

# Installing and Building Apps on a Mac



# Reading App Builder: Installing and Building Apps on a Mac

© 2021, SIL International

Last updated: 25 March 2025

You are free to print this manual for personal use and for training workshops.

The latest version is available at <a href="http://software.sil.org/readingappbuilder/resources/">http://software.sil.org/readingappbuilder/resources/</a>

and on the Help menu of Reading App Builder.

# Contents

1.	Int	roduct	ion	5
2.	Ins	talling	Reading App Builder	5
3.	Ins	talling	Prerequisites for Android	5
	3.1.	Java	Development Kit (JDK)	6
	3.2.	Insta	lling Android Software Development Kit (SDK)	7
	3.2	.1.	Downloading the Android SDK packages from the internet	7
	3.2	.2.	Copying the Android SDK files from someone else	9
4.	Ins	talling	Prerequisites for iOS	10
4	4.1.	Insta	ll Xcode	10
4	4.2.	Verif	y Xcode Installation	11
4	4.3.	Insta	ll Transporter	11
5.	Ins	talling	Aeneas	11
6.	Tes	sting A	pp in iOS Simulator	14
(	6.1.	Run t	he iOS Simulator	14
(	6.2.	Insta	lling Additional Simulators	15
(	6.3.	Manu	ually Installing Apps into the Simulator	15
7.	Cre	ating i	OS Certificates and Provisioning Profiles	15
-	7.1.	Enrol	l in the Apple Developer Program	15
-	7.2.	Creat	e Signing Certificate	16
-	7.3.	Creat	e Provisioning Profile	17
8.	Bui	ilding a	an iOS App	
8	8.1.	Appli	cation builds available for iOS	18
	8.1	.1.	Dedicated App	19
	8.1	.2.	Container App	19
	8.1	.3.	Asset Package	20
	8.1	.4.	Container App Website	20

8.2	2.	App Type (iOS) Tab2	2
8.3	3.	IPA Tab2	3
8.4	1.	Signing (iOS) Tab2	4
9.	Test	ting an iOS App2	7
10.	Usir	ng Xcode to Test an iOS App2	7
11.	Usir	ng DeployGate to Test an iOS App2	8
11	.1.	Creating a DeployGate Account2	8
11	.2.	Uploading your first app2	8
11	.3.	Registering a Device	0
12.	Uplo	oading an iOS App to Apple App Store3	6
13.	Usir	ng Test Flight to Test an iOS App3	7
13. ( 14. /	Usir App	ng Test Flight to Test an iOS App3 Ne Privacy Policy	7 8
<ul> <li>13.</li> <li>14.</li> <li>14</li> </ul>	Usir App .1.	ng Test Flight to Test an iOS App3 Ne Privacy Policy	<b>7</b> 8
<ul> <li>13.</li> <li>14.</li> <li>14</li> <li>14</li> </ul>	Usir App .1. .2.	ng Test Flight to Test an iOS App	<b>7</b> <b>8</b> 8
<ul> <li>13. 1</li> <li>14. 1</li> <li>14</li> <li>14</li> <li>15. 1</li> </ul>	Usir App .1. .2. Buil	ng Test Flight to Test an iOS App	<b>7</b> <b>8</b> 8 8 8 <b>8</b>
<ul> <li>13. 1</li> <li>14. 1</li> <li>14</li> <li>14</li> <li>15. 1</li> <li>16. 1</li> </ul>	Usir App .1. .2. Buil Usir	ng Test Flight to Test an iOS App	7 8 8 8 8 8
13. 1 14. 1 14 14 15. 1 16. 1 16	Usir App .1. .2. Buil Usir .1.	ng Test Flight to Test an iOS App	7 8 8 8 8 8 1
13. 1 14. 1 14 14 15. 1 16. 1 16 16	Usir App .1. .2. Buil Usir .1. .2.	ng Test Flight to Test an iOS App	7 8 8 8 8 8 1 1 5
13. 1 14. 1 14 14 15. 1 16. 1 16 16 16	Usir App .1. .2. Buil Usir .1. .2. .3.	ng Test Flight to Test an iOS App	7 8 8 8 8 1 1 5 6
13. 1 14. 1 14 14 15. 1 16. 1 16 16 16 16	Usir App .1. .2. Buil Usir .1. .2. .3. .4.	ng Test Flight to Test an iOS App	7 8 8 8 8 8 1 5 6

# 1. Introduction

This document provides information on how to install Reading App Builder and build apps on an Apple macOS system. It is possible to build an Android app using RAB on Windows, Linux or Mac, but if you want to build an iOS app for the iPhone or iPad, you will need to build it using a Mac computer.

App Builder Platform	Build Android Apps	Build iOS Apps
Windows	$\checkmark$	
Linux	$\checkmark$	
macOS	$\checkmark$	$\checkmark$

Creating an Android app on a Mac is essentially the same process as it is for Windows or Linux. To create a corresponding iOS app, you will need to enter a few more configuration items.

The apps generated by RAB for iOS will run on iPhones and iPads with iOS 12.2 or higher.

# 2. Installing Reading App Builder

To install the Reading App Builder program files:

1. Download the current Mac installer file (dmg) from the RAB website:

http://software.sil.org/readingappbuilder/download/

- 2. Double click on the file within **Finder** to open the disk image that contains the Reading App Builder application.
- 3. Copy the **Reading App Builder** application to your **Application** folder. This can be done by dragging the Reading App Builder icon from the disk image window to the shortcut of the Applications folder in the same window.

# 3. Installing Prerequisites for Android

If you want to build Android apps, you need to install the following components on your computer:

- 1. Java Development Kit (JDK)
- 2. Android Software Development Kit (SDK)

## 3.1. Java Development Kit (JDK)

You will need version 8 of the Java Development Kit (JDK) to build apps. We recommend you use Zulu, which is a free distribution of OpenJDK from Azul.

1. Go to the Download Zulu Builds of OpenJDK website:

https://www.azul.com/downloads/?version=java-8-lts&os=macos&package=jdk-fx

There are many downloads on this page, but the above link will filter the ones you see (Java Version: Java 8 LTS; Operating System: MacOS; Java Package: Java FX).

2. Scroll down the page until you see the downloads under the heading **Download Azul Zulu Builds of OpenJDK**:

Download	Azul Zulu I Release Updates	)penJDK	Azul Signing Keys Release Notes For PPC32-HF, PPC32-SPE and MIPS32 builds Contact Azul Sales	
Java Version:	Operating System:	Architecture:	Java Package:	
Java 8 (LTS)	macOS	× –Any–	JDK FX	X RESET FILTERS Older Zul
Java 8 (LTS) —				
				Checksum (SHA256) (D JSE 8 Certificate How to install?
<b>8u292b10</b> Zulu: 8.54.0.21 Latest	macOS 10.13 or later	x86 64-bit	JDK FX	Checksum (SHA256) JSE 8 Certificate How to install?

3. You have a choice between two different architectures: **x86 64-bit** and **ARM 64-bit**. You can find the processor type of your machine by clicking on the Apple symbol at the top left of your screen and selecting **About This Mac**. If you have one of the newer Macs with an M1 chip, choose ARM, otherwise you will need x86 (Intel).

Once you have identified your device architecture, you have a choice between a **dmg** file, a **tar.gz** file and a **zip** file. **Download the .dmg file** since it comes with its own installer program.

The file you download will have a filename something like this:

zulu8.54.0.21-ca-fx-jdk8.0.292-macosx\_x64.msi

4. **Double-click the file in Finder** and follow the instructions in the installation wizard to install it. By default, the installer will install the JDK to the following folder:

#### /Library/Java/JavaVirtualMachines/zulu-8.jdk/Contents/Home

**Important**: If you change the JDK install folder to something other than the default folder, you will need to remember the location of the folder so you can tell Reading App Builder where to find the JDK.

## 3.2. Installing Android Software Development Kit (SDK)

The Android Software Development Kit (SDK) is needed for building Android apps. There are two ways of installing the Android SDK:

#### 1. <u>Online</u>: Download the Android SDK packages from the internet:

Use the Android SDK Installation wizard to download and install the command line tools and three additional packages. This method will require an internet connection.

See 3.2.1 for more details.

#### 2. <u>Offline</u>: Copy the Android SDK files from someone else:

If you know someone who has already downloaded and installed the Android SDK, you can copy all the files from them.

This method is especially useful in a training workshop where several people need to install the SDK but have limited internet bandwidth.

See 3.2.2 for more details.

#### 3.2.1. Downloading the Android SDK packages from the internet

To install the Android SDK from the internet:

- 1. Launch Reading App Builder.
- 2. Select **Reading App Builder** > **Preferences** from the main menu.
- 3. Go to the **Android SDK** tab, which is the second tab.

