Flutter resumen documentación

# Accesibility

## exclude semantics

A widget that drops all the semantics of its descendants.

When [excluding](https://api.flutter.dev/flutter/widgets/ExcludeSemantics/excluding.html) is true, this widget (and its subtree) is excluded from the semantics tree.

This can be used to hide descendant widgets that would otherwise be reported but that would only be confusing. For example, the material library's [Chip](https://api.flutter.dev/flutter/material/Chip-class.html) widget hides the avatar since it is redundant with the chip label.

### Constructors

[ExcludeSemantics](https://api.flutter.dev/flutter/widgets/ExcludeSemantics/ExcludeSemantics.html)({[Key](https://api.flutter.dev/flutter/foundation/Key-class.html)? key, [bool](https://api.flutter.dev/flutter/dart-core/bool-class.html) excluding = true, [Widget](https://api.flutter.dev/flutter/widgets/Widget-class.html)? child})

Creates a widget that drops all the semantics of its descendants.

const

### Properties

[child](https://api.flutter.dev/flutter/widgets/SingleChildRenderObjectWidget/child.html) → [Widget](https://api.flutter.dev/flutter/widgets/Widget-class.html)?

The widget below this widget in the tree.

finalinherited

[excluding](https://api.flutter.dev/flutter/widgets/ExcludeSemantics/excluding.html) → [bool](https://api.flutter.dev/flutter/dart-core/bool-class.html)

Whether this widget is excluded in the semantics tree.

final

[hashCode](https://api.flutter.dev/flutter/widgets/Widget/hashCode.html) → [int](https://api.flutter.dev/flutter/dart-core/int-class.html)

The hash code for this object.

read-onlyinherited

[key](https://api.flutter.dev/flutter/widgets/Widget/key.html) → [Key](https://api.flutter.dev/flutter/foundation/Key-class.html)?

Controls how one widget replaces another widget in the tree.

finalinherited

[runtimeType](https://api.flutter.dev/flutter/dart-core/Object/runtimeType.html) → [Type](https://api.flutter.dev/flutter/dart-core/Type-class.html)

A representation of the runtime type of the object.

read-onlyinherited

## merge semantics

A widget that merges the semantics of its descendants.

Causes all the semantics of the subtree rooted at this node to be merged into one node in the semantics tree. For example, if you have a widget with a Text node next to a checkbox widget, this could be used to merge the label from the Text node with the "checked" semantic state of the checkbox into a single node that had both the label and the checked state. Otherwise, the label would be presented as a separate feature than the checkbox, and the user would not be able to be sure that they were related.

Be aware that if two nodes in the subtree have conflicting semantics, the result may be nonsensical. For example, a subtree with a checked checkbox and an unchecked checkbox will be presented as checked. All the labels will be merged into a single string (with newlines separating each label from the other). If multiple nodes in the merged subtree can handle semantic gestures, the first one in tree order will be the one to receive the callbacks.

### Constructors

[MergeSemantics](https://api.flutter.dev/flutter/widgets/MergeSemantics/MergeSemantics.html)({[Key](https://api.flutter.dev/flutter/foundation/Key-class.html)? key, [Widget](https://api.flutter.dev/flutter/widgets/Widget-class.html)? child})

Creates a widget that merges the semantics of its descendants.

const

### Properties

[child](https://api.flutter.dev/flutter/widgets/SingleChildRenderObjectWidget/child.html) → [Widget](https://api.flutter.dev/flutter/widgets/Widget-class.html)?

The widget below this widget in the tree.

finalinherited

[hashCode](https://api.flutter.dev/flutter/widgets/Widget/hashCode.html) → [int](https://api.flutter.dev/flutter/dart-core/int-class.html)

The hash code for this object.

read-onlyinherited

[key](https://api.flutter.dev/flutter/widgets/Widget/key.html) → [Key](https://api.flutter.dev/flutter/foundation/Key-class.html)?

Controls how one widget replaces another widget in the tree.

finalinherited

[runtimeType](https://api.flutter.dev/flutter/dart-core/Object/runtimeType.html) → [Type](https://api.flutter.dev/flutter/dart-core/Type-class.html)

A representation of the runtime type of the object.

read-onlyinherited

## semantics

A widget that annotates the widget tree with a description of the meaning of the widgets.

