Android APIs API level: 19 nstants | Fields |
thods | Inherited
Meth ods | [Expand All]

Activity

extends
android
android.accessibilityservice
ContextThemeWrapper
android.accounts
android.animation
implements ComponentCallbacks2 KeyEvent.Callback Layou Inflater.Factory2
android.app
android.app.admin
java.lang.Object
android.app.backup
↳android.content.Context
android.appwidget
↳android.content.ContextWrapper
android.bluetooth
↳android.view.ContextThemeWrapper
android.content
↳android.app.Activity
android.content.pm
android.content.res
Known Direct Subclasses
android.database
AccountAuthenticatorActivity, ActivityGroup, AliasActivity,
android.database.sqlite
eActivity

Known Indirect Subclasses
Activity, LauncherActivity, PreferenceActivity, Tal Activity

Added in API level 1

Interfaces

ActionBar.OnMenuVisibilityListener
ActionBar.OnNavigationListener
ActionBar.TabListener
Application.ActivityLifecycleCallbacks
Application.OnProvideAssistDataListener
AppCompatCallback
AppCompatDelegate.ProviderNotFoundException
DatePickerDialog.OnDateSetListener
FragmentBreadCrumbs.OnBreadCrumbClickListener
FragmentManager.BackStackEntry
FragmentManager.OnBackStackChangedListener
KeyguardManager.OnKeyguardExitResult
LoaderManager.LoaderCallbacks
PendingIntent.OnFinished
SearchManager.OnCancelListener
SearchManager.OnDismissListener
TimePickerDialog.OnTimeSetListener
UiAutomation.AccessibilityEventFilter
UiAutomation.OnAccessibilityEventListener

Classes

ActionBar
ActionBar.LayoutParams
ActionBar.Tab
Activity
ActivityGroup
ActivityManager

Class Overview

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(android.view.View)` (/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` (/reference/android/R.attr.html#windowIsFloating) set) or embedded inside of another activity (using `ActivityGroup` (/reference/android/app/ActivityGroup.html)). There are two methods almost all subclasses of Activity will implement:

- `onCreate(Bundle)` is where you initialize your activity. Most importantly, here you will usually call `setContentView(int)` with a layout resource defining your UI, and using `findViewById(int)` to retrieve the widgets in that UI that you need to interact with programmatically.
- `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` holding the data).

To be used with `Context.startActivity()` (/reference/android

/content/Context.html#startActivity(android.content.Intent)), all activity classes must have a corresponding <activity> (/reference/android/R.styleable.html#AndroidManifestActivity) declaration in their package's AndroidManifest.xml.

Topics covered here:

1. Fragments
2. Activity Lifecycle
3. Configuration Changes
4. Starting Activities and Getting Results
5. Saving Persistent State
6. Permissions
7. Process Lifecycle

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 (/guide/topics/fundamentals.html) and Tasks and Back Stack (/guide/topics/fundamentals/tasks-and-back-stack.html) developer guides.

You can also find a detailed discussion about how to create activities in the Activities (/guide/topics/fundamentals/activities.html) developer guide.

Fragments

Starting with HONEYCOMB (/reference/android/os/Build.VERSION_CODES.html#HONEYCOMB), Activity implementations can make use of the Fragment (/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 in the foreground of the screen (at the top of the stack), it

Android APIs API level: 19

android
android.accessibilityservice
android.accounts
android.animation
android.app
android.app.admin
android.app.backup
android.appwidget
android.bluetooth
android.content
android.content.pm
android.content.res
android.database
android.database.sqlite

Interfaces

ActionBar.OnMenuVisibilityListener
ActionBar.OnNavigationListener
ActionBar.TabListener
Application.ActivityLifecycleCallbacks
Application.OnProvideAssistDataListener
AppOpsManager.OnOpChangedListener
DatePickerDialog.OnDateSetListener
FragmentBreadCrumbs.OnBreadCrumbClickListener
FragmentManager.BackStackEntry
FragmentManager.OnBackStackChangedListener
KeyguardManager.OnKeyguardExitResult
LoaderManager.LoaderCallbacks
PendingIntent.OnFinished
SearchManager.OnCancelListener
SearchManager.OnDismissListener
TimePickerDialog.OnTimeSetListener
UiAutomation.AccessibilityEventFilter
UiAutomation.OnAccessibilityEventListener

Classes

ActionBar
ActionBar.LayoutParams
ActionBar.Tab
Activity
ActivityGroup
ActivityManager

is *active* or *running*.

- If a new activity is started in the foreground 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.
- If an activity is paused or stopped, the system can drop the activity from memory by either asking it to finish, or simply killing its process. When it is displayed again to the user, it must be 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.

Android APIs API level: 19
android
android.accessibilityservice
android.accounts
android.animation
android.app
android.app.admin
android.app.backup
android.appwidget
android.bluetooth
android.content
android.content.pm
android.content.res
android.database
android.database.sqlite

Interfaces

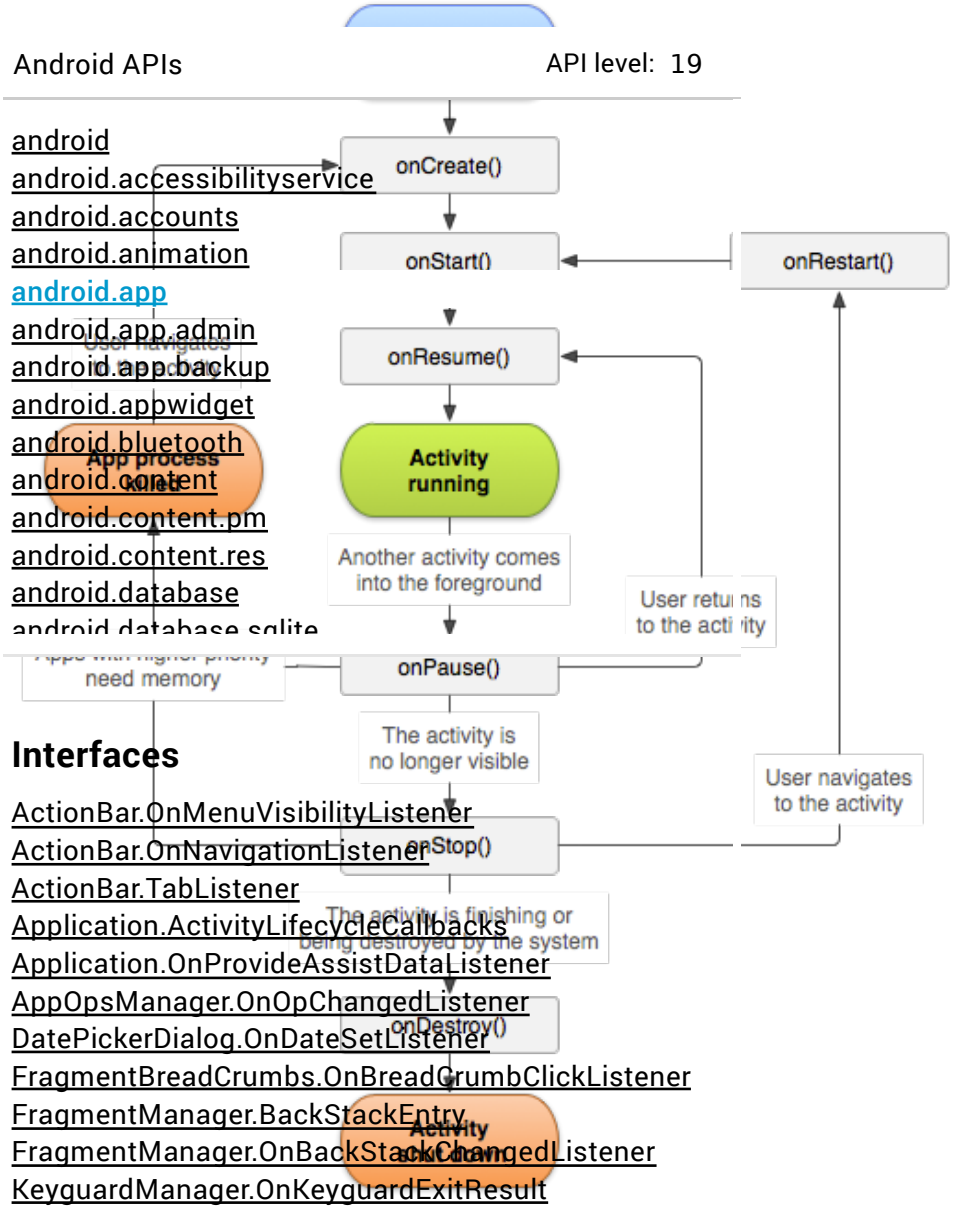
ActionBar.OnMenuVisibilityListener
ActionBar.OnNavigationListener
ActionBar.TabListener
Application.ActivityLifecycleCallbacks
Application.OnProvideAssistDataListener
AppOpsManager.OnOpChangedListener
DatePickerDialog.OnDateSetListener
FragmentBreadCrumbs.OnBreadCrumbClickListener
FragmentManager.BackStackEntry
FragmentManager.OnBackStackChangedListener
KeyguardManager.OnKeyguardExitResult
LoaderManager.LoaderCallbacks
PendingIntent.OnFinished
SearchManager.OnCancelListener
SearchManager.OnDismissListener
TimePickerDialog.OnTimeSetListener
UiAutomation.AccessibilityEventFilter
UiAutomation.OnAccessibilityEventListener

Classes

ActionBar
ActionBar.LayoutParams
ActionBar.Tab
Activity
ActivityGroup
ActivityManager

Android APIs API level: 19

android
android.accessibilityservice
android.accounts
android.animation
android.app
android.app.admin
android.app.backup
android.appwidget
android.bluetooth
android.content
android.content.pm
android.content.res
android.database
android.database.sqlite



Interfaces

ActionBar.OnMenuVisibilityListener
ActionBar.OnNavigationListener
ActionBar.TabListener
Application.ActivityLifecycleCallbacks
Application.OnProvideAssistDataListener
AppOpsManager.OnOpChangedListener
DatePickerDialog.OnDateSetListener
FragmentBreadCrumbs.OnBreadCrumbClickListener
FragmentManager.BackStackEntry
FragmentManager.OnBackStackChangedListener
KeyguardManager.OnKeyguardExitResult
LoaderManager.LoaderCallbacks
PendingIntent.OnFinished
SearchManager.OnCancelListener
SearchManager.OnDismissListener
TimePickerDialog.OnTimeSetListener
UiAutomation.AccessibilityEventFilter
UiAutomation.OnAccessibilityEventListener

Classes

ActionBar
ActionBar.LayoutParams
ActionBar.Tab
Activity
ActivityGroup
ActivityManager

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

- The **entire lifetime** of an activity happens between the f rst call to `onCreate(Bundle)` through to a single final call to `onDestroy()`. An activity will do all setup of "global" state in onCreate(), and 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()` until a corresponding call to `onStop()`. During this time the user can see the activity on-screen, though it may not be in the foreground and ou can maintain resources that are needed to show the activity to the user. For example,

you can register a `BroadcastReceiver` in onStart() to monitor for

Android APIs API level: 19

() when the user
d onStop()

methods can be called multiple times, as the activity be comes visible
and hidden to the user.

- The **foreground lifetime** of an activity happens between a call to onResume() until a corresponding call to onPause(). During this time ng with the user. An activity can frequently go between the resumed and paused states -- for example when the device goes to sleep, when an ac ivity result is delivered, when a new intent is delivered -- so the code i n these methods should be fairly lightweight.

The entire lifecycle of an activity is defined by the follo ing Activity methods. All of these are hooks that you can override to do appropriate work when the activity changes state. All activities will mplement onCreate(Bundle) (/reference/android /app/Activity.html#onCreate(android.os.Bundle)) to do the r initial setup; many will also implement onPause() (/reference/andr id /app/Activity.html#onPause()) to commit changes to data and otherwise prepare to stop interacting with the user. You should al ways call up to your superclass when implementing these methods.

```
public class Activity extends Applicat.onContext {
    protected void onCreate(Bundle savedInstanceSta
    protected void onStart();
    protected void onRestart();
    protected void onResume();
    protected void onPause();
    protected void onStop();
    protected void onDestroy();
}
```

Interfaces

Classes

ng on the movement through an activity's lifecycle ooks like this:

Method	Description	Killable?	Next
	Called when the created. This is		onStart()

Method	Description	Killable?	Next
Android APIs		API level: 19	
onRestart()	your normal static set up: create views, bind data to lists, ... also provides you with a Bundle containing the activity's previously frozen state, if there was one. ... by onStart().	No	onStart()
onStart()	Called after your activity has been stopped, prior to it being started again. Always followed by onStart()	No	onResume() or onStop()
onResume()	Called when the activity is becoming visible to the user. Followed by onPause() if the activity comes to the foreground, or onStop() if it becomes hidden.	No	onPause()
onPause()	Called when the activity will start interacting with the user. At this point your activity is at the ...	No	onStop() or onResume()

Interfaces

ActionBar.OnMenuVisibilityListener
ActionBar.OnNavigationListener
ActionBar.TabListener
Application.ActivityLifecycleCallbacks
Application.OnProvideAssistDataListener
AppOpsManager.OnOpChangedListener
DatePickerDialog.OnDateSetListener
FragmentBreadCrumbs.OnBreadCrumbClickListener
FragmentManager.BackStackEntry
FragmentManager.OnBackStackChangedListener
KeyguardManager.OnKeyguardExitResult
LoaderManager.LoaderCallbacks
PendingIntent.OnFinished
SearchManager.OnCancelListener
SearchManager.OnDismissListener
TimePickerDialog.OnTimeSetListener
UiAutomation.AccessibilityEventFilter
UiAutomation.OnAccessibilityEventListener

Classes

ActionBar
ActionBar.LayoutParams
ActionBar.Tab
Activity
ActivityGroup
ActivityManager

Method	Description	Killable?	Next
Android APIs		API level: 19	
	with user input going to it.		
	Always followed by onPause().		
	Called when the system is about to start resuming a previous activity. This is typically used to commit unsaved changes to		
	stop animations and other things that may be consuming CPU, etc. Implementations of this method must be very quick because the next activity will not be resumed until this method returns. Followed by either onResume() if the activity returns back to the front, or onStop() if it becomes invisible to the user.	Pre-HONEYCOMB	onResume() or onStop()
	Called when the activity is no	Yes	onRestart() or onDestroy()
	the user, because		

Interfaces

ActionBar.OnMenuVisibilityListener
ActionBar.OnNavigationListener
ActionBar.TabListener
Application.ActivityLifecycleCallbacks
Application.OnProvideAssistDataListener
AppOpsManager.OnOpChangedListener
DatePickerDialog.OnDateSetListener
FragmentBreadCrumbs.OnBreadCrumbClickListener
FragmentManager.BackStackEntry
FragmentManager.OnBackStackChangedListener
KeyguardManager.OnKeyguardExitResult
LoaderManager.LoaderCallbacks
PendingIntent.OnFinished
SearchManager.OnCancelListener
SearchManager.OnDismissListener
TimePickerDialog.OnTimeSetListener
UiAutomation.AccessibilityEventFilter
UiAutomation.OnAccessibilityEventListener

Classes

ActionBar
ActionBar.LayoutParams
ActionBar.Tab
Activity
ActivityGroup

Method	Description	Killable?	Next
Android APIs		API level: 19	
	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. Followed by either `onRestart()` if this activity is coming back to interact with the user, or `onDestroy()` if this activity is going away.		
	The final call you receive before your activity is destroyed. This can happen either because the activity is finishing (someone called `finish()` on it, or because the system is temporarily destroying this instance of the activity to save space. You can distinguish between these with the		
`onDestroy()`		Yes	*nothing*

Interfaces

ActionBar.OnMenuVisibilityListener
ActionBar.OnNavigationListener
ActionBar.TabListener
Application.ActivityLifecycleCallbacks
Application.OnProvideAssistDataListener
AppOpsManager.OnOpChangedListener
DatePickerDialog.OnDateSetListener
FragmentBreadCrumbs.OnBreadCrumbClickListener
FragmentManager.BackStackEntry
FragmentManager.OnBackStackChangedListener
KeyguardManager.OnKeyguardExitResult
LoaderManager.LoaderCallbacks
PendingIntent.OnFinished
SearchManager.OnCancelListener
SearchManager.OnDismissListener
TimePickerDialog.OnTimeSetListener
UiAutomation.AccessibilityEventFilter
UiAutomation.OnAccessibilityEventListener

Classes

ActionBar
ActionBar.LayoutParams
ActionBar.Tab
Activity
ActivityGroup
ActivityManager

Method	Description	Killable?	Next

Android APIs API level: 19

android
android.accessibilityservice
android.accounts
android.animation
android.app
android.app.admin
android.app.backup
android.appwidget
android.bluetooth
android.content
android.content.pm
android.content.res
android.database
android.database.sqlite

Interfaces

ActionBar.OnMenuVisibilityListener
ActionBar.OnNavigationListener
ActionBar.TabListener
Application.ActivityLifecycleCallbacks
Application.OnProvideAssistDataListener
AppOpsManager.OnOpChangedListener
DatePickerDialog.OnDateSetListener
FragmentBreadCrumbs.OnBreadCrumbClickListener
FragmentManager.BackStackEntry
FragmentManager.OnBackStackChangedListener
KeyguardManager.OnKeyguardExitResult
LoaderManager.LoaderCallbacks
PendingIntent.OnFinished
SearchManager.OnCancelListener
SearchManager.OnDismissListener
TimePickerDialog.OnTimeSetListener
UiAutomation.AccessibilityEventFilter
UiAutomation.OnAccessibilityEventListener

Classes

ActionBar
ActionBar.LayoutParams
ActionBar.Tab
Activity
ActivityGroup
ActivityManager

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 killed by the system *at any time* without another line of its code being executed. Because of this, you should use the `onPause()` (/reference/android/app/Activity.html#onPause()) method to write any persistent data (such as user edits) to storage. In addition, the method `onSaveInstanceState(Bundle)` (/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)` (/reference/android/app/Activity.html#onCreate(android.os.Bundle)) if the activity needs to be re-created. See the process lifecycle 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()` (/reference/android/app/Activity.html#onPause()) instead of `onSaveInstanceState(Bundle)` (/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` (/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()` (/reference/android/app/Activity.html#onStop()) has returned. This impacts when `onSaveInstanceState(Bundle)` (/reference/android/app/Activity.html#onSaveInstanceState(android.os.Bundle)) may be called (it may be safely called after `onPause()` (/reference/android/app/Activity.html#onPause()) and allows and application to safely wait until `onStop()` (/reference/android/app/Activity.html#onStop()) to save persistent state.

For those methods that are not marked as being killable, the activity's process will not be killed by the system starting from the time the method is called and continuing after it returns. Thus an activity is in the killable state, for example, between after `onPause()` to the start of `onResume()`.

If the configuration of the device (as defined by the

Android APIs API level: 19

android
android.accessibilityservice
android.accounts
android.animation
android.app
android.app.admin
android.app.backup
android.appwidget
android.bluetooth
android.content
android.content.pm
android.content.res
android.database
android.database.sqlite

Interfaces

ActionBar.OnMenuVisibilityListener
ActionBar.OnNavigationListener
ActionBar.TabListener
Application.ActivityLifecycleCallbacks
Application.OnProvideAssistDataListener
AppOpsManager.OnOpChangedListener
DatePickerDialog.OnDateSetListener
FragmentBreadCrumbs.OnBreadCrumbClickListener
FragmentManager.BackStackEntry
FragmentManager.OnBackStackChangedListener
KeyguardManager.OnKeyguardExitResult
LoaderManager.LoaderCallbacks
PendingIntent.OnFinished
SearchManager.OnCancelListener
SearchManager.OnDismissListener
TimePickerDialog.OnTimeSetListener
UiAutomation.AccessibilityEventFilter
UiAutomation.OnAccessibilityEventListener

Classes

ActionBar
ActionBar.LayoutParams
ActionBar.Tab
Activity
ActivityGroup
ActivityManager

ent

isplaying a user

Activity is the primary mechanism for interacting with the user, it includes special support for handling configuration changes.

uch 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()` (/reference/android/app/Activity.html#onPause()), `onStop()` (/reference/android/app/Activity.html#onStop()), and `onDestroy()` (/reference/android/app/Activity.html#onDestroy()) as appropriate. If the activity had been in the foreground or visible to the user, once `onDestroy()` (/reference/android/app/Activity.html#onDestroy()) is called in that instance then a

er

savedInstanceState the previous instance had generated from `onSaveInstanceState(Bundle)` (/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` (/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)` (/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)` (/reference/android/app/Activity.html#onConfigurationChanged(android.content.res.Configuration)) will not be called.

Android APIs API level: 19

android
android.accessibilityservice
android.accounts
android.animation
android.app
android.app.admin
android.app.backup
android.appwidget
android.bluetooth
android.content
android.content.pm
android.content.res
android.database
android.database.sqlite

The `startActivity(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` (`android.content.Intent`), 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)` (`android.app.Activity#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)` (`android.app.Activity#onActivityResult(int, int, android.content.Intent)`) method.

When an activity exits, it can call `setResult(int)` (`android.app.Activity#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;

    protected 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;
    }
```

Interfaces

ActionBar.OnMenuVisibilityListener
ActionBar.OnNavigationListener
ActionBar.TabListener
Application.ActivityLifecycleCallbacks
Application.OnProvideAssistDataListener
AppOpsManager.OnOpChangedListener
DatePickerDialog.OnDateSetListener
FragmentBreadCrumbs.OnBreadCrumbClickListener
FragmentManager.BackStackEntry
FragmentManager.OnBackStackChangedListener
KeyguardManager.OnKeyguardExitResult
LoaderManager.LoaderCallbacks
PendingIntent.OnFinished
SearchManager.OnCancelListener
SearchManager.OnDismissListener
TimePickerDialog.OnTimeSetListener
UiAutomation.AccessibilityEventFilter
UiAutomation.OnAccessibilityEventListener

Classes

ActionBar
ActionBar.LayoutParams
ActionBar.Tab
Activity
ActivityGroup
ActivityManager

```
    protected void onActivityResult(int requestCode
```

Android APIs API level: 19

```
            if (resultCode == RESULT_OK) {
                // A contact was picked.  Here we w
                // to the user.
                startActivity(new Intent(Intent.ACT
            }
        }
    }
```


Saving Persistent State

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

Interfaces

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 a new document is created, 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 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.

Classes

See the content package (/reference/android/content

Android APIs API level: 19

android
android.accessibilityservice
android.accounts
android.animation
android.app
android.app.admin
android.app.backup
android.appwidget
android.bluetooth
android.content
android.content.pm
android.content.res
android.database
android.database.sqlite

nt providers.
and propagate data between themselves.

The Activity class also provides an API for managing internal persistent state associated with an 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) (/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() (/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 savedInstanceSta
        super.onCreate(savedInstanceState);

        SharedPreferences mPrefs = getSharedPrefere
        mCurViewMode = mPrefs.getInt("view_mode", D
    }

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

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

Interfaces

ActionBar.OnMenuVisibilityListener
ActionBar.OnNavigationListener
ActionBar.TabListener
Application.ActivityLifecycleCallbacks
Application.OnProvideAssistDataListener
AppOpsManager.OnOpChangedListener
DatePickerDialog.OnDateSetListener
FragmentBreadCrumbs.OnBreadCrumbClickListener
FragmentManager.BackStackEntry
FragmentManager.OnBackStackChangedListener
KeyguardManager.OnKeyguardExitResult
LoaderManager.LoaderCallbacks
PendingIntent.OnFinished
SearchManager.OnCancelListener
SearchManager.OnDismissListener
TimePickerDialog.OnTimeSetListener
UiAutomation.AccessibilityEventFilter
UiAutomation.OnAccessibilityEventListener

Classes

ActionBar
ActionBar.LayoutParams
ActionBar.Tab
Activity
ActivityGroup
ActivityManager