4. Click the Install Android SDK button.

			0			
Export Folders	Build Settings	Build Folders	After Build	Interface	Files	Slideshows
Code Libraries		Android SDK	Defaults	.	Default	Folders
Android SDK Locatio	n					
The folder where the A	Android Software	Development Kit (Sl	DK) is located. Click	'Install Andro	oid SDK' to	download
and/or install the Andr	old command line	e tools and other pa	ckages.		Install And	
					install And	
Android SDK Dookog						
Android SDK Packag	es					
Android SDK Packag You need four packag	es installed for th	ne Android SDK. If a	ny are listed as 'No	t Found' belov	v, click 'Ins	tall
Android SDK Packag You need four packag Packages' to downl	es es installed for th oad them from th	ne Android SDK. If a e internet.	ny are listed as 'No	t Found' belov	v, click 'Ins	tall
Android SDK Packag You need four packag Packages' to downl Tools	es es installed for th oad them from th Not Found	ne Android SDK. If a e internet.	ny are listed as 'Not Install Packages	t Found' belov	v, click 'Ins	tall
Android SDK Packag You need four packag Packages' to downl Tools Build Tools	es installed for th oad them from th Not Found Not Found	ne Android SDK. If a le internet.	ny are listed as 'No Install Packages Check Installatio	t Found' belov	v, click 'Ins	tall
Android SDK Packag You need four packag Packages' to downl Tools Build Tools Platform Tools	es installed for th oad them from th Not Found Not Found Not Found	ne Android SDK. If a le internet.	ny are listed as 'No Install Packages Check Installatio	t Found' belov	v, click 'Ins	tall
Android SDK Packag You need four packag Packages' to downl Tools Build Tools Platform Tools Platform APl	es installed for th oad them from th Not Found Not Found Not Found Not Found	ne Android SDK. If a e internet.	ny are listed as 'Not Install Packages Check Installatio	t Found' belov 	v, click 'Ins	tall
Android SDK Packag You need four packag Packages' to downl Tools Build Tools Platform Tools Platform APl	es installed for th oad them from th Not Found Not Found Not Found Not Found	ne Android SDK. If a e internet.	ny are listed as 'Not Install Packages Check Installatio	t Found' belov  on	v, click 'Ins	tall
Android SDK Packag You need four packag Packages' to downl Tools Build Tools Platform Tools Platform API	es installed for th oad them from th Not Found Not Found Not Found Not Found	ne Android SDK. If a e internet.	ny are listed as 'Noi Install Packages Check Installatio	t Found' belov  on	v, click 'Ins	tall
Android SDK Packag You need four packag Packages' to downl Tools Build Tools Platform Tools Platform API	es installed for th oad them from th Not Found Not Found Not Found Not Found	ne Android SDK. If a e internet.	ny are listed as 'No Install Packages Check Installatio	t Found' belov	v, click 'Ins	tall
Android SDK Packag You need four packag Packages' to downl Tools Build Tools Platform Tools Platform API	es installed for th oad them from th Not Found Not Found Not Found Not Found	ne Android SDK. If a e internet.	ny are listed as 'No Install Packages Check Installatic	t Found' below	v, click 'Ins OK	tall

5. Follow the instructions on each page of the **Install Android SDK** wizard to download each of the Android SDK packages and install them.

When you are asked to specify a target folder, a good place is: /Users/your-name/Android-SDK.

Four packages will be downloaded and installed:

- Command line tools,
- Build Tools,
- Platform Tools, and
- Platform API.

If the installation was successful, you will see the version numbers displayed in green.

		Settings	
Export Folders	Build Settings Build Fold	ders After Build Interfac	ce Files Slideshow
Code Libraries	Android SDK	C Defaults	Default Folders
Android SDK Locatio	on		
The folder where the A and/or install the Andr	Android Software Development roid command line tools and oth	Kit (SDK) is located. Click 'Install A her packages.	ndroid SDK' to download
/Users/richard/Andro	pid-SDK		Install Android SDK
Android SDK Packag You need four packag Packages' to downl	ges ges installed for the Android SDI load them from the internet.	K. If any are listed as 'Not Found' I	below, click 'Install
Android SDK Packag You need four packag Packages' to downl Tools	ges ges installed for the Android SDI load them from the internet. cmdline-tools 4.0	K. If any are listed as 'Not Found' I	oelow, click 'Install
Android SDK Packag You need four packag Packages' to downl Tools Build Tools	ges ges installed for the Android SDI load them from the internet. cmdline-tools 4.0 28.0.3	K. If any are listed as 'Not Found' I Install Packages Check Installation	below, click 'Install
Android SDK Packag You need four packag Packages' to downl Tools Build Tools Platform Tools	ges ges installed for the Android SD load them from the internet. cmdline-tools 4.0 28.0.3 29.0.6	K. If any are listed as 'Not Found' I Install Packages Check Installation	below, click 'Install
Android SDK Packag You need four packag Packages' to downl Tools Build Tools Platform Tools Platform API	ges jes installed for the Android SDI load them from the internet. cmdline-tools 4.0 28.0.3 29.0.6 Android 11 (API 30)	K. If any are listed as 'Not Found' I Install Packages Check Installation	below, click 'Install
Android SDK Packag You need four packag Packages' to downl Tools Build Tools Platform Tools Platform API	ges installed for the Android SDI load them from the internet. cmdline-tools 4.0 28.0.3 29.0.6 Android 11 (API 30)	K. If any are listed as 'Not Found' I Install Packages Check Installation	below, click 'Install
Android SDK Packag You need four packag Packages' to downl Tools Build Tools Platform Tools Platform API	ges installed for the Android SDI load them from the internet. cmdline-tools 4.0 28.0.3 29.0.6 Android 11 (API 30)	K. If any are listed as 'Not Found' I Install Packages Check Installation	below, click 'Install OK Cancel

If any of the Build Tools, Platform Tools or Platform API is listed as "Not Found" (displayed in red), click the **Install Packages** button to install them.

Click the **Check Installation** button to confirm that all the packages have been installed correctly.

You can skip section 3.2.2 and go straight to section 4.

#### **3.2.2.** Copying the Android SDK files from someone else

If you know someone who has already downloaded and installed the Android SDK and is successfully building apps with it, you can copy all of their Android SDK files to a folder on your computer.

You need to look for the top-level Android SDK folder, such as **/Users/user-name/Android-SDK**, and copy the whole folder and its contents to your computer. A location such as **/Users/your-name/Android-SDK** is good. If it makes it easier, you can zip the folders and then unzip them onto your computer.

Note that there is <u>no setup program to run</u>. Copying the files from one computer to another is sufficient.

**Tip**: A typical Android SDK folder can be quite large (over 1 GB, depending on which additional packages have been installed). To build an app, you do not actually need all of the Android SDK files. If you want to cut down the number of files, here is a list of the essential and optional folders:

Android SDK Folder	Required for building apps?
cmdline-tools (or tools)	Yes
build-tools	Yes (you only need the sub-folder for the latest version)
platforms	Yes (you only need android-30 for now)
platform-tools	Yes
add-ons	No
docs	No
emulator	No, unless you want to use an emulator
extras	No
licenses	Yes
sources	No
system-images	No, unless you want to use an emulator
temp	No

# 4. Installing Prerequisites for iOS

If you want to build iOS apps and upload them to the Apple App Store, you need to install the following components:

- 1. Xcode
- 2. Transporter

# 4.1. Install Xcode

**Xcode** is required to build iOS Apps. To install Xcode, simply search for Xcode in the Mac App Store and install it. Open Xcode at least once to agree to the licensing restrictions and install components.



## 4.2. Verify Xcode Installation

To verify that you have successfully installed Xcode and that it is will be correctly used by Reading App Builder, open Terminal and run the following command to print the path to the active developer directory:

xcode-select -p

```
$ xcode-select -p
/Applications/Xcode.app/Contents/Developer
```

If you have had Xcode Command-line Tools installed previously, it might still be pointing to that installation directory and Reading App Builder will not work correctly.

```
$ xcode-select -p
/Library/Developer/CommandLineTools
```

To correct this situation, run the following command to set the active developer directory.

sudo xcode-select -s /Applications/Xcode.app/Contents/Developer

## 4.3. Install Transporter

**Transporter** is used to upload iOS apps to App Store Connect. To install Transporter, simply search for Transporter in the Mac App Store and install it.



## 5. Installing Aeneas

Aeneas is the audio-text synchronization tool that may be run from within Reading App Builder to create timing files for phrase-by-phrase highlighting. If the apps you are building do not include audio or if the timing files are already available, then there is no need to install it.

To install aeneas:

1. Download the **aeneas tools** for Mac file (dmg) from the RAB website from the section labeled **Audio Synchronization Tools**:

#### http://software.sil.org/readingappbuilder/download/

- Double click on the file within Finder to open the disk image that contains the aeneas-mac-setup install package. Control-click on the install package and select **Open**. You will get a warning that this package is from an unidentified developer. Press **Open**.
- 3. The introduction screen will display. Press Continue.



- 4. A "Read Me" screen will display next followed by a "License" screen. Press **Continue** on both screens.
- 5. Pressing the **Continue** button on the License screen will bring up a screen asking if you agree to the terms of the license. Press **Agree**.
- 6. The "Destination Select" screen selects the default drive. Press **Continue**.
- 7. The "Installation Type" screen displays next for a standard install. Press Install.

