Integration Preparation Guide

Workspace ONE for iOS and iPadOS

Applications for iOS and iPadOS can be integrated with the Omnissa Workspace ONE® platform, by using its mobile software development kit.

The tasks detailed below should be done first, to prepare for integration of your application with the Workspace ONE platform. After completing these tasks, you will be ready to begin the integration work.

This document is part of the Workspace ONE Integration Guide set.

Table of Contents

Welcome

Welcome to the Workspace ONE Integration Preparation Guide for iOS and iPadOS.

Objectives

The objectives of this guide are for you to

- verify your app works.
- verify your developer device is viable.
- use the Workspace ONE Unified Endpoint Manager (UEM) console.
- verify the round trip from integrated developer environment (IDE) to end user device.
- verify you will be able to install the app from your IDE later, when integration work is in progress.

The integration preparation guide doesn't include any work at the code level. That will start in the base integration guide, which is next.

Overview

This is an overview of the steps in the guide.

1. Build your app without Workspace ONE integration.
2. Install the app to a developer device from the IDE.
3. Uninstall the app from the developer device.
4. Generate an iOS Package Archive (IPA) file for the app.
5. Upload the IPA to the UEM as a new internal app.
6. Install Hub on the developer device.
7. Enrol Hub with the UEM console.
8. From Hub, install your app.
9. From the IDE, make a visible change to the app and install that as an upgrade.

Enrolling Hub with the UEM console establishes trust between the device and the UEM. Installing your app from Hub will enable it to be trusted by the UEM subsequently.

Installation Order Note

If you follow the integration guide set, you will install your application on a developer device as follows.

- The first installation will be of a non-integrated version of the application via Workspace ONE.

- Subsequent installations will be of integrated versions of the application from your IDE. This type of installation is referred to as a *side load*.

The side-load installations will be upgrades. The application won't ever be uninstalled after the first installation via Workspace ONE.

It actually isn't necessary to install an application via Workspace ONE if it isn't integrated. It might therefore seem natural to delay installation via Workspace ONE until some integration work has been done. This guide doesn't follow that order though.

The rationale is that installation via Workspace ONE involves activities with which you may be unfamiliar, such as generating an IPA file, and use of the Workspace ONE management console. It's better to do those activities with the application in a known working state.

First Steps

Begin by checking the Prerequisite Conditions.

Prerequisite Conditions

Before you begin, you will need the following.

- Access to a Workspace ONE management console.

  You will need access to a Workspace ONE management console to work on application integration. The management console is sometimes referred to as the UEM, an abbreviation for Unified Endpoint Manager.

  You will need to know the following:

  - Server address.

  - Administrative login credentials.

  You will need the following privileges:

  - Upload a mobile application installer, which will be an iOS Package Archive (IPA) file.

  - Either create an organisation group for an end user, or get the name of an existing group.

  - Either create a new end user with a suitable profile for development purposes, or get the name of an existing suitable user.

  - Either create enrolment credentials for an end user, or get existing credentials.

  Best practice is to have a separate console, or organisation group, for software development.

  Workspace ONE supports installation of custom apps on managed and unmanaged devices. If you have a choice, it will be easier for development if the use of unmanaged devices, sometimes referred to as Hub Registered mode, is allowed in the device enrollment policies.

- Apple account, ideally with a developer membership.

  In case you don't have a developer membership, it is possible to use a personal Apple account for Workspace ONE integration work. A developer account will be better. For a discussion of the differences, see the Appendix: Apple Accounts.

- Developer environment.

  The instructions in the integration guide documents assume you use the Apple Xcode integrated developer environment (IDE). Check the Compatibility table for a recommended version.

  The Xcode IDE can be obtained from Apple, for example by downloading from their developer website here.
  https://developer.apple.com/download/all/

- Developer device.

  You will need a physical iOS or iPadOS device to to work on application integration. You will install the Workspace ONE Intelligent Hub app on the device as part of preparation.

  The iOS and iPadOS simulators cannot be used because they may appear as jailbroken or otherwise compromised to the Hub app. The Hub must be used to install the application that is being integrated at least once during the integration work. After the first installation via Hub, subsequent installations can be made using Xcode in the usual way.

  The device must be set up to trust your computer. When you connect the device for the first time a prompt will be shown on the device, which you must accept to use the device for development purposes.

  The device might also have to be set up for developer use, depending on its operating system version. Instructions for setting up a developer device can be found on the Apple developer website, for example here.
  developer.apple.com/.../enabling-developer-mode-on-a-device

  The device must be compatible with the Apple account that you are using. If you are using a developer account then the device might have to be registered to the associated developer organization. Xcode can do this automatically. Check the Apple developer website for details, for example here.
  developer.apple.com/.../register-a-single-device

  Don't use a device that is already enrolled with a production Workspace ONE console. A device can sometimes be unenrolled from within the Hub application on the device, or by resetting to factory defaults.

When the prerequisite conditions are met, you can start the first Task: Select or create an application to integrate.

Compatibility

Instructions in this document have been followed with the Workspace ONE Software Development Kit (SDK) for iOS and other software, to confirm compatibility.

The following table shows the software versions used for the instructions, and the minimum supported versions if different.

Software	Instructions	Supported
Workspace ONE software development kit for iOS	25.11	
Workspace ONE management console	2506	2410,2406,2402
Workspace ONE Intelligent Hub application	25.09	
Apple iOS or iPadOS Operating System	26.1	16
Apple Xcode integrated development environment	16.4	16

The SDK supports versions of the Swift language that are supported by the above Xcode versions.

Task: Select or create an application to integrate

Integrating an application with Workspace ONE will involve changes to the application source code. You will need access to the Swift source, bundle files, project files, and to any other files required to build the application.

You can integrate Workspace ONE with a mobile application of your own for which you already have the source. If you choose to do that then you can skip this task and continue with the next Task: Generate an application installer file.

You can instead create a sample application from an Xcode project template, by following these instructions. These are provided here for convenience; for definitive information, see the Apple developer website.

1. **Open the Xcode application.**

2. **In the menu, select File, New, Project…**

    This opens a dialog on which you select a project template.

