

Framework

CarPlay

Integrate CarPlay in apps related to audio, communication, navigation, parking, EV charging, food ordering, and more.

iOS 12.0+ | iPadOS 12.0+ | Mac Catalyst 14.0+

Overview

Use the CarPlay framework to create an in-car experience for your app. The framework provides templates for building a version of your app's interface suitable for presentation on a vehicle's displays. Add the templates you want to your app and customize them to suit your content. You control the content of the templates, but the framework controls certain aspects of the template interface elements, such as the touch target size, font size, font color, and highlights.

CarPlay features run when the current device supports CarPlay and when that device is connected to an appropriately equipped vehicle. CarPlay handles variations in vehicle systems, letting you focus on your content. When a person runs your app from their vehicle, the system generates and hosts your app's interface for you. If the device doesn't support CarPlay, the system doesn't try to access your app's CarPlay features.

You can use other technologies to drive portions of your app's CarPlay interface. Messaging apps can include [SiriKit](#) support to allow someone to read or send messages. VoIP apps can use [CallKit](#) to manage incoming and outgoing calls, often in combination with SiriKit call support. Navigation apps can include [MapKit](#) support.

Related Sessions from WWDC20

Session 10635: [Accelerate Your App with CarPlay](#)

Topics

CarPlay Integration



Requesting CarPlay Entitlements

Configure your CarPlay-enabled app with the entitlements it requires.



Displaying Content in CarPlay

Use scenes to present your app's content on the vehicle's built-in screen.



Supporting Previous Versions of iOS

Make your CarPlay-enabled apps compatible with older system versions, such as iOS 13 and earlier.



Using the CarPlay Simulator

Configure Simulator to run and debug your CarPlay-enabled app.

```
class CPTemplateApplicationScene
```

A CarPlay scene that controls your app's user interface.

```
protocol CPTemplateApplicationSceneDelegate
```

The methods for responding to the life cycle events of your app's scene.

```
class CPSessionConfiguration
```

An object that provides vehicle properties and configuration for the CarPlay environment.

General Purpose Templates

Display your app's content using a variety of templates that provide a consistent CarPlay layout and appearance.

```
class CPListTemplate
```

A template that displays and manages a list of items.

```
class CPGridTemplate
```

A template that displays and manages a grid of items.

```
class CPTabBarTemplate
```

A container template that displays and manages other templates, presenting them as tabs.

```
class CPTemplate
```

An abstract base class for interface templates.

`protocol CBarButtonProviding`

The methods that templates use to provide buttons for the navigation bar.

Audio

Templates that are available exclusively to apps with the audio entitlement.

`{}` Integrating CarPlay with Your Music App

Configure your music app to work with CarPlay by displaying a custom UI.

`class CPNowPlayingTemplate`

A shared system template that displays Now Playing information.

Instrument cluster

Symbols that are available for managing the instrument cluster.

`class CPInstrumentClusterController`

`protocol CPInstrumentClusterControllerDelegate`

`class CPTemplateApplicationInstrumentClusterScene`

`protocol CPTemplateApplicationInstrumentClusterSceneDelegate`

Navigation

Symbols that are available exclusively to apps with the navigation entitlement.

`{}` Integrating CarPlay with Your Navigation App

Configure your navigation app to work with CarPlay by displaying your custom map and directions.

`class CPTemplateApplicationDashboardScene`

A CarPlay scene that controls your app's dashboard navigation window.

`protocol CPTemplateApplicationDashboardSceneDelegate`

The methods for responding to the life-cycle events of your navigation app's dashboard scene.

`class CPMAPTemplate`

A template that displays a navigation overlay that your app draws on the map.

`class CPSearchTemplate`

A template that provides the ability to search for a destination and see a list of search results.

`class CPVoiceControlTemplate`

A template that displays a voice control indicator during audio input.

Location and Information

Templates that are available exclusively to apps with the parking, EV-charging, or food-ordering entitlements.

`class CPPointOfInterestTemplate`

A template that displays a map with selectable points of interest.

`class CPIInformationTemplate`

A template that provides information for a point of interest, food order, parking location, or charging location.

`class CPTextButton`

A button that displays a stylized title.

`{}` Integrating CarPlay with your quick-ordering app

Configure your food-ordering app to work with CarPlay.

Maneuvers

`class CPManeuver`

An object that describes a single navigation instruction.

`enum CPManeuverState`

Values that describe the state of a maneuver.

`enum CPManeuverType`

Values that describe types of navigation maneuvers.

Routes, lanes and junctions

`class CPRouteInformation`

A class that describes the characteristic elements of a route.

`class CPLane`

A class that describes characteristics of a lane on a roadway.

`class` `CPLaneGuidance`

A class that provides information that describes the number of lanes on a roadway and navigation instruction variants.

`enum` `CPLaneStatus`

Values that describe the status or preferability of a lane.

`enum` `CPJunctionType`

Values that represent types of roadway junctions.

Communication

Templates that are available exclusively to apps with the communication entitlement.

`class` `CPContactTemplate`

A template that displays information about a person or a business.

Actions and Alerts

`class` `CPActionSheetTemplate`

A template that displays a modal action sheet.

`class` `CPAlertTemplate`

A template that displays a modal alert.

`class` `CPAlertAction`

An object that encapsulates an action the user can perform on an action sheet or alert.

Related Types

`class` `CPButton`

A button that displays an image and invokes a handler when the user taps it.

`class` `CPImageSet`

Light and dark representations of an image.

`let` `CarPlayErrorDomain`: `String`

The domain that CarPlay uses for any errors it provides.

Deprecated

⋮ Deprecated Symbols

Symbols that the CarPlay framework no longer supports.

Reference

⋮ CarPlay Enumerations

⋮ CarPlay Constants

Classes

`class CPlaylistImageRowItemCardElement`

`class CPlaylistImageRowItemCondensedElement`

`class CPlaylistImageRowItemElement`

Abstract superclass for a a row item element object.

`class CPlaylistImageRowItemGridElement`

`class CPlaylistImageRowItemImageGridElement`

`class CPlaylistImageRowItemRowElement`

`class CPMessagesGridItemConfiguration`

`class CPNowPlayingMode`

`class CPNowPlayingModeSports`

The sports mode represents a layout for now playing suited to live-streaming or recorded playback of a sporting event that features exactly two teams.

`class CPNowPlayingSportsClock`

A representation of the amount of time elapsed so far in this event, for events where the clock counts UP.

`class CPNowPlayingSportsEventStatus`

A representation of the status of a sporting event.

`class CPNowPlayingSportsTeam`

A representation of a sports team for the now playing screen, in sports that have exactly two teams.

```
class CPNowPlayingSportsTeamLogo
```

A logo image or, if no image is available, an abbreviation or initialism for this team.

Variables

```
let CPMaximumMessageItemLeadingDetailTextImageSize: CGSize
```

Maximum size of an image for the detailed text leading image.

Functions

```
func NSStringFromCPJunctionType(CPJunctionType) -> String!
```

```
func NSStringFromCPLaneStatus(CPLaneStatus) -> String!
```

```
func NSStringFromCPManeuverType(CPManeuverType) -> String!
```

```
func NSStringFromCPTrafficSide(CPTrafficSide) -> String!
```