```
        }
```

Android APIs API level: 19

android
android.accessibilityservice
android.accounts
android.animation
android.app
android.app.admin
android.app.backup
android.appwidget
android.bluetooth
android.content
android.content.pm
android.content.res
android.database
android.database.sqlite

Permissions

The ability to start a particular Activity can be enforced when it is declared in its manifest's <activity> (/reference/android/R.styleable.html#AndroidManifestActivity) tag. By doing so, other applications will need to declare a corresponding <uses-permission> (/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 (/reference/android/content/Intent.html#FLAG_GRANT_READ_URI_PERMISSION) and/or Intent.FLAG_GRANT_WRITE_URI_PERMISSION (/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 Honeycomb (/reference/android/os/Build.VERSION_CODES.html#GINGERBREAD), if the Activity was already created and a new Intent is being delivered to onNewIntent(Intent) (/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 (/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 one) before it resorts to killing more important processes (the first one).

1. The **foreground activity** (the activity at the top of the screen that

Interfaces

ActionBar.OnMenuVisibilityListener
ActionBar.OnNavigationListener
ActionBar.TabListener
Application.ActivityLifecycleCallbacks
Application.OnProvideAssistDataListener
AppOpsManager.OnOpChangedListener
DatePickerDialog.OnDateSetListener
FragmentBreadCrumbs.OnBreadCrumbClickListener
FragmentManager.BackStackEntry
FragmentManager.OnBackStackChangedListener
KeyguardManager.OnKeyguardExitResult
LoaderManager.LoaderCallbacks
PendingIntent.OnFinished
SearchManager.OnCancelListener
SearchManager.OnDismissListener
TimePickerDialog.OnTimeSetListener
UiAutomation.AccessibilityEventFilter
UiAutomation.OnAccessibilityEventListener

Classes

ActionBar
ActionBar.LayoutParams
ActionBar.Tab
Activity
ActivityGroup
ActivityManager

the user is currently interacting with) is considered the most esort, if it uses erally 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 led 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 (making it visible on the screen again), its `onCreate(Bundle)` (/reference/android/app/Activity.html#onCreate(android.os.Bundle)) method will be called with the savedInstanceState it had previously supplied in `onSaveInstanceState(Bundle)` (/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` (/reference/android/app/Service.html) or `BroadcastReceiver` (/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 will it is executing. To accomplish this, your Activity should start a `Service` (/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.

Interfaces

Classes

Summary

Constants

Android APIs API level: 19

android
android.accessibilityservice
android.accounts
android.animation
android.app
android.app.admin
android.app.backup
android.appwidget
android.bluetooth
android.content
android.content.pm
android.content.res
android.database
android.database.sqlite

Interfaces

ActionBar.OnMenuVisibilityListener
ActionBar.OnNavigationListener
ActionBar.TabListener
Application.ActivityLifecycleCallbacks
Application.OnProvideAssistDataListener
AppOpsManager.OnOpChangedListener
DatePickerDialog.OnDateSetListener
FragmentBreadCrumbs.OnBreadCrumbClickListener
FragmentManager.BackStackEntry
FragmentManager.OnBackStackChangedListener
KeyguardManager.OnKeyguardExitResult
LoaderManager.LoaderCallbacks
PendingIntent.OnFinished
SearchManager.OnCancelListener
SearchManager.OnDismissListener
TimePickerDialog.OnTimeSetListener
UiAutomation.OnAccessibilityEventFilter
UiAutomation.OnAccessibilityEventListener

Inherited Constants [Expand]

- From class android.content.Context
- From interface android.content.ComponentCallbacks2

Classes

ActionBar
ActionBar.LayoutParams
ActionBar.Tab
Activity
ActivityGroup
ActivityManager

int DEFAULT_KEYS_DISABLE
Use with /Mode(int) to during default key handling.

int DEFAULT_KEYS_SEARCH_GLOBAL
Use with setDefaultKe /Mode(int) to turn off default h andling of keys.

Use with setDefaultKe /Mode(int) to specify that unh andled keystrokes will start a global search (typically web search, but some platforms may define alternate metho ls for global search)

See android.app. earchManager (/reference/andro id /app/SearchManage r.html) for more details.

int DEFAULT_KEYS_SEARCH_LOCAL
Use with setDefaultKe /Mode(int) to specify that unh andled keystrokes will start an application-defi ed search.

int DEFAULT_KEYS_SHORTCUT
Use with setDefaultKe /Mode(int) to execute a menu shortcut in default key hanc ling.

int RESULT_CANCELED
Standard activity result: operation cance ed.

int RESULT_FIRST_USER
Start of user-def ned activity results.

int RESULT_OK
Standard activity result: operation succe ded.

Fields

protected static final int[] FOCUSED_STATE_SET

Public Constructors

Activity()

Public Me

Android APIs API level: 19

android
android.accessibilityservice
android.accounts
android.animation
android.app
android.app.admin
android.app.backup
android.appwidget
android.bluetooth
android.content
android.content.pm
android.content.res
android.database
android.database.sqlite

Interfaces

ActionBar.OnMenuVisibilityListener
ActionBar.OnNavigationListener
ActionBar.TabListener
Application.ActivityLifecycleCallbacks
Application.OnProvideAssistDataListener
AppOpsManager.OnOpChangedListener
DatePickerDialog.OnDateSetListener
FragmentBreadCrumbs.OnBreadCrumbClickListener
FragmentManager.BackStackEntry
FragmentManager.OnBackStackChangedListener
KeyguardManager.OnKeyguardExitResult
LoaderManager.LoaderCallbacks
PendingIntent.OnFinished
SearchManager.OnCancelListener
SearchManager.OnDismissListener
TimePickerDialog.OnTimeSetListener
UiAutomation.AccessibilityEventFilter
UiAutomation.OnAccessibilityEventListener

Classes

ActionBar
ActionBar.LayoutParams
ActionBar.Tab
Activity
ActivityGroup
ActivityManager

	roup.LayoutParams params) Add an additional content view t the activity.
void	closeContextMenu () Programmatically closes the mo st recently opened context menu, if s
void	closeOptionsMenu () ons menu.
PendingIntent	createPendingResult (int requestCo de, Intent data, int flags) Create a new PendingIntent obje ct which you can hand to others for t `Intent)` callback.
final void	dismissDialog (int id) *This method was deprecated in AP I level 13. Use the new `DialogFragm` through the Android compatibility package.*
boolean	dispatchGenericMotionEvent (Moti onEvent ev) Called to process generic motion events.
boolean	dispatchKeyEvent (KeyEvent event) Called to process key events.
boolean	dispatchKeyShortcutEvent (KeyEve nt event) Called to process a key shortcut event.
boolean	dispatchPopulateAccessibilityEver t (AccessibilityEvent event) Called to process population of `AccessibilityEvents`.
boolean	dispatchTouchEvent (MotionEvent ev) Called to process touch screen events.
boolean	dispatchTrackballEvent (MotionEve nt ev) Called to process trackball events.
void	dump (String prefix, FileDescriptor fd, PrintWriter writer, String[] args) Print the Activity's state into the given stream.
View	findViewById (int id) Finds a view that was identified by the id attribute from the XML that
void	finish () Call this when your activity is do ne and should be closed.
void	finishActivity (int requestCode) Force finish another activity that you had previously started with `sta`
void	finishActivityFromChild (Activity ch ild, int requestCode) This is called when a child activi y of this one calls its finishActivity().
void	finishAffinity () Finish this activity as well as all ctivities immediately below it in the
void	finishFromChild (Activity child) This is called when a child activi y of this one calls its `finish()` met

final Application

ComponentName

String

int

View

FragmentManager

Interfaces

Intent

Classes

SharedPreferences

getActionBar()

ty's ActionBar.

getApplication()

Return the application that owns this activity.

getCallingActivity()

Return the name of the activity that invoked this activity.

Return the name of the package that invoked this activity.

getChangingConfigurations()

If this activity is being destroyed because it can not handle a configur onConfigurationChanged(Configuration) method is *not* being occurred while in the process of being destroyed.

getComponentName()

Returns complete component name of this activity.

getCurrentFocus()

Calls `getCurrentFocus()` on the Window of this Activity to return t

getFragmentManager()

Return the FragmentManager for interacting with fragments associat

getIntent()

Return the intent that started this activity.

getLastNonConfigurationInstance()

This method was deprecated in API level 13. Use the new `Fragment` API through the Android compatibility package.

getLayoutInflater()

Convenience for calling `getLayoutInflater()`.

getLoaderManager()

Return the LoaderManager for this fragment, creating it if needed.

getLocalClassName()

Returns class name for this activity with the package prefix removed.

getMenuInflater()

Returns a `MenuInflater` with this context.

getParent()

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

getParentActivityIntent()

Obtain an `Intent` that will launch an explicit target activity specified

getPreferences(int mode)

Retrieve a `SharedPreferences` object for accessing preferences th

getRequestedOrientation()

Return the current requested orientation of the activity.

getSystemService (String name)

Android APIs API level: 19 el service by name.

getTaskId ()

int

android

Return the identifier of the task this activity is in.

android.accessibilityservice

final CharSequence getTitle ()

android.accounts

final int getTitleColor ()

android.animation

android.app

final int

Gets the suggested audio stream whose volume should be changed b

android.app.admin

android.app.backup

getWindow ()

android.appwidget

Window

Retrieve the current Window for the activity.

android.bluetooth

android.content

getWindowManager ()

WindowManager

Retrieve the window manager for showing custom windows.

android.content.pm

android.content.res

hasWindowFocus ()

boolean

Returns true if this activity's *main* window currently has window focus

android.database

android.database.sqlite

void

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

isChangingConfigurations ()

Interfaces

boolean

Check to see whether this activity is in the process of being destroyed

ActionBar.OnMenuVisibilityListener

isChild ()

ActionBar.OnNavigationListener

final boolean

Is this activity embedded inside of another activity?

ActionBar.TabListener

isDestroyed ()

Application.ActivityLifecycleCallbacks

boolean

Application.OnProvideAssistDataListener

Returns true if the final onDestroy() call has been made on the Act

AppOpsManager.OnOpChangedListener

isFinishing ()

DatePickerDialog.OnDateSetListener

boolean

Check to see whether this activity is in the process of finishing, either

FragmentBreadCrumbs.OnBreadCrumbClickListener

isImmersive ()

FragmentManager.BackStackEntry

boolean

Bit indicating that this activity is "immersive" and should not be interr

FragmentManager.OnBackStackChangedListener

isTaskRoot ()

KeyguardManager.OnKeyguardExitResult

boolean

LoaderManager.LoaderCallbacks

Returns whether this activity is the root of a task.

PendingIntent.OnFinished

managedQuery (Uri uri, String[] projection, String selection, String[] selec

SearchManager.OnCancelListener

final Cursor

This method was deprecated in API level 11. Use `CursorLoader` *instead*

SearchManager.OnDismissListener

moveTaskToBack (boolean nonRoot)

TimePickerDialog.OnTimeSetListener

boolean

Move the task containing this activity to the back of the activity stack

UiAutomation.AccessibilityEventFilter

UiAutomation.OnAccessibilityEventListener

navigateUpTo (Intent upIntent)

boolean

Navigate from this activity to the activity specified by upIntent, finishi

Classes

navigateUpToFromChild (Activity child, Intent upIntent)

boolean

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

ActionBar

onActionModeFinished (ActionMode mode)

ActionBar.LayoutParams

void

Notifies the activity that an action mode has finished.

ActionBar.Tab

mode)

Activity

Notifies the Activity that an action mode has been started.

ActivityGroup

ActivityManager

Android APIs API level: 19

android
android.accessibilityservice
android.accounts
android.animation
android.app
android.app.admin
android.app.backup
android.appwidget
android.bluetooth
android.content
android.content.pm
android.content.res
android.database
android.database.sqlite

Interfaces

ActionBar.OnMenuVisibilityListener
ActionBar.OnNavigationListener
ActionBar.TabListener
Application.ActivityLifecycleCallbacks
Application.OnProvideAssistDataListener
AppOpsManager.OnOpChangedListener
DatePickerDialog.OnDateSetListener
FragmentBreadCrumbs.OnBreadCrumbClickListener
FragmentManager.BackStackEntry
FragmentManager.OnBackStackChangedListener
KeyguardManager.OnKeyguardExitResult
LoaderManager.LoaderCallbacks
PendingIntent.OnFinished
SearchManager.OnCancelListener
SearchManager.OnDismissListener
TimePickerDialog.OnTimeSetListener
UiAutomation.AccessibilityEventFilter
UiAutomation.OnAccessibilityEventListener

Classes

ActionBar
ActionBar.LayoutParams
ActionBar.Tab
Activity
ActivityGroup
ActivityManager

void
void
boolean
void
void
CharSequence
void
boolean
boolean
View
boolean
View
View
void

onAttachFragment (Fragment fragment)
attached to this activity, immediately after the call to
Fragment.onCreate().

onAttachedToWindow ()
Called when the main window associated with the activity has been a

onBackPressed ()
cted the user's press of the back key

onConfigurationChanged (Configuration newConfig)
Called by the system when the device configuration changes while yo

onContentChanged ()
This hook is called whenever the content view of the screen changes

onContextItemSelected (MenuItem item)
This hook is called whenever an item in a context menu is selected.

onContextMenuClosed (Menu menu)
context menu is being closed (either

selected).

onCreateContextMenu (ContextMenu menu, View v, ContextMenu.Conte
Called when a context menu for the view is about to be shown.

onCreateDescription ()
Generate a new description for this activity.

onCreateNavigateUpTaskStack (TaskStackBuilder builder)
Define the synthetic task stack that will be generated during Up navig

onCreateOptionsMenu (Menu menu)
Initialize the contents of the Activity's standard options menu.

onCreatePanelMenu (int featureId, Menu menu)
Default implementation of onCreatePanelMenu(int, Menu) for a

onCreatePanelView (int featureId)
Default implementation of onCreatePanelView(int) for activities

onCreateThumbnail (Bitmap outBitmap, Canvas canvas)
Generate a new thumbnail for this activity.

onCreateView (View parent, String name, Context context, AttributeSet a
Standard implementation of onCreateView(View, String, Con
getSystemService(String)

onCreateView (String name, Context context, AttributeSet attrs)
Standard implementation of onCreateView(String, Context,
getSystemService(String)

onDetachedFromWindow ()
Called when the main window associated with the activity has been d

onGenericMotionEvent (MotionEvent event)
Called when a generic motion event was not handled by any of the vie

Android APIs API level: 19

android
android.accessibilityservice
android.accounts
android.animation
android.app
android.app.admin
android.app.backup
android.appwidget
android.bluetooth
android.content
android.content.pm
android.content.res
android.database
android.database.sqlite

Interfaces

ActionBar.OnMenuVisibilityListener
ActionBar.OnNavigationListener
ActionBar.TabListener
Application.ActivityLifecycleCallbacks
Application.OnProvideAssistDataListener
AppOpsManager.OnOpChangedListener
DatePickerDialog.OnDateSetListener
FragmentBreadCrumbs.OnBreadCrumbClickListener
FragmentManager.BackStackEntry
FragmentManager.OnBackStackChangedListener
KeyguardManager.OnKeyguardExitResult
LoaderManager.LoaderCallbacks
PendingIntent.OnFinished
SearchManager.OnCancelListener
SearchManager.OnDismissListener
TimePickerDialog.OnTimeSetListener
UiAutomation.AccessibilityEventFilter
UiAutomation.OnAccessibilityEventListener

Classes

ActionBar
ActionBar.LayoutParams
ActionBar.Tab
Activity
ActivityGroup
ActivityManager

boolean
boolean
boolean
boolean
void
boolean
boolean
boolean
boolean
void
void
void
boolean
boolean
boolean
void
Object

onKeyDown (int keyCode, KeyEvent event)
own and not handled by any of the vi
onKeyLongPress (int keyCode, KeyEvent event)
Default implementation of KeyEvent.Callback.onKeyLongPress
onKeyMultiple (int keyCode, int repeatCount, KeyEvent event)
Default implementation of KeyEvent.Callback.onKeyMultiple(
ent event)
Called when a key shortcut event is not handled by any of the views in
onKeyUp (int keyCode, KeyEvent event)
Called when a key was released and not handled by any of the views i
onLowMemory ()
This is called when the overall system is running low on memory, and
onMenuItemSelected (int featureId, MenuItem item)
Default implementation of onMenuItemSelected(int, MenuItem
u menu)
Called when a panel's menu is opened by the user.
onNavigateUp ()
This method is called whenever the user chooses to navigate Up with
onNavigateUpFromChild (Activity child)
This is called when a child activity of this one attempts to navigate up
onOptionsItemSelected (MenuItem item)
This hook is called whenever an item in your options menu is selected
onOptionsMenuClosed (Menu menu)
This hook is called whenever the options menu is being closed (either
selected).
onPanelClosed (int featureId, Menu menu)
Default implementation of onPanelClosed(int, Menu) for activit
onPrepareNavigateUpTaskStack (TaskStackBuilder builder)
Prepare the synthetic task stack that will be generated during Up navi
onPrepareOptionsMenu (Menu menu)
Prepare the Screen's standard options menu to be displayed.
onPreparePanel (int featureId, View view, Menu menu)
Default implementation of onPreparePanel(int, View, Menu) f
onProvideAssistData (Bundle data)
This is called when the user is requesting an assist, to build a full ACT
onRetainNonConfigurationInstance ()
This method was deprecated in API level 13. Use the new Fragment API
through the Android compatibility package.
onSearchRequested ()
This hook is called when the user signals the desire to start a search.

Android APIs API level: 19

android
android.accessibilityservice
android.accounts
android.animation
android.app
android.app.admin
android.app.backup
android.appwidget
android.bluetooth
android.content
android.content.pm
android.content.res
android.database
android.database.sqlite

Interfaces

ActionBar.OnMenuVisibilityListener
ActionBar.OnNavigationListener
ActionBar.TabListener
Application.ActivityLifecycleCallbacks
Application.OnProvideAssistDataListener
AppOpsManager.OnOpChangedListener
DatePickerDialog.OnDateSetListener
FragmentBreadCrumbs.OnBreadCrumbClickListener
FragmentManager.BackStackEntry
FragmentManager.OnBackStackChangedListener
KeyguardManager.OnKeyguardExitResult
LoaderManager.LoaderCallbacks
PendingIntent.OnFinished
SearchManager.OnCancelListener
SearchManager.OnDismissListener
TimePickerDialog.OnTimeSetListener
UiAutomation.AccessibilityEventFilter
UiAutomation.OnAccessibilityEventListener

Classes

ActionBar
ActionBar.LayoutParams
ActionBar.Tab
Activity
ActivityGroup
ActivityManager

boolean
onTouchEvent (MotionEvent event)
t was not handled by any of the view
onTrackballEvent (MotionEvent event)
Called when the trackball was moved and not handled by any of the v
onTrimMemory (int level)
Called when the operating system has determined that it is a good tin
Called whenever a key, touch, or trackball event is dispatched to the a
void
onWindowAttributesChanged (WindowManager.LayoutParams params)
This is called whenever the current window attributes change.
onWindowFocusChanged (boolean hasFocus)
Called when the current Window of the activity gains or loses focus.
onWindowStartingActionMode (ActionMode.Callback callback)
ActionMode
Give the Activity a chance to control the UI for an action mode reques
void
Programmatically opens the context menu for a particular view.
openOptionsMenu ()
void
Programmatically opens the options menu.
overridePendingTransition (int enterAnim, int exitAnim)
void
Call immediately after one of the flavors of startActivity(Intent
recreate ()
void
Cause this Activity to be recreated with a new instance.
registerForContextMenu (View view)
void
Registers a context menu to be shown for the given view (multiple vie
removeDialog (int id)
final void
This method was deprecated in API level 13. Use the new DialogFragm
through the Android compatibility package.
reportFullyDrawn ()
void
Report to the system that your app is now fully drawn, purely for diagn
requestWindowFeature (int featureId)
final boolean
Enable extended window features.
runOnUiThread (Runnable action)
final void
Runs the specified action on the UI thread.
setContentView (int layoutResID)
void
Set the activity content from a layout resource.
setContentView (View view)
void
Set the activity content to an explicit view.
setContentView (View view, ViewGroup.LayoutParams params)
void
Set the activity content to an explicit view.

Android APIs API level: 19

android
android.accessibilityservice
android.accounts
android.animation
android.app
android.app.admin
android.app.backup
android.appwidget
android.bluetooth
android.content
android.content.pm
android.content.res
android.database
android.database.sqlite

Interfaces

ActionBar.OnMenuVisibilityListener
ActionBar.OnNavigationListener
ActionBar.TabListener
Application.ActivityLifecycleCallbacks
Application.OnProvideAssistDataListener
AppOpsManager.OnOpChangedListener
DatePickerDialog.OnDateSetListener
FragmentBreadCrumbs.OnBreadCrumbClickListener
FragmentManager.BackStackEntry
FragmentManager.OnBackStackChangedListener
KeyguardManager.OnKeyguardExitResult
LoaderManager.LoaderCallbacks
PendingIntent.OnFinished
SearchManager.OnCancelListener
SearchManager.OnDismissListener
TimePickerDialog.OnTimeSetListener
UiAutomation.AccessibilityEventFilter
UiAutomation.OnAccessibilityEventListener

Classes

ActionBar
ActionBar.LayoutParams
ActionBar.Tab
Activity
ActivityGroup
ActivityManager

setDefaultKeyMode (int mode)
r this activity.

final void setFeatureDrawable (int featureId, rawable drawable)
Convenience for calling setFea ureDrawable(int, Drawable).

final void setFeatureDrawableAlpha (int featu reId, int alpha)
Convenience for calling setFea ureDrawableAlpha(int, int).

eatureId, int resId)
Convenience for calling setFea ureDrawableResource(int, ir

setFeatureDrawableUri (int feature d, Uri uri)
Convenience for calling setFea ureDrawableUri(int, Uri).

setFinishOnTouchOutside (boolear finish)
Sets whether this activity is finis ed when touched outside its windo

setImmersive (boolean i)
Adjust the current immersive mo de setting.

void
Change the intent returned by ge tIntent().

final void setProgress (int progress)
Sets the progress for the progres s bars in the title.

final void setProgressBarIndeterminate (boo ean indeterminate)
Sets whether the horizontal prog ress bar in the title should be indeter

final void setProgressBarIndeterminateVisib lity (boolean visible)
Sets the visibility of the indeterm inate progress bar in the title.

setProgressBarVisibility (boolean v isible)
Sets the visibility of the progress bar in the title.

void setRequestedOrientation (int reque stedOrientation)
Change the desired orientation o f this activity.

final void setResult (int resultCode)
Call this to set the result that you r activity will return to its caller.

setResult (int resultCode, Intent da a)
Call this to set the result that you r activity will return to its caller.

setSecondaryProgress (int second ryProgress)
Sets the secondary progress for he progress bar in the title.

setTheme (int resid)
Change the title associated with this activity.

void setTitle (CharSequence title)
Change the title associated with this activity.

void setTitleColor (int textColor)

setVisible (boolean visible)
ain window is visible.

Android APIs API level: 19

android
android.accessibilityservice
android.accounts
android.animation
android.app
android.app.admin
android.app.backup
android.appwidget
android.bluetooth
android.content
android.content.pm
android.content.res
android.database
android.database.sqlite

Interfaces

ActionBar.OnMenuVisibilityListener
ActionBar.OnNavigationListener
ActionBar.TabListener
Application.ActivityLifecycleCallbacks
Application.OnProvideAssistDataListener
AppOpsManager.OnOpChangedListener
DatePickerDialog.OnDateSetListener
FragmentBreadCrumbs.OnBreadCrumbClickListener
FragmentManager.BackStackEntry
FragmentManager.OnBackStackChangedListener
KeyguardManager.OnKeyguardExitResult
LoaderManager.LoaderCallbacks
PendingIntent.OnFinished
SearchManager.OnCancelListener
SearchManager.OnDismissListener
TimePickerDialog.OnTimeSetListener
UiAutomation.AccessibilityEventFilter
UiAutomation.OnAccessibilityEventListener

Classes

ActionBar
ActionBar.LayoutParams
ActionBar.Tab
Activity
ActivityGroup
ActivityManager

final boolean

ActionMode

void

void

void

void

void

void

void

boolean

boolean

void

setVolumeControlStream (int streamType)
: volume should be changed by the h

shouldUpRecreateTask (Intent targetIntent)
Returns true if the app should re:reate the task when navigating 'up' f

showDialog (int id, Bundle args)
This method was deprecated in AP I level 13. Use the new `DialogFragm
package.

showDialog (int id)
This method was deprecated in AP I level 13. Use the new `DialogFragm
through the Android compatibility package.

startActionMode (ActionMode.Callback callback)
Start an action mode.

startActivities (Intent[] intents, Bundle options)
Launch a new activity.

startActivities (Intent[] intents)
Same as `startActivities(Intent[], Bundle)` with no options

startActivity (Intent intent)
Same as `startActivity(Intent, Bundle)` with no options spec

startActivity (Intent intent, Bundle options)
Launch a new activity.

startActivityForResult (Intent intent, int requestCode)
Same as calling `startActivityForResult(Intent, int, Bund`

startActivityForResult (Intent intent, int requestCode, Bundle options)
Launch an activity for which you would like a result when it finished.

startActivityFromChild (Activity child, Intent intent, int requestCode, Bun
This is called when a child activity of this one calls its `startActivi`

startActivityFromChild (Activity child, Intent intent, int requestCode)
Same as calling `startActivityFromChild(Activity, Intent`

startActivityFromFragment (Fragment fragment, Intent intent, int reques
This is called when a Fragment in this activity calls its `startActivi`

startActivityFromFragment (Fragment fragment, Intent intent, int reques
Same as calling `startActivityFromFragment(Fragment, Int`

startActivityIfNeeded (Intent intent, int requestCode, Bundle options)
A special variation to launch an activity only if a new activity instance

startActivityIfNeeded (Intent intent, int requestCode)
Same as calling `startActivityIfNeeded(Intent, int, Bund`

startIntentSender (IntentSender intent, Intent fillInIntent, int flagsMask,
Like `startActivity(Intent, Bundle)`, but taking a IntentSende
`int, int, int, Bundle)` for more information.

Android APIs API level: 19

android
android.accessibilityservice
android.accounts
android.animation
android.app
android.app.admin
android.app.backup
android.appwidget
android.bluetooth
android.content
android.content.pm
android.content.res
android.database
android.database.sqlite

Interfaces

ActionBar.OnMenuVisibilityListener
ActionBar.OnNavigationListener
ActionBar.TabListener
Application.ActivityLifecycleCallbacks
Application.OnProvideAssistDataListener
AppOpsManager.OnOpChangedListener
DatePickerDialog.OnDateSetListener
FragmentBreadCrumbs.OnBreadCrumbClickListener
FragmentManager.BackStackEntry
FragmentManager.OnBackStackChangedListener
KeyguardManager.OnKeyguardExitResult
LoaderManager.LoaderCallbacks
PendingIntent.OnFinished
SearchManager.OnCancelListener
SearchManager.OnDismissListener
TimePickerDialog.OnTimeSetListener
UiAutomation.AccessibilityEventFilter
UiAutomation.OnAccessibilityEventListener

Classes

ActionBar
ActionBar.LayoutParams
ActionBar.Tab
Activity
ActivityGroup
ActivityManager

startIntentSender (IntentSender intent, Intent fillInIntent, int flagsMask, ender(IntentSender, Intent,

startIntentSenderForResult (Intent ender intent, int requestCode, Intent
Like `startActivityForResult(Intent, int)`, but allowing you

startIntentSenderForResult (Intent ender intent, int requestCode, Intent
Same as calling `startIntentS enderForResult(IntentSender`

y child, IntentSender intent, int reque

Same as calling `startIntentS enderFromChild(Activity, In`

startIntentSenderFromChild (Activi y child, IntentSender intent, int reque
