



# Recent Support Library Revisions

## Recent revisions

- > [26.0.0 Release](#)
- > [26.0.0 Beta 2](#)
- > [25.4.0](#)
- > [26.0.0 Beta 1](#)
- > [26.0.0 Alpha 1](#)
- > [25.3.1](#)
- > [25.3.0](#)
- > [25.2.0](#)
- > [25.1.1](#)
- > [25.1.0](#)
- > [25.0.1](#)
- > [25.0.0](#)

## See also

- > [Support Library Revisions Archive](#)
- > [Support Library Features](#)

This page provides details about the most recent Support Library package releases. For earlier releases, see the [Support Library Revisions Archive](#).

## Revision 26.0.0 Release

(July 2017)

**Important:** The support libraries are now available through Google's Maven repository. You do not need to download the support repository from the SDK Manager. For more information, see [Support Library Setup](#).

## Important changes

- The minimum SDK version has been increased to 14. As a result, many APIs that existed only for compatibility with pre-14 API levels have been deprecated. Clients of these APIs should migrate to their framework equivalents as noted in the reference page for each deprecated API.
- The Wear UI Library contains classes that help you implement patterns and layouts that work on Wear devices. For more information, see [Using the Wear UI Library](#).
- The [Percent Support module](#) has been deprecated. Clients of this module should migrate to the new [ConstraintLayout](#) widget, which is provided as a separate artifact in SDK Manager.
- [NotificationCompat](#) and its containing classes has been deprecated and will be removed in a future release:
  - Use [NotificationCompat.Builder](#) instead of [v7.app.NotificationCompat.Builder](#). Functionality that previously relied on using the v7 AppCompat Builder has now been folded into the v4 Compat Builder.
  - [DecoratedCustomViewStyle](#) has moved to the [android.support.v4.app](#) package.
  - [MediaStyle](#) and [DecoratedMediaCustomViewStyle](#) are now part of the [media-compat library](#) and can be found in the

[android.support.v4.media](#) package.

## New APIs

- New `fastScrollEnabled` boolean flag for `RecyclerView`. If enabled, `fastScrollHorizontalThumbDrawable`, `fastScrollHorizontalTrackDrawable`, `fastScrollVerticalThumbDrawable`, and `fastScrollVerticalTrackDrawable` must be set.

## API Diffs

- [26.0.0-beta2 → 26.0.0 Release](#)
- [25.4.0 → 26.0.0 Release](#)

## Bug fixes

- Infinite loop in `RecyclerView.toString()`
- `ResourceNotFoundException` running Kotlin project on API 16 AVD after upgrading to Canary 6
- `java.lang.AssertionError` in design view with support library 26.0.0-beta2
- Android Studio layout preview broken for Support Library widgets
- `Preference.setSingleLineTitle()` is ignored if the Preference was not created with attributes
- DAC "Since" annotations are wrong for 25.3.0 / 25.4.0 revisions of Support Library
- `ResourcesCompat.getFont()` throws exception
- Toolbar title not in bold font
- Auto sizing with `maxLines` produces unexpected results
- `NullPointerException` in `TextView.checkForRelayout()`
- `AppCompatActivityAutoSizeHelper.setRawTextSize()` calls `requestLayout()` during layout
- `EmojiAppCompatActivity` crashes
- Autosize `TextView` does not adjust automatically when text is changed
- Screen corruption in Instacart
- `UnsupportedOperationException` in `MenuItemCompat`
- `NotificationCompat` doesn't fully extract actions on API 24 or higher
- `CoordinatorLayout` anchoring problems on layout updates

## Revision 26.0.0 Beta 2

---

(June 2017)

Please note that 26.0.0-beta2 is a pre-release version. Its API surface is subject to change, and it does not necessarily include features or bug fixes from the latest stable versions of Support Library.

**Important:** The support libraries are now available through Google's Maven repository. You do not need to download the support repository from the SDK Manager. For more information, see [Support Library Setup](#).

## New APIs

- New `JobIntentService` class, to help developers schedule tasks in a way that complies with the new Android O [background execution limits](#).

