

[WatchKit](#) / Interacting with Bluetooth peripherals during background app refresh

## Sample Code

# Interacting with Bluetooth peripherals during background app refresh

Keep your complications up-to-date by reading values from a Bluetooth peripheral while your app is running in the background.

[Download](#)

iOS 15.0+ | iPadOS 15.0+ | watchOS 8.0+ | Xcode 14.0+



## Overview

### Note

This sample code project is associated with WWDC 2022 session [10135: Get timely alerts from Bluetooth devices in watchOS](#), and with WWDC 2021 session [10005: Connect Bluetooth Devices to Apple Watch](#).

## Configure the sample code project

This sample runs only on physical devices.

This project contains two targets: BARBluetooth for iOS and BARBluetooth WatchKit App for watchOS.

1. Go to your [Apple Developer account page](#) and create unique App IDs for BARBluetooth, BARBluetooth WatchKit App, and BARBluetooth WatchKit Extension. The IDs in the sample are already in use. You need your own unique identifiers to provision and run the app.
2. Select the BARBluetooth target, click the Signing & Capabilities tab and change the bundle identifier to the ID you created in the previous step.

3. Select the appropriate team from the Team pop-up menu to let Xcode automatically manage your provisioning profile.
4. Repeat steps 2 and 3 for the WatchKit App and the WatchKit Extension targets. The bundle IDs need to be <Your iOS app bundle ID>.watchkitapp and <Your iOS app bundle ID>.watchkitapp.watchkitextension, respectively.
5. For the WatchKit App target, select the Info tab, and change the value of the WKCompanionAppBundleIdentifier key to <Your iOS app bundle ID>.
6. Select the Info.plist file of the WatchKit Extension target, choose NSExtension > NSExtensionAttributes > WKAppBundleIdentifier key, and change the value of the key to <Your iOS app bundle ID>.watchkitapp.
7. Build and run the app on both Apple Watch and iPhone.
8. On the phone, allow Bluetooth access by choosing Settings > Privacy > Bluetooth > BARBluetooth, and run the app.
9. On the watch, add the BARBluetooth widget to the active watch face.
0. On the watch, select the Bluetooth peripheral (the phone) to connect to it.
  1. After connecting, tap "Send temperature alert" on the phone to demonstrate timely alerts.
  2. On the watch, to demonstrate the alert when the peripheral moves out of range and comes back in functionality, turn on the "Scan to alert" switch. Restart scanning and move the phone outside of Bluetooth range, then bring it back in range.

The watch app reads a characteristic value from the phone and updates the complication during background runtime. In watchOS 9, the watch app displays a notification if the peripheral (in this case, the phone app) sends a special value. The watch app also demonstrates displaying a notification as the peripheral goes in and out of advertising range.

---

## See Also

### Runtime management

- ☰ Background execution  
Manage background sessions and tasks.
- ☰ Life cycles  
Receive and respond to life-cycle notifications.
- 📄 Using extended runtime sessions

Create an extended runtime session that continues running your app after the user stops interacting with it.

```
class WKExtendedRuntimeSession
```

A session that continues to run your app after the user has stopped interacting.