added in API level 1 **(https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)**
Summary: Constants (#constants) | Ctors (#pubctors) |
Methods (#pubmethods) | Inherited Methods (#inhmethods) |
[Expand All] (#)

Context

```
public abstract class Context
extends Object (https://developer.android.com/reference/java/lang/Object.html)
```

java.lang.Object (https://developer.android.com/reference/java/lang/Object.html)
↳ android.content.Context

(#)Known Direct Subclasses

ContextWrapper (https://developer.android.com/reference/android/content/ContextWrapper.html),MockContext (https://developer.android.com/reference/android/test/mock/MockContext.html)

(#)Known Indirect Subclasses

AbstractInputMethodService (https://developer.android.com/reference/android/inputmethodservice/AbstractInputMethodService.html),AccessibilityService (https://developer.android.com/reference/android/accessibilityservice/AccessibilityService.html),AccountAuthenticatorActivity (https://developer.android.com/reference/android/accounts/AccountAuthenticatorActivity.html),Activity (https://developer.android.com/reference/android/app/Activity.html),ActivityGroup (https://developer.android.com/reference/android/app/ActivityGroup.html),AliasActivity (https://developer.android.com/reference/android/app/AliasActivity.html),Application (https://developer.android.com/reference/android/app/Application.html),AutofillService (https://developer.android.com/reference/android/service/autofill/AutofillService.html),BackupAgent (https://developer.android.com/reference/android/app/backup/BackupAgent.html),BackupAgentHelper (https://developer.android.com/reference/android/app/backup/BackupAgentHelper.html),CallScreeningService (https://developer.android.com/reference/android/telecom/CallScreeningService.html),CameraPrewarmService (https://developer.android.com/reference/android/service/media/CameraPrewarmService.html),and 40 others. (#)

Interface to global information about an application environment. This is an abstract class whose implementation is provided by the Android system. It allows access to application-specific resources and classes, as well as up-calls for application-level operations such as launching activities, broadcasting and receiving intents, etc.

Summary

Constants

String (https://developer.android.com/reference/java/lang/String.html)	ACCESSIBILITY_SERVICE (https://developer.android.com/reference/android/content/Context.html#ACCESSIBILITY_SERVICE) Use with getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>)) to retrieve a AccessibilityManager (https://developer.android.com/reference/android/view/accessibility/AccessibilityManager.html) for giving the user feedback for UI events through the registered event listeners.
String (https://developer.android.com/reference/java/lang/String.html)	ACCOUNT_SERVICE (https://developer.android.com/reference/android/content/Context.html#ACCOUNT_SERVICE) Use with getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>)) to retrieve a AccountManager (https://developer.android.com/reference/android/accounts/AccountManager.html) for receiving intents at a time of your choosing.
String (https://developer.android.com/reference/java/lang/String.html)	ACTIVITY_SERVICE (https://developer.android.com/reference/android/content/Context.html#ACTIVITY_SERVICE) Use with getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>)) to retrieve a ActivityManager (https://developer.android.com/reference/android/app/ActivityManager.html) for interacting with the global system state.
String (https://developer.android.com/reference/java/lang/String.html)	ALARM_SERVICE (https://developer.android.com/reference/android/content/Context.html#ALARM_SERVICE) Use with getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>)) to retrieve a AlarmManager (https://developer.android.com/reference/android/app/AlarmManager.html) for receiving intents at a time of your choosing.
String (https://developer.android.com/reference/java/lang/String.html)	APPWIDGET_SERVICE (https://developer.android.com/reference/android/content/Context.html#APPWIDGET_SERVICE) Use with getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>)) to retrieve a AppWidgetManager (https://developer.android.com/reference/android/appwidget/AppWidgetManager.html) for accessing AppWidgets.
String (https://developer.android.com/reference/java/lang/String.html)	APP_OPS_SERVICE (https://developer.android.com/reference/android/content/Context.html#APP_OPS_SERVICE) Use with getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>)) to retrieve a AppOpsManager (https://developer.android.com/reference/android/app/AppOpsManager.html) for tracking application operations on the device.

String (https://developer.android.com/reference/java/lang/String.html)	AUDIO_SERVICE (https://developer.android.com/reference/android/content/Context.html#AUDIO_SERVICE) Use with getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>)) to retrieve a AudioManager (https://developer.android.com/reference/android/media/AudioManager.html) for handling management of volume, ringer modes and audio routing.
String (https://developer.android.com/reference/java/lang/String.html)	BATTERY_SERVICE (https://developer.android.com/reference/android/content/Context.html#BATTERY_SERVICE) Use with getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>)) to retrieve a BatteryManager (https://developer.android.com/reference/android/os/BatteryManager.html) for managing battery state.
int	BIND_ABOVE_CLIENT (https://developer.android.com/reference/android/content/Context.html#BIND_ABOVE_CLIENT) Flag for bindService(Intent, ServiceConnection, int) (https://developer.android.com/reference/android/content/Context.html#bindService(android.content.Intent, android.content.ServiceConnection, int)): indicates that the client application binding to this service considers the service to be more important than the app itself.
int	BIND_ADJUST_WITH_ACTIVITY (https://developer.android.com/reference/android/content/Context.html#BIND_ADJUST_WITH_ACTIVITY) Flag for bindService(Intent, ServiceConnection, int) (https://developer.android.com/reference/android/content/Context.html#bindService(android.content.Intent, android.content.ServiceConnection, int)): If binding from an activity, allow the target service's process importance to be raised based on whether the activity is visible to the user, regardless whether another flag is used to reduce the amount that the client process's overall importance is used to impact it.
int	BIND_ALLOW_OOM_MANAGEMENT (https://developer.android.com/reference/android/content/Context.html#BIND_ALLOW_OOM_MANAGEMENT) Flag for bindService(Intent, ServiceConnection, int) (https://developer.android.com/reference/android/content/Context.html#bindService(android.content.Intent, android.content.ServiceConnection, int)): allow the process hosting the bound service to go through its normal memory management.

int	BIND_AUTO_CREATE (https://developer.android.com/reference/android/content/Context.html#BIND_AUTO_CREATE) Flag for bindService(Intent, ServiceConnection, int) (https://developer.android.com/reference/android/content/Context.html#bindService(android.content.Intent, android.content.ServiceConnection, int)): automatically create the service as long as the binding exists.
int	BIND_DEBUG_UNBIND (https://developer.android.com/reference/android/content/Context.html#BIND_DEBUG_UNBIND) Flag for bindService(Intent, ServiceConnection, int) (https://developer.android.com/reference/android/content/Context.html#bindService(android.content.Intent, android.content.ServiceConnection, int)): include debugging help for mismatched calls to unbind.
int	BIND_EXTERNAL_SERVICE (https://developer.android.com/reference/android/content/Context.html#BIND_EXTERNAL_SERVICE) Flag for bindService(Intent, ServiceConnection, int) (https://developer.android.com/reference/android/content/Context.html#bindService(android.content.Intent, android.content.ServiceConnection, int)): The service being bound is an isolated (https://developer.android.com/reference/android/R.attr.html#isolatedProcess), external (https://developer.android.com/reference/android/R.attr.html#externalService) service.
int	BIND_IMPORTANT (https://developer.android.com/reference/android/content/Context.html#BIND_IMPORTANT) Flag for bindService(Intent, ServiceConnection, int) (https://developer.android.com/reference/android/content/Context.html#bindService(android.content.Intent, android.content.ServiceConnection, int)): this service is very important to the client, so should be brought to the foreground process level when the client is.
int	BIND_NOT_FOREGROUND (https://developer.android.com/reference/android/content/Context.html#BIND_NOT_FOREGROUND) Flag for bindService(Intent, ServiceConnection, int) (https://developer.android.com/reference/android/content/Context.html#bindService(android.content.Intent, android.content.ServiceConnection, int)): don't allow this binding to raise the target service's process to the foreground scheduling priority.

int	BIND_WAIVE_PRIORITY (https://developer.android.com/reference/android/content/Context.html#BIND_WAIVE_PRIORITY) Flag for bindService(Intent, ServiceConnection, int) (https://developer.android.com/reference/android/content/Context.html#bindService(android.content.Intent, android.content.ServiceConnection, int)): don't impact the scheduling or memory management priority of the target service's hosting process.
String (https://developer.android.com/reference/java/lang/String.html)	BLUETOOTH_SERVICE (https://developer.android.com/reference/android/content/Context.html#BLUETOOTH_SERVICE) Use with getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>)) to retrieve a BluetoothManager (https://developer.android.com/reference/android/bluetooth/BluetoothManager.html) for using Bluetooth.
String (https://developer.android.com/reference/java/lang/String.html)	CAMERA_SERVICE (https://developer.android.com/reference/android/content/Context.html#CAMERA_SERVICE) Use with getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>)) to retrieve a CameraManager (https://developer.android.com/reference/android/hardware/camera2/CameraManager.html) for interacting with camera devices.
String (https://developer.android.com/reference/java/lang/String.html)	CAPTIONING_SERVICE (https://developer.android.com/reference/android/content/Context.html#CAPTIONING_SERVICE) Use with getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>)) to retrieve a CaptioningManager (https://developer.android.com/reference/android/view/accessibility/CaptioningManager.html) for obtaining captioning properties and listening for changes in captioning preferences.
String (https://developer.android.com/reference/java/lang/String.html)	CARRIER_CONFIG_SERVICE (https://developer.android.com/reference/android/content/Context.html#CARRIER_CONFIG_SERVICE) Use with getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>)) to retrieve a CarrierConfigManager (https://developer.android.com/reference/android/telephony/CarrierConfigManager.html) for reading carrier configuration values.
String (https://developer.android.com/reference/java/lang/String.html)	CLIPBOARD_SERVICE (https://developer.android.com/reference/android/content/Context.html#CLIPBOARD_SERVICE) Use with getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>)) to retrieve a ClipboardManager (https://developer.android.com/reference/android

	/text/ClipboardManager.html) for accessing and modifying ClipboardManager (https://developer.android.com/reference/android/content/ClipboardManager.html) for accessing and modifying the contents of the global clipboard.
String (https://developer.android.com/reference/java/lang/String.html)	COMPANION_DEVICE_SERVICE (https://developer.android.com/reference/android/content/Context.html#COMPANION_DEVICE_SERVICE) Use with getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>)) to retrieve a CompanionDeviceManager (https://developer.android.com/reference/android/companion/CompanionDeviceManager.html) for managing companion devices
String (https://developer.android.com/reference/java/lang/String.html)	CONNECTIVITY_SERVICE (https://developer.android.com/reference/android/content/Context.html#CONNECTIVITY_SERVICE) Use with getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>)) to retrieve a ConnectivityManager (https://developer.android.com/reference/android/net/ConnectivityManager.html) for handling management of network connections.
String (https://developer.android.com/reference/java/lang/String.html)	CONSUMER_IR_SERVICE (https://developer.android.com/reference/android/content/Context.html#CONSUMER_IR_SERVICE) Use with getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>)) to retrieve a ConsumerIrManager (https://developer.android.com/reference/android/hardware/ConsumerIrManager.html) for transmitting infrared signals from the device.
int	CONTEXT_IGNORE_SECURITY (https://developer.android.com/reference/android/content/Context.html#CONTEXT_IGNORE_SECURITY) Flag for use with createPackageContext(String, int) (https://developer.android.com/reference/android/content/Context.html#createPackageContext(java.lang.String, int)): ignore any security restrictions on the Context being requested, allowing it to always be loaded.
int	CONTEXT_INCLUDE_CODE (https://developer.android.com/reference/android/content/Context.html#CONTEXT_INCLUDE_CODE) Flag for use with createPackageContext(String, int) (https://developer.android.com/reference/android/content/Context.html#createPackageContext(java.lang.String, int)): include the application code with the context.
int	CONTEXT_RESTRICTED (https://developer.android.com/reference/android/content/Context.html#CONTEXT_RESTRICTED) Flag for use with createPackageContext(String, int)

	(https://developer.android.com/reference/android/content/Context.html#createPackageContext(java.lang.String, int)): a restricted context may disable specific features.
String (https://developer.android.com/reference/java/lang/String.html)	DEVICE_POLICY_SERVICE (https://developer.android.com/reference/android/content/Context.html#DEVICE_POLICY_SERVICE) Use with getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>)) to retrieve a DevicePolicyManager (https://developer.android.com/reference/android/app/admin/DevicePolicyManager.html) for working with global device policy management.
String (https://developer.android.com/reference/java/lang/String.html)	DISPLAY_SERVICE (https://developer.android.com/reference/android/content/Context.html#DISPLAY_SERVICE) Use with getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>)) to retrieve a DisplayManager (https://developer.android.com/reference/android/hardware/display/DisplayManager.html) for interacting with display devices.
String (https://developer.android.com/reference/java/lang/String.html)	DOWNLOAD_SERVICE (https://developer.android.com/reference/android/content/Context.html#DOWNLOAD_SERVICE) Use with getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>)) to retrieve a DownloadManager (https://developer.android.com/reference/android/app/DownloadManager.html) for requesting HTTP downloads.
String (https://developer.android.com/reference/java/lang/String.html)	DROPBOX_SERVICE (https://developer.android.com/reference/android/content/Context.html#DROPBOX_SERVICE) Use with getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>)) to retrieve a DropBoxManager (https://developer.android.com/reference/android/os/DropBoxManager.html) instance for recording diagnostic logs.
String (https://developer.android.com/reference/java/lang/String.html)	FINGERPRINT_SERVICE (https://developer.android.com/reference/android/content/Context.html#FINGERPRINT_SERVICE) Use with getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>)) to retrieve a FingerprintManager (https://developer.android.com/reference/android/hardware/fingerprint/FingerprintManager.html) for handling management of fingerprints.
String (https://developer.android.com	HARDWARE_PROPERTIES_SERVICE (https://developer.android.com/reference/android/content/Context.html#HARDWARE_PROPERTIES_SERVICE)

/reference/java/lang /String.html)	Use with getSystemService(Class) (https://developer.android.com/reference /android/content/Context.html#getSystemService(java.lang.Class<T>)) to retrieve a HardwarePropertiesManager (https://developer.android.com/reference /android/os/HardwarePropertiesManager.html) for accessing the hardware properties service.
String (https://developer.android.com /reference/java/lang /String.html)	INPUT_METHOD_SERVICE (https://developer.android.com/reference/android /content/Context.html#INPUT_METHOD_SERVICE) Use with getSystemService(Class) (https://developer.android.com/reference /android/content/Context.html#getSystemService(java.lang.Class<T>)) to retrieve a InputMethodManager (https://developer.android.com/reference/android /view/inputmethod/InputMethodManager.html) for accessing input methods.
String (https://developer.android.com /reference/java/lang /String.html)	INPUT_SERVICE (https://developer.android.com/reference/android/content /Context.html#INPUT_SERVICE) Use with getSystemService(Class) (https://developer.android.com/reference /android/content/Context.html#getSystemService(java.lang.Class<T>)) to retrieve a InputManager (https://developer.android.com/reference/android/hardware/input /InputManager.html) for interacting with input devices.
String (https://developer.android.com /reference/java/lang /String.html)	JOB_SCHEDULER_SERVICE (https://developer.android.com/reference/android /content/Context.html#JOB_SCHEDULER_SERVICE) Use with getSystemService(Class) (https://developer.android.com/reference /android/content/Context.html#getSystemService(java.lang.Class<T>)) to retrieve a JobScheduler (https://developer.android.com/reference/android/app/job /JobScheduler.html) instance for managing occasional background tasks.
String (https://developer.android.com /reference/java/lang /String.html)	KEYGUARD_SERVICE (https://developer.android.com/reference/android/content /Context.html#KEYGUARD_SERVICE) Use with getSystemService(Class) (https://developer.android.com/reference /android/content/Context.html#getSystemService(java.lang.Class<T>)) to retrieve a NotificationManager (https://developer.android.com/reference/android /app/NotificationManager.html) for controlling keyguard.
String (https://developer.android.com /reference/java/lang /String.html)	LAUNCHER_APPS_SERVICE (https://developer.android.com/reference/android /content/Context.html#LAUNCHER_APPS_SERVICE) Use with getSystemService(Class) (https://developer.android.com/reference /android/content/Context.html#getSystemService(java.lang.Class<T>)) to retrieve a LauncherApps (https://developer.android.com/reference/android/content /pm/LauncherApps.html) for querying and monitoring launchable apps across profiles of a user.

String (https://developer.android.com/reference/java/lang/String.html)	LAYOUT_INFLATER_SERVICE (https://developer.android.com/reference/android/content/Context.html#LAYOUT_INFLATER_SERVICE) Use with getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>)) to retrieve a LayoutInflater (https://developer.android.com/reference/android/view/LayoutInflater.html) for inflating layout resources in this context.
String (https://developer.android.com/reference/java/lang/String.html)	LOCATION_SERVICE (https://developer.android.com/reference/android/content/Context.html#LOCATION_SERVICE) Use with getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>)) to retrieve a LocationManager (https://developer.android.com/reference/android/location/LocationManager.html) for controlling location updates.
String (https://developer.android.com/reference/java/lang/String.html)	MEDIA_PROJECTION_SERVICE (https://developer.android.com/reference/android/content/Context.html#MEDIA_PROJECTION_SERVICE) Use with getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>)) to retrieve a MediaProjectionManager (https://developer.android.com/reference/android/media/projection/MediaProjectionManager.html) instance for managing media projection sessions.
String (https://developer.android.com/reference/java/lang/String.html)	MEDIA_ROUTER_SERVICE (https://developer.android.com/reference/android/content/Context.html#MEDIA_ROUTER_SERVICE) Use with getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>)) to retrieve a MediaRouter (https://developer.android.com/reference/android/media/MediaRouter.html) for controlling and managing routing of media.
String (https://developer.android.com/reference/java/lang/String.html)	MEDIA_SESSION_SERVICE (https://developer.android.com/reference/android/content/Context.html#MEDIA_SESSION_SERVICE) Use with getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>)) to retrieve a MediaSessionManager (https://developer.android.com/reference/android/media/session/MediaSessionManager.html) for managing media Sessions.
String (https://developer.android.com/reference/java/lang/String.html)	MIDI_SERVICE (https://developer.android.com/reference/android/content/Context.html#MIDI_SERVICE) Use with getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>)) to retrieve a MidiManager (https://developer.android.com/reference/android/media/midi/MidiManager.html) for accessing the MIDI service.

int	MODE_APPEND (https://developer.android.com/reference/android/content/Context.html#MODE_APPEND) File creation mode: for use with openFileOutput(String, int) (https://developer.android.com/reference/android/content/Context.html#openFileOutput(java.lang.String, int)), if the file already exists then write data to the end of the existing file instead of erasing it.
int	MODE_ENABLE_WRITE_AHEAD_LOGGING (https://developer.android.com/reference/android/content/Context.html#MODE_ENABLE_WRITE_AHEAD_LOGGING) Database open flag: when set, the database is opened with write-ahead logging enabled by default.
int	MODE_MULTI_PROCESS (https://developer.android.com/reference/android/content/Context.html#MODE_MULTI_PROCESS) *This constant was deprecated in API level 23. MODE_MULTI_PROCESS does not work reliably in some versions of Android, and furthermore does not provide any mechanism for reconciling concurrent modifications across processes. Applications should not attempt to use it. Instead, they should use an explicit cross-process data management approach such as* ContentProvider (https://developer.android.com/reference/android/content/ContentProvider.html).
int	MODE_NO_LOCALIZED_COLLATORS (https://developer.android.com/reference/android/content/Context.html#MODE_NO_LOCALIZED_COLLATORS) Database open flag: when set, the database is opened without support for localized collators.
int	MODE_PRIVATE (https://developer.android.com/reference/android/content/Context.html#MODE_PRIVATE) File creation mode: the default mode, where the created file can only be accessed by the calling application (or all applications sharing the same user ID).
int	MODE_WORLD_READABLE (https://developer.android.com/reference/android/content/Context.html#MODE_WORLD_READABLE) *This constant was deprecated in API level 17. Creating world-readable files is very dangerous, and likely to cause security holes in applications. It is strongly discouraged; instead, applications should use more formal mechanism for interactions such as* ContentProvider (https://developer.android.com/reference/android/content/ContentProvider.html), BroadcastReceiver (https://developer.android.com/reference/android/content/BroadcastReceiver.html), *and* Service (https://developer.android.com/reference/android/app/Service.html). *There are no guarantees that this access mode will*

	remain on a file, such as when it goes through a backup and restore.
int	MODE_WORLD_WRITEABLE (https://developer.android.com/reference/android/content/Context.html#MODE_WORLD_WRITEABLE) *This constant was deprecated in API level 17. Creating world-writable files is very dangerous, and likely to cause security holes in applications. It is strongly discouraged; instead, applications should use more formal mechanism for interactions such as* ContentProvider (https://developer.android.com/reference/android/content/ContentProvider.html)*,* BroadcastReceiver (https://developer.android.com/reference/android/content/BroadcastReceiver.html)*, and* Service (https://developer.android.com/reference/android/app/Service.html)*. There are no guarantees that this access mode will remain on a file, such as when it goes through a backup and restore.*
String (https://developer.android.com/reference/java/lang/String.html)	NETWORK_STATS_SERVICE (https://developer.android.com/reference/android/content/Context.html#NETWORK_STATS_SERVICE) Use with getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>)) to retrieve a NetworkStatsManager (https://developer.android.com/reference/android/app/usage/NetworkStatsManager.html) for querying network usage stats.
String (https://developer.android.com/reference/java/lang/String.html)	NFC_SERVICE (https://developer.android.com/reference/android/content/Context.html#NFC_SERVICE) Use with getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>)) to retrieve a NfcManager (https://developer.android.com/reference/android/nfc/NfcManager.html) for using NFC.
String (https://developer.android.com/reference/java/lang/String.html)	NOTIFICATION_SERVICE (https://developer.android.com/reference/android/content/Context.html#NOTIFICATION_SERVICE) Use with getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>)) to retrieve a NotificationManager (https://developer.android.com/reference/android/app/NotificationManager.html) for informing the user of background events.
String (https://developer.android.com/reference/java/lang/String.html)	NSD_SERVICE (https://developer.android.com/reference/android/content/Context.html#NSD_SERVICE) Use with getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>)) to retrieve a NsdManager (https://developer.android.com/reference/android/net/nsd/NsdManager.html) for handling management of network service discovery

String (https://developer.android.com/reference/java/lang/String.html)	POWER_SERVICE (https://developer.android.com/reference/android/content/Context.html#POWER_SERVICE) Use with getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>)) to retrieve a PowerManager (https://developer.android.com/reference/android/os/PowerManager.html) for controlling power management, including "wake locks," which let you keep the device on while you're running long tasks.
String (https://developer.android.com/reference/java/lang/String.html)	PRINT_SERVICE (https://developer.android.com/reference/android/content/Context.html#PRINT_SERVICE) PrintManager (https://developer.android.com/reference/android/print/PrintManager.html) for printing and managing printers and print tasks.
int	RECEIVER_VISIBLE_TO_INSTANT_APPS (https://developer.android.com/reference/android/content/Context.html#RECEIVER_VISIBLE_TO_INSTANT_APPS) Flag for registerReceiver(BroadcastReceiver, IntentFilter) (https://developer.android.com/reference/android/content/Context.html#registerReceiver(android.content.BroadcastReceiver, android.content.IntentFilter)): The receiver can receive broadcasts from Instant Apps.
String (https://developer.android.com/reference/java/lang/String.html)	RESTRICTIONS_SERVICE (https://developer.android.com/reference/android/content/Context.html#RESTRICTIONS_SERVICE) Use with getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>)) to retrieve a RestrictionsManager (https://developer.android.com/reference/android/content/RestrictionsManager.html) for retrieving application restrictions and requesting permissions for restricted operations.
String (https://developer.android.com/reference/java/lang/String.html)	SEARCH_SERVICE (https://developer.android.com/reference/android/content/Context.html#SEARCH_SERVICE) Use with getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>)) to retrieve a SearchManager (https://developer.android.com/reference/android/app/SearchManager.html) for handling searches.
String (https://developer.android.com/reference/java/lang/String.html)	SENSOR_SERVICE (https://developer.android.com/reference/android/content/Context.html#SENSOR_SERVICE) Use with getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>)) to retrieve a SensorManager (https://developer.android.com/reference/android/hardware/SensorManager.html) for accessing sensors.

String (https://developer.android.com/reference/java/lang/String.html)	SHORTCUT_SERVICE (https://developer.android.com/reference/android/content/Context.html#SHORTCUT_SERVICE) Use with getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>)) to retrieve a ShortcutManager (https://developer.android.com/reference/android/content/pm/ShortcutManager.html) for accessing the launcher shortcut service.
String (https://developer.android.com/reference/java/lang/String.html)	STORAGE_SERVICE (https://developer.android.com/reference/android/content/Context.html#STORAGE_SERVICE) Use with getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>)) to retrieve a StorageManager (https://developer.android.com/reference/android/os/storage/StorageManager.html) for accessing system storage functions.
String (https://developer.android.com/reference/java/lang/String.html)	STORAGE_STATS_SERVICE (https://developer.android.com/reference/android/content/Context.html#STORAGE_STATS_SERVICE) Use with getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>)) to retrieve a StorageStatsManager (https://developer.android.com/reference/android/app/usage/StorageStatsManager.html) for accessing system storage statistics.
String (https://developer.android.com/reference/java/lang/String.html)	SYSTEM_HEALTH_SERVICE (https://developer.android.com/reference/android/content/Context.html#SYSTEM_HEALTH_SERVICE) Use with getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>)) to retrieve a SystemHealthManager (https://developer.android.com/reference/android/os/health/SystemHealthManager.html) for accessing system health (battery, power, memory, etc) metrics.
String (https://developer.android.com/reference/java/lang/String.html)	TELECOM_SERVICE (https://developer.android.com/reference/android/content/Context.html#TELECOM_SERVICE) Use with getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>)) to retrieve a TelecomManager (https://developer.android.com/reference/android/telecom/TelecomManager.html) to manage telecom-related features of the device.
String (https://developer.android.com/reference/java/lang/String.html)	TELEPHONY_SERVICE (https://developer.android.com/reference/android/content/Context.html#TELEPHONY_SERVICE) Use with getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>)) to retrieve a TelephonyManager (https://developer.android.com/reference/android/telephony/TelephonyManager.html) for handling management the telephony features of the

	device.
String (https://developer.android.com/reference/java/lang/String.html)	TELEPHONY_SUBSCRIPTION_SERVICE (https://developer.android.com/reference/android/content/Context.html#TELEPHONY_SUBSCRIPTION_SERVICE) Use with getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>)) to retrieve a SubscriptionManager (https://developer.android.com/reference/android/telephony/SubscriptionManager.html) for handling management the telephony subscriptions of the device.
String (https://developer.android.com/reference/java/lang/String.html)	TEXT_CLASSIFICATION_SERVICE (https://developer.android.com/reference/android/content/Context.html#TEXT_CLASSIFICATION_SERVICE) Use with getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>)) to retrieve a TextClassificationManager (https://developer.android.com/reference/android/view/textclassifier/TextClassificationManager.html) for text classification services.
String (https://developer.android.com/reference/java/lang/String.html)	TEXT_SERVICES_MANAGER_SERVICE (https://developer.android.com/reference/android/content/Context.html#TEXT_SERVICES_MANAGER_SERVICE) Use with getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>)) to retrieve a TextServicesManager (https://developer.android.com/reference/android/view/textservice/TextServicesManager.html) for accessing text services.
String (https://developer.android.com/reference/java/lang/String.html)	TV_INPUT_SERVICE (https://developer.android.com/reference/android/content/Context.html#TV_INPUT_SERVICE) Use with getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>)) to retrieve a TvInputManager (https://developer.android.com/reference/android/media/tv/TvInputManager.html) for interacting with TV inputs on the device.
String (https://developer.android.com/reference/java/lang/String.html)	UI_MODE_SERVICE (https://developer.android.com/reference/android/content/Context.html#UI_MODE_SERVICE) Use with getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>)) to retrieve a UiModeManager (https://developer.android.com/reference/android/app/UiModeManager.html) for controlling UI modes.
String (https://developer.android.com/reference/java/lang/String.html)	USAGE_STATS_SERVICE (https://developer.android.com/reference/android/content/Context.html#USAGE_STATS_SERVICE) Use with getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>)) to retrieve a

	UsageStatsManager (https://developer.android.com/reference/android/app/usage/UsageStatsManager.html) for querying device usage stats.
String (https://developer.android.com/reference/java/lang/String.html)	USB_SERVICE (https://developer.android.com/reference/android/content/Context.html#USB_SERVICE) Use with getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>)) to retrieve a UsbManager (https://developer.android.com/reference/android/hardware/usb/UsbManager.html) for access to USB devices (as a USB host) and for controlling this device's behavior as a USB device.
String (https://developer.android.com/reference/java/lang/String.html)	USER_SERVICE (https://developer.android.com/reference/android/content/Context.html#USER_SERVICE) Use with getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>)) to retrieve a UserManager (https://developer.android.com/reference/android/os/UserManager.html) for managing users on devices that support multiple users.
String (https://developer.android.com/reference/java/lang/String.html)	VIBRATOR_SERVICE (https://developer.android.com/reference/android/content/Context.html#VIBRATOR_SERVICE) Use with getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>)) to retrieve a Vibrator (https://developer.android.com/reference/android/os/Vibrator.html) for interacting with the vibration hardware.
String (https://developer.android.com/reference/java/lang/String.html)	WALLPAPER_SERVICE (https://developer.android.com/reference/android/content/Context.html#WALLPAPER_SERVICE) Use with getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>)) to retrieve a com.android.server.WallpaperService for accessing wallpapers.
String (https://developer.android.com/reference/java/lang/String.html)	WIFI_AWARE_SERVICE (https://developer.android.com/reference/android/content/Context.html#WIFI_AWARE_SERVICE) Use with getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>)) to retrieve a WifiAwareManager (https://developer.android.com/reference/android/net/wifi/aware/WifiAwareManager.html) for handling management of Wi-Fi Aware.
String (https://developer.android.com/reference/java/lang/String.html)	WIFI_P2P_SERVICE (https://developer.android.com/reference/android/content/Context.html#WIFI_P2P_SERVICE) Use with getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>)) to retrieve a

	WifiP2pManager (https://developer.android.com/reference/android/net/wifi/p2p/WifiP2pManager.html) for handling management of Wi-Fi peer-to-peer connections.
String (https://developer.android.com/reference/java/lang/String.html)	WIFI_SERVICE (https://developer.android.com/reference/android/content/Context.html#WIFI_SERVICE) Use with getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>)) to retrieve a WifiManager (https://developer.android.com/reference/android/net/wifi/WifiManager.html) for handling management of Wi-Fi access.
String (https://developer.android.com/reference/java/lang/String.html)	WINDOW_SERVICE (https://developer.android.com/reference/android/content/Context.html#WINDOW_SERVICE) Use with getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>)) to retrieve a WindowManager (https://developer.android.com/reference/android/view/WindowManager.html) for accessing the system's window manager.

Public constructors
Context (https://developer.android.com/reference/android/content/Context.html#Context())()

Public methods	
abstract boolean	bindService (https://developer.android.com/reference/android/content/Context.html#bindService(android.content.Intent, android.content.ServiceConnection, int))(Intent (https://developer.android.com/reference/android/content/Intent.html) service, ServiceConnection (https://developer.android.com/reference/android/content/ServiceConnection.html) conn, int flags) Connect to an application service, creating it if needed.
abstract int	checkCallingOrSelfPermission (https://developer.android.com/reference/android/content/Context.html#checkCallingOrSelfPermission(java.lang.String))(String (https://developer.android.com/reference/java/lang/String.html) permission) Determine whether the calling process of an IPC *or you* have been granted a particular permission.
abstract int	checkCallingOrSelfUriPermission (https://developer.android.com/reference/android/content/Context.html#checkCallingOrSelfUriPermission(android.net.Uri, int))(Uri (https://developer.android.com/reference/android/net/Uri.html) uri, int

	modeFlags) Determine whether the calling process of an IPC *or you* has been granted permission to access a specific URI.
abstract int	checkCallingPermission (https://developer.android.com/reference/android/content/Context.html#checkCallingPermission(java.lang.String))(String (https://developer.android.com/reference/java/lang/String.html) permission) Determine whether the calling process of an IPC you are handling has been granted a particular permission.
abstract int	checkCallingUriPermission (https://developer.android.com/reference/android/content/Context.html#checkCallingUriPermission(android.net.Uri, int))(Uri (https://developer.android.com/reference/android/net/Uri.html) uri, int modeFlags) Determine whether the calling process and user ID has been granted permission to access a specific URI.
abstract int	checkPermission (https://developer.android.com/reference/android/content/Context.html#checkPermission(java.lang.String, int, int))(String (https://developer.android.com/reference/java/lang/String.html) permission, int pid, int uid) Determine whether the given permission is allowed for a particular process and user ID running in the system.
abstract int	checkSelfPermission (https://developer.android.com/reference/android/content/Context.html#checkSelfPermission(java.lang.String))(String (https://developer.android.com/reference/java/lang/String.html) permission) Determine whether *you* have been granted a particular permission.
abstract int	checkUriPermission (https://developer.android.com/reference/android/content/Context.html#checkUriPermission(android.net.Uri, java.lang.String, java.lang.String, int, int, int))(Uri (https://developer.android.com/reference/android/net/Uri.html) uri, String (https://developer.android.com/reference/java/lang/String.html) readPermission, String (https://developer.android.com/reference/java/lang/String.html) writePermission, int pid, int uid, int modeFlags) Check both a Uri and normal permission.
abstract int	checkUriPermission (https://developer.android.com/reference/android/content/Context.html#checkUriPermission(android.net.Uri, int, int, int))(Uri (https://developer.android.com/reference/android/net/Uri.html) uri, int pid, int uid, int modeFlags) Determine whether a particular process and user ID has been granted

	permission to access a specific URI.
abstract void	clearWallpaper (https://developer.android.com/reference/android/content/Context.html#clearWallpaper())() *This method was deprecated in API level 5. Use* WallpaperManager.clear() (https://developer.android.com/reference/android/app/WallpaperManager.html#clear()) *instead.* *This method requires the caller to hold the permission* SET_WALLPAPER (https://developer.android.com/reference/android/Manifest.permission.html#SET_WALLPAPER).
abstract Context (https://developer.android.com/reference/android/content/Context.html)	createConfigurationContext (https://developer.android.com/reference/android/content/Context.html#createConfigurationContext(android.content.res.Configuration))(Configuration (https://developer.android.com/reference/android/content/res/Configuration.html) overrideConfiguration) Return a new Context object for the current Context but whose resources are adjusted to match the given Configuration.
abstract Context (https://developer.android.com/reference/android/content/Context.html)	createContextForSplit (https://developer.android.com/reference/android/content/Context.html#createContextForSplit(java.lang.String))(String (https://developer.android.com/reference/java/lang/String.html) splitName) Return a new Context object for the given split name.
abstract Context (https://developer.android.com/reference/android/content/Context.html)	createDeviceProtectedStorageContext (https://developer.android.com/reference/android/content/Context.html#createDeviceProtectedStorageContext())() Return a new Context object for the current Context but whose storage APIs are backed by device-protected storage.
abstract Context (https://developer.android.com/reference/android/content/Context.html)	createDisplayContext (https://developer.android.com/reference/android/content/Context.html#createDisplayContext(android.view.Display))(Display (https://developer.android.com/reference/android/view/Display.html) display) Return a new Context object for the current Context but whose resources are adjusted to match the metrics of the given Display.
abstract Context (https://developer.android.com/reference/android/content/Context.html)	createPackageContext (https://developer.android.com/reference/android/content/Context.html#createPackageContext(java.lang.String, int))(String (https://developer.android.com/reference/java/lang/String.html) packageName, int flags) Return a new Context object for the given application name.