## API Diffs

- [26.0.0-beta1 -> 26.0.0-beta2](#)
- [25.4.0 -> 26.0.0-beta2](#)

## Bug fixes

- Android O SDK drop causes loss of italics in TextViews
- Null pointer exception when connecting to MediaBrowserServiceCompat
- TextInputLayout must set hints on onProvideAutofillStructure()
- Stack overflow when using TextView autosize on O

## Revision 25.4.0

(June 2017)

**Important:** The support libraries are now available through Google's Maven repository. You do not need to download the support repository from the SDK Manager. For more information, see [Support Library Setup](#).

## Important changes

- [executePendingTransactions\(\)](#), [commitNow\(\)](#), [popBackStackImmediate\(\)](#), and similar transaction calls are not allowed during [FragmentManager](#) state changes. Reentrant execution of transactions are unsafe and [FragmentManager](#) now enforces this during its state changes.
- Concurrent with this support library release, we are also releasing [multidex](#) version 1.0.2. This version includes the following important changes:
  - Allows multidexing of instrumentation APK.
  - Deprecates MultiDexTestRunner (AndroidJUnitRunner should be used instead).
  - Provides better protection against some bad archive extraction management of the app.
  - Fixes a bug that could lead to abandoned temporary files.
  - Provides faster installation when done in concurrent process.
  - Fixes an installation bug on API 19 and 20.

## New and Modified APIs

Path morphing and path interpolation are supported in [AnimatedVectorDrawableCompat](#). Path morphing allow the shapes changing from one path (specified as [android:valueFrom](#)) to another path (specified as [android:valueTo](#)), in order to provide complex and attractive visual effects. Path interpolation allows the interpolators for [AnimatedVectorDrawableCompat](#) to be specified as paths (specified as [android:pathData](#) in the interpolator's XML).

## API Diffs

- [25.3.0 -> 25.4.0](#)

## Fixed issues

- Null pointer exception when connecting to MediaBrowserServiceCompat
- MediaBrowserCompat.search() API does not work (AOSP issue [262170](#))

- `BrowseFragment.onItemClicked` callbacks broken in 25.3.0
- `NullPointerException` while scrolling up and down in `VerticalGridView` in 25.3.1
- `ClassCastException` in `SimpleArrayMap.allocArrays()`

## Revision 26.0.0 Beta 1

(May 2017)

Please note that 26.0.0-beta1 is a pre-release version. Its API surface is subject to change, and it does not necessarily include features or bug fixes from the latest stable versions of Support Library.

**Important:** The support libraries are now available through Google's Maven repository. You do not need to download the support repository from the SDK Manager. For more information, see [Support Library Setup](#).

### Important changes

- `FragmentManager.setSupportMediaController()` and `FragmentManager.getSupportMediaController()` have been removed. Please use the new static `MediaControllerCompat.setMediaController()` and `MediaControllerCompat.getMediaController()` methods.
- `BottomNavigationView` now calls `onNavigationItemReselected()` when an already-selected item is selected, rather than calling `onNavigationItemSelected()`.
- All instances of the `findViewById()` method now return `<T extends View> T` instead of `View`. This change has the following implications:
  - This may result in existing code now having ambiguous return type, for example if there is both `someMethod(View)` and `someMethod(Textview)` that takes the result of a call to `findViewById()`.
  - When using Java 8 source language, this requires an explicit cast to `View` when the return type is unconstrained (for example, `assertNotNull(findViewById(...)).someViewMethod()`).
  - Overrides of non-final `findViewById()` methods (for example, `Activity.findViewById()`) will need their return type updated.

### New APIs

- `FragmentManager` and `Fragment` have an `isStateSaved()` method to allow querying whether or not a transaction will be allowed without state loss. This is especially useful to check when handling an `onClick()` event before executing any transaction.
- Path motion is supported in `AnimatedVectorDrawableCompat`. Path motion allows one object animator to change two properties at the same time based on one path; the path is specified as `android:pathData` in the animator's XML).
- **Physics-based animation:**
  - New `FlingAnimation` that supports animating with an initial velocity and smoothly slows down.
  - Subclasses of `DynamicAnimation` support animating custom property for any object.
  - Both `SpringAnimation` and `FlingAnimation` can now animate a float value without requiring a `View` or an `Object` to be associated with it.

For more information, see the [Spring animation](#) and [Fling animation](#) preview pages.