• • •	🥪 Install aeneas tools
<ul> <li>Introduction</li> <li>Read Me</li> <li>License</li> <li>Destination Select</li> </ul>	Standard Install on "Macintosh HD" This will take 227.1 MB of space on your computer. Click Install to perform a standard installation of this software on the disk "Macintosh HD".
<ul> <li>Installation Type</li> <li>Installation</li> <li>Summary</li> </ul>	
	Change Install Location
	Customize Go Back Install

- 8. The installer will prompt for credentials to install the software. Enter a username and password with permissions and press **Install Software**.
- 9. At this point the installation will start and will show progress screens until it completes. A terminal window will popup briefly to test the installation. When the application completes successfully, the original screen will show:



#### 10. Press Close.

Aeneas is installed in /usr/local/lib/python2.7/site-packages.

# 6. Testing App in iOS Simulator

When you want to test your app, you can either use a device or a simulator. To test with a device, you will need a signing certificate and provisioning profile (see next section). To test with a simulator, you will need to download Xcode, the integrated development environment used to build and test iOS and Mac apps. Xcode is available for free in the Mac App Store and is quite large (5.46 GB). RAB requires Xcode 9 or greater (which requires macOS Sierra).

To install:

- 1. Go to <u>https://itunes.apple.com/us/app/xcode/id497799835?mt=12</u> or search for "Xcode" in the App Store app on macOS.
- 2. Install from the App Store
- 3. Start Xcode at least once to complete the installation

## 6.1. Run the iOS Simulator

Once you have a project configured and ready to test, click on the **Run iOS App in Simulator** on the toolbar.

• • •	Run iOS Simulator		
To launch an app in the simulat 1. Start the simulator	tor:		
Simulator Type	iPhone X - 11.1	0	Start
2. Build and launch the app			
Project Name	Kuna Gospels	0	Build
Last Built	Not Built Yet		Launch
			Close

Select the simulator you would like to run on and click **Start**. It takes a little bit of time for the simulator to start. If you want to switch simulators, select a different Simulator from the **Simulator Type** combo box and click **Start** again. It will close the previous Simulator and start the new own.

Select the project you want to test (it defaults to the selected project in the Apps list) and click on **Build**. This will build the app for the Simulator in a separate terminal window. When the build is complete, you can click on **Launch** to run the app in the Simulator.

You may close the dialog and make changes to your project. When you restart the Run iOS Simulator dialog again, you will need to Build and Launch again for the changes to be included.

## 6.2. Installing Additional Simulators

You may install simulators for previous versions of iOS by launching Xcode and viewing the preferences dialog and selecting the Components tab.

				Com	ponents				
@ eral Accounts	Behaviors	Avigation Navigation	A Fonts & Colors	Text Editing	Key Bindings	Source Control	Components	Locations	Server & Bot
				Sim	ulators				
Simulator									Size
iOS 11.0	Simulator								2.09 GB
iOS 10.3	.1 Simulato	r							1.98 G8
iOS 10.2	Simulator								1.93 GB
iOS 10.1	Simulator								1.9 GB
iOS 10.0	Simulator								1.89 GB
iOS 9.3	Simulator								
iOS 9.2	Simulator								1.49 68
iOS 9.1	Simulator								1.49 GB
iOS 9.0	Simulator								1.46 GB
iOS 8.4	Simulator								1.37 68
iOS 8.3	Simulator								1.36 GB
iOS 8.2	Simulator								1.32 GB
iOS 8.1	Simulator								1.17 GB
Check for an	d install s	imulator u	pdates autom	atically				Check an	d Install Nov

## 6.3. Manually Installing Apps into the Simulator

If there is some problem with launching the simulator from the **Run iOS Simulator** dialog, you can manually install the app by dragging the built app from the Simulator output folder (found in ~/App Builder/Reading Apps/Sim Output) onto the Simulator. To start the Simulator, launch **Xcode** and from the **Xcode** menu select **Open Developer Tool** > **Simulator**. You will still need to rebuild the app from the **Run iOS Simulator** dialog.

# 7. Creating iOS Certificates and Provisioning Profiles

#### 7.1. Enroll in the Apple Developer Program

To build iOS apps and distribute them through the Apple App Store, you will need to be enrolled in the Apple Developer Program. You can do this as either (i) an individual, or (ii) an organisation. The cost is USD \$99 per year.

To enroll:

- 1. Go to <a href="https://developer.apple.com/programs/enroll/">https://developer.apple.com/programs/enroll/</a>
- 2. Press the Start Your Enrollment button to start.

## 7.2. Create Signing Certificate

When you create an iOS app, it needs to be signed with a certificate.

To create a certificate:

- 1. Go to the <u>Apple Developer</u> website and log in to your account if you are not already.
- 2. Select Certificates, Identifiers & Profiles.
- 3. Click the <sup>•</sup> button to the right of the **Certificates** header.
- 4. On the **Create a New Certificates** page, select **Apple Distribution**. Then press the **Continue** button.
- 5. Upload a Certificate Signing Request. Press the **Learn more** link for instructions on how to use Keychain Access on your Mac computer to complete this task. Then press the **Continue** button.
- 6. On the **Download Your Certificate** page, press the **Download** button to download the certificate (distribution.cer) to your Mac.
- Open the Keychain Access application. Choose the login item from the Keychains section on the left. Choose the File ➤ Import Items... menu item and browse to the Downloads folder and select the certificate (distribution.cer).

Now that the certificate is installed in the Keychain, you will be able to access it from within Reading App Builder.

Note: To work with certificates, you will need the Apple Worldwide Developer Relations (WWDR) Certification Authority. It should have been installed with Xcode. When viewing your certificate in the Keychain Access application, if there is an error stating the certificate is not trusted, then install the certification authority.

To get the certification authority:

1. Download the Apple WWDR Certification Authority from: https://developer.apple.com/certificationauthority/AppleWWDRCA.cer

For signing certificates created after January 28, 2021: https://www.apple.com/certificateauthority/AppleWWDRCAG3.cer

- Open the Keychain Access application. Choose the System item from the Keychains section on the left. Choose the File > Import Items... menu item and browse to the Downloads folder and select the certification authority (AppleWWDRCA.cer). You will be prompted for a username and password that has admin privileges in order to modify the System Keychain.
- 3. Choose the **File ≻ Lock Keychain "System"** menu item.

## 7.3. Create Provisioning Profile

Distribution provisioning profiles are used to for two primary purposes:

- AdHoc to install your app on a limited number of registered devices for testing.
- App Store to submit your app to the App Store.

#### Creating a provisioning profile

To create a new AdHoc provisioning profile, you will need an App ID and at least one registered device. To create a new App Store provisioning profile, you will need just the App ID.

To create an App ID:

- 1. Go to the Apple Developer website and log in to your account if you are not already.
- 2. Select Certificates, Identifiers & Profiles.
- 3. Select **Identifiers** on the left of the page.
- 4. Click the  $\stackrel{•}{\bullet}$  button to the right of the **Identifiers** header.
- 5. Select App IDs and click the Continue button.
- 6. Select **App** and clock the **Continue** button
- 7. Enter a Description of your choice.It can be the App Name from the App > Package page of Reading App Builder.
- 8. For Bundle ID, choose **Explicit** and enter your app package name from the **App** ≻ **Package** page of Reading App Builder.
- Of the App Capabilities, there are only two that may need to be checked. Check the Push Notifications capability if the app uses Firebase Cloud Messaging or Daily Reminders. Check the Associated Domain capability if using Deep Linking.
- 10. Click the **Continue** button.

To register a <u>device</u> (i.e. a specific iPhone or iPad for testing):

- 1. Select **Devices** on the left of the page.
- 2. Click the  $\textcircled{\bullet}$  button to the right of the **Devices** header.
- 3. In the section **Register a Device**, enter a name of the device (such as "John Smith's iPhone") and its UDID (unique device identifier, a sequence of 40 letters and numbers). Press **Continue**.
- 4. On the next page, check that the device information is displayed correctly and click **Register** to confirm.

Note that you can register up to 100 devices of each type (e.g. up to 100 iPhones, 100 iPads) per year of your Apple Developer Program membership. You can remove devices that you no longer need at the beginning of the next membership year.

To create a provisioning profile:

- 1. Select **Profiles** on the left of the page.
- 2. Click the <sup>•</sup> button to the right of the **Profiles** header.
- 3. On the **Register a New Provisioning Profile** page, under **Distribution**, select **Ad Hoc** (for testing) or **App Store** (for submission to the App Store). Click **Continue** to move to the next page.
- 4. On the 'Select an App ID' page, select the App ID from the list of App IDs you have already defined. Click the **Continue** button.
- 5. On the 'Select Certificates' page, select the certificate (Distribution) to include in the profile. Click the **Continue** button.
- If creating an AdHoc provisioning profile:
   On the 'Select Devices' page, check the device(s) that you want to be able to install and run the app. Click the **Continue** button.
- 7. On the 'Review, Name, and Generate' page, give the profile a name of your choice. It is helpful to include "App Store" or "AdHoc" in the name. Click the **Continue** button.
- 8. On the 'Download and Install' page, click **Download** to save your new .mobileprovision file to your computer.