abstract String[] (https://developer.android.com/reference/java/lang/String.html)	databaseList (https://developer.android.com/reference/android/content/Context.html#databaseList())() Returns an array of strings naming the private databases associated with this Context's application package.
abstract boolean	deleteDatabase (https://developer.android.com/reference/android/content/Context.html#deleteDatabase(java.lang.String))(String (https://developer.android.com/reference/java/lang/String.html) name) Delete an existing private SQLiteDatabase associated with this Context's application package.
abstract boolean	deleteFile (https://developer.android.com/reference/android/content/Context.html#deleteFile(java.lang.String))(String (https://developer.android.com/reference/java/lang/String.html) name) Delete the given private file associated with this Context's application package.
abstract boolean	deleteSharedPreferences (https://developer.android.com/reference/android/content/Context.html#deleteSharedPreferences(java.lang.String))(String (https://developer.android.com/reference/java/lang/String.html) name) Delete an existing shared preferences file.
abstract void	enforceCallingOrSelfPermission (https://developer.android.com/reference/android/content/Context.html#enforceCallingOrSelfPermission(java.lang.String, java.lang.String))(String (https://developer.android.com/reference/java/lang/String.html) permission, String (https://developer.android.com/reference/java/lang/String.html) message) If neither you nor the calling process of an IPC you are handling has been granted a particular permission, throw a SecurityException (https://developer.android.com/reference/java/lang/SecurityException.html).
abstract void	enforceCallingOrSelfUriPermission (https://developer.android.com/reference/android/content/Context.html#enforceCallingOrSelfUriPermission(android.net.Uri, int, java.lang.String))(Uri (https://developer.android.com/reference/android/net/Uri.html) uri, int modeFlags, String (https://developer.android.com/reference/java/lang/String.html) message) If the calling process of an IPC *or you* has not been granted permission to access a specific URI, throw SecurityException (https://developer.android.com/reference/java/lang/SecurityException.html).
abstract void	enforceCallingPermission (https://developer.android.com/reference/android/content/Context.html#enforceCallingPermission(java.lang.String, java.lang.String))(String (https://developer.android.com/reference/java/lang

	/String.html) permission, String (https://developer.android.com/reference /java/lang/String.html) message) If the calling process of an IPC you are handling has not been granted a particular permission, throw a SecurityException (https://developer.android.com/reference/java/lang/SecurityException.html).
abstract void	enforceCallingUriPermission (https://developer.android.com/reference /android/content/Context.html#enforceCallingUriPermission(android.net.Uri, int, java.lang.String))(Uri (https://developer.android.com/reference/android /net/Uri.html) uri, int modeFlags, String (https://developer.android.com /reference/java/lang/String.html) message) If the calling process and user ID has not been granted permission to access a specific URI, throw SecurityException (https://developer.android.com /reference/java/lang/SecurityException.html).
abstract void	enforcePermission (https://developer.android.com/reference/android/content /Context.html#enforcePermission(java.lang.String, int, int, java.lang.String)) (String (https://developer.android.com/reference/java/lang/String.html) permission, int pid, int uid, String (https://developer.android.com /reference/java/lang/String.html) message) If the given permission is not allowed for a particular process and user ID running in the system, throw a SecurityException (https://developer.android.com/reference/java/lang/SecurityException.html).
abstract void	enforceUriPermission (https://developer.android.com/reference/android /content/Context.html#enforceUriPermission(android.net.Uri, java.lang.String, java.lang.String, int, int, int, java.lang.String))(Uri (https://developer.android.com/reference/android/net/Uri.html) uri, String (https://developer.android.com/reference/java/lang/String.html) readPermission, String (https://developer.android.com/reference/java/lang /String.html) writePermission, int pid, int uid, int modeFlags, String (https://developer.android.com/reference/java/lang/String.html) message) Enforce both a Uri and normal permission.
abstract void	enforceUriPermission (https://developer.android.com/reference/android /content/Context.html#enforceUriPermission(android.net.Uri, int, int, int, java.lang.String))(Uri (https://developer.android.com/reference/android /net/Uri.html) uri, int pid, int uid, int modeFlags, String (https://developer.android.com/reference/java/lang/String.html) message) If a particular process and user ID has not been granted permission to access a specific URI, throw SecurityException (https://developer.android.com

	/reference/java/lang/SecurityException.html).
abstract String[] (https://developer.android.com/reference/java/lang/String.html)	fileList (https://developer.android.com/reference/android/content/Context.html#fileList())() Returns an array of strings naming the private files associated with this Context's application package.
abstract Context (https://developer.android.com/reference/android/content/Context.html)	getApplicationContext (https://developer.android.com/reference/android/content/Context.html#getApplicationContext())() Return the context of the single, global Application object of the current process.
abstract ApplicationInfo (https://developer.android.com/reference/android/content/pm/ApplicationInfo.html)	getApplicationInfo (https://developer.android.com/reference/android/content/Context.html#getApplicationInfo())() Return the full application info for this context's package.
abstract AssetManager (https://developer.android.com/reference/android/content/res/AssetManager.html)	getAssets (https://developer.android.com/reference/android/content/Context.html#getAssets())() Returns an AssetManager instance for the application's package.
abstract File (https://developer.android.com/reference/java/io/File.html)	getCacheDir (https://developer.android.com/reference/android/content/Context.html#getCacheDir())() Returns the absolute path to the application specific cache directory on the filesystem.
abstract ClassLoader (https://developer.android.com/reference/java/lang/ClassLoader.html)	getClassLoader (https://developer.android.com/reference/android/content/Context.html#getClassLoader())() Return a class loader you can use to retrieve classes in this package.
abstract File (https://developer.android.com/reference/java/io/File.html)	getCodeCacheDir (https://developer.android.com/reference/android/content/Context.html#getCodeCacheDir())() Returns the absolute path to the application specific cache directory on the filesystem designed for storing cached code.
final int	getColor (https://developer.android.com/reference/android/content/Context.html#getColor(int))(int id) Returns a color associated with a particular resource ID and styled for the current theme.
final ColorStateList (https://developer.android.com	getColorStateList (https://developer.android.com/reference/android/content/Context.html#getColorStateList(int))(int id)

/reference/android/content /res/ColorStateList.html)	Returns a color state list associated with a particular resource ID and styled for the current theme.
abstract ContentResolver (https://developer.android.com /reference/android/content /ContentResolver.html)	getContentResolver (https://developer.android.com/reference/android/content /Context.html#getContentResolver())() Return a ContentResolver instance for your application's package.
abstract File (https://developer.android.com /reference/java/io/File.html)	getDataDir (https://developer.android.com/reference/android/content /Context.html#getDataDir())() Returns the absolute path to the directory on the filesystem where all private files belonging to this app are stored.
abstract File (https://developer.android.com /reference/java/io/File.html)	getDatabasePath (https://developer.android.com/reference/android/content /Context.html#getDatabasePath(java.lang.String))(String (https://developer.android.com/reference/java/lang/String.html) name) Returns the absolute path on the filesystem where a database created with openOrCreateDatabase(String, int, SQLiteDatabase.CursorFactory) (https://developer.android.com/reference/android/content /Context.html#openOrCreateDatabase(java.lang.String, int, android.database.sqlite.SQLiteDatabase.CursorFactory)) is stored.
abstract File (https://developer.android.com /reference/java/io/File.html)	getDir (https://developer.android.com/reference/android/content /Context.html#getDir(java.lang.String, int))(String (https://developer.android.com/reference/java/lang/String.html) name, int mode) Retrieve, creating if needed, a new directory in which the application can place its own custom data files.
final Drawable (https://developer.android.com /reference/android/graphics /drawable/Drawable.html)	getDrawable (https://developer.android.com/reference/android/content /Context.html#getDrawable(int))(int id) Returns a drawable object associated with a particular resource ID and styled for the current theme.
abstract File (https://developer.android.com /reference/java/io/File.html)	getExternalCacheDir (https://developer.android.com/reference/android /content/Context.html#getExternalCacheDir())() Returns absolute path to application-specific directory on the primary shared/external storage device where the application can place cache files it owns.
abstract File[] (https://developer.android.com /reference/java/io/File.html)	getExternalCacheDirs (https://developer.android.com/reference/android /content/Context.html#getExternalCacheDirs())() Returns absolute paths to application-specific directories on all shared/external

	storage devices where the application can place cache files it owns.
abstract File (https://developer.android.com/reference/java/io/File.html)	getExternalFilesDir (https://developer.android.com/reference/android/content/Context.html#getExternalFilesDir(java.lang.String))(String (https://developer.android.com/reference/java/lang/String.html) type) Returns the absolute path to the directory on the primary shared/external storage device where the application can place persistent files it owns.
abstract File[] (https://developer.android.com/reference/java/io/File.html)	getExternalFilesDirs (https://developer.android.com/reference/android/content/Context.html#getExternalFilesDirs(java.lang.String))(String (https://developer.android.com/reference/java/lang/String.html) type) Returns absolute paths to application-specific directories on all shared/external storage devices where the application can place persistent files it owns.
abstract File[] (https://developer.android.com/reference/java/io/File.html)	getExternalMediaDirs (https://developer.android.com/reference/android/content/Context.html#getExternalMediaDirs())() Returns absolute paths to application-specific directories on all shared/external storage devices where the application can place media files.
abstract File (https://developer.android.com/reference/java/io/File.html)	getFileStreamPath (https://developer.android.com/reference/android/content/Context.html#getFileStreamPath(java.lang.String))(String (https://developer.android.com/reference/java/lang/String.html) name) Returns the absolute path on the filesystem where a file created with openFileOutput(String, int) (https://developer.android.com/reference/android/content/Context.html#openFileOutput(java.lang.String, int)) is stored.
abstract File (https://developer.android.com/reference/java/io/File.html)	getFilesDir (https://developer.android.com/reference/android/content/Context.html#getFilesDir())() Returns the absolute path to the directory on the filesystem where files created with openFileOutput(String, int) (https://developer.android.com/reference/android/content/Context.html#openFileOutput(java.lang.String, int)) are stored.
abstract Looper (https://developer.android.com/reference/android/os/Looper.html)	getMainLooper (https://developer.android.com/reference/android/content/Context.html#getMainLooper())() Return the Looper for the main thread of the current process.
abstract File (https://developer.android.com/reference/java/io/File.html)	getNoBackupFilesDir (https://developer.android.com/reference/android/content/Context.html#getNoBackupFilesDir())() Returns the absolute path to the directory on the filesystem similar to getFilesDir() (https://developer.android.com/reference/android/content/Context.html#getFilesDir()).

abstract File (https://developer.android.com/reference/java/io/File.html)	getObbDir (https://developer.android.com/reference/android/content/Context.html#getObbDir())() Return the primary shared/external storage directory where this application's OBB files (if there are any) can be found.
abstract File[] (https://developer.android.com/reference/java/io/File.html)	getObbDirs (https://developer.android.com/reference/android/content/Context.html#getObbDirs())() Returns absolute paths to application-specific directories on all shared/external storage devices where the application's OBB files (if there are any) can be found.
abstract String (https://developer.android.com/reference/java/lang/String.html)	getPackageCodePath (https://developer.android.com/reference/android/content/Context.html#getPackageCodePath())() Return the full path to this context's primary Android package.
abstract PackageManager (https://developer.android.com/reference/android/content/pm/PackageManager.html)	getPackageManager (https://developer.android.com/reference/android/content/Context.html#getPackageManager())() Return PackageManager instance to find global package information.
abstract String (https://developer.android.com/reference/java/lang/String.html)	getPackageName (https://developer.android.com/reference/android/content/Context.html#getPackageName())() Return the name of this application's package.
abstract String (https://developer.android.com/reference/java/lang/String.html)	getPackageResourcePath (https://developer.android.com/reference/android/content/Context.html#getPackageResourcePath())() Return the full path to this context's primary Android package.
abstract Resources (https://developer.android.com/reference/android/content/res/Resources.html)	getResources (https://developer.android.com/reference/android/content/Context.html#getResources())() Returns a Resources instance for the application's package.
abstract SharedPreferences (https://developer.android.com/reference/android/content/SharedPreferences.html)	getSharedPreferences (https://developer.android.com/reference/android/content/Context.html#getSharedPreferences(java.lang.String, int))(String (https://developer.android.com/reference/java/lang/String.html) name, int mode) Retrieve and hold the contents of the preferences file 'name', returning a SharedPreferences through which you can retrieve and modify its values.
final String (https://developer.android.com	getString (https://developer.android.com/reference/android/content/Context.html#getString(int, java.lang.Object...))(int resId, Object...

/reference/java/lang /String.html)	(https://developer.android.com/reference/java/lang/Object.html) formatArgs) Returns a localized formatted string from the application's package's default string table, substituting the format arguments as defined in Formatter (https://developer.android.com/reference/java/util/Formatter.html) and format(String, Object...) (https://developer.android.com/reference /java/lang/String.html#format(java.lang.String, java.lang.Object...)).
final String (https://developer.android.com /reference/java/lang /String.html)	getString (https://developer.android.com/reference/android/content /Context.html#getString(int))(int resId) Returns a localized string from the application's package's default string table.
final <T> T	getSystemService (https://developer.android.com/reference/android/content /Context.html#getSystemService(java.lang.Class<T>))(Class (https://developer.android.com/reference/java/lang/Class.html)<T> serviceClass) Return the handle to a system-level service by class.
abstract Object (https://developer.android.com /reference/java/lang /Object.html)	getSystemService (https://developer.android.com/reference/android/content /Context.html#getSystemService(java.lang.String))(String (https://developer.android.com/reference/java/lang/String.html) name) Return the handle to a system-level service by name.
abstract String (https://developer.android.com /reference/java/lang /String.html)	getSystemServiceName (https://developer.android.com/reference/android /content/Context.html#getSystemServiceName(java.lang.Class<?>))(Class (https://developer.android.com/reference/java/lang/Class.html)<?> serviceClass) Gets the name of the system-level service that is represented by the specified class.
final CharSequence (https://developer.android.com /reference/java/lang /CharSequence.html)	getText (https://developer.android.com/reference/android/content /Context.html#getText(int))(int resId) Return a localized, styled CharSequence from the application's package's default string table.
abstract Resources.Theme (https://developer.android.com /reference/android/content /res/Resources.Theme.html)	getTheme (https://developer.android.com/reference/android/content /Context.html#getTheme())() Return the Theme object associated with this Context.
abstract Drawable (https://developer.android.com /reference/android/graphics	getWallpaper (https://developer.android.com/reference/android/content /Context.html#getWallpaper())() *This method was deprecated in API level 5. Use* WallpaperManager.get()

/drawable/Drawable.html)	(https://developer.android.com/reference/android /app/WallpaperManager.html#getDrawable()) *instead.*
abstract int	getWallpaperDesiredMinimumHeight (https://developer.android.com /reference/android/content/Context.html#getWallpaperDesiredMinimumHeight())() *This method was deprecated in API level 5. Use* WallpaperManager.getDesiredMinimumHeight() (https://developer.android.com/reference/android /app/WallpaperManager.html#getDesiredMinimumHeight()) *instead.*
abstract int	getWallpaperDesiredMinimumWidth (https://developer.android.com /reference/android/content/Context.html#getWallpaperDesiredMinimumWidth())() *This method was deprecated in API level 5. Use* WallpaperManager.getDesiredMinimumWidth() (https://developer.android.com/reference/android /app/WallpaperManager.html#getDesiredMinimumWidth()) *instead.*
abstract void	grantUriPermission (https://developer.android.com/reference/android/content /Context.html#grantUriPermission(java.lang.String, android.net.Uri, int))(String (https://developer.android.com/reference/java/lang/String.html) toPackage, Uri (https://developer.android.com/reference/android /net/Uri.html) uri, int modeFlags) Grant permission to access a specific Uri to another package, regardless of whether that package has general permission to access the Uri's content provider.
abstract boolean	isDeviceProtectedStorage (https://developer.android.com/reference/android /content/Context.html#isDeviceProtectedStorage())() Indicates if the storage APIs of this Context are backed by device-protected storage.
boolean	isRestricted (https://developer.android.com/reference/android/content /Context.html#isRestricted())() Indicates whether this Context is restricted.
abstract boolean	moveDatabaseFrom (https://developer.android.com/reference/android/content /Context.html#moveDatabaseFrom(android.content.Context, java.lang.String)) (Context (https://developer.android.com/reference/android/content /Context.html) sourceContext, String (https://developer.android.com /reference/java/lang/String.html) name) Move an existing database file from the given source storage context to this context.

abstract boolean	moveSharedPreferencesFrom (https://developer.android.com/reference/android/content/Context.html#moveSharedPreferencesFrom(android.content.Context, java.lang.String))(Context (https://developer.android.com/reference/android/content/Context.html) sourceContext, String (https://developer.android.com/reference/java/lang/String.html) name) Move an existing shared preferences file from the given source storage context to this context.
final TypedArray (https://developer.android.com/reference/android/content/res/TypedArray.html)	obtainStyledAttributes (https://developer.android.com/reference/android/content/Context.html#obtainStyledAttributes(android.util.AttributeSet, int[]))(AttributeSet (https://developer.android.com/reference/android/util/AttributeSet.html) set, int[] attrs) Retrieve styled attribute information in this Context's theme.
final TypedArray (https://developer.android.com/reference/android/content/res/TypedArray.html)	obtainStyledAttributes (https://developer.android.com/reference/android/content/Context.html#obtainStyledAttributes(android.util.AttributeSet, int[], int, int))(AttributeSet (https://developer.android.com/reference/android/util/AttributeSet.html) set, int[] attrs, int defStyleAttr, int defStyleRes) Retrieve styled attribute information in this Context's theme.
final TypedArray (https://developer.android.com/reference/android/content/res/TypedArray.html)	obtainStyledAttributes (https://developer.android.com/reference/android/content/Context.html#obtainStyledAttributes(int, int[]))(int resid, int[] attrs) Retrieve styled attribute information in this Context's theme.
final TypedArray (https://developer.android.com/reference/android/content/res/TypedArray.html)	obtainStyledAttributes (https://developer.android.com/reference/android/content/Context.html#obtainStyledAttributes(int[]))(int[] attrs) Retrieve styled attribute information in this Context's theme.
abstract FileInputStream (https://developer.android.com/reference/java/io/FileInputStream.html)	openFileInput (https://developer.android.com/reference/android/content/Context.html#openFileInput(java.lang.String))(String (https://developer.android.com/reference/java/lang/String.html) name) Open a private file associated with this Context's application package for reading.
abstract FileOutputStream (https://developer.android.com/reference/java/io/FileOutputStream.html)	openFileOutput (https://developer.android.com/reference/android/content/Context.html#openFileOutput(java.lang.String, int))(String (https://developer.android.com/reference/java/lang/String.html) name, int mode) Open a private file associated with this Context's application package for

	writing.
abstract SQLiteDatabase (https://developer.android.com/reference/android/database/sqlite/SQLiteDatabase.html)	openOrCreateDatabase (https://developer.android.com/reference/android/content/Context.html#openOrCreateDatabase(java.lang.String, int, android.database.sqlite.SQLiteDatabase.CursorFactory, android.database.DatabaseErrorHandler))(String (https://developer.android.com/reference/java/lang/String.html) name, int mode, SQLiteDatabase.CursorFactory (https://developer.android.com/reference/android/database/sqlite/SQLiteDatabase.CursorFactory.html) factory, DatabaseErrorHandler (https://developer.android.com/reference/android/database/DatabaseErrorHandler.html) errorHandler) Open a new private SQLiteDatabase associated with this Context's application package.
abstract SQLiteDatabase (https://developer.android.com/reference/android/database/sqlite/SQLiteDatabase.html)	openOrCreateDatabase (https://developer.android.com/reference/android/content/Context.html#openOrCreateDatabase(java.lang.String, int, android.database.sqlite.SQLiteDatabase.CursorFactory))(String (https://developer.android.com/reference/java/lang/String.html) name, int mode, SQLiteDatabase.CursorFactory (https://developer.android.com/reference/android/database/sqlite/SQLiteDatabase.CursorFactory.html) factory) Open a new private SQLiteDatabase associated with this Context's application package.
abstract Drawable (https://developer.android.com/reference/android/graphics/drawable/Drawable.html)	peekWallpaper (https://developer.android.com/reference/android/content/Context.html#peekWallpaper())() *This method was deprecated in API level 5. Use* WallpaperManager.peek() (https://developer.android.com/reference/android/app/WallpaperManager.html#peekDrawable()) *instead.*
void	registerComponentCallbacks (https://developer.android.com/reference/android/content/Context.html#registerComponentCallbacks(android.content.ComponentCallbacks))(ComponentCallbacks (https://developer.android.com/reference/android/content/ComponentCallbacks.html) callback) Add a new ComponentCallbacks (https://developer.android.com/reference/android/content/ComponentCallbacks.html) to the base application of the Context, which will be called at the same times as the ComponentCallbacks methods of activities and other components are called.
abstract Intent (https://developer.android.com	registerReceiver (https://developer.android.com/reference/android/content/Context.html#registerReceiver(android.content.BroadcastReceiver,

/reference/android/content/Intent.html)	android.content.IntentFilter))(BroadcastReceiver (https://developer.android.com/reference/android/content/BroadcastReceiver.html) receiver, IntentFilter (https://developer.android.com/reference/android/content/IntentFilter.html) filter) Register a BroadcastReceiver to be run in the main activity thread.
abstract Intent (https://developer.android.com/reference/android/content/Intent.html)	registerReceiver (https://developer.android.com/reference/android/content/Context.html#registerReceiver(android.content.BroadcastReceiver, android.content.IntentFilter, int))(BroadcastReceiver (https://developer.android.com/reference/android/content/BroadcastReceiver.html) receiver, IntentFilter (https://developer.android.com/reference/android/content/IntentFilter.html) filter, int flags) Register to receive intent broadcasts, with the receiver optionally being exposed to Instant Apps.
abstract Intent (https://developer.android.com/reference/android/content/Intent.html)	registerReceiver (https://developer.android.com/reference/android/content/Context.html#registerReceiver(android.content.BroadcastReceiver, android.content.IntentFilter, java.lang.String, android.os.Handler, int))(BroadcastReceiver (https://developer.android.com/reference/android/content/BroadcastReceiver.html) receiver, IntentFilter (https://developer.android.com/reference/android/content/IntentFilter.html) filter, String (https://developer.android.com/reference/java/lang/String.html) broadcastPermission, Handler (https://developer.android.com/reference/android/os/Handler.html) scheduler, int flags) Register to receive intent broadcasts, to run in the context of ***scheduler***.
abstract Intent (https://developer.android.com/reference/android/content/Intent.html)	registerReceiver (https://developer.android.com/reference/android/content/Context.html#registerReceiver(android.content.BroadcastReceiver, android.content.IntentFilter, java.lang.String, android.os.Handler))(BroadcastReceiver (https://developer.android.com/reference/android/content/BroadcastReceiver.html) receiver, IntentFilter (https://developer.android.com/reference/android/content/IntentFilter.html) filter, String (https://developer.android.com/reference/java/lang/String.html) broadcastPermission, Handler (https://developer.android.com/reference/android/os/Handler.html) scheduler) Register to receive intent broadcasts, to run in the context of ***scheduler***.
abstract void	removeStickyBroadcast (https://developer.android.com/reference/android/content/Context.html#removeStickyBroadcast(android.content.Intent))(Intent

	(https://developer.android.com/reference/android/content/Intent.html) intent) *This method was deprecated in API level 21. Sticky broadcasts should not be used. They provide no security (anyone can access them), no protection (anyone can modify them), and many other problems. The recommended pattern is to use a non-sticky broadcast to report that something has changed, with another mechanism for apps to retrieve the current value whenever desired.*
abstract void	removeStickyBroadcastAsUser (https://developer.android.com/reference/android/content/Context.html#removeStickyBroadcastAsUser(android.content.Intent, android.os.UserHandle))(Intent (https://developer.android.com/reference/android/content/Intent.html) intent, UserHandle (https://developer.android.com/reference/android/os/UserHandle.html) user) *This method was deprecated in API level 21. Sticky broadcasts should not be used. They provide no security (anyone can access them), no protection (anyone can modify them), and many other problems. The recommended pattern is to use a non-sticky broadcast to report that something has changed, with another mechanism for apps to retrieve the current value whenever desired.*
abstract void	revokeUriPermission (https://developer.android.com/reference/android/content/Context.html#revokeUriPermission(android.net.Uri, int))(Uri (https://developer.android.com/reference/android/net/Uri.html) uri, int modeFlags) Remove all permissions to access a particular content provider Uri that were previously added with grantUriPermission(String, Uri, int) (https://developer.android.com/reference/android/content/Context.html#grantUriPermission(java.lang.String, android.net.Uri, int)) or *any other* mechanism.
abstract void	revokeUriPermission (https://developer.android.com/reference/android/content/Context.html#revokeUriPermission(java.lang.String, android.net.Uri, int))(String (https://developer.android.com/reference/java/lang/String.html) toPackage, Uri (https://developer.android.com/reference/android/net/Uri.html) uri, int modeFlags) Remove permissions to access a particular content provider Uri that were previously added with grantUriPermission(String, Uri, int) (https://developer.android.com/reference/android/content/Context.html#grantUriPermission(java.lang.String, android.net.Uri, int)) for a specific target package.
abstract void	sendBroadcast (https://developer.android.com/reference/android/content/Context.html#sendBroadcast(android.content.Intent, java.lang.String))(Intent

	(https://developer.android.com/reference/android/content/Intent.html) intent, String (https://developer.android.com/reference/java/lang/String.html) receiverPermission) Broadcast the given intent to all interested BroadcastReceivers, allowing an optional required permission to be enforced.
abstract void	sendBroadcast (https://developer.android.com/reference/android/content/Context.html#sendBroadcast(android.content.Intent))(Intent (https://developer.android.com/reference/android/content/Intent.html) intent) Broadcast the given intent to all interested BroadcastReceivers.
abstract void	sendBroadcastAsUser (https://developer.android.com/reference/android/content/Context.html#sendBroadcastAsUser(android.content.Intent, android.os.UserHandle))(Intent (https://developer.android.com/reference/android/content/Intent.html) intent, UserHandle (https://developer.android.com/reference/android/os/UserHandle.html) user) Version of sendBroadcast(Intent) (https://developer.android.com/reference/android/content/Context.html#sendBroadcast(android.content.Intent)) that allows you to specify the user the broadcast will be sent to.
abstract void	sendBroadcastAsUser (https://developer.android.com/reference/android/content/Context.html#sendBroadcastAsUser(android.content.Intent, android.os.UserHandle, java.lang.String))(Intent (https://developer.android.com/reference/android/content/Intent.html) intent, UserHandle (https://developer.android.com/reference/android/os/UserHandle.html) user, String (https://developer.android.com/reference/java/lang/String.html) receiverPermission) Version of sendBroadcast(Intent, String) (https://developer.android.com/reference/android/content/Context.html#sendBroadcast(android.content.Intent, java.lang.String)) that allows you to specify the user the broadcast will be sent to.
abstract void	sendOrderedBroadcast (https://developer.android.com/reference/android/content/Context.html#sendOrderedBroadcast(android.content.Intent, java.lang.String, android.content.BroadcastReceiver, android.os.Handler, int, java.lang.String, android.os.Bundle))(Intent (https://developer.android.com/reference/android/content/Intent.html) intent, String (https://developer.android.com/reference/java/lang/String.html) receiverPermission, BroadcastReceiver (https://developer.android.com/reference/android/content/BroadcastReceiver.html) resultReceiver, Handler (https://developer.android.com/reference/android/os/Handler.html) scheduler, int initialCode, String (https://developer.android.com

	/reference/java/lang/String.html) initialData, Bundle (https://developer.android.com/reference/android/os/Bundle.html) initialExtras) Version of sendBroadcast(Intent) (https://developer.android.com/reference/android/content/Context.html#sendBroadcast(android.content.Intent)) that allows you to receive data back from the broadcast.
abstract void	sendOrderedBroadcast (https://developer.android.com/reference/android/content/Context.html#sendOrderedBroadcast(android.content.Intent, java.lang.String))(Intent (https://developer.android.com/reference/android/content/Intent.html) intent, String (https://developer.android.com/reference/java/lang/String.html) receiverPermission) Broadcast the given intent to all interested BroadcastReceivers, delivering them one at a time to allow more preferred receivers to consume the broadcast before it is delivered to less preferred receivers.
abstract void	sendOrderedBroadcastAsUser (https://developer.android.com/reference/android/content/Context.html#sendOrderedBroadcastAsUser(android.content.Intent, android.os.UserHandle, java.lang.String, android.content.BroadcastReceiver, android.os.Handler, int, java.lang.String, android.os.Bundle))(Intent (https://developer.android.com/reference/android/content/Intent.html) intent, UserHandle (https://developer.android.com/reference/android/os/UserHandle.html) user, String (https://developer.android.com/reference/java/lang/String.html) receiverPermission, BroadcastReceiver (https://developer.android.com/reference/android/content/BroadcastReceiver.html) resultReceiver, Handler (https://developer.android.com/reference/android/os/Handler.html) scheduler, int initialCode, String (https://developer.android.com/reference/java/lang/String.html) initialData, Bundle (https://developer.android.com/reference/android/os/Bundle.html) initialExtras) Version of sendOrderedBroadcast(Intent, String, BroadcastReceiver, Handler, int, String, Bundle) (https://developer.android.com/reference/android/content/Context.html#sendOrderedBroadcast(android.content.Intent, java.lang.String, android.content.BroadcastReceiver, android.os.Handler, int, java.lang.String, android.os.Bundle)) that allows you to specify the user the broadcast will be sent to.
abstract void	sendStickyBroadcast (https://developer.android.com/reference/android/content/Context.html#sendStickyBroadcast(android.content.Intent))(Intent (https://developer.android.com/reference/android/content/Intent.html) intent)

	This method was deprecated in API level 21. Sticky broadcasts should not be used. They provide no security (anyone can access them), no protection (anyone can modify them), and many other problems. The recommended pattern is to use a non-sticky broadcast to report that something has changed, with another mechanism for apps to retrieve the current value whenever desired.
abstract void	sendStickyBroadcastAsUser (https://developer.android.com/reference/android/content/Context.html#sendStickyBroadcastAsUser(android.content.Intent, android.os.UserHandle))(Intent (https://developer.android.com/reference/android/content/Intent.html) intent, UserHandle (https://developer.android.com/reference/android/os/UserHandle.html) user) *This method was deprecated in API level 21. Sticky broadcasts should not be used. They provide no security (anyone can access them), no protection (anyone can modify them), and many other problems. The recommended pattern is to use a non-sticky broadcast to report that something has changed, with another mechanism for apps to retrieve the current value whenever desired.*
abstract void	sendStickyOrderedBroadcast (https://developer.android.com/reference/android/content/Context.html#sendStickyOrderedBroadcast(android.content.Intent, android.content.BroadcastReceiver, android.os.Handler, int, java.lang.String, android.os.Bundle))(Intent (https://developer.android.com/reference/android/content/Intent.html) intent, BroadcastReceiver (https://developer.android.com/reference/android/content/BroadcastReceiver.html) resultReceiver, Handler (https://developer.android.com/reference/android/os/Handler.html) scheduler, int initialCode, String (https://developer.android.com/reference/java/lang/String.html) initialData, Bundle (https://developer.android.com/reference/android/os/Bundle.html) initialExtras) *This method was deprecated in API level 21. Sticky broadcasts should not be used. They provide no security (anyone can access them), no protection (anyone can modify them), and many other problems. The recommended pattern is to use a non-sticky broadcast to report that something has changed, with another mechanism for apps to retrieve the current value whenever desired.*
abstract void	sendStickyOrderedBroadcastAsUser (https://developer.android.com/reference/android/content/Context.html#sendStickyOrderedBroadcastAsUser(android.content.Intent, android.os.UserHandle, android.content.BroadcastReceiver, android.os.Handler, int, java.lang.String, android.os.Bundle))(Intent (https://developer.android.com/reference/android/content/Intent.html) intent,

	UserHandle (https://developer.android.com/reference/android/os/UserHandle.html) user, BroadcastReceiver (https://developer.android.com/reference/android/content/BroadcastReceiver.html) resultReceiver, Handler (https://developer.android.com/reference/android/os/Handler.html) scheduler, int initialCode, String (https://developer.android.com/reference/java/lang/String.html) initialData, Bundle (https://developer.android.com/reference/android/os/Bundle.html) initialExtras) *This method was deprecated in API level 21. Sticky broadcasts should not be used. They provide no security (anyone can access them), no protection (anyone can modify them), and many other problems. The recommended pattern is to use a non-sticky broadcast to report that something has changed, with another mechanism for apps to retrieve the current value whenever desired.*
abstract void	setTheme (https://developer.android.com/reference/android/content/Context.html#setTheme(int))(int resid) Set the base theme for this context.
abstract void	setWallpaper (https://developer.android.com/reference/android/content/Context.html#setWallpaper(android.graphics.Bitmap))(Bitmap (https://developer.android.com/reference/android/graphics/Bitmap.html) bitmap) *This method was deprecated in API level 5. Use* WallpaperManager.set() (https://developer.android.com/reference/android/app/WallpaperManager.html#setBitmap(android.graphics.Bitmap)) *instead.* *This method requires the caller to hold the permission* SET_WALLPAPER (https://developer.android.com/reference/android/Manifest.permission.html#SET_WALLPAPER).
abstract void	setWallpaper (https://developer.android.com/reference/android/content/Context.html#setWallpaper(java.io.InputStream))(InputStream (https://developer.android.com/reference/java/io/InputStream.html) data) *This method was deprecated in API level 5. Use* WallpaperManager.set() (https://developer.android.com/reference/android/app/WallpaperManager.html#setStream(java.io.InputStream)) *instead.* *This method requires the caller to hold the permission* SET_WALLPAPER (https://developer.android.com/reference/android/Manifest.permission.html#SET_WALLPAPER).
abstract void	startActivities (https://developer.android.com/reference/android/content/Context.html#startActivities(android.content.Intent[], android.os.Bundle))

	(Intent[] (https://developer.android.com/reference/android/content/Intent.html) intents, Bundle (https://developer.android.com/reference/android/os/Bundle.html) options) Launch multiple new activities.
abstract void	startActivities (https://developer.android.com/reference/android/content/Context.html#startActivities(android.content.Intent[]))(Intent[] (https://developer.android.com/reference/android/content/Intent.html) intents) Same as startActivities(Intent[], Bundle) (https://developer.android.com/reference/android/content/Context.html#startActivities(android.content.Intent[], android.os.Bundle)) with no options specified.
abstract void	startActivity (https://developer.android.com/reference/android/content/Context.html#startActivity(android.content.Intent))(Intent (https://developer.android.com/reference/android/content/Intent.html) intent) Same as startActivity(Intent, Bundle) (https://developer.android.com/reference/android/content/Context.html#startActivity(android.content.Intent, android.os.Bundle)) with no options specified.
abstract void	startActivity (https://developer.android.com/reference/android/content/Context.html#startActivity(android.content.Intent, android.os.Bundle)) (Intent (https://developer.android.com/reference/android/content/Intent.html) intent, Bundle (https://developer.android.com/reference/android/os/Bundle.html) options) Launch a new activity.
abstract ComponentName (https://developer.android.com/reference/android/content/ComponentName.html)	startForegroundService (https://developer.android.com/reference/android/content/Context.html#startForegroundService(android.content.Intent))(Intent (https://developer.android.com/reference/android/content/Intent.html) service) Similar to startService(Intent) (https://developer.android.com/reference/android/content/Context.html#startService(android.content.Intent)), but with an implicit promise that the Service will call startForeground(int, android.app.Notification) (https://developer.android.com/reference/android/app/Service.html#startForeground(int, android.app.Notification)) once it begins running.
abstract boolean	startInstrumentation (https://developer.android.com/reference/android/content/Context.html#startInstrumentation(android.content.ComponentName, java.lang.String, android.os.Bundle))(ComponentName

	(https://developer.android.com/reference/android/content/ComponentName.html) className, String (https://developer.android.com/reference/java/lang/String.html) profileFile, Bundle (https://developer.android.com/reference/android/os/Bundle.html) arguments) Start executing an Instrumentation (https://developer.android.com/reference/android/app/Instrumentation.html) class.
abstract void	startIntentSender (https://developer.android.com/reference/android/content/Context.html#startIntentSender(android.content.IntentSender, android.content.Intent, int, int, int))(IntentSender (https://developer.android.com/reference/android/content/IntentSender.html) intent, Intent (https://developer.android.com/reference/android/content/Intent.html) fillInIntent, int flagsMask, int flagsValues, int extraFlags) Same as startIntentSender(IntentSender, Intent, int, int, int, Bundle) (https://developer.android.com/reference/android/content/Context.html#startIntentSender(android.content.IntentSender, android.content.Intent, int, int, int, android.os.Bundle)) with no options specified.
abstract void	startIntentSender (https://developer.android.com/reference/android/content/Context.html#startIntentSender(android.content.IntentSender, android.content.Intent, int, int, int, android.os.Bundle))(IntentSender (https://developer.android.com/reference/android/content/IntentSender.html) intent, Intent (https://developer.android.com/reference/android/content/Intent.html) fillInIntent, int flagsMask, int flagsValues, int extraFlags, Bundle (https://developer.android.com/reference/android/os/Bundle.html) options) Like startActivity(Intent, Bundle) (https://developer.android.com/reference/android/content/Context.html#startActivity(android.content.Intent, android.os.Bundle)), but taking a IntentSender to start.
abstract ComponentName (https://developer.android.com/reference/android/content/ComponentName.html)	startService (https://developer.android.com/reference/android/content/Context.html#startService(android.content.Intent))(Intent (https://developer.android.com/reference/android/content/Intent.html) service) Request that a given application service be started.
abstract boolean	stopService (https://developer.android.com/reference/android/content/Context.html#stopService(android.content.Intent))(Intent (https://developer.android.com/reference/android/content/Intent.html) service)

	Request that a given application service be stopped.
abstract void	unbindService (https://developer.android.com/reference/android/content/Context.html#unbindService(android.content.ServiceConnection)) (ServiceConnection (https://developer.android.com/reference/android/content/ServiceConnection.html) conn) Disconnect from an application service.
void	unregisterComponentCallbacks (https://developer.android.com/reference/android/content/Context.html#unregisterComponentCallbacks(android.content.ComponentCallbacks)) (ComponentCallbacks (https://developer.android.com/reference/android/content/ComponentCallbacks.html) callback) Remove a ComponentCallbacks (https://developer.android.com/reference/android/content/ComponentCallbacks.html) object that was previously registered with registerComponentCallbacks(ComponentCallbacks) (https://developer.android.com/reference/android/content/Context.html#registerComponentCallbacks(android.content.ComponentCallbacks)).
abstract void	unregisterReceiver (https://developer.android.com/reference/android/content/Context.html#unregisterReceiver(android.content.BroadcastReceiver)) (BroadcastReceiver (https://developer.android.com/reference/android/content/BroadcastReceiver.html) receiver) Unregister a previously registered BroadcastReceiver.