- **Font support in XML:**
  - `ResourcesCompat.getFont()` allows loading font resources—including font-family XML—that may be used with `TextView.setTypeface()`.
  - When using AppCompat, `TextView` supports specifying a font resource or font-family XML via the `android:fontFamily` XML attribute.
  - Use XML font-family to create families of fonts with style and weight variations. (If you use the support library classes to do this, use the `app:` attributes as well as the `android:` attributes.)

- [Downloadable fonts](#):
  - New `FontsContractCompat` that allows you to request fonts from a font provider instead of bundling them in your app.
  - Fonts can also be requested in XML and used in layouts.
- [Emoji compatibility library](#):
  - `EmojiCompat` can process a given `CharSequence` and add `EmojiSpans`.
  - `EmojiTextView` and other widgets to display emoji.
  - `FontRequestEmojiCompatConfig` to request emoji font from a font provider.
- [Autosizing TextView](#):
  - New methods in `TextViewCompat` as well as XML attributes to control autosizing in `TextView`.
- Leanback playback controls with seek support:
  - New `PlaybackTransportRowPresenter` that renders playback controls with a `SeekBar`.
  - New `PlaybackTransportControlGlue` that works with `PlaybackTransportRowPresenter` and supports seek.
  - New base class `PlaybackSeekDataProvider` for app to provide seek thumbnails to `PlaybackTransportControlGlue`.
- Preferences Data Store:
  - `PreferenceDataStore` now allows you to implement your own preferences storage, set with new methods in `Preference` and `PreferenceManager`.

## Known Issues

- Downloadable Fonts and Emoji compatibility integration with Google Play Services only works on Google Play Services v11+, which is currently available through the [Google Play Services beta program](#).

## Bug fixes

- `MediaBrowserCompat.search()` API does not work (AOSP issue [262170](#))
- `ViewCompat.postInvalidateOnAnimation()` throws exception (AOSP issue [80146](#))
- `onActivityCreated()` called for fragments in destroyed Activity
- `RecyclerView.isComputingLayout()` should return true during prefetch
- When a `Fade` transition is interrupted and reversed, the `View` starts the animation from the beginning. (Fix ported from Android Framework.)
- `Transition.Fade` ignores initial alpha of `View` (AOSP issue [221820](#))

## Revision 26.0.0 Alpha 1

(March 2017)

Please note that 26.0.0-alpha1 is a pre-release version. Its API surface is subject to change, and it does not necessarily include features or bug fixes from the latest stable versions of Support Library.

## Important changes

**Note:** The minimum SDK version has been increased to 14. As a result, many APIs that existed only for API < 14 compatibility have been deprecated. Clients of these APIs should migrate to their framework equivalents as noted in the reference page for each deprecated API.

- The support-percent module has been deprecated. Clients of this module should migrate to the new `ConstraintLayout` widget, which is provided as a separate artifact in SDK Manager.

- The support-fragment module no longer has a dependency on the support-media-compat module.

## New APIs

Many new classes, methods, and constants added to provide backwards-compatible support for platform APIs added in O Preview.

- **IME\_FLAG\_NO\_PERSONALIZED\_LEARNING**: IMEs can listen for "no learning" flags for apps that have a private mode, such as browsers. This feature helps IMEs understand if an app is in a private mode, so they can disable their learning or adaptive functionality while the app is in that mode.

For a complete list of API changes between 25.2.0 and 26.0.0-alpha1, see the [support library API differences report](#).

## Bug fixes

- In some cases simple [AutoTransition](#) animation can be interrupted by view "jumps". (AOSP issue [221816](#))

## Revision 25.3.1

---

(March 2017)

### Fixed issues

- [SwitchCompat](#) requires minimum SDK version of 14 or higher. (AOSP issue [251302](#))
- Physics-based animation [updateListener](#) skips the first frame.
- [BottomNavigationView](#) label animation is broken.

## Revision 25.3.0

---

(March 2017)

### Important changes

Support Library version metadata will automatically be added to [AndroidManifest.xml](#) when building from Gradle, which simplifies tracking versions in public builds. For example:

```
<meta-data android:name="android.support.VERSION" android:value="25.3.0" />
```

## Deprecations

A number of methods and classes have been deprecated in this release. These deprecated APIs will be removed in a future version and developers should migrate away from them. For more information on how to migrate away from a specific API, refer to its documentation.

