there wasn't. In general, if true is returned you will then want to call <code>finish()</code> on yourself.

startPostponedEnterTransition

Added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
void startPostponedEnterTransition ()
```

Begin postponed transitions after `postponeEnterTransition()` ([https://developer.android.com/reference/android/app/Activity.html#postponeEnterTransition\(\)](https://developer.android.com/reference/android/app/Activity.html#postponeEnterTransition())) was called. If

`postponeEnterTransition()` was called, you must call `startPostponedEnterTransition()` to have your Activity start drawing.

startSearch

Added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
void startSearch (String (https://developer.android.com/reference/java/lang/String.html) initi
                boolean selectInitialQuery,
                Bundle (https://developer.android.com/reference/android/os/Bundle.html) appSea
                boolean globalSearch)
```

This hook is called to launch the search UI.

It is typically called from `onSearchRequested()`, either directly from `Activity.onSearchRequested()` or from an overridden version in any given Activity. If your goal is simply to activate search, it is preferred to call `onSearchRequested()`, which may have been overridden elsewhere in your Activity. If your goal is to inject specific data such as context data, it is preferred to *override* `onSearchRequested()`, so that any callers to it will benefit from the override.

Parameters	
<code>initialQuery</code>	String: Any non-null non-empty string will be inserted as pre-entered text in the search query box.
<code>selectInitialQuery</code>	boolean: If true, the initial query will be preselected, which means that any further typing will replace it. This is useful for cases where an

This site uses cookies to store your preferences for site-specific language options.

OK

	inserted query is text that the user entered, and the user would expect to be able to keep typing. <i>This parameter is only meaningful if <code>initialQuery</code> is a non-empty string.</i>
<code>appSearchData</code>	Bundle: An application can insert application-specific context here, in order to improve quality or specificity of its own searches. This data will be returned with SEARCH intent(s). Null if no extra data is required.
<code>globalSearch</code>	boolean: If false, this will only launch the search that has been specifically defined by the application (which is usually defined as a local search). If no default search is defined in the current application or activity, global search will be launched. If true, this will always launch a platform-global (e.g. web-based) search instead.

See also:

`SearchManager` (<https://developer.android.com/reference/android/app/SearchManager.html>)

`onSearchRequested()` ([https://developer.android.com/reference/android/app/Activity.html#onSearchRequested\(\)](https://developer.android.com/reference/android/app/Activity.html#onSearchRequested()))

stopLocalVoiceInteraction

Added in API level 24 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
void stopLocalVoiceInteraction ()
```

Request to terminate the current voice interaction that was previously started using

`startLocalVoiceInteraction(Bundle)` ([https://developer.android.com/reference/android/app/Activity.html#startLocalVoiceInteraction\(android.os.Bundle\)](https://developer.android.com/reference/android/app/Activity.html#startLocalVoiceInteraction(android.os.Bundle))). When the interaction is terminated, `onLocalVoiceInteractionStopped()` ([https://developer.android.com/reference/android/app/Activity.html#onLocalVoiceInteractionStopped\(\)](https://developer.android.com/reference/android/app/Activity.html#onLocalVoiceInteractionStopped())) will be called.

stopLockTask

Added in API level 21 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
void stopLockTask ()
```

This site uses cookies to store your preferences for site-specific language options.

`/app/Activity.html#startLockTask()`). This can only be called by activities that have successfully called `startLockTask` previously. This will allow the user to exit this app and move onto other activities.

Note: This method should only be called when the activity is user-facing. That is, between `onResume()` and `onPause()`.

Note: If there are other tasks below this one that are also locked then calling this method will immediately finish this task and resume the previous locked one, remaining in `lockTask` mode.

See also:

`lockTaskMode` (<https://developer.android.com/reference/android/R.attr.html#lockTaskMode>)

`getLockTaskModeState()` ([https://developer.android.com/reference/android/app/ActivityManager.html#getLockTaskModeState\(\)](https://developer.android.com/reference/android/app/ActivityManager.html#getLockTaskModeState()))

stopManagingCursor

Added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

`void stopManagingCursor (Cursor` (<https://developer.android.com/reference/android/database/Cursor>)

This method was deprecated in API level 11.

Use the new `CursorLoader` (<https://developer.android.com/reference/android/content/CursorLoader.html>) class with `LoaderManager` (<https://developer.android.com/reference/android/app/LoaderManager.html>) instead; this is also available on older platforms through the Android compatibility package.

Given a `Cursor` that was previously given to `startManagingCursor(Cursor)`

([https://developer.android.com/reference/android/app/Activity.html#startManagingCursor\(android.database.Cursor\)](https://developer.android.com/reference/android/app/Activity.html#startManagingCursor(android.database.Cursor))), stop the activity's management of

that cursor.

Warning: After calling this method on a cursor from a managed query, the system *will not* automatically close the cursor and you must call `close()` ([https://developer.android.com/reference/android/database/Cursor.html#close\(\)](https://developer.android.com/reference/android/database/Cursor.html#close())).

Parameters

c	<code>Cursor</code> : The <code>Cursor</code> that was being managed.
---	---

This site uses cookies to store your preferences for site-specific language options.

OK

`startManagingCursor(Cursor)` ([https://developer.android.com/reference/android/app/Activity.html#startManagingCursor\(android.database.Cursor\)](https://developer.android.com/reference/android/app/Activity.html#startManagingCursor(android.database.Cursor)))

takeKeyEvents

Added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

`void takeKeyEvents (boolean get)`

Request that key events come to this activity. Use this if your activity has no views with focus, but the activity still wants a chance to process key events.

