

Aaron Saunders

Use Appcelerator's free and open source Titanium mobile platform

Create native applications using JavaScript and the Titanium framework

Use a cloud service library of over 20 robust prebuilt common services to reduce integration time and risk

Create cross-platform native mobile applications from a single code base



Building Cross-Platform Apps using **Titanium™**, **Alloy**, and **Appcelerator® Cloud Services**

WILEY

Building Cross-Platform Apps Using Titanium™, Alloy, and Appcelerator® Cloud Services

Aaron Saunders

WILEY

This edition first published 2015

© 2015 Aaron Saunders

Registered office

John Wiley & Sons Ltd, The Atrium, Southern Gate,
Chichester, West Sussex, PO19 8SQ, United Kingdom

For details of our global editorial offices, for customer services and for information about how to apply for permission to reuse the copyright material in this book please see our website at www.wiley.com.

The right of the author to be identified as the author of this work has been asserted in accordance with the Copyright, Designs and Patents Act 1988.

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, except as permitted by the UK Copyright, Designs and Patents Act 1988, without the prior permission of the publisher.

Wiley also publishes its books in a variety of electronic formats. Some content that appears in print may not be available in electronic books.

Designations used by companies to distinguish their products are often claimed as trademarks. All brand names and product names used in this book are trade names, service marks, trademarks or registered trademarks of their respective owners. The publisher is not associated with any product or vendor mentioned in this book. This publication is designed to provide accurate and authoritative information in regard to the subject matter covered. It is sold on the understanding that the publisher is not engaged in rendering professional services. If professional advice or other

expert assistance is required, the services of a competent professional should be sought.

Trademarks: Wiley and the Wiley logo are trademarks or registered trademarks of John Wiley & Sons, Inc. and/or its affiliates in the United States and/or other countries, and may not be used without written permission. All other trademarks are the property of their respective owners. John Wiley & Sons, Ltd. is not associated with any product or vendor mentioned in the book.

A catalogue record for this book is available from the British Library.

ISBN 978-1-118-67325-6 (paperback); ISBN 978-1-118-67324-9 (ePub); 978-1-118-67322-5 (ePDF)

Set in 10/12.5 ChaparralPro-Light by TCS/SPS

Printed in the United States by Bind-Rite

I dedicate this book to my father, Dennis Francis Saunders Sr., who supported me in my interest with computers way before computers where commonplace in society. He bought me my first computer—a Timex Sinclair—and I also remember the TRS-80 from Radio Shack. He passed away before the book could be finished, but he is the reason I became involved with computers.

Publisher's Acknowledgements

Some of the people who helped bring this book to market include the following:

Editorial and Production

VP Consumer and Technology Publishing Director:

Michelle Leete

Associate Director-Book Content Management: Martin Tribe

Associate Publisher: Chris Webb

Project Editor: Kezia Endsley

Copy Editor: Kezia Endsley

Technical Editor: Chaim Krause

Editorial Manager: Rev Mengle

Senior Project Editor: Sara Shlaer

Editorial Assistant: Claire Johnson

Marketing

Marketing Manager: Lorna Mein

Assistant Marketing Manager: Dave Allen

About the Author

AARON SAUNDERS is the CEO/Founder of Clearly Innovative Inc., a minority-owned digital solutions provider headquartered in Washington DC with offices in New York City. The firm shapes ideas into viable products and transforms clients' existing technologies into stunning solutions. Clearly Innovative is a leader in early adaption and implementation of cutting edge technologies, from mobile strategy and design to developing innovative web-based solutions. Clearly Innovative provides support and expertise through services focused on product strategy, user experience, design, and development.

Aaron is an experienced software developer with over 30 years of experience and has strong technical, communication, and collaboration abilities. He is highly adept at helping organizations add business value using mobile and web applications.

Aaron has a BA in Computer Science from Ohio Wesleyan University and an MBA with concentrations in Information Technology Strategy and Marketing from the NYU Stern School of Business.

Acknowledgments

This book would never have been started without the encouragement of Kwasi Frye to keep pressuring me to respond to requests to write a book.

This book would have never been completed without the patience and understanding of my wife Andrea Saunders who consistently gave me the time I needed to get this done, which was above and beyond the long hours of running a small digital agency, when I got home nights and

sometime the whole weekend was spent writing code, reviewing chapters, and retesting the application for the book.

Thank you to Appcelerator for the platform you provided for me to start Clearly Innovative on, and thanks to all of the clients we worked with to develop mobile solutions and expand my knowledge of the Appcelerator platform and mobile application development.

