

VideoView

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

```
public class VideoView
```

```
extends SurfaceView implements
```

```
MediaController.MediaPlayerControl
```

```
java.lang.Object
```

```
↳ android.view.View
```

```
↳ android.view.SurfaceView
```

```
↳ android.widget.VideoView
```

Displays a video file. The VideoView class can load images from various sources (such as resources or content providers), takes care of computing its measurement from the video so that it can be used in any layout manager, and provides various display options such as scaling and tinting.

Note: VideoView does not retain its full state when going into the background. In particular, it does not restore the current play state, play position, selected tracks, or any subtitle tracks added via [`addSubtitleSource\(\)`](#).

([https://developer.android.com/reference/android/widget/VideoView.html#addSubtitleSource\(java.io.InputStream,%20android.media.MediaFormat\)\)](https://developer.android.com/reference/android/widget/VideoView.html#addSubtitleSource(java.io.InputStream,%20android.media.MediaFormat))).

Applications should save and restore these on their own in [`Activity onSaveInstanceState\(Bundle\)`](#).

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

[`Activity.onRestoreInstanceState\(Bundle\)`](#).

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

Also note that the audio session id (from [`getAudioSessionId\(\)`](https://developer.android.com/reference/android/widget/VideoView.html#getAudioSessionId()).)

([`getAudioSessionId\(\)`](https://developer.android.com/reference/android/widget/VideoView.html#getAudioSessionId())) may change from its previously returned value when the VideoView is restored.

By default, VideoView requests audio focus with [`AudioManager.AUDIOFOCUS_GAIN`](https://developer.android.com/reference/android/media/AudioManager.html#AUDIOFOCUS_GAIN)

([`AudioManager.AUDIOFOCUS_GAIN`](https://developer.android.com/reference/android/media/AudioManager.html#AUDIOFOCUS_GAIN)). Use [`setAudioFocusRequest\(int\)`](https://developer.android.com/reference/android/widget/VideoView.html#setAudioFocusRequest(int))

([`setAudioFocusRequest\(int\)`](https://developer.android.com/reference/android/widget/VideoView.html#setAudioFocusRequest(int))) to change this behavior.

The default [`AudioAttributes`](https://developer.android.com/reference/android/media/AudioAttributes.html) ([`AudioAttributes`](https://developer.android.com/reference/android/media/AudioAttributes.html)) used during playback have a usage of [`AudioAttributes.USAGE_MEDIA`](https://developer.android.com/reference/android/media/AudioAttributes.html#USAGE_MEDIA) ([`AudioAttributes.USAGE_MEDIA`](https://developer.android.com/reference/android/media/AudioAttributes.html#USAGE_MEDIA)) and a content type of [`AudioAttributes.CONTENT_TYPE_MOVIE`](https://developer.android.com/reference/android/media/AudioAttributes.html#CONTENT_TYPE_MOVIE)

([`AudioAttributes.CONTENT_TYPE_MOVIE`](https://developer.android.com/reference/android/media/AudioAttributes.html#CONTENT_TYPE_MOVIE)), use

[`setAudioAttributes\(AudioAttributes\)`](https://developer.android.com/reference/android/widget/VideoView.html#setAudioAttributes(android.media.AudioAttributes)).

([`setAudioAttributes\(android.media.AudioAttributes\)`](https://developer.android.com/reference/android/widget/VideoView.html#setAudioAttributes(android.media.AudioAttributes))) to modify them.

Summary

Inherited XML attributes

From class [`android.view.View`](https://developer.android.com/reference/android/view/View.html) ([`android.view.View`](https://developer.android.com/reference/android/view/View.html))

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

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

[android:accessibilityTraversalAfter](https://developer.android.com/reference/android/view/View.html#attr_android:accessibilityTraversalAfter) (https://developer.android.com/reference/android/view/View.html#attr_android:accessibilityTraversalAft

[android:accessibilityTraversalBefore](https://developer.android.com/reference/android/view/View.html#attr_android:accessibilityTraversalBefore) (https://developer.android.com/reference/android/view/View.html#attr_android:accessibilityTraversalB

[android:alpha](https://developer.android.com/reference/android/view/View.html#attr_android:alpha) (https://developer.android.com/reference/android/view/View.html#attr_android:alpha)

[android:autofillHints](https://developer.android.com/reference/android/view/View.html#attr_android:autofillHints) (https://developer.android.com/reference/android/view/View.html#attr_android:autofillHints)

[android:autofilledHighlight](https://developer.android.com/reference/android/view/View.html#attr_android:autofilledHighlight) (https://developer.android.com/reference/android/view/View.html#attr_android:autofilledHighlight)

[android:background](https://developer.android.com/reference/android/view/View.html#attr_android:background) (https://developer.android.com/reference/android/view/View.html#attr_android:background)

[android:backgroundTint](https://developer.android.com/reference/android/view/View.html#attr_android:backgroundTint) (https://developer.android.com/reference/android/view/View.html#attr_android:backgroundTint)

[android:backgroundTintMode](https://developer.android.com/reference/android/view/View.html#attr_android:backgroundTintMode) (https://developer.android.com/reference/android/view/View.html#attr_android:backgroundTintMode)

[android:clickable](https://developer.android.com/reference/android/view/View.html#attr_android:clickable) (https://developer.android.com/reference/android/view/View.html#attr_android:clickable)

[android:contentDescription](https://developer.android.com/reference/android/view/View.html#attr_android:contentDescription) (https://developer.android.com/reference/android/view/View.html#attr_android:contentDescription)

[android:contextClickable](https://developer.android.com/reference/android/view/View.html#attr_android:contextClickable) (https://developer.android.com/reference/android/view/View.html#attr_android:contextClickable)

[android:defaultFocusHighlightEnabled](https://developer.android.com/reference/android/view/View.html#attr_android:defaultFocusHighlightEnabled) (https://developer.android.com/reference/android/view/View.html#attr_android:defaultFocusHighlightE

[android:drawingCacheQuality](https://developer.android.com/reference/android/view/View.html#attr_android:drawingCacheQuality) (https://developer.android.com/reference/android/view/View.html#attr_android:drawingCacheQuality)

[android:duplicateParentState](https://developer.android.com/reference/android/view/View.html#attr_android:duplicateParentState) (https://developer.android.com/reference/android/view/View.html#attr_android:duplicateParentState)

[android:elevation](https://developer.android.com/reference/android/view/View.html#attr_android:elevation) (https://developer.android.com/reference/android/view/View.html#attr_android:elevation)

[android:fadeScrollbars](https://developer.android.com/reference/android/view/View.html#attr_android:fadeScrollbars) (https://developer.android.com/reference/android/view/View.html#attr_android:fadeScrollbars)

[android:fadingEdgeLength](https://developer.android.com/reference/android/view/View.html#attr_android:fadingEdgeLength) (https://developer.android.com/reference/android/view/View.html#attr_android:fadingEdgeLength)

[android:filterTouchesWhenObscured](https://developer.android.com/reference/android/view/View.html#attr_android:filterTouchesWhenObscured) (https://developer.android.com/reference/android/view/View.html#attr_android:filterTouchesWhenObscure

[android:fitsSystemWindows](https://developer.android.com/reference/android/view/View.html#attr_android:fitsSystemWindows) (https://developer.android.com/reference/android/view/View.html#attr_android:fitsSystemWindows)

[android:focusable](https://developer.android.com/reference/android/view/View.html#attr_android:focusable) (https://developer.android.com/reference/android/view/View.html#attr_android:focusable)

[android:focusableInTouchMode](https://developer.android.com/reference/android/view/View.html#attr_android:focusableInTouchMode) (https://developer.android.com/reference/android/view/View.html#attr_android:focusableInTouchMode)

[android:focusedByDefault](https://developer.android.com/reference/android/view/View.html#attr_android:focusedByDefault) (https://developer.android.com/reference/android/view/View.html#attr_android:focusedByDefault)

[android:forceHasOverlappingRendering](https://developer.android.com/reference/android/view/View.html#attr_android:forceHasOverlappingRendering) (https://developer.android.com/reference/android/view/View.html#attr_android:forceHasOverlappingRe

[android:foreground](https://developer.android.com/reference/android/view/View.html#attr_android:foreground) (https://developer.android.com/reference/android/view/View.html#attr_android:foreground)

[android:foregroundGravity](https://developer.android.com/reference/android/view/View.html#attr_android:foregroundGravity) (https://developer.android.com/reference/android/view/View.html#attr_android:foregroundGravity)

[android:foregroundTint](https://developer.android.com/reference/android/view/View.html#attr_android:foregroundTint) (https://developer.android.com/reference/android/view/View.html#attr_android:foregroundTint)

[android:foregroundTintMode](https://developer.android.com/reference/android/view/View.html#attr_android:foregroundTintMode) (https://developer.android.com/reference/android/view/View.html#attr_android:foregroundTintMode)

[android:hapticFeedbackEnabled](https://developer.android.com/reference/android/view/View.html#attr_android:hapticFeedbackEnabled) (https://developer.android.com/reference/android/view/View.html#attr_android:hapticFeedbackEnabled)

[android:id](https://developer.android.com/reference/android/view/View.html#attr_android:id) (https://developer.android.com/reference/android/view/View.html#attr_android:id)

[android:importantForAccessibility](https://developer.android.com/reference/android/view/View.html#attr_android:importantForAccessibility) (https://developer.android.com/reference/android/view/View.html#attr_android:importantForAccessibility)

[android:importantForAutofill](https://developer.android.com/reference/android/view/View.html#attr_android:importantForAutofill) (https://developer.android.com/reference/android/view/View.html#attr_android:importantForAutofill)

[android:isScrollContainer](https://developer.android.com/reference/android/view/View.html#attr_android:isScrollContainer) (https://developer.android.com/reference/android/view/View.html#attr_android:isScrollContainer)

[android:keepScreenOn](https://developer.android.com/reference/android/view/View.html#attr_android:keepScreenOn) (https://developer.android.com/reference/android/view/View.html#attr_android:keepScreenOn)

[android:keyboardNavigationCluster](https://developer.android.com/reference/android/view/View.html#attr_android:keyboardNavigationCluster) (https://developer.android.com/reference/android/view/View.html#attr_android:keyboardNavigationCluster)

[android:layerType](https://developer.android.com/reference/android/view/View.html#attr_android:layerType) (https://developer.android.com/reference/android/view/View.html#attr_android:layerType)

[android:layoutDirection](https://developer.android.com/reference/android/view/View.html#attr_android:layoutDirection) (https://developer.android.com/reference/android/view/View.html#attr_android:layoutDirection)

[android:longClickable](https://developer.android.com/reference/android/view/View.html#attr_android:longClickable) (https://developer.android.com/reference/android/view/View.html#attr_android:longClickable)

[android:minHeight](https://developer.android.com/reference/android/view/View.html#attr_android:minHeight) (https://developer.android.com/reference/android/view/View.html#attr_android:minHeight)

[android:minWidth](https://developer.android.com/reference/android/view/View.html#attr_android:minWidth) (https://developer.android.com/reference/android/view/View.html#attr_android:minWidth)

[android:nextClusterForward](https://developer.android.com/reference/android/view/View.html#attr_android:nextClusterForward) (https://developer.android.com/reference/android/view/View.html#attr_android:nextClusterForward)

[android:nextFocusDown](https://developer.android.com/reference/android/view/View.html#attr_android:nextFocusDown) (https://developer.android.com/reference/android/view/View.html#attr_android:nextFocusDown)

[android:nextFocusForward](https://developer.android.com/reference/android/view/View.html#attr_android:nextFocusForward) (https://developer.android.com/reference/android/view/View.html#attr_android:nextFocusForward)

[android:nextFocusLeft](https://developer.android.com/reference/android/view/View.html#attr_android:nextFocusLeft) (https://developer.android.com/reference/android/view/View.html#attr_android:nextFocusLeft)

[android:nextFocusRight](https://developer.android.com/reference/android/view/View.html#attr_android:nextFocusRight) (https://developer.android.com/reference/android/view/View.html#attr_android:nextFocusRight)

[android:nextFocusUp](https://developer.android.com/reference/android/view/View.html#attr_android:nextFocusUp) (https://developer.android.com/reference/android/view/View.html#attr_android:nextFocusUp)

[android:onClick](https://developer.android.com/reference/android/view/View.html#attr_android:onClick) (https://developer.android.com/reference/android/view/View.html#attr_android:onClick)

[android:outlineAmbientShadowColor](https://developer.android.com/reference/android/view/View.html#attr_android:outlineAmbientShadowColor) (https://developer.android.com/reference/android/view/View.html#attr_android:outlineAmbientShadowCol

[android:outlineSpotShadowColor](https://developer.android.com/reference/android/view/View.html#attr_android:outlineSpotShadowColor) (https://developer.android.com/reference/android/view/View.html#attr_android:outlineSpotShadowColor)

[android:padding](https://developer.android.com/reference/android/view/View.html#attr_android:padding) (https://developer.android.com/reference/android/view/View.html#attr_android:padding)

[android:paddingBottom](https://developer.android.com/reference/android/view/View.html#attr_android:paddingBottom) (https://developer.android.com/reference/android/view/View.html#attr_android:paddingBottom)

[android:paddingEnd](https://developer.android.com/reference/android/view/View.html#attr_android:paddingEnd) (https://developer.android.com/reference/android/view/View.html#attr_android:paddingEnd)

[android:paddingHorizontal](https://developer.android.com/reference/android/view/View.html#attr_android:paddingHorizontal) (https://developer.android.com/reference/android/view/View.html#attr_android:paddingHorizontal)

[android:paddingLeft](https://developer.android.com/reference/android/view/View.html#attr_android:paddingLeft) (https://developer.android.com/reference/android/view/View.html#attr_android:paddingLeft)