This is the Provisioning Profile file that you will select in Reading App Builder.

#### Download existing mobile provision files

If other members of your team have already created provisioning profiles, they can be downloaded from the App Developer website by selecting the profile to be downloaded and pressing the **Download** button.

# 8. Building an iOS App

The first step in building an iOS app is to create a new app project following the instructions in the RAB **Building Apps** document. Then follow the instructions below.

## 8.1. Application builds available for iOS

For iOS distribution, three different app types are available.

#### 8.1.1. Dedicated App

This is the standard traditional app and is the default. The content that is defined for this project will be used to generate an iPA file that may be delivered to the App Store for publication. A dedicated app could be viewed as a container app that only runs one preselected asset package that is included in the app at build time.

#### 8.1.2. Container App

App Builder users have encountered issues with the Apple App Store reviewers if too many apps generated by the App Builder are published to the same account. Since the actual application differs only in the content provided, Apple rejects the app as being spam. To avoid these issues, a container app can be generated instead of the standard dedicated app.

The container app does not contain content of its own. It does not have book collections, audio or any of the other normal content. Instead, on its initial startup, it accesses a web site that is maintained by the app developer. The web site provides specially formatted links to multiple asset packages. The app user selects one of these asset packages. The container app downloads it and runs using the downloaded content. From this point, it resembles the dedicated app, running this set of content without further downloads being required.

The user of a container app does have the option of resetting the content at any time. If the user elects to reset the app content, the app asset package currently in use is deleted from his device. The web site accessed at the initial startup is displayed, and the user can once again select the asset package to be run from the available list. The selected package is then downloaded and run in place of the discarded one.

Building a container app will generate an IPA file similar to the dedicated app with the exception that no content is included with the app beyond some configuration information and a reference to the website used by the app. The IPA file is submitted to the Apple App Store in the same manner as the dedicated app.

While most of the settings associated with the app need to be set in the asset package projects, there are a few settings that must be set at the container app level. The first of these are the Firebase related features. Firebase tracks at the app level, and the settings for the *Firebase* tab and the *Security* tab will apply for all app asset packages associated with this container app. The GoogleService-Info.plist file that is used by the iOS Firebase configuration should be associated with the Package Name for the container app set on the package tab. Firebase will track information associated with the container app, not with the individual asset package projects. A project field is included in events generated by asset packages and sent to Firebase and can be used to identify the events that originated with a particular package that is being distributed.

Several other features follow the settings in the asset package project definition but must be accommodated by the container app that loads them. If any of the asset packages that can be loaded by this container app application use *Daily Reminder* features on the *Notifications* tab, these options must be enabled in the container app as well as in the asset package that wants to use the feature. The *IPA* tab also includes a checkbox that must be selected if any of the asset packages include audio files.

On the *Images* tab, the *iOS Icon* and the *iOS Splash Screen* (optional) should be configured for the container app as these will be the images used. The images set in asset packages for these two items will not be used.

#### 8.1.3. Asset Package

Asset packages are used in conjunction with Container Apps. The configuration of an asset package project should be the same as it is for a dedicated app, setting the book collections, audio and feature required for that app. The only exceptions are those mentioned in the section above on Container Apps. However, where the build of Dedicated Apps or Container Apps results in an IPA file to submit to the Apple App Store, the build of an Asset Package project results in a zip file, saved to the IPA output folder, that is intended to be saved to a location where it can be referenced by the web site supporting the Container App. A user running who has downloaded the Container App from the App Store will select this package from the list presented by the Container App Web Site, at which point it will be downloaded to the device, decompressed and run as the project for the app. All of the book collection, audio, font, images, features and configuration information that is usually embedded within the dedicated app is saved in this zip file so that the project can be run as it would if it were configured as a dedicated app.

Dedicated Apps and Container Apps can only be generated on an Apple Mac device. Asset Packages do not require a Mac. Therefore, App Builder can build a project configured as an iOS Asset package on a Mac, a Linux device, or a Windows device. On a Mac Device, selecting *Build iOS App* for a project configured as an Asset Package will generate the zip file in the IPA Output Folder. For testing purposes, so that the app developer can test the device outside of the container app, selecting *Run iOS App in Simulator* will cause an IPA file to be generated in the Simulator Output Folder that will run using the project configuration. The app developer can load the IPA file on an iOS device simulator and test the project before generating the zip file and adding it to their Container App web site.

#### 8.1.4. Container App Website

While not generated by App Builder, a Container App project must have a web site where the asset packages are located and a web page is defined by the user to allow selection of the asset package. The Container App displays the web page referenced by the URL that is part of its configuration. Anything other than links that reference *asset://yourwebsite/yourassetpackage.zip* or <u>https://[something].zip</u> or <u>http://[something].zip</u> are passed through the browser and are displayed to the user. When the user selects a link that starts with *asset://* or a normal URL that ends in *.zip*, that is interpreted as the selection of an asset package. If the URL starts with *asset://*, the *asset://* in the link is replaced with *https://* to generate the destination URL. The file referenced is downloaded, unzipped, and used to initialize the Container App. A very simple example of a Container App Website web page would be:

```
<html>
<head>
<style>
.container {
display: flex;
justify-content: center;
background-color: powderblue;
font-size: 40px;
}
.center {
width: 800px;
}
</style>
</head>
<body>
Container App Test
<div class="container">
Select a translation
<a href="asset://sab-assets-test.s3.amazonaws.com/EnglishGreek2.zip">English</a>
Greek</a>
<a href="https://Test.zip">Test</a>
</div>
</body>
</html>
```

## 8.2. App Type (iOS) Tab



The *App Type* information identifies the type of application being built, as defined in the previous section. Select either *Dedicated App, Container App,* or *Asset Package*.

If the *Container App* option is selected, the *Container App URL* needs to be entered. The URL entered here is the web page that will be displayed to the user by the container app to allow them to select the asset package to be used. The checkbox underneath it must be selected if any asset package used by the Container App will contain audio for books.

If the *Asset Package* option is selected, the *Filename* field under the *Asset Package* option is enabled. This field defines the base name, without path, that will be assigned to the Asset Package zip file. The file will be created in the IPA Output folder when the user selects *Build iOS App* (or on a Linux or Windows system, selects *Build iOS App Asset Package*.

## 8.3. IPA Tab



Click on the IPA tab.

This allows you to set the name of the ipa file to be generated as well as the build and version information.

The remaining fields, under the *IPA File* and *Apple Id* headers are only enabled for a *Dedicated App* or *Container App*.

The *Filename* field on the screen for the *IPA File* section specifies the base name of the ipa file to be generated. If the checkbox at the bottom of the screen for *Append version name to ipa filename* is checked, then the version indicated by the *Version Name* fields is added to the base filename.

The **Build** field referenced as *Build* is also called *Bundle Version String, Bundle Version* or *CFBundleVersion* within Xcode and ITunes Connect. It represents the build number. The *Build* field expects an integer value and should be incremented with each file that is submitted to the ITunes Connect for release or testing.

The **Version Name** field is referenced as *Version, Bundle Short Version String, Bundle versions string, short* and *CFBundleShortVersionString* within Xcode and ITunes Connect. The field is created as a concatenation of the values of the three fields separated by a period. If the final field has a value of 0, then the version string is created from just the first two values. So for values of 1, 2 and 0, the resulting string is "1.2". For values of 1, 2 and 3, the resulting version string is "1.2.3".

The **Apple ID** field is the id assigned by App Store Connect to the application in the app store. It can be obtained from **Apple ID** filed on the App Information tab in the App Store Connect entry for the app as shown below

App Store Connect		Apps	Users and Access	
💮 Kuna Gos	pels ~ App Store Features	TestFlight		
iOS App 2 1.0 Prepare for Submission	App Information This information is used for all platforms of this ap	p. Any changes will	be released with your ne	ext app version.
Add macOS App Add tvOS App	You can now set your app's age rating from describe your app's content more accurate	n the App Informatio	on page. The Gambling a	nd Contests content descript
General	describe your app's content more accurate	ely. Eult Age Ratilig		
App Information	Lessizable information			
Pricing and Availability	Localizable information			
App Privacy	Name ?			
Ratings and Reviews	Kuna Gospels			
Version History				
In-App Purchases	Subtitle ?			
Manage				
App Store Promotions				
	General Information			
	Bundle ID ?			Primary Language ?
	org.wycliffe.app.test.cuk		~	English (U.S.)
	SKU ?			Category ?
	KUNA			Primary
	Apple ID ?			Secondary (optional)
	1346657481			

The **Bundle ID** section allows the user to specify a Bundle ID for the app that is different than the package name. By default, the *Package Name* field from the *Package* tab is used as the Application Bundle ID that is used in creating the provisioning profiles and as the application identifier for the Apple App Store. If the user wants the iOS version of the app to have an identifier that is different than the package name that is used for the Android version of the app, press the *Specify a Bundle ID* button and then enter the ID to be used in the text box labelled *Bundle ID*.