3. **Select the iOS App template, which could be the default, and click Next.**

    This screen capture shows selection in the Xcode user interface.
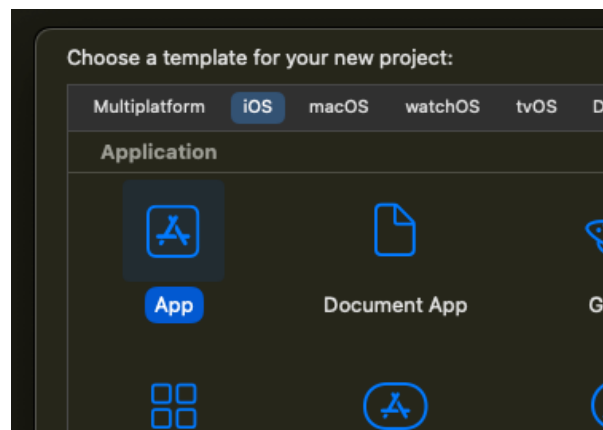
Screen Capture: Xcode select iOS App template

    When you click Next a dialog will open on which you can specify options for the new project.

4. **For the Product Name, enter `ws1app` or any name you like.**

    Tip: If you use all small letters the bundle identifier will also be all small letters and won't have to be edited later.

5. **If you have a developer account, select its organization as the Team.**

   For a discussion about developer and personal accounts see the Appendix: Apple Accounts.

   If you don't have a developer account then you can select a Personal Team instead.

6. **For the Interface, select SwiftUI or Storyboard as you prefer.**

   Both user interfaces are supported by Workspace ONE for iOS. SwiftUI is recommended.

7. **For the Language, select Swift or Objective-C as you prefer.**

   Both languages are supported by Workspace ONE for iOS. Swift is recommended, and will be the only option if you chose SwiftUI for the Interface.

8. **There is no need to Use Core Data, nor to Include Tests.**

   This screen capture shows the dialog as it might appear.
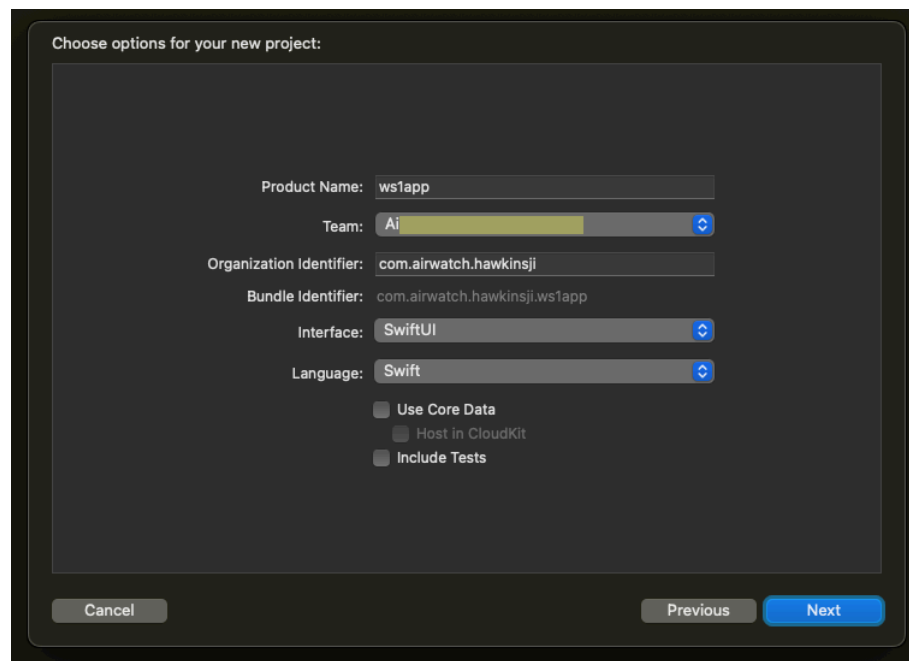
Screen Capture: Xcode new project options

   (The team name begins "Ai" but is otherwise hidden.)

9. **Check the values and click Next.**

10. **Choose a location to save your project.**

    It can be a good idea to create a new sub-directory for the project.

    When you have chosen the location Xcode will generate the project files, which might take a little while. Wait for processing to finish.

11. **Run the new app on a device once.**

    To confirm that the project, your developer environment, and your device are all set up correctly, run the new app on your developer device from Xcode.

    Your device might show an error like Untrusted Developer. The error message will have instructions for how to clear the error condition. Follow the instructions and try again to run the app.

12. **Uninstall the app.**

    Uninstall the app so that you can install it from Workspace ONE.

    This completes the creation of a sample application. You are now ready to continue with the next Task: Generate an application installer file.

Task: Generate an application installer file

Generating an application installer file is a task for application developers. The task is dependent on the Task: Select or create an application to integrate. The following instructions assume that the dependent task is complete already.

You will need to generate an application installer file for your app at least once so that your app can be trusted by the Workspace ONE UEM console and Hub app. Note that you don't need to do any Workspace ONE integration in the application at this stage.

The installer file will be in a format known as IPA, now a back-formed abbreviation for iOS Package Archive. You can generate an IPA file for your app with Xcode. Instructions are provided here for convenience; for definitive information, see the Apple developer website.

You will need an Apple account to generate an IPA file. You will also need to know what type of account you have, specifically whether you have a developer account or not.

- If you have a developer account then you should use that when you are building your app. Follow the Instructions for developer accounts to generate the IPA file.

- If you don't have a developer account, then you will have to follow the Instructions for personal accounts to generate the IPA file.
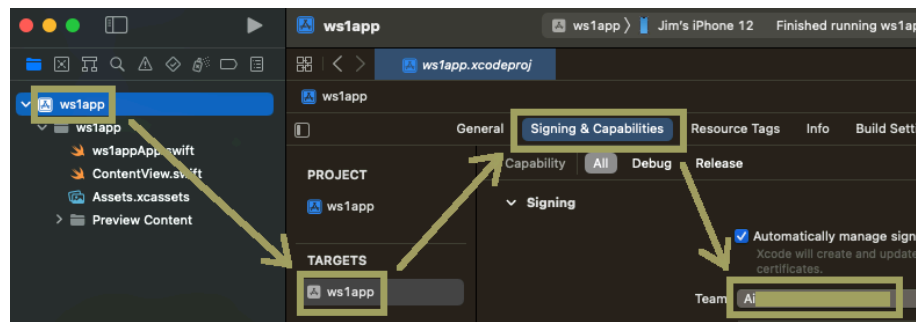
In case you are unsure what account type you have, see the Appendix: Apple Accounts.