[android:paddingRight](https://developer.android.com/reference/android/view/View.html#attr_android:paddingRight) (https://developer.android.com/reference/android/view/View.html#attr_android:paddingRight)

[android:paddingStart](https://developer.android.com/reference/android/view/View.html#attr_android:paddingStart) (https://developer.android.com/reference/android/view/View.html#attr_android:paddingStart)

[android:paddingTop](https://developer.android.com/reference/android/view/View.html#attr_android:paddingTop) (https://developer.android.com/reference/android/view/View.html#attr_android:paddingTop)

[android:paddingVertical](https://developer.android.com/reference/android/view/View.html#attr_android:paddingVertical) (https://developer.android.com/reference/android/view/View.html#attr_android:paddingVertical)

[android:requiresFadingEdge](https://developer.android.com/reference/android/view/View.html#attr_android:requiresFadingEdge) (https://developer.android.com/reference/android/view/View.html#attr_android:requiresFadingEdge)

[android:rotation](https://developer.android.com/reference/android/view/View.html#attr_android:rotation) (https://developer.android.com/reference/android/view/View.html#attr_android:rotation)

[android:rotationX](https://developer.android.com/reference/android/view/View.html#attr_android:rotationX) (https://developer.android.com/reference/android/view/View.html#attr_android:rotationX)

[android:rotationY](https://developer.android.com/reference/android/view/View.html#attr_android:rotationY) (https://developer.android.com/reference/android/view/View.html#attr_android:rotationY)

[android:saveEnabled](https://developer.android.com/reference/android/view/View.html#attr_android:saveEnabled) (https://developer.android.com/reference/android/view/View.html#attr_android:saveEnabled)

[android:scaleX](https://developer.android.com/reference/android/view/View.html#attr_android:scaleX) (https://developer.android.com/reference/android/view/View.html#attr_android:scaleX)

[android:scaleY](https://developer.android.com/reference/android/view/View.html#attr_android:scaleY) (https://developer.android.com/reference/android/view/View.html#attr_android:scaleY)

[android:scrollIndicators](https://developer.android.com/reference/android/view/View.html#attr_android:scrollIndicators) (https://developer.android.com/reference/android/view/View.html#attr_android:scrollIndicators)

[android:scrollX](https://developer.android.com/reference/android/view/View.html#attr_android:scrollX) (https://developer.android.com/reference/android/view/View.html#attr_android:scrollX)

[android:scrollY](https://developer.android.com/reference/android/view/View.html#attr_android:scrollY) (https://developer.android.com/reference/android/view/View.html#attr_android:scrollY)

[android:scrollbarAlwaysDrawHorizontalTrack](https://developer.android.com/reference/android/view/View.html#attr_android:scrollbarAlwaysDrawHorizontalTrack) (https://developer.android.com/reference/android/view/View.html#attr_android:scrollbarAlwaysDrawHorizontalTrack)

[android:scrollbarAlwaysDrawVerticalTrack](https://developer.android.com/reference/android/view/View.html#attr_android:scrollbarAlwaysDrawVerticalTrack) (https://developer.android.com/reference/android/view/View.html#attr_android:scrollbarAlwaysDrawVerticalTrack)

[android:scrollbarDefaultDelayBeforeFade](https://developer.android.com/reference/android/view/View.html#attr_android:scrollbarDefaultDelayBeforeFade) (https://developer.android.com/reference/android/view/View.html#attr_android:scrollbarDefaultDelayBeforeFade)

[android:scrollbarFadeDuration](https://developer.android.com/reference/android/view/View.html#attr_android:scrollbarFadeDuration) (https://developer.android.com/reference/android/view/View.html#attr_android:scrollbarFadeDuration)

[android:scrollbarSize](https://developer.android.com/reference/android/view/View.html#attr_android:scrollbarSize) (https://developer.android.com/reference/android/view/View.html#attr_android:scrollbarSize)

[android:scrollbarStyle](https://developer.android.com/reference/android/view/View.html#attr_android:scrollbarStyle) (https://developer.android.com/reference/android/view/View.html#attr_android:scrollbarStyle)

[android:scrollbarThumbHorizontal](https://developer.android.com/reference/android/view/View.html#attr_android:scrollbarThumbHorizontal) (https://developer.android.com/reference/android/view/View.html#attr_android:scrollbarThumbHorizontal)

[android:scrollbarThumbVertical](https://developer.android.com/reference/android/view/View.html#attr_android:scrollbarThumbVertical) (https://developer.android.com/reference/android/view/View.html#attr_android:scrollbarThumbVertical)

[android:scrollbarTrackHorizontal](https://developer.android.com/reference/android/view/View.html#attr_android:scrollbarTrackHorizontal) (https://developer.android.com/reference/android/view/View.html#attr_android:scrollbarTrackHorizontal)

[android:scrollbarTrackVertical](https://developer.android.com/reference/android/view/View.html#attr_android:scrollbarTrackVertical) (https://developer.android.com/reference/android/view/View.html#attr_android:scrollbarTrackVertical)

[android:scrollbars](https://developer.android.com/reference/android/view/View.html#attr_android:scrollbars) (https://developer.android.com/reference/android/view/View.html#attr_android:scrollbars)

[android:soundEffectsEnabled](https://developer.android.com/reference/android/view/View.html#attr_android:soundEffectsEnabled) (https://developer.android.com/reference/android/view/View.html#attr_android:soundEffectsEnabled)

[android:stateListAnimator](https://developer.android.com/reference/android/view/View.html#attr_android:stateListAnimator) (https://developer.android.com/reference/android/view/View.html#attr_android:stateListAnimator)

[android:tag](https://developer.android.com/reference/android/view/View.html#attr_android:tag) (https://developer.android.com/reference/android/view/View.html#attr_android:tag)

[android:textAlignment](https://developer.android.com/reference/android/view/View.html#attr_android:textAlignment) (https://developer.android.com/reference/android/view/View.html#attr_android:textAlignment)

[android:textDirection](https://developer.android.com/reference/android/view/View.html#attr_android:textDirection) (https://developer.android.com/reference/android/view/View.html#attr_android:textDirection)

[android:theme](https://developer.android.com/reference/android/view/View.html#attr_android:theme) (https://developer.android.com/reference/android/view/View.html#attr_android:theme)

[android:tooltipText](https://developer.android.com/reference/android/view/View.html#attr_android:tooltipText) (https://developer.android.com/reference/android/view/View.html#attr_android:tooltipText)

[android:transformPivotX](https://developer.android.com/reference/android/view/View.html#attr_android:transformPivotX) (https://developer.android.com/reference/android/view/View.html#attr_android:transformPivotX)

[android:transformPivotY](https://developer.android.com/reference/android/view/View.html#attr_android:transformPivotY) (https://developer.android.com/reference/android/view/View.html#attr_android:transformPivotY)

[android:transitionName](https://developer.android.com/reference/android/view/View.html#attr_android:transitionName) (https://developer.android.com/reference/android/view/View.html#attr_android:transitionName)

[android:translationX](https://developer.android.com/reference/android/view/View.html#attr_android:translationX) (https://developer.android.com/reference/android/view/View.html#attr_android:translationX)

[android:translationY](https://developer.android.com/reference/android/view/View.html#attr_android:translationY) (https://developer.android.com/reference/android/view/View.html#attr_android:translationY)

[android:translationZ](https://developer.android.com/reference/android/view/View.html#attr_android:translationZ) (https://developer.android.com/reference/android/view/View.html#attr_android:translationZ)

[android:visibility](https://developer.android.com/reference/android/view/View.html#attr_android:visibility) (https://developer.android.com/reference/android/view/View.html#attr_android:visibility)

Inherited constants

From class [android.view.View](https://developer.android.com/reference/android/view/View.html) (https://developer.android.com/reference/android/view/View.html)

int

ACCESSIBILITY_LIVE_REGION_ASSERTIVE

(https://developer.android.com/reference/android/view/View.html#ACCESSIBILITY_LIVE_REGION_ASSERTIVE)

Live region mode specifying that accessibility services should interrupt ongoing speech to immediately announce changes to this view.

int

ACCESSIBILITY_LIVE_REGION_NONE

(https://developer.android.com/reference/android/view/View.html#ACCESSIBILITY_LIVE_REGION_NONE)

Live region mode specifying that accessibility services should not automatically announce changes to this view.

int

ACCESSIBILITY_LIVE_REGION_POLITE

(https://developer.android.com/reference/android/view/View.html#ACCESSIBILITY_LIVE_REGION_POLITE)

Live region mode specifying that accessibility services should announce changes to this view.

int

AUTOFILL_FLAG_INCLUDE_NOT_IMPORTANT_VIEWS

(https://developer.android.com/reference/android/view/View.html#AUTOFILL_FLAG_INCLUDE_NOT_IMPORTANT_VIEWS)

Flag requesting you to add views that are marked as not important for autofill (see [setImportantForAutofill\(int\)](#)). ([https://developer.android.com/reference/android/view/View.html#setImportantForAutofill\(int\)](https://developer.android.com/reference/android/view/View.html#setImportantForAutofill(int)))) to a [ViewStructure](#) (<https://developer.android.com/reference/android/view/ViewStructure.html>).

String (<https://developer.android.com/reference/java/lang/String.html>) **AUTOFILL_HINT_CREDIT_CARD_EXPIRATION_DATE** (https://developer.android.com/reference/android/view/View.html#AUTOFILL_HINT_CREDIT_CARD_EXPIRATION_DATE)

Hint indicating that this view can be autofilled with a credit card expiration date.

String (<https://developer.android.com/reference/java/lang/String.html>) **AUTOFILL_HINT_CREDIT_CARD_EXPIRATION_DAY**
(https://developer.android.com/reference/android/view/View.html#AUTOFILL_HINT_CREDIT_CARD_EXPIRATION_DAY)

Hint indicating that this view can be autofilled with a credit card expiration day.

String (<https://developer.android.com/reference/java/lang/String.html>) **AUTOFILL_HINT_CREDIT_CARD_EXPIRATION_MONTH**
(https://developer.android.com/reference/android/view/View.html#AUTOFILL_HINT_CREDIT_CARD_EXPIRATION_MONTH)

Hint indicating that this view can be autofilled with a credit card expiration month.

String (<https://developer.android.com/reference/java/lang/String.html>) **AUTOFILL_HINT_CREDIT_CARD_EXPIRATION_YEAR**
(https://developer.android.com/reference/android/view/View.html#AUTOFILL_HINT_CREDIT_CARD_EXPIRATION_YEAR)

Hint indicating that this view can be autofilled with a credit card expiration year.

String (<https://developer.android.com/reference/java/lang/String.html>) **AUTOFILL_HINT_CREDIT_CARD_NUMBER**
(https://developer.android.com/reference/android/view/View.html#AUTOFILL_HINT_CREDIT_CARD_NUMBER)

Hint indicating that this view can be autofilled with a credit card number.

String (<https://developer.android.com/reference/java/lang/String.html>) **AUTOFILL_HINT_CREDIT_CARD_SECURITY_CODE**
(https://developer.android.com/reference/android/view/View.html#AUTOFILL_HINT_CREDIT_CARD_SECURITY_CODE)

Hint indicating that this view can be autofilled with a credit card security code.

String (<https://developer.android.com/reference/java/lang/String.html>) **AUTOFILL_HINT_EMAIL_ADDRESS**
(https://developer.android.com/reference/android/view/View.html#AUTOFILL_HINT_EMAIL_ADDRESS)

Hint indicating that this view can be autofilled with an email address.

String (<https://developer.android.com/reference/java/lang/String.html>) **AUTOFILL_HINT_NAME**
(https://developer.android.com/reference/android/view/View.html#AUTOFILL_HINT_NAME)

Hint indicating that this view can be autofilled with a user's real name.

String (<https://developer.android.com/reference/java/lang/String.html>) **AUTOFILL_HINT_PASSWORD**
(https://developer.android.com/reference/android/view/View.html#AUTOFILL_HINT_PASSWORD)

Hint indicating that this view can be autofilled with a password.

String (<https://developer.android.com/reference/java/lang/String.html>) **AUTOFILL_HINT_PHONE**
(https://developer.android.com/reference/android/view/View.html#AUTOFILL_HINT_PHONE)

Hint indicating that this view can be autofilled with a phone number.

String (<https://developer.android.com/reference/java/lang/String.html>) **AUTOFILL_HINT_POSTAL_ADDRESS**
(https://developer.android.com/reference/android/view/View.html#AUTOFILL_HINT_POSTAL_ADDRESS)

Hint indicating that this view can be autofilled with a postal address.

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

(https://developer.android.com/reference/android/view/View.html#AUTOFILL_HINT_POSTAL_CODE)

Hint indicating that this view can be autofilled with a postal code.