Inherited methods

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

Constants

ACCESSIBILITY_SERVICEadded in API level 4 (https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

String (https://developer.android.com/reference/java/lang/String.html) ACCESSIBILITY_SERVICE

Use with getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>)) to retrieve a AccessibilityManager (https://developer.android.com/reference/android/view/accessibility/AccessibilityManager.html) for giving the user feedback for UI events through the registered event listeners.

See also:

getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>))

AccessibilityManager (https://developer.android.com/reference/android/view/accessibility/AccessibilityManager.html)

Constant Value: "accessibility"

ACCOUNT_SERVICE

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

String (https://developer.android.com/reference/java/lang/String.html) ACCOUNT_SERVICE

Use with getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>)) to retrieve a AccountManager (https://developer.android.com/reference/android/accounts/AccountManager.html) for receiving intents at a time of your choosing.

See also:

getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>))

AccountManager (https://developer.android.com/reference/android/accounts/AccountManager.html)

Constant Value: "account"

ACTIVITY_SERVICE

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

String (https://developer.android.com/reference/java/lang/String.html) ACTIVITY_SERVICE

Use with getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>)) to retrieve a ActivityManager (https://developer.android.com/reference/android/app/ActivityManager.html) for interacting with the global system state.

See also:

getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>))

ActivityManager (https://developer.android.com/reference/android/app/ActivityManager.html)

Constant Value: "activity"

ALARM_SERVICE

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

String (https://developer.android.com/reference/java/lang/String.html) ALARM_SERVICE

Use with getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>)) to retrieve a AlarmManager (https://developer.android.com/reference/android/app/AlarmManager.html) for receiving intents at a time of your choosing.

See also:

getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>))

AlarmManager (https://developer.android.com/reference/android/app/AlarmManager.html)

Constant Value: "alarm"

APPWIDGET_SERVICE

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

String (https://developer.android.com/reference/java/lang/String.html) APPWIDGET_SERVICE

Use with getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>)) to retrieve a AppWidgetManager (https://developer.android.com/reference/android/appwidget/AppWidgetManager.html) for accessing AppWidgets.

See also:

getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>))

Constant Value: "appwidget"

APP_OPS_SERVICE

added in API level 19 (https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

String (https://developer.android.com/reference/java/lang/String.html) APP_OPS_SERVICE

Use with getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>)) to retrieve a AppOpsManager (https://developer.android.com/reference/android/app/AppOpsManager.html) for tracking application operations on the device.

See also:

getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>))

AppOpsManager (https://developer.android.com/reference/android/app/AppOpsManager.html)

Constant Value: "appops"

AUDIO_SERVICE

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

String (https://developer.android.com/reference/java/lang/String.html) AUDIO_SERVICE

Use with getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>)) to retrieve a AudioManager (https://developer.android.com/reference/android/media/AudioManager.html) for handling management of volume, ringer modes and audio routing.

See also:

getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>))

AudioManager (https://developer.android.com/reference/android/media/AudioManager.html)

Constant Value: "audio"

BATTERY_SERVICE

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

String (https://developer.android.com/reference/java/lang/String.html) BATTERY_SERVICE

Use with getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>)) to retrieve a BatteryManager (https://developer.android.com/reference/android/os/BatteryManager.html) for managing battery state.

See also:

getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>))

Constant Value: "batterymanager"

BIND_ABOVE_CLIENT

added in API level 14 (https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

```
int BIND_ABOVE_CLIENT
```

Flag for bindService(Intent, ServiceConnection, int) (https://developer.android.com/reference/android/content/Context.html#bindService(android.content.Intent, android.content.ServiceConnection, int)): indicates that the client application binding to this service considers the service to be more important than the app itself. When set, the platform will try to have the out of memory killer kill the app before it kills the service it is bound to, though this is not guaranteed to be the case.

Constant Value: 8 (0x00000008)

BIND_ADJUST_WITH_ACTIVITY

added in API level 14 (https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

```
int BIND_ADJUST_WITH_ACTIVITY
```

Flag for bindService(Intent, ServiceConnection, int) (https://developer.android.com/reference/android/content/Context.html#bindService(android.content.Intent, android.content.ServiceConnection, int)): If binding from an activity, allow the target service's process importance to be raised based on whether the activity is visible to the user, regardless whether another flag is used to reduce the amount that the client process's overall importance is used to impact it.

Constant Value: 128 (0x00000080)

BIND_ALLOW_OOM_MANAGEMENT

added in API level 14 (https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

```
int BIND_ALLOW_OOM_MANAGEMENT
```

Flag for bindService(Intent, ServiceConnection, int) (https://developer.android.com/reference/android/content/Context.html#bindService(android.content.Intent, android.content.ServiceConnection, int)): allow the process hosting the bound service to go through its normal memory management. It will be treated more like a running service, allowing the system to (temporarily) expunge the process if low on memory or for some other whim it may have, and being more aggressive about making it a candidate to be killed (and restarted) if running for a long time.

Constant Value: 16 (0x00000010)

BIND_AUTO_CREATE

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

`int BIND_AUTO_CREATE`

Flag for `bindService(Intent, ServiceConnection, int)` (https://developer.android.com/reference/android/content/Context.html#bindService(android.content.Intent, android.content.ServiceConnection, int)): automatically create the service as long as the binding exists. Note that while this will create the service, its `onStartCommand(Intent, int, int)` (https://developer.android.com/reference/android/app/Service.html#onStartCommand(android.content.Intent, int, int)) method will still only be called due to an explicit call to `startService(Intent)` (https://developer.android.com/reference/android/content/Context.html#startService(android.content.Intent)). Even without that, though, this still provides you with access to the service object while the service is created.

Note that prior to `ICE_CREAM_SANDWICH` (https://developer.android.com/reference/android/os/Build.VERSION_CODES.html#ICE_CREAM_SANDWICH), not supplying this flag would also impact how important the system consider's the target service's process to be. When set, the only way for it to be raised was by binding from a service in which case it will only be important when that activity is in the foreground. Now to achieve this behavior you must explicitly supply the new flag `BIND_ADJUST_WITH_ACTIVITY` (https://developer.android.com/reference/android/content/Context.html#BIND_ADJUST_WITH_ACTIVITY). For compatibility, old applications that don't specify `BIND_AUTO_CREATE` (https://developer.android.com/reference/android/content/Context.html#BIND_AUTO_CREATE) will automatically have the flags `BIND_WAIVE_PRIORITY` (https://developer.android.com/reference/android/content/Context.html#BIND_WAIVE_PRIORITY) and `BIND_ADJUST_WITH_ACTIVITY` (https://developer.android.com/reference/android/content/Context.html#BIND_ADJUST_WITH_ACTIVITY) set for them in order to achieve the same result.

Constant Value: 1 (0x00000001)

BIND_DEBUG_UNBIND

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

`int BIND_DEBUG_UNBIND`

Flag for `bindService(Intent, ServiceConnection, int)` (https://developer.android.com/reference/android/content/Context.html#bindService(android.content.Intent, android.content.ServiceConnection, int)): include debugging help for mismatched calls to unbind. When this flag is set, the callstack of the following `unbindService(ServiceConnection)` (https://developer.android.com/reference/android/content

/Context.html#unbindService(android.content.ServiceConnection)) call is retained, to be printed if a later incorrect unbind call is made. Note that doing this requires retaining information about the binding that was made for the lifetime of the app, resulting in a leak -- this should only be used for debugging.

Constant Value: 2 (0x00000002)

BIND_EXTERNAL_SERVICE

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

```
int BIND_EXTERNAL_SERVICE
```

Flag for `bindService(Intent, ServiceConnection, int)` (https://developer.android.com/reference/android/content/Context.html#bindService(android.content.Intent, android.content.ServiceConnection, int)): The service being bound is an `isolated` (https://developer.android.com/reference/android/R.attr.html#isolatedProcess), `external` (https://developer.android.com/reference/android/R.attr.html#externalService) service. This binds the service into the calling application's package, rather than the package in which the service is declared.

When using this flag, the code for the service being bound will execute under the calling application's package name and user ID. Because the service must be an isolated process, it will not have direct access to the application's data, though. The purpose of this flag is to allow applications to provide services that are attributed to the app using the service, rather than the application providing the service.

Constant Value: -2147483648 (0x80000000)

BIND_IMPORTANT

added in API level 14 (https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

```
int BIND_IMPORTANT
```

Flag for `bindService(Intent, ServiceConnection, int)` (https://developer.android.com/reference/android/content/Context.html#bindService(android.content.Intent, android.content.ServiceConnection, int)): this service is very important to the client, so should be brought to the foreground process level when the client is. Normally a process can only be raised to the visibility level by a client, even if that client is in the foreground.

Constant Value: 64 (0x00000040)

BIND_NOT_FOREGROUND

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

```
int BIND_NOT_FOREGROUND
```

Flag for `bindService(Intent, ServiceConnection, int)` (https://developer.android.com/reference/android/content/Context.html#bindService(android.content.Intent, android.content.ServiceConnection, int)): don't allow this binding to raise the target service's process to the foreground scheduling priority. It will still be raised to at least the same memory priority as the client (so that its process will not be killable in any situation where the client is not killable), but for CPU scheduling purposes it may be left in the background. This only has an impact in the situation where the binding client is a foreground process and the target service is in a background process.

Constant Value: 4 (0x00000004)

BIND_WAIVE_PRIORITY

added in API level 14 (https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

`int BIND_WAIVE_PRIORITY`

Flag for `bindService(Intent, ServiceConnection, int)` (https://developer.android.com/reference/android/content/Context.html#bindService(android.content.Intent, android.content.ServiceConnection, int)): don't impact the scheduling or memory management priority of the target service's hosting process. Allows the service's process to be managed on the background LRU list just like a regular application process in the background.

Constant Value: 32 (0x00000020)

BLUETOOTH_SERVICE

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

`String` (https://developer.android.com/reference/java/lang/String.html) `BLUETOOTH_SERVICE`

Use with `getSystemService(Class)` (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>)) to retrieve a `BluetoothManager` (https://developer.android.com/reference/android/bluetooth/BluetoothManager.html) for using Bluetooth.

See also:

`getSystemService(Class)` (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>))

Constant Value: "bluetooth"

CAMERA_SERVICE

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

`String` (https://developer.android.com/reference/java/lang/String.html) `CAMERA_SERVICE`

Use with `getSystemService(Class)` (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>)) to retrieve a `CameraManager` (https://developer.android.com/reference/android/hardware/camera2/CameraManager.html) for interacting with camera devices.

See also:

`getSystemService(Class)` (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>))

`CameraManager` (https://developer.android.com/reference/android/hardware/camera2/CameraManager.html)

Constant Value: "camera"

CAPTIONING_SERVICE

added in API level 19 (https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

`String` (https://developer.android.com/reference/java/lang/String.html) `CAPTIONING_SERVICE`

Use with `getSystemService(Class)` (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>)) to retrieve a `CaptioningManager` (https://developer.android.com/reference/android/view/accessibility/CaptioningManager.html) for obtaining captioning properties and listening for changes in captioning preferences.

See also:

`getSystemService(Class)` (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>))

`CaptioningManager` (https://developer.android.com/reference/android/view/accessibility/CaptioningManager.html)

Constant Value: "captioning"

CARRIER_CONFIG_SERVICE

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

`String` (https://developer.android.com/reference/java/lang/String.html) `CARRIER_CONFIG_SERVICE`

Use with `getSystemService(Class)` (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>)) to retrieve a `CarrierConfigManager` (https://developer.android.com/reference/android/telephony/CarrierConfigManager.html) for reading carrier configuration values.

See also:

getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>))

CarrierConfigManager (https://developer.android.com/reference/android/telephony/CarrierConfigManager.html)

Constant Value: "carrier_config"

CLIPBOARD_SERVICE

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

String (https://developer.android.com/reference/java/lang/String.html) CLIPBOARD_SERVICE

Use with getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>)) to retrieve a ClipboardManager (https://developer.android.com/reference/android/text/ClipboardManager.html) for accessing and modifying ClipboardManager (https://developer.android.com/reference/android/content/ClipboardManager.html) for accessing and modifying the contents of the global clipboard.

See also:

getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>))

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

Constant Value: "clipboard"

COMPANION_DEVICE_SERVICE

added in API level 26 (https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

String (https://developer.android.com/reference/java/lang/String.html) COMPANION_DEVICE_SERVICE

Use with getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>)) to retrieve a CompanionDeviceManager (https://developer.android.com/reference/android/companion/CompanionDeviceManager.html) for managing companion devices

See also:

getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>))

CompanionDeviceManager (https://developer.android.com/reference/android/companion/CompanionDeviceManager.html)

Constant Value: "companiondevice"

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

String (https://developer.android.com/reference/java/lang/String.html) CONNECTIVITY_SERVICE

Use with getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>)) to retrieve a ConnectivityManager (https://developer.android.com/reference/android/net/ConnectivityManager.html) for handling management of network connections.

See also:

getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>))

ConnectivityManager (https://developer.android.com/reference/android/net/ConnectivityManager.html)

Constant Value: "connectivity"

CONSUMER_IR_SERVICEadded in API level 19 (https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

String (https://developer.android.com/reference/java/lang/String.html) CONSUMER_IR_SERVICE

Use with getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>)) to retrieve a ConsumerIrManager (https://developer.android.com/reference/android/hardware/ConsumerIrManager.html) for transmitting infrared signals from the device.

See also:

getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>))

ConsumerIrManager (https://developer.android.com/reference/android/hardware/ConsumerIrManager.html)

Constant Value: "consumer_ir"

CONTEXT_IGNORE_SECURITY

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

```
int CONTEXT_IGNORE_SECURITY
```

Flag for use with `createPackageContext(String, int)` (https://developer.android.com/reference/android/content/Context.html#createPackageContext(java.lang.String, int)): ignore any security restrictions on the Context being requested, allowing it to always be loaded. For use with `CONTEXT_INCLUDE_CODE` (https://developer.android.com/reference/android/content/Context.html#CONTEXT_INCLUDE_CODE) to allow code to be loaded into a process even when it isn't safe to do so. Use with extreme care!

Constant Value: 2 (0x00000002)

CONTEXT_INCLUDE_CODE

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

```
int CONTEXT_INCLUDE_CODE
```

Flag for use with `createPackageContext(String, int)` (https://developer.android.com/reference/android/content/Context.html#createPackageContext(java.lang.String, int)): include the application code with the context. This means loading code into the caller's process, so that `getClassLoader()` (https://developer.android.com/reference/android/content/Context.html#getClassLoader()) can be used to instantiate the application's classes. Setting this flags imposes security restrictions on what application context you can access; if the requested application can not be safely loaded into your process, java.lang.SecurityException will be thrown. If this flag is not set, there will be no restrictions on the packages that can be loaded, but `getClassLoader()` (https://developer.android.com/reference/android/content/Context.html#getClassLoader()) will always return the default system class loader.

Constant Value: 1 (0x00000001)

CONTEXT_RESTRICTED

added in API level 4 (https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

```
int CONTEXT_RESTRICTED
```

Flag for use with `createPackageContext(String, int)` (https://developer.android.com/reference/android/content/Context.html#createPackageContext(java.lang.String, int)): a restricted context may disable specific features. For instance, a View associated with a restricted context would ignore particular XML attributes.

Constant Value: 4 (0x00000004)

DEVICE_POLICY_SERVICE added in API level 8 (https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

String (https://developer.android.com/reference/java/lang/String.html) DEVICE_POLICY_SERVICE

Use with getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>)) to retrieve a DevicePolicyManager (https://developer.android.com/reference/android/app/admin/DevicePolicyManager.html) for working with global device policy management.

See also:

getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>))

Constant Value: "device_policy"

DISPLAY_SERVICE added in API level 17 (https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

String (https://developer.android.com/reference/java/lang/String.html) DISPLAY_SERVICE

Use with getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>)) to retrieve a DisplayManager (https://developer.android.com/reference/android/hardware/display/DisplayManager.html) for interacting with display devices.

See also:

getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>))

DisplayManager (https://developer.android.com/reference/android/hardware/display/DisplayManager.html)

Constant Value: "display"

DOWNLOAD_SERVICE added in API level 9 (https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

String (https://developer.android.com/reference/java/lang/String.html) DOWNLOAD_SERVICE

Use with getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>)) to retrieve a DownloadManager (https://developer.android.com/reference/android/app/DownloadManager.html) for requesting HTTP downloads.

See also:

getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>))

Constant Value: "download"

DROPBOX_SERVICE added in API level 8 (https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

String (https://developer.android.com/reference/java/lang/String.html) DROPBOX_SERVICE

Use with getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>)) to retrieve a DropBoxManager (https://developer.android.com/reference/android/os/DropBoxManager.html) instance for recording diagnostic logs.

See also:

getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>))

Constant Value: "dropbox"

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

String (https://developer.android.com/reference/java/lang/String.html) FINGERPRINT_SERVICE

Use with getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>)) to retrieve a FingerprintManager (https://developer.android.com/reference/android/hardware/fingerprint/FingerprintManager.html) for handling management of fingerprints.

See also:

getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>))

FingerprintManager (https://developer.android.com/reference/android/hardware/fingerprint/FingerprintManager.html)

Constant Value: "fingerprint"

HARDWARE_PROPERTIES_SERVICE

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

String (https://developer.android.com/reference/java/lang/String.html) HARDWARE_PROPERTIES_SERVICE

Use with getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>)) to retrieve a HardwarePropertiesManager (https://developer.android.com/reference/android/os/HardwarePropertiesManager.html) for accessing the hardware properties service.

See also:

getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>))

Constant Value: "hardware_properties"

INPUT_METHOD_SERVICE

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

String (https://developer.android.com/reference/java/lang/String.html) INPUT_METHOD_SERVICE

Use with getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>)) to retrieve a InputMethodManager (https://developer.android.com/reference/android/view/inputmethod/InputMethodManager.html) for accessing input methods.

See also:

getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>))

Constant Value: "input_method"

INPUT_SERVICE

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

String (https://developer.android.com/reference/java/lang/String.html) INPUT_SERVICE

Use with getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>)) to retrieve a InputManager (https://developer.android.com/reference/android/hardware/input/InputManager.html) for interacting with input devices.

See also:

getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>))

InputManager (https://developer.android.com/reference/android/hardware/input/InputManager.html)

Constant Value: "input"

JOB_SCHEDULER_SERVICE

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

String (https://developer.android.com/reference/java/lang/String.html) JOB_SCHEDULER_SERVICE

Use with getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>)) to retrieve a JobScheduler (https://developer.android.com/reference/android/app/job/JobScheduler.html) instance for managing occasional background tasks.

See also:

getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>))

JobScheduler (https://developer.android.com/reference/android/app/job/JobScheduler.html)

Constant Value: "jobscheduler"

KEYGUARD_SERVICE

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

String (https://developer.android.com/reference/java/lang/String.html) KEYGUARD_SERVICE

Use with getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>)) to retrieve a NotificationManager (https://developer.android.com/reference/android/app/NotificationManager.html) for controlling keyguard.

See also:

getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>))

KeyguardManager (https://developer.android.com/reference/android/app/KeyguardManager.html)

Constant Value: "keyguard"

LAUNCHER_APPS_SERVICE

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

String (https://developer.android.com/reference/java/lang/String.html) LAUNCHER_APPS_SERVICE

Use with getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>)) to retrieve a LauncherApps (https://developer.android.com/reference/android/content/pm/LauncherApps.html) for querying and monitoring launchable apps across profiles of a user.

See also:

getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>))

LauncherApps (https://developer.android.com/reference/android/content/pm/LauncherApps.html)

Constant Value: "launcherapps"

LAYOUT_INFLATER_SERVICE

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

String (https://developer.android.com/reference/java/lang/String.html) LAYOUT_INFLATER_SERVICE

Use with getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>)) to retrieve a LayoutInflater (https://developer.android.com/reference/android/view/LayoutInflater.html) for inflating layout resources in this context.

See also:

getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>))

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

Constant Value: "layout_inflater"

LOCATION_SERVICE

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

String (https://developer.android.com/reference/java/lang/String.html) LOCATION_SERVICE

Use with getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>)) to retrieve a LocationManager (https://developer.android.com

/reference/android/location/LocationManager.html) for controlling location updates.

See also:

getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>))

LocationManager (https://developer.android.com/reference/android/location/LocationManager.html)

Constant Value: "location"

MEDIA_PROJECTION_SERVICE

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

String (https://developer.android.com/reference/java/lang/String.html) MEDIA_PROJECTION_SERVICE

Use with getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>)) to retrieve a MediaProjectionManager (https://developer.android.com/reference/android/media/projection/MediaProjectionManager.html) instance for managing media projection sessions.

See also:

getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>))

MediaProjectionManager (https://developer.android.com/reference/android/media/projection/MediaProjectionManager.html)

Constant Value: "media_projection"

MEDIA_ROUTER_SERVICE

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

String (https://developer.android.com/reference/java/lang/String.html) MEDIA_ROUTER_SERVICE

Use with getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>)) to retrieve a MediaRouter (https://developer.android.com/reference/android/media/MediaRouter.html) for controlling and managing routing of media.

See also:

getSystemService(Class) (https://developer.android.com/reference/android/content

/Context.html#getSystemService(java.lang.Class<T>))

MediaRouter (https://developer.android.com/reference/android/media/MediaRouter.html)

Constant Value: "media_router"

MEDIA_SESSION_SERVICE

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

String (https://developer.android.com/reference/java/lang/String.html) MEDIA_SESSION_SERVICE

Use with getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>)) to retrieve a MediaSessionManager (https://developer.android.com/reference/android/media/session/MediaSessionManager.html) for managing media Sessions.

See also:

getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>))

MediaSessionManager (https://developer.android.com/reference/android/media/session/MediaSessionManager.html)

Constant Value: "media_session"

MIDI_SERVICE

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

String (https://developer.android.com/reference/java/lang/String.html) MIDI_SERVICE

Use with getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>)) to retrieve a MidiManager (https://developer.android.com/reference/android/media/midi/MidiManager.html) for accessing the MIDI service.

See also:

getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>))

Constant Value: "midi"

MODE_APPEND

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

```
int MODE_APPEND
```

File creation mode: for use with openFileOutput(String, int) (https://developer.android.com/reference/android/content/Context.html#openFileOutput(java.lang.String, int)), if the file already exists then write data to the end of the existing file instead of erasing it.

See also:

openFileOutput(String, int) (https://developer.android.com/reference/android/content/Context.html#openFileOutput(java.lang.String, int))

Constant Value: 32768 (0x00008000)

MODE_ENABLE_WRITE_AHEAD_LOGGING

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

```
int MODE_ENABLE_WRITE_AHEAD_LOGGING
```

Database open flag: when set, the database is opened with write-ahead logging enabled by default.

See also:

openOrCreateDatabase(String, int, CursorFactory) (https://developer.android.com/reference/android/content/Context.html#openOrCreateDatabase(java.lang.String, int, android.database.sqlite.SQLiteDatabase.CursorFactory))

openOrCreateDatabase(String, int, CursorFactory, DatabaseErrorHandler) (https://developer.android.com/reference/android/content/Context.html#openOrCreateDatabase(java.lang.String, int, android.database.sqlite.SQLiteDatabase.CursorFactory, android.database.DatabaseErrorHandler))

enableWriteAheadLogging() (https://developer.android.com/reference/android/database/sqlite/SQLiteDatabase.html#enableWriteAheadLogging())

Constant Value: 8 (0x00000008)

MODE_MULTI_PROCESS

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

```
int MODE_MULTI_PROCESS
```

> **This constant was deprecated in API level 23.**
> MODE_MULTI_PROCESS does not work reliably in some versions of Android, and furthermore does not provide any

> mechanism for reconciling concurrent modifications across processes. Applications should not attempt to use it. Instead, they should use an explicit cross-process data management approach such as `ContentProvider` (https://developer.android.com/reference/android/content/ContentProvider.html).

SharedPreference loading flag: when set, the file on disk will be checked for modification even if the shared preferences instance is already loaded in this process. This behavior is sometimes desired in cases where the application has multiple processes, all writing to the same SharedPreferences file. Generally there are better forms of communication between processes, though.

This was the legacy (but undocumented) behavior in and before Gingerbread (Android 2.3) and this flag is implied when targetting such releases. For applications targetting SDK versions *greater than* Android 2.3, this flag must be explicitly set if desired.

See also:

`getSharedPreferences(String, int)` (https://developer.android.com/reference/android/content/Context.html#getSharedPreferences(java.lang.String, int))

Constant Value: 4 (0x00000004)

MODE_NO_LOCALIZED_COLLATORS

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

```
int MODE_NO_LOCALIZED_COLLATORS
```

Database open flag: when set, the database is opened without support for localized collators.

See also:

`openOrCreateDatabase(String, int, CursorFactory)` (https://developer.android.com/reference/android/content/Context.html#openOrCreateDatabase(java.lang.String, int, android.database.sqlite.SQLiteDatabase.CursorFactory))

`openOrCreateDatabase(String, int, CursorFactory, DatabaseErrorHandler)` (https://developer.android.com/reference/android/content/Context.html#openOrCreateDatabase(java.lang.String, int, android.database.sqlite.SQLiteDatabase.CursorFactory, android.database.DatabaseErrorHandler))

`NO_LOCALIZED_COLLATORS` (https://developer.android.com/reference/android/database/sqlite/SQLiteDatabase.html#NO_LOCALIZED_COLLATORS)

Constant Value: 16 (0x00000010)

MODE_PRIVATE

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

```
int MODE_PRIVATE
```

File creation mode: the default mode, where the created file can only be accessed by the calling application (or all applications sharing the same user ID).

Constant Value: 0 (0x00000000)

MODE_WORLD_READABLE

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

```
int MODE_WORLD_READABLE
```

> **This constant was deprecated in API level 17.**
> Creating world-readable files is very dangerous, and likely to cause security holes in applications. It is strongly discouraged; instead, applications should use more formal mechanism for interactions such as `ContentProvider` (https://developer.android.com/reference/android/content/ContentProvider.html), `BroadcastReceiver` (https://developer.android.com/reference/android/content/BroadcastReceiver.html), and `Service` (https://developer.android.com/reference/android/app/Service.html). There are no guarantees that this access mode will remain on a file, such as when it goes through a backup and restore.

File creation mode: allow all other applications to have read access to the created file.

Starting from `N` (https://developer.android.com/reference/android/os/Build.VERSION_CODES.html#N), attempting to use this mode throws a `SecurityException` (https://developer.android.com/reference/java/lang/SecurityException.html).

See also:

`FileProvider` (https://developer.android.com/https://developer.android.com/reference/android/support/v4/content/FileProvider.html)

`FLAG_GRANT_WRITE_URI_PERMISSION` (https://developer.android.com/reference/android/content/Intent.html#FLAG_GRANT_WRITE_URI_PERMISSION)

Constant Value: 1 (0x00000001)

MODE_WORLD_WRITEABLE

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

```
int MODE_WORLD_WRITEABLE
```

> **This constant was deprecated in API level 17.**
> Creating world-writable files is very dangerous, and likely to cause security holes in applications. It is strongly discouraged; instead, applications should use more formal mechanism for interactions such as `ContentProvider` (https://developer.android.com/reference/android/content/ContentProvider.html), `BroadcastReceiver` (https://developer.android.com/reference/android/content/BroadcastReceiver.html), and `Service` (https://developer.android.com/reference/android/app/Service.html). There are no guarantees that this access mode will remain on a file, such as when it goes through a backup and restore.

File creation mode: allow all other applications to have write access to the created file.

Starting from `N` (https://developer.android.com/reference/android/os/Build.VERSION_CODES.html#N), attempting to use this mode will throw a `SecurityException` (https://developer.android.com/reference/java/lang/SecurityException.html).

See also:

`FileProvider` (https://developer.android.com/https://developer.android.com/reference/android/support/v4/content/FileProvider.html)

`FLAG_GRANT_WRITE_URI_PERMISSION` (https://developer.android.com/reference/android/content/Intent.html#FLAG_GRANT_WRITE_URI_PERMISSION)

Constant Value: 2 (0x00000002)

NETWORK_STATS_SERVICE

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

`String` (https://developer.android.com/reference/java/lang/String.html) `NETWORK_STATS_SERVICE`

Use with `getSystemService(Class)` (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>)) to retrieve a `NetworkStatsManager` (https://developer.android.com/reference/android/app/usage/NetworkStatsManager.html) for querying network usage stats.

See also:

`getSystemService(Class)` (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>))

`NetworkStatsManager` (https://developer.android.com/reference/android/app/usage/NetworkStatsManager.html)

Constant Value: "netstats"

NFC_SERVICE

added in API level 10 (https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

String (https://developer.android.com/reference/java/lang/String.html) NFC_SERVICE

Use with getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>)) to retrieve a NfcManager (https://developer.android.com/reference/android/nfc/NfcManager.html) for using NFC.

See also:

getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>))

Constant Value: "nfc"

NOTIFICATION_SERVICE

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

String (https://developer.android.com/reference/java/lang/String.html) NOTIFICATION_SERVICE

Use with getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>)) to retrieve a NotificationManager (https://developer.android.com/reference/android/app/NotificationManager.html) for informing the user of background events.

See also:

getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>))

NotificationManager (https://developer.android.com/reference/android/app/NotificationManager.html)

Constant Value: "notification"

NSD_SERVICE

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

String (https://developer.android.com/reference/java/lang/String.html) NSD_SERVICE

Use with getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>)) to retrieve a NsdManager (https://developer.android.com/reference/android/net/nsd/NsdManager.html) for handling management of network service discovery

See also:

getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>))

NsdManager (https://developer.android.com/reference/android/net/nsd/NsdManager.html)

Constant Value: "servicediscovery"

POWER_SERVICE

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

String (https://developer.android.com/reference/java/lang/String.html) POWER_SERVICE

Use with getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>)) to retrieve a PowerManager (https://developer.android.com/reference/android/os/PowerManager.html) for controlling power management, including "wake locks," which let you keep the device on while you're running long tasks.

Constant Value: "power"

PRINT_SERVICE

added in API level 19 (https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

String (https://developer.android.com/reference/java/lang/String.html) PRINT_SERVICE

PrintManager (https://developer.android.com/reference/android/print/PrintManager.html) for printing and managing printers and print tasks.

See also:

getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>))

PrintManager (https://developer.android.com/reference/android/print/PrintManager.html)

Constant Value: "print"

RECEIVER_VISIBLE_TO_INSTANT_APPS

added in API level 26 (https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

int RECEIVER_VISIBLE_TO_INSTANT_APPS

Flag for registerReceiver(BroadcastReceiver, IntentFilter) (https://developer.android.com/reference/android/content/Context.html#registerReceiver(android.content.BroadcastReceiver, android.content.IntentFilter)): The receiver can receive broadcasts from Instant Apps.

Constant Value: 1 (0x00000001)

RESTRICTIONS_SERVICE

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

String (https://developer.android.com/reference/java/lang/String.html) RESTRICTIONS_SERVICE

Use with getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>)) to retrieve a RestrictionsManager (https://developer.android.com/reference/android/content/RestrictionsManager.html) for retrieving application restrictions and requesting permissions for restricted operations.

See also:

getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>))

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

Constant Value: "restrictions"

SEARCH_SERVICE

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

String (https://developer.android.com/reference/java/lang/String.html) SEARCH_SERVICE

Use with getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>)) to retrieve a SearchManager (https://developer.android.com/reference/android/app/SearchManager.html) for handling searches.

See also:

getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>))

SearchManager (https://developer.android.com/reference/android/app/SearchManager.html)

Constant Value: "search"

SENSOR_SERVICE

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

String (https://developer.android.com/reference/java/lang/String.html) SENSOR_SERVICE

Use with getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>)) to retrieve a SensorManager (https://developer.android.com/reference/android/hardware/SensorManager.html) for accessing sensors.

See also:

getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>))

SensorManager (https://developer.android.com/reference/android/hardware/SensorManager.html)

Constant Value: "sensor"

SHORTCUT_SERVICE

added in API level 25 (https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

String (https://developer.android.com/reference/java/lang/String.html) SHORTCUT_SERVICE

Use with getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>)) to retrieve a ShortcutManager (https://developer.android.com/reference/android/content/pm/ShortcutManager.html) for accessing the launcher shortcut service.

See also:

getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>))

ShortcutManager (https://developer.android.com/reference/android/content/pm/ShortcutManager.html)

Constant Value: "shortcut"

STORAGE_SERVICE

added in API level 9 (https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

String (https://developer.android.com/reference/java/lang/String.html) STORAGE_SERVICE

Use with getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>)) to retrieve a StorageManager (https://developer.android.com/reference/android/os/storage/StorageManager.html) for accessing system storage functions.

See also:

getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>))

StorageManager (https://developer.android.com/reference/android/os/storage/StorageManager.html)

Constant Value: "storage"

STORAGE_STATS_SERVICE

added in API level 26 (https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

String (https://developer.android.com/reference/java/lang/String.html) STORAGE_STATS_SERVICE

Use with getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>)) to retrieve a StorageStatsManager (https://developer.android.com/reference/android/app/usage/StorageStatsManager.html) for accessing system storage statistics.

See also:

getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>))

StorageStatsManager (https://developer.android.com/reference/android/app/usage/StorageStatsManager.html)

Constant Value: "storagestats"

SYSTEM_HEALTH_SERVICE

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

String (https://developer.android.com/reference/java/lang/String.html) SYSTEM_HEALTH_SERVICE

Use with getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>)) to retrieve a SystemHealthManager (https://developer.android.com/reference/android/os/health/SystemHealthManager.html) for accessing system health (battery, power, memory, etc) metrics.

See also:

getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>))

Constant Value: "systemhealth"

TELECOM_SERVICE

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

String (https://developer.android.com/reference/java/lang/String.html) TELECOM_SERVICE

Use with getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>)) to retrieve a TelecomManager (https://developer.android.com/reference/android/telecom/TelecomManager.html) to manage telecom-related features of the device.

See also:

getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>))

TelecomManager (https://developer.android.com/reference/android/telecom/TelecomManager.html)

Constant Value: "telecom"

TELEPHONY_SERVICE

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

String (https://developer.android.com/reference/java/lang/String.html) TELEPHONY_SERVICE

Use with getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>)) to retrieve a TelephonyManager (https://developer.android.com/reference/android/telephony/TelephonyManager.html) for handling management the telephony features of the device.

See also:

getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>))

TelephonyManager (https://developer.android.com/reference/android/telephony/TelephonyManager.html)

Constant Value: "phone"

TELEPHONY_SUBSCRIPTION_SERVICE

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

String (https://developer.android.com/reference/java/lang/String.html) TELEPHONY_SUBSCRIPTION_SERVICE

Use with getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>)) to retrieve a SubscriptionManager (https://developer.android.com/reference/android/telephony/SubscriptionManager.html) for handling management the telephony subscriptions of the device.