Parameters	
<code>get</code>	<code>boolean</code>

See also:

`takeKeyEvents(boolean)` ([https://developer.android.com/reference/android/view/Window.html#takeKeyEvents\(boolean\)](https://developer.android.com/reference/android/view/Window.html#takeKeyEvents(boolean)))

triggerSearch

Added in API level 5 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

`void triggerSearch (String (https://developer.android.com/reference/java/lang/String.html) query, Bundle (https://developer.android.com/reference/android/os/Bundle.html) appSearchData)`

Similar to `startSearch(String, boolean, Bundle, boolean)` ([https://developer.android.com/reference/android/app/Activity.html#startSearch\(java.lang.String, boolean, android.os.Bundle, boolean\)](https://developer.android.com/reference/android/app/Activity.html#startSearch(java.lang.String, boolean, android.os.Bundle, boolean))), but actually fires off the search query after invoking the search dialog. Made available for testing purposes.

Parameters	
<code>query</code>	String: The query to trigger. If empty, the request will be ignored.
<code>appSearchData</code>	Bundle: An application can insert application-specific context here, in order to improve quality or specificity of its own searches. This data will be returned with SEARCH intent(s). Null if no extra data is required.

This site uses cookies to store your preferences for site-specific language options.

unregisterForContextMenu

Added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
void unregisterForContextMenu (View (https://developer.android.com/reference/android/view/Vie
```

Prevents a context menu to be shown for the given view. This method will remove the `View.OnCreateContextMenuListener` (<https://developer.android.com/reference/android/view/View.OnCreateContextMenuListener.html>) on the view.

Parameters

<code>view</code>	View: The view that should stop showing a context menu.
-------------------	--

See also:

`registerForContextMenu(View)` ([https://developer.android.com/reference/android/app/Activity.html#registerForContextMenu\(android.view.View\)](https://developer.android.com/reference/android/app/Activity.html#registerForContextMenu(android.view.View)))

Protected methods

onActivityResult

Added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
void onActivityResult (int requestCode,
                        int resultCode,
                        Intent (https://developer.android.com/reference/android/content/Intent.html) da
```

Called when an activity you launched exits, giving you the `requestCode` you started it with, the `resultCode` it returned, and any additional data from it. The `resultCode` will be `RESULT_CANCELED` (https://developer.android.com/reference/android/app/Activity.html#RESULT_CANCELED) if the activity explicitly returned that, didn't return any result, or crashed during its operation.

You will receive this call immediately before `onResume()` when your activity is re-starting.

This method is never invoked if your activity sets `noHistory` (https://developer.android.com/reference/android/R.styleable.html#AndroidManifestActivity_noHistory) to `true`.

Parameters

<code>requestCode</code>	int: The integer request code originally supplied to <code>startActivityForResult()</code> , allowing you to identify who this result came from.
--------------------------	---

This site uses cookies to store your preferences for site-specific language options.

OK

resultCode	int: The integer result code returned by the child activity through its setResult().
data	Intent: An Intent, which can return result data to the caller (various data can be attached to Intent "extras").

See also:

startActivityForResult(Intent, int) (https://developer.android.com/reference/android/app/Activity.html#startActivityForResult(android.content.Intent, int))

createPendingResult(int, Intent, int) (https://developer.android.com/reference/android/app/Activity.html#createPendingResult(int, android.content.Intent, int))

setResult(int) (https://developer.android.com/reference/android/app/Activity.html#setResult(int))

onApplyThemeResource

Added in API level 1 (https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

```
void onApplyThemeResource (Resources.Theme (https://developer.android.com/reference/android
    int resid,
    boolean first)
```

Called by `setTheme(int)` (https://developer.android.com/reference/android/view/ContextThemeWrapper.html#setTheme(int)) and `getTheme()` (https://developer.android.com/reference/android/view/ContextThemeWrapper.html#getTheme()) to apply a theme resource to the current Theme object. May be overridden to change the default (simple) behavior. This method will not be called in multiple threads simultaneously.

Parameters	
theme	Resources.Theme: the theme being modified
resid	int: the style resource being applied to <i>theme</i>
first	boolean: true if this is the first time a style is being applied to <i>theme</i>

onChildTitleChanged

Added in API level 1 (https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

This site uses cookies to store your preferences for site-specific language options.

OK

Parameters	
<code>childActivity</code>	<code>Activity</code>
<code>title</code>	<code>CharSequence</code>