Instructions for developer accounts

If you have an Apple developer account you can generate an application installer file by following these instructions. These are provided here for convenience; for definitive information, see the Apple developer website. If you don't have a developer account, or aren't sure what type of account you have, then review the introduction to the Task: Generate an application installer file.

1. **Open the application project in Xcode.**

2. **Ensure that your developer account and organization are the selected team for the app.**

   This screen capture shows the location of team selection in the Xcode user interface.

   Screen Capture: Xcode select team

   For more tips on accounts and organizations, see the Appendix: Apple Accounts.

3. **Ensure that your developer device is connected and is the chosen build target.**

   If necessary, set up your device for developer use, for example by following the instructions on the Apple developer website here.
   developer.apple.com/.../enabling-developer-mode-on-a-device

4. **In the Xcode menu select Product, Archive.**

   Wait for the success message, shown in this screen capture, or for the Organizer window to open.
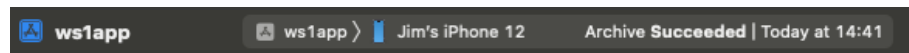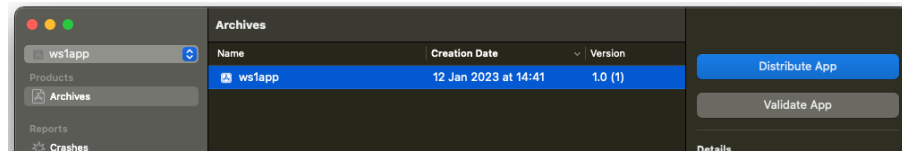

   Screen Capture: Xcode archive succeeded

5. **In the Xcode Organizer Window, select to distribute the archive you just created.**

   If the window didn't open automatically when archiving succeeded, you can open it manually from the Xcode menu by selecting Window, Organizer.

   Click the Distribute App button, shown in this screen capture.

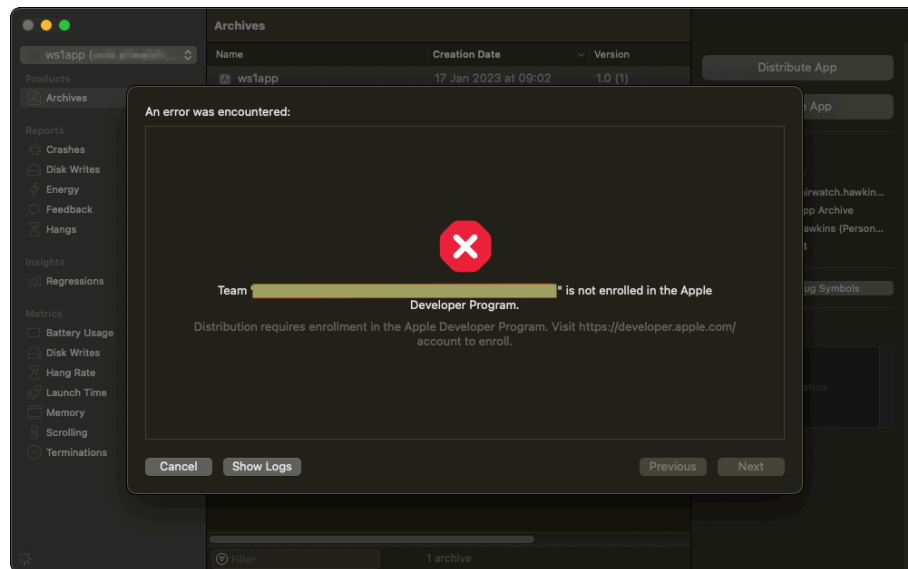   **Screen Capture:** Xcode distribute app

   This will start an interaction spread across multiple screens, beginning with selecting a method of distribution.

6. **Select Development distribution and click Next.**

   All being well, Xcode will process for a short time and then open the next screen, on which you will select options for development distribution.

   If the team selected for the app isn't associated with membership of a developer organization then processing will fail. An error will be displayed that the team isn't enrolled in the developer program.

   This screen capture shows the error message.

Screen Capture: Xcode distribution error

   If this happens then check that you have a developer account and that it is selected for the app. See also the Appendix: Apple Accounts. If you don't have a developer account then you will have to follow the Instructions for personal accounts instead.

7. **On the developer distribution options screen, leave the default selections in place and click Next.**

   The default selections are App Thinning: None and no Additional Options.

   An option like Rebuild from Bitcode might be offered. If you are following the recommended sequence and generating an installer for a non-integrated app then the Bitcode option doesn't matter. However, if the app has been integrated with the Workspace ONE SDK then any Bitcode option must be deselected. The SDK binaries don't support Bitcode.

   When you click Next, the re-sign screen opens.

8. **Select to Automatically manage signing and click Next.**

   Xcode processes for a short time and then opens the next screen, on which you review the package contents.

9. **Check the details and click Export.**

   This opens a file chooser dialog. The default location will be in a new directory at the level above the Xcode project.

10. **Make a note of the location and click Export to finish.**

   Xcode will create an iOS Package Archive (IPA) installer file in the chosen location. Open the location in the Finder and locate the .ipa file.

   This screen capture shows how it might appear.
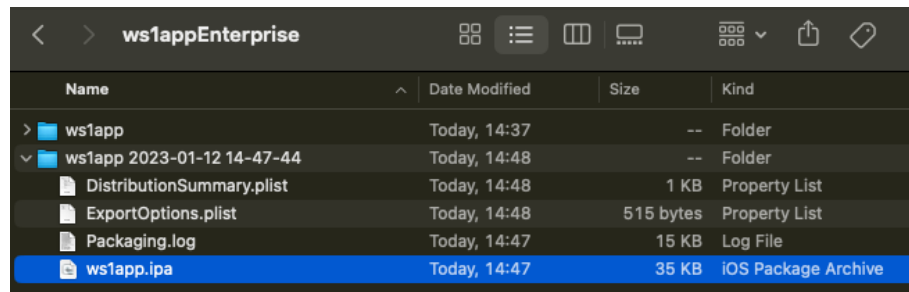
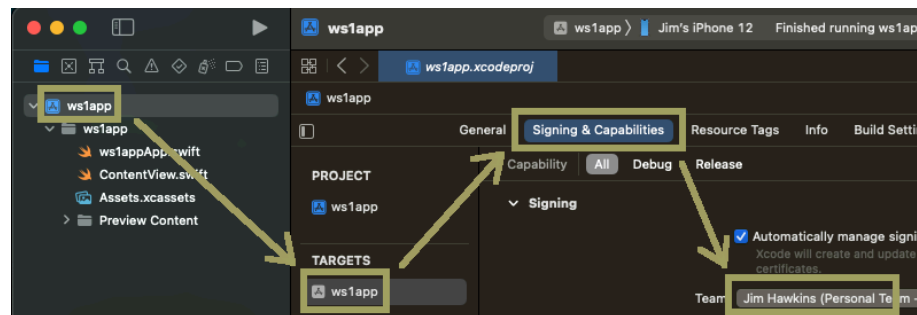Screen Capture: Distribution folder with application installer in Finder

This completes generation of an application installer file for a developer account. You are now ready to continue with the next Task: Add the application to the management console.