#### 8.4. Signing (iOS) Tab

1. Select the **Signing (iOS)** tab to open the iOS signing options for the app.



- 2. Select the **Signing Identity** from the drop down list of signing certificates which have been downloaded and installed to this system in the earlier steps.
- 3. For the **Provisioning Profile** entry, enter or browse to the mobile provisioning file associated with the app that was downloaded in the earlier steps.

4. Click on the **Build iOS App** button at the top of the screen. A terminal window should open. The build script for the iOS App should run within that terminal window.

		Reading App I	luilder					
ew App   Save All   Build Android Ag	p Build IOS App Build Mu	hiple Apps Run IOS Ap	p in Simulator					
Apps My Stories	My Stories: App							
App About Box Analytics	App Name Package	Project APK	Signing (Android)	Signing (iOS)	Expiry	Security	Permissions	
Audio Epub Images Interface Features	IPA File The IPA file is the iOS a via the Apple App Store	ipp package file genera e.	ted by App Builder. This	s is the app insta	allation file	r you distrib	oute to others	
Changes	Filename	My_Stories.ipa						
Colors	Build	1 C (Add 1 for	every app release. Not	seen by user.)				
Styles	Id lagrag ab 00-04		) Ceen by u	ser.)				
iOS Icon Not Found Icon file ITunesArtwork92x.png r	not created for App Store	1						
Signed release IPA built success	sfully.							
-rw-rr 1 hubbard staff 77 eading Apps/Ipa Gutput/My_Storid logout Saving session copying shared history saving historytruncating h completed.	7921301 Dec 1 13:59 /Use os-1.0.ipa	rs/hubbard/App Builde	r/R					
(Process completed)	completed.							

- 5. Examine the terminal window once the shell script has been completed. The message "Signed release IPA built successfully" should appear in the window if the app has been built successfully. (Note that occasionally the terminal window will appear behind the Scripture App Builder and that you have to select the terminal to review the results).
- 6. The results of the build are an IPA file and an app that can be run in the simulator. They can be found in ~/App Builder/Reading Apps/Ipa Output/ and ~/App Builder/Reading Apps/Sim Output/.

Nam	9	^	Date Modified	Size	Kind
▶	Dictionary Apps		Oct 30, 2017, 10:11 AM		Fold
▶	keystore		Nov 16, 2017, 3:30 PM		Fold
<b>v</b> 1	Reading Apps		Today, 2:01 PM		Fold
1	App Projects		Today, 1:50 PM		Fold
	🕫 📄 Ipa Output		Today, 1:59 PM		Fold
	My_Stories-1.0.ipa		Today, 1:59 PM	77.9 MB	iOS /
	🖉 🚞 Sim Output		Today, 2:02 PM		Fold
	My_Stories-1.0		Today, 1:59 PM	57.2 MB	Appl
1 H	Scripture Apps		Nov 13, 2017, 12:16 PM		Fold

# 9. Testing an iOS App

After building the iOS IPA file, you will want to install it and test it on one or more devices before you submit it for publication to the Apple App Store. This manual describes two ways of doing this:

1. Use **Xcode** to install the IPA file to an iPhone or iPad that is connected to your computer.

This method is recommended if you have your test devices with you. It does not involve uploading and downloading the IPA to and from the internet.

2. Use **DeployGate** to upload the IPA file to the internet and share it with limited number of devices to download, install and test.

This method is recommended if you have good internet access and/or you have a team of testers who are elsewhere.

# 10. Using Xcode to Test an iOS App

To test your iOS IPA file using **Xcode**, do the following:

- 1. Launch Xcode and select Window > Devices and Simulators.
- 2. Connect an iPhone or iPad to the Mac using a cable and unlock the device.
- 3. On the Mac, iTunes may launch and show a dialog asking "Do you want to allow this computer to access information..." Click on the **Cancel** button.
  - Note: This feature can be turned off in iTunes. Select iTunes > Preferences..., select the Devices tab, and click on the Prevent iPods, iPhone, and iPad from syncing automatically checkbox.
- 4. On the device, it may prompt to **Trust This Computer**. Tap on the **Trust** button. This may require to enter the Passcode to trust this computer. The device will show up in the **Xcode Devices** window.
- 5. The first time the device is connected to the Mac, Xcode will take some time to Prepare debugger support. This may take some time. Please wait.
- 6. In the **Xcode Devices** window, there will be a section named **INSTALLED APPS**. Click on the + button.
- 7. Find the IPA file to add. Click **Open** after you have selected it.
- 8. Click the **Install** button next to the name of the app.
- 9. Wait until the Mac installs the app to your device.

After the install is complete, you will see the app icon on your device and you can test it.

# 11. Using DeployGate to Test an iOS App

**DeployGate** enables you to test your app and share it with a limited number of users to test. You upload the IPA file for iOS apps or the APK for Android apps and then download them to your phone or tablet device. You can also invite testers to install your app and help with the testing.

A DeployGate app is installed on the testing device. It will show all of the apps that you have uploaded to DeployGate and allows them to be installed on the device.

## 11.1. Creating a DeployGate Account

The first step is to create a DeployGate account. To do this:

- 1. Go to <u>https://deploygate.com</u>
- 2. Press the Get Started button.
- 3. Enter an email address, a user name and a password.
- 4. Press the Sign up for DeployGate button.

Features Pricing Developer Tools Support Log in Sign up
Email
mail@addr.dom (Gravatar support)
Username at least 3 characters
Password
at least 6 characters
By clicking the button, you agree to the terms.

## 11.2. Uploading your first app

Once the sign in screen has been successfully completed, you are presented with a screen that prompts you to upload your app. While there are several methods described for embedding it as part of your build process, the way we have used this to date is to simply upload the IPA file that has been created by locating the ipa file in Finder and then dragging it to the bottom area of the screen where it has a green **Upload App** area:

niunoorei	769 walcome to DeployCatel
	os, welcome to pepioyoate:
DeptoyGate makes it	easy to share in development iOS and Android apps,
allowing users to seamlessly and successfully p	rogress through the prototyping, development, testing, and marketing stages of app creation.
Let's get started	with your own device and see how it works:
1, Stage your app	▶ 2. Log in on your device ▶ 3. Install
it iOS app	Android app
Copy and pacts the following command in Termin	el app Sótr your build gradie to get started.
Contraction of the second second	Open your build, graffe on your project not and will a dependency.
	Sanstath "con-deployatespredies]; 8.4" // add this Time
and a man	
Keyf, edit fölgar engligget samertarter abs// bais	•
Fort etc. Concerning anget and etc. The abs [ bain	•

Dragging the file to the upload area causes a new dialog to be displayed with the name of the file and a text box where you can enter a short note that will be displayed on your profile window and also on the DeployGate app when the user is selecting the app. It is a good place to write a short note on the reason for the update so that it is easy for the testers to see that the app has been updated and to see what the primary reason they need to update is. Complete the screen and press the **Upload** button.

deploy <b>gate</b>	Upload App File	er Tools	Support
	Filename San_Blas_Kuna.ipa Release note Testing documentation		
Hic		ate!	
Dej	Privacy Mode Don't show on your profile (Upgrade my account) Your app packages and data are always kept privatel on DeployGate. Choose this to hide even title and icc from your profile and make the app completely secre	y in et. )ps,	
allowing users to seamlessly a	Cancel	sting, ar	nd marke

When the upload is complete, a new dialog is displayed with a QCode bar code and the option to send an email to your device. The QCode can be read in with your iPhone or iPad which will trigger an installation of the DeployGate app using your app profile.

Alternatively, you can enter an email address at this point, which you would also open on the iPhone to install the DeployGate app with the correct profile. Or you can simply go on and add users and devices later.



After this, the screen that is displayed is what you will normally see when you login. The screen has an entry for each app that you have uploaded. It has an **Upload App** button that can be used to upload new versions of the same app or to upload a new app.

deploy <b>gate</b>		Dashboard Den	veloper Tools Support 🥅 drisoret 768
org wyciffe app cu San Blas #1 Testing docu	Kuna Marriel Males	★0 Visible on profile	ADISTRIBUTIONS
Lipicad App Send to Your Own	Devices: BI QR Code B Email		epp. O Add a link for sharing
		O Options -	A TESTERS (0)
You may write some notes on this application or p	uides for your tetters here.		invite multiple users in one go?
Activities Recent activities on this app.			% API KEY
	III Al Crashes		97e0d5e12d570292c954303 %
Contraction a new app "San Blas Kuna".		.1 minute a	This is the API Key to be used in DeployEate SOK for IOS.
No email notifications sent for no UDID of	hanges or no 105 devices.		Refresh API key?
			DEVICE DATA 50
			= No tracking data yet.