String (<https://developer.android.com/reference/java/lang/String.html>) **AUTOFILL_HINT_USERNAME**
(https://developer.android.com/reference/android/view/View.html#AUTOFILL_HINT_USERNAME)

Hint indicating that this view can be autofilled with a username.

int **AUTOFILL_TYPE_DATE**
(https://developer.android.com/reference/android/view/View.html#AUTOFILL_TYPE_DATE)

Autofill type for a field that contains a date, which is represented by a long representing the number of milliseconds since the standard base time known as "the epoch", namely January 1, 1970, 00:00:00 GMT (see **Date.getTime()**).
([https://developer.android.com/reference/java/util/Date.html#getTime\(\)](https://developer.android.com/reference/java/util/Date.html#getTime())).

int **AUTOFILL_TYPE_LIST**
(https://developer.android.com/reference/android/view/View.html#AUTOFILL_TYPE_LIST)

Autofill type for a selection list field, which is filled by an **int** representing the element index inside the list (starting at 0).

int **AUTOFILL_TYPE_NONE**
(https://developer.android.com/reference/android/view/View.html#AUTOFILL_TYPE_NONE)

Autofill type for views that cannot be autofilled.

int

AUTOFILL_TYPE_TEXT

(https://developer.android.com/reference/android/view/View.html#AUTOFILL_TYPE_TEXT)

Autofill type for a text field, which is filled by a **CharSequence**

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

int

AUTOFILL_TYPE_TOGGLE

(https://developer.android.com/reference/android/view/View.html#AUTOFILL_TYPE_TOGGLE)

Autofill type for a toggleable field, which is filled by a **boolean**.

int

DRAW_FLAG_GLOBAL

(https://developer.android.com/reference/android/view/View.html#DRAW_FLAG_GLOBAL)

Flag indicating that a drag can cross window boundaries.

int

DRAW_FLAG_GLOBAL_PERSISTABLE_URI_PERMISSION

(https://developer.android.com/reference/android/view/View.html#DRAW_FLAG_GLOBAL_PERSISTABLE_URI_PERMISSION)

When this flag is used with **DRAW_FLAG_GLOBAL_URI_READ**

(https://developer.android.com/reference/android/view/View.html#DRAW_FLAG_GLOBAL_URI_READ)

and/or **DRAW_FLAG_GLOBAL_URI_WRITE**

(https://developer.android.com/reference/android/view/View.html#DRAW_FLAG_GLOBAL_URI_WRITE)

, the URI permission grant can be persisted across device reboots until explicitly revoked with **Context.revokeUriPermission(Uri, int)**.

([https://developer.android.com/reference/android/content/Context.html#revokeUriPermission\(android.net.Uri,%20int\)](https://developer.android.com/reference/android/content/Context.html#revokeUriPermission(android.net.Uri,%20int)))
Context.revokeUriPermission}.

int

DRAG_FLAG_GLOBAL_PREFIX_URI_PERMISSION

(https://developer.android.com/reference/android/view/View.html#DRAG_FLAG_GLOBAL_PREFIX_URI_PERMISSION)

When this flag is used with **DRAG_FLAG_GLOBAL_URI_READ**

(https://developer.android.com/reference/android/view/View.html#DRAG_FLAG_GLOBAL_URI_READ)

and/or **DRAG_FLAG_GLOBAL_URI_WRITE**

(https://developer.android.com/reference/android/view/View.html#DRAG_FLAG_GLOBAL_URI_WRITE)

, the URI permission grant applies to any URI that is a prefix match against the original granted URI.

int

DRAG_FLAG_GLOBAL_URI_READ

(https://developer.android.com/reference/android/view/View.html#DRAG_FLAG_GLOBAL_URI_READ)

When this flag is used with **DRAG_FLAG_GLOBAL**

(https://developer.android.com/reference/android/view/View.html#DRAG_FLAG_GLOBAL)

, the drag recipient will be able to request read access to the content URI(s) contained in the **ClipData**

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

int

DRAG_FLAG_GLOBAL_URI_WRITE

(https://developer.android.com/reference/android/view/View.html#DRAG_FLAG_GLOBAL_URI_WRITE)

When this flag is used with **DRAG_FLAG_GLOBAL**
(https://developer.android.com/reference/android/view/View.html#DRAG_FLAG_GLOBAL)
, the drag recipient will be able to request write access to the content URI(s) contained in the **ClipData**
(<https://developer.android.com/reference/android/content/ClipData.html>) object.

int

DRAG_FLAG_OPAQUE
(https://developer.android.com/reference/android/view/View.html#DRAG_FLAG_OPAQUE)

Flag indicating that the drag shadow will be opaque.

int

DRAWING_CACHE_QUALITY_AUTO
(https://developer.android.com/reference/android/view/View.html#DRAWING_CACHE_QUALITY_AUTO)

*This constant was deprecated in API level 28. The view drawing cache was largely made obsolete with the introduction of hardware-accelerated rendering in API 11. With hardware-acceleration, intermediate cache layers are largely unnecessary and can easily result in a net loss in performance due to the cost of creating and updating the layer. In the rare cases where caching layers are useful, such as for alpha animations, **setLayerType(int, Paint)**.*

([https://developer.android.com/reference/android/view/View.html#setLayerType\(int,%20android.graphics.Paint\)](https://developer.android.com/reference/android/view/View.html#setLayerType(int,%20android.graphics.Paint)))

*handles this with hardware rendering. For software-rendered snapshots of a small part of the View hierarchy or individual Views it is recommended to create a **Canvas***

(<https://developer.android.com/reference/android/graphics/Canvas.html>)

*from either a **Bitmap***

(<https://developer.android.com/reference/android/graphics/Bitmap.html>) or

Picture

(<https://developer.android.com/reference/android/graphics/Picture.html>) and call **draw(Canvas)**.

([https://developer.android.com/reference/android/view/View.html#draw\(android.graphics.Canvas\)](https://developer.android.com/reference/android/view/View.html#draw(android.graphics.Canvas)))

on the View. However these software-rendered usages are discouraged and have compatibility issues with hardware-only rendering features such as

Config.HARDWARE

(<https://developer.android.com/reference/android/graphics/Bitmap.Config.html#HARDWARE>)

bitmaps, real-time shadows, and outline clipping. For screenshots of the UI for feedback reports or unit testing the **PixelCopy**.

(<https://developer.android.com/reference/android/view/PixelCopy.html>) API is recommended.

int

DRAWING_CACHE_QUALITY_HIGH

(https://developer.android.com/reference/android/view/View.html#DRAWING_CACHE_QUALITY_HIGH)

This constant was deprecated in API level 28. The view drawing cache was largely made obsolete with the introduction of hardware-accelerated rendering in API 11. With hardware-acceleration, intermediate cache layers are largely unnecessary and can easily result in a net loss in performance due to the cost of creating and updating the layer. In the rare cases where caching layers are useful, such as for alpha animations, **setLayerType(int, Paint)**.

([https://developer.android.com/reference/android/view/View.html#setLayerType\(int,%20android.graphics.Paint\)](https://developer.android.com/reference/android/view/View.html#setLayerType(int,%20android.graphics.Paint)))

handles this with hardware rendering. For software-rendered snapshots of a small part of the View hierarchy or individual Views it is recommended to create a **Canvas**

(<https://developer.android.com/reference/android/graphics/Canvas.html>)

from either a **Bitmap**

(<https://developer.android.com/reference/android/graphics/Bitmap.html>) or

Picture

(<https://developer.android.com/reference/android/graphics/Picture.html>) and call **draw(Canvas)**.

([https://developer.android.com/reference/android/view/View.html#draw\(android.graphics.Canvas\)](https://developer.android.com/reference/android/view/View.html#draw(android.graphics.Canvas)))

on the View. However these software-rendered usages are discouraged and have compatibility issues with hardware-only rendering features such as

Config.HARDWARE

(<https://developer.android.com/reference/android/graphics/Bitmap.Config.html#HARDWARE>)

bitmaps, real-time shadows, and outline clipping. For screenshots of the UI for feedback reports or unit testing the **PixelCopy**.

(<https://developer.android.com/reference/android/view/PixelCopy.html>) API is recommended.

int

DRAWING_CACHE_QUALITY_LOW

(https://developer.android.com/reference/android/view/View.html#DRAWING_CACHE_QUALITY_LOW)

This constant was deprecated in API level 28. The view drawing cache was largely made obsolete with the introduction of hardware-accelerated rendering in API 11. With hardware-acceleration, intermediate cache layers are largely unnecessary and can easily result in a net loss in performance due to the cost of creating and updating the layer. In the rare cases where caching layers are useful, such as for alpha animations, **setLayerType(int, Paint)**.

([https://developer.android.com/reference/android/view/View.html#setLayerType\(int,%20android.graphics.Paint\)](https://developer.android.com/reference/android/view/View.html#setLayerType(int,%20android.graphics.Paint)))

handles this with hardware rendering. For software-rendered snapshots of a small part of the View hierarchy or individual Views it is recommended to create a **Canvas**

(<https://developer.android.com/reference/android/graphics/Canvas.html>) from either a **Bitmap**

(<https://developer.android.com/reference/android/graphics/Bitmap.html>) or **Picture** (<https://developer.android.com/reference/android/graphics/Picture.html>) and call **draw(Canvas)**. ([https://developer.android.com/reference/android/view/View.html#draw\(android.graphics.Canvas\)](https://developer.android.com/reference/android/view/View.html#draw(android.graphics.Canvas))) on the View. However these software-rendered usages are discouraged and have compatibility issues with hardware-only rendering features such as **Config.HARDWARE** (<https://developer.android.com/reference/android/graphics/Bitmap.Config.html#HARDWARE>) bitmaps, real-time shadows, and outline clipping. For screenshots of the UI for feedback reports or unit testing the **PixelCopy** (<https://developer.android.com/reference/android/view/PixelCopy.html>) API is recommended.

int

FIND_VIEWS_WITH_CONTENT_DESCRIPTION

(https://developer.android.com/reference/android/view/View.html#FIND_VIEWS_WITH_CONTENT_DESCRIPTION)

Find find views that contain the specified content description.

int

FIND_VIEWS_WITH_TEXT

(https://developer.android.com/reference/android/view/View.html#FIND_VIEWS_WITH_TEXT)

Find views that render the specified text.

int

FOCUSABLE

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

This view wants keystrokes.

int

FOCUSABLES_ALL

(https://developer.android.com/reference/android/view/View.html#FOCUSABLES_ALL)

View flag indicating whether **addFocusables(ArrayList, int, int)**.
([https://developer.android.com/reference/android/view/View.html#addFocusables\(java.util.ArrayList<android.view.View>,%20int,%20int\)](https://developer.android.com/reference/android/view/View.html#addFocusables(java.util.ArrayList<android.view.View>,%20int,%20int)))
should add all focusable Views regardless if they are focusable in touch mode.

int

FOCUSABLES_TOUCH_MODE

(https://developer.android.com/reference/android/view/View.html#FOCUSABLES_TOUCH_MODE)

View flag indicating whether **addFocusables(ArrayList, int, int)**.
([https://developer.android.com/reference/android/view/View.html#addFocusables\(java.util.ArrayList<android.view.View>,%20int,%20int\)](https://developer.android.com/reference/android/view/View.html#addFocusables(java.util.ArrayList<android.view.View>,%20int,%20int)))
should add only Views focusable in touch mode.

int

FOCUSABLE_AUTO

(https://developer.android.com/reference/android/view/View.html#FOCUSABLE_AUTO)

This view determines focusability automatically.

int

FOCUS_BACKWARD

(https://developer.android.com/reference/android/view/View.html#FOCUS_BACKWARD)

Use with **focusSearch(int)**.
([https://developer.android.com/reference/android/view/View.html#focusSearch\(int\)](https://developer.android.com/reference/android/view/View.html#focusSearch(int)))

.

int

FOCUS_DOWN

(https://developer.android.com/reference/android/view/View.html#FOCUS_DOWN)

Use with **focusSearch(int)**.

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

.

int

FOCUS_FORWARD

(https://developer.android.com/reference/android/view/View.html#FOCUS_FORWARD)

Use with **focusSearch(int)**.

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

.

int

FOCUS_LEFT

(https://developer.android.com/reference/android/view/View.html#FOCUS_LEFT)

Use with **focusSearch(int)**.

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

.

int

FOCUS_RIGHT

(https://developer.android.com/reference/android/view/View.html#FOCUS_RIGHT)

	<p>Use with <u>focusSearch(int)</u>. (https://developer.android.com/reference/android/view/View.html#focusSearch(int))</p> <p>.</p>
int	<p><u>FOCUS_UP</u> (https://developer.android.com/reference/android/view/View.html#FOCUS_UP)</p> <p>Use with <u>focusSearch(int)</u>. (https://developer.android.com/reference/android/view/View.html#focusSearch(int))</p> <p>.</p>
int	<p><u>GONE</u> (https://developer.android.com/reference/android/view/View.html#GONE)</p> <p>This view is invisible, and it doesn't take any space for layout purposes.</p>
int	<p><u>HAPTIC_FEEDBACK_ENABLED</u> (https://developer.android.com/reference/android/view/View.html#HAPTIC_FEEDBACK_ENABLED)</p> <p>View flag indicating whether this view should have haptic feedback enabled for events such as long presses.</p>
int	<p><u>IMPORTANT_FOR_ACCESSIBILITY_AUTO</u> (https://developer.android.com/reference/android/view/View.html#IMPORTANT_FOR_ACCESSIBILITY_AUTO)</p> <p>Automatically determine whether a view is important for accessibility.</p>
int	<p><u>IMPORTANT_FOR_ACCESSIBILITY_NO</u></p>

	<p>(https://developer.android.com/reference/android/view/View.html#IMPORTANT_FOR_ACCESSIBILITY_NO)</p> <p>The view is not important for accessibility.</p>
int	<p><u>IMPORTANT_FOR_ACCESSIBILITY_NO_HIDE_DESCENDANTS</u> (https://developer.android.com/reference/android/view/View.html#IMPORTANT_FOR_ACCESSIBILITY_NO_HIDE_DESCENDANTS)</p> <p>The view is not important for accessibility, nor are any of its descendant views.</p>
int	<p><u>IMPORTANT_FOR_ACCESSIBILITY_YES</u> (https://developer.android.com/reference/android/view/View.html#IMPORTANT_FOR_ACCESSIBILITY_YES)</p> <p>The view is important for accessibility.</p>
int	<p><u>IMPORTANT_FOR_AUTOFILL_AUTO</u> (https://developer.android.com/reference/android/view/View.html#IMPORTANT_FOR_AUTOFILL_AUTO)</p> <p>Automatically determine whether a view is important for autofill.</p>
int	<p><u>IMPORTANT_FOR_AUTOFILL_NO</u> (https://developer.android.com/reference/android/view/View.html#IMPORTANT_FOR_AUTOFILL_NO)</p> <p>The view is not important for autofill, but its children (if any) will be traversed.</p>
int	<p><u>IMPORTANT_FOR_AUTOFILL_NO_EXCLUDE_DESCENDANTS</u> (https://developer.android.com/reference/android/view/View.html#IMPORTANT_FOR_AUTOFILL_NO_EXCLUDE_DESCENDANTS)</p>

