DeviceKit for iOS

DeviceKit provides rich, native iOS device information and event subscriptions for Unity applications. It uses native Swift APIs to expose low-level properties and device events in a simple C# API.

> ⚠️ **Platform Supports: ✅ iOS 13.0+ ✅ iPadOS 13.0+**

Device Info

Property	Description
`Description`	Full name of the device (e.g., `iPad Pro (12.9-inch) (6th generation)`).
`Identifier`	Internal model identifier (e.g., `iPhone14,2`).
`Cpu`	Human-readable CPU name (e.g., `A15 Bionic`).
`Model`	Generic model name (e.g., `iPhone`).
`LocalizedModel`	Localized string for model (e.g., `iPhone`, `iPad`).
`SystemName`	Usually `"iOS"`.
`SystemVersion`	iOS version (e.g., `17.4.1`).
`AppVersion`	`CFBundleShortVersionString` (e.g., 1.2.3)
`BuildVersion`	`CFBundleVersion` (e.g., 153.0)
`BundleIdentifier`	From the app's Info.plist. com.app.test
`AppName`	From the app's Info.plist. MyApp
`InstallDate`	First install date based on Library folder creation.
`GenerateUUID()`	Generates a random UUID string. Like: E2421E3F-6C2B-4B8A-A123-8D5C0E7F4B1A

Display

Property	Description
`ScreenRatio`	Aspect ratio (e.g., `9.0:19.5`). Returns `(0,0)` if parsing fails.
`Diagonal`	Diagonal size in inches. `6.1`
`Ppi`	Pixels per inch. `326`
`Brightness`, `SetBrightness()`	Get/set screen brightness (0–100%).
`IsZoomed`	Whether Display Zoom is enabled.
`HasRoundedDisplayCorners`	If screen corners are rounded.
`SupportsWirelessCharging`	Device supports Qi charging.

🔄 Events

`Event fired when screen brightness changes.`

```
event Action<int> DeviceKit.BrightnessChanged;
```

🔋 Battery

Property	Description
`BatteryLevel`	From `0` to `100`.
`BatteryState`	Enum: `Unknown`, `Unplugged`, `Charging`, `Full`.
`IsLowPowerModeEnabled`	iOS system power mode status.

🔄 Events

`Event fired when battery level or state changes.`

```
event Action<int, BatteryState> DeviceKit.BatteryStateChanged;
```

`Event fired when Low Power Mode toggles.`

```
event Action<bool> DeviceKit.LowPowerModeChanged;
```

🧠 Sensors

Property	Description
`ProximityState`	`true` when the proximity sensor is triggered.

🔄 Events

Event fired when proximity sensor state changes.

```
event Action<bool> DeviceKit.ProximityStateChanged;
```

📷 Cameras

Property	Description
`Cameras`	CSV of available types (e.g., `wide,ultraWide,telephoto`).
`HasCamera`, `HasWideCamera`, `HasUltraWideCamera`, etc.	Booleans.

📡 Audio

Property	Description
`IsSystemMuted`	Returns `true` if system volume is zero.

`OutputVolume`	Float between `0.0` and `1.0`.
`IsOtherAudioPlaying`	Returns `true` if music or podcast is playing.
`CurrentAudioRoute`	List of `AudioRoute` entries (e.g., `BluetoothA2DP:AirPods`).

🔄 Events

Event triggered when the current audio route changes (e.g. AirPods plugged).

```
event Action DeviceKit.AudioRouteChanged;
```

📸 Screen & Media

Property	Description
`IsScreenCaptured`	Returns `true` if screen is being recorded or mirrored.

🔄 Events

Event triggered when the screen capture state changes (e.g. user starts or stops screen recording or mirroring).

```
event Action<bool> DeviceKit.ScreenCapturedChanged;
```

Event triggered when the user takes a screenshot (via system shortcut).

```
event Action DeviceKit.ScreenshotTaken;
```

🌡️ Thermal & Disk

Property	Description
`ThermalState`	Current thermal state of the device. Enum: Unknown, `Nominal`, `Fair`, `Serious`, `Critical`.
`AvailableDiskSpace`	Available disk space (important usage) in bytes.
`AvailableDiskSpaceOpportunistic`	Available disk space (opportunistic usage) in bytes.

🔄 Events

Event fired when thermal state changes.

```
event Action<ThermalState> DeviceKit.ThermalStateChanged;
```

🧬 Biometric & Hardware

Property	Description

`HasTouchID`, `HasFaceID`, `HasBiometric`	Biometric capabilities.
`Has3DTouch`, `Has5GSupport`, `HasLidarSensor`	Device feature checks.
`HasDynamicIsland`, `HasSensorHousing`	Notch/island/sensors.
`HasUSBCConnectivity`	USB-C charging support.
`ApplePencilSupport`	CSV of supported generations.

🧭 System Features

Property	Description
`IsGuidedAccessSessionActive`	True if Guided Access is active.
`IsDarkModeEnabled`	True if system dark mode is active.
`PreferredLanguages`	Returns a list of the user's preferred languages. ["uk-UA", "en-US"]
`SystemLanguageCode`	ISO system language code (e.g., `"en"`).
`ProcessorCount`	Logical cores.
`CountryCode`	Current region/country code e.g. "UA", "US".
`IsTestFlight`	True if installed from TestFlight or Xcode.
`TimestampMs`	Unix timestamp in ms.

🎯 Badge & Notification

Property	Description
`BadgeCount`	Get the current icon badge number.
`SetBadgeCount()`	Set the application's icon badge number.

Requests permission from the user to display notification badges.

```
DeviceKit.RequestBadgePermission(granted ⇒ {
    Debug.Log("Badge allowed: " + granted);
});
```

🧪 Example Usage

```
if (DeviceKit.IsPhone)
    Debug.Log($"Device: {DeviceKit.Description}, PPI: {DeviceKit.Ppi}");

DeviceKit.BrightnessChanged += b ⇒ Debug.Log("New brightness: " + b);
```