Used by assistive technologies, search engines, and other semantic analysis software to determine the meaning of the application.

See also:

* [SemanticsProperties](https://api.flutter.dev/flutter/semantics/SemanticsProperties-class.html), which contains a complete documentation for each of the constructor parameters that belongs to semantics properties.
* [MergeSemantics](https://api.flutter.dev/flutter/widgets/MergeSemantics-class.html), which marks a subtree as being a single node for accessibility purposes.
* [ExcludeSemantics](https://api.flutter.dev/flutter/widgets/ExcludeSemantics-class.html), which excludes a subtree from the semantics tree (which might be useful if it is, e.g., totally decorative and not important to the user).
* [RenderObject.describeSemanticsConfiguration](https://api.flutter.dev/flutter/rendering/RenderObject/describeSemanticsConfiguration.html), the rendering library API through which the [Semantics](https://api.flutter.dev/flutter/widgets/Semantics-class.html) widget is actually implemented.
* [SemanticsNode](https://api.flutter.dev/flutter/semantics/SemanticsNode-class.html), the object used by the rendering library to represent semantics in the semantics tree.
* [SemanticsDebugger](https://api.flutter.dev/flutter/widgets/SemanticsDebugger-class.html), an overlay to help visualize the semantics tree. Can be enabled using [WidgetsApp.showSemanticsDebugger](https://api.flutter.dev/flutter/widgets/WidgetsApp/showSemanticsDebugger.html) or [MaterialApp.showSemanticsDebugger](https://api.flutter.dev/flutter/material/MaterialApp/showSemanticsDebugger.html).