onCreate

Added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
void onCreate (Bundle (https://developer.android.com/reference/android/os/Bundle.html) savedInstanceState)
```

Called when the activity is starting. This is where most initialization should go: calling `setContentView(int)` ([https://developer.android.com/reference/android/app/Activity.html#setContentView\(int\)](https://developer.android.com/reference/android/app/Activity.html#setContentView(int))) to inflate the activity's UI, using `findViewById(int)` ([https://developer.android.com/reference/android/app/Activity.html#findViewById\(int\)](https://developer.android.com/reference/android/app/Activity.html#findViewById(int))) to programmatically interact with widgets in the UI, calling `managedQuery(android.net.Uri, String[], String, String[], String)` ([https://developer.android.com/reference/android/app/Activity.html#managedQuery\(android.net.Uri, java.lang.String\[\], java.lang.String, java.lang.String\[\], java.lang.String\)](https://developer.android.com/reference/android/app/Activity.html#managedQuery(android.net.Uri, java.lang.String[], java.lang.String, java.lang.String[], java.lang.String))) to retrieve cursors for data being displayed, etc.

You can call `finish()` ([https://developer.android.com/reference/android/app/Activity.html#finish\(\)](https://developer.android.com/reference/android/app/Activity.html#finish())) from within this function, in which case `onDestroy()` will be immediately called without any of the rest of the activity lifecycle (`onStart()` ([https://developer.android.com/reference/android/app/Activity.html#onStart\(\)](https://developer.android.com/reference/android/app/Activity.html#onStart())), `onResume()` ([https://developer.android.com/reference/android/app/Activity.html#onResume\(\)](https://developer.android.com/reference/android/app/Activity.html#onResume())), `onPause()` ([https://developer.android.com/reference/android/app/Activity.html#onPause\(\)](https://developer.android.com/reference/android/app/Activity.html#onPause())), etc) executing.

Derived classes must call through to the super class's implementation of this method. If they do not, an exception will be thrown.

Parameters	
<code>savedInstanceState</code>	Bundle: If the activity is being re-initialized after previously being shut down then this Bundle contains the data it most recently supplied in <code>onSaveInstanceState(Bundle)</code> (https://developer.android.com/reference/android/app/Activity.html#onSaveInstanceState(android.os.Bundle)). Note: <i>Otherwise it is null.</i>

This site uses cookies to store your preferences for site-specific language options. OK

`onSaveInstanceState(Bundle)` ([https://developer.android.com/reference/android/app/Activity.html#onSaveInstanceState\(android.os.Bundle\)\)](https://developer.android.com/reference/android/app/Activity.html#onSaveInstanceState(android.os.Bundle)))

`onRestoreInstanceState(Bundle)` ([https://developer.android.com/reference/android/app/Activity.html#onRestoreInstanceState\(android.os.Bundle\)\)](https://developer.android.com/reference/android/app/Activity.html#onRestoreInstanceState(android.os.Bundle)))

`onPostCreate(Bundle)` ([https://developer.android.com/reference/android/app/Activity.html#onPostCreate\(android.os.Bundle\)\)](https://developer.android.com/reference/android/app/Activity.html#onPostCreate(android.os.Bundle)))

onCreateDialog Added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

`Dialog` (<https://developer.android.com/reference/android/app/Dialog.html>) `onCreateDialog (int id)`

This method was deprecated in API level 8.

Old no-arguments version of `onCreateDialog(int, Bundle)` ([https://developer.android.com/reference/android/app/Activity.html#onCreateDialog\(int, android.os.Bundle\)\)](https://developer.android.com/reference/android/app/Activity.html#onCreateDialog(int, android.os.Bundle))).

Parameters

<code>id</code>	<code>int</code>
-----------------	------------------

Returns

<p><code>Dialog</code></p> <p>(https://developer.android.com/reference/android/app/Dialog.html)</p>	
---	--

onCreateDialog Added in API level 8 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

`Dialog` (<https://developer.android.com/reference/android/app/Dialog.html>) `onCreateDialog (int id, Bundle args)` (<https://developer.android.com/reference/android/os/Bundle.html>)

This method was deprecated in API level 13.

Use the new `DialogFragment` (<https://developer.android.com/reference/android/app/DialogFragment.html>) class with `FragmentManager` (<https://developer.android.com/reference/android/app/FragmentManager.html>)

This site uses cookies to store your preferences for site-specific language options.

Callback for creating dialogs that are managed (saved and restored) for you by the activity. The default implementation calls through to `onCreateDialog(int)` ([https://developer.android.com/reference/android/app/Activity.html#onCreateDialog\(int\)](https://developer.android.com/reference/android/app/Activity.html#onCreateDialog(int))) for compatibility. *If you are targeting HONEYCOMB* (https://developer.android.com/reference/android/os/Build.VERSION_CODES.html#HONEYCOMB) *or later, consider instead using a DialogFragment* (<https://developer.android.com/reference/android/app/DialogFragment.html>) *instead*.

If you use `showDialog(int)` ([https://developer.android.com/reference/android/app/Activity.html#showDialog\(int\)](https://developer.android.com/reference/android/app/Activity.html#showDialog(int))), the activity will call through to this method the first time, and hang onto it thereafter. Any dialog that is created by this method will automatically be saved and restored for you, including whether it is showing.

If you would like the activity to manage saving and restoring dialogs for you, you should override this method and handle any ids that are passed to `showDialog(int)` ([https://developer.android.com/reference/android/app/Activity.html#showDialog\(int\)](https://developer.android.com/reference/android/app/Activity.html#showDialog(int))).

If you would like an opportunity to prepare your dialog before it is shown, override `onPrepareDialog(int, Dialog, Bundle)` ([https://developer.android.com/reference/android/app/Activity.html#onPrepareDialog\(int, android.app.Dialog, android.os.Bundle\)](https://developer.android.com/reference/android/app/Activity.html#onPrepareDialog(int, android.app.Dialog, android.os.Bundle))).

Parameters	
id	int: The id of the dialog.
args	Bundle: The dialog arguments provided to <code>showDialog(int, Bundle)</code> (https://developer.android.com/reference/android/app/Activity.html#showDialog(int, android.os.Bundle)).

Returns	
Dialog (https://developer.android.com/reference/android/app/Dialog.html)	The dialog. If you return null, the dialog will not be created.

See also:

`onPrepareDialog(int, Dialog, Bundle)` ([https://developer.android.com/reference/android/app/Activity.html#onPrepareDialog\(int, android.app.Dialog, android.os.Bundle\)](https://developer.android.com/reference/android/app/Activity.html#onPrepareDialog(int, android.app.Dialog, android.os.Bundle)))

`showDialog(int, Bundle)` ([https://developer.android.com/reference/android/app/Activity.html#showDialog\(int, android.os.Bundle\)](https://developer.android.com/reference/android/app/Activity.html#showDialog(int, android.os.Bundle)))

This site uses cookies to store your preferences for site-specific language options.

OK

```
/app/Activity.html#dismissDialog(int))  
  
removeDialog(int) (https://developer.android.com/reference/android  
/app/Activity.html#removeDialog\(int\))
```

onDestroy

Added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
void onDestroy ()
```

Perform any final cleanup before an activity is destroyed. This can happen either because the activity is finishing (someone called `finish()` ([https://developer.android.com/reference/android/app/Activity.html#finish\(\)](https://developer.android.com/reference/android/app/Activity.html#finish())) on it, or because the system is temporarily destroying this instance of the activity to save space. You can distinguish between these two scenarios with the `isFinishing()` ([https://developer.android.com/reference/android/app/Activity.html#isFinishing\(\)](https://developer.android.com/reference/android/app/Activity.html#isFinishing())) method.