	<p>The view is not important for autofill, and its children (if any) will not be traversed.</p>
int	<p><u>IMPORTANT_FOR_AUTOFILL_YES</u> https://developer.android.com/reference/android/view/View.html#IMPORTANT_FOR_AUTOFILL_YES)</p> <p>The view is important for autofill, and its children (if any) will be traversed.</p>
int	<p><u>IMPORTANT_FOR_AUTOFILL_YES_EXCLUDE_DESCENDANTS</u> https://developer.android.com/reference/android/view/View.html#IMPORTANT_FOR_AUTOFILL_YES_EXCLUDE_DESCENDANTS)</p> <p>The view is important for autofill, but its children (if any) will not be traversed.</p>
int	<p><u>INVISIBLE</u> https://developer.android.com/reference/android/view/View.html#INVISIBLE)</p> <p>This view is invisible, but it still takes up space for layout purposes.</p>
int	<p><u>KEEP_SCREEN_ON</u> https://developer.android.com/reference/android/view/View.html#KEEP_SCREEN_ON)</p> <p>View flag indicating that the screen should remain on while the window containing this view is visible to the user.</p>
int	<p><u>LAYER_TYPE_HARDWARE</u> https://developer.android.com/reference/android/view/View.html#LAYER_TYPE_HARDWARE)</p> <p>Indicates that the view has a hardware layer.</p>

int	<u>LAYER_TYPE_NONE</u> (https://developer.android.com/reference/android/view/View.html#LAYER_TYPE_NONE) Indicates that the view does not have a layer.
int	<u>LAYER_TYPE_SOFTWARE</u> (https://developer.android.com/reference/android/view/View.html#LAYER_TYPE_SOFTWARE) Indicates that the view has a software layer.
int	<u>LAYOUT_DIRECTION_INHERIT</u> (https://developer.android.com/reference/android/view/View.html#LAYOUT_DIRECTION_INHERIT) Horizontal layout direction of this view is inherited from its parent.
int	<u>LAYOUT_DIRECTION_LOCALE</u> (https://developer.android.com/reference/android/view/View.html#LAYOUT_DIRECTION_LOCALE) Horizontal layout direction of this view is from deduced from the default language script for the locale.
int	<u>LAYOUT_DIRECTION_LTR</u> (https://developer.android.com/reference/android/view/View.html#LAYOUT_DIRECTION_LTR) Horizontal layout direction of this view is from Left to Right.
int	<u>LAYOUT_DIRECTION_RTL</u> (https://developer.android.com/reference/android/view/View.html#LAYOUT_DIRECTION_RTL)

Horizontal layout direction of this view is from Right to Left.

int

MEASURED_HEIGHT_STATE_SHIFT

(https://developer.android.com/reference/android/view/View.html#MEASURED_HEIGHT_STATE_SHIFT)

Bit shift of **MEASURED_STATE_MASK**

(https://developer.android.com/reference/android/view/View.html#MEASURED_STATE_MASK)

to get to the height bits for functions that combine both width and height into a single int, such as **getMeasuredState()**.

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

and the childState argument of **resolveSizeAndState(int, int, int)**.

([https://developer.android.com/reference/android/view/View.html#resolveSizeAndState\(int,%20int,%20int\)](https://developer.android.com/reference/android/view/View.html#resolveSizeAndState(int,%20int,%20int)))

.

int

MEASURED_SIZE_MASK

(https://developer.android.com/reference/android/view/View.html#MEASURED_SIZE_MASK)

Bits of **getMeasuredWidthAndState()**.

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

and **getMeasuredWidthAndState()**.

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

that provide the actual measured size.

int

MEASURED_STATE_MASK

(https://developer.android.com/reference/android/view/View.html#MEASURED_STATE_MASK)

Bits of getMeasuredWidthAndState().

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

and getMeasuredWidthAndState().

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

that provide the additional state bits.

int

MEASURED_STATE_TOO_SMALL

(https://developer.android.com/reference/android/view/View.html#MEASURED_STATE_TOO_SMALL)

Bit of getMeasuredWidthAndState().

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

and getMeasuredWidthAndState().

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

that indicates the measured size is smaller than the space the view would like to have.

int

NOT_FOCUSABLE

(https://developer.android.com/reference/android/view/View.html#NOT_FOCUSABLE)

This view does not want keystrokes.

int

NO_ID

(https://developer.android.com/reference/android/view/View.html#NO_ID)

Used to mark a View that has no ID.

int

OVER_SCROLL_ALWAYS

	<p>(https://developer.android.com/reference/android/view/View.html#OVER_SCROLL_ALWAYS)</p> <p>Always allow a user to over-scroll this view, provided it is a view that can scroll.</p>
int	<p><u>OVER_SCROLL_IF_CONTENT_SCROLLS</u> (https://developer.android.com/reference/android/view/View.html#OVER_SCROLL_IF_CONTENT_SCROLLS)</p> <p>Allow a user to over-scroll this view only if the content is large enough to meaningfully scroll, provided it is a view that can scroll.</p>
int	<p><u>OVER_SCROLL_NEVER</u> (https://developer.android.com/reference/android/view/View.html#OVER_SCROLL_NEVER)</p> <p>Never allow a user to over-scroll this view.</p>
int	<p><u>SCREEN_STATE_OFF</u> (https://developer.android.com/reference/android/view/View.html#SCREEN_STATE_OFF)</p> <p>Indicates that the screen has changed state and is now off.</p>
int	<p><u>SCREEN_STATE_ON</u> (https://developer.android.com/reference/android/view/View.html#SCREEN_STATE_ON)</p> <p>Indicates that the screen has changed state and is now on.</p>
int	<p><u>SCROLLBARS_INSIDE_INSET</u> (https://developer.android.com/reference/android/view/View.html#SCROLLBARS_INSIDE_INSET)</p>

	The scrollbar style to display the scrollbars inside the padded area, increasing the padding of the view.
int	<u>SCROLLBARS_INSIDE_OVERLAY</u> (https://developer.android.com/reference/android/view/View.html#SCROLLBARS_INSIDE_OVERLAY) The scrollbar style to display the scrollbars inside the content area, without increasing the padding.
int	<u>SCROLLBARS_OUTSIDE_INSET</u> (https://developer.android.com/reference/android/view/View.html#SCROLLBARS_OUTSIDE_INSET) The scrollbar style to display the scrollbars at the edge of the view, increasing the padding of the view.
int	<u>SCROLLBARS_OUTSIDE_OVERLAY</u> (https://developer.android.com/reference/android/view/View.html#SCROLLBARS_OUTSIDE_OVERLAY) The scrollbar style to display the scrollbars at the edge of the view, without increasing the padding.
int	<u>SCROLLBAR_POSITION_DEFAULT</u> (https://developer.android.com/reference/android/view/View.html#SCROLLBAR_POSITION_DEFAULT) Position the scroll bar at the default position as determined by the system.
int	<u>SCROLLBAR_POSITION_LEFT</u> (https://developer.android.com/reference/android/view/View.html#SCROLLBAR_POSITION_LEFT)

	Position the scroll bar along the left edge.
int	<u>SCROLLBAR_POSITION_RIGHT</u> (https://developer.android.com/reference/android/view/View.html#SCROLLBAR_POSITION_RIGHT) Position the scroll bar along the right edge.
int	<u>SCROLL_AXIS_HORIZONTAL</u> (https://developer.android.com/reference/android/view/View.html#SCROLL_AXIS_HORIZONTAL) Indicates scrolling along the horizontal axis.
int	<u>SCROLL_AXIS_NONE</u> (https://developer.android.com/reference/android/view/View.html#SCROLL_AXIS_NONE) Indicates no axis of view scrolling.
int	<u>SCROLL_AXIS_VERTICAL</u> (https://developer.android.com/reference/android/view/View.html#SCROLL_AXIS_VERTICAL) Indicates scrolling along the vertical axis.
int	<u>SCROLL_INDICATOR_BOTTOM</u> (https://developer.android.com/reference/android/view/View.html#SCROLL_INDICATOR_BOTTOM) Scroll indicator direction for the bottom edge of the view.
int	<u>SCROLL_INDICATOR_END</u>

	<p>(https://developer.android.com/reference/android/view/View.html#SCROLL_INDICATOR_END)</p> <p>Scroll indicator direction for the ending edge of the view.</p>
int	<p><u>SCROLL_INDICATOR_LEFT</u> (https://developer.android.com/reference/android/view/View.html#SCROLL_INDICATOR_LEFT)</p> <p>Scroll indicator direction for the left edge of the view.</p>
int	<p><u>SCROLL_INDICATOR_RIGHT</u> (https://developer.android.com/reference/android/view/View.html#SCROLL_INDICATOR_RIGHT)</p> <p>Scroll indicator direction for the right edge of the view.</p>
int	<p><u>SCROLL_INDICATOR_START</u> (https://developer.android.com/reference/android/view/View.html#SCROLL_INDICATOR_START)</p> <p>Scroll indicator direction for the starting edge of the view.</p>
int	<p><u>SCROLL_INDICATOR_TOP</u> (https://developer.android.com/reference/android/view/View.html#SCROLL_INDICATOR_TOP)</p> <p>Scroll indicator direction for the top edge of the view.</p>
int	<p><u>SOUND_EFFECTS_ENABLED</u> (https://developer.android.com/reference/android/view/View.html#SOUND_EFFECTS_ENABLED)</p>

View flag indicating whether this view should have sound effects enabled for events such as clicking and touching.

int

STATUS_BAR_HIDDEN

(https://developer.android.com/reference/android/view/View.html#STATUS_BAR_HIDDEN)

This constant was deprecated in API level 14. Use

SYSTEM_UI_FLAG_LOW_PROFILE

(https://developer.android.com/reference/android/view/View.html#SYSTEM_UI_FLAG_LOW_PROFILE)

instead.

int

STATUS_BAR_VISIBLE

(https://developer.android.com/reference/android/view/View.html#STATUS_BAR_VISIBLE)

This constant was deprecated in API level 14. Use

SYSTEM_UI_FLAG_VISIBLE

(https://developer.android.com/reference/android/view/View.html#SYSTEM_UI_FLAG_VISIBLE)

instead.

int

SYSTEM_UI_FLAG_FULLSCREEN

(https://developer.android.com/reference/android/view/View.html#SYSTEM_UI_FLAG_FULLSCREEN)

Flag for **setSystemUiVisibility(int)**.

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

: View has requested to go into the normal fullscreen mode so that its content can take over the screen while still allowing the user to interact with the application.

int

SYSTEM_UI_FLAG_HIDE_NAVIGATION

(https://developer.android.com/reference/android/view/View.html#SYSTEM_UI_FLAG_HIDE_NAVIGATION)

Flag for **setSystemUiVisibility(int)**.

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

: View has requested that the system navigation be temporarily hidden.

int

SYSTEM_UI_FLAG_IMMERSIVE

(https://developer.android.com/reference/android/view/View.html#SYSTEM_UI_FLAG_IMMERSIVE)

Flag for **setSystemUiVisibility(int)**.

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

: View would like to remain interactive when hiding the navigation bar with

SYSTEM_UI_FLAG_HIDE_NAVIGATION

(https://developer.android.com/reference/android/view/View.html#SYSTEM_UI_FLAG_HIDE_NAVIGATION)

.

int

SYSTEM_UI_FLAG_IMMERSIVE_STICKY

(https://developer.android.com/reference/android/view/View.html#SYSTEM_UI_FLAG_IMMERSIVE_STICKY)

Flag for **setSystemUiVisibility(int)**.

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

: View would like to remain interactive when hiding the status bar with

SYSTEM_UI_FLAG_FULLSCREEN

(https://developer.android.com/reference/android/view/View.html#SYSTEM_UI_FLAG_FULLSCREEN)

and/or hiding the navigation bar with **SYSTEM_UI_FLAG_HIDE_NAVIGATION**
(https://developer.android.com/reference/android/view/View.html#SYSTEM_UI_FLAG_HIDE_NAVIGATION)

.

int

SYSTEM_UI_FLAG_LAYOUT_FULLSCREEN

(https://developer.android.com/reference/android/view/View.html#SYSTEM_UI_FLAG_LAYOUT_FULLSCREEN)

Flag for **setSystemUiVisibility(int)**.

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

: View would like its window to be laid out as if it has requested

SYSTEM_UI_FLAG_FULLSCREEN

(https://developer.android.com/reference/android/view/View.html#SYSTEM_UI_FLAG_FULLSCREEN)

, even if it currently hasn't.

int

SYSTEM_UI_FLAG_LAYOUT_HIDE_NAVIGATION

(https://developer.android.com/reference/android/view/View.html#SYSTEM_UI_FLAG_LAYOUT_HIDE_NAVIGATION)

Flag for **setSystemUiVisibility(int)**.

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

: View would like its window to be laid out as if it has requested

SYSTEM_UI_FLAG_HIDE_NAVIGATION

(https://developer.android.com/reference/android/view/View.html#SYSTEM_UI_FLAG_HIDE_NAVIGATION)

, even if it currently hasn't.

int

SYSTEM_UI_FLAG_LAYOUT_STABLE

(https://developer.android.com/reference/android/view/View.html#SYSTEM_UI_FLAG_LAYOUT_STABLE)

Flag for [setSystemUiVisibility\(int\)](#).

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

: When using other layout flags, we would like a stable view of the content insets given to [fitSystemWindows\(Rect\)](#).