Instructions for personal accounts

If you don't have an Apple developer account you can generate an application installer file by following these instructions. If you do have a developer account then you should use that instead and follow the Instructions for developer accounts. If you aren't sure what type of account you have, then review the introduction to the Task: Generate an application installer file.

1. **Open the application project in Xcode.**

2. **Check the account and organization that are selected team for the app.**

   This screen capture shows the location of team selection in the Xcode user interface.

   Screen Capture: Xcode select team

   For more tips on accounts and organizations, see the Appendix: Apple Accounts.

3. **Ensure that your developer device is connected and is the chosen build target.**

   If necessary, set up your device for developer use, for example by following the instructions on the Apple developer website here. developer.apple.com/.../enabling-developer-mode-on-a-device

4. **In the Xcode menu select Product, Build.**

   Wait for the success message, shown in this screen capture.
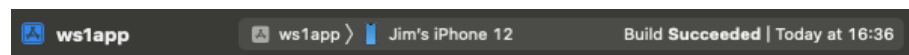

   Screen Capture: Xcode build succeeded

5. **In the Xcode menu select Product, Show Build Folder in Finder.**

6. **In the build folder, locate the .app item for your application.**

   The required item could be under the Products/ directory, in a sub-directory.

   Finder displays it as a plain file with type Application. However, the .app item is actually a type of folder structure. You can cd into it using the command line in a terminal for example.

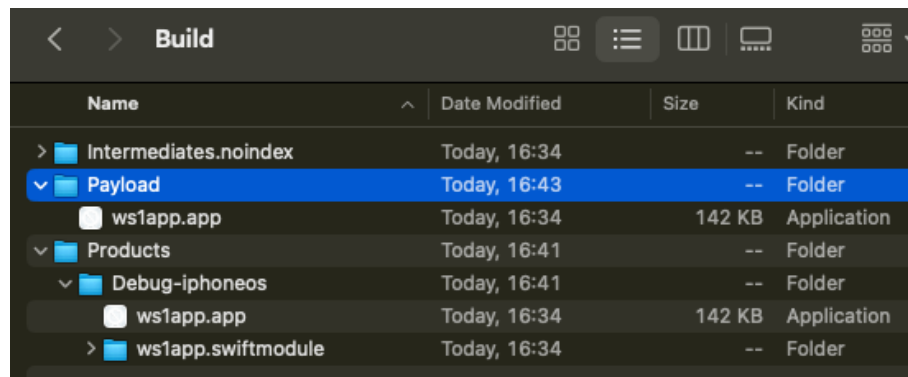   This screen capture shows how the .app item might appear in Finder.
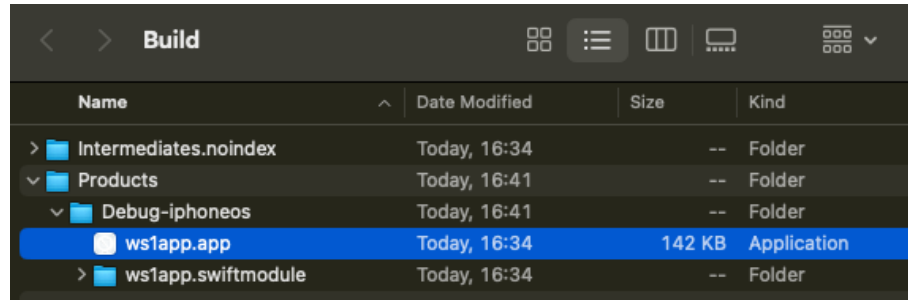
   Screen Capture: Build folder in Finder

7. **Create a new folder named `Payload`**

   The new folder can be in the build folder or elsewhere on your computer.

8. **Copy or move the .app item under the new Payload folder.**

   This screen capture shows how the Payload folder might appear.

   Screen Capture: Payload folder in Finder

   (In this case, the .app has been copied.)

9. **Compress the Payload folder, for example by highlighting it and then selecting File, Compress in the Finder menu.**

   This creates a `Payload.zip` file.

   The Payload folder is no longer required and you can delete it now, if you like.

10. Rename `Payload.zip` to `ws1app.ipa`

Finder might warn you that changing the extension may cause the file to open in a different application. If it does then confirm that you want to use the .ipa extension.

You could change the name part of the file to something different, if you chose a different name for your app for example.

This screen capture shows a possible final state of the build folder and renamed file.
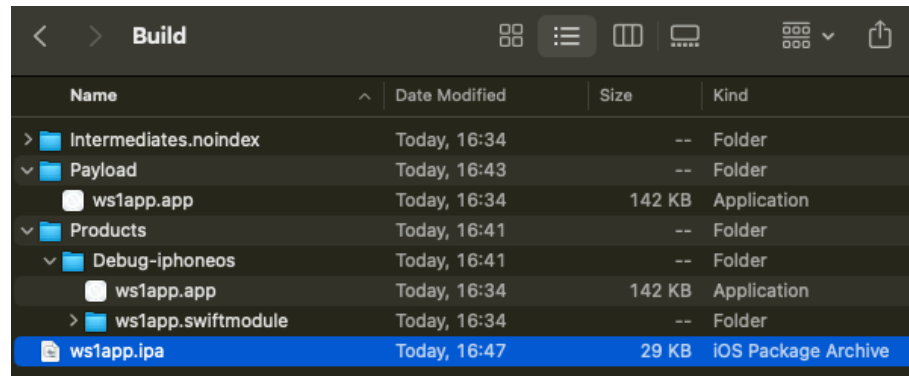
Screen Capture: Build folder with application package in Finder

You could proceed to uploading the .ipa file to the UEM now. However, it's a good idea to test the file first. You can do this by side loading it to your developer device using Xcode as described here.