Like `startActivityFromChild(Activity, Intent, int)`, but
`int, Intent, int, int, i nt)` for more information.

startManagingCursor (Cursor c)
This method was deprecated in AP I level 11. Use the new `CursorLoade` the Android compatibility package

ntent)
Same as calling `startNextMat chingActivity(Intent, Bundl`

startNextMatchingActivity (Intent i ntent, Bundle options)
Special version of starting an ac ivity, for use when you are replacing

startSearch (String initialQuery, bo lean selectInitialQuery, Bundle appS
This hook is called to launch the search UI.

stopManagingCursor (Cursor c)
This method was deprecated in AP I level 11. Use the new `CursorLoade` the Android compatibility package

takeKeyEvents (boolean get)
Request that key events come to this activity.

triggerSearch (String query, Bundle appSearchData)
Similar to `startSearch(Strin g, boolean, Bundle, boolean`

unregisterForContextMenu (View v ew)
Prevents a context menu to be s own for the given view.

Protected Methods

onActivityResult (int requestCode, int resultCode, Intent data)
Called when an activity you launched exits, gi ing you the requestCode you started it with, the resultCod e it returned, and any additional data from it.

onApplyThemeResource (Resources.Theme the ne, int resid, boolean first)
Called by `setTheme(int)` and `getTheme()` to apply a theme resource to the current Theme object.

void onChildTitleChanged (Activity childActivity, Char Sequence title)

void onCreate (Bundle savedInstanceState)

onCreateDialog(int id)
This method was deprecated in API level 8. Old no-arguments version of `onCreateDialog(int, Bundle)`.

onCreateDialog(int id, Bundle args)
This method was deprecated in API level 13. Use the new `DialogFragment` class with `FragmentManager` instead; this is also available on older platforms through the Android compatibility package.

onDestroy()
void
Perform any final cleanup before an activity is destroyed.

onNewIntent(Intent intent)
void
This is called for activities that set launchMode to "singleTop" in their package, or if a client used the `FLAG_ACTIVITY_SINGLE_TOP` flag when calling `startActivity(Intent)`.

onPause()
void
Called as part of the activity lifecycle when an activity is going into the background, but has not (yet) been killed.

onPostCreate(Bundle savedInstanceState)
void
Called when activity start-up is complete (after `onStart()` and `onRestoreInstanceState(Bundle)` have been called).

onPostResume()
void
Called when activity resume is complete (after `onResume()` has been called).

onPrepareDialog(int id, Dialog dialog)
This method was deprecated in API level 8. Old no-arguments version of `onPrepareDialog(int, Dialog, Bundle)`.

onPrepareDialog(int id, Dialog dialog, Bundle args)
This method was deprecated in API level 13. Use the new `DialogFragment` class with `FragmentManager` instead; this is also available on older platforms through the Android compatibility package.

onRestart()
void
Called after `onStop()` when the current activity is being re-displayed to the user (the user has navigated back to it).

onRestoreInstanceState(Bundle savedInstanceState)
void
This method is called after `onStart()` when the activity is being re-initialized from a previously saved state, given here in *savedInstanceState*.

onResume()
void
Called after `onRestoreInstanceState(Bundle)`, `onRestart()`, or `onPause()`, for your activity to start interacting with the user.

onSaveInstanceState(Bundle outState)
void
Called to retrieve per-instance state from an activity before being killed so that the state can be restored in `onCreate(Bundle)` or

Interfaces

Classes

onRestoreInstanceState(Bundle) (the Bundle populated by

Android APIs API level: 19

onStart()

android Called after onCreate(Bundle) — or after onRestart() when the

void activity had been stopped, but is now again being displayed to the

android.accessibilityservice user.

android.accounts

android.animation onStop()

android.app Called when you are no longer visible to the user.

android.app.admin void onTitleChanged (CharSequence title, int color)

android.app.backup

android.appwidget onUserLeaveHint()

android.bluetooth void Called as part of the activity lifecycle when an activity is about to go

android.content into the background as the result of user choice.

android.content.pm

android.content.res **Inherited Methods** [Expand]

android.database

android.databaseandroid From class android.view.ContextThemeWrapper

From class android.content.ContextWrapper

From class android.content.Context

From class java.lang.Object

From interface android.content.ComponentCallbacks

Interfaces

From interface android.content.ComponentCallbacks2

ActionBar.OnMenuVisibilityListener From interface android.view.KeyEvent.Callback

ActionBar.OnNavigationListener From interface android.view.LayoutInflater.Factory

ActionBar.TabListener From interface android.view.LayoutInflater.Factory2

Application.ActivityLifecycleCallbacks From interface android.view.View.OnCreateContextMenuListener

Application.OnProvideAssistDataListener From interface android.view.Window.Callback

AppOpsManager.OnOpChangedListener

DatePickerDialog.OnDateSetListener

FragmentBreadCrumbs.OnBreadCrumbClickListener

FragmentManager.BackStackEntry

FragmentManager.OnBackStackChangedListener

KeyguardManager.OnKeyguardExitResult

LoaderManager.LoaderCallbacks

PendingIntent.OnFinished

SearchManager.OnCancelListener

SearchManager.OnDismissListener

TimePickerDialog.OnTimeSetListener

UiAutomation.AccessibilityEventFilter

UiAutomation.OnAccessibilityEventListener

Constants

public static final int **DEFAULT_KEYS_DIALER** Added in API level 1

Use with setDefaultKeyMode(int) (/reference/android/app/Activity.html#setDefaultKeyMode(int)) to launch the dialer during default key handling.

See Also

setDefaultKeyMode(int)

Constant Value: 1 (0x00000001)

Classes

ActionBar

ActionBar.LayoutParams

ActionBar.Tab

Activity

ActivityGroup

ActivityManager

public static final int **DEFAULT_KEYS_DISABLE** Added in API level 1

Use with setDefaultKeyMode(int) (/reference/android/app/Activity.html#setDefaultKeyMode(int)) to turn off default handling

of keys.

Android APIs API level: 19

setDefaultKeyMode(int)

Constant Value: 0 (0x00000000)

public static final int **DEFAULT_KEYS_DISABLE** Added in API level 1

Use with setDefaultKeyMode(int) (/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 (/reference/android/app/SearchManager.html) for more details.

See Also

setDefaultKeyMode(int)

Constant Value: 4 (0x00000004)

public static final int **DEFAULT_KEYS_SEARCH_LOCAL** Added in API level 1

Use with setDefaultKeyMode(int) (/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 keys will be ignored.)

See android.app.SearchManager (/reference/android/app/SearchManager.html) for more details.

See Also

setDefaultKeyMode(int)

Constant Value: 3 (0x00000003)

public static final int **DEFAULT_KEYS_SHORTCUT** Added in API level 1

Use with setDefaultKeyMode(int) (/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

Interfaces

ActionBar.OnMenuVisibilityListener
ActionBar.OnNavigationListener
ActionBar.TabListener
Application.ActivityLifecycleCallbacks
Application.OnProvideAssistDataListener
AppOpsManager.OnOpChangedListener
DatePickerDialog.OnDateSetListener
FragmentBreadCrumbs.OnBreadCrumbClickListener
FragmentManager.BackStackEntry
FragmentManager.OnBackStackChangedListener
KeyguardManager.OnKeyguardExitResult
LoaderManager.LoaderCallbacks
PendingIntent.OnFinished
SearchManager.OnCancelListener
SearchManager.OnDismissListener
TimePickerDialog.OnTimeSetListener
UiAutomation.AccessibilityEventFilter
UiAutomation.OnAccessibilityEventListener

Classes

ActionBar
ActionBar.LayoutParams
ActionBar.Tab
Activity
ActivityGroup
ActivityManager

See Also

Android APIs API level: 19

Constant Value: 2 (0x00000002)

public static final int **RESULT_CANCELED** Added in API level 1

Standard activity result: operation canceled.

Constant Value: 0 (0x00000000)

public static final int **RESULT_FIRST_USER** Added in API level 1

Start of user-defined activity results.

Constant Value: 1 (0x00000001)

public static final int **RESULT_OK** Added in API level 1

Standard activity result: operation succeeded.

Constant Value: -1 (0xffffffff)

Interfaces

ActionBar.OnMenuVisibilityListener
ActionBar.OnNavigationListener
ActionBar.TabListener
Application.ActivityLifecycleCallbacks
Application.OnProvideAssistDataListener
AppOpsManager.OnOpChangedListener
DatePickerDialog.OnDateSetListener
FragmentBreadCrumbs.OnBreadCrumbClickListener
FragmentManager.BackStackEntry
FragmentManager.OnBackStackChangedListener
KeyguardManager.OnKeyguardExitResult
LoaderManager.LoaderCallbacks
PendingIntent.OnFinished
SearchManager.OnCancelListener
SearchManager.OnDismissListener
TimePickerDialog.OnTimeSetListener
UiAutomation.AccessibilityEventFilter
UiAutomation.OnAccessibilityEventListener

Fields

protected static final int[] **FOCUSED_STATE_SET** Added in API level 1

Public Constructors

public **Activity** () Added in API level 1

Public Methods

public void **addContentView** (View view, ViewGroup.LayoutParams params) Added in API level 1

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

Parameters

view The desired content to display.

params Layout parameters for the view.

Classes

ActionBar
ActionBar.LayoutParams
ActionBar.Tab
Activity
ActivityGroup
ActivityManager

public void **closeContextMenu** () Added in API level 3

Android APIs API level: 19

ontext menu, if showing.

public void **closeOptionsMenu** () Added in API level 1

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

public PendingIntent **createPendingResult** (int requestCode, Intent data, int flags) Added in API level 1

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)` (/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 Private request code for the sender that will be associated with the result data when it is received. The sender can not modify this value, allowing you to identify incoming results.

data Default data to supply in the result, which may be modified by the sender.

flags May be `PendingIntent.FLAG_ONE_SHOT`, `PendingIntent.FLAG_NO_CREATE`, `PendingIntent.FLAG_CANCEL_CURRENT`, `PendingIntent.FLAG_UPDATE_CURRENT`, or any of the flags as supported by `Intent.fillIn()` to control which unspecified parts of the intent that can be supplied when the actual send happens.

Returns

Returns an existing or new PendingIntent matching the given parameters. May return null only if `PendingIntent.FLAG_NO_CREATE` has been supplied.

See Also

PendingIntent

public final void **dismissDialog** (int id) Added in API level 1

Interfaces

ActionBar.OnMenuVisibilityListener
ActionBar.OnNavigationListener
ActionBar.TabListener
Application.ActivityLifecycleCallbacks
Application.OnProvideAssistDataListener
AppOpsManager.OnOpChangedListener
DatePickerDialog.OnDateSetListener
FragmentBreadCrumbs.OnBreadCrumbClickListener
FragmentManager.BackStackEntry
FragmentManager.OnBackStackChangedListener
KeyguardManager.OnKeyguardExitResult
LoaderManager.LoaderCallbacks
PendingIntent.OnFinished
SearchManager.OnCancelListener
SearchManager.OnDismissListener
TimePickerDialog.OnTimeSetListener
UiAutomation.AccessibilityEventFilter
UiAutomation.OnAccessibilityEventListener

Classes

ActionBar
ActionBar.LayoutParams
ActionBar.Tab
Activity
ActivityGroup
ActivityManager

This method was deprecated in API level 13.

Android APIs API level: 19

er (/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).

Parameters

id The id of the managed dialog.

Throws

IllegalArgumentException if the id was not previously shown via showDialog(int).

See Also

onCreateDialog(int, Bundle)
onPrepareDialog(int, Dialog, Bundle)
showDialog(int)
removeDialog(int)

Interfaces

ActionBar.OnMenuVisibilityListener
ActionBar.OnNavigationListener
ActionBar.TabListener
Application.ActivityLifecycleCallbacks
Application.OnProvideAssistDataListener
AppOpsManager.OnOpChangedListener
DatePickerDialog.OnDateSetListener
FragmentBreadCrumbs.OnBreadCrumbClickListener
FragmentManager.BackStackEntry
FragmentManager.OnBackStackChangedListener
KeyguardManager.OnKeyguardExitResult
LoaderManager.LoaderCallbacks
PendingIntent.OnFinished
SearchManager.OnCancelListener
SearchManager.OnDismissListener
TimePickerDialog.OnTimeSetListener
UiAutomation.AccessibilityEventFilter
UiAutomation.OnAccessibilityEventListener

Classes

ActionBar
ActionBar.LayoutParams
ActionBar.Tab
Activity
ActivityGroup
ActivityManager

public boolean **dispatchGenericMotionEvent** (MotionEvent ev) Added in API level 12

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 The generic motion event.

Returns

boolean Return true if this event was consumed.

public boolean **dispatchKeyEvent** (KeyEvent event) Added in API level 1

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 The key event.

Returns

boolean Return true if this event was consumed.

public boolean **dispatchKevShortcutEvent**

Android APIs API level: 19 Added in API level 11

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.

event The key shortcut event.

Returns

True if this event was consumed.

public boolean **dispatchPopulateAccessibilityEvent** (AccessibilityEvent event) Added in API level 4

Dispatch an {@link android.view.accessibility.AccessibilityEvent} (/reference/android/view/accessibility/AccessibilityEvent.html)s.

Parameters

event The event.

Returns

boolean Return true if event population was completed.

public boolean **dispatchTouchEvent** (MotionEvent ev) Added in API level 1

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 events that should be handled normally.

Parameters

ev The touch screen event.

Returns

boolean Return true if this event was consumed.

public boolean **dispatchTrackballEvent** (MotionEvent ev) Added in API level 1

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

Interfaces

ActionBar.OnMenuVisibilityListener
ActionBar.OnNavigationListener
ActionBar.TabListener
Application.ActivityLifecycleCallbacks
Application.OnProvideAssistDataListener
AppOpsManager.OnOpChangedListener
DatePickerDialog.OnDateSetListener
FragmentBreadCrumbs.OnBreadCrumbClickListener
FragmentManager.BackStackEntry
FragmentManager.OnBackStackChangedListener
KeyguardManager.OnKeyguardExitResult
LoaderManager.LoaderCallbacks
PendingIntent.OnFinished
SearchManager.OnCancelListener
SearchManager.OnDismissListener
TimePickerDialog.OnTimeSetListener
UiAutomation.AccessibilityEventFilter
UiAutomation.OnAccessibilityEventListener

Classes

ActionBar
ActionBar.LayoutParams
ActionBar.Tab
Activity
ActivityGroup
ActivityManager

Parameters

Android APIs API level: 19

Returns

android
android.accessibilityservice
android.accounts
android.animation
android.app
android.app.admin
android.app.backup
android.appwidget
android.bluetooth
android.content
android.content.pm
android.content.res
android.database
android.database.sqlite

boolean Return true if this event was consumed.

public void **dump** (String prefix, FileDescriptor fd,

Added in API level 11

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

Parameters

prefix Desired prefix to prepend at each line of output.

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

writer The PrintWriter to which you should dump your state.

This will be closed for you after you return.

args additional arguments to the dump request.

Interfaces

public View **findViewById** (int id)

Added in API level 1

ActionBar.OnMenuVisibilityListener
ActionBar.OnNavigationListener
ActionBar.TabListener
Application.ActivityLifecycleCallbacks
Application.OnProvideAssistDataListener
AppOpsManager.OnOpChangedListener
DatePickerDialog.OnDateSetListener
FragmentBreadCrumbs.OnBreadCrumbClickListener
FragmentManager.BackStackEntry
FragmentManager.OnBackStackChangedListener
KeyguardManager.OnKeyguardExitResult
LoaderManager.LoaderCallbacks
PendingIntent.OnFinished
SearchManager.OnCancelListener
SearchManager.OnDismissListener
TimePickerDialog.OnTimeSetListener
UiAutomation.AccessibilityEventFilter
UiAutomation.OnAccessibilityEventListener

Finds a view that was identified by the id attribute from the XML that was processed in `onCreate(Bundle)` (/reference/android/app/Activity.html#onCreate(android.os.Bundle)).

Returns

The view if found or null otherwise.

public void **finish** ()

Added in API level 1

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

public void **finishActivity** (int requestCode)

Added in API level 1

Force finish another activity that you had previously started with `startActivityForResult(Intent, int)` (/reference/android/app/Activity.html#startActivityForResult(android.content.Intent, int)).

Classes

ActionBar
ActionBar.LayoutParams
ActionBar.Tab
Activity
ActivityGroup
ActivityManager

Parameters

requestCode The request code of the activity that you had started. There are multiple activities started with this request

code. they will all be finished.

Android APIs API level: 19

public void **finishActivityFromChild** (Activity child, int requestCode) Added in API level 1

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

Parameters

child The activity making the call.

requestCode Request code that had been used to start the activity.

public void **finishAffinity** () Added in API level 16

Finish this activity as well as all activities immediately below it in the current task that have the same affinity. This is typically used when an application can be launched on to another task (such as from an ACTION_VIEW of a content type it understands) and the user has used the up navigation 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.

public void **finishFromChild** (Activity child) Added in API level 1

This is called when a child activity of this one calls its `finish()` method. The default implementation simply calls finish() on this activity (the parent), finishing the entire group.

Parameters

child The activity making the call.

See Also

`finish()`

Classes

public ActionBar **getActionBar** () Added in API level 11

Retrieve a reference to this activity's ActionBar.

Returns

The Activity's ActionBar, or null if it does not have one.

Interfaces

ActionBar.OnMenuVisibilityListener
ActionBar.OnNavigationListener
ActionBar.TabListener
Application.ActivityLifecycleCallbacks
Application.OnProvideAssistDataListener
AppOpsManager.OnOpChangedListener
DatePickerDialog.OnDateSetListener
FragmentBreadCrumbs.OnBreadCrumbClickListener
FragmentManager.OnBackStackChangedListener
FragmentManager.OnBackStackChangedListener
KeyguardManager.OnKeyguardExitResult
LoaderManager.LoaderCallbacks
PendingIntent.OnFinished
SearchManager.OnCancelListener
SearchManager.OnDismissListener
TimePickerDialog.OnTimeSetListener
UiAutomation.AccessibilityEventFilter
UiAutomation.OnAccessibilityEventListener

Classes

ActionBar
ActionBar.LayoutParams
ActionBar.Tab
Activity
ActivityGroup
ActivityManager

public final Application **getApplication** () Added in API level 1

Android APIs API level: 19

android
android.accessibilityservice
android.accounts
android.animation
android.app
android.app.admin
android.app.backup
android.appwidget
android.bluetooth
android.content
android.content.pm
android.content.res
android.database
android.database.sqlite

Interfaces

ActionBar.OnMenuVisibilityListener
ActionBar.OnNavigationListener
ActionBar.TabListener
Application.ActivityLifecycleCallbacks
Application.OnProvideAssistDataListener
AppOpsManager.OnOpChangedListener
DatePickerDialog.OnDateSetListener
FragmentBreadCrumbs.OnBreadCrumbClickListener
FragmentManager.BackStackEntry
FragmentManager.OnBackStackChangedListener
KeyguardManager.OnKeyguardExitResult
LoaderManager.LoaderCallbacks
PendingIntent.OnFinished
SearchManager.OnCancelListener
SearchManager.OnDismissListener
TimePickerDialog.OnTimeSetListener
UiAutomation.AccessibilityEventFilter
UiAutomation.OnAccessibilityEventListener

Classes

ActionBar
ActionBar.LayoutParams
ActionBar.Tab
Activity
ActivityGroup
ActivityManager

public ComponentName **getCallingActivity** () Added in API level 1

Return the name of the activity that invoked this activity. This is who the data in `setResult()` (/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)` (/reference/android/app/Activity.html#startActivityForResult(android.content.Intent, int)) form that includes a request code), then the calling package will be null.

Returns

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

public String **getCallingPackage** () Added in API level 1

Return the name of the package that invoked this activity. This is who the data in `setResult()` (/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)` (/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` (/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()` (/reference/android/app/Activity.html#getCallingActivity()) and retrieve the package name from that instead.

Returns

The package of the activity that will receive your reply, or null if

Android APIs API level: 19

public int **getChangingConfigurations** () Added in API level 1

If this activity is being destroyed because it can not handle a configuration parameter being changed (and thus its reference (android/app/Activity.html#onConfigurationChanged(android.content.res.Configur 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

Returns a bit field of the configuration parameters that are changing, as defined by the `Configuration` class.

Interfaces

public ComponentName **getComponentName** () Added in API level 1

Returns complete component name of this activity.

Returns

Returns the complete component name for this activity

public View **getCurrentFocus** () Added in API level 1

Calls `getCurrentFocus()` (/reference/android/view/Window.html#getCurrentFocus()) on the Window of this Activity to return the currently focused view.

Returns

View The current View with focus or null.

See Also

getWindow()

getCurrentFocus()

public FragmentManager **getFragmentManager** () Added in API level 11

Classes

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

public Intent **getIntent** () Added in API level 1

Return the intent that started this activity.

Android APIs API level: 19

public Object **getLastNonConfigurationInstance** () Added in API level 1

This method was deprecated in API level 13.
Use the new `Fragment` (/reference/android/app/Fragment.html) API `setRetainInstance(boolean)` (/reference/android/app/Activity.html#onRetainNonConfigurationInstance()); this is also available on older platforms through the Android compatibility package.

Retrieve the non-configuration instance data that was previously returned by `onRetainNonConfigurationInstance()` (/reference/android/app/Activity.html#onRetainNonConfigurationInstance()). This will be available from the initial `onCreate(Bundle)` (/reference/android/app/Activity.html#onCreate(android.os.Bundle)) and `onStart()` (/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)` (/reference/android/app/Activity.html#onSaveInstanceState(android.os.Bundle)) mechanism) even if this function returns null.

Returns
Returns the object previously returned by `onRetainNonConfigurationInstance()`.

public LayoutInflater **getLayoutInflater** () Added in API level 1

Convenience for calling `getLayoutInflater()` (/reference/android/view/Window.html#getLayoutInflater()).

public LoaderManager **getLoaderManager** () Added in API level 11

Return the LoaderManager for this fragment, creating it if needed.

public String **getLocalClassName** () Added in API level 1

prefix removed. This is the default name used to read and write settings.

Interfaces

ActionBar.OnMenuVisibilityListener
ActionBar.OnNavigationListener
ActionBar.TabListener
Application.ActivityLifecycleCallbacks
Application.OnProvideAssistDataListener
AppOpsManager.OnOpChangedListener
DatePickerDialog.OnDateSetListener
FragmentBreadCrumbs.OnBreadCrumbClickListener
FragmentManager.BackStackEntry
FragmentManager.OnBackStackChangedListener
KeyguardManager.OnKeyguardExitResult
LoaderManager.LoaderCallbacks
PendingIntent.OnFinished
SearchManager.OnCancelListener
SearchManager.OnDismissListener
TimePickerDialog.OnTimeSetListener
UiAutomation.AccessibilityEventFilter
UiAutomation.OnAccessibilityEventListener

Classes

ActionBar
ActionBar.LayoutParams
ActionBar.Tab
Activity
ActivityGroup
ActivityManager

Returns

Android APIs API level: 19

android
android.accessibilityservice
android.accounts
android.animation
android.app
android.app.admin
android.app.backup
android.appwidget
android.bluetooth
android.content
android.content.pm
android.content.res
android.database
android.database.sqlite

public MenuInflater **getMenuInflater** () Added in API level 1

Returns a `MenuInflater` (/reference/android/view/MenuInflater.html) with this context.

public final Activity **getParent** () Added in API level 1

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

public Intent **getParentActivityIntent** () Added in API level 16

Obtain an `Intent` (/reference/android/content/Intent.html) that will launch an explicit target activity specified by this activity's logical parent as declared in application's manifest by the `parentActivityName` (/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

a new Intent targeting the defined parent of this activity or null if there is no valid parent.

public SharedPreferences **getPreferences** (int mode) Added in API level 1

Retrieve a `SharedPreferences` (/reference/android/content/SharedPreferences.html) for accessing preferences that are private to this activity. This simply calls the underlying `getSharedPreferences(String, int)` (/reference/android/content/ContextWrapper.html#getSharedPreferences(java.lang.String, int)) method by passing in this activity's class name as the preferences name.

Parameters

mode Operating mode. Use `MODE_PRIVATE` for the default operation, `MODE_WORLD_READABLE` and `MODE_WORLD_WRITEABLE` to control permissions.

Returns

a `SharedPreferences` (/reference/android/content/SharedPreferences.html) through which you can retrieve and modify the preference values.

Interfaces

ActionBar.OnMenuVisibilityListener
ActionBar.OnNavigationListener
ActionBar.TabListener
Application.ActivityLifecycleCallbacks
Application.OnProvideAssistDataListener
AppOpsManager.OnOpChangedListener
DatePickerDialog.OnDateSetListener
FragmentBreadCrumbs.OnBreadCrumbClickListener
FragmentManager.BackStackEntry
FragmentManager.OnBackStackChangedListener
KeyguardManager.OnKeyguardExitResult
LoaderManager.LoaderCallbacks
PendingIntent.OnFinished
SearchManager.OnCancelListener
SearchManager.OnDismissListener
TimePickerDialog.OnTimeSetListener
UiAutomation.AccessibilityEventFilter
UiAutomation.OnAccessibilityEventListener

Classes

ActionBar
ActionBar.LayoutParams
ActionBar.Tab
Activity
ActivityGroup
ActivityManager

public int **getRequestedOrientation** () Added in API level 1

Android APIs API level: 19

y. This will

either be the orientation requested in its component's manifest, or the last requested orientation given to `setRequestedOrientation(int)` (/reference/android/app/Activity.html#setRequestedOrientation(int)).

Returns

Returns an orientation constant as used in `ActivityInfo.screenOrientation`.

public Object **getSystemService** (String name) Added in API level 1

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:

`WINDOW_SERVICE` ("window")
The top-level window manager in which you can place custom windows. The returned object is a `WindowManager`.

`LAYOUT_INFLATER_SERVICE` ("layout_inflater")
A `LayoutInflater` for inflating layout resources in this context.

`ACTIVITY_SERVICE` ("activity")
An `ActivityManager` for interacting with the global activity state of the system.

`POWER_SERVICE` ("power")
A `PowerManager` for controlling power management.

`ALARM_SERVICE` ("alarm")
A `AlarmManager` for receiving intents at the time of your choosing.

`NOTIFICATION_SERVICE` ("notification")
A `NotificationManager` for informing the user of background events.

`KEYGUARD_SERVICE` ("keyguard")
A `KeyguardManager` for controlling keyguard.

`LOCATION_SERVICE` ("location")
A `LocationManager` for controlling location (e.g., GPS) updates.

`SEARCH_SERVICE` ("search")
A `SearchManager` for handling search.

`VIBRATOR_SERVICE` ("vibrator")
A `Vibrator` for interacting with the vibrator hardware.

`CONNECTIVITY_SERVICE` ("connection")
A `ConnectivityManager` for handling management of

Interfaces

ActionBar.OnMenuVisibilityListener
ActionBar.OnNavigationListener
ActionBar.TabListener
Application.ActivityLifecycleCallbacks
Application.OnProvideAssistDataListener
AppOpsManager.OnOpChangedListener
DatePickerDialog.OnDateSetListener
FragmentBreadCrumbs.OnBreadCrumbClickListener
FragmentManager.BackStackEntry
FragmentManager.OnBackStackChangedListener
KeyguardManager.OnKeyguardExitResult