#### **11.3.** Registering a Device

If the iOS device was not originally in the mobile provisioning profile and if the device has not been previously registered in your Apple Developer account, you need to add it to both. There is a method for manually doing this, but DeployGate provides a way of simplifying the process so that you don't need to go and look up UDID for the device.

First, try to install the app using the method described above. You will not be able to install it because the UDID is not registered for the app. However this will result in the device being registered in DeployGate which allows the following steps.

After attempting to install the app, re-enter DeployGate in your browser and open the entry for your app. As you can see in the screen below, it will show that a new UDID has been registered for the device. Press the **Options** button below and select the **Package Archive** option. Next click the little tag symbol inside the app box to open the UDID list.

deploy <b>gate</b>	Dashboard Develo	per Tools Support 🕅 dmoore1768
org:kalaam.test.ScriptureApp 1.0 (2) Test Universal Ad Heil It Kalaam profile Upload App	★0 Visible on profile	Add a link to share this app with others. Each link can have a different version of app.
Send to Your Own Devices: Bi QR Code 🗃 Email	O Options -	🛦 TESTERS (0)
You may write some notes on this application or guides for your testers here.	▲ Download app package     ⊌ UDID Lookup     ≡ Package Archive	Username or Email + metermultiple users in one got
A mail data a s	曾 Delete this app	% API KEY
ACLIVICIES Recent activities on this app.	·	950cb54905d3224de59c9d5( %) This is the API key to be used to be level to be level to be set of the optimized to be set of the op
O dmoore1768 registered a new UDID. by dmoore1758 This devices UDID is not included in the app. Please export it and re-archive your app from the link below	about 5 hours ago	Battech Alft bayt
contraction of the second seco	about 5 hours ago	
No email notifications sent for no UDID changes or no IDS devices.		- No tracking data yet.

The next screen shows a list of the devices that have been observed by DeployGate or that were included in the provisioning profile. Your new device entry will show up on the screen with a **Not Exist** entry. Make sure that the entry for the new device is checked and then press the **Export Selected UDIDs** button. This will create a file "multiple-device-upload-ios.txt" that can be used on the Apple Developer website to add these devices to the mobile provisioning file.

### deploy**gate**

<b>() Test</b> by dmoore1	768				
Test / Package Archive / UDID List of Memi	bers and Revision #1				
If some UDIDs are "Not Exist" in the profile, this app can't be installed to those devices. In that case, export UDIDs as a file, then import it at "Certificates, identifiers & Profiles", re-create a profile with those devices. Export Selected UDIDs Show 10 8 entries Search Q					
how 10 ¢ entries			Search:	Q,	
how 10 s entries	© PRODUCT © U	¢ digt	Search PROVISIONING PROFILE	Q	¢
how 10 s entries	0 PRODUCT 0 U iPad 3 (WiFi) 9	JDID 0 P0a0d2b9979a47a5d96bb1a1369e0db545ad917	Search: PROVISIONING PROFILE Not Exist	٩	0
how 10 ; entries • USER 0 ROLE C C	PRODUCT     U      iPad 3 (WiFi)     9      iPhone 5 (Global)     3	JDID 0 PDa0d2b9979a47a5d96bb1a1369e0db545ad917 Ic237054805042594f6d8814c7972f931e4ec5b2	Search: PROVISIONING PROFILE Not Exist	٩	0

After logging in to Apple Developer, click on **Certificates, IDs and Profiles**. Press the **All** selection under **Devices** as shown in the illustration below and then select **Register Multiple Devices**. Then press the **Choose File** button.

iOS, tvOS, watchOS *	Add Devices	(+) (×) (Q)
Certificates All Pending Development Production	Registering a New Device or Multiple Devic	es
<ul> <li>Identifiers</li> <li>App IDs</li> <li>Pass Type IDs</li> <li>Website Push IDs</li> <li>ICloud Containers</li> <li>App Groups</li> <li>Merchant IDs</li> </ul>	Pre-Release Software Reminder You may only share Apple pre-release software with employees, contractors, and men organization who are registered as Apple developers and have a demonstrable need to software to develop and test applications on your behalf. Unauthorized distribution of Apple confidential information (including pre-release so may result in the termination of your Apple Developer Program. It may also subject yo liability.	nbers of your o know or use Apple ftware) is prohibited and u to civil and criminal
Devices E All E Apple TV Apple Watch E iPad E iPhone E iPod Touch Provisioning Profiles E All	Register Device     Name your device and enter its Unique Device Identifier (UDID).     Name:     UDID:	
Development     Distribution	Register Multiple Devices     Upload a file containing the devices you wish to register. Please note     100 devices can be included in your file and it may take a few minute     Download sample files     Choose File_	that a maximum of es to process.

Find the multiple-device-upload-ios.txt file that was created by DeployGate and then press **Continue**.

Prending     Development     Production	Registering a New Device or Multiple Devices
<ul> <li>Identifiers</li> <li>App IDs</li> <li>Pass Type IDs</li> <li>Website Push IDs</li> <li>ICloud Containers</li> <li>App Groups</li> <li>Merchant IDs</li> <li>Devices</li> </ul>	Pre-Release Software Reminder You may only share Apple pre-release software with employees, contractors, and members of your organization who are registered as Apple developers and have a demonstrable need to know or use Apple software to develop and test applications on your behalf. Unauthorized distribution of Apple confidential information (including pre-release software) is prohibited and may result in the termination of your Apple Developer Program. It may also subject you to civil and criminal liability.
All Apple TV Apple Watch FiPad Fipa	Register Device         Name your device and enter its Unique Device Identifier (UDID).         Name:         UDID:
<ul> <li>Development</li> <li>Distribution</li> </ul>	<ul> <li>Register Multiple Devices         Upload a file containing the devices you wish to register. Please note that a maximum of         100 devices can be included in your file and it may take a few minutes to process.         Download sample files         Choose File         multiple-device-upload-ios.txt     </li> </ul>
	Cancel Continue

A review screen will be displayed which should list the name to be assigned to the device along with the UDID associated with the device. Review to ensure this is correct and press **Register**.

<ul><li>Pending</li><li>Development</li><li>Production</li></ul>	Review and register.
<ul> <li>Identifiers</li> <li>App IDs</li> <li>Pass Type IDs</li> <li>Website Push IDs</li> <li>iCloud Containers</li> <li>App Groups</li> <li>Merchant IDs</li> </ul>	Confirm the device information is correct. Once this device is registered, you will not be able to edit the UDID and can only edit the name or disable it. Name: dmoore1768 - iPad 3 (WiFi) Model: iPad Wi-Fi (3rd generation) UDID: 9f0a0d2b9979a47a5d96bb1a1369e0db545ad917
Devices	
<ul> <li>Apple TV</li> <li>Apple Watch</li> <li>iPad</li> <li>iPhone</li> <li>iPod Touch</li> <li>Provisioning Profiles</li> <li>All</li> <li>Development</li> <li>Distribution</li> </ul>	You can register 96 more of this device type. The maximum number of each device type that you can register per membership year is: Apple TV: 100 Apple Watch: 100 iPad: 100 iPhone: 100 iPod Touch: 100 You may reset your device list at the start of your next membership year.
	Cancel Back Register

At this point your device has been registered to your Apple Developer account. You now need to add the device to the provisioning profile for your app. Select the provisioning profile being used to test the app. Make sure your test device is checked in the list of devices at the bottom of the screen. Press **Generate**.

**Note:** For the purposes of testing with DeployGate, an AdHoc type of provision profile must be created and used. If you have not selected AdHoc, the list of devices will not be available on the screen.



The following screen will display:



Now you can download the new provisioning profile that has been generated.

Next you need to rebuild the iOS app using the new profile and upload it to DeployGate again. This time when you attempt to install it, the **Install** button should be enabled and will allow you to install your app.

# 12. Uploading an iOS App to Apple App Store

Before attempting to upload the app, you will need to create an entry in App Store Connect (<u>https://appstoreconnect.apple.com</u>).

Select the appropriate distribution **Signing Identity** and **App Store** provisioning profile on the **Signing (iOS)** tab and click Build iOS App. This will create an IPA file in the IPA Output Folder.

Launch **Transporter** and click **Sign In** using the Apple ID for your Apple Developer Account. Drag and drop the IPA file from the IPA Output Folder to Transporter. Once the app is added to Transporter, click on the **Deliver** button. After it is uploaded, you can click on the ... button and select **View in App Store Connect**. Selecting the Activity tab will show the status of the processing of the upload.

# 13. Using Test Flight to Test an iOS App

It will take a little bit of time (around 20 minutes) for the app to be processed in App Store Connect. You will receive an email when the app is done processing. Switching to the Test Flight tab in App Store Connect, you will see that app is Missing Compliance.



Click on the build number and you will be taken to a page where you can click on **Provide Export Compliance Information**. The app uses an encryption algorithm to protect the text of the app.