### Constructors

[Semantics](https://api.flutter.dev/flutter/widgets/Semantics/Semantics.html)({[Key](https://api.flutter.dev/flutter/foundation/Key-class.html)? key, [Widget](https://api.flutter.dev/flutter/widgets/Widget-class.html)? child, [bool](https://api.flutter.dev/flutter/dart-core/bool-class.html) container = false, [bool](https://api.flutter.dev/flutter/dart-core/bool-class.html) explicitChildNodes = false, [bool](https://api.flutter.dev/flutter/dart-core/bool-class.html) excludeSemantics = false, [bool](https://api.flutter.dev/flutter/dart-core/bool-class.html) blockUserActions = false, [bool](https://api.flutter.dev/flutter/dart-core/bool-class.html)? enabled, [bool](https://api.flutter.dev/flutter/dart-core/bool-class.html)? checked, [bool](https://api.flutter.dev/flutter/dart-core/bool-class.html)? mixed, [bool](https://api.flutter.dev/flutter/dart-core/bool-class.html)? selected, [bool](https://api.flutter.dev/flutter/dart-core/bool-class.html)? toggled, [bool](https://api.flutter.dev/flutter/dart-core/bool-class.html)? button, [bool](https://api.flutter.dev/flutter/dart-core/bool-class.html)? slider, [bool](https://api.flutter.dev/flutter/dart-core/bool-class.html)? keyboardKey, [bool](https://api.flutter.dev/flutter/dart-core/bool-class.html)? link, [bool](https://api.flutter.dev/flutter/dart-core/bool-class.html)? header, [bool](https://api.flutter.dev/flutter/dart-core/bool-class.html)? textField, [bool](https://api.flutter.dev/flutter/dart-core/bool-class.html)? readOnly, [bool](https://api.flutter.dev/flutter/dart-core/bool-class.html)? focusable, [bool](https://api.flutter.dev/flutter/dart-core/bool-class.html)? focused, [bool](https://api.flutter.dev/flutter/dart-core/bool-class.html)? inMutuallyExclusiveGroup, [bool](https://api.flutter.dev/flutter/dart-core/bool-class.html)? obscured, [bool](https://api.flutter.dev/flutter/dart-core/bool-class.html)? multiline, [bool](https://api.flutter.dev/flutter/dart-core/bool-class.html)? scopesRoute, [bool](https://api.flutter.dev/flutter/dart-core/bool-class.html)? namesRoute, [bool](https://api.flutter.dev/flutter/dart-core/bool-class.html)? hidden, [bool](https://api.flutter.dev/flutter/dart-core/bool-class.html)? image, [bool](https://api.flutter.dev/flutter/dart-core/bool-class.html)? liveRegion, [bool](https://api.flutter.dev/flutter/dart-core/bool-class.html)? expanded, [int](https://api.flutter.dev/flutter/dart-core/int-class.html)? maxValueLength, [int](https://api.flutter.dev/flutter/dart-core/int-class.html)? currentValueLength, [String](https://api.flutter.dev/flutter/dart-core/String-class.html)? label, [AttributedString](https://api.flutter.dev/flutter/semantics/AttributedString-class.html)? attributedLabel, [String](https://api.flutter.dev/flutter/dart-core/String-class.html)? value, [AttributedString](https://api.flutter.dev/flutter/semantics/AttributedString-class.html)? attributedValue, [String](https://api.flutter.dev/flutter/dart-core/String-class.html)? increasedValue, [AttributedString](https://api.flutter.dev/flutter/semantics/AttributedString-class.html)? attributedIncreasedValue, [String](https://api.flutter.dev/flutter/dart-core/String-class.html)? decreasedValue, [AttributedString](https://api.flutter.dev/flutter/semantics/AttributedString-class.html)? attributedDecreasedValue, [String](https://api.flutter.dev/flutter/dart-core/String-class.html)? hint, [AttributedString](https://api.flutter.dev/flutter/semantics/AttributedString-class.html)? attributedHint, [String](https://api.flutter.dev/flutter/dart-core/String-class.html)? tooltip, [String](https://api.flutter.dev/flutter/dart-core/String-class.html)? onTapHint, [String](https://api.flutter.dev/flutter/dart-core/String-class.html)? onLongPressHint, [TextDirection](https://api.flutter.dev/flutter/dart-ui/TextDirection.html)? textDirection, [SemanticsSortKey](https://api.flutter.dev/flutter/semantics/SemanticsSortKey-class.html)? sortKey, [SemanticsTag](https://api.flutter.dev/flutter/semantics/SemanticsTag-class.html)? tagForChildren, [VoidCallback](https://api.flutter.dev/flutter/dart-ui/VoidCallback.html)? onTap, [VoidCallback](https://api.flutter.dev/flutter/dart-ui/VoidCallback.html)? onLongPress, [VoidCallback](https://api.flutter.dev/flutter/dart-ui/VoidCallback.html)? onScrollLeft, [VoidCallback](https://api.flutter.dev/flutter/dart-ui/VoidCallback.html)? onScrollRight, [VoidCallback](https://api.flutter.dev/flutter/dart-ui/VoidCallback.html)? onScrollUp, [VoidCallback](https://api.flutter.dev/flutter/dart-ui/VoidCallback.html)? onScrollDown, [VoidCallback](https://api.flutter.dev/flutter/dart-ui/VoidCallback.html)? onIncrease, [VoidCallback](https://api.flutter.dev/flutter/dart-ui/VoidCallback.html)? onDecrease, [VoidCallback](https://api.flutter.dev/flutter/dart-ui/VoidCallback.html)? onCopy, [VoidCallback](https://api.flutter.dev/flutter/dart-ui/VoidCallback.html)? onCut, [VoidCallback](https://api.flutter.dev/flutter/dart-ui/VoidCallback.html)? onPaste, [VoidCallback](https://api.flutter.dev/flutter/dart-ui/VoidCallback.html)? onDismiss, [MoveCursorHandler](https://api.flutter.dev/flutter/semantics/MoveCursorHandler.html)? onMoveCursorForwardByCharacter, [MoveCursorHandler](https://api.flutter.dev/flutter/semantics/MoveCursorHandler.html)? onMoveCursorBackwardByCharacter, [SetSelectionHandler](https://api.flutter.dev/flutter/semantics/SetSelectionHandler.html)? onSetSelection, [SetTextHandler](https://api.flutter.dev/flutter/semantics/SetTextHandler.html)? onSetText, [VoidCallback](https://api.flutter.dev/flutter/dart-ui/VoidCallback.html)? onDidGainAccessibilityFocus, [VoidCallback](https://api.flutter.dev/flutter/dart-ui/VoidCallback.html)? onDidLoseAccessibilityFocus, [Map](https://api.flutter.dev/flutter/dart-core/Map-class.html)<[CustomSemanticsAction](https://api.flutter.dev/flutter/semantics/CustomSemanticsAction-class.html), [VoidCallback](https://api.flutter.dev/flutter/dart-ui/VoidCallback.html)>? customSemanticsActions})