([https://developer.android.com/reference/android/view/View.html#fitSystemWindows\(android.graphics.Rect\)](https://developer.android.com/reference/android/view/View.html#fitSystemWindows(android.graphics.Rect)))

.

int

SYSTEM_UI_FLAG_LIGHT_NAVIGATION_BAR

(https://developer.android.com/reference/android/view/View.html#SYSTEM_UI_FLAG_LIGHT_NAVIGATION_BAR)

Flag for [setSystemUiVisibility\(int\)](#).

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

: Requests the navigation bar to draw in a mode that is compatible with light navigation bar backgrounds.

int

SYSTEM_UI_FLAG_LIGHT_STATUS_BAR

(https://developer.android.com/reference/android/view/View.html#SYSTEM_UI_FLAG_LIGHT_STATUS_BAR)

Flag for [setSystemUiVisibility\(int\)](#).

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

: Requests the status bar to draw in a mode that is compatible with light status bar backgrounds.

int

SYSTEM_UI_FLAG_LOW_PROFILE

(https://developer.android.com/reference/android/view/View.html#SYSTEM_UI_FLAG_LOW_PROFILE)

Flag for **setSystemUiVisibility(int)**.

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

: View has requested the system UI to enter an unobtrusive "low profile" mode.

int

SYSTEM_UI_FLAG_VISIBLE

(https://developer.android.com/reference/android/view/View.html#SYSTEM_UI_FLAG_VISIBLE)

Special constant for **setSystemUiVisibility(int)**.

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

: View has requested the system UI (status bar) to be visible (the default).

int

SYSTEM_UI_LAYOUT_FLAGS

(https://developer.android.com/reference/android/view/View.html#SYSTEM_UI_LAYOUT_FLAGS)

Flags that can impact the layout in relation to system UI.

int

TEXT_ALIGNMENT_CENTER

(https://developer.android.com/reference/android/view/View.html#TEXT_ALIGNMENT_CENTER)

Center the paragraph, e.g.

int

TEXT_ALIGNMENT_GRAVITY

(https://developer.android.com/reference/android/view/View.html#TEXT_ALIGNMENT_GRAVITY)

Default for the root view.

int	<u>TEXT_ALIGNMENT_INHERIT</u> (https://developer.android.com/reference/android/view/View.html#TEXT_ALIGNMENT_INHERIT) Default text alignment.
int	<u>TEXT_ALIGNMENT_TEXT_END</u> (https://developer.android.com/reference/android/view/View.html#TEXT_ALIGNMENT_TEXT_END) Align to the end of the paragraph, e.g.
int	<u>TEXT_ALIGNMENT_TEXT_START</u> (https://developer.android.com/reference/android/view/View.html#TEXT_ALIGNMENT_TEXT_START) Align to the start of the paragraph, e.g.
int	<u>TEXT_ALIGNMENT_VIEW_END</u> (https://developer.android.com/reference/android/view/View.html#TEXT_ALIGNMENT_VIEW_END) Align to the end of the view, which is ALIGN_RIGHT if the view's resolved layoutDirection is LTR, and ALIGN_LEFT otherwise.
int	<u>TEXT_ALIGNMENT_VIEW_START</u> (https://developer.android.com/reference/android/view/View.html#TEXT_ALIGNMENT_VIEW_START) Align to the start of the view, which is ALIGN_LEFT if the view's resolved layoutDirection is LTR, and ALIGN_RIGHT otherwise.
int	<u>TEXT_DIRECTION_ANY_RTL</u>

(https://developer.android.com/reference/android/view/View.html#TEXT_DIRECTION_ANY_RTL)

Text direction is using "any-RTL" algorithm.

int

TEXT_DIRECTION_FIRST_STRONG

(https://developer.android.com/reference/android/view/View.html#TEXT_DIRECTION_FIRST_STRONG)

Text direction is using "first strong algorithm".

int

TEXT_DIRECTION_FIRST_STRONG_LTR

(https://developer.android.com/reference/android/view/View.html#TEXT_DIRECTION_FIRST_STRONG_LTR)

Text direction is using "first strong algorithm".

int

TEXT_DIRECTION_FIRST_STRONG_RTL

(https://developer.android.com/reference/android/view/View.html#TEXT_DIRECTION_FIRST_STRONG_RTL)

Text direction is using "first strong algorithm".

int

TEXT_DIRECTION_INHERIT

(https://developer.android.com/reference/android/view/View.html#TEXT_DIRECTION_INHERIT)

Text direction is inherited through **ViewGroup**

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

int

TEXT_DIRECTION_LOCALE

(https://developer.android.com/reference/android/view/View.html#TEXT_DIRECTION_LOCALE)

	Text direction is coming from the system Locale.
int	<u>TEXT_DIRECTION_LTR</u> (https://developer.android.com/reference/android/view/View.html#TEXT_DIRECTION_LTR) Text direction is forced to LTR.
int	<u>TEXT_DIRECTION_RTL</u> (https://developer.android.com/reference/android/view/View.html#TEXT_DIRECTION_RTL) Text direction is forced to RTL.
<u>String</u> (https://developer.android.com/reference/java/lang/String.html)	<u>VIEW_LOG_TAG</u> (https://developer.android.com/reference/android/view/View.html#VIEW_LOG_TAG) The logging tag used by this class with android.util.Log.
int	<u>VISIBLE</u> (https://developer.android.com/reference/android/view/View.html#VISIBLE) This view is visible.

Inherited fields

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

public static final Property (<https://developer.android.com/reference/android/util/Property.html>) **<View** (<https://developer.android.com/reference/android/view/View.html>) **Float** (<https://developer.android.com/reference/java/lang/Float.html>)>

```
protected static final int[]
```

```
protected static final int[]
```

```
protected static final int[]
```

```
protected static final int[]
```

```
protected static final int[]
```

```
protected static final int[]
```

```
protected static final int[]
```

```
protected static final int[]
```

```
protected static final int[]
```

```
protected static final int[]
```

```
protected static final int[]
```

```
protected static final int[]
```

```
protected static final int[]
```

```
protected static final int[]
```

```
protected static final int[]
```

```
protected static final int[]
```

```
protected static final int[]
```

```
protected static final int[]
```

```
protected static final int[]
```

```
protected static final int[]
```

```
protected static final int[]
```

```
protected static final int[]
```

```
protected static final int[]
```

```
protected static final int[]
```

```
protected static final int[]
```

```
protected static final int[]
```

```
protected static final int[]
```

```
protected static final int[]
```

```
protected static final int[]
```

public static final Property (<https://developer.android.com/reference/android/util/Property.html>)<**View** (<https://developer.android.com/refere>
Float (<https://developer.android.com/reference/java/lang/Float.html>)>

public static final Property (<https://developer.android.com/reference/android/util/Property.html>)<**View** (<https://developer.android.com/refere>
Float (<https://developer.android.com/reference/java/lang/Float.html>)>

public static final Property (<https://developer.android.com/reference/android/util/Property.html>)<**View** (<https://developer.android.com/refere>
Float (<https://developer.android.com/reference/java/lang/Float.html>)>

public static final Property (<https://developer.android.com/reference/android/util/Property.html>)<**View** (<https://developer.android.com/refere>
Float (<https://developer.android.com/reference/java/lang/Float.html>)>

`public static final Property (https://developer.android.com/reference/android/util/Property.html)<View (https://developer.android.com/refere
Float (https://developer.android.com/reference/java/lang/Float.html)>`

`protected static final int[]`

```
protected static final int[]
```

```
public static final Property (https://developer.android.com/reference/android/util/Property.html)<View (https://developer.android.com/refere  
Float (https://developer.android.com/reference/java/lang/Float.html)>
```

```
public static final Property (https://developer.android.com/reference/android/util/Property.html)<View (https://developer.android.com/refere  
Float (https://developer.android.com/reference/java/lang/Float.html)>
```

public static final Property (<https://developer.android.com/reference/android/util/Property.html>)<**View** (<https://developer.android.com/refere>
Float (<https://developer.android.com/reference/java/lang/Float.html>)>

```
protected static final int[]
```

```
public static final Property (https://developer.android.com/reference/android/util/Property.html)<View (https://developer.android.com/refere  
Float (https://developer.android.com/reference/java/lang/Float.html)>
```

```
public static final Property (https://developer.android.com/reference/android/util/Property.html)<View (https://developer.android.com/refere  
Float (https://developer.android.com/reference/java/lang/Float.html)>
```

```
public static final Property (https://developer.android.com/reference/android/util/Property.html)<View (https://developer.android.com/refere  
Float (https://developer.android.com/reference/java/lang/Float.html)>
```

Public constructors

VideoView ([\(https://developer.android.com/reference/android/widget/VideoView.html#VideoView\(android.content.Context\)\)](https://developer.android.com/reference/android/widget/VideoView.html#VideoView(android.content.Context))) (**Context** (<https://developer.android.com/reference/android/content/Context.html>) **context**)

VideoView ([\(https://developer.android.com/reference/android/widget/VideoView.html#VideoView\(android.content.Context,%20android.util.AttributeSet\)\)](https://developer.android.com/reference/android/widget/VideoView.html#VideoView(android.content.Context,%20android.util.AttributeSet))) (**Context** (<https://developer.android.com/reference/android/content/Context.html>) **context**, **AttributeSet** (<https://developer.android.com/reference/android/util/AttributeSet.html>) **attrs**)

VideoView
([\(https://developer.android.com/reference/android/widget/VideoView.html#VideoView\(android.content.Context,%20android.util.AttributeSet,%20int\)\)](https://developer.android.com/reference/android/widget/VideoView.html#VideoView(android.content.Context,%20android.util.AttributeSet,%20int)))
(**Context** (<https://developer.android.com/reference/android/content/Context.html>) **context**, **AttributeSet** (<https://developer.android.com/reference/android/util/AttributeSet.html>) **attrs**, **int defStyleAttr**)

VideoView
([\(https://developer.android.com/reference/android/widget/VideoView.html#VideoView\(android.content.Context,%20android.util.AttributeSet,%20int,%20int\)\)](https://developer.android.com/reference/android/widget/VideoView.html#VideoView(android.content.Context,%20android.util.AttributeSet,%20int,%20int)))
(**Context** (<https://developer.android.com/reference/android/content/Context.html>) **context**, **AttributeSet** (<https://developer.android.com/reference/android/util/AttributeSet.html>) **attrs**, **int defStyleAttr**, **int defStyleRes**)

Public methods

void	<u>addSubtitleSource</u> ((https://developer.android.com/reference/android/widget/VideoView.html#addSubtitleSource(java.io.InputStream,%20android.media.MediaFormat))) (<u>InputStream</u> (https://developer.android.com/reference/java/io/InputStream.html) is , <u>MediaFormat</u> (https://developer.android.com/reference/android/media/MediaFormat.html) format)
-------------	---

Adds an external subtitle source file (from the provided input stream.) Note that a single external subtitle source may contain multiple or no supported tracks in it.

boolean	<u>canPause</u> (https://developer.android.com/reference/android/widget/VideoView.html#canPause()) ()
boolean	<u>canSeekBackward</u> (https://developer.android.com/reference/android/widget/VideoView.html#canSeekBackward()) ()
boolean	<u>canSeekForward</u> (https://developer.android.com/reference/android/widget/VideoView.html#canSeekForward()) ()
void	<u>draw</u> (https://developer.android.com/reference/android/widget/VideoView.html#draw(android.graphics.Canvas)) (<u>Canvas</u> (https://developer.android.com/reference/android/graphics/Canvas.html) canvas) Manually render this view (and all of its children) to the given Canvas.
<u>CharSequence</u> (https://developer.android.com/reference/java/lang/CharSequence.html)	<u>getAccessibilityClassName</u> (https://developer.android.com/reference/android/widget/VideoView.html#getAccessibilityClassName()) () Return the class name of this object to be used for accessibility purposes.
int	<u>getAudioSessionId</u> (https://developer.android.com/reference/android/widget/VideoView.html#getAudioSessionId()) () Get the audio session id for the player used by this VideoView.
int	<u>getBufferPercentage</u> (https://developer.android.com/reference/android/widget/VideoView.html#getBufferPercentage()) ()
int	<u>getCurrentPosition</u> (https://developer.android.com/reference/android/widget/VideoView.html#getCurrentPosition()) ()
int	<u>getDuration</u> (https://developer.android.com/reference/android/widget/VideoView.html#getDuration()) ()
boolean	<u>isPlaying</u> (https://developer.android.com/reference/android/widget/VideoView.html#isPlaying()) ()
boolean	<u>onKeyDown</u> (https://developer.android.com/reference/android/widget/VideoView.html#onKeyDown(int,%20android.view.KeyEvent)) (int keyCode , <u>KeyEvent</u> (https://developer.android.com/reference/android/view/KeyEvent.html) event)

Default implementation of **KeyEvent.Callback.onKeyDown()**.