#### [ExifInterface](#)

The boolean method [getLatLong\(float\[\]\)](#) is deprecated. Instead, use the new method [getLatLong\(\)](#), which takes no arguments and returns [double\[\]](#).

#### [mediacompat](#)

[PlaybackStateCompat.Builder.setErrorMessage\(CharSequence\)](#) is deprecated. Instead, use the new method [setErrorMessage\(int, CharSequence\)](#), which is passed an error code and an optional description.

[EXTRA\\_SUGGESTION\\_KEYWORDS](#) is deprecated. Instead, use the [MediaBrowserCompat](#) search functionality.

#### [v7.recyclerview](#)

[LinearLayoutManager.getInitialItemPrefetchCount\(\)](#) has been renamed to

[LinearLayoutManager.getInitialPrefetchItemCount\(\)](#). The old name is still supported but will be removed in a future release.

## New and Modified APIs

### [appcompat-v7](#)

The new method [ActionBarDrawerToggle.setDrawerSlideAnimationEnabled\(boolean\)](#) simplifies disabling the navigation drawer toggle icon's animation.

### [customtabs](#)

Added support for message channels. See the [CustomTabsService.requestPostMessageChannel\(\)](#) and [CustomTabsService.postMessage\(\)](#) reference for details.

### [dynamic-animation](#)

New physics-based animation library that provides a set of APIs for building animations that dynamically react to user input.

### [leanback-v17](#)

Added support for parallax backgrounds. See the [Parallax](#) reference for details.

Added [TimePicker](#) widget for picking times on a TV interface.

### [mediacompat](#)

Added search functionality. See the [MediaBrowserCompat.search\(\)](#) and [MediaBrowserServiceCompat.onSearch\(\)](#) reference for details.

Added support for shuffle and repeat modes. See the [MediaSessionCompat.setRepeatMode\(\)](#) and [setShuffleModeEnabled\(\)](#) reference for details.

## Fixed issues

- [StaggeredGridLayoutManager](#) throws [IllegalArgumentException](#) (AOSP issue [230295](#))
- [RecyclerView](#) prefetch does not properly handle a [RecyclerView](#) that is attached but not onscreen
- [LinearLayout](#) not recognized by Robolectric
- When [Activity](#) is destroyed, [onActivityCreated\(\)](#) is improperly called for its fragments
- [AppCompatImageView](#) constructor causes [ArrayIndexOutOfBoundsException](#)
- Poor UI performance in [Call.Details](#) activity transition

## Revision 25.2.0

---

(February 2017)

## Important Changes

### Fixed issues

- This release fixes a severe mediarouter issue in which using an A2DP device and media routing APIs could cause the device to become unresponsive, requiring a reboot.
- The [FragmentManager.FragmentLifecycleCallbacks](#) class is now static.

### Fixed issues

- Showing a slide presentation with screen mirroring causes device to disconnect from Wi-Fi
- Media button did not properly handle media apps that did not register themselves with `setMediaButtonReceiver()`
- `VectorDrawable` error with string resource (AOSP issue [232407](#))
- `TextInputLayout` overlays hint and text if text is set by XML (AOSP issue [230171](#))
- Memory leak in `MediaControllerCompat` (AOSP issue [231441](#))
- `RecyclerViewLayoutTest.triggerFocusSearchInOnRecycledCallback()` crashing
- `RecyclerView` crashes when recycling view holders (AOSP issue [225762](#))
- `getAllowGeneratedReplies()` incorrectly returns false for actions inside a `WearableExtender`

## Revision 25.1.1

(January 2017)

**Important:** There is a known bug in the `android.support.v7.media.MediaRouter` class in revision 25.1.1 and 25.1.0 of the Support Library. If your app uses the v7 `MediaRouter`, you should update to [Support Library Revision 25.2.0](#), which fixes this bug.

## Important Changes

- Fragment transactions can now be optimized within and across transactions. Optimizing fragment transaction operations can eliminate operations that cancel. For example, suppose two transactions are executed together, one that adds a fragment A and a second one that replaces fragment A with fragment B. In this case, the first operation might be canceled, and only fragment B added. That means that fragment A might not go through the creation/destruction lifecycle.