See also:

getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>))

SubscriptionManager (https://developer.android.com/reference/android/telephony/SubscriptionManager.html)

Constant Value: "telephony_subscription_service"

TEXT_CLASSIFICATION_SERVICE

added in API level 26 (https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

String (https://developer.android.com/reference/java/lang/String.html) TEXT_CLASSIFICATION_SERVICE

Use with getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>)) to retrieve a TextClassificationManager (https://developer.android.com/reference/android/view/textclassifier/TextClassificationManager.html) for text classification services.

See also:

getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>))

TextClassificationManager (https://developer.android.com/reference/android/view/textclassifier/TextClassificationManager.html)

Constant Value: "textclassification"

TEXT_SERVICES_MANAGER_SERVICE

added in API level 14 (https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

String (https://developer.android.com/reference/java/lang/String.html) TEXT_SERVICES_MANAGER_SERVICE

Use with getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>)) to retrieve a TextServicesManager (https://developer.android.com/reference/android/view/textservice/TextServicesManager.html) for accessing text services.

See also:

getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>))

Constant Value: "textservices"

TV_INPUT_SERVICE

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

String (https://developer.android.com/reference/java/lang/String.html) TV_INPUT_SERVICE

Use with getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>)) to retrieve a TvInputManager (https://developer.android.com/reference/android/media/tv/TvInputManager.html) for interacting with TV inputs on the device.

See also:

getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>))

TvInputManager (https://developer.android.com/reference/android/media/tv/TvInputManager.html)

Constant Value: "tv_input"

UI_MODE_SERVICE

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

String (https://developer.android.com/reference/java/lang/String.html) UI_MODE_SERVICE

Use with getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>)) to retrieve a UiModeManager (https://developer.android.com/reference/android/app/UiModeManager.html) for controlling UI modes.

See also:

getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>))

Constant Value: "uimode"

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

String (https://developer.android.com/reference/java/lang/String.html) USAGE_STATS_SERVICE

Use with getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>)) to retrieve a UsageStatsManager (https://developer.android.com/reference/android/app/usage/UsageStatsManager.html) for querying device usage stats.

See also:

getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>))

UsageStatsManager (https://developer.android.com/reference/android/app/usage/UsageStatsManager.html)

Constant Value: "usagestats"

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

String (https://developer.android.com/reference/java/lang/String.html) USB_SERVICE

Use with getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>)) to retrieve a UsbManager (https://developer.android.com/reference/android/hardware/usb/UsbManager.html) for access to USB devices (as a USB host) and for controlling this device's behavior as a USB device.

See also:

getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>))

UsbManager (https://developer.android.com/reference/android/hardware/usb/UsbManager.html)

Constant Value: "usb"

USER_SERVICE added in API level 17 (https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

String (https://developer.android.com/reference/java/lang/String.html) USER_SERVICE

Use with getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>)) to retrieve a UserManager (https://developer.android.com

/reference/android/os/UserManager.html) for managing users on devices that support multiple users.

See also:

getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>))

UserManager (https://developer.android.com/reference/android/os/UserManager.html)

Constant Value: "user"

VIBRATOR_SERVICE

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

String (https://developer.android.com/reference/java/lang/String.html) VIBRATOR_SERVICE

Use with getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>)) to retrieve a Vibrator (https://developer.android.com/reference/android/os/Vibrator.html) for interacting with the vibration hardware.

See also:

getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>))

Vibrator (https://developer.android.com/reference/android/os/Vibrator.html)

Constant Value: "vibrator"

WALLPAPER_SERVICE

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

String (https://developer.android.com/reference/java/lang/String.html) WALLPAPER_SERVICE

Use with getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>)) to retrieve a com.android.server.WallpaperService for accessing wallpapers.

See also:

getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>))

Constant Value: "wallpaper"

WIFI_AWARE_SERVICE

added in API level 26 (https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

String (https://developer.android.com/reference/java/lang/String.html) WIFI_AWARE_SERVICE

Use with getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>)) to retrieve a WifiAwareManager (https://developer.android.com/reference/android/net/wifi/aware/WifiAwareManager.html) for handling management of Wi-Fi Aware.

See also:

getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>))

WifiAwareManager (https://developer.android.com/reference/android/net/wifi/aware/WifiAwareManager.html)

Constant Value: "wifiaware"

WIFI_P2P_SERVICE

added in API level 14 (https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

String (https://developer.android.com/reference/java/lang/String.html) WIFI_P2P_SERVICE

Use with getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>)) to retrieve a WifiP2pManager (https://developer.android.com/reference/android/net/wifi/p2p/WifiP2pManager.html) for handling management of Wi-Fi peer-to-peer connections.

See also:

getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>))

WifiP2pManager (https://developer.android.com/reference/android/net/wifi/p2p/WifiP2pManager.html)

Constant Value: "wifip2p"

WIFI_SERVICE

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

String (https://developer.android.com/reference/java/lang/String.html) WIFI_SERVICE

Use with getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>)) to retrieve a WifiManager (https://developer.android.com/reference/android/net/wifi/WifiManager.html) for handling management of Wi-Fi access.

See also:

getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>))

WifiManager (https://developer.android.com/reference/android/net/wifi/WifiManager.html)

Constant Value: "wifi"

WINDOW_SERVICE

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

String (https://developer.android.com/reference/java/lang/String.html) WINDOW_SERVICE

Use with getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>)) to retrieve a WindowManager (https://developer.android.com/reference/android/view/WindowManager.html) for accessing the system's window manager.

See also:

getSystemService(Class) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>))

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

Constant Value: "window"

Public constructors

Context

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

Context ()

Public methods

bindService

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

```
boolean bindService (Intent (https://developer.android.com/reference/android/content/Intent.html) service,
                ServiceConnection (https://developer.android.com/reference/android/content/ServiceConnection.h
                int flags)
```

Connect to an application service, creating it if needed. This defines a dependency between your application and the service. The given *conn* will receive the service object when it is created and be told if it dies and restarts. The service will be considered required by the system only for as long as the calling context exists. For example, if this Context is an Activity that is stopped, the service will not be required to continue running until the Activity is resumed.

This function will throw `SecurityException` (https://developer.android.com/reference/java/lang/SecurityException.html) if you do not have permission to bind to the given service.

> Note: this method *can not be called from a* `BroadcastReceiver` (https://developer.android.com/reference/android/content/BroadcastReceiver.html) *component*. A pattern you can use to communicate from a BroadcastReceiver to a Service is to call `startService(Intent)` (https://developer.android.com/reference/android/content/Context.html#startService(android.content.Intent)) with the arguments containing the command to be sent, with the service calling its `stopSelf(int)` (https://developer.android.com/reference/android/app/Service.html#stopSelf(int)) method when done executing that command. See the API demo App/Service/Service Start Arguments Controller for an illustration of this. It is okay, however, to use this method from a BroadcastReceiver that has been registered with `registerReceiver(BroadcastReceiver, IntentFilter)` (https://developer.android.com/reference/android/content/Context.html#registerReceiver(android.content.BroadcastReceiver, android.content.IntentFilter)), since the lifetime of this BroadcastReceiver is tied to another object (the one that registered it).

Parameters	
`service`	`Intent`: Identifies the service to connect to. The Intent must specify an explicit component name.
`conn`	`ServiceConnection`: Receives information as the service is started and stopped. This must be a valid ServiceConnection object; it must not be null.
`flags`	`int`: Operation options for the binding. May be 0, `BIND_AUTO_CREATE` (https://developer.android.com/reference/android/content/Context.html#BIND_AUTO_CREATE), `BIND_DEBUG_UNBIND` (https://developer.android.com/reference/android/content/Context.html#BIND_DEBUG_UNBIND), `BIND_NOT_FOREGROUND` (https://developer.android.com/reference/android/content/Context.html#BIND_NOT_FOREGROUND), `BIND_ABOVE_CLIENT` (https://developer.android.com/reference/android/content/Context.html#BIND_ABOVE_CLIENT), `BIND_ALLOW_OOM_MANAGEMENT` (https://developer.android.com/reference/android/content/Context.html#BIND_ALLOW_OOM_MANAGEMENT), or `BIND_WAIVE_PRIORITY` (https://developer.android.com/reference/android/content/Context.html#BIND_WAIVE_PRIORITY).

Returns

`boolean`	If you have successfully bound to the service, `true` is returned; `false` is returned if the connection is not made so you will not receive the service object. However, you should still call `unbindService(ServiceConnection)` (https://developer.android.com/reference/android/content/Context.html#unbindService(android.content.ServiceConnection)) to release the connection.

Throws	
`SecurityException` (https://developer.android.com/reference/java/lang/SecurityException.html)	If the caller does not have permission to access the service or the service can not be found.

See also:

`unbindService(ServiceConnection)` (https://developer.android.com/reference/android/content/Context.html#unbindService(android.content.ServiceConnection))

`startService(Intent)` (https://developer.android.com/reference/android/content/Context.html#startService(android.content.Intent))

`BIND_AUTO_CREATE` (https://developer.android.com/reference/android/content/Context.html#BIND_AUTO_CREATE)

`BIND_DEBUG_UNBIND` (https://developer.android.com/reference/android/content/Context.html#BIND_DEBUG_UNBIND)

`BIND_NOT_FOREGROUND` (https://developer.android.com/reference/android/content/Context.html#BIND_NOT_FOREGROUND)

checkCallingOrSelfPermission

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

```
int checkCallingOrSelfPermission (String (https://developer.android.com/reference/java/lang/String.html) pe
```

Determine whether the calling process of an IPC *or you* have been granted a particular permission. This is the same as `checkCallingPermission(String)` (https://developer.android.com/reference/android/content/Context.html#checkCallingPermission(java.lang.String)), except it grants your own permissions if you are not currently processing an IPC. Use with care!

Parameters	
`permission`	`String`: The name of the permission being checked. This value must never be `null`.

Returns

int	PERMISSION_GRANTED (https://developer.android.com/reference/android/content/pm/PackageManager.html#PERMISSION_GRANTED) if the calling pid/uid is allowed that permission, or PERMISSION_DENIED (https://developer.android.com/reference/android/content/pm/PackageManager.html#PERMISSION_DENIED) if it is not. Value is PERMISSION_GRANTED (https://developer.android.com/reference/android/content/pm/PackageManager.html#PERMISSION_GRANTED) or PERMISSION_DENIED (https://developer.android.com/reference/android/content/pm/PackageManager.html#PERMISSION_DENIED).

See also:

checkPermission(String, String) (https://developer.android.com/reference/android/content/pm/PackageManager.html#checkPermission(java.lang.String, java.lang.String))

checkPermission(String, int, int) (https://developer.android.com/reference/android/content/Context.html#checkPermission(java.lang.String, int, int))

checkCallingPermission(String) (https://developer.android.com/reference/android/content/Context.html#checkCallingPermission(java.lang.String))

checkCallingOrSelfUriPermission

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

```
int checkCallingOrSelfUriPermission (Uri (https://developer.android.com/reference/android/net/Uri.html) uri
                int modeFlags)
```

Determine whether the calling process of an IPC *or you* has been granted permission to access a specific URI. This is the same as checkCallingUriPermission(Uri, int) (https://developer.android.com/reference/android/content/Context.html#checkCallingUriPermission(android.net.Uri, int)), except it grants your own permissions if you are not currently processing an IPC. Use with care!

Parameters	
uri	Uri: The uri that is being checked.
modeFlags	int: The access modes to check. Value is either 0 or combination of FLAG_GRANT_READ_URI_PERMISSION (https://developer.android.com/reference/android/content/Intent.html#FLAG_GRANT_READ_URI_PERMISSION) or FLAG_GRANT_WRITE_URI_PERMISSION (https://developer.android.com/reference/android/content/Intent.html#FLAG_GRANT_WRITE_URI_PERMISSION).

Returns

int	PERMISSION_GRANTED (https://developer.android.com/reference/android/content/pm/PackageManager.html#PERMISSION_GRANTED) if the caller is allowed to access that uri, or PERMISSION_DENIED (https://developer.android.com/reference/android/content/pm/PackageManager.html#PERMISSION_DENIED) if it is not. Value is PERMISSION_GRANTED (https://developer.android.com/reference/android/content/pm/PackageManager.html#PERMISSION_GRANTED) or PERMISSION_DENIED (https://developer.android.com/reference/android/content/pm/PackageManager.html#PERMISSION_DENIED).

See also:

checkCallingUriPermission(Uri, int) (https://developer.android.com/reference/android/content/Context.html#checkCallingUriPermission(android.net.Uri, int))

checkCallingPermission

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

```
int checkCallingPermission (String (https://developer.android.com/reference/java/lang/String.html) permissi
```

Determine whether the calling process of an IPC you are handling has been granted a particular permission. This is basically the same as calling checkPermission(String, int, int) (https://developer.android.com/reference/android/content/Context.html#checkPermission(java.lang.String, int, int)) with the pid and uid returned by getCallingPid() (https://developer.android.com/reference/android/os/Binder.html#getCallingPid()) and getCallingUid() (https://developer.android.com/reference/android/os/Binder.html#getCallingUid()). One important difference is that if you are not currently processing an IPC, this function will always fail. This is done to protect against accidentally leaking permissions; you can use checkCallingOrSelfPermission(String) (https://developer.android.com/reference/android/content/Context.html#checkCallingOrSelfPermission(java.lang.String)) to avoid this protection.

Parameters	
permission	String: The name of the permission being checked. This value must never be null.

Returns	
int	PERMISSION_GRANTED (https://developer.android.com/reference/android/content/pm/PackageManager.html#PERMISSION_GRANTED) if the calling pid/uid is allowed that permission, or PERMISSION_DENIED (https://developer.android.com/reference/android/content/pm/PackageManager.html#PERMISSION_DENIED) if it is not. Value is PERMISSION_GRANTED (https://developer.android.com/reference/android/content/pm/PackageManager.html#PERMISSION_GRANTED) or PERMISSION_DENIED (https://developer.android.com

/reference/android/content/pm/PackageManager.html#PERMISSION_DENIED).

See also:

checkPermission(String, String) (https://developer.android.com/reference/android/content/pm/PackageManager.html#checkPermission(java.lang.String, java.lang.String))

checkPermission(String, int, int) (https://developer.android.com/reference/android/content/Context.html#checkPermission(java.lang.String, int, int))

checkCallingOrSelfPermission(String) (https://developer.android.com/reference/android/content/Context.html#checkCallingOrSelfPermission(java.lang.String))

checkCallingUriPermission

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

```
int checkCallingUriPermission (Uri (https://developer.android.com/reference/android/net/Uri.html) uri,
                int modeFlags)
```

Determine whether the calling process and user ID has been granted permission to access a specific URI. This is basically the same as calling checkUriPermission(Uri, int, int, int) (https://developer.android.com/reference/android/content/Context.html#checkUriPermission(android.net.Uri, int, int, int)) with the pid and uid returned by getCallingPid() (https://developer.android.com/reference/android/os/Binder.html#getCallingPid()) and getCallingUid() (https://developer.android.com/reference/android/os/Binder.html#getCallingUid()). One important difference is that if you are not currently processing an IPC, this function will always fail.

Parameters	
uri	Uri: The uri that is being checked.
modeFlags	int: The access modes to check. Value is either 0 or combination of FLAG_GRANT_READ_URI_PERMISSION (https://developer.android.com/reference/android/content/Intent.html#FLAG_GRANT_READ_URI_PERMISSION) or FLAG_GRANT_WRITE_URI_PERMISSION (https://developer.android.com/reference/android/content/Intent.html#FLAG_GRANT_WRITE_URI_PERMISSION).

Returns	
int	PERMISSION_GRANTED (https://developer.android.com/reference/android/content/pm/PackageManager.html#PERMISSION_GRANTED) if the caller is allowed to access that uri, or PERMISSION_DENIED (https://developer.android.com/reference/android/content/pm/PackageManager.html#PERMISSION_DENIED) if it is not.

Value is PERMISSION_GRANTED (https://developer.android.com/reference/android/content/pm/PackageManager.html#PERMISSION_GRANTED) or PERMISSION_DENIED (https://developer.android.com/reference/android/content/pm/PackageManager.html#PERMISSION_DENIED).

See also:

checkUriPermission(Uri, int, int, int) (https://developer.android.com/reference/android/content/Context.html#checkUriPermission(android.net.Uri, int, int, int))

checkPermission

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

```
int checkPermission (String (https://developer.android.com/reference/java/lang/String.html) permission,
                int pid,
                int uid)
```

Determine whether the given permission is allowed for a particular process and user ID running in the system.

Parameters	
permission	String: The name of the permission being checked. This value must never be null.
pid	int: The process ID being checked against. Must be > 0.
uid	int: The user ID being checked against. A uid of 0 is the root user, which will pass every permission check.

Returns	
int	PERMISSION_GRANTED (https://developer.android.com/reference/android/content/pm/PackageManager.html#PERMISSION_GRANTED) if the given pid/uid is allowed that permission, or PERMISSION_DENIED (https://developer.android.com/reference/android/content/pm/PackageManager.html#PERMISSION_DENIED) if it is not. Value is PERMISSION_GRANTED (https://developer.android.com/reference/android/content/pm/PackageManager.html#PERMISSION_GRANTED) or PERMISSION_DENIED (https://developer.android.com/reference/android/content/pm/PackageManager.html#PERMISSION_DENIED).

See also:

checkPermission(String, String) (https://developer.android.com/reference/android/content/pm/PackageManager.html#checkPermission(java.lang.String, java.lang.String))

checkCallingPermission(String) (https://developer.android.com/reference/android/content

/Context.html#checkCallingPermission(java.lang.String))

checkSelfPermission

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

```
int checkSelfPermission (String (https://developer.android.com/reference/java/lang/String.html) permission)
```

Determine whether *you* have been granted a particular permission.

Parameters	
permission	String: The name of the permission being checked. This value must never be null.

Returns	
int	PERMISSION_GRANTED (https://developer.android.com/reference/android/content/pm/PackageManager.html#PERMISSION_GRANTED) if you have the permission, or PERMISSION_DENIED (https://developer.android.com/reference/android/content/pm/PackageManager.html#PERMISSION_DENIED) if not. Value is PERMISSION_GRANTED (https://developer.android.com/reference/android/content/pm/PackageManager.html#PERMISSION_GRANTED) or PERMISSION_DENIED (https://developer.android.com/reference/android/content/pm/PackageManager.html#PERMISSION_DENIED).

See also:

checkPermission(String, String) (https://developer.android.com/reference/android/content/pm/PackageManager.html#checkPermission(java.lang.String, java.lang.String))

checkCallingPermission(String) (https://developer.android.com/reference/android/content/Context.html#checkCallingPermission(java.lang.String))

checkUriPermission

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

```
int checkUriPermission (Uri (https://developer.android.com/reference/android/net/Uri.html) uri,
                String (https://developer.android.com/reference/java/lang/String.html) readPermission,
                String (https://developer.android.com/reference/java/lang/String.html) writePermission,
                int pid,
                int uid,
                int modeFlags)
```

Check both a Uri and normal permission. This allows you to perform both checkPermission(String, int, int) (https://developer.android.com/reference/android/content/Context.html#checkPermission(java.lang.String, int, int)) and checkUriPermission(Uri, int, int, int) (https://developer.android.com/reference/android/content/Context.html#checkUriPermission(android.net.Uri, int, int, int)) in one call.

Parameters	
uri	Uri: The Uri whose permission is to be checked, or null to not do this check.
readPermission	String: The permission that provides overall read access, or null to not do this check.
writePermission	String: The permission that provides overall write access, or null to not do this check.
pid	int: The process ID being checked against. Must be > 0.
uid	int: The user ID being checked against. A uid of 0 is the root user, which will pass every permission check.
modeFlags	int: The access modes to check. Value is either 0 or combination of FLAG_GRANT_READ_URI_PERMISSION (https://developer.android.com/reference/android/content/Intent.html#FLAG_GRANT_READ_URI_PERMISSION) or FLAG_GRANT_WRITE_URI_PERMISSION (https://developer.android.com/reference/android/content/Intent.html#FLAG_GRANT_WRITE_URI_PERMISSION).

Returns	
int	PERMISSION_GRANTED (https://developer.android.com/reference/android/content/pm/PackageManager.html#PERMISSION_GRANTED) if the caller is allowed to access that uri or holds one of the given permissions, or PERMISSION_DENIED (https://developer.android.com/reference/android/content/pm/PackageManager.html#PERMISSION_DENIED) if it is not. Value is PERMISSION_GRANTED (https://developer.android.com/reference/android/content/pm/PackageManager.html#PERMISSION_GRANTED) or PERMISSION_DENIED (https://developer.android.com/reference/android/content/pm/PackageManager.html#PERMISSION_DENIED).

checkUriPermission

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

```
int checkUriPermission (Uri (https://developer.android.com/reference/android/net/Uri.html) uri,
                int pid,
                int uid,
                int modeFlags)
```

Determine whether a particular process and user ID has been granted permission to access a specific URI. This only checks for permissions that have been explicitly granted -- if the given process/uid has more general access to the

URI's content provider then this check will always fail.

Parameters	
uri	Uri: The uri that is being checked.
pid	int: The process ID being checked against. Must be > 0.
uid	int: The user ID being checked against. A uid of 0 is the root user, which will pass every permission check.
modeFlags	int: The access modes to check. Value is either 0 or combination of FLAG_GRANT_READ_URI_PERMISSION (https://developer.android.com/reference/android/content/Intent.html#FLAG_GRANT_READ_URI_PERMISSION) or FLAG_GRANT_WRITE_URI_PERMISSION (https://developer.android.com/reference/android/content/Intent.html#FLAG_GRANT_WRITE_URI_PERMISSION).

Returns	
int	PERMISSION_GRANTED (https://developer.android.com/reference/android/content/pm/PackageManager.html#PERMISSION_GRANTED) if the given pid/uid is allowed to access that uri, or PERMISSION_DENIED (https://developer.android.com/reference/android/content/pm/PackageManager.html#PERMISSION_DENIED) if it is not. Value is PERMISSION_GRANTED (https://developer.android.com/reference/android/content/pm/PackageManager.html#PERMISSION_GRANTED) or PERMISSION_DENIED (https://developer.android.com/reference/android/content/pm/PackageManager.html#PERMISSION_DENIED).

See also:

checkCallingUriPermission(Uri, int) (https://developer.android.com/reference/android/content/Context.html#checkCallingUriPermission(android.net.Uri, int))

clearWallpaper

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

```
void clearWallpaper ()
```

> **This method was deprecated in API level 5.**
> Use WallpaperManager.clear() (https://developer.android.com/reference/android/app/WallpaperManager.html#clear()) instead.

This method requires the caller to hold the permission SET_WALLPAPER (https://developer.android.com/reference/android/Manifest.permission.html#SET_WALLPAPER).

Throws	
`IOException` (https://developer.android.com/reference/java/io/IOException.html)	

createConfigurationContext

added in API level 17 (https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

`Context` (https://developer.android.com/reference/android/content/Context.html) `createConfigurationContext (Co`

Return a new Context object for the current Context but whose resources are adjusted to match the given Configuration. Each call to this method returns a new instance of a Context object; Context objects are not shared, however common state (ClassLoader, other Resources for the same configuration) may be so the Context itself can be fairly lightweight.

Parameters	
`overrideConfiguration`	`Configuration`: A `Configuration` (https://developer.android.com/reference/android/content/res/Configuration.html) specifying what values to modify in the base Configuration of the original Context's resources. If the base configuration changes (such as due to an orientation change), the resources of this context will also change except for those that have been explicitly overridden with a value here. This value must never be `null`.

Returns	
`Context` (https://developer.android.com/reference/android/content/Context.html)	A `Context` (https://developer.android.com/reference/android/content/Context.html) with the given configuration override.

createContextForSplit

added in API level 26 (https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

`Context` (https://developer.android.com/reference/android/content/Context.html) `createContextForSplit (String`

Return a new Context object for the given split name. The new Context has a ClassLoader and Resources object that can access the split's and all of its dependencies' code/resources. Each call to this method returns a new instance of a

Context object; Context objects are not shared, however common state (ClassLoader, other Resources for the same split) may be so the Context itself can be fairly lightweight.

Parameters	
`splitName`	`String`: The name of the split to include, as declared in the split's `AndroidManifest.xml`.

Returns	
`Context` (https://developer.android.com/reference/android/content/Context.html)	A `Context` (https://developer.android.com/reference/android/content/Context.html) with the given split's code and/or resources loaded.

Throws	
`PackageManager.NameNotFoundException` (https://developer.android.com/reference/android/content/pm/PackageManager.NameNotFoundException.html)	

createDeviceProtectedStorageContext

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

`Context (https://developer.android.com/reference/android/content/Context.html) createDeviceProtectedStorageCo`

Return a new Context object for the current Context but whose storage APIs are backed by device-protected storage.

On devices with direct boot, data stored in this location is encrypted with a key tied to the physical device, and it can be accessed immediately after the device has booted successfully, both *before and after* the user has authenticated with their credentials (such as a lock pattern or PIN).

Because device-protected data is available without user authentication, you should carefully limit the data you store using this Context. For example, storing sensitive authentication tokens or passwords in the device-protected area is strongly discouraged.

If the underlying device does not have the ability to store device-protected and credential-protected data using different keys, then both storage areas will become available at the same time. They remain as two distinct storage locations on disk, and only the window of availability changes.

Each call to this method returns a new instance of a Context object; Context objects are not shared, however common state (ClassLoader, other Resources for the same configuration) may be so the Context itself can be fairly lightweight.

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

See also:

isDeviceProtectedStorage() (https://developer.android.com/reference/android/content/Context.html#isDeviceProtectedStorage())

createDisplayContext

added in API level 17 (https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

```
Context (https://developer.android.com/reference/android/content/Context.html) createDisplayContext (Display
```

Return a new Context object for the current Context but whose resources are adjusted to match the metrics of the given Display. Each call to this method returns a new instance of a Context object; Context objects are not shared, however common state (ClassLoader, other Resources for the same configuration) may be so the Context itself can be fairly lightweight. The returned display Context provides a WindowManager (https://developer.android.com/reference/android/view/WindowManager.html) (see getSystemService(String) (https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.String))) that is configured to show windows on the given display. The WindowManager's getDefaultDisplay() (https://developer.android.com/reference/android/view/WindowManager.html#getDefaultDisplay()) method can be used to retrieve the Display from the returned Context.

Parameters	
display	Display: A Display (https://developer.android.com/reference/android/view/Display.html) object specifying the display for whose metrics the Context's resources should be tailored and upon which new windows should be shown. This value must never be null.

Returns	
Context (https://developer.android.com/reference/android/content/Context.html)	A Context (https://developer.android.com/reference/android/content/Context.html) for the display.

createPackageContext

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

```
Context (https://developer.android.com/reference/android/content/Context.html) createPackageContext (String (
                int flags)
```

Return a new Context object for the given application name. This Context is the same as what the named application gets when it is launched, containing the same resources and class loader. Each call to this method returns a new instance of a Context object; Context objects are not shared, however they share common state (Resources, ClassLoader, etc) so the Context instance itself is fairly lightweight.

Throws PackageManager.NameNotFoundException (https://developer.android.com/reference/android/content/pm/PackageManager.NameNotFoundException.html) if there is no application with the given package name.

Throws SecurityException (https://developer.android.com/reference/java/lang/SecurityException.html) if the Context requested can not be loaded into the caller's process for security reasons (see CONTEXT_INCLUDE_CODE (https://developer.android.com/reference/android/content/Context.html#CONTEXT_INCLUDE_CODE) for more information}.

Parameters	
packageName	String: Name of the application's package.
flags	int: Option flags. Value is either 0 or combination of CONTEXT_INCLUDE_CODE (https://developer.android.com/reference/android/content/Context.html#CONTEXT_INCLUDE_CODE), CONTEXT_IGNORE_SECURITY (https://developer.android.com/reference/android/content/Context.html#CONTEXT_IGNORE_SECURITY) or CONTEXT_RESTRICTED (https://developer.android.com/reference/android/content/Context.html#CONTEXT_RESTRICTED).

Returns	
Context (https://developer.android.com/reference/android/content/Context.html)	A Context (https://developer.android.com/reference/android/content/Context.html) for the application.

Throws	
SecurityException (https://developer.android.com/reference/java/lang/SecurityException.html)	
	if there is no application with the given package name.
PackageManager.NameNotFoundException (https://developer.android.com/reference/android/content	

/pm/PackageManager.NameNotFoundException.html)

databaseList

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

String[] (https://developer.android.com/reference/java/lang/String.html) databaseList ()

Returns an array of strings naming the private databases associated with this Context's application package.

Returns	
String[] (https://developer.android.com/reference/java/lang/String.html)	Array of strings naming the private databases.

See also:

openOrCreateDatabase(String, int, SQLiteDatabase.CursorFactory) (https://developer.android.com/reference/android/content/Context.html#openOrCreateDatabase(java.lang.String, int, android.database.sqlite.SQLiteDatabase.CursorFactory))

deleteDatabase(String) (https://developer.android.com/reference/android/content/Context.html#deleteDatabase(java.lang.String))

deleteDatabase

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

boolean deleteDatabase (String (https://developer.android.com/reference/java/lang/String.html) name)

Delete an existing private SQLiteDatabase associated with this Context's application package.

Parameters	
name	String: The name (unique in the application package) of the database.

Returns	
boolean	true if the database was successfully deleted; else false.

See also:

openOrCreateDatabase(String, int, SQLiteDatabase.CursorFactory) (https://developer.android.com

/reference/android/content/Context.html#openOrCreateDatabase(java.lang.String, int, android.database.sqlite.SQLiteDatabase.CursorFactory))

deleteFile

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

```
boolean deleteFile (String (https://developer.android.com/reference/java/lang/String.html) name)
```

Delete the given private file associated with this Context's application package.

Parameters	
name	String: The name of the file to delete; can not contain path separators.

Returns	
boolean	true if the file was successfully deleted; else false.

See also:

openFileInput(String) (https://developer.android.com/reference/android/content/Context.html#openFileInput(java.lang.String))

openFileOutput(String, int) (https://developer.android.com/reference/android/content/Context.html#openFileOutput(java.lang.String, int))

fileList() (https://developer.android.com/reference/android/content/Context.html#fileList())

delete() (https://developer.android.com/reference/java/io/File.html#delete())

deleteSharedPreferences

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

```
boolean deleteSharedPreferences (String (https://developer.android.com/reference/java/lang/String.html) nam
```

Delete an existing shared preferences file.

Parameters	
name	String: The name (unique in the application package) of the shared preferences file.

Returns	
boolean	true if the shared preferences file was successfully deleted; else false.

See also:

getSharedPreferences(String, int) (https://developer.android.com/reference/android/content/Context.html#getSharedPreferences(java.lang.String, int))

enforceCallingOrSelfPermission

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

```
void enforceCallingOrSelfPermission (String (https://developer.android.com/reference/java/lang/String.html)
                String (https://developer.android.com/reference/java/lang/String.html) message)
```

If neither you nor the calling process of an IPC you are handling has been granted a particular permission, throw a SecurityException (https://developer.android.com/reference/java/lang/SecurityException.html). This is the same as enforceCallingPermission(String, String) (https://developer.android.com/reference/android/content/Context.html#enforceCallingPermission(java.lang.String, java.lang.String)), except it grants your own permissions if you are not currently processing an IPC. Use with care!

Parameters	
permission	String: The name of the permission being checked. This value must never be null.
message	String: A message to include in the exception if it is thrown. This value may be null.

See also:

checkCallingOrSelfPermission(String) (https://developer.android.com/reference/android/content/Context.html#checkCallingOrSelfPermission(java.lang.String))

enforceCallingOrSelfUriPermission

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

```
void enforceCallingOrSelfUriPermission (Uri (https://developer.android.com/reference/android/net/Uri.html)
                int modeFlags,
                String (https://developer.android.com/reference/java/lang/String.html) message)
```

If the calling process of an IPC *or you* has not been granted permission to access a specific URI, throw SecurityException (https://developer.android.com/reference/java/lang/SecurityException.html). This is the same as enforceCallingUriPermission(Uri, int, String) (https://developer.android.com/reference/android/content/Context.html#enforceCallingUriPermission(android.net.Uri, int, java.lang.String)), except it grants your own permissions if you are not currently processing an IPC. Use with care!

Parameters	
uri	Uri: The uri that is being checked.
modeFlags	int: The access modes to enforce. Value is either 0 or combination of FLAG_GRANT_READ_URI_PERMISSION (https://developer.android.com/reference/android/content/Intent.html#FLAG_GRANT_READ_URI_PERMISSION) or FLAG_GRANT_WRITE_URI_PERMISSION (https://developer.android.com/reference/android/content/Intent.html#FLAG_GRANT_WRITE_URI_PERMISSION).
message	String: A message to include in the exception if it is thrown.

See also:

checkCallingOrSelfUriPermission(Uri, int) (https://developer.android.com/reference/android/content/Context.html#checkCallingOrSelfUriPermission(android.net.Uri, int))

enforceCallingPermission

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

```
void enforceCallingPermission (String (https://developer.android.com/reference/java/lang/String.html) permi
                String (https://developer.android.com/reference/java/lang/String.html) message)
```

If the calling process of an IPC you are handling has not been granted a particular permission, throw a SecurityException (https://developer.android.com/reference/java/lang/SecurityException.html). This is basically the same as calling enforcePermission(String, int, int, String) (https://developer.android.com/reference/android/content/Context.html#enforcePermission(java.lang.String, int, int, java.lang.String)) with the pid and uid returned by getCallingPid() (https://developer.android.com/reference/android/os/Binder.html#getCallingPid()) and getCallingUid() (https://developer.android.com/reference/android/os/Binder.html#getCallingUid()). One important difference is that if you are not currently processing an IPC, this function will always throw the SecurityException. This is done to protect against accidentally leaking permissions; you can use enforceCallingOrSelfPermission(String, String) (https://developer.android.com/reference/android/content/Context.html#enforceCallingOrSelfPermission(java.lang.String, java.lang.String)) to avoid this protection.

Parameters	
permission	String: The name of the permission being checked. This value must never be null.
message	String: A message to include in the exception if it is thrown. This value may be null.

See also:

checkCallingPermission(String) (https://developer.android.com/reference/android/content/Context.html#checkCallingPermission(java.lang.String))

enforceCallingUriPermission

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

```
void enforceCallingUriPermission (Uri (https://developer.android.com/reference/android/net/Uri.html) uri,
                int modeFlags,
                String (https://developer.android.com/reference/java/lang/String.html) message)
```

If the calling process and user ID has not been granted permission to access a specific URI, throw SecurityException (https://developer.android.com/reference/java/lang/SecurityException.html). This is basically the same as calling enforceUriPermission(Uri, int, int, int, String) (https://developer.android.com/reference/android/content/Context.html#enforceUriPermission(android.net.Uri, int, int, int, java.lang.String)) with the pid and uid returned by getCallingPid() (https://developer.android.com/reference/android/os/Binder.html#getCallingPid()) and getCallingUid() (https://developer.android.com/reference/android/os/Binder.html#getCallingUid()). One important difference is that if you are not currently processing an IPC, this function will always throw a SecurityException.

Parameters	
uri	Uri: The uri that is being checked.
modeFlags	int: The access modes to enforce. Value is either 0 or combination of FLAG_GRANT_READ_URI_PERMISSION (https://developer.android.com/reference/android/content/Intent.html#FLAG_GRANT_READ_URI_PERMISSION) or FLAG_GRANT_WRITE_URI_PERMISSION (https://developer.android.com/reference/android/content/Intent.html#FLAG_GRANT_WRITE_URI_PERMISSION).
message	String: A message to include in the exception if it is thrown.

See also:

checkCallingUriPermission(Uri, int) (https://developer.android.com/reference/android/content/Context.html#checkCallingUriPermission(android.net.Uri, int))

enforcePermission

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

```
void enforcePermission (String (https://developer.android.com/reference/java/lang/String.html) permission,
                int pid,
                int uid,
                String (https://developer.android.com/reference/java/lang/String.html) message)
```

If the given permission is not allowed for a particular process and user ID running in the system, throw a SecurityException (https://developer.android.com/reference/java/lang/SecurityException.html).

Parameters	
permission	String: The name of the permission being checked. This value must never be null.
pid	int: The process ID being checked against. Must be > 0.
uid	int: The user ID being checked against. A uid of 0 is the root user, which will pass every permission check.
message	String: A message to include in the exception if it is thrown. This value may be null.