Note: do not count on this method being called as a place for saving data! For example, if an activity is editing data in a content provider, those edits should be committed in either `onPause()`

([https://developer.android.com/reference/android/app/Activity.html#onPause\(\)](https://developer.android.com/reference/android/app/Activity.html#onPause())) or

*`onSaveInstanceState(Bundle)` ([https://developer.android.com/reference/android](https://developer.android.com/reference/android/app/Activity.html#onSaveInstanceState(android.os.Bundle))
[/app/Activity.html#onSaveInstanceState\(android.os.Bundle\)](https://developer.android.com/reference/android/app/Activity.html#onSaveInstanceState(android.os.Bundle))), *not here*. This method is usually*

implemented to free resources like threads that are associated with an activity, so that a destroyed activity does not leave such things around while the rest of its application is still running. There are situations where the system will simply kill the activity's hosting process without calling this method (or any others) in it, so it should not be used to do things that are intended to remain around after the process goes away.

Derived classes must call through to the super class's implementation of this method. If they do not, an exception will be thrown.

See also:

`onPause()` ([https://developer.android.com/reference/android/app/Activity.html#onPause\(\)](https://developer.android.com/reference/android/app/Activity.html#onPause()))

`onStop()` ([https://developer.android.com/reference/android/app/Activity.html#onStop\(\)](https://developer.android.com/reference/android/app/Activity.html#onStop()))

`finish()` ([https://developer.android.com/reference/android/app/Activity.html#finish\(\)](https://developer.android.com/reference/android/app/Activity.html#finish()))

`isFinishing()` ([https://developer.android.com/reference/android/app/Activity.html#isFinishing\(\)](https://developer.android.com/reference/android/app/Activity.html#isFinishing()))

This site uses cookies to store your preferences for site-specific language options.

OK

onNewIntent

Added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
void onNewIntent (Intent (https://developer.android.com/reference/android/content/Intent.html) )
```

This is called for activities that set `launchMode` to "singleTop" in their package, or if a client used the `FLAG_ACTIVITY_SINGLE_TOP` (https://developer.android.com/reference/android/content/Intent.html#FLAG_ACTIVITY_SINGLE_TOP) flag when calling `startActivity(Intent)` ([https://developer.android.com/reference/android/app/Activity.html#startActivity\(android.content.Intent\)](https://developer.android.com/reference/android/app/Activity.html#startActivity(android.content.Intent))). In either case, when the activity is re-launched while at the top of the activity stack instead of a new instance of the activity being started, `onNewIntent()` will be called on the existing instance with the `Intent` that was used to re-launch it.

An activity will always be paused before receiving a new intent, so you can count on `onResume()` ([https://developer.android.com/reference/android/app/Activity.html#onResume\(\)](https://developer.android.com/reference/android/app/Activity.html#onResume())) being called after this method.

Note that `getIntent()` ([https://developer.android.com/reference/android/app/Activity.html#getIntent\(\)](https://developer.android.com/reference/android/app/Activity.html#getIntent())) still returns the original `Intent`. You can use `setIntent(Intent)` ([https://developer.android.com/reference/android/app/Activity.html#setIntent\(android.content.Intent\)](https://developer.android.com/reference/android/app/Activity.html#setIntent(android.content.Intent))) to update it to this new `Intent`.

Parameters	
<code>intent</code>	<code>Intent</code> : The new intent that was started for the activity.

See also:

- `getIntent()` ([https://developer.android.com/reference/android/app/Activity.html#getIntent\(\)](https://developer.android.com/reference/android/app/Activity.html#getIntent()))
- `setIntent(Intent)` ([https://developer.android.com/reference/android/app/Activity.html#setIntent\(android.content.Intent\)](https://developer.android.com/reference/android/app/Activity.html#setIntent(android.content.Intent)))
- `onResume()` ([https://developer.android.com/reference/android/app/Activity.html#onResume\(\)](https://developer.android.com/reference/android/app/Activity.html#onResume()))

onPause

Added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
void onPause ()
```

Called as part of the activity lifecycle when an activity is going into the background, but has not (yet)

This site uses cookies to store your preferences for site-specific language options.

OK

When activity B is launched in front of activity A, this callback will be invoked on A. B will not be created until A's `onPause()` ([https://developer.android.com/reference/android/app/Activity.html#onPause\(\)](https://developer.android.com/reference/android/app/Activity.html#onPause())) returns, so be sure to not do anything lengthy here.

This callback is mostly used for saving any persistent state the activity is editing, to present a "edit in place" model to the user and making sure nothing is lost if there are not enough resources to start the new activity without first killing this one. This is also a good place to do things like stop animations and other things that consume a noticeable amount of CPU in order to make the switch to the next activity as fast as possible, or to close resources that are exclusive access such as the camera.

In situations where the system needs more memory it may kill paused processes to reclaim resources. Because of this, you should be sure that all of your state is saved by the time you return from this function. In general `onSaveInstanceState(Bundle)` ([https://developer.android.com/reference/android/app/Activity.html#onSaveInstanceState\(android.os.Bundle\)](https://developer.android.com/reference/android/app/Activity.html#onSaveInstanceState(android.os.Bundle))) is used to save per-instance state in the activity and this method is used to store global persistent data (in content providers, files, etc.)