Click Yes to the first dialog that the app uses encryption and then click Next.

Export Compliance Information

Does your app use encryption? Select Yes even if your app only uses the standard encryption within Apple's operating system.



Click Yes to the second dialog to indicate that the app qualifies for an exemption due to (b) Limited to intellectual property and copyright protection and then click Start Internal Testing.

Export Compliance Information

Does your app qualify for any of the exemptions provided in Category 5, Part 2 of the U.S. Export Administration Regulations?



Make sure that your app meets the criteria of the exemption listed below. You are responsible for the proper classification of your product. Incorrectly classifying your app may lead to you being in violation of U.S. export laws and could make you subject to penalties, including your app being removed from the App Store.

You can select Yes for this question if the encryption of your app is:

- (a) Specially designed for medical end-use
- (b) Limited to intellectual property and copyright protection
- (c) Limited to authentication, digital signature, or the decryption of data or files
- (d) Specially designed and limited for banking use or "money transactions"; or
- (e) Limited to "fixed" data compression or coding techniques

You can add App Store Connect users (normally users in your organization) to test the app. There is a link at the left for **Add External Testers**. This will require the app to go through Beta App Review.

# 14. Apple Privacy Policy

Prior to publishing your app through **App Store Connect**, you will be required to complete the privacy policy details section. The answers to these questions will be used to create the privacy policy section in your app store entry that describes to the user how you are using their data. The types of data associated with each question is documented by Apple at <a href="https://developer.apple.com/app-store/app-privacy-details/">https://developer.apple.com/app-store/app-privacy-details/</a>. The answers associated with the SAB app depend upon the use of analytics and what data is saved if you have user registration defined for your app.

# 14.1. Data Types

Under the initial *Data Collection* screen in this section, if the app does not use analytics or user registration, you may indicate that the app does not collect any data. If either of these features are in use, then you should respond that we do collect data from this app. If you answer *Yes* to this question, then you need to specify in the next screen the types of data collected. For an app that has analytics enabled, *Product Interaction* under *Usage Data* and *Crash Data* under *Diagnostics* should be selected. If the app has User Registration enabled, then some of the fields under *Contact Info* may need to be checked depending upon the fields you have configured for the user to enter.

# 14.2. Product Interaction

This section should only be required if analytics or user registration are enabled. If analytics is enabled, then the *Analytics* entry should be checked to indicate that the data is being used to track user behavior. If user registration is enabled, you may also want to check the *App Functionality* entry. On the next screen, you should check the "*No, product interaction data collected from this app is not linked to the user's identity*" entry. On the final screen, regarding tracking, you may click the "*No, we do not product interaction data for tracking purposes*".

# 15. Building from Terminal

Reading App Builder has a command line interface which allows you to create a new app and build it, or load an existing app and build it.

The base command calls java to access the jar file within the Reading App Builder application followed by a series of options described below. The base command is:

java -jar "/Applications/Reading App Builder.app/Contents/Java/bin/reading-appbuilder.jar" The available parameters are:

Option	Description
-new	Create a new app project
-load <project></project>	Load an existing app project
-build	Build app project (use with either -new or -load)
-no-save	Do not save changes to app (use with -load)
-resign	Resign iOS Template App (use with either -new or -load)
-?	Show command line help
-n <app-name></app-name>	Set app name. Enclose the name in "double quotes" if it contains spaces.
-p <package-name></package-name>	Set package name, e.g. com.myorg.language.appname
-b <filename></filename>	Add book or bundle file. This could be a USFM file or a zipped set of USFM files. It could also be a Digital Bible Library text release bundle.
-i <filename></filename>	Include additional parameters file. Use the full path of the file and enclose it in "double quotes" if there is a space in the path.
-a <filename></filename>	Set about box text, contained in text file. Use the full path of the file and enclose it in "double quotes" if there is a space in the path.
-f <fontname></fontname>	Set font name or filename, e.g. "Charis SIL Compact", "c:\fonts\myfont.ttf" The font name must be one of the items in the list of fonts in the New App wizard. For other fonts, specify the full path to the font filename.
-g	Use Grandroid
-ic <filename></filename>	Add launcher icon (one or more .png files). Use the full path of the files and enclose them in "double quotes" if there is a space in the path.

-l <lang-code></lang-code>	Set language for menu items and settings, e.g. en, fr, es
-ft <feature=value></feature=value>	Set a feature, e.g. book-select=grid
-vc <integer></integer>	Set version code, e.g. 1, 2, 3, or +1 to increment the current version code by 1.
-vn <string></string>	Set version name, e.g. 1.0, 2.1.4, or use +1, +0.1, +0.0.1 to increment the current value.
-ks <filename></filename>	Set keystore filename. Use the full path of the file and enclose it in "double quotes" if there is a space in the path.
-ksp <password></password>	Set keystore password
-ka <alias></alias>	Set key alias
-kap <password></password>	Set key alias password
-fp <folder=path></folder=path>	Set a folder path, e.g. "app.builder=/Developer/Reading App Builder".
-ta <target-api></target-api>	Set Target API, e.g. 21 for Android 5.0, 22 for Android 5.1.
-si <signing identity&gt;</signing 	Set Signing Identity to use for iOS Resigning
-pp <provisioning profile&gt;</provisioning 	Set full path to provisioning profile for iOS resigning
-bn <integer></integer>	Set build number for ipa file, e.g. 1, 2, 3, or +1 to increment by 1
-vs <string></string>	Set version string for ipa file, e.g. 1.0, 2.1.4 or +1, +0.1, +0.0.1

#### **Examples:**

Java -jar

"/Applications/Reading App Builder.app/Contents/Java/bin/reading-app-builder.jar" load "Mali" -resign -bn "5" -vs "2.3.2" -si "iPhone Distribution: Summer Institute
of Linguistics, Inc (SIL) (4YF5X97M4H) " -pp
"/Users/builder/Documents/MobileProvision/AdHoc\_org.wycliffe.app.mali.mobileprovisi
on"

# 16. Using Firebase in an iOS App

The *Building Apps* document contains general information about setting up the app to support Firebase Analytics, Crashlytics, Messaging and Real Time Database. This section will focus on the steps required to make these features work with an iOS app.

## 16.1. Adding an iOS App

The first step is to go to your Firebase Console (<u>https://console.firebase.google.com/u/0/</u>) and select the entry for the application you are working on. Press the "Add App" button, then select the iOS button.



The next screen will request general information about your application. The iOS bundle ID should match the Bundle ID for the app. This is the SAB Package setting from the Package Tab for the App unless a separate Bundle ID is set in the IPA Tab . The App Nickname can be set to whatever you want this app to be referenced as within Firebase. The App Store ID is the same as the Apple ID field in the SAB IPA tab described in Section 9. Once these fields are entered press "Register App".

×	Ac	dd Firebase to your iOS app	
	1	Register app	
		iOS bundle ID 💿	
		org.sil.sample	8
		App nickname (optional)	
		Sample App	
		App Store ID (optional) ⑦	
		1234567890	
		Register app	
	2	Download config file	
	3	Add Firebase SDK	
	4	Add initialization code	
	5	Next steps	

The next screen allows you to download the config file for the application. Press "Download GoogleServiceInfo.plist" to download the Firebase configuration file to your local machine. It should be mentioned here that if your app has the Real Time Database option set in RAB and you have not yet configured the Real Time Database within Firebase Console, you should either do that first or download the plist file again after that configuration has been completed. Download the file to a location where you can locate it. Later in this section, instructions will be provided for adding this file into your RAB application. After downloading the file, press "Next" to continue.



The screens which follow the download step are all instructional screens providing information about programming changes that need to be made to add Firebase to an app. This work has already been done within RAB and can be ignored here.

The final screen just provides a means to return to the console. The addition of the iOS app to the Firebase console is complete.