See also:

checkPermission(String, int, int) (https://developer.android.com/reference/android/content/Context.html#checkPermission(java.lang.String, int, int))

enforceUriPermission

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

```
void enforceUriPermission (Uri (https://developer.android.com/reference/android/net/Uri.html) uri,
                String (https://developer.android.com/reference/java/lang/String.html) readPermission,
                String (https://developer.android.com/reference/java/lang/String.html) writePermission,
                int pid,
                int uid,
                int modeFlags,
                String (https://developer.android.com/reference/java/lang/String.html) message)
```

Enforce both a Uri and normal permission. This allows you to perform both enforcePermission(String, int, int, String) (https://developer.android.com/reference/android/content/Context.html#enforcePermission(java.lang.String, int, int, java.lang.String)) and enforceUriPermission(Uri, int, int, int, String) (https://developer.android.com/reference/android/content/Context.html#enforceUriPermission(android.net.Uri, int, int, int, java.lang.String)) in one call.

Parameters	
uri	Uri: The Uri whose permission is to be checked, or null to not do this check.

readPermission	String: The permission that provides overall read access, or null to not do this check.
writePermission	String: The permission that provides overall write access, or null to not do this check.
pid	int: The process ID being checked against. Must be > 0.
uid	int: The user ID being checked against. A uid of 0 is the root user, which will pass every permission check.
modeFlags	int: The access modes to enforce. Value is either 0 or combination of FLAG_GRANT_READ_URI_PERMISSION (https://developer.android.com/reference/android/content/Intent.html#FLAG_GRANT_READ_URI_PERMISSION) or FLAG_GRANT_WRITE_URI_PERMISSION (https://developer.android.com/reference/android/content/Intent.html#FLAG_GRANT_WRITE_URI_PERMISSION).
message	String: A message to include in the exception if it is thrown. This value may be null.

See also:

checkUriPermission(Uri, String, String, int, int, int) (https://developer.android.com/reference/android/content/Context.html#checkUriPermission(android.net.Uri, java.lang.String, java.lang.String, int, int, int))

enforceUriPermission

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

```
void enforceUriPermission (Uri (https://developer.android.com/reference/android/net/Uri.html) uri,
                int pid,
                int uid,
                int modeFlags,
                String (https://developer.android.com/reference/java/lang/String.html) message)
```

If a particular process and user ID has not been granted permission to access a specific URI, throw SecurityException (https://developer.android.com/reference/java/lang/SecurityException.html). This only checks for permissions that have been explicitly granted -- if the given process/uid has more general access to the URI's content provider then this check will always fail.

Parameters	
uri	Uri: The uri that is being checked.
pid	int: The process ID being checked against. Must be > 0.
uid	int: The user ID being checked against. A uid of 0 is the root user, which will pass every permission check.

modeFlags	int: The access modes to enforce. Value is either 0 or combination of FLAG_GRANT_READ_URI_PERMISSION (https://developer.android.com/reference/android/content/Intent.html#FLAG_GRANT_READ_URI_PERMISSION) or FLAG_GRANT_WRITE_URI_PERMISSION (https://developer.android.com/reference/android/content/Intent.html#FLAG_GRANT_WRITE_URI_PERMISSION).
message	String: A message to include in the exception if it is thrown.

See also:

checkUriPermission(Uri, int, int, int) (https://developer.android.com/reference/android/content/Context.html#checkUriPermission(android.net.Uri, int, int, int))

fileList

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

String[] (https://developer.android.com/reference/java/lang/String.html) fileList ()

Returns an array of strings naming the private files associated with this Context's application package.

Returns	
String[] (https://developer.android.com/reference/java/lang/String.html)	Array of strings naming the private files.

See also:

openFileInput(String) (https://developer.android.com/reference/android/content/Context.html#openFileInput(java.lang.String))

openFileOutput(String, int) (https://developer.android.com/reference/android/content/Context.html#openFileOutput(java.lang.String, int))

deleteFile(String) (https://developer.android.com/reference/android/content/Context.html#deleteFile(java.lang.String))

getApplicationContext

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

Context (https://developer.android.com/reference/android/content/Context.html) getApplicationContext ()

Return the context of the single, global Application object of the current process. This generally should only be used if you need a Context whose lifecycle is separate from the current context, that is tied to the lifetime of the process rather than the current component.

Consider for example how this interacts with `registerReceiver(BroadcastReceiver, IntentFilter)` (https://developer.android.com/reference/android/content/Context.html#registerReceiver(android.content.BroadcastReceiver, android.content.IntentFilter)):

- If used from an Activity context, the receiver is being registered within that activity. This means that you are expected to unregister before the activity is done being destroyed; in fact if you do not do so, the framework will clean up your leaked registration as it removes the activity and log an error. Thus, if you use the Activity context to register a receiver that is static (global to the process, not associated with an Activity instance) then that registration will be removed on you at whatever point the activity you used is destroyed.
- If used from the Context returned here, the receiver is being registered with the global state associated with your application. Thus it will never be unregistered for you. This is necessary if the receiver is associated with static data, not a particular component. However using the ApplicationContext elsewhere can easily lead to serious leaks if you forget to unregister, unbind, etc.

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

getApplicationInfo

added in API level 4 (https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

ApplicationInfo (https://developer.android.com/reference/android/content/pm/ApplicationInfo.html) getApplicatic

Return the full application info for this context's package.

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

getAssets

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

`AssetManager (https://developer.android.com/reference/android/content/res/AssetManager.html) getAssets ()`

Returns an AssetManager instance for the application's package.

Note: Implementations of this method should return an AssetManager instance that is consistent with the Resources instance returned by `getResources()` (https://developer.android.com/reference/android/content/Context.html#getResources()). For example, they should share the same `Configuration` (https://developer.android.com/reference/android/content/res/Configuration.html) object.

Returns	
`AssetManager` (https://developer.android.com/reference/android/content/res/AssetManager.html)	an AssetManager instance for the application's package

See also:

`getResources()` (https://developer.android.com/reference/android/content/Context.html#getResources())

getCacheDir

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

`File (https://developer.android.com/reference/java/io/File.html) getCacheDir ()`

Returns the absolute path to the application specific cache directory on the filesystem.

The system will automatically delete files in this directory as disk space is needed elsewhere on the device. The system will always delete older files first, as reported by `lastModified()` (https://developer.android.com/reference/java/io/File.html#lastModified()). If desired, you can exert more control over how files are deleted using `setCacheBehaviorGroup(File, boolean)` (https://developer.android.com/reference/android/os/storage/StorageManager.html#setCacheBehaviorGroup(java.io.File, boolean)) and `setCacheBehaviorTombstone(File, boolean)` (https://developer.android.com/reference/android/os/storage/StorageManager.html#setCacheBehaviorTombstone(java.io.File, boolean)).

Apps are strongly encouraged to keep their usage of cache space below the quota returned by `getCacheQuotaBytes(java.util.UUID)` (https://developer.android.com/reference/android/os/storage/StorageManager.html#getCacheQuotaBytes(java.util.UUID)). If your app goes above this quota, your cached files will be some of the first to be deleted when additional disk space is needed. Conversely, if your app stays under this quota, your cached files will be some of the last to be deleted when additional disk space is needed.

Note that your cache quota will change over time depending on how frequently the user interacts with your app, and depending on how much system-wide disk space is used.

The returned path may change over time if the calling app is moved to an adopted storage device, so only relative paths should be persisted.

Apps require no extra permissions to read or write to the returned path, since this path lives in their private storage.

Returns	
`File` (https://developer.android.com/reference/java/io/File.html)	The path of the directory holding application cache files.

See also:

`openFileOutput(String, int)` (https://developer.android.com/reference/android/content/Context.html#openFileOutput(java.lang.String, int))

`getFileStreamPath(String)` (https://developer.android.com/reference/android/content/Context.html#getFileStreamPath(java.lang.String))

`getDir(String, int)` (https://developer.android.com/reference/android/content/Context.html#getDir(java.lang.String, int))

`getExternalCacheDir()` (https://developer.android.com/reference/android/content/Context.html#getExternalCacheDir())

getClassLoader

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

`ClassLoader` (https://developer.android.com/reference/java/lang/ClassLoader.html) `getClassLoader ()`

Return a class loader you can use to retrieve classes in this package.

Returns	
`ClassLoader` (https://developer.android.com/reference/java/lang/ClassLoader.html)	

getCodeCacheDir

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

```
File (https://developer.android.com/reference/java/io/File.html) getCodeCacheDir ()
```

Returns the absolute path to the application specific cache directory on the filesystem designed for storing cached code.

The system will delete any files stored in this location both when your specific application is upgraded, and when the entire platform is upgraded.

This location is optimal for storing compiled or optimized code generated by your application at runtime.

The returned path may change over time if the calling app is moved to an adopted storage device, so only relative paths should be persisted.

Apps require no extra permissions to read or write to the returned path, since this path lives in their private storage.

Returns	
`File` (https://developer.android.com/reference/java/io/File.html)	The path of the directory holding application code cache files.

getColor

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

```
int getColor (int id)
```

Returns a color associated with a particular resource ID and styled for the current theme.

Parameters	
`id`	`int`: The desired resource identifier, as generated by the aapt tool. This integer encodes the package, type, and resource entry. The value 0 is an invalid identifier.

Returns	
`int`	A single color value in the form 0xAARRGGBB.

Throws	
`Resources.NotFoundException` (https://developer.android.com/reference/android/content/res/Resources.NotFoundException.html)	if the given ID does not exist.

getColorStateList

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

`ColorStateList (https://developer.android.com/reference/android/content/res/ColorStateList.html) getColorStateL`

Returns a color state list associated with a particular resource ID and styled for the current theme.

Parameters	
id	int: The desired resource identifier, as generated by the aapt tool. This integer encodes the package, type, and resource entry. The value 0 is an invalid identifier.

Returns	
ColorStateList (https://developer.android.com/reference/android/content/res/ColorStateList.html)	A color state list, or null if the resource could not be resolved.

Throws	
Resources.NotFoundException (https://developer.android.com/reference/android/content/res/Resources.NotFoundException.html)	if the given ID does not exist.

getContentResolver

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

`ContentResolver (https://developer.android.com/reference/android/content/ContentResolver.html) getContentResol`

Return a ContentResolver instance for your application's package.

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

getDataDir

File (https://developer.android.com/reference/java/io/File.html) getDataDir ()

Returns the absolute path to the directory on the filesystem where all private files belonging to this app are stored. Apps should not use this path directly; they should instead use getFilesDir() (https://developer.android.com/reference/android/content/Context.html#getFilesDir()), getCacheDir() (https://developer.android.com/reference/android/content/Context.html#getCacheDir()), getDir(String, int) (https://developer.android.com/reference/android/content/Context.html#getDir(java.lang.String, int)), or other storage APIs on this class.

The returned path may change over time if the calling app is moved to an adopted storage device, so only relative paths should be persisted.

No additional permissions are required for the calling app to read or write files under the returned path.

Returns	
File (https://developer.android.com/reference/java/io/File.html)	

See also:

dataDir (https://developer.android.com/reference/android/content/pm/ApplicationInfo.html#dataDir)

getDatabasePath

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

File (https://developer.android.com/reference/java/io/File.html) getDatabasePath (String (https://developer.and

Returns the absolute path on the filesystem where a database created with openOrCreateDatabase(String, int, SQLiteDatabase.CursorFactory) (https://developer.android.com/reference/android/content/Context.html#openOrCreateDatabase(java.lang.String, int, android.database.sqlite.SQLiteDatabase.CursorFactory)) is stored.

The returned path may change over time if the calling app is moved to an adopted storage device, so only relative paths should be persisted.

Parameters	
name	String: The name of the database for which you would like to get its path.

Returns	
File (https://developer.android.com	An absolute path to the given database.

/reference/java/io/File.html)

See also:

openOrCreateDatabase(String, int, SQLiteDatabase.CursorFactory) (https://developer.android.com/reference/android/content/Context.html#openOrCreateDatabase(java.lang.String, int, android.database.sqlite.SQLiteDatabase.CursorFactory))

getDir

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

```
File (https://developer.android.com/reference/java/io/File.html) getDir (String (https://developer.android.com/re
                int mode)
```

Retrieve, creating if needed, a new directory in which the application can place its own custom data files. You can use the returned File object to create and access files in this directory. Note that files created through a File object will only be accessible by your own application; you can only set the mode of the entire directory, not of individual files.

The returned path may change over time if the calling app is moved to an adopted storage device, so only relative paths should be persisted.

Apps require no extra permissions to read or write to the returned path, since this path lives in their private storage.

Parameters	
name	String: Name of the directory to retrieve. This is a directory that is created as part of your application data.
mode	int: Operating mode. Value is either 0 or combination of MODE_PRIVATE (https://developer.android.com/reference/android/content/Context.html#MODE_PRIVATE), MODE_WORLD_READABLE (https://developer.android.com/reference/android/content/Context.html#MODE_WORLD_READABLE), MODE_WORLD_WRITEABLE (https://developer.android.com/reference/android/content/Context.html#MODE_WORLD_WRITEABLE) or MODE_APPEND (https://developer.android.com/reference/android/content/Context.html#MODE_APPEND).

Returns	
File (https://developer.android.com/reference/java/io/File.html)	A File (https://developer.android.com/reference/java/io/File.html) object for the requested directory. The directory will have been created if it does not already exist.

See also:

openFileOutput(String, int) (https://developer.android.com/reference/android/content

/Context.html#openFileOutput(java.lang.String, int))

getDrawable

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

Drawable (https://developer.android.com/reference/android/graphics/drawable/Drawable.html) getDrawable (int id)

Returns a drawable object associated with a particular resource ID and styled for the current theme.

Parameters	
id	int: The desired resource identifier, as generated by the aapt tool. This integer encodes the package, type, and resource entry. The value 0 is an invalid identifier.

Returns	
Drawable (https://developer.android.com/reference/android/graphics/drawable/Drawable.html)	An object that can be used to draw this resource, or null if the resource could not be resolved.

Throws	
Resources.NotFoundException (https://developer.android.com/reference/android/content/res/Resources.NotFoundException.html)	if the given ID does not exist.

getExternalCacheDir

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

File (https://developer.android.com/reference/java/io/File.html) getExternalCacheDir ()

Returns absolute path to application-specific directory on the primary shared/external storage device where the application can place cache files it owns. These files are internal to the application, and not typically visible to the user as media.

This is like getCacheDir() (https://developer.android.com/reference/android/content/Context.html#getCacheDir()) in that these files will be deleted when the application is uninstalled, however there are some important differences:

- The platform does not always monitor the space available in shared storage, and thus may not automatically delete these files. Apps should always manage the maximum space used in this location. Currently the only time files here will be deleted by the platform is when running on JELLY_BEAN_MR1 (https://developer.android.com/reference/android/os/Build.VERSION_CODES.html#JELLY_BEAN_MR1) or later and isExternalStorageEmulated(File) (https://developer.android.com/reference/android/os/Environment.html#isExternalStorageEmulated(java.io.File)) returns true.

- Shared storage may not always be available, since removable media can be ejected by the user. Media state can be checked using getExternalStorageState(File) (https://developer.android.com/reference/android/os/Environment.html#getExternalStorageState(java.io.File)).
- There is no security enforced with these files. For example, any application holding WRITE_EXTERNAL_STORAGE (https://developer.android.com/reference/android/Manifest.permission.html#WRITE_EXTERNAL_STORAGE) can write to these files.

If a shared storage device is emulated (as determined by isExternalStorageEmulated(File) (https://developer.android.com/reference/android/os/Environment.html#isExternalStorageEmulated(java.io.File))), its contents are backed by a private user data partition, which means there is little benefit to storing data here instead of the private directory returned by getCacheDir() (https://developer.android.com/reference/android/content/Context.html#getCacheDir()).

Starting in KITKAT (https://developer.android.com/reference/android/os/Build.VERSION_CODES.html#KITKAT), no permissions are required to read or write to the returned path; it's always accessible to the calling app. This only applies to paths generated for package name of the calling application. To access paths belonging to other packages, WRITE_EXTERNAL_STORAGE (https://developer.android.com/reference/android/Manifest.permission.html#WRITE_EXTERNAL_STORAGE) and/or READ_EXTERNAL_STORAGE (https://developer.android.com/reference/android/Manifest.permission.html#READ_EXTERNAL_STORAGE) are required.

On devices with multiple users (as described by UserManager (https://developer.android.com/reference/android/os/UserManager.html)), each user has their own isolated shared storage. Applications only have access to the shared storage for the user they're running as.

The returned path may change over time if different shared storage media is inserted, so only relative paths should be persisted.

Returns	
File (https://developer.android.com/reference/java/io/File.html)	the absolute path to application-specific directory. May return null if shared storage is not currently available.

See also:

getCacheDir() (https://developer.android.com/reference/android/content/Context.html#getCacheDir())

getExternalCacheDirs() (https://developer.android.com/reference/android/content/Context.html#getExternalCacheDirs())

`getExternalStorageState(File)` (https://developer.android.com/reference/android/os/Environment.html#getExternalStorageState(java.io.File))

`isExternalStorageEmulated(File)` (https://developer.android.com/reference/android/os/Environment.html#isExternalStorageEmulated(java.io.File))

`isExternalStorageRemovable(File)` (https://developer.android.com/reference/android/os/Environment.html#isExternalStorageRemovable(java.io.File))

getExternalCacheDirs

added in API level 19 (https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

`File[]` (https://developer.android.com/reference/java/io/File.html) `getExternalCacheDirs ()`

Returns absolute paths to application-specific directories on all shared/external storage devices where the application can place cache files it owns. These files are internal to the application, and not typically visible to the user as media.

This is like `getCacheDir()` (https://developer.android.com/reference/android/content/Context.html#getCacheDir()) in that these files will be deleted when the application is uninstalled, however there are some important differences:

- The platform does not always monitor the space available in shared storage, and thus may not automatically delete these files. Apps should always manage the maximum space used in this location. Currently the only time files here will be deleted by the platform is when running on `JELLY_BEAN_MR1` (https://developer.android.com/reference/android/os/Build.VERSION_CODES.html#JELLY_BEAN_MR1) or later and `isExternalStorageEmulated(File)` (https://developer.android.com/reference/android/os/Environment.html#isExternalStorageEmulated(java.io.File)) returns true.
- Shared storage may not always be available, since removable media can be ejected by the user. Media state can be checked using `getExternalStorageState(File)` (https://developer.android.com/reference/android/os/Environment.html#getExternalStorageState(java.io.File)).
- There is no security enforced with these files. For example, any application holding `WRITE_EXTERNAL_STORAGE` (https://developer.android.com/reference/android/Manifest.permission.html#WRITE_EXTERNAL_STORAGE) can write to these files.

If a shared storage device is emulated (as determined by `isExternalStorageEmulated(File)` (https://developer.android.com/reference/android/os/Environment.html#isExternalStorageEmulated(java.io.File))), it's contents are backed by a private user data partition, which means there is little benefit to storing data here instead of the private directory returned by `getCacheDir()` (https://developer.android.com/reference/android/content/Context.html#getCacheDir()).

Shared storage devices returned here are considered a stable part of the device, including physical media slots under a protective cover. The returned paths do not include transient devices, such as USB flash drives connected to handheld devices.

An application may store data on any or all of the returned devices. For example, an app may choose to store large files on the device with the most available space, as measured by StatFs (https://developer.android.com/reference/android/os/StatFs.html).

No additional permissions are required for the calling app to read or write files under the returned path. Write access outside of these paths on secondary external storage devices is not available.

The returned paths may change over time if different shared storage media is inserted, so only relative paths should be persisted.

Returns	
File[] (https://developer.android.com/reference/java/io/File.html)	the absolute paths to application-specific directories. Some individual paths may be null if that shared storage is not currently available. The first path returned is the same as getExternalCacheDir() (https://developer.android.com/reference/android/content/Context.html#getExternalCacheDir()).

See also:

getExternalCacheDir() (https://developer.android.com/reference/android/content/Context.html#getExternalCacheDir())

getExternalStorageState(File) (https://developer.android.com/reference/android/os/Environment.html#getExternalStorageState(java.io.File))

isExternalStorageEmulated(File) (https://developer.android.com/reference/android/os/Environment.html#isExternalStorageEmulated(java.io.File))

isExternalStorageRemovable(File) (https://developer.android.com/reference/android/os/Environment.html#isExternalStorageRemovable(java.io.File))

getExternalFilesDir

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

```
File (https://developer.android.com/reference/java/io/File.html) getExternalFilesDir (String (https://develope
```

Returns the absolute path to the directory on the primary shared/external storage device where the application can place persistent files it owns. These files are internal to the applications, and not typically visible to the user as media.

This is like getFilesDir() (https://developer.android.com/reference/android/content/Context.html#getFilesDir()) in that these files will be deleted when the application is uninstalled, however there are some important differences:

- Shared storage may not always be available, since removable media can be ejected by the user. Media state can be checked using getExternalStorageState(File) (https://developer.android.com/reference/android/os/Environment.html#getExternalStorageState(java.io.File)).
- There is no security enforced with these files. For example, any application holding WRITE_EXTERNAL_STORAGE (https://developer.android.com/reference/android/Manifest.permission.html#WRITE_EXTERNAL_STORAGE) can write to these files.

If a shared storage device is emulated (as determined by isExternalStorageEmulated(File) (https://developer.android.com/reference/android/os/Environment.html#isExternalStorageEmulated(java.io.File))), it's contents are backed by a private user data partition, which means there is little benefit to storing data here instead of the private directories returned by getFilesDir() (https://developer.android.com/reference/android/content/Context.html#getFilesDir()), etc.

Starting in KITKAT (https://developer.android.com/reference/android/os/Build.VERSION_CODES.html#KITKAT), no permissions are required to read or write to the returned path; it's always accessible to the calling app. This only applies to paths generated for package name of the calling application. To access paths belonging to other packages, WRITE_EXTERNAL_STORAGE (https://developer.android.com/reference/android/Manifest.permission.html#WRITE_EXTERNAL_STORAGE) and/or READ_EXTERNAL_STORAGE (https://developer.android.com/reference/android/Manifest.permission.html#READ_EXTERNAL_STORAGE) are required.

On devices with multiple users (as described by UserManager (https://developer.android.com/reference/android/os/UserManager.html)), each user has their own isolated shared storage. Applications only have access to the shared storage for the user they're running as.

The returned path may change over time if different shared storage media is inserted, so only relative paths should be persisted.

Here is an example of typical code to manipulate a file in an application's shared storage:

```
void createExternalStoragePrivateFile() {
    // Create a path where we will place our private file on external
    // storage.
    File file = new File(getExternalFilesDir(null), "DemoFile.jpg");

    try {
        // Very simple code to copy a picture from the application's
        // resource into the external file.  Note that this code does
        // no error checking, and assumes the picture is small (does not
        // try to copy it in chunks).  Note that if external storage is
        // not currently mounted this will silently fail.
        InputStream is = getResources().openRawResource(R.drawable.balloons);
        OutputStream os = new FileOutputStream(file);
        byte[] data = new byte[is.available()];
        is.read(data);
        os.write(data);
        is.close();
        os.close();
    } catch (IOException e) {
        // Unable to create file, likely because external storage is
        // not currently mounted.
        Log.w("ExternalStorage", "Error writing " + file, e);
    }
}

void deleteExternalStoragePrivateFile() {
    // Get path for the file on external storage.  If external
    // storage is not currently mounted this will fail.
    File file = new File(getExternalFilesDir(null), "DemoFile.jpg");
    if (file != null) {
        file.delete();
    }
}

boolean hasExternalStoragePrivateFile() {
    // Get path for the file on external storage.  If external
    // storage is not currently mounted this will fail.
    File file = new File(getExternalFilesDir(null), "DemoFile.jpg");
    if (file != null) {
        return file.exists();
    }
    return false;
}
```

If you supply a non-null *type* to this function, the returned file will be a path to a sub-directory of the given type. Though these files are not automatically scanned by the media scanner, you can explicitly add them to the media database with `MediaScannerConnection.scanFile` (https://developer.android.com/reference/android/media/MediaScannerConnection.html#scanFile(android.content.Context, java.lang.String[], java.lang.String[], android.media.MediaScannerConnection.OnScanCompletedListener)). Note that this is not the same as `Environment.getExternalStoragePublicDirectory()` (https://developer.android.com/reference/android/os/Environment.html#getExternalStoragePublicDirectory(java.lang.String)), which provides directories of media shared by all applications. The directories returned here are owned by the application, and their contents will be removed when the application is uninstalled. Unlike `Environment.getExternalStoragePublicDirectory()` (https://developer.android.com/reference/android/os/Environment.html#getExternalStoragePublicDirectory(java.lang.String)), the directory returned here will be automatically created for you.

Here is an example of typical code to manipulate a picture in an application's shared storage and add it to the media database:

```
void createExternalStoragePrivatePicture() {
    // Create a path where we will place our picture in our own private
    // pictures directory.  Note that we don't really need to place a
    // picture in DIRECTORY_PICTURES, since the media scanner will see
    // all media in these directories; this may be useful with other
    // media types such as DIRECTORY_MUSIC however to help it classify
    // your media for display to the user.
    File path = getExternalFilesDir(Environment.DIRECTORY_PICTURES);
    File file = new File(path, "DemoPicture.jpg");

    try {
        // Very simple code to copy a picture from the application's
        // resource into the external file.  Note that this code does
        // no error checking, and assumes the picture is small (does not
        // try to copy it in chunks).  Note that if external storage is
        // not currently mounted this will silently fail.
        InputStream is = getResources().openRawResource(R.drawable.balloons);
        OutputStream os = new FileOutputStream(file);
        byte[] data = new byte[is.available()];
        is.read(data);
        os.write(data);
        is.close();
        os.close();

        // Tell the media scanner about the new file so that it is
        // immediately available to the user.
        MediaScannerConnection.scanFile(this,
                new String[] { file.toString() }, null,
                new MediaScannerConnection.OnScanCompletedListener() {
            public void onScanCompleted(String path, Uri uri) {
                Log.i("ExternalStorage", "Scanned " + path + ":");
                Log.i("ExternalStorage", "-> uri=" + uri);
            }
        });
    } catch (IOException e) {
        // Unable to create file, likely because external storage is
        // not currently mounted.
        Log.w("ExternalStorage", "Error writing " + file, e);
    }
}

void deleteExternalStoragePrivatePicture() {
    // Create a path where we will place our picture in the user's
    // public pictures directory and delete the file.  If external
    // storage is not currently mounted this will fail.
    File path = getExternalFilesDir(Environment.DIRECTORY_PICTURES);
    if (path != null) {
        File file = new File(path, "DemoPicture.jpg");
        file.delete();
    }
}

boolean hasExternalStoragePrivatePicture() {
    // Create a path where we will place our picture in the user's
    // public pictures directory and check if the file exists.  If
    // external storage is not currently mounted this will think the
    // picture doesn't exist.
    File path = getExternalFilesDir(Environment.DIRECTORY_PICTURES);
    if (path != null) {
        File file = new File(path, "DemoPicture.jpg");
        return file.exists();
    }
    return false;
}
```

Parameters	
type	String: The type of files directory to return. May be null for the root of the files directory or one of the following constants for a subdirectory: DIRECTORY_MUSIC (https://developer.android.com/reference/android/os/Environment.html#DIRECTORY_MUSIC), DIRECTORY_PODCASTS (https://developer.android.com/reference/android/os/Environment.html#DIRECTORY_PODCASTS), DIRECTORY_RINGTONES (https://developer.android.com/reference/android/os/Environment.html#DIRECTORY_RINGTONES), DIRECTORY_ALARMS (https://developer.android.com/reference/android/os/Environment.html#DIRECTORY_ALARMS), DIRECTORY_NOTIFICATIONS (https://developer.android.com/reference/android/os/Environment.html#DIRECTORY_NOTIFICATIONS), DIRECTORY_PICTURES (https://developer.android.com/reference/android/os/Environment.html#DIRECTORY_PICTURES), or DIRECTORY_MOVIES (https://developer.android.com/reference/android/os/Environment.html#DIRECTORY_MOVIES).

Returns	
File (https://developer.android.com/reference/java/io/File.html)	the absolute path to application-specific directory. May return null if shared storage is not currently available.

See also:

getFilesDir() (https://developer.android.com/reference/android/content/Context.html#getFilesDir())

getExternalFilesDirs(String) (https://developer.android.com/reference/android/content/Context.html#getExternalFilesDirs(java.lang.String))

getExternalStorageState(File) (https://developer.android.com/reference/android/os/Environment.html#getExternalStorageState(java.io.File))

isExternalStorageEmulated(File) (https://developer.android.com/reference/android/os/Environment.html#isExternalStorageEmulated(java.io.File))

isExternalStorageRemovable(File) (https://developer.android.com/reference/android/os/Environment.html#isExternalStorageRemovable(java.io.File))

getExternalFilesDirs

added in API level 19 (https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

File[] (https://developer.android.com/reference/java/io/File.html) getExternalFilesDirs (String (https://deve

Returns absolute paths to application-specific directories on all shared/external storage devices where the application can place persistent files it owns. These files are internal to the application, and not typically visible to the user as media.

This is like getFilesDir() (https://developer.android.com/reference/android/content/Context.html#getFilesDir()) in that these files will be deleted when the application is uninstalled, however there are some important differences:

- Shared storage may not always be available, since removable media can be ejected by the user. Media state can be checked using getExternalStorageState(File) (https://developer.android.com/reference/android/os/Environment.html#getExternalStorageState(java.io.File)).
- There is no security enforced with these files. For example, any application holding WRITE_EXTERNAL_STORAGE (https://developer.android.com/reference/android/Manifest.permission.html#WRITE_EXTERNAL_STORAGE) can write to these files.

If a shared storage device is emulated (as determined by isExternalStorageEmulated(File) (https://developer.android.com/reference/android/os/Environment.html#isExternalStorageEmulated(java.io.File))), it's contents are backed by a private user data partition, which means there is little benefit to storing data here instead of the private directories returned by getFilesDir() (https://developer.android.com/reference/android/content/Context.html#getFilesDir()), etc.

Shared storage devices returned here are considered a stable part of the device, including physical media slots under a protective cover. The returned paths do not include transient devices, such as USB flash drives connected to handheld devices.

An application may store data on any or all of the returned devices. For example, an app may choose to store large files on the device with the most available space, as measured by StatFs (https://developer.android.com/reference/android/os/StatFs.html).

No additional permissions are required for the calling app to read or write files under the returned path. Write access outside of these paths on secondary external storage devices is not available.

The returned path may change over time if different shared storage media is inserted, so only relative paths should be persisted.

Parameters	
type	String: The type of files directory to return. May be null for the root of the files directory or one of the following constants for a subdirectory: DIRECTORY_MUSIC (https://developer.android.com/reference/android/os/Environment.html#DIRECTORY_MUSIC), DIRECTORY_PODCASTS (https://developer.android.com/reference/android/os/Environment.html#DIRECTORY_PODCASTS), DIRECTORY_RINGTONES

(https://developer.android.com/reference/android/os/Environment.html#DIRECTORY_RINGTONES), DIRECTORY_ALARMS (https://developer.android.com/reference/android/os/Environment.html#DIRECTORY_ALARMS), DIRECTORY_NOTIFICATIONS (https://developer.android.com/reference/android/os/Environment.html#DIRECTORY_NOTIFICATIONS), DIRECTORY_PICTURES (https://developer.android.com/reference/android/os/Environment.html#DIRECTORY_PICTURES), or DIRECTORY_MOVIES (https://developer.android.com/reference/android/os/Environment.html#DIRECTORY_MOVIES).

Returns	
File[] (https://developer.android.com/reference/java/io/File.html)	the absolute paths to application-specific directories. Some individual paths may be null if that shared storage is not currently available. The first path returned is the same as getExternalFilesDir(String) (https://developer.android.com/reference/android/content/Context.html#getExternalFilesDir(java.lang.String)).

See also:

getExternalFilesDir(String) (https://developer.android.com/reference/android/content/Context.html#getExternalFilesDir(java.lang.String))

getExternalStorageState(File) (https://developer.android.com/reference/android/os/Environment.html#getExternalStorageState(java.io.File))

isExternalStorageEmulated(File) (https://developer.android.com/reference/android/os/Environment.html#isExternalStorageEmulated(java.io.File))

isExternalStorageRemovable(File) (https://developer.android.com/reference/android/os/Environment.html#isExternalStorageRemovable(java.io.File))

getExternalMediaDirs

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

```
File[] (https://developer.android.com/reference/java/io/File.html) getExternalMediaDirs ()
```

Returns absolute paths to application-specific directories on all shared/external storage devices where the application can place media files. These files are scanned and made available to other apps through MediaStore (https://developer.android.com/reference/android/provider/MediaStore.html).

This is like getExternalFilesDirs(String) (https://developer.android.com/reference/android/content/Context.html#getExternalFilesDirs(java.lang.String)) in that these files will be deleted when the application is uninstalled, however there are some important differences:

- Shared storage may not always be available, since removable media can be ejected by the user. Media state can be checked using `getExternalStorageState(File)` (https://developer.android.com/reference/android/os/Environment.html#getExternalStorageState(java.io.File)).
- There is no security enforced with these files. For example, any application holding `WRITE_EXTERNAL_STORAGE` (https://developer.android.com/reference/android/Manifest.permission.html#WRITE_EXTERNAL_STORAGE) can write to these files.

Shared storage devices returned here are considered a stable part of the device, including physical media slots under a protective cover. The returned paths do not include transient devices, such as USB flash drives connected to handheld devices.

An application may store data on any or all of the returned devices. For example, an app may choose to store large files on the device with the most available space, as measured by `StatFs` (https://developer.android.com/reference/android/os/StatFs.html).

No additional permissions are required for the calling app to read or write files under the returned path. Write access outside of these paths on secondary external storage devices is not available.

The returned paths may change over time if different shared storage media is inserted, so only relative paths should be persisted.

Returns	
`File[]` (https://developer.android.com/reference/java/io/File.html)	the absolute paths to application-specific directories. Some individual paths may be `null` if that shared storage is not currently available.

See also:

`getExternalStorageState(File)` (https://developer.android.com/reference/android/os/Environment.html#getExternalStorageState(java.io.File))

`isExternalStorageEmulated(File)` (https://developer.android.com/reference/android/os/Environment.html#isExternalStorageEmulated(java.io.File))

`isExternalStorageRemovable(File)` (https://developer.android.com/reference/android/os/Environment.html#isExternalStorageRemovable(java.io.File))

getFileStreamPath

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

```
File (https://developer.android.com/reference/java/io/File.html) getFileStreamPath (String (https://developer.a
```

Returns the absolute path on the filesystem where a file created with openFileOutput(String, int) (https://developer.android.com/reference/android/content/Context.html#openFileOutput(java.lang.String, int)) is stored.

The returned path may change over time if the calling app is moved to an adopted storage device, so only relative paths should be persisted.

Parameters	
name	String: The name of the file for which you would like to get its path.

Returns	
File (https://developer.android.com/reference/java/io/File.html)	An absolute path to the given file.

See also:

openFileOutput(String, int) (https://developer.android.com/reference/android/content/Context.html#openFileOutput(java.lang.String, int))

getFilesDir() (https://developer.android.com/reference/android/content/Context.html#getFilesDir())

getDir(String, int) (https://developer.android.com/reference/android/content/Context.html#getDir(java.lang.String, int))

getFilesDir

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

```
File (https://developer.android.com/reference/java/io/File.html) getFilesDir ()
```

Returns the absolute path to the directory on the filesystem where files created with openFileOutput(String, int) (https://developer.android.com/reference/android/content/Context.html#openFileOutput(java.lang.String, int)) are stored.

The returned path may change over time if the calling app is moved to an adopted storage device, so only relative paths should be persisted.

No additional permissions are required for the calling app to read or write files under the returned path.

Returns	
File (https://developer.android.com/reference/java/io/File.html)	The path of the directory holding application files.

See also:

`openFileOutput(String, int)` (https://developer.android.com/reference/android/content/Context.html#openFileOutput(java.lang.String, int))

`getFileStreamPath(String)` (https://developer.android.com/reference/android/content/Context.html#getFileStreamPath(java.lang.String))

`getDir(String, int)` (https://developer.android.com/reference/android/content/Context.html#getDir(java.lang.String, int))

getMainLooper

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

`Looper` (https://developer.android.com/reference/android/os/Looper.html) `getMainLooper ()`

Return the Looper for the main thread of the current process. This is the thread used to dispatch calls to application components (activities, services, etc).