LoaderManager.LoaderCallbacks
PendingIntent.OnFinished
SearchManager.OnCancelListener
SearchManager.OnDismissListener
TimePickerDialog.OnTimeSetListener
UiAutomation.AccessibilityEventFilter
UiAutomation.OnAccessibilityEventListener

Classes

ActionBar
ActionBar.LayoutParams
ActionBarTab
Activity
ActivityGroup
ActivityManager

network connections.

Android APIs API level: 19

ctivity.

INPUT_METHOD_SERVICE ("input_method")

An InputMethodManager for management of i nput methods.

UI_MODE_SERVICE ("uimode")

An UiModeManager for controlling UI modes.

An DownloadManager for requesting HTTP dowr loads

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

Parameters

The name of the desired service

Returns

The service or null if the name does not exist.

Interfaces

ActionBar.OnMenuVisibilityListener
ActionBar.OnNavigationListener
ActionBar.TabListener
Application.ActivityLifecycleCallbacks
Application.OnProvideAssistDataListener
AppOpsManager.OnOpChangedListener
DatePickerDialog.OnDateSetListener
FragmentBreadCrumbs.OnBreadCrumbClickListener
FragmentManager.BackStackEntry
FragmentManager.OnBackStackChangedListener
KeyguardManager.OnKeyguardExitResult
LoaderManager.LoaderCallbacks
PendingIntent.OnFinished
SearchManager.OnCancelListener
SearchManager.OnDismissListener
TimePickerDialog.OnTimeSetListener
UiAutomation.AccessibilityEventFilter
UiAutomation.OnAccessibilityEventListener

Classes

ActionBar
ActionBar.LayoutParams
ActionBar.Tab
Activity
ActivityGroup
ActivityManager

public int **getTaskId** () Added in API level 1

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

Returns

Task identifier, an opaque integer.

public final CharSequence **getTitle** () Added in API level 1

public final int **getTitleColor** () Added in API level 1

public final int **getVolumeControlStream** () Added in API level 1

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

Returns

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

See Also

setVolumeControlStream(int)

public Window **getWindow** () Added in API level 1

Retrieve the current `Window` (/reference/android/view/Window.html) for the Window

Android APIs API level: 19

Returns

Window The current window, or null if the activity is not visual.

Added in API level 1

Retrieve the window manager for showing custom windows.

public boolean hasWindowFocus ()

Added in API level 3

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

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

See Also

- `onWindowAttributesChanged(android.view.WindowManager.LayoutParams)`

public void invalidateOptionsMenu ()

Added in API level 11

Declare that the options menu has changed, so should be recreated. The `onCreateOptionsMenu(Menu)` (/reference/android/app/Activity.html#onCreateOptionsMenu(android.view.Menu)) method will be called the next time it needs to be displayed.

public boolean isChangingConfigurations ()

Added in API level 11

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()` (/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()` (/reference/android/app/Activity.html#onRetainNonConfigurationInstance()).

Returns

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

public final boolean isChild ()

Added in API level 1

Interfaces

- ActionBar.OnMenuVisibilityListener
- ActionBar.OnNavigationListener
- ActionBar.TabListener
- Application.ActivityLifecycleCallbacks
- Application.OnProvideAssistDataListener
- AppOpsManager.OnOpChangedListener
- DatePickerDialog.OnDateSetListener
- FragmentBreadCrumbs.OnBreadCrumbClickListener
- FragmentManager.BackStackEntry
- FragmentManager.OnBackStackChangedListener
- KeyguardManager.OnKeyguardExitResult
- LoaderManager.LoaderCallbacks
- PendingIntent.OnFinished
- SearchManager.OnCancelListener
- SearchManager.OnDismissListener
- TimePickerDialog.OnTimeSetListener
- UiAutomation.AccessibilityEventFilter
- UiAutomation.OnAccessibilityEventListener

Classes

- ActionBar
- ActionBar.LayoutParams
- ActionBar.Tab
- Activity
- ActivityGroup
- ActivityManager

public boolean **isDestroyed** () Added in API level 17

Android APIs API level: 19

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()](/reference/android/app/Activity.html#onStop()) to determine whether the state needs to be cleaned up or will be recreated when the activity is started again.

Returns

true if this activity is in the process of being destroyed and recreated with a new configuration

See Also

- [onStop()](/reference/android/app/Activity.html#onStop())
- [getChangingConfigurations()](/reference/android/app/Activity.html#getChangingConfigurations())
- [getConfigChanges(int)](/reference/android/content/pm/ActivityInfo.html#getConfigChanges(int))

public boolean **isFinishing** () Added in API level 1

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

Returns

Returns true if the activity is in the process of finishing, else false.

See Also

- [finish()](/reference/android/app/Activity.html#finish())

public boolean **isImmersive** () Added in API level 18

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](/reference/android/R.attr.html#immersive) but may be changed at runtime by [setImmersive(boolean)](/reference/android/app/Activity.html#setImmersive(boolean)).

See Also

- [setImmersive(boolean)](/reference/android/app/Activity.html#setImmersive(boolean))
- [FLAG_IMMERSIVE](/reference/android/app/ActivityManager.RecentTaskInfo.html#FLAG_IMMERSIVE)

public boolean **isTaskRoot** () Added in API level 1

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

Returns

True if this is the root activity, else false.

public final Cursor **managedQuery** (Uri uri, String[] projection, String selection, String[] selectionArgs, String sortOrder) Added in API level 1

Use [CursorLoader](/reference/android/content/CursorLoader.html) (/reference/android/content/CursorLoader.html)

Interfaces

ActionBar.OnMenuVisibilityListener
ActionBar.OnNavigationListener
ActionBar.TabListener
Application.ActivityLifecycleCallbacks
Application.OnProvideAssistDataListener
AppOpsManager.OnOpChangedListener
DatePickerDialog.OnDateSetListener
FragmentBreadCrumbs.OnBreadCrumbClickListener
FragmentManager.BackStackEntry
FragmentManager.OnBackStackChangedListener
KeyguardManager.OnKeyguardExitResult
LoaderManager.LoaderCallbacks
PendingIntent.OnFinished
SearchManager.OnCancelListener
SearchManager.OnDismissListener
TimePickerDialog.OnTimeSetListener
UiAutomation.AccessibilityEventFilter
UiAutomation.OnAccessibilityEventListener

Classes

ActionBar
ActionBar.LayoutParams
ActionBar.Tab
Activity
ActivityGroup
ActivityManager

instead.

Android APIs API level: 19

android
android.accessibilityservice
android.accounts
android.animation
android.app
android.app.admin
android.app.backup
android.appwidget
android.bluetooth
android.content
android.content.pm
android.content.res
android.database
android.database.sqlite

Interfaces

ActionBar.OnMenuVisibilityListener
ActionBar.OnNavigationListener
ActionBar.TabListener
Application.ActivityLifecycleCallbacks
Application.OnProvideAssistDataListener
AppOpsManager.OnOpChangedListener
DatePickerDialog.OnDateSetListener
FragmentBreadCrumbs.OnBreadCrumbClickListener
FragmentManager.BackStackEntry
FragmentManager.OnBackStackChangedListener
KeyguardManager.OnKeyguardExitResult
LoaderManager.LoaderCallbacks
PendingIntent.OnFinished
SearchManager.OnCancelListener
SearchManager.OnDismissListener
TimePickerDialog.OnTimeSetListener
UiAutomation.AccessibilityEventFilter
UiAutomation.OnAccessibilityEventListener

Classes

ActionBar
ActionBar.LayoutParams
ActionBar.Tab
Activity
ActivityGroup
ActivityManager

ng[],
String, String[], String) (/reference/androi /content
/ContentResolver.html#query(android.net.Uri, java.lang .String[],
java.lang.String, java.lang.String[], java.lang.Strinc)) that gives the
resulting Cursor (/reference/android/database/Cursor .html) to call
oid
/app/Activity.html#startManagingCursor(android.databas e.Cursor)) so
that the activity will manage its lifecycle for you. *If yo u are targeting*
HONEYCOMB (/reference/android
/os/Build.VERSION_CODES.html#HONEYCOMB) or later, consid er instead using
LoaderManager (/reference/android/app/LoaderManag er.html) instead,
available via getLoaderManager() (/reference/andr oid
/app/Activity.html#getLoaderManager()).

Warning: Do not call close() (/reference/android/da tabase
/Cursor.html#close()) on a cursor obtained using this method,
because the activity will do that for you at the appro riate time.
However, if you call stopManagingCursor(Curso) (/reference
/android
/app/Activity.html#stopManagingCursor(android.database .Cursor)) on a
cursor from a managed query, the system *will not* aut matically
close the cursor and, in that case, you must call clo se()
(/reference/android/database/Cursor.html#close()).

Parameters

uri	The URI of the content provider to query.
projection	List of columns to return.
selection	SQL WHERE clause.
selectionArgs	The arguments to selection, if any ?s are present
sortOrder	SQL ORDER BY clause.

Returns

The Cursor that was returned by query().

See Also

query(android.net.Uri, String[], String, String[], String)
startManagingCursor(Cursor)

public boolean **moveTaskToBack** (boolean nonRo) added in API level 1

Move the task containing this activity to the back of the activity

stack. The activity's order within the task is unchanged.

Android APIs API level: 19

nonRoot 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.

Returns

If the task was moved (or it was already at the back) true is returned, else false.

public boolean **navigateUpTo** (Intent upIntent) Added in API level 16

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)` (/reference/android/app/Activity.html#shouldUpRecreateTask(android.content.Intent)).

Parameters

upIntent An Intent representing the target destination for up navigation

Returns

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

public boolean **navigateUpToFromChild** (Activity child, Intent upIntent) Added in API level 16

This is called when a child activity of this one calls its `navigateUpTo(Intent)` (/reference/android/app/Activity.html#navigateUpTo(android.content.Intent)) method. The default implementation simply calls navigateUpTo(upIntent) on this

Interfaces

ActionBar.OnMenuVisibilityListener
ActionBar.OnNavigationListener
ActionBar.TabListener
Application.ActivityLifecycleCallbacks
Application.OnProvideAssistDataListener
AppOpsManager.OnOpChangedListener
DatePickerDialog.OnDateSetListener
FragmentBreadCrumbs.OnBreadCrumbClickListener
FragmentManager.BackStackEntry
FragmentManager.OnBackStackChangedListener
KeyguardManager.OnKeyguardExitResult
LoaderManager.LoaderCallbacks
PendingIntent.OnFinished
SearchManager.OnCancelListener
SearchManager.OnDismissListener
TimePickerDialog.OnTimeSetListener
UiAutomation.AccessibilityEventFilter
UiAutomation.OnAccessibilityEventListener

Classes

ActionBar
ActionBar.LayoutParams
ActionBar.Tab
Activity
ActivityGroup
ActivityManager

activity (the parent).

Android APIs API level: 19

android
android.accessibilityservice
android.accounts
android.animation
android.app
android.app.admin
android.app.backup
android.appwidget
android.bluetooth
android.content
android.content.pm
android.content.res
android.database
android.database.sqlite

child The activity making the call.
upIntent An intent representing the target destination for up navigation

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

public void **onActionModeFinished** (ActionMode mode) Added in API level 11

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

Parameters

mode The action mode that just finished.

public void **onActionModeStarted** (ActionMode mode) Added in API level 11

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

Parameters

mode The new action mode.

public void **onAttachFragment** (Fragment fragment) Added in API level 11

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

public void **onAttachedToWindow** () Added in API level 5

Called when the main window associated with the activity has been `View.onAttachedToWindow()` (/reference/androi

Interfaces

ActionBar.OnMenuVisibilityListener
ActionBar.OnNavigationListener
ActionBar.TabListener
Application.ActivityLifecycleCallbacks
Application.OnProvideAssistDataListener
AppOpsManager.OnOpChangedListener
DatePickerDialog.OnDateSetListener
FragmentBreadCrumbs.OnBreadCrumbClickListener
FragmentManager.BackStackEntry
FragmentManager.OnBackStackChangedListener
KeyguardManager.OnKeyguardExitResult
LoaderManager.LoaderCallbacks
PendingIntent.OnFinished
SearchManager.OnCancelListener
SearchManager.OnDismissListener
TimePickerDialog.OnTimeSetListener
UiAutomation.AccessibilityEventFilter
UiAutomation.OnAccessibilityEventListener

Classes

ActionBar
ActionBar.LayoutParams
ActionBar.Tab
Activity
ActivityGroup
ActivityManager

/view/View.html#onAttachedToWindow()) for more information.

Android APIs API level: 19

onAttachedToWindow()

android
android.accessibilityservice
android.accounts
android.animation
android.app
android.app.admin
android.app.backup
android.appwidget
android.bluetooth
android.content
android.content.pm
android.content.res
android.database
android.database.sqlite

public void **onBackPressed** () Added in API level 5

s of the back key. The default implementation simply finishes the current activity, but you can override this to do whatever you want.

public void **onConfigurationChanged** (Configuration newConfig) Added in API level 1

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 (/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

newConfig The new device configuration.

public void **onContentChanged** () Added in API level 1

This hook is called whenever the content view of the screen changes (due to a call to Window.setContentView (/reference/android/view/Window.html#setContentView(android.view.View, android.view.ViewGroup.LayoutParams)) or Window.addContentView (/reference/android/view/Window.html#addContentView(android.view.View, android.view.ViewGroup.LayoutParams))).

public boolean **onContextItemSelected** (MenuItem item) Added in API level 1

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

Interfaces

ActionBar.OnMenuVisibilityListener
ActionBar.OnNavigationListener
ActionBar.TabListener
Application.ActivityLifecycleCallbacks
Application.OnProvideAssistDataListener
AppOpsManager.OnOpChangedListener
DatePickerDialog.OnDateSetListener
FragmentBreadCrumbs.OnBreadCrumbClickListener
FragmentManager.BackStackEntry
FragmentManager.OnBackStackChangedListener
KeyguardManager.OnKeyguardExitResult
LoaderManager.LoaderCallbacks
PendingIntent.OnFinished
SearchManager.OnCancelListener
SearchManager.OnDismissListener
TimePickerDialog.OnTimeSetListener
UiAutomation.AccessibilityEventFilter
UiAutomation.OnAccessibilityEventListener

Classes

ActionBar
ActionBar.LayoutParams
ActionBar.Tab
Activity
ActivityGroup
ActivityManager

any items for which you would like to do processing without those

Android APIs API level: 19

Use `getMenuInfo()` (/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

item The context menu item that was selected.

Returns

boolean Return false to allow normal context menu processing to proceed, true to consume it here.

public void **onContextMenuClosed** (Menu menu) Added in API level 1

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

Parameters

menu The context menu that is being closed.

public void **onCreateContextMenu** (ContextMenu menu, View v, ContextMenu.ContextMenuInfo menuInfo) Added in API level 1

Called when a context menu for the view is about to be shown. Unlike `onCreateOptionsMenu(Menu)` (/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` (/reference/android/widget/AdapterView.html) subclasses, this can be found in the `menuInfo`).

Use `onContextItemSelected(android.view.MenuItem)` (/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

Interfaces

ActionBar.OnMenuVisibilityListener
ActionBar.OnNavigationListener
ActionBar.TabListener
Application.ActivityLifecycleCallbacks
Application.OnProvideAssistDataListener
AppOpsManager.OnOpChangedListener
DatePickerDialog.OnDateSetListener
FragmentBreadCrumbs.OnBreadCrumbClickListener
FragmentManager.BackStackEntry
FragmentManager.OnBackStackChangedListener
KeyguardManager.OnKeyguardExitResult
LoaderManager.LoaderCallbacks
PendingIntent.OnFinished
SearchManager.OnCancelListener
SearchManager.OnDismissListener
TimePickerDialog.OnTimeSetListener
UiAutomation.AccessibilityEventFilter
UiAutomation.OnAccessibilityEventListener

Classes

ActionBar
ActionBar.LayoutParams
ActionBar.Tab
Activity
ActivityGroup
ActivityManager

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

Added in API level 1

Generate a new description for this activity. This method is called before pausing the activity and can, if desired, return some textual description of its current state to be displayed to the user.

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

A description of what the user is doing. It should be short and sweet (only a few words).

See Also

- `onBackCommitted()`
- `onBackInvoked()`
- `onBackPressed()`
- `onCreateDescription()`
- `onCreateNavigateUpTaskStack(TaskStackBuilder)`
- `onCreateThumbnail(Bitmap, Canvas)`
- `onCreateView(String, Context, AttributeSet)`
- `onDetachedFromWindow()`
- `onEnterAnimationComplete()`
- `onGenericMotionEvent(MotionEvent)`
- `onGetDirectActions(CancellationSignal, Consumer)`
- `onKeyDown(int, KeyEvent)`
- `onKeyLongPress(int, KeyEvent)`
- `onKeyMultiple(int, int, KeyEvent)`
- `onKeyShortcut(int, KeyEvent)`
- `onKeyUp(int, KeyEvent)`
- `onLocalVoiceInteractionStarted()`
- `onLocalVoiceInteractionStopped()`
- `onLowMemory()`
- `onMenuItemSelected(int, MenuItem)`
- `onMenuOpened(int, Menu)`
- `onMultiWindowModeChanged(boolean)`
- `onNavigateUp()`
- `onNavigateUpFromChild(Activity)`
- `onNewIntent(Intent)`
- `onOptionsItemSelected(MenuItem)`
- `onOptionsMenuClosed(Menu)`
- `onPanelClosed(int, Menu)`
- `onPause()`

public void **onCreateNavigateUpTaskStack**(TaskStackBuilder builder)

Added in API level 16

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` (/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()` (/reference/android/app/Activity.html#onNavigateUp()) if `shouldUpRecreateTask(Intent)` (/reference/android/app/Activity.html#shouldUpRecreateTask(android.content.Intent)) returns true when supplied with the intent returned by `getParentActivityIntent()` (/reference/androi

Interfaces

- ActionBar.OnMenuVisibilityListener
- ActionBar.OnNavigationListener
- ActionBar.TabListener
- Application.ActivityLifecycleCallbacks
- Application.OnProvideAssistDataListener
- AppOpsManager.OnOpChangedListener
- DatePickerDialog.OnDateSetListener
- FragmentBreadCrumbs.OnBreadCrumbClickListener
- FragmentManager.BackStackEntry
- FragmentManager.OnBackStackChangedListener
- KeyguardManager.OnKeyguardExitResult
- LoaderManager.LoaderCallbacks
- PendingIntent.OnFinished
- SearchManager.OnCancelListener
- SearchManager.OnDismissListener
- TimePickerDialog.OnTimeSetListener
- UiAutomation.AccessibilityEventFilter
- UiAutomation.OnAccessibilityEventListener

Classes

- ActionBar
- ActionBar.LayoutParams
- ActionBar.Tab
- Activity
- ActivityGroup
- ActivityManager

Applications that wish to supply extra Intent parameters to the

Android APIs API level: 19

uilder)

(/reference/android

(app/Activity.html#onPrepareNavigateUpTaskStack(android.app.TaskStackB

uilder))

builder An empty TaskStackBuilder - the application should add Intents representing the desired task stack

public boolean **onCreateOptionsMenu** (Menu menu) Added in API level 1

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

nu is displayed.

To update the menu every time it is displayed, see onPrepareOptionsMenu(Menu) (/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 (/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) (/reference/android/app/Activity.html#onOptionsItemSelected(android.view.MenuItem)) method to handle them there.

Parameters

menu The options menu in which you place your items.

Returns

You must return true for the menu to be displayed; if you return false it will not be shown.

See Also

onPrepareOptionsMenu(Menu)

onOptionsItemSelected(MenuItem)

Interfaces

ActionBar.OnMenuVisibilityListener
ActionBar.OnNavigationListener
ActionBar.TabListener
Application.ActivityLifecycleCallbacks
Application.OnProvideAssistDataListener
AppOpsManager.OnOpChangedListener
DatePickerDialog.OnDateSetListener
FragmentBreadCrumbs.OnBreadCrumbClickListener
FragmentManager.BackStackEntry
FragmentManager.OnBackStackChangedListener
KeyguardManager.OnKeyguardExitResult
LoaderManager.LoaderCallbacks
PendingIntent.OnFinished
SearchManager.OnCancelListener
SearchManager.OnDismissListener
TimePickerDialog.OnTimeSetListener
UiAutomation.AccessibilityEventFilter
UiAutomation.OnAccessibilityEventListener

Classes

ActionBar
ActionBar.LayoutParams
ActionBar.Tab
Activity
ActivityGroup
ActivityManager

public boolean **onCreatePanelMenu** (int featureId, Menu menu)

Added in API level 1

Default implementation of `onCreatePanelMenu(int, Menu)` for activities. This calls through to the new `onCreateOptionsMenu(Menu)` method for the `FEATURE_OPTIONS_PANEL` panel, so that subclasses of Activity don't need to deal with feature codes.

Parameters

featureId The panel being created.

menu The menu inside the panel.

Returns

boolean You must return true for the panel to be displayed; if you return false it will not be shown.

public View **onCreatePanelView** (int featureId)

Added in API level 1

Default implementation of `onCreatePanelView(int)` for activities. This simply returns null so that all panel sub-windows will have the default menu behavior.

Parameters

featureId Which panel is being created.

Returns

view The top-level view to place in the panel.

public boolean **onCreateThumbnail** (Bitmap outBitmap, Canvas canvas)

Added in API level 1

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

Interfaces

Classes

outBitmap The bitmap to contain the thumbnail.

Android APIs API level: 19

Returns

Return true if you have drawn into the bitmap; otherwise after you return it will be filled with a default thumbnail.

See Also

onCreateDescription()
onSaveInstanceState(Bundle)
onPause()

public View **onCreateView** (View parent, String name, Context context, AttributeSet attrs) Added in API level 11

Standard implementation of `onCreateView(View, String, Context, AttributeSet)` (/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(String)` (/reference/android/app/Activity.html#getSystemService(java.lang.String)). This implementation handles tags to embed fragments inside of the activity.

Parameters

parent The parent that the created view will be placed in; *note that this may be null*.

name Tag name to be inflated.

context The context the view is being created in.

attrs Inflation attributes as specified in XML file.

Returns

View Newly created view. Return null for the default behavior.

See Also

createView(String, String, AttributeSet)
getLayoutInflater()

public View **onCreateView** (String name, Context context, AttributeSet attrs) Added in API level 1

Standard implementation of `onCreateView(String, Context, AttributeSet)` (/reference/android/app/Activity.html#onCreateView(java.lang.String, android.content.Context, android.util.AttributeSet)) used when

Interfaces

ActionBar.OnMenuVisibilityListener
ActionBar.OnNavigationListener
ActionBar.TabListener
Application.ActivityLifecycleCallbacks
Application.OnProvideAssistDataListener
AppOpsManager.OnOpChangedListener
DatePickerDialog.OnDateSetListener
FragmentBreadCrumbs.OnBreadCrumbClickListener
FragmentManager.BackStackEntry
FragmentManager.OnBackStackChangedListener
KeyguardManager.OnKeyguardExitResult
LoaderManager.LoaderCallbacks
PendingIntent.OnFinished
SearchManager.OnCancelListener
SearchManager.OnDismissListener
TimePickerDialog.OnTimeSetListener
UiAutomation.AccessibilityEventFilter
UiAutomation.OnAccessibilityEventListener

Classes

ActionBar
ActionBar.LayoutParams
ActionBar.Tab
Activity
ActivityGroup
ActivityManager

inflating with the LayoutInflater returned by

Android APIs API level: 19

This implementation does nothing and is for pre-HONEYCOMB (/reference/android/os/Build.VERSION_CODES.html#HONEYCOMB) apps. Newer apps should use onCreateView(View, String, Context, AttributeSet) (/reference/android/app/Activity.html#onCreateView(android.view.View, java.lang.String, android.content.Context, android.util.AttributeSet)).

Parameters

name Tag name to be inflated.

context The context the view is being created in.

attrs Inflation attributes as specified in XML file.

Returns

View Newly created view. Return null for the default behavior.

See Also

- createView(String, String, AttributeSet)
- getLayoutInflater()

public void **onDetachedFromWindow** () Added in API level 5

Called when the main window associated with the activity has been detached from the window manager. See View.onDetachedFromWindow() (/reference/android/view/View.html#onDetachedFromWindow()) for more information.

See Also

- onDetachedFromWindow()

public boolean **onGenericMotionEvent** (MotionEvent event) Added in API level 12

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 (/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

Interfaces

- ActionBar.OnMenuVisibilityListener
- ActionBar.OnNavigationListener
- ActionBar.TabListener
- Application.ActivityLifecycleCallbacks
- Application.OnProvideAssistDataListener
- ApplicationErrorReport.CrashInfo.UntrustedStateModuleApi
- DatePickerDialog.OnDateSetListener
- FragmentBreadCrumbs.OnBreadCrumbClickListener
- FragmentManager.BackStackEntry
- FragmentManager.OnBackStackChangedListener
- KeyguardManager.OnKeyguardExitResult
- LoaderManager.LoaderCallbacks
- PendingIntent.OnFinished
- SearchManager.OnCancelListener
- SearchManager.OnDismissListener
- TimePickerDialog.OnTimeSetListener
- UiAutomation.AccessibilityEventFilter
- UiAutomation.OnAccessibilityEventListener

Classes

- ActionBar
- ActionBar.LayoutParams
- ActionBar.Tab
- Activity
- ActivityGroup
- ActivityManager

Android APIs API level: 19

android
android.accessibilityservice
android.accounts
android.animation
android.app
android.app.admin
android.app.backup
android.appwidget
android.bluetooth
android.content
android.content.pm
android.content.res
android.database
android.database.sqlite
android.drm
android.gesture
android.graphics
android.graphics.drawable
android.graphics.drawable.shapes
android.graphics.pdf
android.hardware
android.hardware.camera2
android.hardware.camera2.params
android.hardware.display
android.hardware.fingerprint
android.hardware.input
android.hardware.usb
android.location
android.media
android.media.audiofx
android.media.browse
android.media.effect
android.media.midi
android.media.projection
android.media.session
android.media.tv
android.net
android.net.http
android.net.nsd
android.net.rtp
android.net.sip
android.net.wifi
android.net.wifi.p2p
android.net.wifi.p2p.nsd
android.nfc
android.nfc.cardemulation
android.nfc.tech
android.opengl
android.os
android.os.storage
android.preference
android.print
android.print.pdf
android.printservice
android.provider
android.renderscript
android.sax
android.security
android.security.keystore
android.service.carrier
android.service.chooser
android.service.dreams
android.service.gatt
android.service.media
android.service.notification
android.service.restrictions
android.service.textservice
android.service.voice
android.service.wallpaper
android.speech
android.speech.tts
android.support.v4.accessibilityservice
android.support.v4.app
android.support.v4.content
android.support.v4.content.pm
android.support.v4.content.res
android.support.v4.database
android.support.v4.database.sqlite
android.support.v4.graphics
android.support.v4.graphics.drawable
android.support.v4.hardware.display
android.support.v4.hardware.fingerprint
android.support.v4.media
android.support.v4.media.session
android.support.v4.net
android.support.v4.os
android.support.v4.print
android.support.v4.provider
android.support.v4.text
android.support.v4.text.util
android.support.v4.util
android.support.v4.view
android.support.v4.view.accessibility
android.support.v4.view.animation
android.support.v4.widget
android.support.v7.app
android.support.v7.appcompat
android.support.v7.graphics
android.support.v7.graphics.drawable
android.support.v7.media
android.support.v7.mediarouter
android.support.v7.mediarouter.app
android.support.v7.mediarouter.media
android.support.v7.palette
android.support.v7.preference
android.support.v7.recyclerview
android.support.v7.util
android.support.v7.view
android.support.v7.widget
android.support.wear.view
android.support.wear.widget
android.support.wear.widget.drawer
android.system
android.telephony
android.telephony.cdma
android.telephony.gsm
android.test
android.test.mock