11. **In the Xcode menu select Window, Devices and Simulators.**
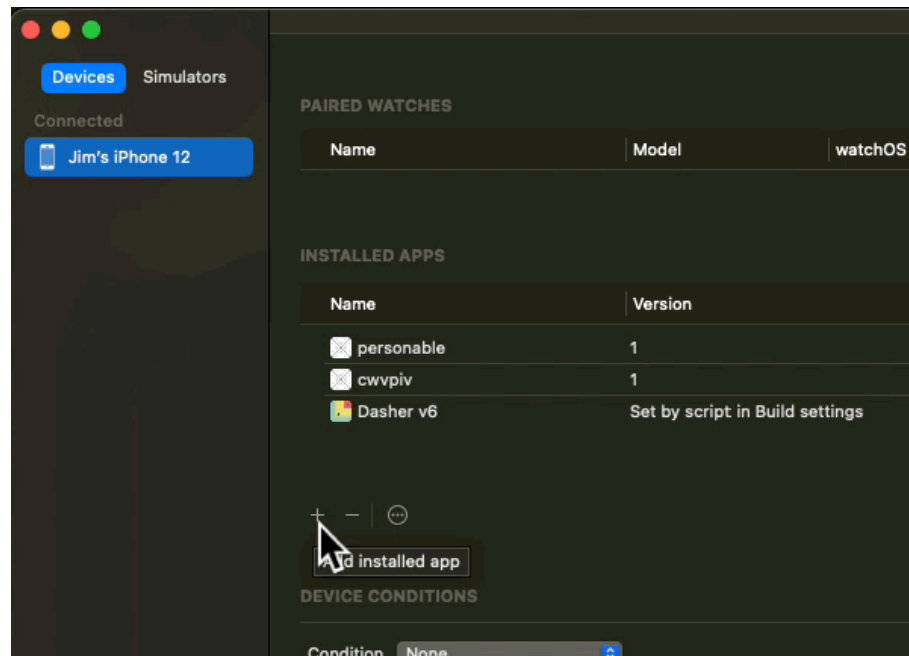
This opens the Xcode device management window.

12. **In the left pane, select Devices and your developer device, if it isn't selected by default.**

13. **In the right pane, scroll down to the list of installed apps.**

14. **Click the plus button under the list of installed apps.**

    This screen capture shows the location of the button.

    Screen Capture: Xcode add app to device

    This opens a file chooser.

15. **In the file chooser, select the .ipa file.**

    Tip: If you have the .ipa file highlighted in Finder, you can copy its path by pressing Cmd+Alt+C. If you then type an oblique, **/**, in the Xcode file chooser, it will open a text box in which you can paste the copied path. Delete the oblique before pasting.

    Xcode will install the app in the .ipa file to your developer device.

16. **Check the app opens and the user interface appears as expected.**

17. **Remove the app from the device.**

    You can do this on the device, or in the Xcode device manager window.

    This completes generation of an application installer file for a personal account. You are now ready to continue with the next Task: Add the application to the management console.

Task: Add the application to the management console

Adding an application to the management console is a task for application developers. You will need the iOS Package Archive (IPA) file from the Task: Generate an application installer file. The following instructions assume that you have the IPA file already.

You can add an application to the Workspace ONE Unified Endpoint Manager (UEM) console by following these instructions. These are provided here for application developer convenience and aren't intended to replace the system administrator user guides for the Workspace ONE product.

Upload the application installer

Uploading the application installer is part of the Task: Add the application to the management console. Proceed as follows.

1. **Log in to the UEM and select the Organization Group (OG) of the end user that you are using for development.**

2. **Navigate to: Resources, Apps, Native.**

   Depending on your console version and customization the navigation could be different.

   - Resources might be labelled Apps & Books instead.

   - Apps might be labelled Applications instead.

   However you navigate, a list of applications will open.

3. **Select the Internal tab if it isn't selected by default.**

4. **Expand the Add control and select Application File.**

   The following screen capture shows the location in the user interface.
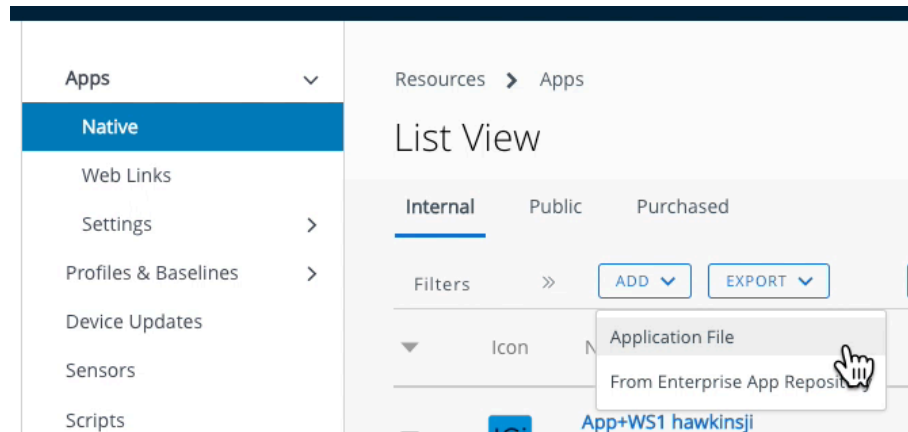
   Screen capture: Add Application File in console

   This opens the Add Application dialog.

5. **Select the group of the end user that you are using for development if it isn't selected by default.**

6. **Click Upload, which opens the Add dialog.**

7. **On the Add dialog: select Local File, then click Choose file.**

   This opens a file chooser dialog.

8. **Locate and select your IPA file, then click Save.**

   Tip: If you have the .ipa file highlighted in Finder, you can copy its path by pressing Cmd+Alt+C. If you then type an oblique, **/**, in the browser file chooser, it will open a text box in which you can paste the copied path. Delete the oblique first.

   The file will be uploaded and progress will be indicated on the screen.

   When the upload finishes, the Add dialog closes and you return to the Add Application dialog. The file name will have been filled in.

9. **On the Add Application dialog, click Continue.**

   This opens the next step, which is a dialog in which the details can be edited.

   Continue with the instructions Configure and publish the application.

Configure and publish the application

Configuring and publishing the application is part of the Task: Add the application to the management console. These instructions come after the Upload the application installer instructions.

