API Collection

Drag and Drop

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

Overview

With very little programming on your part, custom-view objects can be dragged and dropped anywhere. Objects become part of this dragging mechanism by conforming to dragging protocols: Draggable objects conform to the `NSDraggingSource` protocol, and destination objects (that is, receivers of a drop) conform to the `NSDraggingDestination` protocol. AppKit hides all the details of tracking the cursor and displaying the dragged image.

> **Note**
>
> To learn how to adopt drag and drop in your iOS app, see Drag and drop.

To learn how to use drag and drop for an image view, see Supporting Drag and Drop Through File Promises. To use drag and drop in a table view, see Supporting Table View Drag and Drop Through File Promises. For an example of drag and drop in a collection view, see Supporting Collection View Drag and Drop Through File Promises, and for an outline view: Navigating Hierarchical Data Using Outline and Split Views.

Topics

Drag Sources

Originate content from a drag source by creating items to represent that content.

`protocol` NSDraggingSource

A set of methods that are implemented by the source object in a dragging session.

`class` NSDraggingItem

A single dragged item within a dragging session.

`class` NSDraggingSession

The encapsulation of a drag-and-drop action that supports modification of the drag while in progress.

`class` NSDraggingImageComponent

A single object in a dragging item.

Drop Targets

Receive dragged content in your app's objects.

`protocol` NSDraggingDestination

A set of methods that the destination object (or recipient) of a dragged image must implement.

`protocol` NSDraggingInfo

A set of methods that supply information about a dragging session.

`protocol` NSSpringLoadingDestination

A set of methods that the destination object (or recipient) of a dragged object can implement to support spring-loading.

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.

- Touch Bar

  Display interactive content and controls in the Touch Bar.

- Accessibility for AppKit

  Make your AppKit apps accessible to everyone who uses macOS.