android.test.suitebuilder
android.test.suitebuilder.annotation
android.text
android.text.format
android.text.method
android.text.style
android.text.util
android.transition
android.util
android.view
android.view.accessibility
android.view.animation
android.view.inputmethod
android.view.textservice
android.webkit
android.widget
dalvik.annotation
dalvik.bytecode
dalvik.system
java.awt.font
java.beans
java.io
java.lang
java.lang.annotation
java.lang.ref
java.lang.reflect
java.math
java.net
java.nio
java.nio.channels
java.nio.channels.spi
java.nio.charset
java.nio.charset.spi
java.security
java.security.acl
java.security.cert
java.security.interfaces
java.security.spec
java.sql
java.text
java.text.spi
java.util
java.util.concurrent
java.util.concurrent.atomic
java.util.concurrent.locks
java.util.jar
java.util.logging
java.util.prefs
java.util.regex
java.util.zip
javax.crypto
javax.crypto.interfaces
javax.crypto.spec
javax.microedition.khronos.egl
javax.microedition.khronos.opengles
javax.net
javax.net.ssl
javax.security.auth
javax.security.auth.callback
javax.security.auth.login
javax.security.auth.spi
javax.security.auth.x500
javax.security.cert
javax.sql
javax.xml.parsers
javax.xml.transform
javax.xml.transform.dom
javax.xml.transform.sax
javax.xml.transform.stream
javax.xml.validation
javax.xml.xpath
junit.framework
junit.runner
org.apache.http.conn.scheme
org.apache.http.conn.ssl
org.apache.http.cookie
org.apache.http.entity
org.apache.http.impl.conn.tsccm
org.apache.http.message
org.apache.http.params
org.apache.http.protocol
org.apache.http.util
org.json
org.w3c.dom
org.w3c.dom.ls
org.xml.sax
org.xml.sax.ext
org.xml.sax.helpers
org.xmlpull.v1
org.xmlpull.v1.sax2

Generic motion events with source class SOURCE_CLASS_POINTER (/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) (/reference/android/view/View.html#onGenericMotionEvent(android.view.MotionEvent)) for an

Parameters

event The generic motion event being processed.

Returns

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

onKeyDown (int keyCode, KeyEvent event)

Added in API level 1

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 (/reference/android/view/KeyEvent.html#KEYCODE_BACK) by calling onBackPressed() (/reference/android/app/Activity.html#onBackPressed()). Though the behavior varies based on the application compatibility mode: for ECLAIR (/reference/android/os/Build.VERSION_CODES.html#ECLAIR) or later applications, it will set up the dispatch to call onKeyUp(int, KeyEvent) (/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) (/reference/android/app/Activity.html#setDefaultKeyMode(int)).

Parameters

keyCode The value in event.getKeyCode().

event Description of the key event.

Interfaces

ActionBar.OnMenuVisibilityListener
ActionBar.OnNavigationListener
ActionBar.TabListener
Application.ActivityLifecycleCallbacks
Application.OnProvideAssistDataListener
AppOpsManager.OnOpChangedListener
DatePickerDialog.OnDateSetListener
FragmentBreadCrumbs.OnBreadCrumbClickListener
FragmentManager.BackStackEntry
FragmentManager.OnBackStackChangedListener
KeyguardManager.OnKeyguardExitResult
LoaderManager.LoaderCallbacks
PendingIntent.OnFinished
SearchManager.OnCancelListener
SearchManager.OnDismissListener
TimePickerDialog.OnTimeSetListener
UiAutomation.AccessibilityEventFilter
UiAutomation.OnAccessibilityEventListener

Classes

ActionBar
ActionBar.LayoutParams
ActionBar.Tab
Activity
ActivityGroup
ActivityManager

Returns

Android APIs API level: 19 ated further,

v of false to indicate that you have not handled this event and it should continue to be propagated.

android

See Also

android.accessibilityservice

android.accounts

`onKeyUp(int, KeyEvent)`

android.animation

android.app

android.app.admin

public boolean **onKeyLongPress** (int keyCode, KeyEvent event)

android.appwidget

Added in API level 5

android.bluetooth

android.content

Default implementation of `KeyEvent.Callback.onKeyLongPress()` (/reference/android/view/KeyEvent.Callback.html#onKeyLongPress(int, android.view.KeyEvent)): always returns false (doesn't handle the

android.content.pm

android.content.res

android.database

android.database.sqlite

android.drm

Parameters

Interfaces

keyCode The value in event.getKeyCode().

event Description of the key event.

ActionBar.OnMenuVisibilityListener

ActionBar.OnNavigationListener

Returns

ActionBar.TabListener

If you handled the event, return true. If you want to allow the event to be handled by the next receiver, return false.

Application.ActivityLifecycleCallbacks

Application.OnProvideAssistDataListener

AppOpsManager.OnOpChangedListener

public boolean **onKeyMultiple** (int keyCode, int repeatCount, KeyEvent event)

DatePickerDialog.OnDateSetListener

FragmentBreadCrumbs.OnBreadCrumbClickListener

Added in API level 1

FragmentManager.BackStackEntry

FragmentManager.OnBackStackChangedListener

Default implementation of `KeyEvent.Callback.onKeyMultiple()` (/reference/android/view/KeyEvent.Callback.html#onKeyMultiple(int, int, android.view.KeyEvent)): always returns false (doesn't handle the

KeyguardManager.OnKeyguardExitResult

LoaderManager.LoaderCallbacks

PendingIntent.OnFinished

event)

SearchManager.OnCancelListener

SearchManager.OnDismissListener

Parameters

TimePickerDialog.OnTimeSetListener

UiAutomation.AccessibilityEventFilter

keyCode The value in event.getKeyCode().

UiAutomation.OnAccessibilityEventListener

repeatCount Number of pairs as returned by event.getRepeatCount().

event Description of the key event.

Classes

Returns

ActionBar

If you handled the event, return true. If you want to allow the event to be handled by the next receiver, return false.

ActionBar.LayoutParams

ActionBar.Tab

Activity

ActivityGroup

public boolean **onKeyShortcut** (int keyCode,

ActivityManager

KeyEvent event)

Added in API level 11

Android APIs API level: 19

ny of the views in the Activity. Override this method to implement glc bal key shortcuts for the Activity. Key shortcuts can also be implemented by setting the `shortcut` (`reference/android /view/MenuItem.html#setShortcut(char, char)`) property of menu items.

keyCode The value in event.getKeyCode().

event Description of the key event.

Returns

True if the key shortcut was handled.

public boolean **onKeyUp** (int keyCode, KeyEvent

Added in API level 1

Called when a key was released and not handled by a ny of the views inside of the activity. So, for example, key presses wh ile the cursor is inside a TextView will not trigger the event (unless it s a navigation to another object) because a TextView handles its own key presses.

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

Parameters

keyCode The value in event.getKeyCode().

event Description of the key event.

Returns

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

onKeyDown(int, KeyEvent)

KeyEvent

public void **onLowMemory** ()

Added in API level 1

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 defi ned, generally it will happen when all background process have been killed. That is, before reaching the point of killing processes hosting service and

Interfaces

ActionBar.OnMenuVisibilityListener
ActionBar.OnNavigationListener
ActionBar.TabListener
Application.ActivityLifecycleCallbacks
Application.OnProvideAssistDataListener
AppOpsManager.OnOpChangedListener
DatePickerDialog.OnDateSetListener
FragmentBreadCrumbs.OnBreadCrumbClickListener
FragmentManager.BackStackEntry
FragmentManager.OnBackStackChangedListener
KeyguardManager.OnKeyguardExitResult
LoaderManager.LoaderCallbacks
PendingIntent.OnFinished
SearchManager.OnCancelListener
SearchManager.OnDismissListener
TimePickerDialog.OnTimeSetListener
UiAutomation.AccessibilityEventFilter
UiAutomation.OnAccessibilityEventListener

Classes

ActionBar
ActionBar.LayoutParams
ActionBar.Tab
Activity
ActivityGroup
ActivityManager

android
android.accessibilityservice
android.accounts
android.animation
android.app
android.app.admin
android.app.backup
android.appwidget
android.bluetooth
android.content
android.content.pm
android.content.res
android.database
android.database.sqlite
ActionBar.OnMenuVisibilityListener
ActionBar.OnNavigationListener
ActionBar.TabListener
Application.ActivityLifecycleCallbacks
Application.OnProvideAssistDataListener
AppOpsManager.OnOpChangedListener
DatePickerDialog.OnDateSetListener
FragmentBreadCrumbs.OnBreadCrumbClickListener
FragmentManager.BackStackEntry
FragmentManager.OnBackStackChangedListener
KeyguardManager.OnKeyguardExitResult
LoaderManager.LoaderCallbacks
PendingIntent.OnFinished
SearchManager.OnCancelListener
SearchManager.OnDismissListener
TimePickerDialog.OnTimeSetListener
UiAutomation.AccessibilityEventFilter
UiAutomation.OnAccessibilityEventListener

Classes

ActionBar
ActionBar.LayoutParams
ActionBar.Tab
Activity
ActivityGroup
ActivityManager

You should implement this method to release any caches or other e system will om this method.

Preferably, you should implement onTrimMemory(int) (/reference/android/content/ComponentCallbacks2.html#onTrimMemory(int)) from (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() (/reference/android/content/ComponentCallbacks.html#onLowMemory()) method as a fallback for older versions, which can be treated the same as onTrimMemory(int) (/reference/android/content/ComponentCallbacks2.html#onTrimMemory(int)) with the /ComponentCallbacks2.html#TRIM_MEMORY_COMPLETE) level.

Interfaces

public boolean **onMenuItemSelected** (int featureId, MenuItem item) Added in API level 1

Default implementation of onMenuItemSelected(int, MenuItem) (/reference/android/view/Window.Callback.html#onMenuItemSelected(int, android.view.MenuItem)) for activities. This calls through to the new onOptionsItemSelected(MenuItem) (/reference/android/app/Activity.html#onOptionsItemSelected(android.view.MenuItem)) method for the FEATURE_OPTIONS_PANEL (/reference/android/view/Window.html#FEATURE_OPTIONS_PANEL) panel, so that subclasses of Activity don't need to deal with feature codes.

Parameters

featureId The panel that the menu is in.

item The menu item that was selected.

Returns

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).

public boolean **onMenuOpened** (int featureId, Menu menu) Added in API level 1

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

Parameters

featureId The panel that the menu is in.

menu The menu that is opened.

The default implementation returns true.

public boolean **onNavigateUp** () Added in API level 16

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` (/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)` (/reference/android/app/Activity.html#onPrepareNavigateUpTaskStack(android.app.TaskStackBuilder)) to supply those arguments.

See Tasks and Back Stack (/guide/topics/fundamentals/tasks-and-back-stack.html) from the developer guide and Navigation (/design/patterns/navigation.html) from the design guide for more information about navigating within your app.

See the `TaskStackBuilder` (/reference/android/app/TaskStackBuilder.html) class and the Activity methods `getParentActivityIntent()` (/reference/android/app/Activity.html#getParentActivityIntent()), `shouldUpRecreateTask(Intent)` (/reference/android/app/Activity.html#shouldUpRecreateTask(android.content.Intent)), and `navigateUpTo(Intent)` (/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

true if Up navigation completed successfully and this Activity was finished, false otherwise.

public boolean **onNavigateUpFromChild** (Activity

Interfaces

ActionBar.OnMenuVisibilityListener
ActionBar.OnNavigationListener
ActionBar.TabListener
Application.ActivityLifecycleCallbacks
Application.OnProvideAssistDataListener
AppOpsManager.OnOpChangedListener
DatePickerDialog.OnDateSetListener
FragmentBreadCrumbs.OnBreadCrumbClickListener
FragmentManager.BackStackEntry
FragmentManager.OnBackStackChangedListener
KeyguardManager.OnKeyguardExitResult
LoaderManager.LoaderCallbacks
PendingIntent.OnFinished
SearchManager.OnCancelListener
SearchManager.OnDismissListener
TimePickerDialog.OnTimeSetListener
UiAutomation.AccessibilityEventFilter
UiAutomation.OnAccessibilityEventListener

Classes

ActionBar
ActionBar.LayoutParams
ActionBar.Tab
Activity
ActivityGroup
ActivityManager

child) Added in API level 16

Android APIs API level: 19

ts to navigate up. The default implementation simply calls onNavigateUp() on this activity (the parent).

android
android.accessibilityservice
android.accounts
android.animation
android.app
android.app.admin
android.app.backup
android.appwidget
android.bluetooth
android.content
android.content.pm
android.content.res
android.database
android.database.sqlite

Parameters

this The activity making the call.

public boolean **onOptionsItemSelected** (MenuItem item) Added in API level 1

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

item The menu item that was selected.

Returns

boolean Return false to allow normal menu processing to proceed, true to consume it here.

See Also

onCreateOptionsMenu(Menu)

Interfaces

ActionBar.OnMenuVisibilityListener
ActionBar.OnNavigationListener
ActionBar.TabListener
Application.ActivityLifecycleCallbacks
Application.OnProvideAssistDataListener
AppOpsManager.OnOpChangedListener
DatePickerDialog.OnDateSetListener
FragmentBreadCrumbs.OnBreadCrumbClickListener
FragmentManager.BackStackEntry
FragmentManager.OnBackStackChangedListener
KeyguardManager.OnKeyguardExitResult
LoaderManager.LoaderCallbacks
PendingIntent.OnFinished
SearchManager.OnCancelListener
SearchManager.OnDismissListener
TimePickerDialog.OnTimeSetListener
UiAutomation.AccessibilityEventFilter
UiAutomation.OnAccessibilityEventListener

public void **onOptionsMenuClosed** (Menu menu) Added in API level 1

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 The options menu as last shown or first initialized by onCreateOptionsMenu().

Classes

ActionBar
ActionBar.LayoutParams
ActionBar.Tab
Activity
ActivityGroup
ActivityManager

public void **onPanelClosed** (int featureId, Menu menu) Added in API level 1

Default implementation of onPanelClosed(int, Menu) for activities. This calls through to the deprecated onPanelClosed(int, android.view.Menu)

onOptionsMenuClosed(Menu) (/reference/android

Android APIs API level: 19 nu)) method for

the FEATURE_OPTIONS_PANEL (/reference/android

/view/Window.html#FEATURE_OPTIONS_PANEL) panel, so tha subclasses of

Activity don't need to deal with feature codes. For co ntext menus

FEATURE_CONTEXT_MENU (/reference/android

/view/Window.html#FEATURE_CONTEXT_MENU)), the

onContextMenuClosed(Menu) (/reference/androi

/app/Activity.html#onContextMenuClosed(android.view.Me nu)) will be

called.

Parameters

featureId The panel that is being displayed.

menu If onCreatePanelView() returned null, th is is the Menu being displayed in the panel.

public void **onPrepareNavigateUpTaskStack** (TaskStackBuilder builder) Added in API level 16

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

This method receives the TaskStackBuilder (/re erence/android /app/TaskStackBuilder.html) with the constructed serie s of Intents as generated by onCreateNavigateUpTaskStack(TaskStackBu ilder) (/reference/android

/app/Activity.html#onCreateNavigateUpTaskStack(androic .app.TaskStackBu

ilder)). If any extra Intents should be added to these in ents before launching the new task, the application should overri de this method and edit the builder.

Parameters

builder A TaskStackBuilder that has been popula ted with Intents by onCreateNavigateUpTaskStac k.

public boolean **onPrepareOptionsMenu** (Menu menu) Added in API level 1

Prepare the Screen's standard options menu to be di splayed. This is called right before the menu is shown, every time it is shown. You can use this method to efficiently enable/disable iten s or otherwise dynamically modify the contents.

The default implementation updates the system mer u items based

Interfaces

ActionBar.OnMenuVisibilityListener
ActionBar.OnNavigationListener
ActionBar.TabListener
Application.ActivityLifecycleCallbacks
Application.OnProvideAssistDataListener
AppOpsManager.OnOpChangedListener
DatePickerDialog.OnDateSetListener
FragmentBreadCrumbs.OnBreadCrumbClickListener
FragmentManager.BackStackEntry
FragmentManager.OnBackStackChangedListener
KeyguardManager.OnKeyguardExitResult
LoaderManager.LoaderCallbacks
PendingIntent.OnFinished
SearchManager.OnCancelListener
SearchManager.OnDismissListener
TimePickerDialog.OnTimeSetListener
UiAutomation.AccessibilityEventFilter
UiAutomation.OnAccessibilityEventListener

Classes

ActionBar
ActionBar.LayoutParams
ActionBar.Tab
Activity
ActivityGroup

on the activity's state. Deriving classes should always call through to

Android APIs API level: 19

Parameters

menu The options menu as last shown or first in tialized by onCreateOptionsMenu().

Returns

You must return true for the menu to be displayed, if you return false it will not be shown.

See Also

onCreateOptionsMenu(Menu)

public boolean **onPreparePanel** (int featureId, View view, Menu menu) Added in API level 1

Default implementation of onPreparePanel(int, View, Menu) (/reference/android/view/Window.Callback.html#onPreparePanel(int, android.view.View, android.view.Menu)) for activities. This calls through to the new onPrepareOptionsMenu(Menu) (/reference/android/app/Activity.html#onPrepareOptionsMenu(android.view.Menu)) if the featureId is FEATURE_OPTIONS_PANEL (/reference/android/view/Window.html#FEATURE_OPTIONS_PANEL) panel, so that subclasses of Activity don't need to deal with feature codes.

Parameters

featureId The panel that is being displayed.

view The View that was returned by onCreatePanelView() if it returned null, this is the Menu being displayed in the panel.

Returns

boolean You must return true for the panel to be displayed; if you return false it will not be shown.

public void **onProvideAssistData** (Bundle data) Added in API level 18

This is called when the user is requesting an assist, to build a full ACTION_ASSIST (/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 (/reference/android/content/Intent.html#EXTRA_ASSIST_CONTEXT) part of the assist Intent. The

Interfaces

ActionBar.OnMenuVisibilityListener
ActionBar.OnNavigationListener
ActionBar.TabListener
Application.ActivityLifecycleCallbacks
Application.OnProvideAssistDataListener
AppOpsManager.OnOpChangedListener
DatePickerDialog.OnDateSetListener
FragmentBreadCrumbs.OnBreadCrumbClickListener
FragmentManager.BackStackEntry
FragmentManager.OnBackStackChangedListener
KeyguardManager.OnKeyguardExitResult
LoaderManager.LoaderCallbacks
PendingIntent.OnFinished
SearchManager.OnCancelListener
SearchManager.OnDismissListener
TimePickerDialog.OnTimeSetListener
UiAutomation.AccessibilityEventFilter
UiAutomation.OnAccessibilityEventListener

Classes

ActionBar
ActionBar.LayoutParams
ActionBar.Tab
Activity
ActivityGroup
ActivityManager

default implementation does nothing.

Android APIs API level: 19

lbacks that had been registered with `Application.registerOnProvideAssistDataListener` (/reference/android/app/Application.html#registerOnProvideAssistDataListener(android.app.

android
android.accessibilityservice
android.accounts
android.animation
android.app
android.app.admin
android.app.backup
android.appwidget
android.bluetooth
android.content
android.content.pm
android.content.res
android.database
android.database.sqlite

public Object **onRetainNonConfigurationInstance** () Added in API level 1

This method was deprecated in API level 13.
Use the new `Fragment` (/reference/android/app/Fragment.html) API `setRetainInstance(boolean)` (/reference/android/app/Fragment.html#setRetainInstance(boolean)) instead; this is also available on older platforms through the Android compatibility package.

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()` (/reference/android/app/Activity.html#getLastNonConfigurationInstance()) in the new activity instance. *If you are targeting `HONEYCOMB` (/reference/android/os/Build.VERSION_CODES.html#HONEYCOMB) or later, consider instead using a `Fragment` (/reference/android/app/Fragment.html) with `Fragment.setRetainInstance(boolean)` (/reference/android/app/Fragment.html#setRetainInstance(boolean)).*

This function is called purely as an optimization, and you must not rely on it being called. When it is called, a number of guarantees will be made to help optimize configuration switching:

- The function will be called between `onStop()` and `onDestroy()`.
- A new instance of the activity will *always* be immediately created after this one's `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()` method of the [Activity](#)
- These guarantees are designed so that an activity can use this API

Interfaces

ActionBar.OnMenuVisibilityListener
ActionBar.OnNavigationListener
ActionBar.TabListener
Application.ActivityLifecycleCallbacks
Application.OnProvideAssistDataListener
AppOpsManager.OnOpChangedListener
DatePickerDialog.OnDateSetListener
FragmentBreadCrumbs.OnBreadCrumbClickListener
FragmentManager.BackStackEntry
FragmentManager.OnBackStackChangedListener
KeyguardManager.OnKeyguardExitResult
LoaderManager.LoaderCallbacks
PendingIntent.OnFinished
SearchManager.OnCancelListener
SearchManager.OnDismissListener
TimePickerDialog.OnTimeSetListener
UiAutomation.AccessibilityEventFilter
UiAutomation.OnAccessibilityEventListener

Classes

ActionBar
ActionBar.LayoutParams
ActionBar.Tab
Activity
ActivityGroup
ActivityManager

to propagate extensive state from the old to new activity instance,

Android APIs API level: 19

ly actively
any data that
may change based on the configuration, including ar y data loaded
from resources such as strings, layouts, or drawables.

The guarantee of no message handling during the sw itch to the next
activity simplifies use with active objects. For examp le if your
retained state is an AsyncTask (/reference/android/os/AsyncTask.html) you are guaranteed that its call ba ck functions
(like onPostExecute(Result) (/reference/android/os/AsyncTask.html#onPostExecute(Result))) will not be called from the
call here until you execute the next instance's onCreate(Bundle)
(/reference/android/app/Activity.html#onCreate(android.os.Bundle)).

(Note however that there is of course no such guarantee for
doInBackground(Params...) (/reference/androi at is running in a
separate thread.)

Returns

Return any Object holding the desired state to propagate to the
next activity instance.

Interfaces

ActionBar.OnMenuVisibilityListener
ActionBar.OnNavigationListener
ActionBar.TabListener
Application.ActivityLifecycleCallbacks
Application.OnProvideAssistDataListener
AppOpsManager.OnOpChangedListener
DatePickerDialog.OnDateSetListener
FragmentBreadCrumbs.OnBreadCrumbClickListener
FragmentManager.BackStackEntry
FragmentManager.OnBackStackChangedListener
KeyguardManager.OnKeyguardExitResult
LoaderManager.LoaderCallbacks
PendingIntent.OnFinished
SearchManager.OnCancelListener
SearchManager.OnDismissListener
TimePickerDialog.OnTimeSetListener
UiAutomation.AccessibilityEventFilter
UiAutomation.OnAccessibilityEventListener

Classes

ActionBar
ActionBar.LayoutParams
ActionBar.Tab
Activity
ActivityGroup
ActivityManager

public boolean onSearchRequested ()

Added in API level 1

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, search button, or other wid gets within your
activity. Unless overridden, calling this function is the same as calling
startSearch(null, false, null, false) (/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 SearchMana ger (/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).

Returns

Returns true if search launched, and false if activity blocks it.
The default implementation always returns true.

See Also

public boolean **onTouchEvent** (MotionEvent event) Added in API level 1

Android APIs API level: 19

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.

android
android.accessibilityservice
android.accounts
android.animation
android.app
android.app.admin
android.app.backup
android.appwidget
android.bluetooth
android.content
android.content.pm
android.content.res
android.database
android.database.sqlite

Parameters

event The touch screen event being processed.

Returns

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

public boolean **onTrackballEvent** (MotionEvent event) Added in API level 1

d 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 The trackball event being processed.

Returns

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

public void **onTrimMemory** (int level) Added in API level 14

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)` (/reference/android.app.ActivityManager.RunningAppProcessInfo)).

Interfaces

ActionBar.OnMenuVisibilityListener
ActionBar.OnNavigationListener
ActionBar.TabListener
Application.ActivityLifecycleCallbacks
Application.OnProvideAssistDataListener
AppOpsManager.OnOpChangedListener
DatePickerDialog.OnDateSetListener
FragmentBreadCrumbs.OnBreadCrumbClickListener
FragmentManager.BackStackEntry
FragmentManager.OnBackStackChangedListener
KeyguardManager.OnKeyguardExitResult
LoaderManager.LoaderCallbacks
PendingIntent.OnFinished
SearchManager.OnCancelListener
SearchManager.OnDismissListener
TimePickerDialog.OnTimeSetListener
UiAutomation.AccessibilityEventFilter
UiAutomation.OnAccessibilityEventListener

Classes

ActionBar
ActionBar.LayoutParams
ActionBar.Tab
Activity
ActivityGroup
ActivityManager

Parameters

Android APIs API level: 19

amount of
trimming the application may like to perform. May be
TRIM_MEMORY_COMPLETE, TRIM_MEMORY_MODERATE,
TRIM_MEMORY_BACKGROUND,
TRIM_MEMORY_UI_HIDDEN,
TRIM_MEMORY_RUNNING_CRITICAL,
TRIM_MEMORY_RUNNING_LOW,
TRIM_MEMORY_RUNNING_MODERATE.

android
android.accessibilityservice
android.accounts
android.animation
android.app
android.app.admin
android.app.backup
android.appwidget
android.bluetooth
android.content
android.content.pm
android.content.res
android.database
android.database.sqlite

public void **onUserInteraction** () Added in API level 3

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() (/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() (/reference/android/app/Activity.html#onUserLeaveHint()) callback will be accompanied by calls to onUserInteraction() (/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()

Interfaces

ActionBar.OnMenuVisibilityListener
ActionBar.OnNavigationListener
ActionBar.TabListener
Application.ActivityLifecycleCallbacks
Application.OnProvideAssistDataListener
AppOpsManager.OnOpChangedListener
DatePickerDialog.OnDateSetListener
FragmentBreadCrumbs.OnBreadCrumbClickListener
FragmentManager.BackStackEntry
FragmentManager.OnBackStackChangedListener
KeyguardManager.OnKeyguardExitResult
LoaderManager.LoaderCallbacks
PendingIntent.OnFinished
SearchManager.OnCancelListener
SearchManager.OnDismissListener
TimePickerDialog.OnTimeSetListener
UiAutomation.AccessibilityEventFilter
UiAutomation.OnAccessibilityEventListener

public void **onWindowAttributesChanged** (WindowManager.LayoutParams params) Added in API level 1

This is called whenever the current window attributes change.

Classes

ActionBar
ActionBar.LayoutParams
ActionBar.Tab
Activity
ActivityGroup
ActivityManager

public void **onWindowFocusChanged** (boolean hasFocus) Added in API level 1

Called when the current Window (/reference/android/view/Window.html) of the activity gains or loses focus. This is the best
The default
implementation clears the key tracking state, so should always be

called.

Android APIs API level: 19

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() (/reference/android/app/Activity.html#onResume()).

As a general rule, however, a resumed activity will have window focus... unless it has displayed other 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

hasFocus Whether the window of this activity has focus.

See Also

- hasWindowFocus()
- onResume()
- onWindowFocusChanged(boolean)

public ActionMode onWindowStartingActionMode (ActionMode.Callback callback)

Added in API level 11

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) (/reference/android/app/Activity.html#onActionModeStarted(android.view.ActionMode)).

Parameters

callback The callback that should control the new action mode

Returns

The new action mode, or `null` if the activity does not want to provide special handling for this action mode. (It will be handled by the system.)

public void openContextMenu (View view)

Added in API level 1

Interfaces

- ActionBar.OnMenuVisibilityListener
- ActionBar.OnNavigationListener
- ActionBar.TabListener
- Application.ActivityLifecycleCallbacks
- Application.OnProvideAssistDataListener
- AppOpsManager.OnOpChangedListener
- DatePickerDialog.OnDateSetListener
- FragmentBreadCrumbs.OnBreadCrumbClickListener
- FragmentManager.BackStackEntry
- FragmentManager.OnBackStackChangedListener
- KeyguardManager.OnKeyguardExitResult
- LoaderManager.LoaderCallbacks
- PendingIntent.OnFinished
- SearchManager.OnCancelListener
