

[AVKit](#) / [AVPictureInPictureController](#)

Class

AVPictureInPictureController

A controller that responds to user-initiated Picture in Picture playback of video in a floating, resizable window.

iOS 9.0+ | iPadOS 9.0+ | Mac Catalyst 13.1+ | macOS 10.15+ | tvOS 14.0+ | visionOS 1.0+

```
class AVPictureInPictureController
```

Mentioned in

- 📄 Adopting Picture in Picture in a Custom Player
- 📄 Adopting Picture in Picture for video calls

Overview

To use Picture in Picture, you need to configure your app to support background audio playback. See [Configuring your app for media playback](#) for more details.

Before presenting a user interface to start Picture in Picture, call the [isPictureInPictureSupported\(\)](#) method to determine if the current device supports the feature, and check the [isPictureInPicturePossible](#) property value to determine whether PiP is possible in the current context.

Important

The framework doesn't support subclassing [AVPictureInPictureController](#).

Topics

Creating a Controller

```
init(contentSource: AVPictureInPictureController.ContentSource)
```

Creates a Picture in Picture controller with a content source.

```
convenience init?(playerLayer: AVPlayerLayer)
```

Creates a Picture in Picture controller with a player layer.

Configuring the Content Source

```
var contentSource: AVPictureInPictureController.ContentSource?
```

The source of the controller's content.

```
class ContentSource
```

An object that represents the source of the content to present in Picture in Picture.

Accessing the Player Layer

```
var playerLayer: AVPlayerLayer
```

The layer that displays the video content.

Configuring Playback Behavior

```
var requiresLinearPlayback: Bool
```

A Boolean value that determines whether the controller allows the user to skip media content.

Accessing the Delegate Object

```
var delegate: (any AVPictureInPictureControllerDelegate)?
```

A delegate object for a Picture in Picture controller.

```
protocol AVPictureInPictureControllerDelegate
```

A protocol to adopt to respond to Picture in Picture events.

Accessing Picture in Picture State

```
class func isPictureInPictureSupported() -> Bool
```

Returns a Boolean value that indicates whether the current device supports Picture in Picture.

```
var isPictureInPicturePossible: Bool
```

A Boolean value that indicates whether Picture in Picture playback is currently possible.

```
var isPictureInPictureActive: Bool
```

A Boolean value that indicates whether the Picture in Picture window is onscreen.

```
var isPictureInPictureSuspended: Bool
```

A Boolean value that indicates whether the system suspends the controller's Picture in Picture window.

Controlling Picture in Picture Playback

```
var canStopPictureInPicture: Bool
```

A Boolean value that indicates whether Picture in Picture is active and is able to stop.

```
var canStartPictureInPictureAutomaticallyFromInline: Bool
```

A Boolean value that indicates whether Picture in Picture starts automatically when the controller embeds its content inline and the app transitions to the background.

```
func startPictureInPicture()
```

Starts Picture in Picture, if possible.

```
func stopPictureInPicture()
```

Stops Picture in Picture, if active.

Retrieving Picture in Picture Template Images

```
class var pictureInPictureButtonStartImage: UIImage
```

A system-default template image for the button that starts Picture in Picture in your app.

```
class var pictureInPictureButtonStopImage: UIImage
```

A system-default template image for the button that stops Picture in Picture in your app.

```
class func pictureInPictureButtonStartImage(compatibleWith: UITraitCollection?) -> UIImage
```

Returns a system-default template image that's compatible with a trait collection for the button that starts Picture in Picture in your app.

```
class func pictureInPictureButtonStopImage(compatibleWith: UITraitCollection?) -> UIImage
```

Returns a system-default template image that's compatible with a trait collection for the button that stops Picture in Picture in your app.

Instance Methods

```
func invalidatePlaybackState()
```

Invalidate the controller's current playback state and fetches the updated state from the sample buffer playback delegate object.

Relationships

Inherits From

NSObject

Conforms To

CVarArg
CustomDebugStringConvertible
CustomStringConvertible
Equatable
Hashable
NSObjectProtocol

See Also

Picture in Picture

{} Adopting Picture in Picture Playback in tvOS

Add advanced multitasking capabilities to your video apps by using Picture in Picture playback in tvOS.

 Adopting Picture in Picture in a Standard Player

Add Picture in Picture (PiP) playback to your app using a player view controller.

 Adopting Picture in Picture in a Custom Player

Add controls to your custom player user interface to invoke Picture in Picture (PiP) playback.

 Adopting Picture in Picture for video calls

Add multitasking capability to your video-call apps by using Picture in Picture (PiP).

 Accessing the camera while multitasking on iPad

Operate the camera in Split View, Slide Over, Picture in Picture, and Stage Manager modes.