By definition, this method returns the same result as would be obtained by calling `Looper.getMainLooper()` (https://developer.android.com/reference/android/os/Looper.html#getMainLooper()).

Returns	
`Looper` (https://developer.android.com/reference/android/os/Looper.html)	The main looper.

getNoBackupFilesDir

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

`File` (https://developer.android.com/reference/java/io/File.html) `getNoBackupFilesDir ()`

Returns the absolute path to the directory on the filesystem similar to `getFilesDir()` (https://developer.android.com/reference/android/content/Context.html#getFilesDir()). The difference is that files placed under this directory will be excluded from automatic backup to remote storage. See `BackupAgent` (https://developer.android.com/reference/android/app/backup/BackupAgent.html) for a full discussion of the automatic backup mechanism in Android.

The returned path may change over time if the calling app is moved to an adopted storage device, so only relative paths should be persisted.

No additional permissions are required for the calling app to read or write files under the returned path.

Returns	
`File` (https://developer.android.com/reference/java/io/File.html)	The path of the directory holding application files that will not be automatically backed up to remote storage.

See also:

`openFileOutput(String, int)` (https://developer.android.com/reference/android/content/Context.html#openFileOutput(java.lang.String, int))

`getFileStreamPath(String)` (https://developer.android.com/reference/android/content/Context.html#getFileStreamPath(java.lang.String))

`getDir(String, int)` (https://developer.android.com/reference/android/content/Context.html#getDir(java.lang.String, int))

`BackupAgent` (https://developer.android.com/reference/android/app/backup/BackupAgent.html)

getObbDir

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

`File` (https://developer.android.com/reference/java/io/File.html) `getObbDir ()`

Return the primary shared/external storage directory where this application's OBB files (if there are any) can be found. Note if the application does not have any OBB files, this directory may not exist.

This is like `getFilesDir()` (https://developer.android.com/reference/android/content/Context.html#getFilesDir()) in that these files will be deleted when the application is uninstalled, however there are some important differences:

- Shared storage may not always be available, since removable media can be ejected by the user. Media state can be checked using `getExternalStorageState(File)` (https://developer.android.com/reference/android/os/Environment.html#getExternalStorageState(java.io.File)).
- There is no security enforced with these files. For example, any application holding `WRITE_EXTERNAL_STORAGE` (https://developer.android.com/reference/android/Manifest.permission.html#WRITE_EXTERNAL_STORAGE) can write to these files.

Starting in `KITKAT` (https://developer.android.com/reference/android/os/Build.VERSION_CODES.html#KITKAT), no permissions are required to read or write to the path that this method returns. However, starting from `M` (https://developer.android.com/reference/android/os/Build.VERSION_CODES.html#M), to read the OBB expansion files, you must declare the `READ_EXTERNAL_STORAGE` (https://developer.android.com/reference/android

/Manifest.permission.html#READ_EXTERNAL_STORAGE) permission in the app manifest and ask for permission at runtime as follows:

```
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"
android:maxSdkVersion="23" />
```

Starting from N (https://developer.android.com/reference/android/os/Build.VERSION_CODES.html#N), READ_EXTERNAL_STORAGE (https://developer.android.com/reference/android/Manifest.permission.html#READ_EXTERNAL_STORAGE) permission is not required, so don't ask for this permission at runtime. To handle both cases, your app must first try to read the OBB file, and if it fails, you must request READ_EXTERNAL_STORAGE (https://developer.android.com/reference/android/Manifest.permission.html#READ_EXTERNAL_STORAGE) permission at runtime.

The following code snippet shows how to do this:

```
File obb = new File(obb_filename);
boolean open_failed = false;

try {
    BufferedReader br = new BufferedReader(new FileReader(obb));
    open_failed = false;
    ReadObbFile(br);
} catch (IOException e) {
    open_failed = true;
}

if (open_failed) {
    // request READ_EXTERNAL_STORAGE permission before reading OBB file
    ReadObbFileWithPermission();
}
```

On devices with multiple users (as described by UserManager (https://developer.android.com/reference/android/os/UserManager.html)), multiple users may share the same OBB storage location. Applications should ensure that multiple instances running under different users don't interfere with each other.

Returns	
File (https://developer.android.com/reference/java/io/File.html)	the absolute path to application-specific directory. May return null if shared storage is not currently available.

See also:

getObbDirs() (https://developer.android.com/reference/android/content/Context.html#getObbDirs())

getExternalStorageState(File) (https://developer.android.com/reference/android

/os/Environment.html#getExternalStorageState(java.io.File))

isExternalStorageEmulated(File) (https://developer.android.com/reference/android/os/Environment.html#isExternalStorageEmulated(java.io.File))

isExternalStorageRemovable(File) (https://developer.android.com/reference/android/os/Environment.html#isExternalStorageRemovable(java.io.File))

getObbDirs

added in API level 19 (https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

File[] (https://developer.android.com/reference/java/io/File.html) getObbDirs ()

Returns absolute paths to application-specific directories on all shared/external storage devices where the application's OBB files (if there are any) can be found. Note if the application does not have any OBB files, these directories may not exist.

This is like getFilesDir() (https://developer.android.com/reference/android/content/Context.html#getFilesDir()) in that these files will be deleted when the application is uninstalled, however there are some important differences:

- Shared storage may not always be available, since removable media can be ejected by the user. Media state can be checked using getExternalStorageState(File) (https://developer.android.com/reference/android/os/Environment.html#getExternalStorageState(java.io.File)).
- There is no security enforced with these files. For example, any application holding WRITE_EXTERNAL_STORAGE (https://developer.android.com/reference/android/Manifest.permission.html#WRITE_EXTERNAL_STORAGE) can write to these files.

Shared storage devices returned here are considered a stable part of the device, including physical media slots under a protective cover. The returned paths do not include transient devices, such as USB flash drives connected to handheld devices.

An application may store data on any or all of the returned devices. For example, an app may choose to store large files on the device with the most available space, as measured by StatFs (https://developer.android.com/reference/android/os/StatFs.html).

No additional permissions are required for the calling app to read or write files under the returned path. Write access outside of these paths on secondary external storage devices is not available.

Returns	
File[] (https://developer.android.com/reference/java/io/File.html)	the absolute paths to application-specific directories. Some individual paths may be null if that shared storage is not currently available. The first path returned is the same as getObbDir() (https://developer.android.com/reference

/android/content/Context.html#getObbDir())

See also:

getObbDir() (https://developer.android.com/reference/android/content/Context.html#getObbDir())

getExternalStorageState(File) (https://developer.android.com/reference/android/os/Environment.html#getExternalStorageState(java.io.File))

isExternalStorageEmulated(File) (https://developer.android.com/reference/android/os/Environment.html#isExternalStorageEmulated(java.io.File))

isExternalStorageRemovable(File) (https://developer.android.com/reference/android/os/Environment.html#isExternalStorageRemovable(java.io.File))

getPackageCodePath

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

```
String (https://developer.android.com/reference/java/lang/String.html) getPackageCodePath ()
```

Return the full path to this context's primary Android package. The Android package is a ZIP file which contains application's primary code and assets.

Note: this is not generally useful for applications, since they should not be directly accessing the file system.

Returns	
String (https://developer.android.com/reference/java/lang/String.html)	String Path to the code and assets.

getPackageManager

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

```
PackageManager (https://developer.android.com/reference/android/content/pm/PackageManager.html) getPackageManag
```

Return PackageManager instance to find global package information.

Returns	
PackageManager (https://developer.android.com	

/reference/android/content
/pm/PackageManager.html)

getPackageName

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

`String (https://developer.android.com/reference/java/lang/String.html) getPackageName ()`

Return the name of this application's package.

Returns	
String (https://developer.android.com/reference/java/lang/String.html)	

getPackageResourcePath

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

`String (https://developer.android.com/reference/java/lang/String.html) getPackageResourcePath ()`

Return the full path to this context's primary Android package. The Android package is a ZIP file which contains the application's primary resources.

Note: this is not generally useful for applications, since they should not be directly accessing the file system.

Returns	
String (https://developer.android.com/reference/java/lang/String.html)	String Path to the resources.

getResources

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

`Resources (https://developer.android.com/reference/android/content/res/Resources.html) getResources ()`

Returns a Resources instance for the application's package.

Note: Implementations of this method should return a Resources instance that is consistent with the AssetManager instance returned by `getAssets()` (https://developer.android.com/reference/android/content/Context.html#getAssets()). For example, they should share the same `Configuration` (https://developer.android.com/reference/android/content/res/Configuration.html) object.

Returns	
`Resources` (https://developer.android.com/reference/android/content/res/Resources.html)	a Resources instance for the application's package

See also:

`getAssets()` (https://developer.android.com/reference/android/content/Context.html#getAssets())

getSharedPreferences

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

```
SharedPreferences (https://developer.android.com/reference/android/content/SharedPreferences.html) getSharedPre
                    int mode)
```

Retrieve and hold the contents of the preferences file 'name', returning a SharedPreferences through which you can retrieve and modify its values. Only one instance of the SharedPreferences object is returned to any callers for the same name, meaning they will see each other's edits as soon as they are made.

Parameters	
`name`	`String`: Desired preferences file. If a preferences file by this name does not exist, it will be created when you retrieve an editor (SharedPreferences.edit()) and then commit changes (Editor.commit()).
`mode`	`int`: Operating mode. Value is either `0` or combination of `MODE_PRIVATE` (https://developer.android.com/reference/android/content/Context.html#MODE_PRIVATE), `MODE_WORLD_READABLE` (https://developer.android.com/reference/android/content/Context.html#MODE_WORLD_READABLE), `MODE_WORLD_WRITEABLE` (https://developer.android.com/reference/android/content/Context.html#MODE_WORLD_WRITEABLE) or `MODE_MULTI_PROCESS` (https://developer.android.com/reference/android/content/Context.html#MODE_MULTI_PROCESS).

Returns	
`SharedPreferences` (https://developer.android.com	The single `SharedPreferences` (https://developer.android.com/reference/android/content/SharedPreferences.html) instance that can be used to retrieve

/reference/android/content /SharedPreferences.html)	and modify the preference values.

See also:

MODE_PRIVATE (https://developer.android.com/reference/android/content/Context.html#MODE_PRIVATE)

getString

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

```
String (https://developer.android.com/reference/java/lang/String.html) getString (int resId,
                Object... (https://developer.android.com/reference/java/lang/Object.html) formatArgs)
```

Returns a localized formatted string from the application's package's default string table, substituting the format arguments as defined in Formatter (https://developer.android.com/reference/java/util/Formatter.html) and format(String, Object...) (https://developer.android.com/reference/java/lang/String.html#format(java.lang.String, java.lang.Object...)).

Parameters	
resId	int: Resource id for the format string
formatArgs	Object: The format arguments that will be used for substitution.

Returns	
String (https://developer.android.com/reference/java/lang/String.html)	The string data associated with the resource, formatted and stripped of styled text information. This value will never be null.

getString

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

```
String (https://developer.android.com/reference/java/lang/String.html) getString (int resId)
```

Returns a localized string from the application's package's default string table.

Parameters	
resId	int: Resource id for the string

Returns	
String (https://developer.android.com/reference/java/lang/String.html)	The string data associated with the resource, stripped of styled text information. This value will never be null.

getSystemService

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

```
T getSystemService (Class (https://developer.android.com/reference/java/lang/Class.html)<T> serviceClass)
```

Return the handle to a system-level service by class.

Currently available classes are: WindowManager (https://developer.android.com/reference/android/view/WindowManager.html), LayoutInflater (https://developer.android.com/reference/android/view/LayoutInflater.html), ActivityManager (https://developer.android.com/reference/android/app/ActivityManager.html), PowerManager (https://developer.android.com/reference/android/os/PowerManager.html), AlarmManager (https://developer.android.com/reference/android/app/AlarmManager.html), NotificationManager (https://developer.android.com/reference/android/app/NotificationManager.html), KeyguardManager (https://developer.android.com/reference/android/app/KeyguardManager.html), LocationManager (https://developer.android.com/reference/android/location/LocationManager.html), SearchManager (https://developer.android.com/reference/android/app/SearchManager.html), Vibrator (https://developer.android.com/reference/android/os/Vibrator.html), ConnectivityManager (https://developer.android.com/reference/android/net/ConnectivityManager.html), WifiManager (https://developer.android.com/reference/android/net/wifi/WifiManager.html), AudioManager (https://developer.android.com/reference/android/media/AudioManager.html), MediaRouter (https://developer.android.com/reference/android/media/MediaRouter.html), TelephonyManager (https://developer.android.com/reference/android/telephony/TelephonyManager.html), SubscriptionManager (https://developer.android.com/reference/android/telephony/SubscriptionManager.html), InputMethodManager (https://developer.android.com/reference/android/view/inputmethod/InputMethodManager.html), UiModeManager (https://developer.android.com/reference/android/app/UiModeManager.html), DownloadManager (https://developer.android.com/reference/android/app/DownloadManager.html), BatteryManager (https://developer.android.com/reference/android/os/BatteryManager.html), JobScheduler (https://developer.android.com/reference/android/app/job/JobScheduler.html), NetworkStatsManager (https://developer.android.com/reference/android/app/usage/NetworkStatsManager.html).

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

Parameters	
serviceClass	Class: The class of the desired service. This value must never be null.

Returns	
T	The service or null if the class is not a supported system service.

getSystemService

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

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

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 (https://developer.android.com/reference/android/content/Context.html#WINDOW_SERVICE) ("window")

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

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

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

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

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

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

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

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

> A AlarmManager (https://developer.android.com/reference/android/app/AlarmManager.html) for receiving intents

at the time of your choosing.

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

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

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

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

LOCATION_SERVICE (https://developer.android.com/reference/android/content/Context.html#LOCATION_SERVICE) ("location")

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

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

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

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

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

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

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

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

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

WIFI_AWARE_SERVICE (https://developer.android.com/reference/android/content/Context.html#WIFI_AWARE_SERVICE) ("wifiaware")

A WifiAwareManager (https://developer.android.com/reference/android/net/wifi/aware/WifiAwareManager.html) for management of Wi-Fi Aware discovery and connectivity.

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

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

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

An InputMethodManager (https://developer.android.com/reference/android/view/inputmethod/InputMethodManager.html) for management of input methods.

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

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

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

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

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

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

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

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

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

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

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

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

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

Parameters	
name	String: The name of the desired service. This value must never be null.

Returns	
Object (https://developer.android.com/reference/java/lang/Object.html)	The service or null if the name does not exist.

See also:

WINDOW_SERVICE (https://developer.android.com/reference/android/content/Context.html#WINDOW_SERVICE)

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

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

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

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

ActivityManager (https://developer.android.com/reference/android/app/ActivityManager.html)

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

PowerManager (https://developer.android.com/reference/android/os/PowerManager.html)

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

AlarmManager (https://developer.android.com/reference/android/app/AlarmManager.html)

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

NotificationManager (https://developer.android.com/reference/android/app/NotificationManager.html)

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

KeyguardManager (https://developer.android.com/reference/android/app/KeyguardManager.html)

LOCATION_SERVICE (https://developer.android.com/reference/android/content/Context.html#LOCATION_SERVICE)

LocationManager (https://developer.android.com/reference/android/location/LocationManager.html)

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

SearchManager (https://developer.android.com/reference/android/app/SearchManager.html)

SENSOR_SERVICE (https://developer.android.com/reference/android/content/Context.html#SENSOR_SERVICE)

SensorManager (https://developer.android.com/reference/android/hardware/SensorManager.html)

STORAGE_SERVICE (https://developer.android.com/reference/android/content/Context.html#STORAGE_SERVICE)

StorageManager (https://developer.android.com/reference/android/os/storage/StorageManager.html)

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

Vibrator (https://developer.android.com/reference/android/os/Vibrator.html)

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

ConnectivityManager (https://developer.android.com/reference/android/net/ConnectivityManager.html)

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

WifiManager (https://developer.android.com/reference/android/net/wifi/WifiManager.html)

AUDIO_SERVICE (https://developer.android.com/reference/android/content/Context.html#AUDIO_SERVICE)

AudioManager (https://developer.android.com/reference/android/media/AudioManager.html)

MEDIA_ROUTER_SERVICE (https://developer.android.com/reference/android/content/Context.html#MEDIA_ROUTER_SERVICE)

MediaRouter (https://developer.android.com/reference/android/media/MediaRouter.html)

TELEPHONY_SERVICE (https://developer.android.com/reference/android/content/Context.html#TELEPHONY_SERVICE)

TelephonyManager (https://developer.android.com/reference/android/telephony/TelephonyManager.html)

TELEPHONY_SUBSCRIPTION_SERVICE (https://developer.android.com/reference/android/content/Context.html#TELEPHONY_SUBSCRIPTION_SERVICE)

SubscriptionManager (https://developer.android.com/reference/android/telephony/SubscriptionManager.html)

CARRIER_CONFIG_SERVICE (https://developer.android.com/reference/android/content/Context.html#CARRIER_CONFIG_SERVICE)

CarrierConfigManager (https://developer.android.com/reference/android/telephony/CarrierConfigManager.html)

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

InputMethodManager (https://developer.android.com/reference/android/view/inputmethod/InputMethodManager.html)

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

UiModeManager (https://developer.android.com/reference/android/app/UiModeManager.html)

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

DownloadManager (https://developer.android.com/reference/android/app/DownloadManager.html)

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

BatteryManager (https://developer.android.com/reference/android/os/BatteryManager.html)

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

JobScheduler (https://developer.android.com/reference/android/app/job/JobScheduler.html)

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

NetworkStatsManager (https://developer.android.com/reference/android/app/usage/NetworkStatsManager.html)

HardwarePropertiesManager (https://developer.android.com/reference/android/os/HardwarePropertiesManager.html)

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

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

```
String (https://developer.android.com/reference/java/lang/String.html) getSystemServiceName (Class (https://de
```

Gets the name of the system-level service that is represented by the specified class.

Parameters	
`serviceClass`	`Class`: The class of the desired service. This value must never be `null`.

Returns	
`String` (https://developer.android.com/reference/java/lang/String.html)	The service name or null if the class is not a supported system service.

getText

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

`CharSequence (https://developer.android.com/reference/java/lang/CharSequence.html) getText (int resId)`

Return a localized, styled CharSequence from the application's package's default string table.

Parameters	
`resId`	`int`: Resource id for the CharSequence text

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

getTheme

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

`Resources.Theme (https://developer.android.com/reference/android/content/res/Resources.Theme.html) getTheme ()`

Return the Theme object associated with this Context.

Returns	
`Resources.Theme` (https://developer.android.com/reference/android/content/res/Resources.Theme.html)	

getWallpaper

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

`Drawable (https://developer.android.com/reference/android/graphics/drawable/Drawable.html) getWallpaper ()`

> **This method was deprecated in API level 5.**
> Use `WallpaperManager.get() (https://developer.android.com/reference/android/app/WallpaperManager.html#getDrawable())` instead.

Returns	
`Drawable (https://developer.android.com/reference/android/graphics/drawable/Drawable.html)`	

getWallpaperDesiredMinimumHeight

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

`int getWallpaperDesiredMinimumHeight ()`

> **This method was deprecated in API level 5.**
> Use `WallpaperManager.getDesiredMinimumHeight() (https://developer.android.com/reference/android/app/WallpaperManager.html#getDesiredMinimumHeight())` instead.

Returns	
`int`	

getWallpaperDesiredMinimumWidth

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

`int getWallpaperDesiredMinimumWidth ()`

> **This method was deprecated in API level 5.**
> Use `WallpaperManager.getDesiredMinimumWidth() (https://developer.android.com/reference/android/app/WallpaperManager.html#getDesiredMinimumWidth())` instead.

Returns	
`int`	

grantUriPermission

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

```
void grantUriPermission (String (https://developer.android.com/reference/java/lang/String.html) toPackage,
                Uri (https://developer.android.com/reference/android/net/Uri.html) uri,
                int modeFlags)
```

Grant permission to access a specific Uri to another package, regardless of whether that package has general permission to access the Uri's content provider. This can be used to grant specific, temporary permissions, typically in response to user interaction (such as the user opening an attachment that you would like someone else to display).

Normally you should use `Intent.FLAG_GRANT_READ_URI_PERMISSION` (https://developer.android.com/reference/android/content/Intent.html#FLAG_GRANT_READ_URI_PERMISSION) or `Intent.FLAG_GRANT_WRITE_URI_PERMISSION` (https://developer.android.com/reference/android/content/Intent.html#FLAG_GRANT_WRITE_URI_PERMISSION) with the Intent being used to start an activity instead of this function directly. If you use this function directly, you should be sure to call `revokeUriPermission(Uri, int)` (https://developer.android.com/reference/android/content/Context.html#revokeUriPermission(android.net.Uri, int)) when the target should no longer be allowed to access it.

To succeed, the content provider owning the Uri must have set the `grantUriPermissions` (https://developer.android.com/reference/android/R.styleable.html#AndroidManifestProvider_grantUriPermissions) attribute in its manifest or included the `<grant-uri-permissions>` (https://developer.android.com/reference/android/R.styleable.html#AndroidManifestGrantUriPermission) tag.

Parameters	
toPackage	String: The package you would like to allow to access the Uri.
uri	Uri: The Uri you would like to grant access to.
modeFlags	int: The desired access modes. Value is either 0 or combination of FLAG_GRANT_READ_URI_PERMISSION (https://developer.android.com/reference/android/content/Intent.html#FLAG_GRANT_READ_URI_PERMISSION), FLAG_GRANT_WRITE_URI_PERMISSION (https://developer.android.com/reference/android/content/Intent.html#FLAG_GRANT_WRITE_URI_PERMISSION), FLAG_GRANT_PERSISTABLE_URI_PERMISSION (https://developer.android.com/reference/android/content/Intent.html#FLAG_GRANT_PERSISTABLE_URI_PERMISSION) or FLAG_GRANT_PREFIX_URI_PERMISSION (https://developer.android.com/reference/android/content/Intent.html#FLAG_GRANT_PREFIX_URI_PERMISSION).

See also:

`revokeUriPermission(Uri, int)` (https://developer.android.com/reference/android/content/Context.html#revokeUriPermission(android.net.Uri, int))

isDeviceProtectedStorage

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

```
boolean isDeviceProtectedStorage ()
```

Indicates if the storage APIs of this Context are backed by device-protected storage.

Returns	
boolean	

See also:

createDeviceProtectedStorageContext() (https://developer.android.com/reference/android/content/Context.html#createDeviceProtectedStorageContext())

isRestricted

added in API level 4 (https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

```
boolean isRestricted ()
```

Indicates whether this Context is restricted.

Returns	
boolean	true if this Context is restricted, false otherwise.

See also:

CONTEXT_RESTRICTED (https://developer.android.com/reference/android/content/Context.html#CONTEXT_RESTRICTED)

moveDatabaseFrom

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

```
boolean moveDatabaseFrom (Context (https://developer.android.com/reference/android/content/Context.html) sour
                String (https://developer.android.com/reference/java/lang/String.html) name)
```

Move an existing database file from the given source storage context to this context. This is typically used to migrate data between storage locations after an upgrade, such as migrating to device protected storage.

The database must be closed before being moved.

Parameters	
sourceContext	Context: The source context which contains the existing database to move.

name	String: The name of the database file.

Returns	
boolean	true if the move was successful or if the database didn't exist in the source context, otherwise false.

See also:

createDeviceProtectedStorageContext() (https://developer.android.com/reference/android/content/Context.html#createDeviceProtectedStorageContext())

moveSharedPreferencesFrom

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

```
boolean moveSharedPreferencesFrom (Context (https://developer.android.com/reference/android/content/Context
                String (https://developer.android.com/reference/java/lang/String.html) name)
```

Move an existing shared preferences file from the given source storage context to this context. This is typically used to migrate data between storage locations after an upgrade, such as moving to device protected storage.

Parameters	
sourceContext	Context: The source context which contains the existing shared preferences to move.
name	String: The name of the shared preferences file.

Returns	
boolean	true if the move was successful or if the shared preferences didn't exist in the source context, otherwise false.

See also:

createDeviceProtectedStorageContext() (https://developer.android.com/reference/android/content/Context.html#createDeviceProtectedStorageContext())

obtainStyledAttributes

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

```
TypedArray (https://developer.android.com/reference/android/content/res/TypedArray.html) obtainStyledAttribute
                int[] attrs)
```

Retrieve styled attribute information in this Context's theme. See `obtainStyledAttributes(AttributeSet, int[], int, int)` (https://developer.android.com/reference/android/content/res/Resources.Theme.html#obtainStyledAttributes(android.util.AttributeSet, int[], int, int)) for more information.

Parameters	
set	AttributeSet
attrs	int

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

See also:

`obtainStyledAttributes(AttributeSet, int[], int, int)` (https://developer.android.com/reference/android/content/res/Resources.Theme.html#obtainStyledAttributes(android.util.AttributeSet, int[], int, int))

obtainStyledAttributes

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

```
TypedArray (https://developer.android.com/reference/android/content/res/TypedArray.html) obtainStyledAttribute
                int[] attrs,
                int defStyleAttr,
                int defStyleRes)
```

Retrieve styled attribute information in this Context's theme. See `obtainStyledAttributes(AttributeSet, int[], int, int)` (https://developer.android.com/reference/android/content/res/Resources.Theme.html#obtainStyledAttributes(android.util.AttributeSet, int[], int, int)) for more information.

Parameters	
set	AttributeSet
attrs	int
defStyleAttr	int
defStyleRes	int

Returns

TypedArray (https://developer.android.com/reference/android/content/res/TypedArray.html)

See also:

obtainStyledAttributes(AttributeSet, int[], int, int) (https://developer.android.com/reference/android/content/res/Resources.Theme.html#obtainStyledAttributes(android.util.AttributeSet, int[], int, int))

obtainStyledAttributes

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

```
TypedArray (https://developer.android.com/reference/android/content/res/TypedArray.html) obtainStyledAttribute
                int[] attrs)
```

Retrieve styled attribute information in this Context's theme. See obtainStyledAttributes(int, int[]) (https://developer.android.com/reference/android/content/res/Resources.Theme.html#obtainStyledAttributes(int, int[])) for more information.

Parameters	
resid	int
attrs	int

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

Throws	
Resources.NotFoundException (https://developer.android.com/reference/android/content/res/Resources.NotFoundException.html)	

See also:

obtainStyledAttributes(int, int[]) (https://developer.android.com/reference/android/content

/res/Resources.Theme.html#obtainStyledAttributes(int, int[]))

obtainStyledAttributes

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

`TypedArray (https://developer.android.com/reference/android/content/res/TypedArray.html) obtainStyledAttribute`

Retrieve styled attribute information in this Context's theme. See `obtainStyledAttributes(int[])` (https://developer.android.com/reference/android/content/res/Resources.Theme.html#obtainStyledAttributes(int[])) for more information.

Parameters	
attrs	int

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

See also:

`obtainStyledAttributes(int[])` (https://developer.android.com/reference/android/content/res/Resources.Theme.html#obtainStyledAttributes(int[]))

openFileInput

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

`FileInputStream (https://developer.android.com/reference/java/io/FileInputStream.html) openFileInput (String`

Open a private file associated with this Context's application package for reading.

Parameters	
name	String: The name of the file to open; can not contain path separators.

Returns	
FileInputStream (https://developer.android.com	The resulting FileInputStream (https://developer.android.com/reference/java/io/FileInputStream.html).

/reference/java/io

/FileInputStream.html)

Throws	
FileNotFoundException (https://developer.android.com/reference/java/io/FileNotFoundException.html)	

See also:

openFileOutput(String, int) (https://developer.android.com/reference/android/content/Context.html#openFileOutput(java.lang.String, int))

fileList() (https://developer.android.com/reference/android/content/Context.html#fileList())

deleteFile(String) (https://developer.android.com/reference/android/content/Context.html#deleteFile(java.lang.String))

FileInputStream(String) (https://developer.android.com/reference/java/io/FileInputStream.html#FileInputStream(java.lang.String))

openFileOutput

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

```
FileOutputStream (https://developer.android.com/reference/java/io/FileOutputStream.html) openFileOutput (Stri
                    int mode)
```

Open a private file associated with this Context's application package for writing. Creates the file if it doesn't already exist.

No additional permissions are required for the calling app to read or write the returned file.

Parameters	
name	String: The name of the file to open; can not contain path separators.
mode	int: Operating mode. Value is either 0 or combination of MODE_PRIVATE (https://developer.android.com/reference/android/content/Context.html#MODE_PRIVATE), MODE_WORLD_READABLE (https://developer.android.com/reference/android/content/Context.html#MODE_WORLD_READABLE), MODE_WORLD_WRITEABLE (https://developer.android.com/reference/android/content/Context.html#MODE_WORLD_WRITEABLE) or MODE_APPEND (https://developer.android.com/reference/android/content/Context.html#MODE_APPEND).

Returns	
FileOutputStream (https://developer.android.com/reference/java/io/FileOutputStream.html)	The resulting FileOutputStream (https://developer.android.com/reference/java/io/FileOutputStream.html).

Throws	
FileNotFoundException (https://developer.android.com/reference/java/io/FileNotFoundException.html)	

See also:

MODE_APPEND (https://developer.android.com/reference/android/content/Context.html#MODE_APPEND)

MODE_PRIVATE (https://developer.android.com/reference/android/content/Context.html#MODE_PRIVATE)

openFileInput(String) (https://developer.android.com/reference/android/content/Context.html#openFileInput(java.lang.String))

fileList() (https://developer.android.com/reference/android/content/Context.html#fileList())

deleteFile(String) (https://developer.android.com/reference/android/content/Context.html#deleteFile(java.lang.String))

FileOutputStream(String) (https://developer.android.com/reference/java/io/FileOutputStream.html#FileOutputStream(java.lang.String))

openOrCreateDatabase

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

```
SQLiteDatabase (https://developer.android.com/reference/android/database/sqlite/SQLiteDatabase.html) openOrCreat
                int mode,
                SQLiteDatabase.CursorFactory (https://developer.android.com/reference/android/database/sqli
                DatabaseErrorHandler (https://developer.android.com/reference/android/database/DatabaseErrorH
```

Open a new private SQLiteDatabase associated with this Context's application package. Creates the database file if it doesn't exist.

Accepts input param: a concrete instance of DatabaseErrorHandler (https://developer.android.com/reference/android/database/DatabaseErrorHandler.html) to be used to handle corruption when sqlite reports database corruption.

Parameters	
`name`	`String`: The name (unique in the application package) of the database.
`mode`	`int`: Operating mode. Value is either `0` or combination of `MODE_PRIVATE` (https://developer.android.com/reference/android/content/Context.html#MODE_PRIVATE), `MODE_WORLD_READABLE` (https://developer.android.com/reference/android/content/Context.html#MODE_WORLD_READABLE), `MODE_WORLD_WRITEABLE` (https://developer.android.com/reference/android/content/Context.html#MODE_WORLD_WRITEABLE), `MODE_ENABLE_WRITE_AHEAD_LOGGING` (https://developer.android.com/reference/android/content/Context.html#MODE_ENABLE_WRITE_AHEAD_LOGGING) or `MODE_NO_LOCALIZED_COLLATORS` (https://developer.android.com/reference/android/content/Context.html#MODE_NO_LOCALIZED_COLLATORS).
`factory`	`SQLiteDatabase.CursorFactory`: An optional factory class that is called to instantiate a cursor when query is called.
`errorHandler`	`DatabaseErrorHandler`: the `DatabaseErrorHandler` (https://developer.android.com/reference/android/database/DatabaseErrorHandler.html) to be used when sqlite reports database corruption. if null, `DefaultDatabaseErrorHandler` (https://developer.android.com/reference/android/database/DefaultDatabaseErrorHandler.html) is assumed.

Returns	
`SQLiteDatabase` (https://developer.android.com/reference/android/database/sqlite/SQLiteDatabase.html)	The contents of a newly created database with the given name.

Throws	
`SQLiteException` (https://developer.android.com/reference/android/database/sqlite/SQLiteException.html)	if the database file could not be opened.

See also:

`MODE_PRIVATE` (https://developer.android.com/reference/android/content/Context.html#MODE_PRIVATE)

`MODE_ENABLE_WRITE_AHEAD_LOGGING` (https://developer.android.com/reference/android/content/Context.html#MODE_ENABLE_WRITE_AHEAD_LOGGING)

`MODE_NO_LOCALIZED_COLLATORS` (https://developer.android.com/reference/android/content

/Context.html#MODE_NO_LOCALIZED_COLLATORS)

deleteDatabase(String) (https://developer.android.com/reference/android/content/Context.html#deleteDatabase(java.lang.String))

openOrCreateDatabase

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

```
SQLiteDatabase (https://developer.android.com/reference/android/database/sqlite/SQLiteDatabase.html) openOrCreat
                int mode,
                SQLiteDatabase.CursorFactory (https://developer.android.com/reference/android/database/sqli
```

Open a new private SQLiteDatabase associated with this Context's application package. Create the database file if it doesn't exist.

Parameters	
name	String: The name (unique in the application package) of the database.
mode	int: Operating mode. Value is either 0 or combination of MODE_PRIVATE (https://developer.android.com/reference/android/content/Context.html#MODE_PRIVATE), MODE_WORLD_READABLE (https://developer.android.com/reference/android/content/Context.html#MODE_WORLD_READABLE), MODE_WORLD_WRITEABLE (https://developer.android.com/reference/android/content/Context.html#MODE_WORLD_WRITEABLE), MODE_ENABLE_WRITE_AHEAD_LOGGING (https://developer.android.com/reference/android/content/Context.html#MODE_ENABLE_WRITE_AHEAD_LOGGING) or MODE_NO_LOCALIZED_COLLATORS (https://developer.android.com/reference/android/content/Context.html#MODE_NO_LOCALIZED_COLLATORS).
factory	SQLiteDatabase.CursorFactory: An optional factory class that is called to instantiate a cursor when query is called.

Returns	
SQLiteDatabase (https://developer.android.com/reference/android/database/sqlite/SQLiteDatabase.html)	The contents of a newly created database with the given name.

Throws	
SQLiteException (https://developer.android.com/reference/android/database	if the database file could not be opened.

/sqlite/SQLiteException.html)

See also:

MODE_PRIVATE (https://developer.android.com/reference/android/content/Context.html#MODE_PRIVATE)

MODE_ENABLE_WRITE_AHEAD_LOGGING (https://developer.android.com/reference/android/content/Context.html#MODE_ENABLE_WRITE_AHEAD_LOGGING)

MODE_NO_LOCALIZED_COLLATORS (https://developer.android.com/reference/android/content/Context.html#MODE_NO_LOCALIZED_COLLATORS)

deleteDatabase(String) (https://developer.android.com/reference/android/content/Context.html#deleteDatabase(java.lang.String))

peekWallpaper

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

```
Drawable (https://developer.android.com/reference/android/graphics/drawable/Drawable.html) peekWallpaper ()
```

> **This method was deprecated in API level 5.**
> Use `WallpaperManager.peek()` (https://developer.android.com/reference/android/app/WallpaperManager.html#peekDrawable()) instead.

Returns	
Drawable (https://developer.android.com/reference/android/graphics/drawable/Drawable.html)	

registerComponentCallbacks

added in API level 14 (https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

```
void registerComponentCallbacks (ComponentCallbacks (https://developer.android.com/reference/android/cont
```

Add a new `ComponentCallbacks` (https://developer.android.com/reference/android/content/ComponentCallbacks.html) to the base application of the Context, which will be called at the same times as the ComponentCallbacks methods of activities and other components are called. Note that you *must* be sure to use `unregisterComponentCallbacks(ComponentCallbacks)` (https://developer.android.com/reference/android/content/Context.html#unregisterComponentCallbacks(android.content.ComponentCallbacks)) when appropriate in the future; this will not be removed for you.

Parameters	
callback	ComponentCallbacks: The interface to call. This can be either a ComponentCallbacks (https://developer.android.com/reference/android/content/ComponentCallbacks.html) or ComponentCallbacks2 (https://developer.android.com/reference/android/content/ComponentCallbacks2.html) interface.

registerReceiver

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

```
Intent (https://developer.android.com/reference/android/content/Intent.html) registerReceiver (BroadcastRece
                IntentFilter (https://developer.android.com/reference/android/content/IntentFilter.html) filte
```

Register a BroadcastReceiver to be run in the main activity thread. The ***receiver*** will be called with any broadcast Intent that matches ***filter***, in the main application thread.