([https://developer.android.com/reference/android/view/KeyEvent.Callback.html#onKeyDown\(int,%20android.view.KeyEvent\)](https://developer.android.com/reference/android/view/KeyEvent.Callback.html#onKeyDown(int,%20android.view.KeyEvent)))

: perform press of the view when **KeyEvent.KEYCODE_DPAD_CENTER**

(https://developer.android.com/reference/android/view/KeyEvent.html#KEYCODE_DPAD_CENTER) or

KeyEvent.KEYCODE_ENTER (https://developer.android.com/reference/android/view/KeyEvent.html#KEYCODE_ENTER) is released, if the view is enabled and clickable.

boolean

onTouchEvent

([https://developer.android.com/reference/android/widget/VideoView.html#onTouchEvent\(android.view.MotionEvent\)](https://developer.android.com/reference/android/widget/VideoView.html#onTouchEvent(android.view.MotionEvent)))

(**MotionEvent** (<https://developer.android.com/reference/android/view/MotionEvent.html>) **ev**)

Implement this method to handle touch screen motion events.

boolean

onTrackballEvent

([https://developer.android.com/reference/android/widget/VideoView.html#onTrackballEvent\(android.view.MotionEvent\)](https://developer.android.com/reference/android/widget/VideoView.html#onTrackballEvent(android.view.MotionEvent)))

(**MotionEvent** (<https://developer.android.com/reference/android/view/MotionEvent.html>) **ev**)

Implement this method to handle trackball motion events.

void

pause ([https://developer.android.com/reference/android/widget/VideoView.html#pause\(\)](https://developer.android.com/reference/android/widget/VideoView.html#pause())) ()

int

resolveAdjustedSize

([https://developer.android.com/reference/android/widget/VideoView.html#resolveAdjustedSize\(int,%20int\)](https://developer.android.com/reference/android/widget/VideoView.html#resolveAdjustedSize(int,%20int))) (**int**

desiredSize, **int measureSpec**)

void

resume ([https://developer.android.com/reference/android/widget/VideoView.html#resume\(\)](https://developer.android.com/reference/android/widget/VideoView.html#resume())) ()

void

seekTo ([https://developer.android.com/reference/android/widget/VideoView.html#seekTo\(int\)](https://developer.android.com/reference/android/widget/VideoView.html#seekTo(int))) (**int msec**)

void

setAudioAttributes

([https://developer.android.com/reference/android/widget/VideoView.html#setAudioAttributes\(android.media.AudioAttributes\)](https://developer.android.com/reference/android/widget/VideoView.html#setAudioAttributes(android.media.AudioAttributes)))

(**AudioAttributes** (<https://developer.android.com/reference/android/media/AudioAttributes.html>) **attributes**)

Sets the **AudioAttributes** (<https://developer.android.com/reference/android/media/AudioAttributes.html>) to be used during the playback of the video.

void	<u>setAudioFocusRequest</u> (<code>https://developer.android.com/reference/android/widget/VideoView.html#setAudioFocusRequest(int)</code>)(int focusGain) Sets which type of audio focus will be requested during the playback, or configures playback to not request audio focus.
void	<u>setMediaController</u> (<code>https://developer.android.com/reference/android/widget/VideoView.html#setMediaController(android.widget.MediaController)</code>) (<u>MediaController</u> (<code>https://developer.android.com/reference/android/widget/MediaController.html</code>) controller)
void	<u>setOnCompletionListener</u> (<code>https://developer.android.com/reference/android/widget/VideoView.html#setOnCompletionListener(android.media.MediaPlayer.OnCompletionListener)</code>) (<u>MediaPlayer.OnCompletionListener</u> (<code>https://developer.android.com/reference/android/media/MediaPlayer.OnCompletionListener.html</code>) l) Register a callback to be invoked when the end of a media file has been reached during playback.
void	<u>setOnErrorListener</u> (<code>https://developer.android.com/reference/android/widget/VideoView.html#setOnErrorListener(android.media.MediaPlayer.OnErrorListener)</code>) (<u>MediaPlayer.OnErrorListener</u> (<code>https://developer.android.com/reference/android/media/MediaPlayer.OnErrorListener.html</code>) l) Register a callback to be invoked when an error occurs during playback or setup.
void	<u>setOnInfoListener</u> (<code>https://developer.android.com/reference/android/widget/VideoView.html#setOnInfoListener(android.media.MediaPlayer.OnInfoListener)</code>)

(MediaPlayer.OnInfoListener

(<https://developer.android.com/reference/android/media/MediaPlayer.OnInfoListener.html>) 1)

Register a callback to be invoked when an informational event occurs during playback or setup.

void

setOnPreparedListener

([https://developer.android.com/reference/android/widget/VideoView.html#setOnPreparedListener\(android.media.MediaPlayer.OnPreparedListener\)](https://developer.android.com/reference/android/widget/VideoView.html#setOnPreparedListener(android.media.MediaPlayer.OnPreparedListener)))

(MediaPlayer.OnPreparedListener

(<https://developer.android.com/reference/android/media/MediaPlayer.OnPreparedListener.html>) 1)

Register a callback to be invoked when the media file is loaded and ready to go.

void

setVideoPath

([https://developer.android.com/reference/android/widget/VideoView.html#setVideoPath\(java.lang.String\)](https://developer.android.com/reference/android/widget/VideoView.html#setVideoPath(java.lang.String)))(String (<https://developer.android.com/reference/java/lang/String.html>) **path**)

Sets video path.

void

setVideoURI

([https://developer.android.com/reference/android/widget/VideoView.html#setVideoURI\(android.net.Uri,%20java.util.Map%20java.lang.String,%20java.lang.String%\)](https://developer.android.com/reference/android/widget/VideoView.html#setVideoURI(android.net.Uri,%20java.util.Map%20java.lang.String,%20java.lang.String%)))

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

(<https://developer.android.com/reference/java/util/Map.html>)<String

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

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

Sets video URI using specific headers.

void

setVideoURI ([https://developer.android.com/reference/android/widget/VideoView.html#setVideoURI\(android.net.Uri\)](https://developer.android.com/reference/android/widget/VideoView.html#setVideoURI(android.net.Uri)))

(**Uri** (<https://developer.android.com/reference/android/net/Uri.html>) **uri**)

Sets video URI.

void	<u>start</u> (https://developer.android.com/reference/android/widget/VideoView.html#start()) ()
void	<u>stopPlayback</u> (https://developer.android.com/reference/android/widget/VideoView.html#stopPlayback()) ()
void	<u>suspend</u> (https://developer.android.com/reference/android/widget/VideoView.html#suspend()) ()

Protected methods

void	<u>onAttachedToWindow</u> (https://developer.android.com/reference/android/widget/VideoView.html#onAttachedToWindow()) ()
-------------	--

This is called when the view is attached to a window.

void	<u>onDetachedFromWindow</u> (https://developer.android.com/reference/android/widget/VideoView.html#onDetachedFromWindow()) ()
-------------	--

This is called when the view is detached from a window.

void	<u>onLayout</u> (https://developer.android.com/reference/android/widget/VideoView.html#onLayout(boolean,%20int,%20int,%20int,%20int)) (boolean changed, int left, int top, int right, int bottom)
-------------	--

Called from layout when this view should assign a size and position to each of its children.

void	<u>onMeasure</u> (https://developer.android.com/reference/android/widget/VideoView.html#onMeasure(int,%20int)) (int widthMeasureSpec, int heightMeasureSpec)
-------------	--

Measure the view and its content to determine the measured width and the measured height.

Inherited methods

From class **android.view.SurfaceView** (<https://developer.android.com/reference/android/view/SurfaceView.html>)

void	<u>dispatchDraw</u>
-------------	----------------------------

(<https://developer.android.com/reference/android/view/graphics/Canvas>)

(**Canvas** (<https://developer.android.com/reference/android/graphics/Canvas>))

Called by draw to draw the child views.

void

draw

(<https://developer.android.com/reference/android/view/graphics/Canvas>)

(**Canvas** (<https://developer.android.com/reference/android/graphics/Canvas>))

Manually render this view (and all of its children) to the g

boolean

gatherTransparentRegion

([https://developer.android.com/reference/android/view/Region\(android.graphics.Region\)](https://developer.android.com/reference/android/view/Region(android.graphics.Region)))

(**Region** (<https://developer.android.com/reference/android/graphics/Region>))

This is used by the RootView to perform an optimization or several SurfaceView.

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

getHolder

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

Return the SurfaceHolder providing access and control o surface.

void

onAttachedToWindow

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

()

This is called when the view is attached to a window.

void

onDetachedFromWindow

([https://developer.android.com/reference/android/view/Window\(\)](https://developer.android.com/reference/android/view/Window()))
()

This is called when the view is detached from a window.

void

onMeasure

([https://developer.android.com/reference/android/view/View#onMeasure\(int,int\)](https://developer.android.com/reference/android/view/View#onMeasure(int,int)))
(**int widthMeasureSpec, int heightMeasureSpec**)

Measure the view and its content to determine the measurement

void

onWindowVisibilityChanged

([https://developer.android.com/reference/android/view/View#onWindowVisibilityChanged\(int\)](https://developer.android.com/reference/android/view/View#onWindowVisibilityChanged(int)))
(**int visibility**)

Called when the window containing has change its visibility
([https://developer.android.com/reference/android/view/View#onWindowVisibilityChanged\(int\)](https://developer.android.com/reference/android/view/View#onWindowVisibilityChanged(int)))
([https://developer.android.com/reference/android/view/View#onWindowVisibilityChanged\(int\)](https://developer.android.com/reference/android/view/View#onWindowVisibilityChanged(int)))
([https://developer.android.com/reference/android/view/View#onWindowVisibilityChanged\(int\)](https://developer.android.com/reference/android/view/View#onWindowVisibilityChanged(int)))

void

setSecure

([https://developer.android.com/reference/android/view/View#setSecure\(boolean\)](https://developer.android.com/reference/android/view/View#setSecure(boolean)))
(**boolean isSecure**)

Control whether the surface view's content should be treated as secure, preventing it from appearing in screenshots or from being viewed on non-secure displays.

void

setVisibility

(<https://developer.android.com/reference/android/view/>
(int visibility))

Set the visibility state of this view.

void

setZOrderMediaOverlay

(<https://developer.android.com/reference/android/view/>
verlay(boolean))
(boolean isMediaOverlay)

Control whether the surface view's surface is placed on t
the window (but still behind the window itself).

void

setZOrderOnTop

(<https://developer.android.com/reference/android/view/>
p(boolean))
(boolean onTop)

Control whether the surface view's surface is placed on

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

void

void

void

void

void

void

void

void

void

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

void

void

void

boolean

boolean

boolean

void

void

void

void

boolean

boolean

boolean

boolean

boolean

boolean

final void

void

final void

boolean

void

void

static int

`int`

`int`

`int`

void

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

`int`

`int`

`int`

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

void

void

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

boolean

void

void

boolean

void

void

void

boolean

boolean

boolean

boolean

boolean

boolean

boolean

boolean

boolean

boolean

boolean

boolean

void

boolean

void

void

void

void

void

void

void

void

void

boolean

boolean

boolean

void

void

void

void

void

void

void

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

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

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

```
void
```

boolean

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

void

void

static int

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

int

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

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

int

int

float

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

IBinder (<https://developer.android.com/reference/android/os/IBinder.html>)

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

final AutofillId (<https://developer.android.com/reference/android/view/autofill/AutofillId.html>)

int

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

Drawable (<https://developer.android.com/reference/android/graphics/drawable/Drawable.html>)

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

PorterDuff.Mode (<https://developer.android.com/reference/android/graphics/PorterDuff.Mode.html>)

int

final int

float

int

float

boolean

Rect (<https://developer.android.com/reference/android/graphics/Rect.html>)

final boolean

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

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

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

final boolean

`static int`

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

`final int[]`

Bitmap (<https://developer.android.com/reference/android/graphics/Bitmap.html>)

`int`

int

void

long

float

boolean

boolean

int

ArrayList (<https://developer.android.com/reference/java/util/ArrayList.html>)<**View** (<https://developer.android.com/reference/android/view/View.html>),

void

Drawable (<https://developer.android.com/reference/android/graphics/drawable/Drawable.html>)

int

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

PorterDuff.Mode (<https://developer.android.com/reference/android/graphics/PorterDuff.Mode.html>)

final boolean

boolean

Handler (<https://developer.android.com/reference/android/os/Handler.html>)

final boolean

final int

void

int

int

`int`

`int`

`int`

`boolean`

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

int

`int`

`int`

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

`final int`

float

int

final boolean

void

void

Matrix (<https://developer.android.com/reference/android/graphics/Matrix.html>)

final int

final int

`final int`

`final int`

final int

int

int

int

int

int

int

`int`

`int`

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

`int`

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

`int`

`int`

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

int

int

int

int

int

int

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

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

float

float

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

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

final boolean

final int

float

int

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

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

float

float

float

float

float

int

int

int

int

int

`final int`

`final int`

`int`

StateListAnimator (<https://developer.android.com/reference/android/animation/StateListAnimator.html>)

int

int

int

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

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

int

int

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

`final int`

`float`

`int`

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

ArrayList (<https://developer.android.com/reference/java/util/ArrayList.html>)<**View** (<https://developer.android.com/reference/android/view/View.html>),

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

float

float

float

int

int

int

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

int

final int

int

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

int

IBinder (<https://developer.android.com/reference/android/os/IBinder.html>)

int

void

float

float

float

boolean

boolean

boolean

boolean

boolean

boolean

boolean

boolean

boolean

static [View](https://developer.android.com/reference/android/view/View.html) (https://developer.android.com/reference/android/view/View.html)

void

void

void

void

void

boolean

boolean

boolean

boolean

boolean

boolean

boolean

boolean

boolean

boolean

final boolean

final boolean

boolean

final boolean

boolean

boolean

boolean

boolean

boolean

boolean

final boolean

boolean

boolean

boolean

final boolean

boolean

boolean

boolean

boolean

boolean

boolean

boolean

boolean

boolean

boolean

boolean

boolean

boolean

boolean

boolean

boolean

boolean

`boolean`

`final boolean`

boolean

boolean

boolean

boolean

boolean

void

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

void

final void

```
static int[]
```

```
void
```

void

void

void

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

void

void

boolean

boolean

void

void

`int[]`

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

`void`

void

boolean

void

void

final void

boolean

void

void

void

boolean

void

boolean

void

void

boolean

boolean

boolean

boolean

boolean

boolean

void

void

void

void

void

void

void

void

void

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

void

void

Parcelable (<https://developer.android.com/reference/android/os/Parcelable.html>)

void

void

boolean

void

void

boolean

boolean

void

void

void

void

void

boolean

boolean

boolean

boolean

boolean

boolean

boolean

boolean

boolean

void

boolean

boolean

void

void

void

void

void

void

void

void

void

void

boolean

void

void

void

void

void

final boolean

final boolean

boolean

final boolean

void

void

boolean

boolean

final void

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

void

static int

static int

boolean

void

void

void

void

void

void

void

void

void

void

void

void

void

void

void

void

void

void

void

void

void

void

void

void

final void

void

void

void

void

void

void

void

void

void

void

void

void

void

void

void

void

void

void

void

void

void

void

void

void

void

void

void

void

void

void

void

void

void

void

void

void

void

void

void

final void

void

final void

void

void

void

void

void

void

void

void

void

void

void

void

void

void

void

void

void

void

void

void

void

void

void

void

void

void

void

void

void

void

void

void

void

final void

final void

void

void

void

void

void

void

void

void

void

void

void

void

void

void

void

void

void

void

void

void

void

void

void

void

void

void

void

final void

void

final void

void

void

void

void

void

void

void

void

void

void

void

void

boolean

boolean

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

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

void

final boolean

`final boolean`

boolean

void

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

void

void

final void

boolean

boolean

boolean

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

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

clone (https://developer.android.com/reference/java/lang/Object

Creates and returns a copy of this object.