- SearchManager.OnDismissListener
- TimePickerDialog.OnTimeSetListener
- UiAutomation.AccessibilityEventFilter
- UiAutomation.OnAccessibilityEventListener

Classes

- ActionBar
- ActionBar.LayoutParams
- ActionBar.Tab
- Activity
- ActivityGroup
- ActivityManager

Programmatically opens the context menu for a particular `view`. The

Android APIs API level: 19

android
android.accessibilityservice
android.accounts
android.animation
android.app
android.app.admin
android.app.backup
android.appwidget
android.bluetooth
android.content
android.content.pm
android.content.res
android.database
android.database.sqlite

(/reference/android/app/Activity.html#registerForContextMenu(android.view.View)).

Parameters

view The view to show the context menu for.

public void **openOptionsMenu** () Added in API level 1

Programmatically opens the options menu. If the options menu is already open, this method does nothing.

public void **overridePendingTransition** (int enterAnim, int exitAnim) Added in API level 5

`startActivity(Intent)` (/reference/android/app/Activity.html#startActivity(android.content.Intent)) or `finish()` (/reference/android/app/Activity.html#finish()) to specify an explicit transition animation to perform next.

As of `JELLY_BEAN` (/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` (/reference/android/app/ActivityOptions.html) bundle to {@link #startActivity(Intent, 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

enterAnim A resource ID of the animation resource to use for the incoming activity. Use 0 for no animation.

exitAnim A resource ID of the animation resource to use for the outgoing activity. Use 0 for no animation.

public void **recreate** () Added in API level 11

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()` (/reference/android/app/Activity.html#onDestroy()) and then created after it.

Interfaces

ActionBar.OnMenuVisibilityListener
ActionBar.OnNavigationListener
ActionBar.TabListener
Application.ActivityLifecycleCallbacks
Application.OnProvideAssistDataListener
AppOpsManager.OnOpChangedListener
DatePickerDialog.OnDateSetListener
FragmentBreadCrumbs.OnBreadCrumbClickListener
FragmentManager.BackStackEntry
FragmentManager.OnBackStackChangedListener
KeyguardManager.OnKeyguardExitResult
LoaderManager.LoaderCallbacks
PendingIntent.OnFinished
SearchManager.OnCancelListener
SearchManager.OnDismissListener
TimePickerDialog.OnTimeSetListener
UiAutomation.AccessibilityEventFilter
UiAutomation.OnAccessibilityEventListener

Classes

ActionBar
ActionBar.LayoutParams
ActionBar.Tab
Activity
ActivityGroup
ActivityManager

public void **registerForContextMenu** (View view) Added in API level 1

Android APIs API level: 19

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` (/reference/android/view/View.OnCreateContextMenuListener.html) on the view to this activity, so `onCreateContextMenu(ContextMenu, View, ContextMenu.ContextMenuInfo)` (/reference/android/app/Activity.html#onCreateContextMenu(android.view.ContextMenu, android.view.View, android.view.ContextMenu.ContextMenuInfo)) will be called when it is time to show the context menu.

Parameters

view The view that should show a context menu

See Also

- `unregisterForContextMenu(View)`

public final void **removeDialog** (int id) Added in API level 1

This method was deprecated in API level 13.
Use the new `DialogFragment` (/reference/android/app/DialogFragment.html) class with `FragmentManager` (/reference/android/app/FragmentManager.html) instead; this is also available on older platforms through the Android compatibility package.

Removes any internal references to a dialog managed by this Activity. If the dialog is showing, it will dismiss it as part of the clean up.

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` (/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 The id of the managed dialog.

See Also

- `onCreateDialog(int, Bundle)`
- `onPrepareDialog(int, Dialog, Bundle)`
- `dismissDialog(int)`

Interfaces

- ActionBar.OnMenuVisibilityListener
- ActionBar.OnNavigationListener
- ActionBar.TabListener
- Application.ActivityLifecycleCallbacks
- Application.OnProvideAssistDataListener
- AppOpsManager.OnOpChangedListener
- DatePickerDialog.OnDateSetListener
- FragmentBreadCrumbs.OnBreadCrumbClickListener
- FragmentManager.BackStackEntry
- FragmentManager.OnBackStackChangedListener
- KeyguardManager.OnKeyguardExitResult
- LoaderManager.LoaderCallbacks
- PendingIntent.OnFinished
- SearchManager.OnCancelListener
- SearchManager.OnDismissListener
- TimePickerDialog.OnTimeSetListener
- UiAutomation.AccessibilityEventFilter
- UiAutomation.OnAccessibilityEventListener

Classes

- ActionBar
- ActionBar.LayoutParams
- ActionBar.Tab
- Activity
- ActivityGroup
- ActivityManager

public void **reportFullyDrawn** ()

Added in API level 19

Android APIs API level: 19

, purely for diagnostic purposes (calling it does not impact the v sible behavior of the activity). This is only used to help instrument a pplication launch times, so that the app can report when it is fu ly in a usable state, without this, the only thing the system itself ca n 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)` (/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.

android
android.accessibilityservice
android.accounts
android.animation
android.app
android.app.admin
android.app.backup
android.appwidget
android.bluetooth
android.content
android.content.pm
android.content.res
android.database
android.database.sqlite
android.drawable
android.gesture
android.graphics
android.graphics.drawable
android.graphics.drawable.shapes
android.graphics.pdf
android.hardware
android.hardware.camera2
android.hardware.camera2.params
android.hardware.display
android.hardware.fingerprint
android.hardware.input
android.hardware.usb
android.inputmethodservice
android.location
android.media
android.media.audiofx
android.media.browse
android.media.effect
android.media.midi
android.media.projection
android.media.session
android.media.tv
android.mtp
android.net
android.net.http
android.net.nsd
android.net.rtp
android.net.sip
android.net.wifi
android.net.wifi.p2p
android.net.wifi.p2p.nsd
android.nfc
android.nfc.cardemulation
android.nfc.tech
android.opengl
android.os
android.os.storage
android.preference
android.print
android.print.pdf
android.printservice
android.provider
android.renderscript
android.sax
android.security
android.security.keystore
android.service.carrier
android.service.chooser
android.service.dreams
android.service.gatt
android.service.notification
android.service.textservice
android.service.voice
android.service.wallpaper
android.speech
android.speech.tts
android.support.annotation
android.support.customtabs
android.support.design.widget
android.support.multidex
android.support.percent
android.support.transition
android.support.v13.app
android.support.v17.leanback
android.support.v17.leanback.app
android.support.v17.leanback.database
android.support.v17.leanback.graphics
android.support.v17.leanback.system
android.support.v17.leanback.widget
android.support.v17.preference
android.support.v4.app
android.support.v4.content
android.support.v4.content.pm
android.support.v4.content.res
android.support.v4.graphics
android.support.v4.graphics.drawable
android.support.v4.media
android.support.v4.media.session
android.support.v4.net
android.support.v4.os
android.support.v4.print
android.support.v4.provider
android.support.v4.text
android.support.v4.text.util
android.support.v4.util
android.support.v4.view
android.support.v4.view.accessibility
android.support.v4.view.animation
android.support.v4.widget
android.support.v7.app
android.support.v7.appcompat
android.support.v7.cardview
android.support.v7.gridlayout
android.support.v7.media
android.support.v7.mediarouter
android.support.v7.palette
android.support.v7.preference
android.support.v7.recyclerview
android.support.v7.widget
android.support.wear.ambient
android.support.wear.internal.widget
android.support.wear.internal.widget.drawer
android.support.wear.widget
android.support.wear.widget.drawer
android.system
android.telecom
android.telephony
android.telephony.cdma
android.telephony.gsm
android.test
android.test.mock
android.test.suitebuilder
android.test.suitebuilder.annotation
android.text
android.text.format
android.text.method
android.text.style
android.text.util
android.transition
android.util
android.view
android.view.accessibility
android.view.animation
android.view.inputmethod
android.view.textservice
android.webkit
android.widget
dalvik.annotation
dalvik.bytecode
dalvik.system
java.awt.font
java.beans
java.io
java.lang
java.lang.annotation
java.lang.ref
java.lang.reflect
java.math
java.net
java.nio
java.nio.channels
java.nio.channels.spi
java.nio.charset
java.nio.charset.spi
java.security
java.security.acl
java.security.cert
java.security.interfaces
java.security.spec
java.sql
java.text
java.text.spi
java.util
java.util.concurrent
java.util.concurrent.atomic
java.util.concurrent.locks
java.util.jar
java.util.logging
java.util.prefs
java.util.regex
java.util.zip
javax.crypto
javax.crypto.interfaces
javax.crypto.spec
javax.microedition.khronos.egl
javax.microedition.khronos.opengles
javax.net
javax.net.ssl
javax.security.auth
javax.security.auth.callback
javax.security.auth.login
javax.security.auth.spi
javax.security.auth.x500
javax.security.cert
javax.sql
javax.xml.datatype
javax.xml.namespace
javax.xml.parsers
javax.xml.transform
javax.xml.transform.dom
javax.xml.transform.sax
javax.xml.transform.stream
javax.xml.validation
javax.xml.xpath
junit.framework
org.apache.http.conn.scheme
org.apache.http.conn.ssl
org.apache.http.cookie
org.apache.http.entity
org.apache.http.impl.conn.tsccm
org.apache.http.message
org.apache.http.params
org.apache.http.protocol
org.apache.http.util
org.json
org.w3c.dom
org.w3c.dom.ls
org.xml.sax
org.xml.sax.ext
org.xml.sax.helpers
org.xmlpull.v1
org.xmlpull.v1.sax2

public void **requestWindowFeature** (int featureId)

Added in API level 1

Enable extended window features. This is a convenience for calling `getWindow().requestFeature()` (/reference/android/view/Window.html#requestFeature(int)).

Parameters

featureId The desired feature as defined in `Window`.

Returns

Returns true if the requested feature is supported and now enabled.

See Also

`requestFeature(int)`

Interfaces

ActionBar.OnMenuVisibilityListener
ActionBar.OnNavigationListener
ActionBar.TabListener
Application.ActivityLifecycleCallbacks
Application.OnProvideAssistDataListener
AppOpsManager.OnOpChangedListener
DatePickerDialog.OnDateSetListener
FragmentBreadCrumbs.OnBreadCrumbClickListener
FragmentManager.BackStackEntry
FragmentManager.OnBackStackChangedListener
KeyguardManager.OnKeyguardExitResult
LoaderManager.LoaderCallbacks
PendingIntent.OnFinished
SearchManager.OnCancelListener
SearchManager.OnDismissListener
TimePickerDialog.OnTimeSetListener
UiAutomation.AccessibilityEventFilter
UiAutomation.OnAccessibilityEventListener

public final void **runOnUiThread** (Runnable action)

Added in API level 1

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

action the action to run on the UI thread

Classes

ActionBar
ActionBar.LayoutParams
ActionBar.Tab
Activity
ActivityGroup
ActivityManager

public void **setContentView** (int layoutResID)

Added in API level 1

esource will be inflated, adding all top-level views to the activity.

Parameters

Android APIs API level: 19

See Also

`setContentView(android.view.View)`
`setContentView(android.view.View, android.view.ViewGroup.LayoutParams)`

android
android.accessibilityservice
android.accounts
android.animation
android.app
android.app.admin
android.app.backup
android.bluetooth
android.content
android.content.pm
android.content.res
android.database
android.database.sqlite

public void **setContentView** (View view) Added in API level 1

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. When calling this method, the layout parameters of the specified view are ignored. Both the width and the height of the view are set by default to `MATCH_PARENT` (/reference/android/view/ViewGroup.LayoutParams.html#MATCH_PARENT). To use your own

`setContentView(android.view.View, android.view.ViewGroup.LayoutParams)` (/reference/android/app/Activity.html#setContentView(android.view.View, android.view.ViewGroup.LayoutParams)) instead.

Parameters

view The desired content to display.

See Also

`setContentView(int)`
`setContentView(android.view.View, android.view.ViewGroup.LayoutParams)`

Interfaces

ActionBar.OnMenuVisibilityListener
ActionBar.OnNavigationListener
ActionBar.TabListener
Application.ActivityLifecycleCallbacks
Application.OnProvideAssistDataListener
AppOpsManager.OnOpChangedListener
DatePickerDialog.OnDateSetListener
FragmentBreadCrumbs.OnBreadCrumbClickListener
FragmentManager.BackStackEntry
FragmentManager.OnBackStackChangedListener
KeyguardManager.OnKeyguardExitResult
LoaderManager.LoaderCallbacks
PendingIntent.OnFinished
SearchManager.OnCancelListener
SearchManager.OnDismissListener
TimePickerDialog.OnTimeSetListener
UiAutomation.AccessibilityEventFilter
UiAutomation.OnAccessibilityEventListener

public void **setContentView** (View view, ViewGroup.LayoutParams params) Added in API level 1

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 The desired content to display.

params Layout parameters for the view.

See Also

`setContentView(android.view.View)`
`setContentView(int)`

Classes

ActionBar
ActionBar.LayoutParams
ActionBar.Tab
Activity
ActivityGroup
ActivityManager

public final void **setDefaultKeyMode** (int mode)

Select the default key handling for this activity. This controls what
Android APIs API level: 19 led. The default
android
android.accessibilityservice
android.accounts
android.animation
android.app
android.app.admin
android.app.backup
android.appwidget
android.bluetooth
android.content
android.content.pm
android.content.res
android.database
android.database.sqlite
(/app/Activity.html#DEFAULT_KEYS_DISABLE)) will simply drop them on the floor. Other modes allow you to launch the dialer (DEFAULT_KEYS_DIALER (/reference/android/app/Activity.html#DEFAULT_KEYS_DIALER)), execute a shortcut in your down (DEFAULT_KEYS_SHORTCUT (/reference/android/app/Activity.html#DEFAULT_KEYS_SHORTCUT)), or launch a search (DEFAULT_KEYS_SEARCH_LOCAL (/reference/android/app/Activity.html#DEFAULT_KEYS_SEARCH_LOCAL) and DEFAULT_KEYS_SEARCH_GLOBAL (/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 The desired default key mode constant.

See Also

DEFAULT_KEYS_DISABLE
DEFAULT_KEYS_DIALER
DEFAULT_KEYS_SHORTCUT
DEFAULT_KEYS_SEARCH_LOCAL
DEFAULT_KEYS_SEARCH_GLOBAL
onKeyDown(int, KeyEvent)

public final void **setFeatureDrawable** (int featureId, Drawable drawable) Added in API level 1

Convenience for calling setFeatureDrawable(int, Drawable) (/reference/android/view/Window.html#setFeatureDrawable(int, android.graphics.drawable.Drawable)).

public final void **setFeatureDrawableAlpha** (int featureId, int alpha) Added in API level 1

Convenience for calling setFeatureDrawableAlpha(int, int) (/reference/android/view/Window.html#setFeatureDrawableAlpha(int, int)).

Interfaces

ActionBar.OnMenuVisibilityListener
ActionBar.OnNavigationListener
ActionBar.TabListener
Application.ActivityLifecycleCallbacks
Application.OnProvideAssistDataListener
AppOpsManager.OnOpChangedListener
DatePickerDialog.OnDateSetListener
FragmentBreadCrumbs.OnBreadCrumbClickListener
FragmentManager.BackStackEntry
FragmentManager.OnBackStackChangedListener
KeyguardManager.OnKeyguardExitResult
LoaderManager.LoaderCallbacks
PendingIntent.OnFinished
SearchManager.OnCancelListener
SearchManager.OnDismissListener
TimePickerDialog.OnTimeSetListener
UiAutomation.AccessibilityEventFilter
UiAutomation.OnAccessibilityEventListener

Classes

ActionBar
ActionBar.LayoutParams
ActionBar.Tab
Activity
ActivityGroup
ActivityManager

public final void **setFeatureDrawableResource** (int

Android APIs API level: 19 Added in API level 1

Convenience for calling setFeatureDrawableResource(int, int) (/reference/android/view/Window.html#setFeatureDrawableResource(int, int)).

android
android.accessibilityservice
android.accounts
android.animation
android.app
android.app.admin
android.app.backup
android.appwidget
android.bluetooth
android.content
android.content.pm
android.content.res
android.database
android.database.sqlite

public final void **setFeatureDrawableUri** (int, Uri) Added in API level 1

Convenience for calling setFeatureDrawableUri(int, Uri) (/reference/android/view/Window.html#setFeatureDrawableUri(int, android.net.Uri)).

public void **setFinishOnTouchOutside** (boolean finish) Added in API level 11

Sets whether this activity is finished when touched outside its window's bounds.

Interfaces

public void **setImmersive** (boolean i) Added in API level 18

ActionBar.OnMenuVisibilityListener
ActionBar.OnNavigationListener
ActionBar.TabListener
Application.ActivityLifecycleCallbacks
Application.OnProvideAssistDataListener
AppOpsManager.OnOpChangedListener
DatePickerDialog.OnDateSetListener
FragmentBreadCrumbs.OnBreadCrumbClickListener
FragmentManager.BackStackEntry
FragmentManager.OnBackStackChangedListener
KeyguardManager.OnKeyguardExitResult
LoaderManager.LoaderCallbacks
PendingIntent.OnFinished
SearchManager.OnCancelListener
SearchManager.OnDismissListener
TimePickerDialog.OnTimeSetListener
UiAutomation.OnAccessibilityEventListener

Adjust the current immersive mode setting. Note that changing this value will have no effect on the activity's ActivityInfo (/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 (/reference/android/content/pm/ActivityInfo.html#flags) will always have its FLAG_IMMERSIVE (/reference/android/content/pm/ActivityInfo.html#FLAG_IMMERSIVE) bit set.

See Also

- isImmersive()
- FLAG_IMMERSIVE

public void **setIntent** (Intent newIntent) Added in API level 1

Change the intent returned by getIntent() (/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) (/reference/android/app/Activity.html#onNewIntent(android.content.Intent)).

Classes

ActionBar
ActionBar.LayoutParams
ActionBar.Tab
Activity
ActivityGroup
ActivityManager

Parameters

newIntent The new Intent object to return from getIntent

See Also

Android APIs API level: 19

android
android.accessibilityservice
android.accounts
android.animation
android.app
android.app.admin
android.app.backup
android.appwidget
android.bluetooth
android.content
android.content.pm
android.content.res
android.database
android.database.sqlite

Interfaces

ActionBar.OnMenuVisibilityListener
ActionBar.OnNavigationListener
ActionBar.TabListener
Application.ActivityLifecycleCallbacks
Application.OnProvideAssistDataListener
AppOpsManager.OnOpChangedListener
DatePickerDialog.OnDateSetListener
FragmentBreadCrumbs.OnBreadCrumbClickListener
FragmentManager.BackStackEntry
FragmentManager.OnBackStackChangedListener
KeyguardManager.OnKeyguardExitResult
LoaderManager.LoaderCallbacks
PendingIntent.OnFinished
SearchManager.OnCancelListener
SearchManager.OnDismissListener
TimePickerDialog.OnTimeSetListener
UiAutomation.AccessibilityEventFilter
UiAutomation.OnAccessibilityEventListener

Classes

ActionBar
ActionBar.LayoutParams
ActionBar.Tab
Activity
ActivityGroup
ActivityManager

public final void **setProgress** (int progress) Added in API level 1

Set 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)` (/reference/android/app/Activity.html#requestWindowFeature(int)).

Parameters

progress 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.

public final void **setProgressBarIndeterminate** (boolean indeterminate) Added in API level 1

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 `requestWindowFeature(int)` (/reference/android/app/Activity.html#requestWindowFeature(int)).

Parameters

indeterminate Whether the horizontal progress bar should be indeterminate.

public final void **setProgressBarIndeterminateVisibility** (boolean visible) Added in API level 1

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)` (/reference/android/app/Activity.html#requestWindowFeature(int)).

Parameters

visible Whether to show the progress bars in the title.

public final void **setProgressBarVisibility** (boolean

visible) Added in API level 1

Android APIs API level: 19

In order for the progress bar to be shown, the feature must be requested via `requestWindowFeature(int)` (/reference/android/app/Activity.html#requestWindowFeature(int)).

visible Whether to show the progress bars in the title.

public void **setRequestedOrientation** (int requestedOrientation) Added in API level 1

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

requestedOrientation An orientation constant as used in `ActivityInfo.screenOrientation`.

public final void **setResult** (int resultCode) Added in API level 1

Call this to set the result that your activity will return to its caller.

Parameters

resultCode The result code to propagate back to the originating activity, often RESULT_CANCELED or RESULT_OK

See Also

RESULT_CANCELED
RESULT_OK
RESULT_FIRST_USER
`setResult(int, Intent)`

public final void **setResult** (int resultCode, Intent data) Added in API level 1

Call this to set the result that your activity will return to its caller.

As of `GINGERBREAD` (/reference/android

u supply here

`Intent.FLAG_GRANT_READ_URI_PERMISSION`

Interfaces

ActionBar.OnMenuVisibilityListener
ActionBar.OnNavigationListener
ActionBar.TabListener
Application.ActivityLifecycleCallbacks
Application.OnProvideAssistDataListener
AppOpsManager.OnOpChangedListener
DatePickerDialog.OnDateSetListener
FragmentBreadCrumbs.OnBreadCrumbClickListener
FragmentManager.BackStackEntry
FragmentManager.OnBackStackChangedListener
KeyguardManager.OnKeyguardExitResult
LoaderManager.LoaderCallbacks
PersistableBundle.OnFinished
SearchManager.OnCancelListener
SearchManager.OnDismissListener
TimePickerDialog.OnTimeSetListener
UiAutomation.AccessibilityEventFilter
UiAutomation.OnAccessibilityEventListener

Classes

ActionBar
ActionBar.LayoutParams
ActionBar.Tab
Activity
ActivityGroup
ActivityManager

(/reference/android/content

Android APIs API level: 19

/reference
android
android.accessibilityservice
android.accounts
android.animation
android.app
android.app.admin
android.app.backup
android.appwidget
android.bluetooth
android.content
android.content.pm
android.content.res
android.database
android.database.sqlite

IntentFLAG_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 URI permissions it already holds).

Parameters

resultCode	The result code to propagate back to the originating activity, often RESULT_CANCELED or RESULT_OK
data	The data to propagate back to the originating activity.

See Also

RESULT_CANCELED
RESULT_OK
RESULT_FIRST_USER
setResult(int)

Interfaces

ActionBar.OnMenuVisibilityListener
ActionBar.OnNavigationListener
ActionBar.TabListener
Application.ActivityLifecycleCallbacks
Application.OnProvideAssistDataListener
AppOpsManager.OnOpChangedListener
DatePickerDialog.OnDateSetListener
FragmentBreadCrumbs.OnBreadCrumbClickListener
FragmentManager.BackStackEntry
FragmentManager.OnBackStackChangedListener
KeyguardManager.OnKeyguardExitResult
LoaderManager.LoaderCallbacks
PendingIntent.OnFinished
SearchManager.OnCancelListener
SearchManager.OnDismissListener
TimePickerDialog.OnTimeSetListener
UiAutomation.AccessibilityEventFilter
UiAutomation.OnAccessibilityEventListener

Classes

ActionBar
ActionBar.LayoutParams
ActionBar.Tab
Activity
ActivityGroup
ActivityManager

public final void **setSecondaryProgress** (int secondaryProgress) Added in API level 1

Sets the secondary progress for the progress bar in the title. This progress is drawn between the primary progress (set via setProgress(int)) (/reference/android and the backgrou/widget/ProgressBar.html#setProgress(int)) nd. 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) (/reference/android/view/Window.html#requestFeature(int)).

Parameters

secondaryProgress	The secondary progress for the progress bar. Valid ranges are from 0 to 10000 (both inclusive).

public void **setTitle** (int titleId) Added in API level 1

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.

Android APIs API level: 19

public void **setTitle** (CharSequence title) Added in API level 1

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.

public void **setTitleColor** (int textColor) Added in API level 1

public void **setVisible** (boolean visible) Added in API level 3

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 prevents any tests for a window to become active (which would cause onTopResumedActivityChanged() to be called) during that time.

The default value for this is taken from the `windowNoDisplay` (/reference/android/R.attr.html#windowNoDisplay) attribute of the activity's theme.

public final void **setVolumeControlStream** (int streamType) Added in API level 1

Suggests an audio stream whose volume should be changed by the hardware volume controls.

The suggested audio stream will be tied to the window of this Activity. If the Activity is switched, the stream set here is no longer the suggested stream. The client does not need to save and restore the old suggested stream value in onPause and onResume.

Parameters

streamType The type of the audio stream whose volume should be changed by the hardware volume controls. 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`.

public boolean **shouldUpRecreateTask** (Intent targetIntent) Added in API level 16

Returns true if the app should recreate the task when navigating 'up'

Interfaces

ActionBar.OnMenuVisibilityListener
ActionBar.OnNavigationListener
ActionBar.TabListener
Application.ActivityLifecycleCallbacks
Application.OnProvideAssistDataListener
AppOpsManager.OnOpChangedListener
DatePickerDialog.OnDateSetListener
FragmentBreadCrumbs.OnBreadCrumbClickListener
FragmentManager.BackStackEntry
FragmentManager.OnBackStackChangedListener
KeyguardManager.OnKeyguardExitResult
LoaderManager.LoaderCallbacks
PendingIntent.OnFinished
SearchManager.OnCancelListener
SearchManager.OnDismissListener
TimePickerDialog.OnTimeSetListener
UiAutomation.AccessibilityEventFilter
UiAutomation.OnAccessibilityEventListener

Classes

ActionBar
ActionBar.LayoutParams
ActionBar.Tab
Activity
ActivityGroup
ActivityManager

from this activity by using targetIntent.

Android APIs API level: 19

navigateUpTo(Intent) (/reference/android/app/Activity.html#navigateUpTo(android.content.Intent)) using the same parameters to correctly perform up navigation. If this method returns false, the app should synthesize a new task stack by using TaskStackBuilder.create(Context) (/reference/android/app/TaskStackBuilder.html) or another similar mechanism to perform up navigation.

Parameters

targetIntent An intent representing the target destination for up navigation

Returns

true if navigating up should recreate a new task stack, false if the same task should be used for the destination

public final boolean **showDialog** (int id, Bundle args) Added in API level 8

This method was deprecated in API level 13.
Use the new DialogFragment (/reference/android/app/DialogFragment.html) class with FragmentManager (/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) (/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 (/reference/android/os/Build.VERSION_CODES.html#HONEYCOMB) or later, consider instead using a DialogFragment (/reference/android/app/DialogFragment.html) instead.*

Each time a dialog is shown, onPrepareDialog(int, Dialog, Bundle) (/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 The id of the managed dialog.

args Arguments to pass through to the dialog. These will be saved and restored for the dialog. Note that if the dialog is already created, onCreateDialog(int, Bundle)

Interfaces

ActionBar.OnMenuVisibilityListener
ActionBar.OnNavigationListener
ActionBar.TabListener
Application.ActivityLifecycleCallbacks
Application.OnProvideAssistDataListener
AppOpsManager.OnOpChangedListener
DatePickerDialog.OnDateSetListener
FragmentBreadCrumbs.OnBreadCrumbClickListener
FragmentManager.BackStackEntry
FragmentManager.OnBackStackChangedListener
KeyguardManager.OnKeyguardExitResult
LoaderManager.LoaderCallbacks
PendingIntent.OnFinished
SearchManager.OnCancelListener
SearchManager.OnDismissListener
TimePickerDialog.OnTimeSetListener
UiAutomation.AccessibilityEventFilter
UiAutomation.OnAccessibilityEventListener

Classes

ActionBar
ActionBar.LayoutParams
ActionBar.Tab
Activity
ActivityGroup
ActivityManager

will not be called with the new arguments but
Android APIs API level: 19 `ıle)` will
be. If you need to rebuild the dialog, call
`removeDialog(int)` first.

android
android.accessibilityservice
android.accounts
android.animation
android.app
android.app.admin
android.app.backup
android.appwidget
android.bluetooth
android.content
android.content.pm
android.content.res
android.database
android.database.sqlite

Returns
Returns true if the Dialog was created; false is returned if it is not created because the `onCreateDialog(int, Bundle)` returns false.