The system may broadcast Intents that are "sticky" -- these stay around after the broadcast has finished, to be sent to any later registrations. If your IntentFilter matches one of these sticky Intents, that Intent will be returned by this function **and** sent to your ***receiver*** as if it had just been broadcast.

There may be multiple sticky Intents that match ***filter***, in which case each of these will be sent to ***receiver***. In this case, only one of these can be returned directly by the function; which of these that is returned is arbitrarily decided by the system.

If you know the Intent your are registering for is sticky, you can supply null for your ***receiver***. In this case, no receiver is registered -- the function simply returns the sticky Intent that matches ***filter***. In the case of multiple matches, the same rules as described above apply.

See BroadcastReceiver (https://developer.android.com/reference/android/content/BroadcastReceiver.html) for more information on Intent broadcasts.

As of ICE_CREAM_SANDWICH (https://developer.android.com/reference/android/os/Build.VERSION_CODES.html#ICE_CREAM_SANDWICH), receivers registered with this method will correctly respect the setPackage(String) (https://developer.android.com/reference/android/content/Intent.html#setPackage(java.lang.String)) specified for an Intent being broadcast. Prior to that, it would be ignored and delivered to all matching registered receivers. Be careful if using this for security.

> Note: this method *cannot be called from a* BroadcastReceiver (https://developer.android.com/reference/android/content/BroadcastReceiver.html) *component;* that is, from a BroadcastReceiver that is declared in an application's manifest. It is okay, however, to call this method from another BroadcastReceiver that has itself been registered at run time with registerReceiver(BroadcastReceiver, IntentFilter) (https://developer.android.com

> /reference/android/content/Context.html#registerReceiver(android.content.BroadcastReceiver, android.content.IntentFilter)), since the lifetime of such a registered BroadcastReceiver is tied to the object that registered it.

Parameters	
receiver	BroadcastReceiver: The BroadcastReceiver to handle the broadcast. This value may be null.
filter	IntentFilter: Selects the Intent broadcasts to be received.

Returns	
Intent (https://developer.android.com /reference/android/content /Intent.html)	The first sticky intent found that matches *filter*, or null if there are none.

See also:

registerReceiver(BroadcastReceiver, IntentFilter, String, Handler) (https://developer.android.com /reference/android/content/Context.html#registerReceiver(android.content.BroadcastReceiver, android.content.IntentFilter, java.lang.String, android.os.Handler))

sendBroadcast(Intent) (https://developer.android.com/reference/android/content /Context.html#sendBroadcast(android.content.Intent))

unregisterReceiver(BroadcastReceiver) (https://developer.android.com/reference/android/content /Context.html#unregisterReceiver(android.content.BroadcastReceiver))

registerReceiver

added in API level 26 (https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

```
Intent (https://developer.android.com/reference/android/content/Intent.html) registerReceiver (BroadcastRece
                IntentFilter (https://developer.android.com/reference/android/content/IntentFilter.html) filte
                int flags)
```

Register to receive intent broadcasts, with the receiver optionally being exposed to Instant Apps. See registerReceiver(BroadcastReceiver, IntentFilter) (https://developer.android.com/reference/android /content/Context.html#registerReceiver(android.content.BroadcastReceiver, android.content.IntentFilter)) for more information. By default Instant Apps cannot interact with receivers in other applications, this allows you to expose a receiver that Instant Apps can interact with.

See BroadcastReceiver (https://developer.android.com/reference/android/content/BroadcastReceiver.html) for more information on Intent broadcasts.

As of ICE_CREAM_SANDWICH (https://developer.android.com/reference/android/os/Build.VERSION_CODES.html#ICE_CREAM_SANDWICH), receivers registered with this method will correctly respect the setPackage(String) (https://developer.android.com/reference/android/content/Intent.html#setPackage(java.lang.String)) specified for an Intent being broadcast. Prior to that, it would be ignored and delivered to all matching registered receivers. Be careful if using this for security.

Parameters	
receiver	BroadcastReceiver: The BroadcastReceiver to handle the broadcast. This value may be null.
filter	IntentFilter: Selects the Intent broadcasts to be received.
flags	int: Additional options for the receiver. May be 0 or RECEIVER_VISIBLE_TO_INSTANT_APPS (https://developer.android.com/reference/android/content/Context.html#RECEIVER_VISIBLE_TO_INSTANT_APPS).

Returns	
Intent (https://developer.android.com/reference/android/content/Intent.html)	The first sticky intent found that matches *filter*, or null if there are none.

See also:

registerReceiver(BroadcastReceiver, IntentFilter) (https://developer.android.com/reference/android/content/Context.html#registerReceiver(android.content.BroadcastReceiver, android.content.IntentFilter))

sendBroadcast(Intent) (https://developer.android.com/reference/android/content/Context.html#sendBroadcast(android.content.Intent))

unregisterReceiver(BroadcastReceiver) (https://developer.android.com/reference/android/content/Context.html#unregisterReceiver(android.content.BroadcastReceiver))

registerReceiver

added in API level 26 (https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

```
Intent (https://developer.android.com/reference/android/content/Intent.html) registerReceiver (BroadcastRece
                    IntentFilter (https://developer.android.com/reference/android/content/IntentFilter.html) filte
                    String (https://developer.android.com/reference/java/lang/String.html) broadcastPermission,
                    Handler (https://developer.android.com/reference/android/os/Handler.html) scheduler,
                int flags)
```

Register to receive intent broadcasts, to run in the context of *scheduler*. See `registerReceiver(BroadcastReceiver, IntentFilter, int)` (https://developer.android.com/reference/android/content/Context.html#registerReceiver(android.content.BroadcastReceiver, android.content.IntentFilter, int)) and `registerReceiver(BroadcastReceiver, IntentFilter, String, Handler)` (https://developer.android.com/reference/android/content/Context.html#registerReceiver(android.content.BroadcastReceiver, android.content.IntentFilter, java.lang.String, android.os.Handler)) for more information.

See `BroadcastReceiver` (https://developer.android.com/reference/android/content/BroadcastReceiver.html) for more information on Intent broadcasts.

As of `ICE_CREAM_SANDWICH` (https://developer.android.com/reference/android/os/Build.VERSION_CODES.html#ICE_CREAM_SANDWICH), receivers registered with this method will correctly respect the `setPackage(String)` (https://developer.android.com/reference/android/content/Intent.html#setPackage(java.lang.String)) specified for an Intent being broadcast. Prior to that, it would be ignored and delivered to all matching registered receivers. Be careful if using this for security.

Parameters	
`receiver`	`BroadcastReceiver`: The BroadcastReceiver to handle the broadcast.
`filter`	`IntentFilter`: Selects the Intent broadcasts to be received.
`broadcastPermission`	`String`: String naming a permissions that a broadcaster must hold in order to send an Intent to you. If null, no permission is required.
`scheduler`	`Handler`: Handler identifying the thread that will receive the Intent. If null, the main thread of the process will be used.
`flags`	`int`: Additional options for the receiver. May be 0 or `RECEIVER_VISIBLE_TO_INSTANT_APPS` (https://developer.android.com/reference/android/content/Context.html#RECEIVER_VISIBLE_TO_INSTANT_APPS).

Returns	
`Intent` (https://developer.android.com/reference/android/content/Intent.html)	The first sticky intent found that matches *filter*, or null if there are none.

See also:

`registerReceiver(BroadcastReceiver, IntentFilter, int)` (https://developer.android.com/reference/android/content/Context.html#registerReceiver(android.content.BroadcastReceiver, android.content.IntentFilter, int))

`registerReceiver(BroadcastReceiver, IntentFilter, String, Handler)` (https://developer.android.com

/reference/android/content/Context.html#registerReceiver(android.content.BroadcastReceiver, android.content.IntentFilter, java.lang.String, android.os.Handler))

`sendBroadcast(Intent)` (https://developer.android.com/reference/android/content/Context.html#sendBroadcast(android.content.Intent))

`unregisterReceiver(BroadcastReceiver)` (https://developer.android.com/reference/android/content/Context.html#unregisterReceiver(android.content.BroadcastReceiver))

registerReceiver

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

```
Intent (https://developer.android.com/reference/android/content/Intent.html) registerReceiver (BroadcastRece
                IntentFilter (https://developer.android.com/reference/android/content/IntentFilter.html) filte
                String (https://developer.android.com/reference/java/lang/String.html) broadcastPermission,
                Handler (https://developer.android.com/reference/android/os/Handler.html) scheduler)
```

Register to receive intent broadcasts, to run in the context of *scheduler*. See `registerReceiver(BroadcastReceiver, IntentFilter)` (https://developer.android.com/reference/android/content/Context.html#registerReceiver(android.content.BroadcastReceiver, android.content.IntentFilter)) for more information. This allows you to enforce permissions on who can broadcast intents to your receiver, or have the receiver run in a different thread than the main application thread.

See `BroadcastReceiver` (https://developer.android.com/reference/android/content/BroadcastReceiver.html) for more information on Intent broadcasts.

As of `ICE_CREAM_SANDWICH` (https://developer.android.com/reference/android/os/Build.VERSION_CODES.html#ICE_CREAM_SANDWICH), receivers registered with this method will correctly respect the `setPackage(String)` (https://developer.android.com/reference/android/content/Intent.html#setPackage(java.lang.String)) specified for an Intent being broadcast. Prior to that, it would be ignored and delivered to all matching registered receivers. Be careful if using this for security.

Parameters	
`receiver`	`BroadcastReceiver`: The BroadcastReceiver to handle the broadcast.
`filter`	`IntentFilter`: Selects the Intent broadcasts to be received.
`broadcastPermission`	`String`: String naming a permissions that a broadcaster must hold in order to send an Intent to you. If null, no permission is required.
`scheduler`	`Handler`: Handler identifying the thread that will receive the Intent. If null, the main thread of the process will be used.

Returns	
Intent (https://developer.android.com/reference/android/content/Intent.html)	The first sticky intent found that matches *filter*, or null if there are none.

See also:

registerReceiver(BroadcastReceiver, IntentFilter) (https://developer.android.com/reference/android/content/Context.html#registerReceiver(android.content.BroadcastReceiver, android.content.IntentFilter))

sendBroadcast(Intent) (https://developer.android.com/reference/android/content/Context.html#sendBroadcast(android.content.Intent))

unregisterReceiver(BroadcastReceiver) (https://developer.android.com/reference/android/content/Context.html#unregisterReceiver(android.content.BroadcastReceiver))

removeStickyBroadcast

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

```
void removeStickyBroadcast (Intent (https://developer.android.com/reference/android/content/Intent.html) int
```

> **This method was deprecated in API level 21.**
> Sticky broadcasts should not be used. They provide no security (anyone can access them), no protection (anyone can modify them), and many other problems. The recommended pattern is to use a non-sticky broadcast to report that *something* has changed, with another mechanism for apps to retrieve the current value whenever desired.

Remove the data previously sent with sendStickyBroadcast(Intent) (https://developer.android.com/reference/android/content/Context.html#sendStickyBroadcast(android.content.Intent)), so that it is as if the sticky broadcast had never happened.

Requires the BROADCAST_STICKY (https://developer.android.com/reference/android/Manifest.permission.html#BROADCAST_STICKY) permission.

Parameters	
intent	Intent: The Intent that was previously broadcast.

See also:

sendStickyBroadcast(Intent) (https://developer.android.com/reference/android/content/Context.html#sendStickyBroadcast(android.content.Intent))

removeStickyBroadcastAsUser added in API level 17 (https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

```
void removeStickyBroadcastAsUser (Intent (https://developer.android.com/reference/android/content/Intent.htm
                UserHandle (https://developer.android.com/reference/android/os/UserHandle.html) user)
```

> **This method was deprecated in API level 21.**
> Sticky broadcasts should not be used. They provide no security (anyone can access them), no protection (anyone can modify them), and many other problems. The recommended pattern is to use a non-sticky broadcast to report that *something* has changed, with another mechanism for apps to retrieve the current value whenever desired.

Version of `removeStickyBroadcast(Intent)` (https://developer.android.com/reference/android/content/Context.html#removeStickyBroadcast(android.content.Intent)) that allows you to specify the user the broadcast will be sent to. This is not available to applications that are not pre-installed on the system image.

You must hold the `BROADCAST_STICKY` (https://developer.android.com/reference/android/Manifest.permission.html#BROADCAST_STICKY) permission in order to use this API. If you do not hold that permission, `SecurityException` (https://developer.android.com/reference/java/lang/SecurityException.html) will be thrown.

Requires the `BROADCAST_STICKY` (https://developer.android.com/reference/android/Manifest.permission.html#BROADCAST_STICKY) permission.

Parameters	
`intent`	`Intent`: The Intent that was previously broadcast.
`user`	`UserHandle`: UserHandle to remove the sticky broadcast from.

See also:

`sendStickyBroadcastAsUser(Intent, UserHandle)` (https://developer.android.com/reference/android/content/Context.html#sendStickyBroadcastAsUser(android.content.Intent, android.os.UserHandle))

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

```
void revokeUriPermission (Uri (https://developer.android.com/reference/android/net/Uri.html) uri,
                int modeFlags)
```

Remove all permissions to access a particular content provider Uri that were previously added with `grantUriPermission(String, Uri, int)` (https://developer.android.com/reference/android/content/Context.html#grantUriPermission(java.lang.String, android.net.Uri, int)) or *any other* mechanism. The given Uri will match all previously granted Uris that are the same or a sub-path of the given Uri. That is, revoking "content://foo/target" will revoke both "content://foo/target" and "content://foo/target/sub", but not "content://foo". It will not remove any prefix grants that exist at a higher level.

Prior to LOLLIPOP (https://developer.android.com/reference/android/os/Build.VERSION_CODES.html#LOLLIPOP), if you did not have regular permission access to a Uri, but had received access to it through a specific Uri permission grant, you could not revoke that grant with this function and a SecurityException (https://developer.android.com/reference/java/lang/SecurityException.html) would be thrown. As of LOLLIPOP (https://developer.android.com/reference/android/os/Build.VERSION_CODES.html#LOLLIPOP), this function will not throw a security exception, but will remove whatever permission grants to the Uri had been given to the app (or none).

Unlike revokeUriPermission(String, Uri, int) (https://developer.android.com/reference/android/content/Context.html#revokeUriPermission(java.lang.String, android.net.Uri, int)), this method impacts all permission grants matching the given Uri, for any package they had been granted to, through any mechanism this had happened (such as indirectly through the clipboard, activity launch, service start, etc). That means this can be potentially dangerous to use, as it can revoke grants that another app could be strongly expecting to stick around.

Parameters	
uri	Uri: The Uri you would like to revoke access to.
modeFlags	int: The access modes to revoke. Value is either 0 or combination of FLAG_GRANT_READ_URI_PERMISSION (https://developer.android.com/reference/android/content/Intent.html#FLAG_GRANT_READ_URI_PERMISSION) or FLAG_GRANT_WRITE_URI_PERMISSION (https://developer.android.com/reference/android/content/Intent.html#FLAG_GRANT_WRITE_URI_PERMISSION).

See also:

grantUriPermission(String, Uri, int) (https://developer.android.com/reference/android/content/Context.html#grantUriPermission(java.lang.String, android.net.Uri, int))

revokeUriPermission

added in API level 26 (https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

```
void revokeUriPermission (String (https://developer.android.com/reference/java/lang/String.html) toPackage,
                Uri (https://developer.android.com/reference/android/net/Uri.html) uri,
                int modeFlags)
```

Remove permissions to access a particular content provider Uri that were previously added with grantUriPermission(String, Uri, int) (https://developer.android.com/reference/android/content/Context.html#grantUriPermission(java.lang.String, android.net.Uri, int)) for a specific target package. The given Uri will match all previously granted Uris that are the same or a sub-path of the given Uri. That is, revoking "content://foo/target" will revoke both "content://foo/target" and "content://foo/target/sub", but not "content://foo". It will not remove any prefix grants that exist at a higher level.

Unlike `revokeUriPermission(Uri, int)` (https://developer.android.com/reference/android/content/Context.html#revokeUriPermission(android.net.Uri, int)), this method will *only* revoke permissions that had been explicitly granted through `grantUriPermission(String, Uri, int)` (https://developer.android.com/reference/android/content/Context.html#grantUriPermission(java.lang.String, android.net.Uri, int)) and only for the package specified. Any matching grants that have happened through other mechanisms (clipboard, activity launching, service starting, etc) will not be removed.

Parameters	
`toPackage`	`String`: The package you had previously granted access to.
`uri`	`Uri`: The Uri you would like to revoke access to.
`modeFlags`	`int`: The access modes to revoke. Value is either `0` or combination of `FLAG_GRANT_READ_URI_PERMISSION` (https://developer.android.com/reference/android/content/Intent.html#FLAG_GRANT_READ_URI_PERMISSION) or `FLAG_GRANT_WRITE_URI_PERMISSION` (https://developer.android.com/reference/android/content/Intent.html#FLAG_GRANT_WRITE_URI_PERMISSION).

See also:

`grantUriPermission(String, Uri, int)` (https://developer.android.com/reference/android/content/Context.html#grantUriPermission(java.lang.String, android.net.Uri, int))

sendBroadcast

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

```
void sendBroadcast (Intent (https://developer.android.com/reference/android/content/Intent.html) intent,
                String (https://developer.android.com/reference/java/lang/String.html) receiverPermission)
```

Broadcast the given intent to all interested BroadcastReceivers, allowing an optional required permission to be enforced. This call is asynchronous; it returns immediately, and you will continue executing while the receivers are run. No results are propagated from receivers and receivers can not abort the broadcast. If you want to allow receivers to propagate results or abort the broadcast, you must send an ordered broadcast using `sendOrderedBroadcast(Intent, String)` (https://developer.android.com/reference/android/content/Context.html#sendOrderedBroadcast(android.content.Intent, java.lang.String)).

See `BroadcastReceiver` (https://developer.android.com/reference/android/content/BroadcastReceiver.html) for more information on Intent broadcasts.

Parameters

intent	Intent: The Intent to broadcast; all receivers matching this Intent will receive the broadcast.
receiverPermission	String: (optional) String naming a permission that a receiver must hold in order to receive your broadcast. If null, no permission is required.

See also:

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

registerReceiver(BroadcastReceiver, IntentFilter) (https://developer.android.com/reference/android/content/Context.html#registerReceiver(android.content.BroadcastReceiver, android.content.IntentFilter))

sendBroadcast(Intent) (https://developer.android.com/reference/android/content/Context.html#sendBroadcast(android.content.Intent))

sendOrderedBroadcast(Intent, String) (https://developer.android.com/reference/android/content/Context.html#sendOrderedBroadcast(android.content.Intent, java.lang.String))

sendOrderedBroadcast(Intent, String, BroadcastReceiver, Handler, int, String, Bundle) (https://developer.android.com/reference/android/content/Context.html#sendOrderedBroadcast(android.content.Intent, java.lang.String, android.content.BroadcastReceiver, android.os.Handler, int, java.lang.String, android.os.Bundle))

sendBroadcast

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

```
void sendBroadcast (Intent (https://developer.android.com/reference/android/content/Intent.html) intent)
```

Broadcast the given intent to all interested BroadcastReceivers. This call is asynchronous; it returns immediately, and you will continue executing while the receivers are run. No results are propagated from receivers and receivers can not abort the broadcast. If you want to allow receivers to propagate results or abort the broadcast, you must send an ordered broadcast using sendOrderedBroadcast(Intent, String) (https://developer.android.com/reference/android/content/Context.html#sendOrderedBroadcast(android.content.Intent, java.lang.String)).

See BroadcastReceiver (https://developer.android.com/reference/android/content/BroadcastReceiver.html) for more information on Intent broadcasts.

Parameters	
intent	Intent: The Intent to broadcast; all receivers matching this Intent will receive the broadcast.

See also:

BroadcastReceiver (https://developer.android.com/reference/android/content/BroadcastReceiver.html) registerReceiver(BroadcastReceiver, IntentFilter) (https://developer.android.com/reference/android/content/Context.html#registerReceiver(android.content.BroadcastReceiver, android.content.IntentFilter))

sendBroadcast(Intent, String) (https://developer.android.com/reference/android/content/Context.html#sendBroadcast(android.content.Intent, java.lang.String))

sendOrderedBroadcast(Intent, String) (https://developer.android.com/reference/android/content/Context.html#sendOrderedBroadcast(android.content.Intent, java.lang.String))

sendOrderedBroadcast(Intent, String, BroadcastReceiver, Handler, int, String, Bundle) (https://developer.android.com/reference/android/content/Context.html#sendOrderedBroadcast(android.content.Intent, java.lang.String, android.content.BroadcastReceiver, android.os.Handler, int, java.lang.String, android.os.Bundle))

sendBroadcastAsUser

added in API level 17 (https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

```
void sendBroadcastAsUser (Intent (https://developer.android.com/reference/android/content/Intent.html) inten
                UserHandle (https://developer.android.com/reference/android/os/UserHandle.html) user)
```

Version of sendBroadcast(Intent) (https://developer.android.com/reference/android/content/Context.html#sendBroadcast(android.content.Intent)) that allows you to specify the user the broadcast will be sent to. This is not available to applications that are not pre-installed on the system image.

Requires the permission.

Parameters	
intent	Intent: The intent to broadcast
user	UserHandle: UserHandle to send the intent to.

See also:

sendBroadcast(Intent) (https://developer.android.com/reference/android/content/Context.html#sendBroadcast(android.content.Intent))

sendBroadcastAsUser

added in API level 17 (https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

```
void sendBroadcastAsUser (Intent (https://developer.android.com/reference/android/content/Intent.html) inten
                UserHandle (https://developer.android.com/reference/android/os/UserHandle.html) user,
                String (https://developer.android.com/reference/java/lang/String.html) receiverPermission)
```

Version of sendBroadcast(Intent, String) (https://developer.android.com/reference/android/content/Context.html#sendBroadcast(android.content.Intent, java.lang.String)) that allows you to specify the user the broadcast will be sent to. This is not available to applications that are not pre-installed on the system image.

Requires the permission.

Parameters	
intent	Intent: The Intent to broadcast; all receivers matching this Intent will receive the broadcast.
user	UserHandle: UserHandle to send the intent to.
receiverPermission	String: (optional) String naming a permission that a receiver must hold in order to receive your broadcast. If null, no permission is required.

See also:

sendBroadcast(Intent, String) (https://developer.android.com/reference/android/content/Context.html#sendBroadcast(android.content.Intent, java.lang.String))

sendOrderedBroadcast

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

```
void sendOrderedBroadcast (Intent (https://developer.android.com/reference/android/content/Intent.html) inte
                String (https://developer.android.com/reference/java/lang/String.html) receiverPermission,
                BroadcastReceiver (https://developer.android.com/reference/android/content/BroadcastReceiver.h
                Handler (https://developer.android.com/reference/android/os/Handler.html) scheduler,
                int initialCode,
                String (https://developer.android.com/reference/java/lang/String.html) initialData,
                Bundle (https://developer.android.com/reference/android/os/Bundle.html) initialExtras)
```

Version of sendBroadcast(Intent) (https://developer.android.com/reference/android/content/Context.html#sendBroadcast(android.content.Intent)) that allows you to receive data back from the broadcast. This is accomplished by supplying your own BroadcastReceiver when calling, which will be treated as a final receiver at the end of the broadcast -- its onReceive(Context, Intent) (https://developer.android.com/reference/android/content/BroadcastReceiver.html#onReceive(android.content.Context, android.content.Intent)) method will be called with the result values collected from the other receivers. The broadcast will be serialized in the same way as calling sendOrderedBroadcast(Intent, String) (https://developer.android.com/reference/android/content/Context.html#sendOrderedBroadcast(android.content.Intent, java.lang.String)).

Like sendBroadcast(Intent) (https://developer.android.com/reference/android/content/Context.html#sendBroadcast(android.content.Intent)), this method is asynchronous; it will return before resultReceiver.onReceive() is called.

See BroadcastReceiver (https://developer.android.com/reference/android/content/BroadcastReceiver.html) for more information on Intent broadcasts.

Parameters	
intent	Intent: The Intent to broadcast; all receivers matching this Intent will receive the broadcast. This value must never be null.
receiverPermission	String: String naming a permissions that a receiver must hold in order to receive your broadcast. If null, no permission is required.
resultReceiver	BroadcastReceiver: Your own BroadcastReceiver to treat as the final receiver of the broadcast. This value may be null.
scheduler	Handler: A custom Handler with which to schedule the resultReceiver callback; if null it will be scheduled in the Context's main thread.
initialCode	int: An initial value for the result code. Often Activity.RESULT_OK.
initialData	String: An initial value for the result data. Often null.
initialExtras	Bundle: An initial value for the result extras. Often null.

See also:

sendBroadcast(Intent) (https://developer.android.com/reference/android/content/Context.html#sendBroadcast(android.content.Intent))

sendBroadcast(Intent, String) (https://developer.android.com/reference/android/content/Context.html#sendBroadcast(android.content.Intent, java.lang.String))

sendOrderedBroadcast(Intent, String) (https://developer.android.com/reference/android/content/Context.html#sendOrderedBroadcast(android.content.Intent, java.lang.String))

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

registerReceiver(BroadcastReceiver, IntentFilter) (https://developer.android.com/reference/android/content/Context.html#registerReceiver(android.content.BroadcastReceiver, android.content.IntentFilter))

RESULT_OK (https://developer.android.com/reference/android/app/Activity.html#RESULT_OK)

sendOrderedBroadcast

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

```
void sendOrderedBroadcast (Intent (https://developer.android.com/reference/android/content/Intent.html) inte
                String (https://developer.android.com/reference/java/lang/String.html) receiverPermission)
```

Broadcast the given intent to all interested BroadcastReceivers, delivering them one at a time to allow more preferred receivers to consume the broadcast before it is delivered to less preferred receivers. This call is asynchronous; it returns immediately, and you will continue executing while the receivers are run.

See BroadcastReceiver (https://developer.android.com/reference/android/content/BroadcastReceiver.html) for more information on Intent broadcasts.

Parameters	
intent	Intent: The Intent to broadcast; all receivers matching this Intent will receive the broadcast.
receiverPermission	String: (optional) String naming a permissions that a receiver must hold in order to receive your broadcast. If null, no permission is required.

See also:

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

registerReceiver(BroadcastReceiver, IntentFilter) (https://developer.android.com/reference/android/content/Context.html#registerReceiver(android.content.BroadcastReceiver, android.content.IntentFilter))

sendBroadcast(Intent) (https://developer.android.com/reference/android/content/Context.html#sendBroadcast(android.content.Intent))

sendOrderedBroadcast(Intent, String, BroadcastReceiver, Handler, int, String, Bundle) (https://developer.android.com/reference/android/content/Context.html#sendOrderedBroadcast(android.content.Intent, java.lang.String, android.content.BroadcastReceiver, android.os.Handler, int, java.lang.String, android.os.Bundle))

sendOrderedBroadcastAsUser

added in API level 17 (https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

```
void sendOrderedBroadcastAsUser (Intent (https://developer.android.com/reference/android/content/Intent.html
                UserHandle (https://developer.android.com/reference/android/os/UserHandle.html) user,
                String (https://developer.android.com/reference/java/lang/String.html) receiverPermission,
                BroadcastReceiver (https://developer.android.com/reference/android/content/BroadcastReceiver.h
                Handler (https://developer.android.com/reference/android/os/Handler.html) scheduler,
                int initialCode,
                String (https://developer.android.com/reference/java/lang/String.html) initialData,
                Bundle (https://developer.android.com/reference/android/os/Bundle.html) initialExtras)
```

Version of sendOrderedBroadcast(Intent, String, BroadcastReceiver, Handler, int, String, Bundle) (https://developer.android.com/reference/android/content/Context.html#sendOrderedBroadcast(android.content.Intent, java.lang.String, android.content.BroadcastReceiver, android.os.Handler, int, java.lang.String, android.os.Bundle)) that allows you to specify the user the broadcast will be sent to. This is not available to applications that are not pre-installed on the system image.

See `BroadcastReceiver` (https://developer.android.com/reference/android/content/BroadcastReceiver.html) for more information on Intent broadcasts.

Requires the permission.

Parameters	
`intent`	`Intent`: The Intent to broadcast; all receivers matching this Intent will receive the broadcast.
`user`	`UserHandle`: UserHandle to send the intent to.
`receiverPermission`	`String`: String naming a permissions that a receiver must hold in order to receive your broadcast. If null, no permission is required.
`resultReceiver`	`BroadcastReceiver`: Your own BroadcastReceiver to treat as the final receiver of the broadcast.
`scheduler`	`Handler`: A custom Handler with which to schedule the resultReceiver callback; if null it will be scheduled in the Context's main thread.
`initialCode`	`int`: An initial value for the result code. Often Activity.RESULT_OK.
`initialData`	`String`: An initial value for the result data. Often null.
`initialExtras`	`Bundle`: An initial value for the result extras. Often null.

See also:

`sendOrderedBroadcast(Intent, String, BroadcastReceiver, Handler, int, String, Bundle)` (https://developer.android.com/reference/android/content/Context.html#sendOrderedBroadcast(android.content.Intent, java.lang.String, android.content.BroadcastReceiver, android.os.Handler, int, java.lang.String, android.os.Bundle))

sendStickyBroadcast

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

```
void sendStickyBroadcast (Intent (https://developer.android.com/reference/android/content/Intent.html) inten
```

> **This method was deprecated in API level 21.**
> Sticky broadcasts should not be used. They provide no security (anyone can access them), no protection (anyone can modify them), and many other problems. The recommended pattern is to use a non-sticky broadcast to report that *something* has changed, with another mechanism for apps to retrieve the current value whenever desired.

Perform a `sendBroadcast(Intent)` (https://developer.android.com/reference/android/content/Context.html#sendBroadcast(android.content.Intent)) that is "sticky," meaning the Intent you are sending stays around after the broadcast is complete, so that others can quickly retrieve that data through the return value of `registerReceiver(BroadcastReceiver, IntentFilter)` (https://developer.android.com/reference/android/content/Context.html#registerReceiver(android.content.BroadcastReceiver, android.content.IntentFilter)). In all other ways, this behaves the same as `sendBroadcast(Intent)` (https://developer.android.com/reference/android/content/Context.html#sendBroadcast(android.content.Intent)).

Requires the `BROADCAST_STICKY` (https://developer.android.com/reference/android/Manifest.permission.html#BROADCAST_STICKY) permission.

Parameters	
`intent`	`Intent`: The Intent to broadcast; all receivers matching this Intent will receive the broadcast, and the Intent will be held to be re-broadcast to future receivers.

See also:

`sendBroadcast(Intent)` (https://developer.android.com/reference/android/content/Context.html#sendBroadcast(android.content.Intent))

`sendStickyOrderedBroadcast(Intent, BroadcastReceiver, Handler, int, String, Bundle)` (https://developer.android.com/reference/android/content/Context.html#sendStickyOrderedBroadcast(android.content.Intent, android.content.BroadcastReceiver, android.os.Handler, int, java.lang.String, android.os.Bundle))

sendStickyBroadcastAsUser

added in API level 17 (https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

```
void sendStickyBroadcastAsUser (Intent (https://developer.android.com/reference/android/content/Intent.html)
                UserHandle (https://developer.android.com/reference/android/os/UserHandle.html) user)
```

> **This method was deprecated in API level 21.**
> Sticky broadcasts should not be used. They provide no security (anyone can access them), no protection (anyone can modify them), and many other problems. The recommended pattern is to use a non-sticky broadcast to report that *something* has changed, with another mechanism for apps to retrieve the current value whenever desired.

Version of `sendStickyBroadcast(Intent)` (https://developer.android.com/reference/android/content/Context.html#sendStickyBroadcast(android.content.Intent)) that allows you to specify the user the broadcast will be sent to. This is not available to applications that are not pre-installed on the system image.

Requires the `BROADCAST_STICKY` (https://developer.android.com/reference/android/Manifest.permission.html#BROADCAST_STICKY) permission.

Parameters	
intent	Intent: The Intent to broadcast; all receivers matching this Intent will receive the broadcast, and the Intent will be held to be re-broadcast to future receivers.
user	UserHandle: UserHandle to send the intent to.

See also:

sendBroadcast(Intent) (https://developer.android.com/reference/android/content/Context.html#sendBroadcast(android.content.Intent))

sendStickyOrderedBroadcast

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

```
void sendStickyOrderedBroadcast (Intent (https://developer.android.com/reference/android/content/Intent.html
                BroadcastReceiver (https://developer.android.com/reference/android/content/BroadcastReceiver.h
                Handler (https://developer.android.com/reference/android/os/Handler.html) scheduler,
                int initialCode,
                String (https://developer.android.com/reference/java/lang/String.html) initialData,
                Bundle (https://developer.android.com/reference/android/os/Bundle.html) initialExtras)
```

> **This method was deprecated in API level 21.**
> Sticky broadcasts should not be used. They provide no security (anyone can access them), no protection (anyone can modify them), and many other problems. The recommended pattern is to use a non-sticky broadcast to report that *something* has changed, with another mechanism for apps to retrieve the current value whenever desired.

Version of sendStickyBroadcast(Intent) (https://developer.android.com/reference/android/content/Context.html#sendStickyBroadcast(android.content.Intent)) that allows you to receive data back from the broadcast. This is accomplished by supplying your own BroadcastReceiver when calling, which will be treated as a final receiver at the end of the broadcast -- its onReceive(Context, Intent) (https://developer.android.com/reference/android/content/BroadcastReceiver.html#onReceive(android.content.Context, android.content.Intent)) method will be called with the result values collected from the other receivers. The broadcast will be serialized in the same way as calling sendOrderedBroadcast(Intent, String) (https://developer.android.com/reference/android/content/Context.html#sendOrderedBroadcast(android.content.Intent, java.lang.String)).

Like sendBroadcast(Intent) (https://developer.android.com/reference/android/content/Context.html#sendBroadcast(android.content.Intent)), this method is asynchronous; it will return before resultReceiver.onReceive() is called. Note that the sticky data stored is only the data you initially supply to the broadcast, not the result of any changes made by the receivers.

See BroadcastReceiver (https://developer.android.com/reference/android/content/BroadcastReceiver.html) for more information on Intent broadcasts.

Requires the BROADCAST_STICKY (https://developer.android.com/reference/android/Manifest.permission.html#BROADCAST_STICKY) permission.

Parameters	
intent	Intent: The Intent to broadcast; all receivers matching this Intent will receive the broadcast.
resultReceiver	BroadcastReceiver: Your own BroadcastReceiver to treat as the final receiver of the broadcast.
scheduler	Handler: A custom Handler with which to schedule the resultReceiver callback; if null it will be scheduled in the Context's main thread.
initialCode	int: An initial value for the result code. Often Activity.RESULT_OK.
initialData	String: An initial value for the result data. Often null.
initialExtras	Bundle: An initial value for the result extras. Often null.

See also:

sendBroadcast(Intent) (https://developer.android.com/reference/android/content/Context.html#sendBroadcast(android.content.Intent))

sendBroadcast(Intent, String) (https://developer.android.com/reference/android/content/Context.html#sendBroadcast(android.content.Intent, java.lang.String))

sendOrderedBroadcast(Intent, String) (https://developer.android.com/reference/android/content/Context.html#sendOrderedBroadcast(android.content.Intent, java.lang.String))

sendStickyBroadcast(Intent) (https://developer.android.com/reference/android/content/Context.html#sendStickyBroadcast(android.content.Intent))

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

registerReceiver(BroadcastReceiver, IntentFilter) (https://developer.android.com/reference/android/content/Context.html#registerReceiver(android.content.BroadcastReceiver, android.content.IntentFilter))

RESULT_OK (https://developer.android.com/reference/android/app/Activity.html#RESULT_OK)

sendStickyOrderedBroadcastAsUser

added in API level 17 (https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

```
void sendStickyOrderedBroadcastAsUser (Intent (https://developer.android.com/reference/android/content/Inte
                UserHandle (https://developer.android.com/reference/android/os/UserHandle.html) user,
                BroadcastReceiver (https://developer.android.com/reference/android/content/BroadcastReceiver.h
                Handler (https://developer.android.com/reference/android/os/Handler.html) scheduler,
                int initialCode,
                String (https://developer.android.com/reference/java/lang/String.html) initialData,
                Bundle (https://developer.android.com/reference/android/os/Bundle.html) initialExtras)
```

> **This method was deprecated in API level 21.**
> Sticky broadcasts should not be used. They provide no security (anyone can access them), no protection (anyone can modify them), and many other problems. The recommended pattern is to use a non-sticky broadcast to report that *something* has changed, with another mechanism for apps to retrieve the current value whenever desired.