Thanks to the team at Wiley who has been patient with me through the changes in the underlying Appcelerator platform that caused chapter rewrites, changes in the mobile user interface that required new screenshots, and delays in scheduling due to personal matters.

Thanks to Chaim Krause for being a great technical reviewer. I hope you learned something through the process.

Building Cross-Platform Apps Using Titanium™, Alloy, and Appcelerator® Cloud Services

Table of Contents

[Introduction](#)

[Chapter 1: Installing and Configuring Appcelerator](#)

[Setting Up Titanium](#)

[Installing Titanium on the Mac](#)

[Installing Titanium Studio IDE](#)

[Installing Xcode](#)

[Installing the iOS Simulator](#)

[Installing the Titanium Command-Line Interface to Use an Alternate IDE](#)

[Installing the Android SDK](#)

[Installing Titanium Studio on Windows](#)

[Installing Titanium Studio](#)

[Installing Android SDK](#)

[Summary](#)

[Chapter 2: Introducing Appcelerator Cloud Services](#)

[Using the Appcelerator Cloud Services Console](#)

[Using Appcelerator Cloud Services REST API](#)

[Installing curl on a Device](#)

[Simple Test with the REST API](#)

[Integrating Appcelerator Cloud Services](#)

[Simple Example of Integrating Appcelerator Cloud Services](#)

[Summary](#)

[Chapter 3: Appcelerator Titanium Alloy Overview](#)

[Understanding the Model-View-Controller \(MVC\) Framework](#)

[Using Appcelerator Alloy with the MVC Framework](#)

[Backbone.js](#)

[Backbone.js in Alloy: Models and Collections](#)

[Using Sync Adapters](#)

[Basic Sync Adapter Construction](#)

[Backbone Model Events](#)

[Model-View Data Binding](#)

[Demo Project for Model View Binding](#)

[Creating the Model File](#)

[Creating the Collection Object](#)

[Data Binding with Models in Appcelerator Titanium Alloy](#)

[Updating the cars.js Controller File](#)

[Creating the New Controller/View for the Detail Display](#)

[Completing the Controller for the Detail View](#)

[Creating Widgets](#)

[Creating a More Complex Widget](#)

[Summary](#)

[Chapter 4: Building a Cross-Platform Social Photo-Sharing Application](#)

[Using Balsamiq to Design Mockups](#)

[Walking Through the Phone-Sharing App](#)

[User Accounts](#)

[Camera](#)

[Photo Uploading](#)

[Social Integration with Facebook](#)

[Finding Friends](#)

[Commenting and Rating of Media](#)

[Push Notifications](#)

[Application Flow](#)

[Summary](#)

[Chapter 5: Development Process for Cross-Platform Apps](#)

[Creating the Project for This Chapter](#)

[Preconfiguring Appcelerator Cloud Services](#)

[Creating the User Interface](#)

[Creating the Tab Group Files](#)

[Enabling the Camera Functionality on the Feed Tab](#)

[Adding a Custom Table Row to TableView](#)

[Integrating the Camera Functionality into the Application](#)

[Accessing the Device Camera in Appcelerator](#)

[Adding Camera API Calls to Feed Controller](#)

[Revisiting the FeedRow Controller](#)

[Revisiting the Feed Controller to Add the Rows to the Table](#)

[Adding Some Style to the Feed Table](#)

[Using the Android ActionBar for the Camera Button](#)

[Setting Up the index.xml View to Support the ActionBar](#)

[Modifying the index.xml View to Support the ActionBar](#)

[Adding the Alloy Sync Adapter and Appcelerator Cloud Services](#)

[Creating the User Model](#)

[Extending Alloy Models](#)

[Logging the User In](#)

[Creating Appcelerator Cloud Service Sync Adapter](#)

[Creating the Photo Model](#)

[Modifying the ACS Sync Adapter to Support the Photo Model](#)

[Model and Sync Adapter Working Together](#)

[Summary](#)

[Chapter 6: Integrating Comments](#)

[Creating the Comment Table View Layout](#)

[Rendering the Rows Using a Different View and Controller](#)

[Styling the Views to Match the Mockups](#)

[Adding Logic to the Controllers](#)

[Calling the New Controller from feed.js](#)

[Coding the comment.js Controller](#)

[Cross-Platform Support in Comment View](#)

[Coding the commentRow Controller](#)

[Adding Models and Collections for Querying Comments](#)

[Finishing the Comment Controllers](#)

[The commentRow Controller](#)