Creates a semantic annotation.

[Semantics.fromProperties](https://api.flutter.dev/flutter/widgets/Semantics/Semantics.fromProperties.html)({[Key](https://api.flutter.dev/flutter/foundation/Key-class.html)? key, [Widget](https://api.flutter.dev/flutter/widgets/Widget-class.html)? child, [bool](https://api.flutter.dev/flutter/dart-core/bool-class.html) container = false, [bool](https://api.flutter.dev/flutter/dart-core/bool-class.html) explicitChildNodes = false, [bool](https://api.flutter.dev/flutter/dart-core/bool-class.html) excludeSemantics = false, [bool](https://api.flutter.dev/flutter/dart-core/bool-class.html) blockUserActions = false, required [SemanticsProperties](https://api.flutter.dev/flutter/semantics/SemanticsProperties-class.html) properties})

Creates a semantic annotation using [SemanticsProperties](https://api.flutter.dev/flutter/semantics/SemanticsProperties-class.html).

const

### Properties

[blockUserActions](https://api.flutter.dev/flutter/widgets/Semantics/blockUserActions.html) → [bool](https://api.flutter.dev/flutter/dart-core/bool-class.html)

Whether to block user interactions for the rendering subtree.

final

[child](https://api.flutter.dev/flutter/widgets/SingleChildRenderObjectWidget/child.html) → [Widget](https://api.flutter.dev/flutter/widgets/Widget-class.html)?

The widget below this widget in the tree.

finalinherited

[container](https://api.flutter.dev/flutter/widgets/Semantics/container.html) → [bool](https://api.flutter.dev/flutter/dart-core/bool-class.html)

If [container](https://api.flutter.dev/flutter/widgets/Semantics/container.html) is true, this widget will introduce a new node in the semantics tree. Otherwise, the semantics will be merged with the semantics of any ancestors (if the ancestor allows that).

final

[excludeSemantics](https://api.flutter.dev/flutter/widgets/Semantics/excludeSemantics.html) → [bool](https://api.flutter.dev/flutter/dart-core/bool-class.html)

Whether to replace all child semantics with this node.

final

[explicitChildNodes](https://api.flutter.dev/flutter/widgets/Semantics/explicitChildNodes.html) → [bool](https://api.flutter.dev/flutter/dart-core/bool-class.html)

Whether descendants of this widget are allowed to add semantic information to the [SemanticsNode](https://api.flutter.dev/flutter/semantics/SemanticsNode-class.html) annotated by this widget.

final

[hashCode](https://api.flutter.dev/flutter/widgets/Widget/hashCode.html) → [int](https://api.flutter.dev/flutter/dart-core/int-class.html)

The hash code for this object.

read-onlyinherited

[key](https://api.flutter.dev/flutter/widgets/Widget/key.html) → [Key](https://api.flutter.dev/flutter/foundation/Key-class.html)?

Controls how one widget replaces another widget in the tree.

finalinherited

[properties](https://api.flutter.dev/flutter/widgets/Semantics/properties.html) → [SemanticsProperties](https://api.flutter.dev/flutter/semantics/SemanticsProperties-class.html)

Contains properties used by assistive technologies to make the application more accessible.

final

[runtimeType](https://api.flutter.dev/flutter/dart-core/Object/runtimeType.html) → [Type](https://api.flutter.dev/flutter/dart-core/Type-class.html)

A representation of the runtime type of the object.

read-onlyinherited