

---

[Bundle Resources](#) / Information Property List

## Property List

# Information Property List

A resource containing key-value pairs that identify and configure a bundle.

iOS 2.0+ | iPadOS 2.0+ | Mac Catalyst 13.0+ | macOS 10.0+ | tvOS 9.0+ | visionOS 1.0+ | watchOS 2.0+

---

## Details

Type  
Object

## Mentioned in

 Placing content in a bundle

## Discussion

Bundles, which represent executables of different kinds, contain an information property list file. This collection of key-value pairs specifies how the system interprets the associated bundle. Some key-value pairs characterize the bundle itself, and others configure the app, framework, or other entity that the bundle represents. Some keys are required, and others are specific to particular features of the executable.

The information property list file always has the name `Info.plist`. The file name is case-sensitive and must begin with a capital letter I. Its location within the bundle depends on both the bundle type and the platform. For example, iOS app bundles store the file in the bundle's root directory, whereas macOS app bundles place the `Info.plist` file in the `Contents` directory.

You typically rely on Xcode to create the information property list file for your apps and other executable targets, as described in [Managing your app's information property list values](#). When you let Xcode prepare your bundle's information property list, you don't need to edit the file directly. For information on manually editing property list files, see [Editing property list files](#).

To access values in an information property list programmatically, you use an instance of the [Bundle](#) class, which represents a bundle on disk. You can get the value for a few common keys by accessing properties of the bundle instance. For example, the [bundleIdentifier](#) property contains the value associated with the [CFBundleIdentifier](#) key. You can obtain the value for an arbitrary key using the [object\(forInfoDictionaryKey:\)](#) method.

---

# Topics

## Essentials

- 📄 Managing your app's information property list values

Customize the information property list values for your app using Xcode.

## Core settings

- ☰ Bundle configuration

Define basic characteristics of a bundle, like its name, type, and version.

- ☰ User interface

Configure an app's scenes, storyboards, icons, fonts, and other user interface elements.

- ☰ App execution

Control app launch, execution, and termination.

## Services

- ☰ Protected resources

Control an app's access to protected system services and user data.

- ☰ Data and storage

Regulate documents, URLs, and other kinds of data movement and storage.

- ☰ App services

Configure services provided by the app, like support for giving directions or using game controllers.

- ☰ Kernel and drivers

Configure device drivers provided by the app.

## AVFoundation

### NSAlwaysAllowMicrophoneModeControl

A Boolean value that indicates if a person can configure a microphone mode regardless of whether the microphone is in an active state.

## BrowserEngineKit

### BEEmbeddedWebBrowserEngine

### BEEmbeddedWebBrowserEngineVersion

## Dictionaries

### IOParentMatch

### IOPropertyMatch

The device-specific keys the system must match in order to use your driver.

### NSPinnedCAIdentities

A list of allowed Certificate Authority certificates for a given domain name.

### NSPinnedLeafIdentities

A list of allowed leaf certificates for a given domain name.

## Type Aliases

### IOClass

The name of the class to instantiate from your driver.

### IOMatchCategory

### IONameMatch

One or more strings that contain the names of possible provider objects in the system registry.

### IOPathMatch

### IOProviderClass

The name of the class that your driver expects to provide the implementation for its provider object.

## **IOResourceMatch**

One or more system-specific or device-specific resources that your driver requires.

## **IOUserClass**

The name of your driver's main class, which is the entry point for interacting with your driver's code.

## **IOUserClientClass**

The name of the class to instantiate when the system requires a client connection to the driver.

## **IOUserServerName**

The name that the system uses to facilitate communication between your driver and other clients.

## **NSExceptionAllowsInsecureHTTPLoads**

A Boolean value indicating whether to allow insecure HTTP loads.

## **NSExceptionMinimumTLSVersion**

The minimum Transport Layer Security (TLS) version for network connections.

## **NSExceptionRequiresForwardSecrecy**

A Boolean value indicating whether to override the perfect forward secrecy requirement.

## **NSIncludesSubdomains**

A Boolean value that indicates whether to extend the configuration to subdomains of the given domain.

## **NSRequiresCertificateTransparency**

An obsolete Boolean value indicating whether to require Certificate Transparency.

## **OSAScriptingDefinition**

## **ProvidesOneTimeCodes**

Allows this credential provider to show up in one-time-code text fields, to allowing filling time-based verification codes.

## **ProvidesTextToInsert**

Allows this credential provider to show up in the system AutoFill context menu, so that it can fill text in any text field.

## **SupportsConditionalPasskeyRegistration**

Indicates that the credential provider supports automatic passkey upgrades.

---

## See Also

### Property Lists

#### Entitlements

Key-value pairs that grant an executable permission to use a service or technology.

#### ☰ Privacy manifest files

Describe the data your app or third-party SDK collects and the required reasons APIs it uses.