[Connecting the Dots . . . Showing the Comment List](#)

[Back to the feed and feedRow Controllers](#)

[Adding a New Comment to a Photo](#)

[Creating a New Comment Controller and View](#)

[Adding Code to the Comment Input Controller](#)

[Back to the Comment.js Controller](#)

[Saving the Comment and Updating the Table](#)

[Deleting Comments](#)

[Summary](#)

Chapter 7: Integrating User Accounts with Appcelerator Cloud Services

[Adding the Login User Interface](#)

[Updating the User Model](#)

[User Create Account Method](#)

[User Logout Method](#)

[Additional User Management Methods](#)

[Updating the Index Controller](#)

[Set Up the Basics in the Index Controller](#)

[Creating the Login Controller](#)

[Logging in the User](#)

[Creating the User Account](#)

[Using Facebook for Account Creation](#)

[Setting Up an Application to Use the Facebook Module](#)

[Facebook Button in the login.xml File](#)

[Facebook Method in the User Model](#)

[Facebook Handler in Login Controller](#)

[Updating User with Facebook Information](#)

[Check for Facebook Authentication on Startup](#)

[Logging Out of Facebook](#)

[Summary](#)

Chapter 8: Working with Friends and Followers

[Creating the CommonJS Library in Alloy](#)

[Adding the Code](#)

[Adding the Friends User Interface](#)

[Finishing Up the ListView with Style](#)

[Introduction to Appcelerator Cloud Services Friends Object](#)

[Modifying the ACS Sync Adapter to Support User Queries](#)

[Modifying the ACS Sync Adapter to Support Friends](#)

[Creating the Friend Relationship](#)

[Finding Friend Relationships Based on a User's ID](#)

[Removing Friend Relationships from a User](#)

[Extending the User Model to Support User-Specific Friends Functionality](#)

[Integrating ListView Data-Binding with Friends Collections](#)

[Revisiting the friends.xml File](#)

[Integrating ListView Data-Binding with the Friends Controller](#)

[Displaying All Users](#)

[Displaying the Friends List](#)

[Working with User and Friends Lists](#)

[Removing a Friend from the Friends List](#)

[Updating the Application to Be Friend- and Location-Aware](#)

[Summary](#)

[Chapter 9: Working with Maps and Locations](#)

[Associating GPS Information When Saving a Photo](#)

[Modifying the Photo Model](#)

[Getting GPS Information from a Device](#)

[Creating a CommonJS Library for Geolocation](#)

[Updating the Feed Controller to Add Location to a Photo](#)

[Displaying the Photo Location on a Map](#)

[Android Support for Google Maps v2](#)

[Adding the Map Component to MapView XML](#)

[Displaying a Map of Photos Near Your Location](#)

[Querying ACS Photo Objects Using Your Current Location](#)

[Updating the User Interface to Show a Map View](#)

[Changes in the feed.js Controller](#)

[Responding to Clicks on Map Annotations](#)

[Summary](#)

[Chapter 10: Sharing via Facebook, Email, and Twitter](#)

[Creating the CommonJS Library for Sharing Functions](#)

[Facebook Permissions and Reauthorization](#)

[Sharing to the Facebook Wall](#)

[Sharing to the Facebook Album](#)

[Revisiting and Refactoring the Progress Window Library](#)

[Sharing to a Facebook Album](#)

[Sharing an Image as an Email Attachment](#)

[Twitter Integration with the social.js Module](#)

[Setting Up Your Twitter Developer Account](#)

[Adding social.js to Your Project](#)

[Adding the shareImage Function](#)

[Including the social.js Library in the Application](#)

[Adding Functionality to the sharing.js Library](#)

[Summary](#)

[Chapter 11: Push Notifications](#)

[Setting Up Push Notifications on Your Development Platform](#)

[Apple Push Notifications Configuration](#)

[Google Push Notifications Configuration](#)

[Configuring Push Notifications in Appcelerator Cloud Services](#)

[Creating the Push Notifications Library in an Application](#)

[Creating the pushNotifications.js Library](#)

[Getting the iOS Token](#)

[Getting the Android Token](#)

[Registering Callbacks](#)

[Integrating Push in Your Application](#)

[Registering for Push Notifications When the User Logs In](#)

[Sending Notifications Using the Appcelerator Cloud Services Console](#)

[Sending a Push Notification](#)

[Sending a Notification When Posting a Photo](#)

[Sending a Notification When Commenting on Photos](#)

[Sending a Notification When Adding a New Friend](#)

