

[AppKit](#) / Touch Bar

API Collection

Touch Bar

Display interactive content and controls in the Touch Bar.

Topics

Essentials

{ } Integrating a Toolbar and Touch Bar into Your App

Provide users quick access to your app's features from a toolbar and corresponding Touch Bar.

{ } Creating and Customizing the Touch Bar

Adopt Touch Bar support by displaying interactive content and controls for your macOS apps.

`class NSTouchBar`

An object that provides dynamic contextual controls in the Touch Bar of supported models of MacBook Pro.

`protocol NSTouchBarDelegate`

A protocol that allows you to provide the items for a bar dynamically.

`protocol NSTouchBarProvider`

A protocol that an object adopts to create a bar object in your app.

Touch Bar items

`class NSTouchBarItem`

A UI control shown in the Touch Bar on supported models of MacBook Pro.

```
class NSCandidateListTouchBarItem
```

A bar item that, along with its delegate, provides a list of textual suggestions for the current text view.

```
class NSColorPickerTouchBarItem
```

A bar item that provides a system-defined color picker.

```
class NSCustomTouchBarItem
```

A bar item that contains a responder of your choice, such as a view, a button, or a scrubber.

```
class NSGroupTouchBarItem
```

A bar item that provides a bar to contain other items.

```
class NSPopoverTouchBarItem
```

A bar item that provides a two-state control that can expand into its second state, showing the contents of a bar that it owns.

```
class NSSharingServicePickerTouchBarItem
```

A bar item that, along with its delegate, provides a list of objects eligible for sharing.

```
class NSSliderTouchBarItem
```

A bar item that provides a slider control for choosing a value in a range.

```
class NSS stepperTouchBarItem
```

A bar item that provides a stepper control for incrementing or decrementing a value.

```
class NSUserInterfaceCompressionOptions
```

An object that specifies how user interface elements resize themselves when space is constrained.

```
class NSButtonTouchBarItem
```

A bar item that provides a button.

```
class NS PickerTouchBarItem
```

A bar item that provides a picker control with multiple options.

```
enum ControlRepresentation
```

Constants that specify display styles for picker bar items.

```
enum SelectionMode
```

Constants that specify selection modes for picker bar items.

Scrubbers

`class NSScrubber`

A customizable item picker control for the Touch Bar.

`protocol NSScrubberDataSource`

A set of methods that a scrubber data source object implements to provide items to the scrubber from an associated data collection in your app.

`protocol NSScrubberDelegate`

A set of methods that a scrubber delegate implements to respond to user interactions.

Scrubber items

`class NSScrubberItemView`

An item at a specific index position in the scrubber.

`class NSScrubberArrangedView`

An abstract base class for the views whose layout is managed by a scrubber.

`class NSScrubberImageItemView`

A concrete view subclass for displaying images in a scrubber items.

`class NSScrubberSelectionStyle`

An abstract class that provides decorative accessory views for selected and highlighted items within a scrubber control.

`class NSScrubberSelectionView`

An abstract base class for specifying the appearance of a highlighted or selected item in a scrubber.

`class NSScrubberTextItemView`

A concrete view subclass for displaying text for an item in a scrubber.

Scrubber layouts

`class NSScrubberFlowLayout`

A concrete layout object that arranges items end-to-end in a linear strip.

`protocol NSScrubberFlowLayoutDelegate`

A protocol that a scrubber delegate can adopt to provide the size of an item.

class NSScrubberProportionalLayout

A concrete layout object that sizes each item to some fraction of the scrubber's visible size.

class NSScrubberLayoutAttributes

The layout of a scrubber item.

class NSScrubberLayout

An abstract class that describes the layout of items within a scrubber control.

See Also

User Interactions

☰ Mouse, Keyboard, and Trackpad

Handle events related to mouse, keyboard, and trackpad input.

☰ Menus, Cursors, and the Dock

Implement menus and cursors to facilitate interactions with your app, and use your app's Dock tile to convey updated information.

☰ Gestures

Encapsulate your app's event-handling logic in gesture recognizers so that you can reuse that code throughout your app.

☰ Drag and Drop

Support the direct manipulation of your app's content using drag and drop.

☰ Accessibility for AppKit

Make your AppKit apps accessible to everyone who uses macOS.