See Also
- `Dialog`
- `onCreateDialog(int, Bundle)`
- `onPrepareDialog(int, Dialog, Bundle)`
- `dismissDialog(int)`
- `removeDialog(int)`

public final void **showDialog** (int id) Added in API level 1

This method was deprecated in API level 13.
Use the new `DialogFragment` (/reference/android/app/DialogFragment.html) class with `FragmentManager` (/reference/android/app/FragmentManager.html) instead; this is also available on older platforms through the Android compatibility package.

Simple version of `showDialog(int, Bundle)` (/reference/android/app/Activity.html#showDialog(int, android.os.Bundle)) that does not take any arguments. Simply calls `showDialog(int, Bundle)` (/reference/android/app/Activity.html#showDialog(int, android.os.Bundle)) with null arguments.

public ActionMode **startActionMode** (ActionMode.Callback callback) Added in API level 11

Parameters
- *callback* Callback that will manage lifecycle events for this context mode

Returns
The ContextMode that was started, or null if it was canceled

See Also
- `ActionMode`

public void **startActivities** (Intent[] intents, Bundle

Interfaces
ActionBar.OnMenuVisibilityListener
ActionBar.OnNavigationListener
ActionBar.TabListener
Application.ActivityLifecycleCallbacks
Application.OnProvideAssistDataListener
AppOpsManager.OnOpChangedListener
DatePickerDialog.OnDateSetListener
FragmentBreadCrumbs.OnBreadCrumbClickListener
FragmentManager.BackStackEntry
FragmentManager.OnBackStackChangedListener
KeyguardManager.OnKeyguardExitResult
LoaderManager.LoaderCallbacks
PendingIntent.OnFinished
SearchManager.OnCancelListener
SearchManager.OnDismissListener
TimePickerDialog.OnTimeSetListener
UiAutomation.AccessibilityEventFilter
UiAutomation.OnAccessibilityEventListener

Classes
ActionBar
ActionBar.LayoutParams
ActionBar.Tab
Activity
ActivityGroup
ActivityManager

options) Added in API level 16

Android APIs API level: 19

android
android.accessibilityservice
android.accounts
android.animation
android.app
android.app.admin
android.app.backup
android.appwidget
android.bluetooth
android.content
android.content.pm
android.content.res
android.database
android.database.sqlite

ation 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 (/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 (/reference/android/content/ActivityNotFoundException.html) if there was no Activity found to run the given Intent.

Parameters

intents The intents to start.

options Additional options for how the Activity should be started. See Context.startActivity(Intent, Bundle) for more details.

Throws

android.content.ActivityNotFoundException

See Also

ERROR(#startActivities(Intent[])}/{@link #startActivities(Intent[], Bundle)})
startActivityForResult(Intent, int)

public void **startActivities** (Intent[] intents) Added in API level 11

Same as startActivities(Intent[], Bundle) (/reference/android/app/Activity.html#startActivities(android.content.Intent[], android.os.Bundle)) with no options specified.

Parameters

intents The intents to start.

Throws

android.content.ActivityNotFoundException

See Also

ERROR(#startActivities(Intent[], Bundle)}/{@link #startActivities(Intent[], Bundle)})
startActivityForResult(Intent, int)

public void **startActivity** (Intent intent) Added in API level 1

Interfaces

ActionBar.OnMenuVisibilityListener
ActionBar.OnNavigationListener
ActionBar.TabListener
Application.ActivityLifecycleCallbacks
Application.OnProvideAssistDataListener
AppOpsManager.OnOpChangedListener
DatePickerDialog.OnDateSetListener
FragmentBreadCrumbs.OnBreadCrumbClickListener
FragmentManager.BackStackEntry
FragmentManager.OnBackStackChangedListener
KeyguardManager.OnKeyguardExitResult
LoaderManager.LoaderCallbacks
PendingIntent.OnFinished
SearchManager.OnCancelListener
SearchManager.OnDismissListener
TimePickerDialog.OnTimeSetListener
UiAutomation.AccessibilityEventFilter
UiAutomation.OnAccessibilityEventListener

Classes

ActionBar
ActionBar.LayoutParams
ActionBar.Tab
Activity
ActivityGroup
ActivityManager

Same as `startActivity(Intent, Bundle)` (/reference/android

Android APIs API level: 19

android.os.Bundle)) with no options specified.

android

Parameters

android.accessibilityservice

android.accounts *intent* The intent to start.

android.animation

android.app

android.appwidget *options* Additional options for how the Activity should be

See Also

android.bluetooth

android.content

android.content.pm

android.content.res

android.database

android.database.sqlite

public void **startActivity** (Intent intent, Bundle

Added in API level 16

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` (/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` (/reference/android/content/ActivityNotFoundException.html) if there was no Activity found to run the given Intent.

Parameters

intent The intent to start.

options Additional options for how the Activity should be started. See `Context.startActivity(Intent, Bundle)` for more details.

Throws

android.content.ActivityNotFoundException

See Also

`startActivityForResult(Intent, int)`

Interfaces

ActionBar.OnMenuVisibilityListener

ActionBar.OnNavigationListener

ActionBar.TabListener

Application.ActivityLifecycleCallbacks

Application.OnProvideAssistDataListener

AppOpsManager.OnOpChangedListener

DatePickerDialog.OnDateSetListener

FragmentBreadCrumbs.OnBreadCrumbClickListener

FragmentManager.BackStackEntry

FragmentManager.OnBackStackChangedListener

KeyguardManager.OnKeyguardExitResult

LoaderManager.LoaderCallbacks

PendingIntent.OnFinished

SearchManager.OnCancelListener

SearchManager.OnDismissListener

TimePickerDialog.OnTimeSetListener

UiAutomation.AccessibilityEventFilter

UiAutomation.OnAccessibilityEventListener

Classes

ActionBar

ActionBar.LayoutParams

ActionBar.Tab

Activity

ActivityGroup

ActivityManager

requestCode)

Same as calling `startActivityForResult(Intent, int, Bundle)` with no options.

Parameters

intent The intent to start.

requestCode If >= 0, this code will be returned in onActivityResult() when the activity exits.

Throws

android.content.ActivityNotFoundException

See Also

`startActivity(Intent)`

public void **startActivityForResult** (Intent intent, int requestCode, Bundle options) Added in API level 16

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)` (/reference/android/app/Activity.html#startActivity(android.content.Intent)) (the activity is not launched as a sub-activity).

Note that this method should only be used with Intent protocols that are defined to return a result. In other protocols (such as `ACTION_MAIN` (/reference/android/content/Intent.html#ACTION_MAIN) or `ACTION_VIEW` (/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` (/reference/android/content/ActivityNotFoundException.html) if there was no Activity found to run the given Intent.

Parameters

Interfaces

ActionBar.OnMenuVisibilityListener
ActionBar.OnNavigationListener
ActionBar.TabListener
Application.ActivityLifecycleCallbacks
Application.OnProvideAssistDataListener
AppOpsManager.OnOpChangedListener
DatePickerDialog.OnDateSetListener
FragmentBreadCrumbs.OnBreadCrumbClickListener
FragmentManager.BackStackEntry
FragmentManager.OnBackStackChangedListener
KeyguardManager.OnKeyguardExitResult
LoaderManager.LoaderCallbacks
PendingIntent.OnFinished
SearchManager.OnCancelListener
SearchManager.OnDismissListener
TimePickerDialog.OnTimeSetListener
UiAutomation.AccessibilityEventFilter
UiAutomation.OnAccessibilityEventListener

Classes

ActionBar
ActionBar.LayoutParams
ActionBar.Tab
Activity
ActivityGroup
ActivityManager

intent	The intent to start.
	onActivityResult() when the activity exits.
options	Additional options for how the Activity should be started. See Context.startActivity(Intent, Bundle) for more details.

Throws

android.content.ActivityNotFoundException

See Also

startActivity(Intent)

public void startActivityFromChild (Activity child, Intent intent, int requestCode, Bundle options) Added in API level 16

This is called when a child activity of this one calls its `startActivity(Intent)` (/reference/android/app/Activity.html#startActivity(android.content.Intent)) or `startActivityForResult(Intent, int)` (/reference/android/app/Activity.html#startActivityForResult(android.content.Intent, int)) method.

This method throws `ActivityNotFoundException` (/reference/android/content/ActivityNotFoundException.html) if there was no Activity found to run the given Intent.

Parameters

child	The activity making the call.
intent	The intent to start.
requestCode	Reply request code. < 0 if reply is not requested.
options	Additional options for how the Activity should be started. See `Context.startActivity(Intent, Bundle)` for more details.

Throws

android.content.ActivityNotFoundException

See Also

`startActivity(Intent)`
`startActivityForResult(Intent, int)`

public void startActivityFromChild (Activity child,

Android APIs API level: 19

android
android.accessibilityservice
android.accounts
android.animation
android.app
android.app.admin
android.app.backup
android.appwidget
android.bluetooth
android.content
android.content.pm
android.content.res
android.database

Interfaces

ActionBar.OnMenuVisibilityListener
ActionBar.OnNavigationListener
ActionBar.TabListener
Application.ActivityLifecycleCallbacks
Application.OnProvideAssistDataListener
AppOpsManager.OnOpChangedListener
DatePickerDialog.OnDateSetListener
FragmentBreadCrumbs.OnBreadCrumbClickListener
FragmentManager.BackStackEntry
FragmentManager.OnBackStackChangedListener
KeyguardManager.OnKeyguardExitResult
LoaderManager.LoaderCallbacks
PendingIntent.OnFinished
SearchManager.OnCancelListener
SearchManager.OnDismissListener
TimePickerDialog.OnTimeSetListener
UiAutomation.AccessibilityEventFilter
UiAutomation.OnAccessibilityEventListener

Classes

ActionBar
ActionBar.LayoutParams
ActionBar.Tab
Activity
ActivityGroup

Intent intent, int requestCode) Added in API level 1

Android APIs API level: 19

`Intent, int, Bundle)` (/reference/android/app/Activity.html#startActivityFromChild(android.app.Activity, android.content.Intent, int, android.os.Bundle)) with no options.

Parameters

child	The activity making the call.
intent	The intent to start.
requestCode	Reply request code. < 0 if reply is not requested.

Throws

android.content.ActivityNotFoundException

See Also

`startActivityForResult(Intent, int)`

public void **startActivityFromFragment** (Fragment fragment, Intent intent, int requestCode, Bundle options) Added in API level 16

This is called when a Fragment in this activity calls its `startActivity(Intent)` (/reference/android/app/Fragment.html#startActivity(android.content.Intent)) or `startActivityForResult(Intent, int)` (/reference/android/app/Fragment.html#startActivityForResult(android.content.Intent, int)) method.

This is the same as calling `startActivityForResult(Intent, int)` (/reference/android/content/ActivityNotFoundException.html) if the e was no Activity found to run the given Intent.

Parameters

fragment	The fragment making the call.
intent	The intent to start
requestCode	Reply request code. < 0 if reply is not requested.
options	Additional options for how the Activity should be started. See `Context.startActivity(Intent, Bundle)` for more details.

Throws

android.content.ActivityNotFoundException

Interfaces

ActionBar.OnMenuVisibilityListener
ActionBar.OnNavigationListener
ActionBar.TabListener
Application.ActivityLifecycleCallbacks
Application.OnProvideAssistDataListener
AppOpsManager.OnOpChangedListener
DatePickerDialog.OnDateSetListener
FragmentBreadCrumbs.OnBreadCrumbClickListener
FragmentManager.BackStackEntry
FragmentManager.OnBackStackChangedListener
KeyguardManager.OnKeyguardExitResult
LoaderManager.LoaderCallbacks
PendingIntent.OnFinished
SearchManager.OnCancelListener
SearchManager.OnDismissListener
TimePickerDialog.OnTimeSetListener
UiAutomation.AccessibilityEventFilter
UiAutomation.OnAccessibilityEventListener

Classes

ActionBar
ActionBar.LayoutParams
ActionBar.Tab
Activity
ActivityGroup
ActivityManager

See Also

Android APIs API level: 19

startActivityForResult(Intent, int)

android

android.accessibilityservice

public void **startActivityFromFragment** (Fragment fragment, Intent intent, int requestCode) Added in API level 11

android.accounts

android.animation

android.app

Initiate a ragment, Intent, int, Bundle) (/reference/android/app/Activity.html#startActivityFromFragment(android.app.Fragment, android.content.Intent, int, android.os.Bundle)) with no options.

android.app.admin

android.app.backup

android.appwidget

android.bluetooth

android.content

Parameters

android.content.pm

fragment The fragment making the call.

android.content.res

intent The intent to start.

android.database

android.database.sqlite *requestCode* If >= 0, this code will be returned in onActivityResult() when the activity exits.

Throws

android.content.ActivityNotFoundException

Interfaces

See Also

ActionBar.OnMenuVisibilityListener

ActionBar.OnNavigationListener

startActivity(Intent)

ActionBar.TabListener

startActivityForResult(Intent, int)

Application.ActivityLifecycleCallbacks

Application.OnProvideAssistDataListener

public boolean **startActivityIfNeeded** (Intent intent, int requestCode, Bundle options) Added in API level 16

AppOpsManager.OnOpChangedListener

DatePickerDialog.OnDateSetListener

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) (/reference/android/app/Activity.html#startActivityForResult(android.content.Intent, int)) except: if you are using the FLAG_ACTIVITY_SINGLE_TOP (/reference/android/content/Intent.html#FLAG_ACTIVITY_SINGLE_TOP) flag, or the singleTop launchMode (/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) (/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

Parameters

FragmentBreadCrumbs.OnBreadCrumbClickListener

FragmentManager.BackStackEntry

FragmentManager.OnBackStackChangedListener

KeyguardManager.OnKeyguardExitResult

LoaderManager.LoaderCallbacks

PendingIntent.OnFinished

SearchManager.OnCancelListener

SearchManager.OnDismissListener

TimePickerDialog.OnTimeSetListener

UiAutomation.AccessibilityEventFilter

UiAutomation.OnAccessibilityEventListener

Classes

ActionBar

ActionBar.LayoutParams

ActionBar.Tab

Activity

ActivityGroup

ActivityManager

intent The intent to start.

Android APIs API level: 19

onActivityResult() when the activity exits, as described in `startActivityForResult(Intent, int)`.

options Additional options for how the Activity should be started. See `Context.startActivity(Intent, Bundle)` for more details.

Returns

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)`
`startActivityForResult(Intent, int)`

public boolean **startActivityIfNeeded** (Intent intent, int requestCode)

Added in API level 1

Same as calling `startActivityIfNeeded(Intent, int, Bundle)` (/reference/android/app/Activity.html#startActivityIfNeeded(android.content.Intent, int, android.os.Bundle)) with no options.

Parameters

intent The intent to start.

requestCode If >= 0, this code will be returned in onActivityResult() when the activity exits, as described in `startActivityForResult(Intent, int)`.

Returns

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)`
`startActivityForResult(Intent, int)`

public void **startIntentSender** (IntentSender intent, Intent fillInIntent, int flagsMask, int flagsValues, int extraFlags, Bundle options)

Added in API level 16

Interfaces

ActionBar.OnMenuVisibilityListener
ActionBar.OnNavigationListener
ActionBar.TabListener
Application.ActivityLifecycleCallbacks
Application.OnProvideAssistDataListener
AppOpsManager.OnOpChangedListener
DatePickerDialog.OnDateSetListener
FragmentBreadCrumbs.OnBreadCrumbClickListener
FragmentManager.BackStackEntry
FragmentManager.OnBackStackChangedListener
KeyguardManager.OnKeyguardExitResult
LoaderManager.LoaderCallbacks
PendingIntent.OnFinished
SearchManager.OnCancelListener
SearchManager.OnDismissListener
TimePickerDialog.OnTimeSetListener
UiAutomation.AccessibilityEventFilter
UiAutomation.OnAccessibilityEventListener

Classes

ActionBar
ActionBar.LayoutParams
ActionBar.Tab
Activity
ActivityGroup
ActivityManager

/app/Activity.html#startActivity(android.content.Intent,

Android APIs API level: 19

android
android.accessibilityservice
android.accounts
android.animation
android.app
android.app.admin
android.app.backup
android.appwidget
android.bluetooth
android.content
android.content.pm
android.content.res
android.database
android.database.sqlite

see

startIntentSender(IntentSender, Intent, int, int, int, Bundle) (/reference/android/app/Activity.html#startIntentSenderForResult(android.content.IntentSender, android.content.Intent, int, int, int, android.os.Bundle))

for more information.

Parameters

intent The IntentSender to launch.

fillInIntent If non-null, this will be provided as the intent parameter to sendIntent(Context, int, Intent, IntentSender.OnFinished, Handler).

flagsMask Intent flags in the original IntentSender that you

flagsValues Desired values for any bits set in *flagsMask*

extraFlags Always set to 0.

options Additional options for how the Activity should be started. See Context.startActivity(Intent, Bundle) for more details. If options have also been supplied by the IntentSender, options given here will override any that conflict with those given by the IntentSender.

Throws

IntentSender.SendIntentException

Interfaces

ActionBar.OnMenuVisibilityListener
ActionBar.OnNavigationListener
ActionBar.TabListener
Application.ActivityLifecycleCallbacks
Application.OnProvideAssistDataListener
AppOpsManager.OnOpChangedListener
DatePickerDialog.OnDateSetListener
FragmentBreadCrumbs.OnBreadCrumbClickListener
FragmentManager.BackStackEntry
FragmentManager.OnBackStackChangedListener
KeyguardManager.OnKeyguardExitResult
LoaderManager.LoaderCallbacks
PendingIntent.OnFinished
SearchManager.OnCancelListener
SearchManager.OnDismissListener
TimePickerDialog.OnTimeSetListener
UiAutomation.AccessibilityEventFilter
UiAutomation.OnAccessibilityEventListener

public void startIntentSender (IntentSender intent, Intent fillInIntent, int flagsMask, int flagsValues, int extraFlags)

Added in API level 5

Same as calling startIntentSender(IntentSender, Intent, int, int, int, Bundle) (/reference/android/app/Activity.html#startIntentSenderForResult(android.content.IntentSender, android.content.Intent, int, int, int, android.os.Bundle)) with no options.

Classes

Parameters

ActionBar
ActionBar.LayoutParams
ActionBar.Tab
Activity
ActivityGroup
ActivityManager

intent The IntentSender to launch.

fillInIntent If non-null, this will be provided as the intent

t, int, Intent, IntentSender.OnFinished,

Handler).

Android APIs API level: 19

android
android.accessibilityservice
android.accounts
android.animation
android.app
android.app.admin
android.app.backup
android.appwidget
android.bluetooth
android.content
android.content.pm
android.content.res
android.database
android.database.sqlite

ler that you would like to change.

flagsValues Desired values for any bits set in *flagsMask*

extraFlags Always set to 0.

Throws

IntentSender.SendIntentException

public void **startIntentSenderForResult**(IntentSender intent, int requestCode, Intent fillInIntent, int flagsMask, int flagsValues, int extraFlags, Bundle options)

Added in API level 16

Like `startActivityForResult(Intent, int)` (/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)` (/reference/android/app/Activity.html#startActivityForResult(android.content.Intent, int)) here, in which case no additional action will be executed (such as sending a broadcast) as if you had called `IntentSender.sendIntent` (/reference/android/content/IntentSender.html#sendIntent(android.content.Context, int, android.content.Intent, android.content.IntentSender.OnFinished, android.os.Handler)) on it.

Parameters

intent The IntentSender to launch.

requestCode If >= 0, this code will be returned in onActivityResult() when the activity exits.

fillInIntent If non-null, this will be provided as the intent parameter to `sendIntent(Context, int, Intent, IntentSender.OnFinished, Handler)`.

flagsMask Intent flags in the original IntentSender that you would like to change.

flagsValues Desired values for any bits set in *flagsMask*

extraFlags Always set to 0.

Interfaces

ActionBar.OnMenuVisibilityListener
ActionBar.OnNavigationListener
ActionBar.TabListener
ApplicationErrorReport.CrashInfo
Application.ActivityLifecycleCallbacks
Application.OnProvideAssistDataListener
AppOpsManager.OnOpChangedListener
DatePickerDialog.OnDateSetListener
FragmentBreadCrumbs.OnBreadCrumbClickListener
FragmentManager.BackStackEntry
FragmentManager.OnBackStackChangedListener
KeyguardManager.OnKeyguardExitResult
LoaderManager.LoaderCallbacks
PendingIntent.OnFinished
SearchManager.OnCancelListener
SearchManager.OnDismissListener
TimePickerDialog.OnTimeSetListener
UiAutomation.AccessibilityEventFilter
UiAutomation.OnAccessibilityEventListener

Classes

ActionBar
ActionBar.LayoutParams
ActionBar.Tab
Activity
ActivityGroup
ActivityManager

options Additional options for how the Activity should

Android APIs API level: 19

android
android.accessibilityservice
android.accounts
android.animation
android.app
android.app.admin
android.app.backup
android.appwidget
android.bluetooth
android.content
android.content.pm
android.content.res
android.database
android.database.sqlite

`Bundle)` for more details. If options have also been supplied by the IntentSender, options given here will override any that conflict with those given by the IntentSender.

Throws

IntentSender.SendIntentException

public void **startIntentSenderForResult** (IntentSender intent, int requestCode, Intent fillInIntent, int flagsMask, int flagsValues, int extraFlags)

Added in API level 5

`startIntentSenderForResult(IntentSender, int, Intent, int, int, int, Bundle)` (/reference/android/app/Activity.html#startIntentSenderForResult(android.content.IntentSender, int, android.content.Intent, int, int, int, android.os.Bundle)) with no options.

Parameters

intent The IntentSender to launch.

requestCode If >= 0, this code will be returned in onActivityResult() when the activity exits.

fillInIntent If non-null, this will be provided as the intent parameter to `sendIntent(Context, int, Intent, IntentSender.OnFinished, Handler)`.

flagsMask Intent flags in the original IntentSender that you would like to change.

flagsValues Desired values for any bits set in *flagsMask*

extraFlags Always set to 0.

Throws

IntentSender.SendIntentException

Interfaces

ActionBar.OnMenuVisibilityListener
ActionBar.OnNavigationListener
ActionBar.TabListener
Application.ActivityLifecycleCallbacks
Application.OnProvideAssistDataListener
AppOpsManager.OnOpChangedListener
DatePickerDialog.OnDateSetListener
FragmentBreadCrumbs.OnBreadCrumbClickListener
FragmentManager.BackStackEntry
FragmentManager.OnBackStackChangedListener
KeyguardManager.OnKeyguardExitResult
LoaderManager.LoaderCallbacks
PendingIntent.OnFinished
SearchManager.OnCancelListener
SearchManager.OnDismissListener
TimePickerDialog.OnTimeSetListener
UiAutomation.AccessibilityEventFilter
UiAutomation.OnAccessibilityEventListener

Classes

ActionBar
ActionBar.LayoutParams
ActionBar.Tab
Activity
ActivityGroup
ActivityManager

public void **startIntentSenderFromChild** (Activity child, IntentSender intent, int requestCode, Intent fillInIntent, int flagsMask, int flagsValues, int

Added in API level 5

Same as calling `startIntentSenderFromChild(Activity,`

IntentSender, int, Intent, int, int, int, Bundle)

Android APIs API level: 19

/app/Activity.html#startIntentSenderFromChild(android.app.Activity, android.content.IntentSender, int, android.content.Intent, int, int, int, android.os.Bundle) with no options.

Throws

IntentSender.SendIntentException

public void **startManagingCursor** (Cursor c) Added in API level 1

This method was deprecated in API level 11.

Use the new `CursorLoader` (/reference/android/content/CursorLoader.html) class with `LoaderManager` (/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` (/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()` (/reference/android/database/Cursor.html#deactivate()) on the given Cursor, and when it is later restarted it will call `requery()` (/reference/android/database/Cursor.html#requery()) for you. When the activity is destroyed, all

are targeting HONEYCOMB (/reference/android

/os/Build.VERSION_CODES.html#HONEYCOMB) or later, consider instead using

Android APIs API level: 19 *er.html) instead,*

available via getLoaderManager() (/reference/and oid

/app/Activity.html#getLoaderManager()).

android
android.accessibilityservice
android.accounts
android.animation
android.app
android.app.admin
android.app.backup
android.appwidget
android.bluetooth
android.content
android.content.pm
android.content.res
android.database
android.database.sqlite

Warning: Do not call `close()` (/reference/android/database/Cursor.html#close()) on cursor obtained from `managedQuery(Uri, ...)` (/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)` (/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()` (/reference/android/database/Cursor.html#close())

Parameters

cursor	The Cursor to be managed.

See Also

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

`stopManagingCursor(Cursor)`

Interfaces

ActionBar.OnMenuVisibilityListener
ActionBar.OnNavigationListener
ActionBar.TabListener
Application.ActivityLifecycleCallbacks
Application.OnProvideAssistDataListener
AppOpsManager.OnOpChangedListener
DatePickerDialog.OnDateSetListener
FragmentBreadCrumbs.OnBreadCrumbClickListener
FragmentManager.BackStackEntry
FragmentManager.OnBackStackChangedListener
KeyguardManager.OnKeyguardExitResult
LoaderManager.LoaderCallbacks
PendingIntent.OnFinished
SearchManager.OnCancelListener
SearchManager.OnDismissListener
TimePickerDialog.OnTimeSetListener
UiAutomation.AccessibilityEventFilter
UiAutomation.OnAccessibilityEventListener

Classes

ActionBar
ActionBar.LayoutParams
ActionBar.Tab
Activity
ActivityGroup
ActivityManager

public boolean **startNextMatchingActivity** (Intent intent)

Added in API level 1

Same as calling `startNextMatchingActivity(Intent, Bundle)` (/reference/android/app/Activity.html#startNextMatchingActivity(android.content.Intent, android.os.Bundle)) with no options.

Parameters

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.

Returns

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.

public boolean **startNextMatchingActivity** (Intent

Android APIs API level: 19 Added in API level 16

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)` (/reference/android/app/Activity.html#onCreate(android.os.Bundle)) with the Intent returned by `getIntent()` (/reference/android/app/Activity.html#getIntent()).

Parameters

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 Additional options for how the Activity should be started. See `Context.startActivity(Intent, Bundle)` for more details.

Returns

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.

public void **startSearch** (String initialQuery, boolean selectInitialQuery, Bundle appSearchData, boolean globalSearch)

Added in API level 1

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 overriden 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

initialQuery Any non-null non-empty string will be inserted as pre-entered text in the search query box.

selectInitialQuery If true, the intial query will be preselected, which means that any further typing will replace it. This is useful for cases where an

Interfaces

ActionBar.OnMenuVisibilityListener
ActionBar.OnNavigationListener
ActionBar.TabListener
Application.ActivityLifecycleCallbacks
Application.OnProvideAssistDataListener
AppOpsManager.OnOpChangedListener
DatePickerDialog.OnDateSetListener
FragmentBreadCrumbs.OnBreadCrumbClickListener
FragmentManager.BackStackEntry
FragmentManager.OnBackStackChangedListener
KeyguardManager.OnKeyguardExitResult
LoaderManager.LoaderCallbacks
PendingIntent.OnFinished
SearchManager.OnCancelListener
SearchManager.OnDismissListener
TimePickerDialog.OnTimeSetListener
UiAutomation.AccessibilityEventFilter
UiAutomation.OnAccessibilityEventListener

Classes

ActionBar
ActionBar.LayoutParams
ActionBar.Tab
Activity
ActivityGroup
ActivityManager

Android APIs API level: 19

android
android.accessibilityservice
android.accounts
android.animation
android.app
android.app.admin
android.app.backup
android.appwidget
android.bluetooth
android.content
android.content.pm
android.content.res
android.database
android.database.sqlite

entire pre-formed query is being inserted. If placed at the end of the inserted query. This is useful when the inserted query is text that the user entered, and the user would expect to be able to keep typing. *This parameter is only meaningful if initialQuery is a non-empty*

appSearchData

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.