After receiving this call you will usually receive a following call to `onStop()` ([https://developer.android.com/reference/android/app/Activity.html#onStop\(\)](https://developer.android.com/reference/android/app/Activity.html#onStop())) (after the next activity has been resumed and displayed), however in some cases there will be a direct call back to `onResume()` ([https://developer.android.com/reference/android/app/Activity.html#onResume\(\)](https://developer.android.com/reference/android/app/Activity.html#onResume())) without going through the stopped state.

Derived classes must call through to the super class's implementation of this method. If they do not, an exception will be thrown.

See also:

`onResume()` ([https://developer.android.com/reference/android/app/Activity.html#onResume\(\)](https://developer.android.com/reference/android/app/Activity.html#onResume()))

`onSaveInstanceState(Bundle)` ([https://developer.android.com/reference/android/app/Activity.html#onSaveInstanceState\(android.os.Bundle\)](https://developer.android.com/reference/android/app/Activity.html#onSaveInstanceState(android.os.Bundle)))

`onStop()` ([https://developer.android.com/reference/android/app/Activity.html#onStop\(\)](https://developer.android.com/reference/android/app/Activity.html#onStop()))

onPostCreate

Added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

`void onPostCreate (Bundle` (<https://developer.android.com/reference/android/os/Bundle.html>) `) save`

This site uses cookies to store your preferences for site-specific language options.

OK

[/android/app/Activity.html#onStart\(\)](#) and [onRestoreInstanceState\(Bundle\)](#)

(<https://developer.android.com/reference/android>

[/app/Activity.html#onRestoreInstanceState\(android.os.Bundle\)](#)) have been called). Applications will generally not implement this method; it is intended for system classes to do final initialization after application code has run.

Derived classes must call through to the super class's implementation of this method. If they do not, an exception will be thrown.

Parameters	
<code>savedInstanceState</code>	Bundle: If the activity is being re-initialized after previously being shut down then this Bundle contains the data it most recently supplied in onSaveInstanceState(Bundle) (https://developer.android.com/reference/android/app/Activity.html#onSaveInstanceState(android.os.Bundle)). Note: <i>Otherwise it is null.</i>

See also:

[onCreate\(Bundle\)](#) (<https://developer.android.com/reference/android>

[/app/Activity.html#create\(android.os.Bundle\)](#))

onPostResume Added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
void onPostResume ()
```

Called when activity resume is complete (after [onResume\(\)](#) ([https://developer.android.com/reference/android/app/Activity.html#onResume\(\)](https://developer.android.com/reference/android/app/Activity.html#onResume())) has been called). Applications will generally not implement this method; it is intended for system classes to do final setup after application resume code has run.

Derived classes must call through to the super class's implementation of this method. If they do not, an exception will be thrown.

See also:

[onResume\(\)](#) ([https://developer.android.com/reference/android/app/Activity.html#onResume\(\)](https://developer.android.com/reference/android/app/Activity.html#onResume()))

This site uses cookies to store your preferences for site-specific language options.

onPrepareDialog

Added in API level 8 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
void onPrepareDialog (int id,
    Dialog (https://developer.android.com/reference/android/app/Dialog.html) dialog,
    Bundle (https://developer.android.com/reference/android/os/Bundle.html) args)
```

This method was deprecated in API level 13.

Use the new [DialogFragment](https://developer.android.com/reference/android/app/DialogFragment.html) (<https://developer.android.com/reference/android/app/DialogFragment.html>) class with [FragmentManager](https://developer.android.com/reference/android/app/FragmentManager.html) (<https://developer.android.com/reference/android/app/FragmentManager.html>) instead; this is also available on older platforms through the Android compatibility package.

Provides an opportunity to prepare a managed dialog before it is being shown. The default implementation calls through to [onPrepareDialog\(int, Dialog\)](https://developer.android.com/reference/android/app/Activity.html#onPrepareDialog(int, android.app.Dialog)) ([https://developer.android.com/reference/android/app/Activity.html#onPrepareDialog\(int, android.app.Dialog\)](https://developer.android.com/reference/android/app/Activity.html#onPrepareDialog(int, android.app.Dialog))) for compatibility.

Override this if you need to update a managed dialog based on the state of the application each time it is shown. For example, a time picker dialog might want to be updated with the current time. You should call through to the superclass's implementation. The default implementation will set this Activity as the owner activity on the Dialog.

Parameters	
id	int: The id of the managed dialog.
dialog	Dialog: The dialog.
args	Bundle: The dialog arguments provided to showDialog(int, Bundle) (https://developer.android.com/reference/android/app/Activity.html#showDialog(int, android.os.Bundle)).

See also:

[onCreateDialog\(int, Bundle\)](https://developer.android.com/reference/android/app/Activity.html#onCreateDialog(int, android.os.Bundle)) ([https://developer.android.com/reference/android/app/Activity.html#onCreateDialog\(int, android.os.Bundle\)](https://developer.android.com/reference/android/app/Activity.html#onCreateDialog(int, android.os.Bundle)))

[showDialog\(int\)](https://developer.android.com/reference/android/app/Activity.html#showDialog(int)) ([https://developer.android.com/reference/android/app/Activity.html#showDialog\(int\)](https://developer.android.com/reference/android/app/Activity.html#showDialog(int)))

[dismissDialog\(int\)](https://developer.android.com/reference/android/app/Activity.html#dismissDialog(int)) ([https://developer.android.com/reference/android/app/Activity.html#dismissDialog\(int\)](https://developer.android.com/reference/android/app/Activity.html#dismissDialog(int)))

[removeDialog\(int\)](https://developer.android.com/reference/android/app/Activity.html#removeDialog(int)) ([https://developer.android.com/reference/android/app/Activity.html#removeDialog\(int\)](https://developer.android.com/reference/android/app/Activity.html#removeDialog(int)))

This site uses cookies to store your preferences for site-specific language options.