Proceed as follows.

1. **In the application details dialog, append your user name to the application name, if you like.**

    Appending your user name will make clear, to anybody with access to the console, that you are responsible for this application.

2. **In the same dialog, ensure that an SDK profile is selected.**

   If you are following the recommended installation order and have uploaded an app without Workspace ONE integration, then this step is in theory unnecessary. In practice however, this is a convenient point at which to make the selection and it might be forgotten later when integration work is in progress and it's essential.

   Select the SDK tab in the dialog, then select an SDK profile if one isn't selected already. The SDK tab might be under a More drop-down.
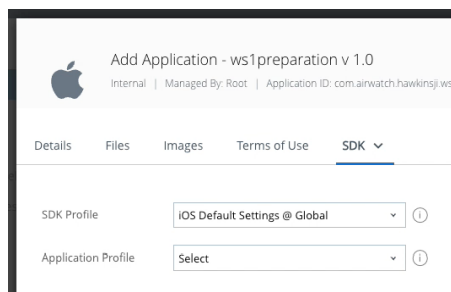
   The following screen captures show the interaction and a selected profile.
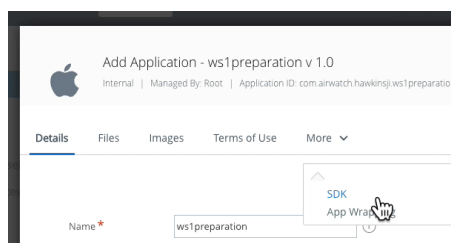
   Screen capture: Select SDK tab
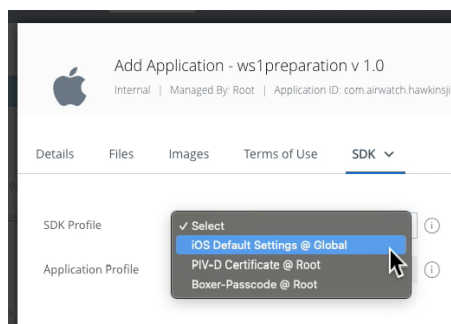
   Screen capture: Selecting an SDK Profile

   Screen capture: SDK Profile selected

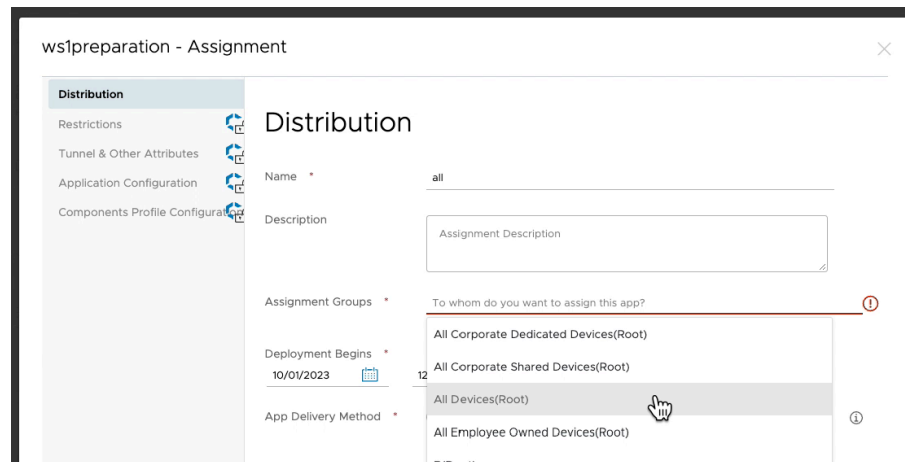   (In the above screen capture, the selected profile is "iOS Default Settings @ Global".)

   Ignore the Application Profile selection.

Click Save & Assign. The Assignment Distribution dialog will open.

3. **Set the Name and Assignment Groups as follows.**

   - Set Name to `all` or any other name you like.

   - Select the assignment group for all devices in the Organization Group (OG).

   The following screen capture shows selection of the group.

Screen capture: Selecting assignment to all devices

(In the above screen capture the OG name is Root.)

There is no need to add or enable any other items, like restrictions, at this time.

Click Create and a dialog will open in which you can review the new assignment.

4. **On the assignment review dialog, click Save.**

   This opens the Preview Assigned Devices dialog.

5. **Check that your developer device appears in the preview of assigned devices.**

   If it doesn't appear, retrace your steps and check you followed all the instructions. Also check which UEM Organization Group (OG) the end user is enrolled into. That OG must be the same as the one the app was added to, or be a child OG of the one the app was added to.

6. **Click Publish.**

   This finalizes the addition and returns you to the Applications part of the console. The detail view of the app you just added will be open.

This completes adding the application to the management console. You are now ready to continue with the next Task: Install the application via Workspace ONE.
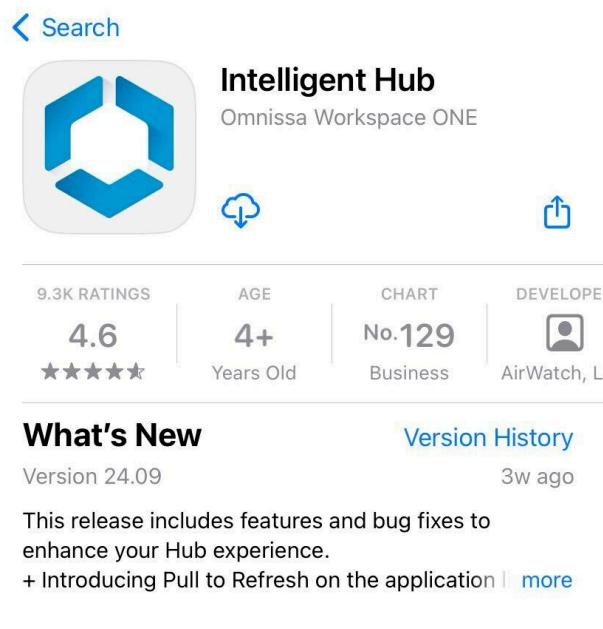
Task: Install the application via Workspace ONE

Installing the application via Workspace ONE is a platform integration task for iOS and iPadOS application developers. The task is dependent on the Task: Add the application to the management console. The following instructions assume that the dependent task is complete already.

Proceed as follows.

