

Release Notes for iOS 24.1.0

- iOS
 - [Minimum Requirements](#)
 - [New Features](#)
 - [Resolved Issues](#)
 - [Known Issues](#)

iOS

Minimum Requirements

- iOS 15.0 device or iPadOS 15.0 device
- tvOS devices and app extensions are no longer supported
- visionOS for Vision Pro devices is not supported

New Features

- A fix supports the instrumentation and use of the newer WKNavigationDelegate method introduced by Apple in iOS 13. The app can supply the new version of decidePolicyForNavigationAction, the old version, or can supply neither.
 - Apple Documentation: <https://developer.apple.com/documentation/webkit/wknavigationdelegate/3223382-webview>
- SDK imports have to be renamed from "WS1Intelligence" to "**WS1IntelligenceSDK**"
- **dsym upload shell script has been updated to parse the latest service account json file. Please re-create and re-download the service account json file to support the dsym upload/parsing scripts. Instructions to generate the service account json is here - [link](#)**
- WS1IntelligenceSDK enable API takes in a WS1 config object outside of the AppID. This config object should be used to enable DEX and inject all entitlements that an app supports as shown below -

```
let config = WS1Config.default()
config.enableMachExceptionHandler = true
// enable DEX for DEX capabilities
config.enableDEX = true
// Inject all entitlements that App supports that SDK requires.
config.entitlements = [.wifi_info()]

WS1Intelligence.enable(withAppID: "App ID", config: config)
```

- A list of all entitlements / permissions that **WS1IntelligenceSDK** requires is listed below. If the entitlements / permissions are not given / available then those attributes are not sent to the Intel backend.

Permissions / Entitlements Required

Attribute	Platform	Entitlements	Permissions
bluetooth_connected	MacOS	Bluetooth	
ssid	iOS	wifi_info	Location
bssid	iOS	wifi_info	Location
wifi_signal_strength	iOS	wifi_info	Location

- IntelSDK is also compatible with M1/M2 machines.
- Introduced more modern external distribution options:
 - XCFramework
 - Swift Package Manager
- Apps integrating the SDK should now also set an instance of type **WS1UEMDelegate**. (**WS1UEMDelegate must be set before enabling IntelSDK**). This is to publish the following UEM specific attributes **serialNumber**, **deviceUDID**, **username** to the Intel backend. Integration code is shown below -

WS1UEMDataDelegate

```
@objc public protocol WS1UEMDataDelegate: AnyObject {
    var serialNumber: String? { get }
    var deviceUDID: String? { get }
    var username: String? { get }
}
```

Sample Implementation

```
#import "WS1UEMAttributeKeys.h"
#import "WS1Intelligence.h"

@interface AppDelegate () <WS1UEMDataDelegate>
@end

@implementation AppDelegate

- (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:(NSDictionary *)
launchOptions {
    WS1Config *config = [WS1Config defaultConfig];
    [WS1Intelligence setUEMProviderDelegate:self];
    [WS1Intelligence enableWithConfig:config];
}

- (NSString *) deviceUDID {
    return [self getAppConfig:[WS1UEMAttributeKeys intelSDKDeviceUDID]];
}

- (NSString *) serialNumber {
    return [self getAppConfig:[WS1UEMAttributeKeys intelSDKSerialNumber]];
}

- (NSString *) username {
    return [self getAppConfig:[WS1UEMAttributeKeys intelSDKSerialNumber]];
}

- (NSString *) getAppConfig:(NSString*) key {
    NSDictionary<NSString*, id> *dictionary = [[NSUserDefaults standardUserDefaults] objectForKey:
[WS1UEMAttributeKeys managedAppConfigKey]];
    return [dictionary valueForKey:key];
}

@end
```

Resolved Issues

- Fixed a bug that could crash the app if the user reset the device's clock to an earlier time than the last use of the app.
- Fixed a bug that caused a crash if a URL specified by the app was nil.
- improved the way the "App Start" userflow is automatically generated. This should give more reliable times in the reports.
- Resolved an issue when instrumenting WKWebView if loaded from storyboard.

Known Issues

- This version currently does not support Apple Privacy [manifest](#) policy