A side effect of this optimization is that fragments might have state changes out of the expected order. For example, suppose one transaction adds fragment A, a second adds fragment B, then a third removes fragment A. Without optimization, fragment B could expect that while it is being created, fragment A will also exist because fragment A will be removed after fragment B is added. With optimization, fragment B cannot be sure that fragment A will exist while B is being created, because fragment A's creation and destruction may be removed by the optimization.

This optimization is disabled by default. To enable the optimization, call `FragmentManager.setAllowOptimization(true)`.

- Fragments can now postpone their transitions and animations until they are ready using `Fragment.postponeEnterTransition()` and `Fragment.startPostponedEnterTransition()`. This API is similar to `Activity.postponeEnterTransition()` and `Activity.startPostponedEnterTransition()` used with Activity Transitions.

## Fixed issues

- `MediaSessionCompatTest` fails with `IllegalArgumentException`
- `DetailsFragment.installTitleView()` is not called in 25.1.0
- Fragment transaction keeps ghost view on exit (AOSP issue [230679](#))
- `BottomNavigationView` needs spacing between item icon and text (AOSP issue [230653](#))
- Selected listeners are missing from the new `PlaybackFragment` and `PlaybackSupportFragment`
- `TextInputLayout` focus does not change properly in emulator from support library version 25.1.0 (AOSP issue [230461](#))
- Cannot replace the menu of a `BottomNavigationView` (AOSP issue [230343](#))
- `RecyclerView` with `StaggeredGridLayoutManager` crashes with full-span items (AOSP issue [230295](#))
- Crash in `MediaSessionCompat` when using `setCallback(null)`
- `PlaybackGlueHostOld` and `PlaybackSupportGlueHostOld` don't notify callbacks when playback row changes



- [PlaybackOverlayFragment](#) example `/test` can not start playing
- [RecyclerViewFocusRecoveryTest](#) is failing on API 15
- "Screenshots" row is focused to the top of the screen
- [RecyclerViewLayoutTest.triggerFocusSearchInOnRecycledCallback\(\)](#) crashes on API 15
- [setActions\(\)](#) in [onSubactionClicked\(\)](#) is broken
- [RecyclerView](#) crashes when recycling some view holders

## Revision 25.1.0

(December 2016)

**Important:** There is a known bug in the [android.support.v7.media.MediaRouter](#) class in revision 25.1.1 and 25.1.0 of the Support Library. If your app uses the v7 [MediaRouter](#), you should update to [Support Library Revision 25.2.0](#), which fixes this bug.

## Important Changes

- Clients of nested [RecyclerView](#) widgets (for example, vertical scrolling list of horizontal scrolling lists) can get significant performance benefits by hinting the inner [RecyclerView](#) widgets' layout managers how many items to prepare before being scrolled on screen. Call [LinearLayoutManager.setInitialPrefetchItemCount\(N\)](#), where *N* is the number of views visible per inner item. For example, if your inner, horizontal lists show a minimum of three and a half item views at a time, you can improve performance by calling [LinearLayoutManager.setInitialPrefetchItemCount\(4\)](#). Doing so allows [RecyclerView](#) to create all relevant views early, while the outer [RecyclerView](#) is scrolling, which significantly reduces the amount of stuttering during scrolls.
- [FragmentActivity.setSupportMediaController\(\)](#) and [FragmentActivity.getSupportMediaController\(\)](#) have been deprecated. Please use the new static [MediaControllerCompat.setMediaController\(\)](#) and [MediaControllerCompat.getMediaController\(\)](#) methods.
- When a client specifies a widget tint via appcompat tinting (for example, `appcompat:buttonTint`), the client is responsible for providing all necessary states (such as "disabled", "pressed", etc.). This is consistent with how widget tints are specified when using framework tinting.

