

[Wi-Fi Aware](#) / Adopting Wi-Fi Aware

Article

Adopting Wi-Fi Aware

Add entitlements and declare your app's services.



Overview

The Wi-Fi Aware™ technology enables devices to securely discover, pair, and communicate with nearby devices without an internet connection or access point. Your app can use the Wi-Fi Aware framework to securely establish peer-to-peer (P2P) connections between Wi-Fi devices.

If you intend your app to use Wi-Fi Aware capabilities, you need to add the `com.apple.developer.wifi-aware` entitlement. The value of the entitlement is an array of capability strings, where the strings specify the Wi-Fi Aware operations and API capabilities your app intends to use.

Capability string	Permission description
Publish	Publish a service and allow incoming connections.
Subscribe	Subscribe to a service and make outgoing connections.

Declare services

Services are the specific functionality and protocols that your app can either provide to other devices or consume from other devices. There are two supported service roles in Wi-Fi Aware:

- **Publisher:** Your app hosts the service and acts as a server that allows incoming connections from other paired devices. For example, your app publishes its availability to other devices to print documents.

- **Subscriber:** Your app uses the service and acts as the client that makes outgoing connections to other paired devices. For example, your app finds friends to connect to in a game.

Wi-Fi Aware can support multiple services simultaneously, and can also act as a publisher, subscriber, or both. As such, your app can:

- Publish one or more services.
- Subscribe to one or more services.
- Simultaneously publish and subscribe the same services.
- Publish one set of services, while subscribing to a different set of services.

Note

Your app can only publish a given service at most once per device, so it can't publish multiple instances of the same service. To minimize resource cost, avoid simultaneously subscribing to the same service multiple times.

Declare the services you want to use with the WiFiAwareServices key in your app's Information property list. The key's value is the dictionary where *keys* are names of services your app uses and *values* are dictionaries of service configuration information, which is in the Info pane of the target editor in Xcode.

Each key under the WiFiAwareServices entry is the fully qualified name of a service as it's sent over the air. It's populated in the name field of the resulting [WAPublishableService](#) and [WASubscribableService](#) structures in the API. For a service name to be valid, it must conform to the rules in [RFC 6763](#) and [RFC 6335](#).

The service string needs to have a unique name that:

- Only uses characters such as a–z, A–Z, 0–9, and hyphen (-)
- Uses at least one letter a–z or A–Z
- Doesn't start or end with a hyphen (-)
- Doesn't exceed 15 characters

The fully qualified service name string needs to have:

- An underscore (_) before the name of the component
- A dot (.) separator
- A protocol suffix of _tcp or _udp

The following examples show service names that follow the naming conventions:

Name component	Protocol component	Fully qualified service name string in Info.plist
example-service	_tcp	_example-service._tcp
example-service	_udp	_example-service._udp

Use the exact same name in the Info.plist. Invalid service names in the Info.plist cause your app to crash.

Configure a service

For each service name key in the WiFiAwareServices dictionary, there's a corresponding value dictionary. This dictionary contains a set of properties that configure each individual service. The configuration includes whether the service is published, subscribed, or both:

Configuration key (String)	Configuration value	Use
Publishable	An empty dictionary	Publishes the configured service
Subscribable	An empty dictionary	Subscribes the configured service

Important

One or both of the Publishable and Subscribable keys must be present. If neither key is present, the framework crashes your app.

The following is an example of an Info.plist that declares a single service _example-service._tcp that's both published and subscribed:

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/Prop
<plist version="1.0">
<dict>
  <key>CFBundleIdentifier</key>
  <string>com.example.MyApp</string>
  <key>WiFiAwareServices</key>
  <dict>
    ...
  </dict>
</dict>

```

```
<key>_example-service._tcp</key>
<dict>
    <key>Publishable</key>
    <dict/>
    <key>Subscribable</key>
    <dict/>
</dict>
</dict>
</plist>
```

Access API services

After you declare the services your app uses in `Info.plist`, your app can access those services with the Wi-Fi Aware API. For each Publishable service, the system creates a [WAPublishableService](#) available in `allServices`. Similarly, for each Subscribable service, the system creates a [WASubscribableService](#) available in `allServices`.

See Also

Essentials

{} Building peer-to-peer apps

Communicate with nearby devices over a secure, high-throughput, low-latency connection by using Wi-Fi Aware.

📄 Connecting devices for peer-to-peer Wi-Fi

Make outgoing and accept incoming secure connections with paired devices.

`com.apple.developer.wifi-aware`

The entitlement the system requires for an app to use the Wi-Fi Aware framework.

`WiFiAwareServices`

Dictionaries of Wi-Fi Aware services that the app can publish or subscribe to.