×	Ac	ld Firebase to your iOS app
	<b>?</b>	Register app iOS bundle ID: org.sil.sample, App nickname: Sample App, App Store ID: 1234567890
	0	Download config file
	0	Add Firebase SDK
	0	Add initialization code
	5	Next steps
		You're all set!
		Make sure to check out the <u>documentation</u> [2] to learn how to get started with each Firebase product that you want to use in your app.
		You can also explore sample Firebase apps 2.
		Or, continue to the console to explore Firebase.
		Previous Continue to console

## 16.2. iOS Configuration for Firebase in SAB

After the app has been added to the Firebase configuration through Firebase Console, the configuration file that was downloaded needs to be added to SAB. On the **Firebase** tab, in the **Firebase Configuration** section on the bottom half of the screen, press the **iOS** tab. Press the **Browse** button to locate the *GoogleService-Info.plist* file that was downloaded during the Firebase configuration process above. Select that file to add it to the app configuration. Note that if Firebase features are checked as enabled, the iOS app will not build until the plist file has been added.

y stories:	Арр											
pp Name	Package	Project	APK	IPA	Signing (Android)	Signing (iOS)	Expiry	Firebase	Deep Linking	Security	Permissions	
Firebase c specify the	an provide a configuratio	range of to on details b	ols for yo elow.	our app	such as analytics, o	crash reporting a	nd push no	otifications. T	o use Firebase in	ı your app, <u>c</u>	reate a Firebase projec	tor the app and
Firebase F	eatures											
Which Fire	base feature	s would yo	u like to	use in t	nis app?							
🗸 Fire	base Analyt	ics (for trac	king app	usage	and user engagemer	nt)						
🗸 Fire	base Crash	lytics (for cr	ash repo	rting)								
🗸 Fire	base Messa	aging (for pu	ush notifi	cations)								
🗹 Fire	base Realti	me Databas	se (for sa	ving us	er details)							
Firebase C	onfiguratio	n										
Android	iOS V	Veb										
Ornerit	the Fireber		tion file f		Occurring Info m	lind) This can be			Firebasa Decisat	0		
Specing				011031	Googledervice-Inito.p	hisy. This carrie	Guinnau	ieu nom you	Filebase Flojeci	Settings pai	ye.	Desuga
xr<br D0</td <td>CTYPE plist</td> <td>PUBLIC -</td> <td>//Apple//</td> <td>-8°?&gt; DTD PL</td> <td>IST 1.0//EN" "http://</td> <td>www.apple.com/</td> <td>DTDs/Prop</td> <td>ertyList-1.0.0</td> <td>dtd"&gt;</td> <td></td> <td></td> <td>Browse</td>	CTYPE plist	PUBLIC -	//Apple//	-8°?> DTD PL	IST 1.0//EN" "http://	www.apple.com/	DTDs/Prop	ertyList-1.0.0	dtd">			Browse
<plis< place<="" td=""><td>t version="1</td><td>.0"&gt;</td><td></td><td></td><td></td><td></td><td></td><td>-</td><td></td><td></td><td></td><td>Clear</td></plis<>	t version="1	.0">						-				Clear
				10								

## 16.3. Security Feature Support in iOS App

The iOS app supports a subset of the security features available in RAB. These features can be accessed through the **Security** tab within RAB. iOS only supports the default *Allow anyone to use this app* and the *Require each user to register with the details when they first use the app* features. The other features are available for Android apps but are not implemented in the iOS app at this time. The registration configuration for this feature works identically for both Android and iOS. No special steps have to be taken specifically for iOS to configure this option if it is selected.

	Fackage	Project	APK	IPA	Signing (Android)	Signing (iOS)	Expiry	Firebase	Deep Linking	Security	Permissions
Compres	sion and Enc	ryption									
By default	, App Builder	will compre	ess and	encrypt	t the data in the app.	This significantly	y reduces	app size. Tu	m compression o	ff if you wan	t to use the editor feature.
Co	mpress and e	encrypt data	a in the a	рр							
_											
Restricted	Users	v anvone to	uea thie	200.0	restrict its use to a	limited number of	f known d	avices The r	estricted option o	ould be used	ful if you are testing the app with a small g
of people	and want to n	nake sure it	t is not di	istribut	ed any further.	innited number o	i known d	evices. The h	estricted option c	Jouid De usei	ui ii you are testing the app with a small g
	ow anyone to	install and	use this	app							
_ •••											
	ow only selec	ted devices	s to use t	this app	P						
		Specify De	vices								
🔘 Re	quire a devic	e-specific a	access co	ode to	use this app						
		Configure A	ccess								
🔘 Re	quire each us	ser to registe	er with th	neir det	ails when they first u	ise the app					
	Co	onfigure Reg	sistration.	••							
🔿 Sh	ow a calculat	or on start-u	up. Enter	ing a c	ode will give access	to the app conte	nt.				
	0	Calculator S	ettinas								

## 16.4. Firebase Messaging

If Firebase Cloud Messaging is configured to allow push notifications for the app, several changes are required for the app's configuration information in the Apple App Store. The first thing that is required is that the App's configuration must be changed in Apple Developer to include Push Notifications. Under *Certficates, Identifiers, and Profiles*, select the app that is being configured. Enable the Push Notifications option from the list.

-		
U	NJ NFC Tag Reading	
D	VPN Personal VPN 🔘	
	Push Notifications ③	Configure Certificates (0)
0	Sign In with Apple	Configure
	🛞 sirikit 💿	
	System Extension	
	Time Sensitive Notifications	
D	S User Management	
D	Wallet 🕡	
	🗧 Wireless Accessory Configuration 💿	

After making these changes, the provisioning profile for the app should be updated for the new capability and the entry updated in SAB.

For Firebase to send notification to the app, it will need to have a key uploaded to it. The preferred manner is to go to the Keys section in Apple Developer and select to create an Apple Push Notifications service key, using the first option on the screen below. One key of this type is created to be used with all apps associated with this organization. If a key has already been created, then use that.

evelope	1 1	Summer Institute of L	inguistics, Inc (SIL) - 3YE4.
ertifi	cates, Identifiers &	Profiles	
< All Keys			
Regis	ter a New Key		Contin
Key Name	1		
You canno	t use special characters such as @, &, *, ', ", -, .		
ENABLE	NAME	DESCRIPTION	
	Apple Push Notifications service (APNs)	Establish connectivity between your notification server and the Apple Push Notification service. One key is used for all of your apps. Learn more	
	DeviceCheck	Access the DeviceCheck and AppAttest APIs to get data that your associated server can use in its business logic to protect your business while maintaining user privacy. Learn more	
	MapKit JS	Use Apple Maps on your websites. Show a map, display search results, provide directions, and more. Learn more O There are no identifiers available that can be associated with the key	Configure
	Media Services (MusicKit, ShazamKit)	Access the Apple Music catalog and make personalized requests for authorized users, and check audio signatures against the Shazam music catalog. O There are no identifiers available that can be associated with the key	Configure
	Sign in with Apple	Enable your apps to allow users to authenticate in your application with their Apple ID. Configuration is required to enable this feature. O There are no identifiers available that can be associated with the key	Configure
	ClassKit Catalog	Publish all of your ClassKit app activities to teachers creating Handouts in Apple	

#### C

At the time the key is created, a .p8 file is generated which can be used for all apps. You are allowed to make two keys through this interface, so that you can create a new key and revoke the old one, but only one is active at a time.

**Certificates, Identifiers & Profiles** 

< All Keys View Key Details		Download	Revoke	Edit
Name SIL APNS Key ID 55438BX8MT				
Enabled Services				
NAME	CONFIGURATION			
Apple Push Notifications service (APNs)				

Once the p8 file for this key has been downloaded, the file can be uploaded to the Firebase configuration for cloud messaging for this app.

Proje	ect setting	S Integrations	Service accounts Data	privacy Users and permissions	App Check (BETA)	
		Project creden	tials			
						Add server key
		Key	Token			
		Server key	AAAAVZW_lql:APA911 ZRdOygMPJY_mGjCg	bHGkdH5CPgl1liU5JEx-W-1PugA7_3y10JBj Wk5AoEEvM-9ubRFtTdQ5ucvQpNg0lfF9V2	AlHMzeynfJeLli2lldmakKnnzLm6 _01	BnjpXzHgkEN3GwGuQw
		Sender ID ②	)			
		367584581	282			
		iOS app config	juration			
		iOS apps		Firebase Cloud Messaging can	use either an APNs authentication	key or APNs certificate to connect
		iOS Engli org.w	i <b>sh Greek</b> rycliffe.app.englishgreek2	with APNs		to for the continuato to connect
				APNs Authentication Key		
				File	Key ID Team	ID
				APNs Auth Key	55438BX8MT	3YE4W86L3G Delete

## 16.5. Firebase Crashlytics for iOS

If Crashlytics is configured for the app in Firebase, the dSYM file associated with the app should be uploaded to get the detailed information about any crashes. To upload dSYMs, you'll need to use the *upload-symbols* command line tool that ships with the Crashlytics SDK. One way to obtain the tool is to install the Crashlytics SDK and locate the tool in the

"FirebaseCrashlytics" folder where you installed the SDK. The tool has also been uploaded to an App Builder share drive. The link to this file is:

https://drive.google.com/file/d/1RJIDZyQUbfhoqg9IGZg1YggNMJQQVeue/view?usp=sharing

After downloading this file, cd to the download location and enter "chmod +x uploadsymbols" to make the script executable. Otherwise a "Permission denied" may be encountered when attempting to run the script.

The dSYM file that is required is supplied as part of the build process if Crashlytics is configured for the application. The dSYM file is created in the ipa output folder with the same name as the ipa file, with a dSYM extension.

Upload the dSYM file using the *GoogleService-Info.plist* file associated with this application by opening a Terminal window. The command line to upload the dSYM file is:

/path/to/upload-symbols -gsp /path/to/GoogleService-Info.plist -p ios /path/to/TemplateApp.app.dSYM

For the example where all three files have been placed in the same directory and the Terminal window has cd'd to that directory, the command would be:

./upload-symbols -gsp ./GoogleService-Info.plist -p ios ./TemplateApp.app.dSYM