globalSearch

If false, this will only launch the search that has been specifically defined by the application (which is usually defined as a ch 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
- onSearchRequested()

Interfaces

ActionBar.OnMenuVisibilityListener
ActionBar.OnNavigationListener
ActionBar.TabListener
Application.ActivityLifecycleCallbacks
Application.OnProvideAssistDataListener
AppOpsManager.OnOpChangedListener
DatePickerDialog.OnDateSetListener
FragmentBreadCrumbs.OnBreadCrumbClickListener
FragmentManager.BackStackEntry
FragmentManager.OnBackStackChangedListener
KeyguardManager.OnKeyguardExitResult
LoaderManager.LoaderCallbacks
PendingIntent.OnFinished
SearchManager.OnCancelListener
SearchManager.OnDismissListener
TimePickerDialog.OnTimeSetListener
UiAutomation.AccessibilityEventFilter
UiAutomation.OnAccessibilityEventListener

Classes

ActionBar
ActionBar.LayoutParams
ActionBar.Tab
Activity
ActivityGroup
ActivityManager

public void stopManagingCursor (Cursor c)

Added in API level 1

This method was deprecated in API level 11.
Use the new CursorLoader (/reference/android/content/CursorLoader.html) class with LoaderManager (/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) (/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() (/reference/android/database/Cursor.html#close()).

Parameters

- c: The Cursor that was being managed.

See Also

- startManagingCursor(Cursor)

`startManagingCursor(Cursor)`

Android APIs API level: 19

public void **takeKeyEvents** (boolean get) Added in API level 1

Request that key events come to this activity. Use th s if your activity has no views with focus, but the activity still wants a chance to process key events.

See Also

`takeKeyEvents(boolean)`

public void **triggerSearch** (String query, Bundle appSearchData) Added in API level 5

Similar to `startSearch(String, boolean, Bundle, boolean)`, but actually fires off the search query after invoking the search dialog. Made available for testing purposes.

Parameters

query The query to trigger. If empty, the request will be ignored.

appSearchData 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.

public void **unregisterForContextMenu** (View view) Added in API level 1

Prevents a context menu to be shown for the given view. This method will remove the `View.OnCreateContextMenuListener` (/reference/android/view/View.OnCreateContextMenuListener.html) on the view.

Parameters

view The view that should stop showing a context menu.

See Also

`registerForContextMenu(View)`

Interfaces

ActionBar.OnMenuVisibilityListener
ActionBar.OnNavigationListener
ActionBar.TabListener
Application.ActivityLifecycleCallbacks
Application.OnProvideAssistDataListener
AppOpsManager.OnOpChangedListener
DatePickerDialog.OnDateSetListener
FragmentBreadCrumbs.OnBreadCrumbClickListener
FragmentManager.BackStackEntry
FragmentManager.OnBackStackChangedListener
KeyguardManager.OnKeyguardExitResult
LoaderManager.LoaderCallbacks
PendingIntent.OnFinished
SearchManager.OnCancelListener
SearchManager.OnDismissListener
TimePickerDialog.OnTimeSetListener
UiAutomation.AccessibilityEventFilter
UiAutomation.OnAccessibilityEventListener

Classes

ActionBar
ActionBar.LayoutParams
ActionBar.Tab
Activity
ActivityGroup
ActivityManager

protected void **onActivityResult** (int requestCode, int

Android APIs API level: 19 Added in API level 1

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 (/reference/android/app/Activity.html#RESULT_CANCELED) if the activity ...shed during its operation.

You will receive this call immediately before onResume() when your activity is re-starting.

Parameters

requestCode The integer request code originally supplied to startActivityForResult(), allowing you to identify who this result came from.

resultCode The integer result code returned by the child activity through its setResult().

data An Intent, which can return result data to the caller (various data can be attached to Intent "extras").

See Also

- startActivityFromChild(Activity, Intent, int)
- createPendingResult(int, Intent, int)
- setResult(int)

protected void **onApplyThemeResource** (Resources.Theme theme, int resid, boolean first) Added in API level 1

Called by setTheme(int) (/reference/android/view/ContextThemeWrapper.html#setTheme(int)) and getTheme() (/reference/android/view/ContextThemeWrapper.html#getTheme()) to apply a theme resource to the current Theme object. Can override to change the default (simple) behavior. This method will not be called in multiple threads simultaneously.

Parameters

theme The Theme object being modified.

resid The theme style resource being applied to *theme*.

first Set to true if this is the first time a style is being applied to *theme*.

protected void **onChildTitleChanged** (Activity

Interfaces

ActionBar.OnMenuVisibilityListener
ActionBar.OnNavigationListener
ActionBar.TabListener
Application.ActivityLifecycleCallbacks
Application.OnProvideAssistDataListener
AppOpsManager.OnOpChangedListener
DatePickerDialog.OnDateSetListener
FragmentBreadCrumbs.OnBreadCrumbClickListener
FragmentManager.BackStackEntry
FragmentManager.OnBackStackChangedListener
KeyguardManager.OnKeyguardExitResult
LoaderManager.LoaderCallbacks
PendingIntent.OnFinished
SearchManager.OnCancelListener
SearchManager.OnDismissListener
TimePickerDialog.OnTimeSetListener
UiAutomation.AccessibilityEventFilter
UiAutomation.OnAccessibilityEventListener

Classes

ActionBar
ActionBar.LayoutParams
ActionBar.Tab
Activity
ActivityGroup
ActivityManager

childActivity, CharSequence title) Added in API level 1

Android APIs API level: 19

protected void **onCreate** (Bundle savedInstanceState) Added in API level 1

android
android.accessibilityservice
android.accounts
android.animation
android.app
android.appwidget
android.backup
android.bluetooth
android.content
android.content.pm
android.content.res
android.database
android.database.sqlite

Called when the activity is starting. This is where most initialization should go: calling `setContentView(int)` (/reference/android/app/Activity.html#setContentView(int)) to inflate the activity's UI, using `findViewById(int)` (/reference/android/app/Activity.html#findViewById(int)) to programmatically interact with widgets in the UI, calling `managedQuery(android.net.Uri, String[], String, String[], String)` (/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()` (/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()` (/reference/android/app/Activity.html#onStart()), `onResume()` (/reference/android/app/Activity.html#onResume()), `onPause()` (/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

savedInstanceState 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)`. ***Note: Otherwise it is null.***

See Also

`onStart()`
`onSaveInstanceState(Bundle)`
`onRestoreInstanceState(Bundle)`
`onPostCreate(Bundle)`

Interfaces

ActionBar.OnMenuVisibilityListener
ActionBar.OnNavigationListener
ActionBar.TabListener
Application.ActivityLifecycleCallbacks
Application.OnProvideAssistDataListener
AppOpsManager.OnOpChangedListener
DatePickerDialog.OnDateSetListener
FragmentBreadCrumbs.OnBreadCrumbClickListener
FragmentManager.BackStackEntry
FragmentManager.OnBackStackChangedListener
KeyguardManager.OnKeyguardExitResult
LoaderManager.LoaderCallbacks
PendingIntent.OnFinished
SearchManager.OnCancelListener
SearchManager.OnDismissListener
TimePickerDialog.OnTimeSetListener
UiAutomation.AccessibilityEventFilter
UiAutomation.OnAccessibilityEventListener

Classes

ActionBar
ActionBar.LayoutParams
ActionBar.Tab
Activity
ActivityGroup
ActivityManager

protected Dialog **onCreateDialog** (int id) Added in API level 1

This method was deprecated in API level 8.

Old no arguments version of `onCreateDialog(int, Bundle)`

(/reference/android/app/Activity.html#onCreateDialog(int,

Android APIs API level: 19

android
android.accessibilityservice
android.accounts
android.animation
android.app
android.app.admin
android.app.backup
android.appwidget
android.bluetooth
android.content
android.content.pm
android.content.res
android.database
android.database.sqlite

protected Dialog **onCreateDialog** (int id, Bundle args)

Added in API level 8

This method was deprecated in API level 13.

/app/DialogFragment.html) class with FragmentManager (/reference/android/app/FragmentManager.html) instead; this is also available on older platforms through the Android compatibility package.

Callback for creating dialogs that are managed (saved and restored) for you by the activity. The default implementation calls through to onCreateDialog(int) (/reference/android/app/Activity.html#onCreateDialog(int)) for compatibility. *If you are targeting HONEYCOMB (/reference/android/os/Build.VERSION_CODES.html#HONEYCOMB) or later, consider instead using a DialogFragment (/reference/android/app/DialogFragment.html) instead.*

If you use showDialog(int) (/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) (/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) (/reference/android/app/Activity.html#onPrepareDialog(int, android.app.Dialog, android.os.Bundle)).

Parameters

id	The id of the dialog.
args	The dialog arguments provided to showDialog(int, Bundle).

Returns

The dialog	If you return null, the dialog will not be created.

See Also

Interfaces

ActionBar.OnMenuVisibilityListener
ActionBar.OnNavigationListener
ActionBar.TabListener
Application.ActivityLifecycleCallbacks
Application.OnProvideAssistDataListener
AppOpsManager.OnOpChangedListener
DatePickerDialog.OnDateSetListener
FragmentBreadCrumbs.OnBreadCrumbClickListener
FragmentManager.BackStackEntry
FragmentManager.OnBackStackChangedListener
KeyguardManager.OnKeyguardExitResult
LoaderManager.LoaderCallbacks
PendingIntent.OnFinished
SearchManager.OnCancelListener
SearchManager.OnDismissListener
TimePickerDialog.OnTimeSetListener
UiAutomation.AccessibilityEventFilter
UiAutomation.OnAccessibilityEventListener

Classes

ActionBar
ActionBar.LayoutParams
ActionBar.Tab
Activity
ActivityGroup
ActivityManager

`onPrepareDialog(int, Dialog, Bundle)`

Android APIs API level: 19

`removeDialog(int)`

android
android.accessibilityservice
android.accounts
android.animation
android.app
android.app.admin
android.app.backup
android.appwidget
android.bluetooth
android.content
android.content.pm
android.content.res
android.database
android.database.sqlite
android.graphics
android.graphics.drawable
android.graphics.drawable.shapes
android.hardware
android.hardware.display
android.hardware.input
android.hardware.usb
android.inputmethodservice
android.location
android.media
android.media.audiofx
android.media.effect
android.media.midi
android.media.tv
android.net
android.net.http
android.net.nsd
android.net.rtp
android.net.sip
android.net.wifi
android.net.wifi.p2p
android.nfc
android.nfc.cardemulation
android.nfc.tech
android.opengl
android.os
android.os.storage
android.preference
android.print
android.print.pdf
android.printservice
android.provider
android.renderscript
android.sax
android.service.dreams
android.service.notification
android.service.textservice
android.service.wallpaper
android.speech
android.speech.tts
android.support.v13.app
android.support.v4.app
android.support.v4.content
android.support.v4.content.pm
android.support.v4.content.res
android.support.v4.graphics.drawable
android.support.v4.os
android.support.v4.print
android.support.v4.provider
android.support.v4.view
android.support.v4.view.accessibility
android.support.v4.widget
android.system
android.test
android.test.mock
android.test.suitebuilder
android.test.suitebuilder.annotation
android.text
android.text.format
android.text.method
android.text.style
android.text.util
android.util
android.view
android.view.accessibility
android.view.animation
android.view.inputmethod
android.view.textservice
android.webkit
android.widget
dalvik.annotation
dalvik.bytecode
dalvik.system
java.awt.font
java.beans
java.io
java.lang
java.lang.annotation
java.lang.ref
java.lang.reflect
java.math
java.net
java.nio
java.nio.channels
java.nio.channels.spi
java.nio.charset
java.nio.charset.spi
java.security
java.security.acl
java.security.cert
java.security.interfaces
java.security.spec
java.sql
java.text
java.text.spi
java.util
java.util.concurrent
java.util.concurrent.atomic
java.util.concurrent.locks
java.util.jar
java.util.logging
java.util.prefs
java.util.regex
java.util.zip
javax.crypto
javax.crypto.interfaces
javax.crypto.spec
javax.microedition.khronos.egl
javax.microedition.khronos.opengles
javax.net
javax.net.ssl
javax.security.auth
javax.security.auth.callback
javax.security.auth.kerberos
javax.security.auth.login
javax.security.auth.spi
javax.security.auth.x500
javax.security.cert
javax.sql
javax.xml.datatype
javax.xml.parsers
javax.xml.transform
javax.xml.transform.dom
javax.xml.transform.sax
javax.xml.transform.stream
javax.xml.validation
javax.xml.xpath
junit.framework
org.apache.http.conn.scheme
org.apache.http.conn.ssl
org.apache.http.cookie
org.apache.http.entity
org.apache.http.message
org.apache.http.params
org.apache.http.protocol
org.apache.http.util
org.json
org.w3c.dom
org.w3c.dom.bootstrap
org.w3c.dom.ls
org.xml.sax
org.xml.sax.ext
org.xml.sax.helpers
org.xmlpull.v1
org.xmlpull.v1.sax2
org.xmlpull.v1.util
org.xmlpull.v1.wrapper

Interfaces

ActionBar.OnMenuVisibilityListener
ActionBar.OnNavigationListener
ActionBar.TabListener
Application.ActivityLifecycleCallbacks
Application.OnProvideAssistDataListener
AppOpsManager.OnOpChangedListener
DatePickerDialog.OnDateSetListener
FragmentBreadCrumbs.OnBreadCrumbClickListener
FragmentManager.BackStackEntry
FragmentManager.OnBackStackChangedListener
KeyguardManager.OnKeyguardExitResult
LoaderManager.LoaderCallbacks
PendingIntent.OnFinished
SearchManager.OnCancelListener
SearchManager.OnDismissListener
TimePickerDialog.OnTimeSetListener
UiAutomation.AccessibilityEventFilter
UiAutomation.OnAccessibilityEventListener

Classes

ActionBar
ActionBar.LayoutParams
ActionBar.Tab
Activity
ActivityGroup
ActivityManager

protected void **onDestroy** ()

Added in API level 1

yed. This can
happen either because the activity is finishing (some one called `finish()` (/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()` (/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()` (/reference/android/app/Activity.html#onPause()) or `onSaveInstanceState(Bundle)` (/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()`
`onStop()`
`finish()`
`isFinishing()`

protected void **onNewIntent** (Intent intent)

Added in API level 1

This is called for activities that set launchMode to "singleTop" in their package, or if a client used the `FLAG_ACTIVITY_SINGLE_TOP` (/reference/android/content/Intent.html#FLAG_ACTIVITY_SINGLE_TOP) flag when calling `startActivity(Intent)` (/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, with the Intent

Android APIs API level: 19

android
android.accessibilityservice
android.accounts
android.animation
android.app
android.app.admin
android.app.backup
android.appwidget
android.bluetooth
android.content
android.content.pm
android.content.res
android.database
android.database.sqlite

An activity will always be paused before receiving a new intent, so you can count on `onResume()` (/reference/android/app/Activity.html#onResume()) being called after this method.

Note that `getIntent()` (/reference/android/app/Activity.html#getIntent()) still returns the original Intent. You can use `setIntent(Intent)` (/reference/android/app/Activity.html#setIntent(android.content.Intent)) to update it to this new Intent.

Parameters

intent The new intent that was started for the activity.

See Also

`getIntent()`
`setIntent(Intent)`
`onResume()`

Interfaces

ActionBar.OnMenuVisibilityListener
ActionBar.OnNavigationListener
ActionBar.TabListener
Application.ActivityLifecycleCallbacks
Application.OnProvideAssistDataListener
AppOpsManager.OnOpChangedListener
DatePickerDialog.OnDateSetListener
FragmentBreadCrumbs.OnBreadCrumbClickListener
FragmentManager.BackStackEntry
FragmentManager.OnBackStackChangedListener
KeyguardManager.OnKeyguardExitResult
LoaderManager.LoaderCallbacks
PendingIntent.OnFinished
SearchManager.OnCancelListener
SearchManager.OnDismissListener
TimePickerDialog.OnTimeSetListener
UiAutomation.AccessibilityEventFilter
UiAutomation.OnAccessibilityEventListener

protected void **onPause** () Added in API level 1

Called as part of the activity lifecycle when an activity is going into the background, but has not (yet) been killed. The counterpart to `onResume()` (/reference/android/app/Activity.html#onResume()).

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()` (/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)` (/reference/android/app/Activity.html#onSaveInstanceState(android.os.Bundle)) is

Classes

ActionBar
ActionBar.LayoutParams
ActionBar.Tab
Activity
ActivityGroup
ActivityManager

used to save per-instance state in the activity and this method is
lers, files, etc.)

Android APIs API level: 19

After receiving this call you will usually receive a follo wing call to
onStop() (/reference/android/app/Activity.html#onStop()) (after the
next activity has been resumed and displayed), howe ver in some
cases there will be a direct call back to onResume() (/reference
hrough 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()
- onSaveInstanceState(Bundle)
- onStop()

protected void **onPostCreate** (Bundle savedInstanceState) Added in API level 1

Called when activity start-up is complete (after onSt art()
(/reference/android/app/Activity.html#onStart()) and
onRestoreInstanceState(Bundle) (/reference/android
/app/Activity.html#onRestoreInstanceState(android.os.B undle)) have
been called). Applications will generally not impleme nt this method;
it is intended for system classes to do final initializat ion 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

savedInstanceState If the activity is being re-initi alized after
previously being shut down then this
Bundle contains the data it most
recently supplied in
onSaveInstanceState(B undle).
Note: Otherwise it is null.

See Also

- onRestoreInstanceState(Bundle)

protected void **onPostResume** () Added in API level 1

sume()
(/reference/android/app/Activity.html#onResume()) has been called).

Interfaces

- ActionBar.OnMenuVisibilityListener
- ActionBar.OnNavigationListener
- ActionBar.TabListener
- Application.ActivityLifecycleCallbacks
- Application.OnProvideAssistDataListener
- AppOpsManager.OnOpChangedListener
- DatePickerDialog.OnDateSetListener
- FragmentBreadCrumbs.OnBreadCrumbClickListener
- FragmentManager.BackStackEntry
- FragmentManager.OnBackStackChangedListener
- KeyguardManager.OnKeyguardExitResult
- LoaderManager.LoaderCallbacks
- PendingIntent.OnFinished
- SearchManager.OnCancelListener
- SearchManager.OnDismissListener
- TimePickerDialog.OnTimeSetListener
- UiAutomation.AccessibilityEventFilter
- UiAutomation.OnAccessibilityEventListener

Classes

- ActionBar
- ActionBar.LayoutParams
- ActionBar.Tab
- Activity
- ActivityGroup
- ActivityManager

Applications will generally not implement this method; it is intended
resume code

Android APIs API level: 19

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

protected void **onPrepareDialog** (int id, Dialog dialog) Added in API level 1

This method was deprecated in API level 8.
Old no-arguments version of `onPrepareDialog(int, Dialog, Bundle)` (/reference/android/app/Activity.html#onPrepareDialog(int, android.app.Dialog, android.os.Bundle)).

protected void **onPrepareDialog** (int id, Dialog dialog, Bundle args) Added in API level 8

This method was deprecated in API level 13.
Use the new `DialogFragment` (/reference/android/app/DialogFragment.html) class with `FragmentManager` (/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)` (/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 The id of the managed dialog.

dialog The dialog.

args The dialog arguments provided to

See Also

Interfaces

ActionBar.OnMenuVisibilityListener
ActionBar.OnNavigationListener
ActionBar.TabListener
Application.ActivityLifecycleCallbacks
Application.OnProvideAssistDataListener
AppOpsManager.OnOpChangedListener
DatePickerDialog.OnDateSetListener
FragmentBreadCrumbs.OnBreadCrumbClickListener
FragmentManager.BackStackEntry
FragmentManager.OnBackStackChangedListener
KeyguardManager.OnKeyguardExitResult
LoaderManager.LoaderCallbacks
PendingIntent.OnFinished
SearchManager.OnCancelListener
SearchManager.OnDismissListener
TimePickerDialog.OnTimeSetListener
UiAutomation.AccessibilityEventFilter
UiAutomation.OnAccessibilityEventListener

Classes

ActionBar
ActionBar.LayoutParams
ActionBar.Tab
Activity
ActivityGroup
ActivityManager

onCreateDialog(int, Bundle)

Android APIs API level: 19

removeDialog(int)

protected void **onRestart** () Added in API level 1

Called after [onStop()](/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()](/reference/android/app/Activity.html#onStart()) and then [onResume()](/reference/android/app/Activity.html#onResume()).

For activities that are using raw [Cursor](/reference/android/database/Cursor.html) objects (instead of creating them through [managedQuery(android.net.Uri, String[], String, String[], String)](/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()](/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()
- onStart()
- onResume()

Interfaces

Classes

protected void **onRestoreInstanceState** (Bundle savedInstanceState) Added in API level 1

This method is called after [onStart()](/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)](/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)](/reference

/android/app/Activity.html#onSaveInstanceState(android.os.Bundle)).

Android APIs API level: 19

/app/Activity.html#onStart()) and onPostCreate(Bundle) (/reference/android/app/Activity.html#onPostCreate(android.os.Bundle)).

Parameters

ed in onSaveInstanceState(Bundle).

See Also

- onCreate(Bundle)
- onPostCreate(Bundle)
- onResume()
- onSaveInstanceState(Bundle)

Added in API level 1

Called after onRestoreInstanceState(Bundle) (/reference/android/app/Activity.html#onRestoreInstanceState(android.os.Bundle)), onRestart() (/reference/android/app/Activity.html#onRestart()), or onPause() (/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) (/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)
- onRestart()
- onPostResume()
- onPause()

protected void **onSaveInstanceState** (Bundle outState)

Added in API level 1

Called to retrieve per-instance state from an activity before being killed so that the state can be restored in onCreate(Bundle)

Interfaces

- ActionBar.OnMenuVisibilityListener
- ActionBar.OnNavigationListener
- ActionBar.TabListener
- Application.ActivityLifecycleCallbacks
- Application.OnProvideAssistDataListener
- AppOpsManager.OnOpChangedListener
- DatePickerDialog.OnDateSetListener
- FragmentBreadCrumbs.OnBreadCrumbClickListener
- FragmentManager.BackStackEntry
- FragmentManager.OnBackStackChangedListener
- KeyguardManager.OnKeyguardExitResult
- LoaderManager.LoaderCallbacks
- PendingIntent.OnFinished
- SearchManager.OnCancelListener
- SearchManager.OnDismissListener
- TimePickerDialog.OnTimeSetListener
- UiAutomation.AccessibilityEventFilter
- UiAutomation.OnAccessibilityEventListener

Classes

- ActionBar
- ActionBar.LayoutParams
- ActionBar.Tab
- Activity
- ActivityGroup
- ActivityManager

(/reference/android/app/Activity.html#onCreate(android.os.Bundle)) or

Android APIs API level: 19

/app/Activity.html#onRestoreInstanceState(android.os.Bundle)) (the Bundle (/reference/android/os/Bundle.html) populated by this method will be passed to both).

This method is called before an activity may be killed so that when it 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) (/reference /android/app/Activity.html#onCreate(android.os.Bundle)) or onRestoreInstanceState(Bundle) (/reference/android /app/Activity.html#onRestoreInstanceState(android.os.Bundle)).

Do not confuse this method with activity lifecycle callbacks such as onPause() (/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() (/reference/android /app/Activity.html#onStop()) which is called before destruction. One example of when onPause() (/reference/android /app/Activity.html#onPause()) and onStop() (/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) (/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() (/reference /android/app/Activity.html#onPause()) is called and not onSaveInstanceState(Bundle) (/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) (/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() (/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 (which is restored by the default implementation of onRestoreInstanceState(Bundle) (/reference/android

Interfaces

ActionBar.OnMenuVisibilityListener
ActionBar.OnNavigationListener
ActionBar.TabListener
Application.ActivityLifecycleCallbacks
Application.OnProvideAssistDataListener
AppOpsManager.OnOpChangedListener
DatePickerDialog.OnDateSetListener
FragmentBreadCrumbs.OnBreadCrumbClickListener
FragmentManager.BackStackEntry
FragmentManager.OnBackStackChangedListener
KeyguardManager.OnKeyguardExitResult
LoaderManager.LoaderCallbacks
PendingIntent.OnFinished
SearchManager.OnCancelListener
SearchManager.OnDismissListener
TimePickerDialog.OnTimeSetListener
UiAutomation.AccessibilityEventFilter
UiAutomation.OnAccessibilityEventListener

Classes

ActionBar
ActionBar.LayoutParams
ActionBar.Tab
Activity
ActivityGroup
ActivityManager

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

Android APIs API level: 19 ot captured by

gh 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() (/reference/android about whether it will occur before or after onPause() (/reference/android/app/Activity.html#onPause()).

Parameters

outState Bundle in which to place your saved state.

See Also

- onCreate(Bundle)
- onRestoreInstanceState(Bundle)
- onPause()

protected void **onStart** () Added in API level 1

Called after onCreate(Bundle) (/reference/android/app/Activity.html#onCreate(android.os.Bundle)) — or after onRestart() (/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() (/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)
- onStop()
- onResume()

protected void **onStop** () Added in API level 1

Called when you are no longer visible to the user. You will next receive either onRestart() (/reference/android/app/Activity.html#onRestart()), onDestroy() (/reference/android/app/Activity.html#onDestroy()), or nothing, depending on later user activity.

Note that this method may never be called, in low memory situations where the system does not have enough memory to keep your

Interfaces

- ActionBar.OnMenuVisibilityListener
- ActionBar.OnNavigationListener
- ActionBar.TabListener
- Application.ActivityLifecycleCallbacks
- Application.OnProvideAssistDataListener
- AppOpsManager.OnOpChangedListener
- DatePickerDialog.OnDateSetListener
- FragmentBreadCrumbs.OnBreadCrumbClickListener
- FragmentManager.BackStackEntry
- FragmentManager.OnBackStackChangedListener
- KeyguardManager.OnKeyguardExitResult
- LoaderManager.LoaderCallbacks
- PendingIntent.OnFinished
- SearchManager.OnCancelListener
- SearchManager.OnDismissListener
- TimePickerDialog.OnTimeSetListener
- UiAutomation.AccessibilityEventFilter
- UiAutomation.OnAccessibilityEventListener

Classes

- ActionBar
- ActionBar.LayoutParams
- ActionBar.Tab
- Activity
- ActivityGroup
- ActivityManager

activity's process running after its onPause() (/reference/android

Android APIs API level: 19

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()
onSaveInstanceState(Bundle)
onDestroy()

protected void **onTitleChanged** (CharSequence title, int color) Added in API level 1

protected void **onUserLeaveHint** () Added in API level 3

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() (/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() (/reference/android/app/Activity.html#onUserLeaveHint()) will not be called on the activity being interrupted. In cases when it is invoked, this method is called right before the activity's onPause() (/reference/android/app/Activity.html#onPause()) callback.

This callback and onUserInteraction() (/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()

Interfaces

ActionBar.OnMenuVisibilityListener
ActionBar.OnNavigationListener
ActionBar.TabListener
Application.ActivityLifecycleCallbacks
Application.OnProvideAssistDataListener
AppOpsManager.OnOpChangedListener
DatePickerDialog.OnDateSetListener
FragmentBreadCrumbs.OnBreadCrumbClickListener
FragmentManager.BackStackEntry
FragmentManager.OnBackStackChangedListener
KeyguardManager.OnKeyguardExitResult
LoaderManager.LoaderCallbacks
PendingIntent.OnFinished
SearchManager.OnCancelListener
SearchManager.OnDismissListener
TimePickerDialog.OnTimeSetListener
UiAutomation.AccessibilityEventFilter
UiAutomation.OnAccessibilityEventListener

Classes

ActionBar
ActionBar.LayoutParams
ActionBar.Tab
Activity
ActivityGroup
ActivityManager