OK

onPrepareDialog

Added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
void onPrepareDialog (int id,
    Dialog (https://developer.android.com/reference/android/app/Dialog.html) dialog)
```

This method was deprecated in API level 8.

Old no-arguments version of `onPrepareDialog(int, Dialog, Bundle)` ([https://developer.android.com/reference/android/app/Activity.html#onPrepareDialog\(int, android.app.Dialog, android.os.Bundle\)](https://developer.android.com/reference/android/app/Activity.html#onPrepareDialog(int, android.app.Dialog, android.os.Bundle))).

Parameters	
id	int
dialog	Dialog

onRestart

Added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
void onRestart ()
```

Called after `onStop()` ([https://developer.android.com/reference/android/app/Activity.html#onStop\(\)](https://developer.android.com/reference/android/app/Activity.html#onStop())) when the current activity is being re-displayed to the user (the user has navigated back to it). It will be followed by `onStart()` ([https://developer.android.com/reference/android/app/Activity.html#onStart\(\)](https://developer.android.com/reference/android/app/Activity.html#onStart())) and then `onResume()` ([https://developer.android.com/reference/android/app/Activity.html#onResume\(\)](https://developer.android.com/reference/android/app/Activity.html#onResume())).

For activities that are using raw `Cursor` (<https://developer.android.com/reference/android/database/Cursor.html>) objects (instead of creating them through `managedQuery(android.net.Uri, String[], String, String[], String)` ([https://developer.android.com/reference/android/app/Activity.html#managedQuery\(android.net.Uri, java.lang.String\[\], java.lang.String, java.lang.String\[\], java.lang.String\)](https://developer.android.com/reference/android/app/Activity.html#managedQuery(android.net.Uri, java.lang.String[], java.lang.String, java.lang.String[], java.lang.String)))), this is usually the place where the cursor should be requeried (because you had deactivated it in `onStop()` ([https://developer.android.com/reference/android/app/Activity.html#onStop\(\)](https://developer.android.com/reference/android/app/Activity.html#onStop()))).

Derived classes must call through to the super class's implementation of this method. If they do not, an exception will be thrown.

See also:

[onStop\(\)](https://developer.android.com/reference/android/app/Activity.html#onStop()) ([https://developer.android.com/reference/android/app/Activity.html#onStop\(\)](https://developer.android.com/reference/android/app/Activity.html#onStop()))

This site uses cookies to store your preferences for site-specific language options.

OK

`onResume()` ([https://developer.android.com/reference/android/app/Activity.html#onResume\(\)](https://developer.android.com/reference/android/app/Activity.html#onResume()))

onRestoreInstanceState

Added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

`void onRestoreInstanceState (Bundle` (<https://developer.android.com/reference/android/os/Bundle>)

This method is called after `onStart()` ([https://developer.android.com/reference/android/app/Activity.html#onStart\(\)](https://developer.android.com/reference/android/app/Activity.html#onStart())) when the activity is being re-initialized from a previously saved state, given here in ***savedInstanceState***. Most implementations will simply use `onCreate(Bundle)` ([https://developer.android.com/reference/android/app/Activity.html#onCreate\(android.os.Bundle\)](https://developer.android.com/reference/android/app/Activity.html#onCreate(android.os.Bundle))) to restore their state, but it is sometimes convenient to do it here after all of the initialization has been done or to allow subclasses to decide whether to use your default implementation. The default implementation of this method performs a restore of any view state that had previously been frozen by `onSaveInstanceState(Bundle)` ([https://developer.android.com/reference/android/app/Activity.html#onSaveInstanceState\(android.os.Bundle\)](https://developer.android.com/reference/android/app/Activity.html#onSaveInstanceState(android.os.Bundle))).

This method is called between `onStart()` ([https://developer.android.com/reference/android/app/Activity.html#onStart\(\)](https://developer.android.com/reference/android/app/Activity.html#onStart())) and `onPostCreate(Bundle)` ([https://developer.android.com/reference/android/app/Activity.html#onPostCreate\(android.os.Bundle\)](https://developer.android.com/reference/android/app/Activity.html#onPostCreate(android.os.Bundle))).

Parameters	
<code>savedInstanceState</code>	Bundle : the data most recently supplied in <code>onSaveInstanceState(Bundle)</code> (https://developer.android.com/reference/android/app/Activity.html#onSaveInstanceState(android.os.Bundle)).

See also:

`onCreate(Bundle)` ([https://developer.android.com/reference/android/app/Activity.html#onCreate\(android.os.Bundle\)](https://developer.android.com/reference/android/app/Activity.html#onCreate(android.os.Bundle)))

`onPostCreate(Bundle)` ([https://developer.android.com/reference/android/app/Activity.html#onPostCreate\(android.os.Bundle\)](https://developer.android.com/reference/android/app/Activity.html#onPostCreate(android.os.Bundle)))

`onResume()` ([https://developer.android.com/reference/android/app/Activity.html#onResume\(\)](https://developer.android.com/reference/android/app/Activity.html#onResume()))

`onSaveInstanceState(Bundle)` ([https://developer.android.com/reference/android/app/Activity.html#onSaveInstanceState\(android.os.Bundle\)](https://developer.android.com/reference/android/app/Activity.html#onSaveInstanceState(android.os.Bundle)))

This site uses cookies to store your preferences for site-specific language options.