## New and Modified APIs

- Added [ExifInterface support library](#). This library unbundles support for reading Exif information from JPEG and raw formatted files and setting the Exif information on JPEG image files.
- [Snackbar](#) has been refactored to allow apps to display custom content. [BaseTransientBottomBar](#) is the new base class that exposes the general sliding and animations behavior.
- Added a new [leanback.media](#) package which contains helper classes to integrate media players into Android TV applications.
- Added [SeekBarPreference](#) with customizable layout and attributes to the [v7 preference support library](#).
- Added [ArraySet](#) class to the v4 support library. This class corresponds to the framework [ArraySet](#) class that was introduced with API level 23.
- [RecyclerView](#) RecyclerView item prefetching improvements:
  - Nested [RecyclerView](#) prefetch enables prefetching of content from a [RecyclerView](#) within another scrolling [RecyclerView](#), with API to control how much prefetching is done:
    - [LinearLayoutManager.setInitialPrefetchItemCount\(\)](#)
    - [LinearLayoutManager.getInitialPrefetchItemCount\(\)](#)
  - APIs added for custom [LayoutManager](#) objects to implement to enable prefetching during scrolls and flings
    - [RecyclerView.LayoutManager.LayoutPrefetchRegistry\(\)](#)

- `RecyclerView.LayoutManager.collectAdjacentPrefetchPositions()`
- `RecyclerView.LayoutManager.collectInitialPrefetchPositions()`
- Improvements to prefetching to do only as much create/bind work as possible in the time between frames

## Fixed issues

- Password visibility toggle fails accessibility tests.
- Appcompat doesn't respect `state_enabled` on pre-L devices.
- Added focus recovery mechanism to `RecyclerView`. This also fixed support pref fragments broken focus when using DPAD navigation such as on Android TV devices.
- Leanback: BrowseFragment crashes with headers disabled and empty adapter.
- Appcompat: `AlertDialog` is too wide.
- `InputContentInfoCompat` calls `requestPermission()` when it should call `releasePermission()`.
- `MediaBrowserCompat` crashes.
- CoordinatorLayout measures/lays out views when visibility is set to `GONE`.
- Could not tint `AnimatedVectorDrawableCompat` on API level below 24
- Leanback library triggers spurious lint errors
- Palette library caused test failures on every API level
- `RecyclerView` failed tests on Leanback
- `RecyclerView` crashes when recycling view holders (AOSP issue [225762](#))
- `Fragment.onDestroy()` not called for fragment in backstack
- `CollapsingToolbarLayout` scrim is not drawn when collapsed
- `CoordinatorLayout.offsetChildByInset()` throws `IllegalArgumentException`
- Animating `RecyclerView` items detach inner `RecyclerViews`, prevent future prefetches
- Attached `RecyclerView` items can't be nested-prefetched
- Prefetch data for nested `RecyclerView` items is discarded during first layout
- `RecyclerView` prefetch fails if two drag events arrive at same position
- `RecyclerView` should speculatively layout while RenderThread is rendering
- Night-configured color resources converted to Drawables are not always properly purged from Resources cache
- `FloatingActionButton`: Programmatically setting BackgroundTintList does not work properly (AOSP issue [227428](#))
- `TextInputLayout`: Typeface is not getting set for ErrorView (AOSP issue [227803](#))
- `TextInputLayout` always falls back to light error color below API 23 (AOSP issue [221992](#))
- `FloatingActionButton` shows as pressed when pointer leaves

A complete list of public bug fixes is available on the [AOSP Issue Tracker](#).

## Deprecations

A number of methods and classes have been deprecated in this release. These deprecated APIs will be removed in a future version and developers should migrate away from them. For more information on how to migrate away from a specific API, refer to its documentation.