Version of `sendStickyOrderedBroadcast(Intent, BroadcastReceiver, Handler, int, String, Bundle)` (https://developer.android.com/reference/android/content/Context.html#sendStickyOrderedBroadcast(android.content.Intent, android.content.BroadcastReceiver, android.os.Handler, int, java.lang.String, android.os.Bundle)) that allows you to specify the user the broadcast will be sent to. This is not available to applications that are not pre-installed on the system image.

See `BroadcastReceiver` (https://developer.android.com/reference/android/content/BroadcastReceiver.html) for more information on Intent broadcasts.

Requires the `BROADCAST_STICKY` (https://developer.android.com/reference/android/Manifest.permission.html#BROADCAST_STICKY) permission.

Parameters	
`intent`	`Intent`: The Intent to broadcast; all receivers matching this Intent will receive the broadcast.
`user`	`UserHandle`: UserHandle to send the intent to.
`resultReceiver`	`BroadcastReceiver`: Your own BroadcastReceiver to treat as the final receiver of the broadcast.
`scheduler`	`Handler`: A custom Handler with which to schedule the resultReceiver callback; if null it will be scheduled in the Context's main thread.
`initialCode`	`int`: An initial value for the result code. Often Activity.RESULT_OK.
`initialData`	`String`: An initial value for the result data. Often null.
`initialExtras`	`Bundle`: An initial value for the result extras. Often null.

See also:

`sendStickyOrderedBroadcast(Intent, BroadcastReceiver, Handler, int, String, Bundle)` (https://developer.android.com/reference/android/content/Context.html#sendStickyOrderedBroadcast(android.content.Intent, android.content.BroadcastReceiver, android.os.Handler, int, java.lang.String, android.os.Bundle))

setTheme

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

```
void setTheme (int resid)
```

Set the base theme for this context. Note that this should be called before any views are instantiated in the Context (for example before calling `setContentView(View)` (https://developer.android.com/reference/android/app/Activity.html#setContentView(android.view.View)) or `inflate(int, ViewGroup)` (https://developer.android.com/reference/android/view/LayoutInflater.html#inflate(int, android.view.ViewGroup))).

Parameters	
`resid`	`int`: The style resource describing the theme.

setWallpaper

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

```
void setWallpaper (Bitmap (https://developer.android.com/reference/android/graphics/Bitmap.html) bitmap)
```

> **This method was deprecated in API level 5.**
> Use `WallpaperManager.set()` (https://developer.android.com/reference/android/app/WallpaperManager.html#setBitmap(android.graphics.Bitmap)) instead.

This method requires the caller to hold the permission `SET_WALLPAPER` (https://developer.android.com/reference/android/Manifest.permission.html#SET_WALLPAPER).

Parameters	
`bitmap`	`Bitmap`

Throws	
`IOException` (https://developer.android.com/reference/java/io/IOException.html)	

setWallpaper

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

```
void setWallpaper (InputStream (https://developer.android.com/reference/java/io/InputStream.html) data)
```

> **This method was deprecated in API level 5.**

> Use WallpaperManager.set() (https://developer.android.com/reference/android/app/WallpaperManager.html#setStream(java.io.InputStream)) instead.

This method requires the caller to hold the permission SET_WALLPAPER (https://developer.android.com/reference/android/Manifest.permission.html#SET_WALLPAPER).

Parameters	
data	InputStream

Throws	
IOException (https://developer.android.com/reference/java/io/IOException.html)	

startActivities

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

```
void startActivities (Intent[] (https://developer.android.com/reference/android/content/Intent.html) intents,
                Bundle (https://developer.android.com/reference/android/os/Bundle.html) options)
```

Launch multiple new activities. This is generally the same as calling startActivity(Intent) (https://developer.android.com/reference/android/content/Context.html#startActivity(android.content.Intent)) for the first Intent in the array, that activity during its creation calling startActivity(Intent) (https://developer.android.com/reference/android/content/Context.html#startActivity(android.content.Intent)) for the second entry, etc. Note that unlike that approach, generally none of the activities except the last in the array will be created at this point, but rather will be created when the user first visits them (due to pressing back from the activity on top).

This method throws ActivityNotFoundException (https://developer.android.com/reference/android/content/ActivityNotFoundException.html) if there was no Activity found for *any* given Intent. In this case the state of the activity stack is undefined (some Intents in the list may be on it, some not), so you probably want to avoid such situations.

Parameters	
intents	Intent: An array of Intents to be started.
options	Bundle: Additional options for how the Activity should be started. See startActivity(Intent, Bundle) (https://developer.android.com/reference/android/content/Context.html#startActivity(android.content.Intent, android.os.Bundle)) Context.startActivity(Intent, Bundle)} for more details.

Throws	
ActivityNotFoundException (https://developer.android.com/reference/android/content/ActivityNotFoundException.html)	

See also:

startActivities(Intent[]) (https://developer.android.com/reference/android/content/Context.html#startActivities(android.content.Intent[]))

resolveActivity(Intent, int) (https://developer.android.com/reference/android/content/pm/PackageManager.html#resolveActivity(android.content.Intent, int))

startActivities

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

```
void startActivities (Intent[] (https://developer.android.com/reference/android/content/Intent.html) intents
```

Same as startActivities(Intent[], Bundle) (https://developer.android.com/reference/android/content/Context.html#startActivities(android.content.Intent[], android.os.Bundle)) with no options specified.

Parameters	
intents	Intent: An array of Intents to be started.

Throws	
ActivityNotFoundException (https://developer.android.com/reference/android/content/ActivityNotFoundException.html)	

See also:

startActivities(Intent[], Bundle) (https://developer.android.com/reference/android/content/Context.html#startActivities(android.content.Intent[], android.os.Bundle))

resolveActivity(Intent, int) (https://developer.android.com/reference/android/content/pm/PackageManager.html#resolveActivity(android.content.Intent, int))

startActivity

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

```
void startActivity (Intent (https://developer.android.com/reference/android/content/Intent.html) intent)
```

Same as `startActivity(Intent, Bundle)` (https://developer.android.com/reference/android/content/Context.html#startActivity(android.content.Intent, android.os.Bundle)) with no options specified.

Parameters	
`intent`	`Intent`: The description of the activity to start.

Throws	
`ActivityNotFoundException` (https://developer.android.com/reference/android/content/ActivityNotFoundException.html)	`

See also:

`startActivity(Intent, Bundle)` (https://developer.android.com/reference/android/content/Context.html#startActivity(android.content.Intent, android.os.Bundle))

`resolveActivity(Intent, int)` (https://developer.android.com/reference/android/content/pm/PackageManager.html#resolveActivity(android.content.Intent, int))

startActivity

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

```
void startActivity (Intent (https://developer.android.com/reference/android/content/Intent.html) intent,
                Bundle (https://developer.android.com/reference/android/os/Bundle.html) options)
```

Launch a new activity. You will not receive any information about when the activity exits.

Note that if this method is being called from outside of an `Activity` (https://developer.android.com/reference/android/app/Activity.html) Context, then the Intent must include the `FLAG_ACTIVITY_NEW_TASK` (https://developer.android.com/reference/android/content/Intent.html#FLAG_ACTIVITY_NEW_TASK) launch flag. This is because, without being started from an existing Activity, there is no existing task in which to place the new activity and thus it needs to be placed in its own separate task.

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

Parameters

`intent`	`Intent`: The description of the activity to start.
`options`	`Bundle`: Additional options for how the Activity should be started. May be null if there are no options. See `ActivityOptions` (https://developer.android.com/reference/android/app/ActivityOptions.html) for how to build the Bundle supplied here; there are no supported definitions for building it manually.

Throws	
`ActivityNotFoundException` (https://developer.android.com/reference/android/content/ActivityNotFoundException.html)	

See also:

`startActivity(Intent)` (https://developer.android.com/reference/android/content/Context.html#startActivity(android.content.Intent))

`resolveActivity(Intent, int)` (https://developer.android.com/reference/android/content/pm/PackageManager.html#resolveActivity(android.content.Intent, int))

startForegroundService

added in API level 26 (https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

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

Similar to `startService(Intent)` (https://developer.android.com/reference/android/content/Context.html#startService(android.content.Intent)), but with an implicit promise that the Service will call `startForeground(int, android.app.Notification)` (https://developer.android.com/reference/android/app/Service.html#startForeground(int, android.app.Notification)) once it begins running. The service is given an amount of time comparable to the ANR interval to do this, otherwise the system will automatically stop the service and declare the app ANR.

Unlike the ordinary `startService(Intent)` (https://developer.android.com/reference/android/content/Context.html#startService(android.content.Intent)), this method can be used at any time, regardless of whether the app hosting the service is in a foreground state.

Parameters	
`service`	`Intent`: Identifies the service to be started. The Intent must be fully explicit (supplying a component name). Additional values may be included in the Intent extras to supply arguments along with this specific start call.

Returns	
ComponentName (https://developer.android.com/reference/android/content/ComponentName.html)	If the service is being started or is already running, the ComponentName (https://developer.android.com/reference/android/content/ComponentName.html) of the actual service that was started is returned; else if the service does not exist null is returned.

Throws	
SecurityException (https://developer.android.com/reference/java/lang/SecurityException.html)	If the caller does not have permission to access the service or the service can not be found.

See also:

stopService(Intent) (https://developer.android.com/reference/android/content/Context.html#stopService(android.content.Intent))

startForeground(int, android.app.Notification) (https://developer.android.com/reference/android/app/Service.html#startForeground(int, android.app.Notification))

startInstrumentation

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

```
boolean startInstrumentation (ComponentName (https://developer.android.com/reference/android/content/Compon
                String (https://developer.android.com/reference/java/lang/String.html) profileFile,
                Bundle (https://developer.android.com/reference/android/os/Bundle.html) arguments)
```

Start executing an Instrumentation (https://developer.android.com/reference/android/app/Instrumentation.html) class. The given Instrumentation component will be run by killing its target application (if currently running), starting the target process, instantiating the instrumentation component, and then letting it drive the application.

This function is not synchronous -- it returns as soon as the instrumentation has started and while it is running.

Instrumentation is normally only allowed to run against a package that is either unsigned or signed with a signature that the the instrumentation package is also signed with (ensuring the target trusts the instrumentation).

Parameters	
className	ComponentName: Name of the Instrumentation component to be run. This value must never be null.
profileFile	String: Optional path to write profiling data as the instrumentation runs, or null for no profiling.

arguments	Bundle: Additional optional arguments to pass to the instrumentation, or null.

Returns	
boolean	true if the instrumentation was successfully started, else false if it could not be found.

startIntentSender

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

```
void startIntentSender (IntentSender (https://developer.android.com/reference/android/content/IntentSender.ht
                Intent (https://developer.android.com/reference/android/content/Intent.html) fillInIntent,
                int flagsMask,
                int flagsValues,
                int extraFlags)
```

Same as startIntentSender(IntentSender, Intent, int, int, int, Bundle) (https://developer.android.com/reference/android/content/Context.html#startIntentSender(android.content.IntentSender, android.content.Intent, int, int, int, android.os.Bundle)) with no options specified.

Parameters	
intent	IntentSender: The IntentSender to launch.
fillInIntent	Intent: If non-null, this will be provided as the intent parameter to sendIntent(Context, int, Intent, IntentSender.OnFinished, Handler) (https://developer.android.com/reference/android/content/IntentSender.html#sendIntent(android.content.Context, int, android.content.Intent, android.content.IntentSender.OnFinished, android.os.Handler)).
flagsMask	int: Intent flags in the original IntentSender that you would like to change. Value is either 0 or combination of FLAG_FROM_BACKGROUND (https://developer.android.com/reference/android/content/Intent.html#FLAG_FROM_BACKGROUND), FLAG_DEBUG_LOG_RESOLUTION (https://developer.android.com/reference/android/content/Intent.html#FLAG_DEBUG_LOG_RESOLUTION), FLAG_EXCLUDE_STOPPED_PACKAGES (https://developer.android.com/reference/android/content/Intent.html#FLAG_EXCLUDE_STOPPED_PACKAGES), FLAG_INCLUDE_STOPPED_PACKAGES (https://developer.android.com/reference/android/content/Intent.html#FLAG_INCLUDE_STOPPED_PACKAGES), FLAG_RECEIVER_REGISTERED_ONLY (https://developer.android.com/reference/android/content/Intent.html#FLAG_RECEIVER_REGISTERED_ONLY), FLAG_RECEIVER_REPLACE_PENDING (https://developer.android.com/reference/android/content/Intent.html#FLAG_RECEIVER_REPLACE_PENDING), FLAG_RECEIVER_FOREGROUND (https://developer.android.com/reference/android/content/Intent.html#FLAG_RECEIVER_FOREGROUND),

FLAG_RECEIVER_NO_ABORT (https://developer.android.com/reference/android/content/Intent.html#FLAG_RECEIVER_NO_ABORT), FLAG_ACTIVITY_CLEAR_TOP (https://developer.android.com/reference/android/content/Intent.html#FLAG_ACTIVITY_CLEAR_TOP), FLAG_ACTIVITY_FORWARD_RESULT (https://developer.android.com/reference/android/content/Intent.html#FLAG_ACTIVITY_FORWARD_RESULT), FLAG_ACTIVITY_PREVIOUS_IS_TOP (https://developer.android.com/reference/android/content/Intent.html#FLAG_ACTIVITY_PREVIOUS_IS_TOP), FLAG_ACTIVITY_EXCLUDE_FROM_RECENTS (https://developer.android.com/reference/android/content/Intent.html#FLAG_ACTIVITY_EXCLUDE_FROM_RECENTS), FLAG_ACTIVITY_BROUGHT_TO_FRONT (https://developer.android.com/reference/android/content/Intent.html#FLAG_ACTIVITY_BROUGHT_TO_FRONT), FLAG_RECEIVER_VISIBLE_TO_INSTANT_APPS (https://developer.android.com/reference/android/content/Intent.html#FLAG_RECEIVER_VISIBLE_TO_INSTANT_APPS), FLAG_ACTIVITY_LAUNCHED_FROM_HISTORY (https://developer.android.com/reference/android/content/Intent.html#FLAG_ACTIVITY_LAUNCHED_FROM_HISTORY), FLAG_ACTIVITY_NEW_DOCUMENT (https://developer.android.com/reference/android/content/Intent.html#FLAG_ACTIVITY_NEW_DOCUMENT), FLAG_ACTIVITY_NO_USER_ACTION (https://developer.android.com/reference/android/content/Intent.html#FLAG_ACTIVITY_NO_USER_ACTION), FLAG_ACTIVITY_REORDER_TO_FRONT (https://developer.android.com/reference/android/content/Intent.html#FLAG_ACTIVITY_REORDER_TO_FRONT), FLAG_ACTIVITY_NO_ANIMATION (https://developer.android.com/reference/android/content/Intent.html#FLAG_ACTIVITY_NO_ANIMATION), FLAG_ACTIVITY_CLEAR_TASK (https://developer.android.com/reference/android/content/Intent.html#FLAG_ACTIVITY_CLEAR_TASK), FLAG_ACTIVITY_TASK_ON_HOME (https://developer.android.com/reference/android/content/Intent.html#FLAG_ACTIVITY_TASK_ON_HOME), FLAG_ACTIVITY_RETAIN_IN_RECENTS (https://developer.android.com/reference/android/content/Intent.html#FLAG_ACTIVITY_RETAIN_IN_RECENTS) or FLAG_ACTIVITY_LAUNCH_ADJACENT (https://developer.android.com/reference/android/content/Intent.html#FLAG_ACTIVITY_LAUNCH_ADJACENT).

flagsValues

int: Desired values for any bits set in *flagsMask*
Value is either 0 or combination of FLAG_FROM_BACKGROUND (https://developer.android.com/reference/android/content/Intent.html#FLAG_FROM_BACKGROUND), FLAG_DEBUG_LOG_RESOLUTION (https://developer.android.com/reference/android/content/Intent.html#FLAG_DEBUG_LOG_RESOLUTION), FLAG_EXCLUDE_STOPPED_PACKAGES (https://developer.android.com/reference/android/content/Intent.html#FLAG_EXCLUDE_STOPPED_PACKAGES), FLAG_INCLUDE_STOPPED_PACKAGES (https://developer.android.com/reference/android/content/Intent.html#FLAG_INCLUDE_STOPPED_PACKAGES), FLAG_RECEIVER_REGISTERED_ONLY

	(https://developer.android.com/reference/android/content/Intent.html#FLAG_RECEIVER_REGISTERED_ONLY), FLAG_RECEIVER_REPLACE_PENDING (https://developer.android.com/reference/android/content/Intent.html#FLAG_RECEIVER_REPLACE_PENDING), FLAG_RECEIVER_FOREGROUND (https://developer.android.com/reference/android/content/Intent.html#FLAG_RECEIVER_FOREGROUND), FLAG_RECEIVER_NO_ABORT (https://developer.android.com/reference/android/content/Intent.html#FLAG_RECEIVER_NO_ABORT), FLAG_ACTIVITY_CLEAR_TOP (https://developer.android.com/reference/android/content/Intent.html#FLAG_ACTIVITY_CLEAR_TOP), FLAG_ACTIVITY_FORWARD_RESULT (https://developer.android.com/reference/android/content/Intent.html#FLAG_ACTIVITY_FORWARD_RESULT), FLAG_ACTIVITY_PREVIOUS_IS_TOP (https://developer.android.com/reference/android/content/Intent.html#FLAG_ACTIVITY_PREVIOUS_IS_TOP), FLAG_ACTIVITY_EXCLUDE_FROM_RECENTS (https://developer.android.com/reference/android/content/Intent.html#FLAG_ACTIVITY_EXCLUDE_FROM_RECENTS), FLAG_ACTIVITY_BROUGHT_TO_FRONT (https://developer.android.com/reference/android/content/Intent.html#FLAG_ACTIVITY_BROUGHT_TO_FRONT), FLAG_RECEIVER_VISIBLE_TO_INSTANT_APPS (https://developer.android.com/reference/android/content/Intent.html#FLAG_RECEIVER_VISIBLE_TO_INSTANT_APPS), FLAG_ACTIVITY_LAUNCHED_FROM_HISTORY (https://developer.android.com/reference/android/content/Intent.html#FLAG_ACTIVITY_LAUNCHED_FROM_HISTORY), FLAG_ACTIVITY_NEW_DOCUMENT (https://developer.android.com/reference/android/content/Intent.html#FLAG_ACTIVITY_NEW_DOCUMENT), FLAG_ACTIVITY_NO_USER_ACTION (https://developer.android.com/reference/android/content/Intent.html#FLAG_ACTIVITY_NO_USER_ACTION), FLAG_ACTIVITY_REORDER_TO_FRONT (https://developer.android.com/reference/android/content/Intent.html#FLAG_ACTIVITY_REORDER_TO_FRONT), FLAG_ACTIVITY_NO_ANIMATION (https://developer.android.com/reference/android/content/Intent.html#FLAG_ACTIVITY_NO_ANIMATION), FLAG_ACTIVITY_CLEAR_TASK (https://developer.android.com/reference/android/content/Intent.html#FLAG_ACTIVITY_CLEAR_TASK), FLAG_ACTIVITY_TASK_ON_HOME (https://developer.android.com/reference/android/content/Intent.html#FLAG_ACTIVITY_TASK_ON_HOME), FLAG_ACTIVITY_RETAIN_IN_RECENTS (https://developer.android.com/reference/android/content/Intent.html#FLAG_ACTIVITY_RETAIN_IN_RECENTS) or FLAG_ACTIVITY_LAUNCH_ADJACENT (https://developer.android.com/reference/android/content/Intent.html#FLAG_ACTIVITY_LAUNCH_ADJACENT).
extraFlags	int: Always set to 0.

Throws

IntentSender.SendIntentException (https://developer.android.com/reference/android/content/IntentSender.SendIntentException.html)

See also:

startActivity(Intent) (https://developer.android.com/reference/android/content/Context.html#startActivity(android.content.Intent))

startIntentSender(IntentSender, Intent, int, int, int, Bundle) (https://developer.android.com/reference/android/content/Context.html#startIntentSender(android.content.IntentSender, android.content.Intent, int, int, int, android.os.Bundle))

startIntentSender

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

```
void startIntentSender (IntentSender (https://developer.android.com/reference/android/content/IntentSender.ht
                Intent (https://developer.android.com/reference/android/content/Intent.html) fillInIntent,
                int flagsMask,
                int flagsValues,
                int extraFlags,
                Bundle (https://developer.android.com/reference/android/os/Bundle.html) options)
```

Like startActivity(Intent, Bundle) (https://developer.android.com/reference/android/content/Context.html#startActivity(android.content.Intent, android.os.Bundle)), but taking a IntentSender to start. If the IntentSender is for an activity, that activity will be started as if you had called the regular startActivity(Intent) (https://developer.android.com/reference/android/content/Context.html#startActivity(android.content.Intent)) here; otherwise, its associated action will be executed (such as sending a broadcast) as if you had called IntentSender.sendIntent (https://developer.android.com/reference/android/content/IntentSender.html#sendIntent(android.content.Context, int, android.content.Intent, android.content.IntentSender.OnFinished, android.os.Handler)) on it.

Parameters	
intent	IntentSender: The IntentSender to launch.
fillInIntent	Intent: If non-null, this will be provided as the intent parameter to sendIntent(Context, int, Intent, IntentSender.OnFinished, Handler) (https://developer.android.com/reference/android/content/IntentSender.html#sendIntent(android.content.Context, int, android.content.Intent, android.content.IntentSender.OnFinished, android.os.Handler)).

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

Value is either `0` or combination of `FLAG_FROM_BACKGROUND` (https://developer.android.com/reference/android/content/Intent.html#FLAG_FROM_BACKGROUND), `FLAG_DEBUG_LOG_RESOLUTION` (https://developer.android.com/reference/android/content/Intent.html#FLAG_DEBUG_LOG_RESOLUTION), `FLAG_EXCLUDE_STOPPED_PACKAGES` (https://developer.android.com/reference/android/content/Intent.html#FLAG_EXCLUDE_STOPPED_PACKAGES), `FLAG_INCLUDE_STOPPED_PACKAGES` (https://developer.android.com/reference/android/content/Intent.html#FLAG_INCLUDE_STOPPED_PACKAGES), `FLAG_RECEIVER_REGISTERED_ONLY` (https://developer.android.com/reference/android/content/Intent.html#FLAG_RECEIVER_REGISTERED_ONLY), `FLAG_RECEIVER_REPLACE_PENDING` (https://developer.android.com/reference/android/content/Intent.html#FLAG_RECEIVER_REPLACE_PENDING), `FLAG_RECEIVER_FOREGROUND` (https://developer.android.com/reference/android/content/Intent.html#FLAG_RECEIVER_FOREGROUND), `FLAG_RECEIVER_NO_ABORT` (https://developer.android.com/reference/android/content/Intent.html#FLAG_RECEIVER_NO_ABORT), `FLAG_ACTIVITY_CLEAR_TOP` (https://developer.android.com/reference/android/content/Intent.html#FLAG_ACTIVITY_CLEAR_TOP), `FLAG_ACTIVITY_FORWARD_RESULT` (https://developer.android.com/reference/android/content/Intent.html#FLAG_ACTIVITY_FORWARD_RESULT), `FLAG_ACTIVITY_PREVIOUS_IS_TOP` (https://developer.android.com/reference/android/content/Intent.html#FLAG_ACTIVITY_PREVIOUS_IS_TOP), `FLAG_ACTIVITY_EXCLUDE_FROM_RECENTS` (https://developer.android.com/reference/android/content/Intent.html#FLAG_ACTIVITY_EXCLUDE_FROM_RECENTS), `FLAG_ACTIVITY_BROUGHT_TO_FRONT` (https://developer.android.com/reference/android/content/Intent.html#FLAG_ACTIVITY_BROUGHT_TO_FRONT), `FLAG_RECEIVER_VISIBLE_TO_INSTANT_APPS` (https://developer.android.com/reference/android/content/Intent.html#FLAG_RECEIVER_VISIBLE_TO_INSTANT_APPS), `FLAG_ACTIVITY_LAUNCHED_FROM_HISTORY` (https://developer.android.com/reference/android/content/Intent.html#FLAG_ACTIVITY_LAUNCHED_FROM_HISTORY), `FLAG_ACTIVITY_NEW_DOCUMENT` (https://developer.android.com/reference/android/content/Intent.html#FLAG_ACTIVITY_NEW_DOCUMENT), `FLAG_ACTIVITY_NO_USER_ACTION` (https://developer.android.com/reference/android/content/Intent.html#FLAG_ACTIVITY_NO_USER_ACTION), `FLAG_ACTIVITY_REORDER_TO_FRONT` (https://developer.android.com/reference/android/content/Intent.html#FLAG_ACTIVITY_REORDER_TO_FRONT), `FLAG_ACTIVITY_NO_ANIMATION` (https://developer.android.com/reference/android/content/Intent.html#FLAG_ACTIVITY_NO_ANIMATION), `FLAG_ACTIVITY_CLEAR_TASK` (https://developer.android.com/reference/android/content/Intent.html#FLAG_ACTIVITY_CLEAR_TASK), `FLAG_ACTIVITY_TASK_ON_HOME` (https://developer.android.com/reference/android/content

	/Intent.html#FLAG_ACTIVITY_TASK_ON_HOME), FLAG_ACTIVITY_RETAIN_IN_RECENTS (https://developer.android.com/reference/android/content/Intent.html#FLAG_ACTIVITY_RETAIN_IN_RECENTS) or FLAG_ACTIVITY_LAUNCH_ADJACENT (https://developer.android.com/reference/android/content/Intent.html#FLAG_ACTIVITY_LAUNCH_ADJACENT).
flagsValues	int: Desired values for any bits set in *flagsMask* Value is either 0 or combination of FLAG_FROM_BACKGROUND (https://developer.android.com/reference/android/content/Intent.html#FLAG_FROM_BACKGROUND), FLAG_DEBUG_LOG_RESOLUTION (https://developer.android.com/reference/android/content/Intent.html#FLAG_DEBUG_LOG_RESOLUTION), FLAG_EXCLUDE_STOPPED_PACKAGES (https://developer.android.com/reference/android/content/Intent.html#FLAG_EXCLUDE_STOPPED_PACKAGES), FLAG_INCLUDE_STOPPED_PACKAGES (https://developer.android.com/reference/android/content/Intent.html#FLAG_INCLUDE_STOPPED_PACKAGES), FLAG_RECEIVER_REGISTERED_ONLY (https://developer.android.com/reference/android/content/Intent.html#FLAG_RECEIVER_REGISTERED_ONLY), FLAG_RECEIVER_REPLACE_PENDING (https://developer.android.com/reference/android/content/Intent.html#FLAG_RECEIVER_REPLACE_PENDING), FLAG_RECEIVER_FOREGROUND (https://developer.android.com/reference/android/content/Intent.html#FLAG_RECEIVER_FOREGROUND), FLAG_RECEIVER_NO_ABORT (https://developer.android.com/reference/android/content/Intent.html#FLAG_RECEIVER_NO_ABORT), FLAG_ACTIVITY_CLEAR_TOP (https://developer.android.com/reference/android/content/Intent.html#FLAG_ACTIVITY_CLEAR_TOP), FLAG_ACTIVITY_FORWARD_RESULT (https://developer.android.com/reference/android/content/Intent.html#FLAG_ACTIVITY_FORWARD_RESULT), FLAG_ACTIVITY_PREVIOUS_IS_TOP (https://developer.android.com/reference/android/content/Intent.html#FLAG_ACTIVITY_PREVIOUS_IS_TOP), FLAG_ACTIVITY_EXCLUDE_FROM_RECENTS (https://developer.android.com/reference/android/content/Intent.html#FLAG_ACTIVITY_EXCLUDE_FROM_RECENTS), FLAG_ACTIVITY_BROUGHT_TO_FRONT (https://developer.android.com/reference/android/content/Intent.html#FLAG_ACTIVITY_BROUGHT_TO_FRONT), FLAG_RECEIVER_VISIBLE_TO_INSTANT_APPS (https://developer.android.com/reference/android/content/Intent.html#FLAG_RECEIVER_VISIBLE_TO_INSTANT_APPS), FLAG_ACTIVITY_LAUNCHED_FROM_HISTORY (https://developer.android.com/reference/android/content/Intent.html#FLAG_ACTIVITY_LAUNCHED_FROM_HISTORY), FLAG_ACTIVITY_NEW_DOCUMENT (https://developer.android.com/reference/android/content/Intent.html#FLAG_ACTIVITY_NEW_DOCUMENT), FLAG_ACTIVITY_NO_USER_ACTION (https://developer.android.com/reference/android/content/Intent.html#FLAG_ACTIVITY_NO_USER_ACTION), FLAG_ACTIVITY_REORDER_TO_FRONT (https://developer.android.com/reference/android/content

	/Intent.html#FLAG_ACTIVITY_REORDER_TO_FRONT), FLAG_ACTIVITY_NO_ANIMATION (https://developer.android.com/reference/android/content /Intent.html#FLAG_ACTIVITY_NO_ANIMATION), FLAG_ACTIVITY_CLEAR_TASK (https://developer.android.com/reference/android/content/Intent.html#FLAG_ACTIVITY_CLEAR_TASK), FLAG_ACTIVITY_TASK_ON_HOME (https://developer.android.com/reference/android/content /Intent.html#FLAG_ACTIVITY_TASK_ON_HOME), FLAG_ACTIVITY_RETAIN_IN_RECENTS (https://developer.android.com/reference/android/content /Intent.html#FLAG_ACTIVITY_RETAIN_IN_RECENTS) or FLAG_ACTIVITY_LAUNCH_ADJACENT (https://developer.android.com/reference/android/content /Intent.html#FLAG_ACTIVITY_LAUNCH_ADJACENT).
extraFlags	int: Always set to 0.
options	Bundle: Additional options for how the Activity should be started. See startActivity(Intent, Bundle) (https://developer.android.com/reference/android/content /Context.html#startActivity(android.content.Intent, android.os.Bundle)) 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. This value may be null.

Throws	
IntentSender.SendIntentException (https://developer.android.com/reference /android/content /IntentSender.SendIntentException.html)	

See also:

startActivity(Intent, Bundle) (https://developer.android.com/reference/android/content /Context.html#startActivity(android.content.Intent, android.os.Bundle))

startIntentSender(IntentSender, Intent, int, int, int) (https://developer.android.com/reference/android /content/Context.html#startIntentSender(android.content.IntentSender, android.content.Intent, int, int, int))

startService

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

ComponentName (https://developer.android.com/reference/android/content/ComponentName.html) startService (Inter

Request that a given application service be started. The Intent should either contain the complete class name of a specific service implementation to start, or a specific package name to target. If the Intent is less specified, it logs a warning about this. In this case any of the multiple matching services may be used. If this service is not already running, it will be instantiated and started (creating a process for it if needed); if it is running then it remains running.

Every call to this method will result in a corresponding call to the target service's `onStartCommand(Intent, int, int)` (https://developer.android.com/reference/android/app/Service.html#onStartCommand(android.content.Intent, int, int)) method, with the *intent* given here. This provides a convenient way to submit jobs to a service without having to bind and call on to its interface.

Using startService() overrides the default service lifetime that is managed by `bindService(Intent, ServiceConnection, int)` (https://developer.android.com/reference/android/content/Context.html#bindService(android.content.Intent, android.content.ServiceConnection, int)): it requires the service to remain running until `stopService(Intent)` (https://developer.android.com/reference/android/content/Context.html#stopService(android.content.Intent)) is called, regardless of whether any clients are connected to it. Note that calls to startService() do not nest: no matter how many times you call startService(), a single call to `stopService(Intent)` (https://developer.android.com/reference/android/content/Context.html#stopService(android.content.Intent)) will stop it.

The system attempts to keep running services around as much as possible. The only time they should be stopped is if the current foreground application is using so many resources that the service needs to be killed. If any errors happen in the service's process, it will automatically be restarted.

This function will throw `SecurityException` (https://developer.android.com/reference/java/lang/SecurityException.html) if you do not have permission to start the given service.

> **Note:** Each call to startService() results in significant work done by the system to manage service lifecycle surrounding the processing of the intent, which can take multiple milliseconds of CPU time. Due to this cost, startService() should not be used for frequent intent delivery to a service, and only for scheduling significant work. Use `bound services` (https://developer.android.com/reference/android/content/Context.html#bindService(android.content.Intent, android.content.ServiceConnection, int)) for high frequency calls.

Parameters	
`service`	`Intent`: Identifies the service to be started. The Intent must be fully explicit (supplying a component name). Additional values may be included in the Intent extras to supply arguments along with this specific start call.

Returns	
`ComponentName` (https://developer.android.com/reference/android/content	If the service is being started or is already running, the `ComponentName` (https://developer.android.com/reference/android/content/ComponentName.html) of the actual service that was started is returned; else if the service does not exist

/ComponentName.html)	null is returned.

Throws	
SecurityException (https://developer.android.com/reference/java/lang/SecurityException.html)	If the caller does not have permission to access the service or the service can not be found.
IllegalStateException (https://developer.android.com/reference/java/lang/IllegalStateException.html)	If the application is in a state where the service can not be started (such as not in the foreground in a state when services are allowed).

See also:

stopService(Intent) (https://developer.android.com/reference/android/content/Context.html#stopService(android.content.Intent))

bindService(Intent, ServiceConnection, int) (https://developer.android.com/reference/android/content/Context.html#bindService(android.content.Intent, android.content.ServiceConnection, int))

stopService

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

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

Request that a given application service be stopped. If the service is not running, nothing happens. Otherwise it is stopped. Note that calls to startService() are not counted -- this stops the service no matter how many times it was started.

Note that if a stopped service still has ServiceConnection (https://developer.android.com/reference/android/content/ServiceConnection.html) objects bound to it with the BIND_AUTO_CREATE (https://developer.android.com/reference/android/content/Context.html#BIND_AUTO_CREATE) set, it will not be destroyed until all of these bindings are removed. See the Service (https://developer.android.com/reference/android/app/Service.html) documentation for more details on a service's lifecycle.

This function will throw SecurityException (https://developer.android.com/reference/java/lang/SecurityException.html) if you do not have permission to stop the given service.

Parameters

service	Intent: Description of the service to be stopped. The Intent must be either fully explicit (supplying a component name) or specify a specific package name it is targetted to.

Returns	
boolean	If there is a service matching the given Intent that is already running, then it is stopped and true is returned; else false is returned.

Throws	
SecurityException (https://developer.android.com/reference/java/lang/SecurityException.html)	If the caller does not have permission to access the service or the service can not be found.
IllegalStateException (https://developer.android.com/reference/java/lang/IllegalStateException.html)	If the application is in a state where the service can not be started (such as not in the foreground in a state when services are allowed).

See also:

startService(Intent) (https://developer.android.com/reference/android/content/Context.html#startService(android.content.Intent))

unbindService

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

```
void unbindService (ServiceConnection (https://developer.android.com/reference/android/content/ServiceConnect
```

Disconnect from an application service. You will no longer receive calls as the service is restarted, and the service is now allowed to stop at any time.

Parameters	
conn	ServiceConnection: The connection interface previously supplied to bindService(). This parameter must not be null.

See also:

bindService(Intent, ServiceConnection, int) (https://developer.android.com/reference/android/content/Context.html#bindService(android.content.Intent, android.content.ServiceConnection, int))

unregisterComponentCallbacks

added in API level 14 (https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

```
void unregisterComponentCallbacks (ComponentCallbacks (https://developer.android.com/reference/android/cc
```

Remove a `ComponentCallbacks` (https://developer.android.com/reference/android/content/ComponentCallbacks.html) object that was previously registered with `registerComponentCallbacks(ComponentCallbacks)` (https://developer.android.com/reference/android/content/Context.html#registerComponentCallbacks(android.content.ComponentCallbacks)).

Parameters	
`callback`	`ComponentCallbacks`

unregisterReceiver

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

```
void unregisterReceiver (BroadcastReceiver (https://developer.android.com/reference/android/content/Broadcas
```

Unregister a previously registered BroadcastReceiver. *All* filters that have been registered for this BroadcastReceiver will be removed.

Parameters	
`receiver`	`BroadcastReceiver`: The BroadcastReceiver to unregister.

See also:

`registerReceiver(BroadcastReceiver, IntentFilter)` (https://developer.android.com/reference/android/content/Context.html#registerReceiver(android.content.BroadcastReceiver, android.content.IntentFilter))

Follow @AndroidDev on Twitter

Follow Android Developers on Google+

Check out Android Developers on YouTube