OK

onResume

Added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
void onResume ()
```

Called after `onRestoreInstanceState(Bundle)` ([https://developer.android.com/reference/android/app/Activity.html#onRestoreInstanceState\(android.os.Bundle\)](https://developer.android.com/reference/android/app/Activity.html#onRestoreInstanceState(android.os.Bundle))), `onRestart()` ([https://developer.android.com/reference/android/app/Activity.html#onRestart\(\)](https://developer.android.com/reference/android/app/Activity.html#onRestart())), or `onPause()` ([https://developer.android.com/reference/android/app/Activity.html#onPause\(\)](https://developer.android.com/reference/android/app/Activity.html#onPause())), for your activity to start interacting with the user. This is a good place to begin animations, open exclusive-access devices (such as the camera), etc.

Keep in mind that `onResume` is not the best indicator that your activity is visible to the user; a system window such as the keyguard may be in front. Use `onWindowFocusChanged(boolean)` ([https://developer.android.com/reference/android/app/Activity.html#onWindowFocusChanged\(boolean\)](https://developer.android.com/reference/android/app/Activity.html#onWindowFocusChanged(boolean))) to know for certain that your activity is visible to the user (for example, to resume a game).

Derived classes must call through to the super class's implementation of this method. If they do not, an exception will be thrown.

See also:

`onRestoreInstanceState(Bundle)` ([https://developer.android.com/reference/android/app/Activity.html#onRestoreInstanceState\(android.os.Bundle\)](https://developer.android.com/reference/android/app/Activity.html#onRestoreInstanceState(android.os.Bundle)))

`onRestart()` ([https://developer.android.com/reference/android/app/Activity.html#onRestart\(\)](https://developer.android.com/reference/android/app/Activity.html#onRestart()))

`onPostResume()` ([https://developer.android.com/reference/android/app/Activity.html#onPostResume\(\)](https://developer.android.com/reference/android/app/Activity.html#onPostResume()))

`onPause()` ([https://developer.android.com/reference/android/app/Activity.html#onPause\(\)](https://developer.android.com/reference/android/app/Activity.html#onPause()))

onSaveInstanceState

Added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
void onSaveInstanceState (Bundle (https://developer.android.com/reference/android/os/Bundle.html))
```

Called to retrieve per-instance state from an activity before being killed so that the state can be restored in `onCreate(Bundle)` ([https://developer.android.com/reference/android/app/Activity.html#onCreate\(android.os.Bundle\)](https://developer.android.com/reference/android/app/Activity.html#onCreate(android.os.Bundle))) or `onRestoreInstanceState(Bundle)` ([https://developer.android.com/reference/android/app/Activity.html#onRestoreInstanceState\(android.os.Bundle\)](https://developer.android.com/reference/android/app/Activity.html#onRestoreInstanceState(android.os.Bundle))) (the `Bundle` (<https://developer.android.com/reference/android/os/Bundle.html>) populated by this method will be

This site uses cookies to store your preferences for site-specific language options.

This method is called before an activity may be killed so that when it comes back some time in the future it can restore its state. For example, if activity B is launched in front of activity A, and at some point activity A is killed to reclaim resources, activity A will have a chance to save the current state of its user interface via this method so that when the user returns to activity A, the state of the user interface can be restored via `onCreate(Bundle)` ([https://developer.android.com/reference/android/app/Activity.html#onCreate\(android.os.Bundle\)](https://developer.android.com/reference/android/app/Activity.html#onCreate(android.os.Bundle))) or `onRestoreInstanceState(Bundle)` ([https://developer.android.com/reference/android/app/Activity.html#onRestoreInstanceState\(android.os.Bundle\)](https://developer.android.com/reference/android/app/Activity.html#onRestoreInstanceState(android.os.Bundle))).

Do not confuse this method with activity lifecycle callbacks such as `onPause()` ([https://developer.android.com/reference/android/app/Activity.html#onPause\(\)](https://developer.android.com/reference/android/app/Activity.html#onPause())), which is always called when an activity is being placed in the background or on its way to destruction, or `onStop()` ([https://developer.android.com/reference/android/app/Activity.html#onStop\(\)](https://developer.android.com/reference/android/app/Activity.html#onStop())) which is called before destruction. One example of when `onPause()` ([https://developer.android.com/reference/android/app/Activity.html#onPause\(\)](https://developer.android.com/reference/android/app/Activity.html#onPause())) and `onStop()` ([https://developer.android.com/reference/android/app/Activity.html#onStop\(\)](https://developer.android.com/reference/android/app/Activity.html#onStop())) is called and not this method is when a user navigates back from activity B to activity A: there is no need to call `onSaveInstanceState(Bundle)` ([https://developer.android.com/reference/android/app/Activity.html#onSaveInstanceState\(android.os.Bundle\)](https://developer.android.com/reference/android/app/Activity.html#onSaveInstanceState(android.os.Bundle))) on B because that particular instance will never be restored, so the system avoids calling it. An example when `onPause()` ([https://developer.android.com/reference/android/app/Activity.html#onPause\(\)](https://developer.android.com/reference/android/app/Activity.html#onPause())) is called and not `onSaveInstanceState(Bundle)` ([https://developer.android.com/reference/android/app/Activity.html#onSaveInstanceState\(android.os.Bundle\)](https://developer.android.com/reference/android/app/Activity.html#onSaveInstanceState(android.os.Bundle))) is when activity B is launched in front of activity A: the system may avoid calling `onSaveInstanceState(Bundle)` ([https://developer.android.com/reference/android/app/Activity.html#onSaveInstanceState\(android.os.Bundle\)](https://developer.android.com/reference/android/app/Activity.html#onSaveInstanceState(android.os.Bundle))) on activity A if it isn't killed during the lifetime of B since the state of the user interface of A will stay intact.