- `android.support.design.widget`

- `Snackbar.setCallback()`
- `android.support.v17.leanback.app`
  - `BackgroundManager.getDefaultDimLayer()`
  - `BackgroundManager.getDimLayer()`
  - `BackgroundManager.setDimLayer()`
  - `MediaControllerGlue.MediaControllerGlue(Context, PlaybackOverlayFragment, int[])`
  - `MediaControllerGlue.MediaControllerGlue(Context, PlaybackOverlayFragment, int[], int[])`
  - `PlaybackControlGlue.PlaybackControlGlue(Context, PlaybackOverlayFragment, int[])`
  - `PlaybackControlGlue.PlaybackControlGlue(Context, PlaybackOverlayFragment, int[], int[])`
  - `PlaybackControlGlue.getFragment()`
  - `PlaybackControlGlue.getOnItemClickListener()`
  - `PlaybackControlGlue.onRowChanged()`
  - `PlaybackControlGlue.pausePlayback()`
  - `PlaybackControlGlue.skipToNext()`
  - `PlaybackControlGlue.skipToPrevious()`
  - `PlaybackControlGlue.startPlayback()`
  - `PlaybackControlSupportGlue`
  - `PlaybackOverlayFragment`
  - `PlaybackOverlaySupportFragment`
- `android.support.v17.leanback.widget`
  - `BaseCardView.getExtraVisibility()`
  - `BaseCardView.setExtraVisibility()`
  - `GuidedActionsStylist.onEditingModeChange()`
  - `GuidedActionsStylist.setEditingMode()`
  - `GuidedActionsStylist.setExpandedViewHolder()`
  - `GuidedActionsStylist.startExpandedTransition()`
- `android.support.v4.app`
  - `FragmentActivity.getSupportMediaController()`
  - `FragmentActivity.setSupportMediaController()`

## Revision 25.0.1

---

(November 2016)

### Fixed issues

- The `TextInputLayout` password toggle is now disabled by default to avoid unnecessarily overwriting developer-specified end drawables. It may be manually enabled via the `passwordToggleEnabled` XML attribute.
- `BottomNavigationView` items are now single line to match Material spec.
- `RecyclerView` crashes during prefetch if layout manager is null.

- [BottomNavigationView](#) elevation is now set properly. (AOSP issue [226182](#))
- [BottomNavigationView](#) crashing when adding menu items programmatically. (AOSP issue [225731](#))
- Fix to [TextInputLayout](#) left+right compound drawables. (AOSP issue [225836](#))
- [RecyclerView](#) crashes when recycling view holders. (AOSP issue [225762](#))
- Leanback: TalkBack frequently says the word "null" in split-screen views.
- [RecyclerView](#): Rendering problems in Android Studio. (AOSP issue [225753](#))
- [BottomNavigationView](#) still shows menu item as selected after `onNavigationItemSelected()` returns false. (AOSP issue [225898](#))
- ForwardingListener throws [NoSuchMethodError](#). (AOSP issue [225647](#))
- [TextInputEditText](#) does not show hints in IME extract mode. (AOSP issue [221880](#))

A complete list of public bug fixes is available on the [AOSP Issue Tracker](#).

## Revision 25.0.0

---

(October 2016)

### Important changes

- [ContextCompat](#) constructor has been made protected. This class should not be publicly instantiated, but it may be extended by support libraries targeting newer API levels.
- [ActivityCompat](#) constructor has been made protected. This class should not be publicly instantiated, but it may be extended by support libraries targeting newer API levels.
- [getReferrer\(Activity\)](#) has been made static.
- [android.support.design.widget.CoordinatorLayout.Behavior.isDirty\(CoordinatorLayout, V\)](#) has been removed. Any client implementations of this method should be removed.
- [android.support.v4.media.session.MediaSessionCompat.obtain\(Context, Object\)](#) has been removed. Usages should be replaced with the more appropriately named method [fromMediaSession\(\)](#).
- [android.support.v4.media.session.MediaSessionCompat.QueueItem.obtain\(Object\)](#) has been removed. Usages should be replaced with the more appropriately named method [MediaSessionCompat.QueueItem#fromQueueItem](#).
- [android.support.v7.widget.Space](#) has been removed. Usages should be replaced with [android.support.v4.widget.Space](#).

### New APIs

- [android.support.design.widget.BottomNavigationView](#) class implements the [bottom navigation](#) pattern from the Material Design spec.
- New [android.support.v13.view.inputmethod](#) package includes classes for accessing [android.view.inputmethod.InputConnection](#) features introduced after API level 13.
- [android.v7.widget.RecyclerView.DividerItemDecoration](#) class provides a base implementation for vertical or horizontal dividers between items.
- New decorated styles in [NotificationCompat](#), [DecoratedCustomViewStyle](#) and [DecoratedMediaCustomViewStyle](#), mirror classes added in platform API 24.

### Fixed issues

A complete list of public bug fixes is available on the [AOSP Issue Tracker](#).