boolean

equals (https://developer.android.com/reference/java/lang/Object
(**Object** (https://developer.android.com/reference/java/lang/Objec

Indicates whether some other object is "equal to" this one.

void

finalize (https://developer.android.com/reference/java/lang/Objc

Called by the garbage collector on an object when garbage collectio
references to the object.

final **Class** (https://developer.android.com/reference/java/lang/Class.html)<?> **getClass** (https://developer.android.com/reference/java/lang/Objc

Returns the runtime class of this **Object**.

int	<u>hashCode</u> (https://developer.android.com/reference/java/lang/Object#hashCode) Returns a hash code value for the object.
final void	<u>notify</u> (https://developer.android.com/reference/java/lang/Object#notify) Wakes up a single thread that is waiting on this object's monitor.
final void	<u>notifyAll</u> (https://developer.android.com/reference/java/lang/Object#notifyAll) Wakes up all threads that are waiting on this object's monitor.
<u>String</u> (https://developer.android.com/reference/java/lang/String.html)	<u>toString</u> (https://developer.android.com/reference/java/lang/Object#toString) Returns a string representation of the object.
final void	<u>wait</u> (https://developer.android.com/reference/java/lang/Object.html#wait(long, int, int)) millis, int nanos) Causes the current thread to wait until another thread invokes the <u>notify</u> (https://developer.android.com/reference/java/lang/Object.html#notify) (https://developer.android.com/reference/java/lang/Object.html#notifyAll) some other thread interrupts the current thread, or a certain amount
final void	<u>wait</u> (https://developer.android.com/reference/java/lang/Object.html#wait(long)) Causes the current thread to wait until either another thread invokes (https://developer.android.com/reference/java/lang/Object.html#notify) (https://developer.android.com/reference/java/lang/Object.html#notifyAll) specified amount of time has elapsed.
final void	<u>wait</u> (https://developer.android.com/reference/java/lang/Object.html#wait(long, int, int)) Causes the current thread to wait until another thread invokes the <u>notify</u> (https://developer.android.com/reference/java/lang/Object.html#notify)

notifyAll() (<https://developer.android.com/reference/java/lang>
for this object.

From interface **android.graphics.drawable.Drawable.Callback** (<https://developer.android.com/reference/android/graphics/drawable/Drawable>

abstract void **invalidateDrawable**
([https://developer.android.com/reference/android/graphics/drawable/Drawable.Callback.html#invalidateDrawable\(an](https://developer.android.com/reference/android/graphics/drawable/Drawable.Callback.html#invalidateDrawable(android.graphics.drawable.Drawable))
(**Drawable** (<https://developer.android.com/reference/android/graphics/drawable/Drawable.html>) **who**)

Called when the drawable needs to be redrawn.

abstract void **scheduleDrawable**
([https://developer.android.com/reference/android/graphics/drawable/Drawable.Callback.html#scheduleDrawable\(andi](https://developer.android.com/reference/android/graphics/drawable/Drawable.Callback.html#scheduleDrawable(android.graphics.drawable.Drawable,java.lang.Runnable,long))
[ava.lang.Runnable,%20long\)\)](https://developer.android.com/reference/java/lang/Runnable.html)
(**Drawable** (<https://developer.android.com/reference/android/graphics/drawable/Drawable.html>) **who**, **Runnable**
(<https://developer.android.com/reference/java/lang/Runnable.html>) **what**, **long when**)

A Drawable can call this to schedule the next frame of its animation.

abstract void **unscheduleDrawable**
([https://developer.android.com/reference/android/graphics/drawable/Drawable.Callback.html#unscheduleDrawable\(;](https://developer.android.com/reference/android/graphics/drawable/Drawable.Callback.html#unscheduleDrawable(android.graphics.drawable.Drawable,java.lang.Runnable))
[,%20java.lang.Runnable\)\)](https://developer.android.com/reference/java/lang/Runnable.html)
(**Drawable** (<https://developer.android.com/reference/android/graphics/drawable/Drawable.html>) **who**, **Runnable**
(<https://developer.android.com/reference/java/lang/Runnable.html>) **what**)

A Drawable can call this to unschedule an action previously scheduled with **scheduleDrawable(Drawable, Runnable, long)**
([https://developer.android.com/reference/android/graphics/drawable/Drawable.Callback.html#scheduleDrawable\(ani](https://developer.android.com/reference/android/graphics/drawable/Drawable.Callback.html#scheduleDrawable(android.graphics.drawable.Drawable,java.lang.Runnable,long))
[20java.lang.Runnable,%20long\)\)](https://developer.android.com/reference/java/lang/Runnable.html)
.

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

abstract boolean **onKeyDown** ([https://developer.android.com/reference/android/view/KeyEvent.Callback.html#onKeyDown\(int,%20andr](https://developer.android.com/reference/android/view/KeyEvent.Callback.html#onKeyDown(int,int,KeyEvent))
[keyCode](https://developer.android.com/reference/android/view/KeyEvent.html), **KeyEvent** (<https://developer.android.com/reference/android/view/KeyEvent.html>) **event**)

Called when a key down event has occurred.

abstract boolean

onKeyLongPress ([https://developer.android.com/reference/android/view/KeyEvent.Callback.html#onKeyLongPress\(int,%20int,%20KeyEvent\)](https://developer.android.com/reference/android/view/KeyEvent.Callback.html#onKeyLongPress(int,%20int,%20KeyEvent)) **keyCode**, **KeyEvent** (<https://developer.android.com/reference/android/view/KeyEvent.html>) **event**)

Called when a long press has occurred.

abstract boolean

onKeyMultiple ([https://developer.android.com/reference/android/view/KeyEvent.Callback.html#onKeyMultiple\(int,%20int,%20int,%20KeyEvent\)](https://developer.android.com/reference/android/view/KeyEvent.Callback.html#onKeyMultiple(int,%20int,%20int,%20KeyEvent)) **int keyCode**, **int count**, **KeyEvent** (<https://developer.android.com/reference/android/view/KeyEvent.html>) **e**

Called when a user's interaction with an analog control, such as flinging a trackball, generates simulated down/up event: quick succession.

abstract boolean

onKeyUp ([https://developer.android.com/reference/android/view/KeyEvent.Callback.html#onKeyUp\(int,%20android.view.KeyEvent\)](https://developer.android.com/reference/android/view/KeyEvent.Callback.html#onKeyUp(int,%20android.view.KeyEvent)) **int keyCode**, **KeyEvent** (<https://developer.android.com/reference/android/view/KeyEvent.html>) **event**)

Called when a key up event has occurred.

From interface **android.view.accessibility.AccessibilityEventSource** (<https://developer.android.com/reference/android/view/accessibility/AccessibilityEventSource.html>)

abstract void

sendAccessibilityEvent

([https://developer.android.com/reference/android/view/accessibility/AccessibilityEventSource.html#sendAccessibilityEvent\(int,%20AccessibilityEvent\)](https://developer.android.com/reference/android/view/accessibility/AccessibilityEventSource.html#sendAccessibilityEvent(int,%20AccessibilityEvent)) **int eventType**, **AccessibilityEvent event**)

Handles the request for sending an **AccessibilityEvent** (<https://developer.android.com/reference/android/view/accessibility/AccessibilityEvent.html>) given the event type.

abstract void

sendAccessibilityEventUnchecked

([https://developer.android.com/reference/android/view/accessibility/AccessibilityEventSource.html#sendAccessibilityEventUnchecked\(int,%20AccessibilityEvent\)](https://developer.android.com/reference/android/view/accessibility/AccessibilityEventSource.html#sendAccessibilityEventUnchecked(int,%20AccessibilityEvent)) **int eventType**, **AccessibilityEvent event**)

(**AccessibilityEvent** (<https://developer.android.com/reference/android/view/accessibility/AccessibilityEvent.html>) **event**)

Handles the request for sending an **AccessibilityEvent** (<https://developer.android.com/reference/android/view/accessibility/AccessibilityEvent.html>)

From interface [android.widget.MediaController.MediaPlayerControl](https://developer.android.com/reference/android/widget/MediaController.MediaPlayerControl) (https://developer.android.com/reference/android/widget/MediaConti

abstract boolean	<u>canPause</u> (https://developer.android.com/reference/android/widget/MediaController.MediaPlayerControl.html#canP:
abstract boolean	<u>canSeekBackward</u> (https://developer.android.com/reference/android/widget/MediaController.MediaPlayerControl.htmr
abstract boolean	<u>canSeekForward</u> (https://developer.android.com/reference/android/widget/MediaController.MediaPlayerControl.html
abstract int	<u>getAudioSessionId</u> (https://developer.android.com/reference/android/widget/MediaController.MediaPlayerControl.l Get the audio session id for the player used by this VideoView.
abstract int	<u>getBufferPercentage</u> (https://developer.android.com/reference/android/widget/MediaController.MediaPlayerContr
abstract int	<u>getCurrentPosition</u> (https://developer.android.com/reference/android/widget/MediaController.MediaPlayerControl
abstract int	<u>getDuration</u> (https://developer.android.com/reference/android/widget/MediaController.MediaPlayerControl.html#ge
abstract boolean	<u>isPlaying</u> (https://developer.android.com/reference/android/widget/MediaController.MediaPlayerControl.html#isPlay
abstract void	<u>pause</u> (https://developer.android.com/reference/android/widget/MediaController.MediaPlayerControl.html#pause()) ()
abstract void	<u>seekTo</u> (https://developer.android.com/reference/android/widget/MediaController.MediaPlayerControl.html#seekTo(in
abstract void	<u>start</u> (https://developer.android.com/reference/android/widget/MediaController.MediaPlayerControl.html#start()) ()

Public constructors

VideoView

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

```
public VideoView (Context context)
```

Parameters

context	Context
---------	---------

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

```
public VideoView (Context context,  
                 AttributeSet attrs)
```

Parameters

context	Context
---------	---------

attrs	AttributeSet
-------	--------------

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

```
public VideoView (Context context,  
                 AttributeSet attrs,  
                 int defStyleAttr)
```

Parameters

context	Context
attrs	AttributeSet
defStyleAttr	int

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

```
public VideoView (Context context,  
                 AttributeSet attrs,  
                 int defStyleAttr,  
                 int defStyleRes)
```

Parameters

context	Context
attrs	AttributeSet
defStyleAttr	int
defStyleRes	int

Public methods

addSubtitleSource

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

```
public void addSubtitleSource (InputStream is,
                             MediaFormat format)
```

Adds an external subtitle source file (from the provided input stream.) Note that a single external subtitle source may contain multiple or no supported tracks in it. If the source contained at least one track in it, one will receive an [**MediaPlayer.MEDIA_INFO_METADATA_UPDATE**](https://developer.android.com/reference/android/media/MediaPlayer.html#MEDIA_INFO_METADATA_UPDATE) (https://developer.android.com/reference/android/media/MediaPlayer.html#MEDIA_INFO_METADATA_UPDATE) info message. Otherwise, if reading the source takes excessive time, one will receive a [**MediaPlayer.MEDIA_INFO_SUBTITLE_TIMED_OUT**](https://developer.android.com/reference/android/media/MediaPlayer.html#MEDIA_INFO_SUBTITLE_TIMED_OUT) (https://developer.android.com/reference/android/media/MediaPlayer.html#MEDIA_INFO_SUBTITLE_TIMED_OUT) message. If the source contained no supported track (including an empty source file or null input stream), one will receive a [**MediaPlayer.MEDIA_INFO_UNSUPPORTED_SUBTITLE**](https://developer.android.com/reference/android/media/MediaPlayer.html#MEDIA_INFO_UNSUPPORTED_SUBTITLE) (https://developer.android.com/reference/android/media/MediaPlayer.html#MEDIA_INFO_UNSUPPORTED_SUBTITLE) message. One can find the total number of available tracks using [**MediaPlayer.getTrackInfo\(\)**](https://developer.android.com/reference/android/media/MediaPlayer.html#getTrackInfo()) ([https://developer.android.com/reference/android/media/MediaPlayer.html#getTrackInfo\(\)](https://developer.android.com/reference/android/media/MediaPlayer.html#getTrackInfo())) to see what additional tracks become available after this method call.