1. **Install and enrol the Workspace ONE Intelligent Hub application.**

   Workspace ONE supports installation of custom apps on managed and unmanaged devices. In either case, the Hub app must be installed and enrolled before the custom app is installed. The Hub can be installed from the Apple app store. Search for "workspace one intelligent hub", for example.

   This screen capture shows the top of the Apple app store page for the Hub app.

   Screen Capture: Apple app store page for Workspace ONE Intelligent Hub

   If you are installing Hub other than from the app store then check the Compatibility table for a recommended version of Hub.

   Follow the instructions in the Hub application to complete enrollment. You will need to know the server address and a set of end user enrollment credentials. Enrolling Hub establishes trust between the device and the UEM console.

   Tip: Set a device passcode before you begin enrollment. Typical UEM configurations will require a passcode, as a security policy. If a device passcode isn't set at the start of the enrollment interaction, you will be forced to set it as an enrollment step, which sometimes doesn't go smoothly.

   Warning: The Hub application cannot be enrolled with more than one management console at a time. If the Hub is already installed and enrolled on your developer device, then it must now be unenrolled, or removed and re-

installed. Unenrolling or removing the Hub may cause removal of any associated applications from the device.

You may want to try different enrolment options such as managed and unmanaged modes, or into a different organization group (OG) with other policy settings. In all cases, it will be necessary to unenroll the Hub, or wipe it from the console, and uninstall Hub and all the apps before trying the next option.

2. **Install your application from Hub.**

Your application can now be installed from the Hub app on the device. Find it in the Apps tab and select to install it.

In case your application doesn't appear, try these troubleshooting steps.

- Open the Hub application Apps tab and pull down to refresh.
- Open the Hub application and select Support, then your developer device which will be labelled as (Current), then Sync Device.
- Terminate the Hub using the device task manager, then open Hub again.

If you are using an Apple developer account, then the installed app can now be run. If you are using only a personal account then the installed app might actually be a placeholder that cannot be run. The next instruction fixes this.

3. **Side load your application from Xcode.**

To confirm that your setup is ready to begin Workspace ONE integration work, connect your developer device to your computer and install and run your app from Xcode in the usual way. Don't uninstall the app, so the Xcode side load will be installed as an upgrade.

Make a superficial but obvious change to your app first, so you can check that the side load worked. For example, if your app has a hello-world type of message in its user interface, change it to "Hello World 2".

If you earlier used a personal account to generate the IPA file, the side load will now overwrite the placeholder with the full app that can be run on the device.

This completes the task.

Warning: Don't uninstall Hub nor your app from this point until you finish the SDK integration work.

You can now proceed to the Next Steps.

Next Steps

This completes the preparation for integrating your mobile app with the Workspace ONE platform. You are now ready to start code-level integration in your application.

See the Base Integration guide for instructions. The Base Integration Guide is available

- in Markdown format, in the repository that also holds the sample code: https://github.com/euc-releases/…BaseIntegration.md TBD

- in Portable Document Format (PDF), on the website: https://developer.omnissa.com/…BaseIntegration.pdf TBD

Appendix: Apple Accounts

You will need an Apple account to complete the work of integrating an iOS or iPadOS application with Workspace ONE. For the purposes of that work there are two types of Apple account, developer and personal. Some instructions for checking your account type are given below. These are provided here for convenience; for a full discussion of account types and memberships, see the Apple developer website, for example here.
https://developer.apple.com/support/compare-memberships/

(Sensitive information in the screen captures in this section has been hidden.)

How to check account type

A developer account is one that is a member of a developer organization. Follow these instructions to check the organizations of which your account is a member.

1. **Log in to your account on the developer.apple.com website.**

2. **If you are a member of multiple organizations, select one of them.**

   This screen capture shows selection in the Apple developer website.
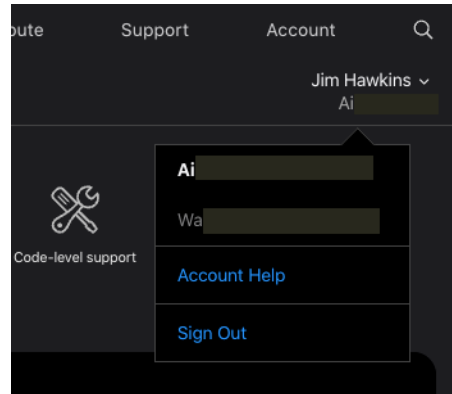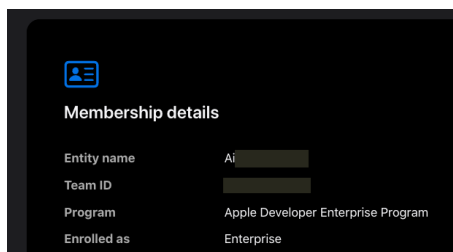

Screen Capture: Apple organization selection

   In this screen capture the user is a member of two organizations. The name of one organization starts with "Ai" the name of the other starts with "Wa". The organization starting "Ai" is selected.
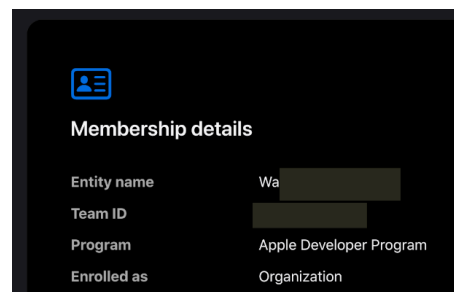
   If you aren't a member of any organizations, then you have a *personal account* and can skip the remaining steps.

3. **Scroll down to the Membership details.**

   The top of the section might look like one of these screen captures.

Screen Capture: Apple membership details for developer enterprise organization

Screen Capture: Apple membership details for developer organization

4. **Check if the word Developer appears in the Program name for the organization.**

    If it does then you have a *developer account*. Note that this applies to both the developer program and the developer enterprise program.

    If it doesn't then you have a *personal account*, unless you're a member of multiple organizations, see above, and one of the others has Developer in the Program name.

The account type has now been determined.

Association of Apple accounts with application projects

An Apple organization of some kind is associated with every iOS or iPadOS app, at build time. The associated organization may be referred to as having *signed* the app. You can select which organization to associate with an app in Xcode.