The default implementation takes care of most of the UI per-instance state for you by calling `onSaveInstanceState()` ([https://developer.android.com/reference/android/view/View.html#onSaveInstanceState\(\)](https://developer.android.com/reference/android/view/View.html#onSaveInstanceState())) on each view in the hierarchy that has an id, and by saving the id of the currently focused view (all of which is restored by the default implementation of `onRestoreInstanceState(Bundle)` ([https://developer.android.com/reference/android/app/Activity.html#onRestoreInstanceState\(android.os.Bundle\)](https://developer.android.com/reference/android/app/Activity.html#onRestoreInstanceState(android.os.Bundle)))). If you override this method to save additional information not captured by each individual view, you will likely want to call through to the default implementation, otherwise be prepared to save all of the state of each view yourself.

If called, this method will occur before `onStop()` (<https://developer.android.com/reference/android...>)

This site uses cookies to store your preferences for site-specific language options.

Parameters	
<code>outState</code>	Bundle: Bundle in which to place your saved state.

See also:

`onCreate(Bundle)` ([https://developer.android.com/reference/android/app/Activity.html#onCreate\(android.os.Bundle\)](https://developer.android.com/reference/android/app/Activity.html#onCreate(android.os.Bundle)))

`onRestoreInstanceState(Bundle)` ([https://developer.android.com/reference/android/app/Activity.html#onRestoreInstanceState\(android.os.Bundle\)](https://developer.android.com/reference/android/app/Activity.html#onRestoreInstanceState(android.os.Bundle)))

`onPause()` ([https://developer.android.com/reference/android/app/Activity.html#onPause\(\)](https://developer.android.com/reference/android/app/Activity.html#onPause()))

onStart

Added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
void onStart ()
```

Called after `onCreate(Bundle)` ([https://developer.android.com/reference/android/app/Activity.html#onCreate\(android.os.Bundle\)](https://developer.android.com/reference/android/app/Activity.html#onCreate(android.os.Bundle))) — or after `onRestart()`

([https://developer.android.com/reference/android/app/Activity.html#onRestart\(\)](https://developer.android.com/reference/android/app/Activity.html#onRestart())) when the activity had been stopped, but is now again being displayed to the user. It will be followed by `onResume()` ([https://developer.android.com/reference/android/app/Activity.html#onResume\(\)](https://developer.android.com/reference/android/app/Activity.html#onResume())).

Derived classes must call through to the super class's implementation of this method. If they do not, an exception will be thrown.

See also:

`onCreate(Bundle)` ([https://developer.android.com/reference/android/app/Activity.html#onCreate\(android.os.Bundle\)](https://developer.android.com/reference/android/app/Activity.html#onCreate(android.os.Bundle)))

`onStop()` ([https://developer.android.com/reference/android/app/Activity.html#onStop\(\)](https://developer.android.com/reference/android/app/Activity.html#onStop()))

`onResume()` ([https://developer.android.com/reference/android/app/Activity.html#onResume\(\)](https://developer.android.com/reference/android/app/Activity.html#onResume()))

onStop

Added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

This site uses cookies to store your preferences for site-specific language options.

OK

([https://developer.android.com/reference/android/app/Activity.html#onRestart\(\)](https://developer.android.com/reference/android/app/Activity.html#onRestart())), `onDestroy()` ([https://developer.android.com/reference/android/app/Activity.html#onDestroy\(\)](https://developer.android.com/reference/android/app/Activity.html#onDestroy())), or nothing, depending on later user activity.

Derived classes must call through to the super class's implementation of this method. If they do not, an exception will be thrown.

See also:

`onRestart()` ([https://developer.android.com/reference/android/app/Activity.html#onRestart\(\)](https://developer.android.com/reference/android/app/Activity.html#onRestart()))

`onResume()` ([https://developer.android.com/reference/android/app/Activity.html#onResume\(\)](https://developer.android.com/reference/android/app/Activity.html#onResume()))

`onSaveInstanceState(Bundle)` ([https://developer.android.com/reference/android/app/Activity.html#onSaveInstanceState\(android.os.Bundle\)](https://developer.android.com/reference/android/app/Activity.html#onSaveInstanceState(android.os.Bundle)))

`onDestroy()` ([https://developer.android.com/reference/android/app/Activity.html#onDestroy\(\)](https://developer.android.com/reference/android/app/Activity.html#onDestroy()))

onTitleChanged Added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
void onTitleChanged (CharSequence (https://developer.android.com/reference/java/lang/CharSequence)
                    int color)
```

Parameters	
<code>title</code>	<code>CharSequence</code>
<code>color</code>	<code>int</code>

onUserLeaveHint Added in API level 3 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
void onUserLeaveHint ()
```

Called as part of the activity lifecycle when an activity is about to go into the background as the result of user choice. For example, when the user presses the Home key, `onUserLeaveHint()` ([https://developer.android.com/reference/android/app/Activity.html#onUserLeaveHint\(\)](https://developer.android.com/reference/android/app/Activity.html#onUserLeaveHint())) will be called, but when an incoming phone call causes the in-call Activity to be automatically brought to the foreground, `onUserLeaveHint()` (<https://developer.android.com/reference/android...>)

This site uses cookies to store your preferences for site-specific language options.

OK

([https://developer.android.com/reference/android/app/Activity.html#onPause\(\)](https://developer.android.com/reference/android/app/Activity.html#onPause())) callback.

This callback and `onUserInteraction()` ([https://developer.android.com/reference/android/app/Activity.html#onUserInteraction\(\)](https://developer.android.com/reference/android/app/Activity.html#onUserInteraction())) are intended to help activities manage status bar notifications intelligently; specifically, for helping activities determine the proper time to cancel a notification.

See also:

`onUserInteraction()` ([https://developer.android.com/reference/android/app/Activity.html#onUserInteraction\(\)](https://developer.android.com/reference/android/app/Activity.html#onUserInteraction()))

This site uses cookies to store your preferences for site-specific language options.