

[App Clips](#) / Choosing the right functionality for your App Clip

Article

Choosing the right functionality for your App Clip

Review frameworks available to App Clips and identify functionality that makes a great App Clip.



Overview

An App Clip is a lightweight version of your app that offers some of its functionality when and where it's needed, or gives people a way to try a demo version of your app. App Clips offer a focused feature set, and are designed to launch instantly, protect user privacy, and preserve resources. As a result, an App Clip comes with some limitations. Before you create your App Clip, first review the technology available, and identify the functionality that makes a great App Clip.

Note

Your full app can offer multiple App Clip experiences, but you have to package them as a single App Clip target. Additionally, the full app must include the same functionality as the App Clip.

Keep your App Clip within size limitations

To ensure a fast launch experience, App Clips must be small. Aim to keep your App Clip binary well below the applicable limit:

iOS version	Maximum size of the uncompressed App Clip binary
iOS 15 and earlier	10 MB

iOS version	Maximum size of the uncompressed App Clip binary
iOS 16 and earlier	15 MB
iOS 17 and later	100 MB, with additional requirements and limitations (see below)

On devices running iOS 17 and later, the uncompressed App Clip binary can be up to 100 MB in size if you meet the following conditions:

- The App Clip only supports digital invocations — for example, from your website or Spotlight search.
- The App Clip doesn't support physical invocations such as App Clip Codes, QR codes, or NFC tags.
- People use your App Clip in situations where a reliable internet connection is likely; for example, at home.
- Your App Clip doesn't support iOS 16 and earlier.

Additionally, you can use the App Clip demo link that App Store Connect generates to use the 100 MB size limit and support physical invocations from App Clip Codes, NFC tags, and QR codes.

For more information, refer to [Configuring App Clip experiences](#) and [Verify the size of your App Clip](#).

If your App Clip needs to download additional assets; for example, if you offer a demo version of your game; use [Background Assets](#) to download additional assets. For more information, refer to [Download additional assets](#).

Review available frameworks and APIs

App Clips make use of [SwiftUI](#) and [UIKit](#), and have access to the same frameworks as your full app. However, the following frameworks provide no functionality at runtime: [App Intents](#), [Assets Library](#), [Background Tasks](#), [CallKit](#), [CareKit](#), [Contacts](#), [Contacts UI](#), [Core Motion](#), [EventKit](#), [EventKit UI](#), [File Provider](#), [File Provider UI](#), [HealthKit](#), [HomeKit](#), [Media Player](#), [Messages](#), [Message UI](#), [Nearby Interaction](#), [PhotoKit](#), [ResearchKit](#), [SensorKit](#), and [Speech](#).

For most unavailable frameworks, using them in an App Clip doesn't result in compile-time errors, but their APIs return values that indicate unavailability, empty data, or error codes at runtime. For example, HealthKit's [isHealthDataAvailable\(\)](#) returns `false` when you call it from an App Clip.

App Clips can't perform background activity. For example, they can't make use of:

- Background networking with [URLSession](#)

- Functionality enabled by the Background Modes capability as described in [Pushing background updates to your App](#)
- The ability to maintain Bluetooth connections while the App Clip isn't in use

Some frameworks are available to App Clips but offer only limited functionality, or using them requires special consideration:

Advanced networking features and low-level networking APIs

Advanced networking features like [Bonjour](#) and low-level networking APIs like [CFSocket](#) or POSIX functions aren't available to App Clips. Instead, use [URLSession](#) or the [Network](#) framework.

App extensions

App Clips can't include app extensions, but they can include a widget extension to offer Live Activities. For more information, refer to [Offering Live Activities with your App Clip](#).

[Core Telephony](#)

Functionality provided by [Core Telephony](#) is available to App Clips. However, they can't provision cellular plan eSIMs or use functionality that carrier apps with suitable entitlements use. For example, an App Clip can't use [CTCellularPlanProvisioning](#) and [CTCellularPlanProvisioningRequest](#).

[CloudKit](#)

[CloudKit](#) isn't available to App Clips in iOS 14 or 15. Starting with iOS 16, App Clips can read their public iCloud database. However, App Clips can't write data to a public database or use private or shared containers. Additionally, they can't use iCloud Documents or iCloud key-value storage. To learn more about using CloudKit in your App Clip, refer to the [Access your public iCloud database](#) section of [Sharing data between your App Clip and your full app](#).

Face ID

App Clips can't use Face ID because the [NSFaceIDUsageDescription](#) entitlement isn't available to them. However, you can use the [Local Authentication](#) framework to authenticate people with Touch ID.

Note that App Clips may configure Wi-Fi networks using the [Hotspot Configuration Entitlement](#). Additionally, to connect to an authentication provider, they may initialize an [ASWebAuthenticationSession](#) using `init(url:callback:completionHandler:)`.

Preserve user privacy

App Clips come with limitations that help to protect user privacy and prevent user tracking across apps and App Clips, for example:

- Functionality provided by [SKAdNetwork](#) isn't available.
- App Clips can't request authorization to track a person with [App Tracking Transparency](#).

- Both `name` and `identifierForVendor` return an empty string.
- App Clips can't request continuous location access. However, you can call `requestWhenInUseAuthorization()` to request the When In Use authorization, which resets automatically the next day at 4:00 a.m.
- In iOS 17 and later, App Clips can request the `Pass Type IDs Entitlement` to read passes stored in the Wallet app. On devices that run iOS 16 or earlier, where App Clips can't read passes stored in the Wallet app, App Clips can add a pass to the Wallet app and check if this pass is already present. For more information, refer to [Checking Whether a Pass Is in the Library](#).
- App Clips can't share data with any other app except its corresponding full app. For more information, refer to [Sharing data between your App Clip and your full app](#).

To help protect user data, App Clips can't access:

- Apple Music and Media
- Data from apps like Calendar, Contacts, Files, Health, Messages, Reminders, and Photos
- Motion and fitness data

Reserve certain functionality for your full app

App Clips that aren't demo versions of full apps provide an in-the-moment experience and focus on offering the quickest possible solution to an everyday task, so some functionality works best in your full app. If your App Clip offers an in-the-moment experience, reserve the following functionality for the full app:

- App extensions
- Customization and settings, for example, creation of a settings bundle
- Data handoff and document opening
- In-app purchases
- Low-level UNIX functionality, for example, BSD notifications
- Multiple scenes on iPad
- On-demand resources and [Background Assets](#)
- Promoting other apps
- Registration of custom URL schemes
- Requests for reviews of the full app by using StoreKit's `requestReview(in:)` method

- Searching for paired Bluetooth devices
-

See Also

Essentials

-  Configuring App Clip experiences

Review how people launch your App Clip with invocation URLs, default and demo links, and advanced App Clip experiences.

-  App Clips updates

Learn about important changes in App Clips.