Parameters

is	InputStream : input stream containing the subtitle data. It will be closed by the media framework.
----	---

format

MediaFormat: the format of the subtitle track(s). Must contain at least the mime type (**MediaFormat.KEY_MIME** (https://developer.android.com/reference/android/media/MediaFormat.html#KEY_MIME)) and the language (**MediaFormat.KEY_LANGUAGE** (https://developer.android.com/reference/android/media/MediaFormat.html#KEY_LANGUAGE)) of the file. If the file itself contains the language information, specify "und" for the language.

canPause

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

```
public boolean canPause ()
```

Returns

boolean

canSeekBackward

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

```
public boolean canSeekBackward ()
```

Returns

boolean

canSeekForward

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

```
public boolean canSeekForward ()
```

Returns

boolean

draw

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

```
public void draw (Canvas canvas)
```

Manually render this view (and all of its children) to the given Canvas. The view must have already done a full layout before this function is called. When implementing a view, implement **[onDraw\(android.graphics.Canvas\)](https://developer.android.com/reference/android/graphics/Canvas.html)**.

([https://developer.android.com/reference/android/view/View.html#onDraw\(android.graphics.Canvas\)](https://developer.android.com/reference/android/view/View.html#onDraw(android.graphics.Canvas))) instead of overriding this method. If you do need to override this method, call the superclass version.

Parameters

canvas	Canvas: The Canvas to which the View is rendered.
---------------	--

getAccessibilityClassName

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

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

Return the class name of this object to be used for accessibility purposes. Subclasses should only override this if they are implementing something that should be seen as a completely new class of view when used by accessibility, unrelated to the class it is deriving from.

This is used to fill in [**AccessibilityNodeInfo.setClassName**](#)

([https://developer.android.com/reference/android/view/accessibility/AccessibilityNodeInfo.html#setClassName\(java.lang.CharSequence\)](https://developer.android.com/reference/android/view/accessibility/AccessibilityNodeInfo.html#setClassName(java.lang.CharSequence)))).

Returns

[**CharSequence**](#)

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

getAudioSessionId

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

```
public int getAudioSessionId ()
```

Get the audio session id for the player used by this `VideoView`. This can be used to apply audio effects to the audio track of a video.

Returns

int	The audio session, or 0 if there was an error.
------------	--

getBufferPercentage

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

```
public int getBufferPercentage ()
```

Returns

int

getCurrentPosition

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

```
public int getCurrentPosition ()
```

Returns

int

getDuration

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

```
public int getDuration ()
```

Returns

int

isPlaying

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

```
public boolean isPlaying ()
```

Returns

boolean

onKeyDown

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

```
public boolean onKeyDown (int keyCode,  
    KeyEvent event)
```

Default implementation of [KeyEvent.Callback.onKeyDown\(\)](https://developer.android.com/reference/android/view/KeyEvent.Callback.html#onKeyDown(int,%20android.view.KeyEvent)).

([https://developer.android.com/reference/android/view/KeyEvent.Callback.html#onKeyDown\(int,%20android.view.KeyEvent\)](https://developer.android.com/reference/android/view/KeyEvent.Callback.html#onKeyDown(int,%20android.view.KeyEvent))): perform press of the view when [KeyEvent.KEYCODE_DPAD_CENTER](https://developer.android.com/reference/android/view/KeyEvent.html#KEYCODE_DPAD_CENTER) or [KeyEvent.KEYCODE_ENTER](https://developer.android.com/reference/android/view/KeyEvent.html#KEYCODE_ENTER) is released, if the view is enabled and clickable.

Key presses in software keyboards will generally NOT trigger this listener, although some may elect to do so in some situations. Do not rely on this to catch software key presses.

Parameters

keyCode **int**: a key code that represents the button pressed, from [KeyEvent](https://developer.android.com/reference/android/view/KeyEvent.html) (<https://developer.android.com/reference/android/view/KeyEvent.html>)

event	KeyEvent : the KeyEvent object that defines the button action
-------	--

Returns

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

onTouchEvent

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

```
public boolean onTouchEvent (MotionEvent (https://developer.android.com/reference/android/view/MotionEvent.html) ev)
```

Implement this method to handle touch screen motion events.

If this method is used to detect click actions, it is recommended that the actions be performed by implementing and calling [performClick\(\)](https://developer.android.com/reference/android/view/View.html#performClick()) ([https://developer.android.com/reference/android/view/View.html#performClick\(\)](https://developer.android.com/reference/android/view/View.html#performClick())). This will ensure consistent system behavior, including:

- obeying click sound preferences
- dispatching OnClickListener calls
- handling [ACTION_CLICK](https://developer.android.com/reference/android/view/accessibility/AccessibilityNodeInfo.html#ACTION_CLICK) (https://developer.android.com/reference/android/view/accessibility/AccessibilityNodeInfo.html#ACTION_CLICK) when accessibility features are enabled

Parameters

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

Returns

<code>boolean</code>	True if the event was handled, false otherwise.
----------------------	---

onTrackballEvent

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

```
public boolean onTrackballEvent (MotionEvent ev)
```

Implement this method to handle trackball motion events. The *relative* movement of the trackball since the last event can be retrieve with [`MotionEvent.getX\(\)`](https://developer.android.com/reference/android/view/MotionEvent.html#getX()) and [`MotionEvent.getY\(\)`](https://developer.android.com/reference/android/view/MotionEvent.html#getY()). These are normalized so that a movement of 1 corresponds to the user pressing one DPAD key (so they will often be fractional values, representing the more fine-grained movement information available from a trackball).

Parameters

<code>ev</code>	MotionEvent: The motion event.
-----------------	---------------------------------------

Returns

<code>boolean</code>	True if the event was handled, false otherwise.
----------------------	---

pause

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

```
public void pause ()
```

resolveAdjustedSize

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

```
public int resolveAdjustedSize (int desiredSize,  
                                int measureSpec)
```

Parameters

desiredSize	int
--------------------	------------

measureSpec	int
--------------------	------------

Returns

int

resume

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

```
public void resume ()
```

seekTo

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

```
public void seekTo (int msec)
```

Parameters

msec	int
------	-----

setAudioAttributes

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

```
public void setAudioAttributes (AudioAttributes attribute;
```

Sets the [AudioAttributes](https://developer.android.com/reference/android/media/AudioAttributes.html) to be used during the playback of the video.

Parameters

attributes	AudioAttributes: non-null AudioAttributes .
------------	---

setAudioFocusRequest

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

```
public void setAudioFocusRequest (int focusGain)
```

Sets which type of audio focus will be requested during the playback, or configures playback to not request audio focus. Valid values for focus requests are `AudioManager.AUDIOFOCUS_GAIN` (https://developer.android.com/reference/android/media/AudioManager.html#AUDIOFOCUS_GAIN), `AudioManager.AUDIOFOCUS_GAIN_TRANSIENT` (https://developer.android.com/reference/android/media/AudioManager.html#AUDIOFOCUS_GAIN_TRANSIENT), `AudioManager.AUDIOFOCUS_GAIN_TRANSIENT_MAY_DUCK` (https://developer.android.com/reference/android/media/AudioManager.html#AUDIOFOCUS_GAIN_TRANSIENT_MAY_DUCK), and `AudioManager.AUDIOFOCUS_GAIN_TRANSIENT_EXCLUSIVE` (https://developer.android.com/reference/android/media/AudioManager.html#AUDIOFOCUS_GAIN_TRANSIENT_EXCLUSIVE). Or use `AudioManager.AUDIOFOCUS_NONE` (https://developer.android.com/reference/android/media/AudioManager.html#AUDIOFOCUS_NONE) to express that audio focus should not be requested when playback starts. You can for instance use this when playing a silent animation through this class, and you don't want to affect other audio applications playing in the background.

Parameters

focusGain	int: the type of audio focus gain that will be requested, or <code>AudioManager.AUDIOFOCUS_NONE</code> (https://developer.android.com/reference/android/media/AudioManager.html#AUDIOFOCUS_NONE) to disable the use audio focus during playback.
-----------	--

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

`public void setMediaController` (`MediaController` (<https://developer.android.com/reference/android/widget/MediaController.html>) controller

Parameters

controller	MediaController
------------	-----------------

setOnCompletionListener

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

```
public void setOnCompletionListener (MediaPlayer.OnCompletionListener (https://developer.android.com/reference/android/media/Media
```

Register a callback to be invoked when the end of a media file has been reached during playback.

Parameters

1 **MediaPlayer.OnCompletionListener**: The callback that will be run

setOnErrorListener

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

```
public void setOnErrorListener (MediaPlayer.OnErrorListener (https://developer.android.com/reference/android/media/MediaPlayer.OnErrc
```

Register a callback to be invoked when an error occurs during playback or setup. If no listener is specified, or if the listener returned false, `VideoView` will inform the user of any errors.

Parameters

1 **MediaPlayer.OnErrorListener**: The callback that will be run

setOnInfoListener

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

```
public void setOnInfoListener (MediaPlayer.OnInfoListener (https://developer.android.com/reference/android/media/MediaPlayer.OnInfoListener
```

Register a callback to be invoked when an informational event occurs during playback or setup.

Parameters

1 **MediaPlayer.OnInfoListener**: The callback that will be run

setOnPreparedListener

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

```
public void setOnPreparedListener (MediaPlayer.OnPreparedListener (https://developer.android.com/reference/android/media/MediaPlayer.OnPreparedListener
```

Register a callback to be invoked when the media file is loaded and ready to go.

Parameters

1 **MediaPlayer.OnPreparedListener**: The callback that will be run

setVideoPath

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

```
public void setVideoPath (String path)
```

Sets video path.

Parameters

path	String: the path of the video.
-------------	---------------------------------------

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

```
public void setVideoURI (Uri uri,  
                        Map <String
```

Sets video URI using specific headers.

Parameters

uri	Uri: the URI of the video.
------------	-----------------------------------

headers	Map: the headers for the URI request. Note that the cross domain redirection is allowed by default, but that can be changed with key/value pairs through the headers parameter with "android-allow-cross-domain-redirect" as the key and "0" or "1" as the value to disallow or allow cross domain redirection.
----------------	--

setVideoURI

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

```
public void setVideoURI (Uri uri)
```

Sets video URI.

Parameters

uri	Uri: the URI of the video.
------------	-----------------------------------

start

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

```
public void start ()
```

stopPlayback

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

```
public void stopPlayback ()
```

suspend

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

```
public void suspend ()
```

Protected methods

onAttachedToWindow

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

```
protected void onAttachedToWindow ()
```

This is called when the view is attached to a window. At this point it has a Surface and will start drawing. Note that this function is guaranteed to be called before `onDraw(android.graphics.Canvas)`.

([https://developer.android.com/reference/android/view/View.html#onDraw\(android.graphics.Canvas\)](https://developer.android.com/reference/android/view/View.html#onDraw(android.graphics.Canvas))), however it may be called any time before the first onDraw -- including before or after `onMeasure(int, int)`.

([https://developer.android.com/reference/android/view/View.html#onMeasure\(int,%20int\)](https://developer.android.com/reference/android/view/View.html#onMeasure(int,%20int))).

onDetachedFromWindow

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

```
protected void onDetachedFromWindow ()
```

This is called when the view is detached from a window. At this point it no longer has a surface for drawing.

onLayout

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

```
protected void onLayout (boolean changed,  
                        int left,  
                        int top,
```

```
int right,  
int bottom)
```

Called from layout when this view should assign a size and position to each of its children. Derived classes with children should override this method and call layout on each of their children.

Parameters

changed	boolean: This is a new size or position for this view
----------------	--

left	int: Left position, relative to parent
-------------	---

top	int: Top position, relative to parent
------------	--

right	int: Right position, relative to parent
--------------	--

bottom	int: Bottom position, relative to parent
---------------	---

onMeasure

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

```
protected void onMeasure (int widthMeasureSpec,  
int heightMeasureSpec)
```

Measure the view and its content to determine the measured width and the measured height. This method is invoked by `measure(int, int)` ([https://developer.android.com/reference/android/view/View.html#measure\(int,%20int\)](https://developer.android.com/reference/android/view/View.html#measure(int,%20int))) and should be overridden by subclasses to provide

accurate and efficient measurement of their contents.

CONTRACT: When overriding this method, you *must* call `setMeasuredDimension(int, int)`.

([https://developer.android.com/reference/android/view/View.html#setMeasuredDimension\(int,%20int\)](https://developer.android.com/reference/android/view/View.html#setMeasuredDimension(int,%20int))) to store the measured width and height of this view. Failure to do so will trigger an `IllegalStateException`, thrown by `measure(int, int)`.

([https://developer.android.com/reference/android/view/View.html#measure\(int,%20int\)](https://developer.android.com/reference/android/view/View.html#measure(int,%20int))). Calling the superclass' `onMeasure(int, int)`.

([https://developer.android.com/reference/android/view/View.html#onMeasure\(int,%20int\)](https://developer.android.com/reference/android/view/View.html#onMeasure(int,%20int))) is a valid use.

The base class implementation of `measure` defaults to the background size, unless a larger size is allowed by the `MeasureSpec`.

Subclasses should override `onMeasure(int, int)` ([https://developer.android.com/reference/android/view/View.html#onMeasure\(int,%20int\)](https://developer.android.com/reference/android/view/View.html#onMeasure(int,%20int))) to provide better measurements of their content.

If this method is overridden, it is the subclass's responsibility to make sure the measured height and width are at least the view's minimum height and width (`getSuggestedMinimumHeight()`).

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

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

Parameters

widthMeasureSpec	int : horizontal space requirements as imposed by the parent. The requirements are encoded with <code>View.MeasureSpec</code> (https://developer.android.com/reference/android/view/View.MeasureSpec.html).
-------------------------	---

heightMeasureSpec	int : vertical space requirements as imposed by the parent. The requirements are encoded with <code>View.MeasureSpec</code> (https://developer.android.com/reference/android/view/View.MeasureSpec.html).
--------------------------	---

Last updated June 6, 2018.



Twitter

Follow @AndroidDev on
Twitter



Google+

Follow Android Developers on
Google+



YouTube

Check out Android Developers
on YouTube