1. **Open the app project in Xcode.**
2. **In the navigator select the project itself.**
   By default, the project itself will be the first item in the navigator.
3. **In the project editor, select the app target that you're using for integration.**
   By default, each app project has only a single target.
4. **In the target editor, select the Signing & Capabilities tab.**
5. **Select the required account and organization with the Team drop-down menu.**

These screen captures show the location in the Xcode user interface and the expanded Team drop-down menu.
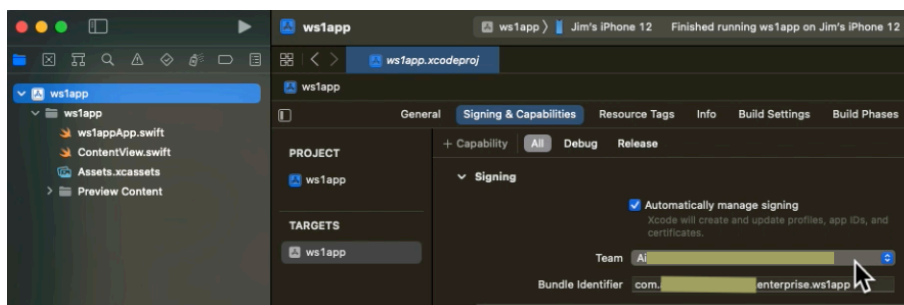
Screen Capture: Xcode target editor Signing and Capabilities tab

Screen Capture: Xcode target editor Team drop-down

In this screen capture there are two Personal Team organizations and three other organizations. The name of the selected organization begins Ai.

Account availability in the integrated developer environment

Your Apple account and organization membership must appear in Xcode in order to complete Workspace ONE integration work.

You can check this by opening Xcode and selecting Xcode, Settings, and then the Accounts tab.

This screen capture shows how the tab might appear.
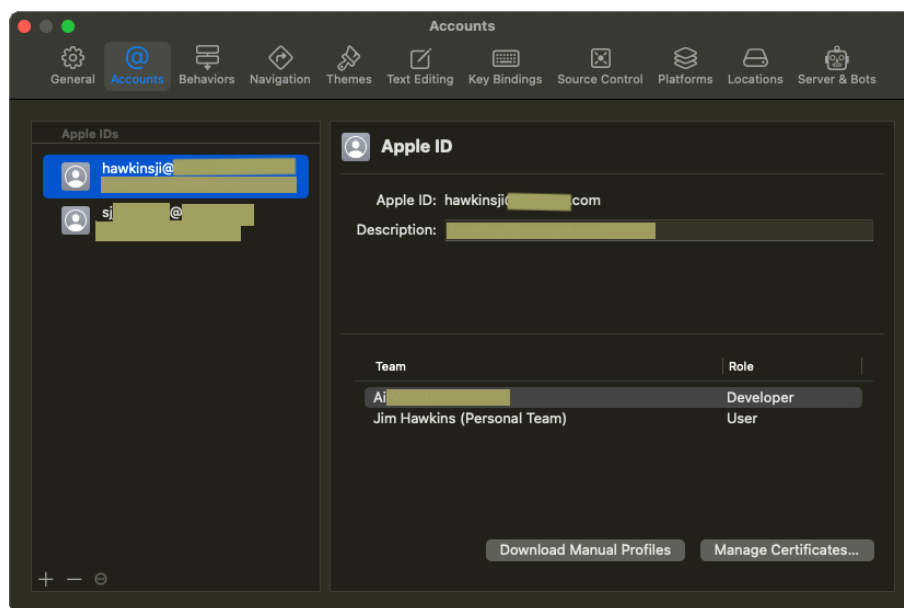
Screen Capture: Xcode Settings for Accounts

In this screen capture there are two Apple accounts, of which the first is selected. The Apple ID of the selected account is a omnissa.com email address. The Apple ID of the second begins sj and is an email address. The selected Apple ID is a member of an organization whose name begins Ai.

If the Apple ID you want to use doesn't appear, you can add it by clicking the plus button at the bottom left of the window. Also ensure that the organization membership you want to use appears on this screen. Membership issues can sometimes be resolved by removing and re-adding the account, or by closing and re-opening Xcode, or by power cycling your developer computer, or by liaising with the account administrator.

Benefits of a developer account during integration work

There are some benefits to using a developer account instead of a personal account during integration work. The benefits relate to the generation of iOS Package Archive (IPA) installer files for your application.

- You can generate IPA files automatically using just Xcode if you are using a developer account. If you are using a personal account you must take additional manual steps every time you generate an IPA file.

- The IPA file generated by a developer account can be installed by Workspace ONE on a device and run. The IPA file generated by a personal account can be installed by Workspace ONE on a device but will only be a placeholder. You will have to side load over the app from Xcode in order to run it. This limitation isn't important during integration work because you will anyway be side loading quite often during the process.

- The IPA file generated by a developer account can be installed and run on, at least, any device in the same organization. The IPA file generated by a personal account can be installed and run only on devices registered to the same Apple ID.

For instructions and background, see the Task: Generate an application installer file.

Document Information

Published Locations

This document is available

- in Markdown format, in the repository that also holds the sample code:
  github.com/euc-releases/.../IntegrationGuideForiOS/...Preparation/

- in Portable Document Format (PDF), on the website:
  developer.omnisaa.com/.../WorkspaceONE_iOS_IntegrationPreparation.pdf

Revision History

19may2023	First correct publication, for 23.04 SDK for iOS.
08jun2023	Publication for 23.06 SDK for iOS.
19jul2023	Publication for 23.07 SDK for iOS.
11sep2023	Publication for 23.09 SDK for iOS.
31oct2023	Publication for 23.10 SDK for iOS.
14Dec2023	Publication for 23.12 SDK for iOS.
29Jan2024	Publication for 24.01 SDK for iOS.
15May2024	Publication for 24.04 SDK for iOS.
01Jul2024	Publication for 24.06 SDK for iOS.
06Aug2024	Publication for 24.07 SDK for iOS.
30Sep2024	Publication for 24.09 SDK for iOS.
28Nov2024	Publication for 24.11 SDK for iOS.
11Feb2025	Publication for 25.02 SDK for iOS.
10Apr2025	Publication for 25.04 SDK for iOS.
30May2025	Publication for 25.04.1 SDK for iOS.
18Jun2025	Publication for 25.06 SDK for iOS.
29Aug2025	Publication for 25.08 SDK for iOS.
10Nov2025	Publication for 25.11 SDK for iOS.

License

This software is licensed under the Omnissa Software Development Kit (SDK) License Agreement; you may not use this software except in compliance with the License.

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language