[Unregistering from Push Notifications When Logging Out](#)

[Further Integration of Push Notifications in Your Application](#)

[Summary](#)

[Chapter 12: Settings and User Management](#)

[Getting Started: View, Style, Controller](#)

[Editing the View](#)

[Editing the User Information in the Header Section](#)

[Editing the User Information Style](#)

[Handling Logout on Android and iOS](#)

[Logging the User Out](#)

[Logging Out of Appcelerator Push Notifications](#)

[Logging Out from Appcelerator Cloud Services](#)

[Logging Out from Social Media](#)

[Returning to the Login Screen](#)

[Setting the User's Profile Picture](#)

[Adding a Few Performance Enhancements](#)

[Returning to the Feed Controller for Performance and UI Enhancement](#)

[Additional Information from the User Account](#)

[Adding Content to the Main View in the Settings Tab](#)

[Platform-Specific User Interface for Switch Control](#)

[Handling the Switch Initialization Values](#)

[Displaying Push Notification Status](#)

[Changing the Push Notification Status](#)

[Summary](#)

Chapter 13: Going to Market: Deploying to the App Store and to Google Play

Process Overview

[Registering for a Developer Account](#)

[Signing Your Application](#)

[Creating an App Record and Filling Out Metadata](#)

[Publishing Your Binary to the Store](#)

iOS App Store Submission Process

[Signing Up for an iOS Developer Account](#)

[Signing Your iOS Application](#)

[Creating an iTunes Connect Record](#)

[Publishing from Titanium Studio](#)

Uploading Your Binary to the App Store

Google Play Submission Process

[Signing Up for a Google Play Developer Account](#)

[Generating a Keystore for Publishing](#)

Publishing to Google Play

Summary

Introduction

THIS BOOK IS a high-level overview of using Appcelerator Titanium Alloy and Appcelerator Cloud Services to build native, cross-platform solutions in JavaScript. There is and will probably for a long time be an argument about the benefits of cross-platform solutions like Appcelerator, yet competitors continue to enter the field.

First and foremost, Appcelerator Titanium Alloy is not Phonegap; Appcelerator renders native user interfaces and provides a robust and extensible framework of APIs to interact with the native device on iOS, Android, Blackberry, and Windows Phone. Phonegap uses the mobile device's web browser to render the user interface of the application and a collection of modules to interact with the device's native capabilities. It is an acceptable solution for some, but I choose to focus on Appcelerator in my business, Clearly Innovative, because I believed, from a business and cost perspective, that we could provide the client with the native performance at a much better price point.

This book walks you through the process of building a photo-sharing application for the iOS and Android platforms on the Appcelerator Platform using the powerful new Alloy framework. This application integrates Appcelerator Platform's MBaaS (mobile backend-as-a-service) and Appcelerator Cloud Services to create users, store photos, and implement push notifications in mobile applications.

The Appcelerator Community is very active through the Q&A forums, specific websites such as <http://www.tidev.io/>, and community-driven conferences like <http://ticonf.org/>. The Appcelerator company website documentation, training videos, and open

sourced samples can provide you with additional supporting materials to help you get started in building your own great solution.

Building a great mobile solution is a fun yet sometimes challenging experience. It is my hope that this book makes the process a bit more enjoyable and manageable.

—Aaron K. Saunders

@aaronksaunders

<https://github.com/aaronksaunders>

Chapter 1

Installing and Configuring Appcelerator

THIS CHAPTER BRIEFLY outlines the installation process for the Appcelerator tools. More detailed step-by-step instructions can be found on the Appcelerator site at http://docs.appcelerator.com/titanium/3.0/#%21/guide/Setting_up_Studio-section-37540095_SettingupStudio-installingstudio.

Setting Up Titanium

To install Titanium Studio, download the installer from the Appcelerator website. You will need to log in using your Appcelerator credentials, so register for an account if you haven't done so already.

After launching Titanium Studio, you will need to configure native SDKs for each platform you want to support. Android works on both Mac and PC, but to develop for iOS you will need a Mac. See

http://docs.appcelerator.com/titanium/3.0/#!/guide/Quick_Start for more information.

Installing Titanium on the Mac

To install Titanium Studio on the Mac, download Titanium Studio from Appcelerator and install it. Then install Xcode and the Android SDK. The following sections cover this process in detail.

Installing Titanium Studio IDE

1. Register for an account at www.appcelerator.com.
2. Download Titanium Studio at www.appcelerator.com/titanium/download, as shown in Figure 1-1. The download will begin automatically.

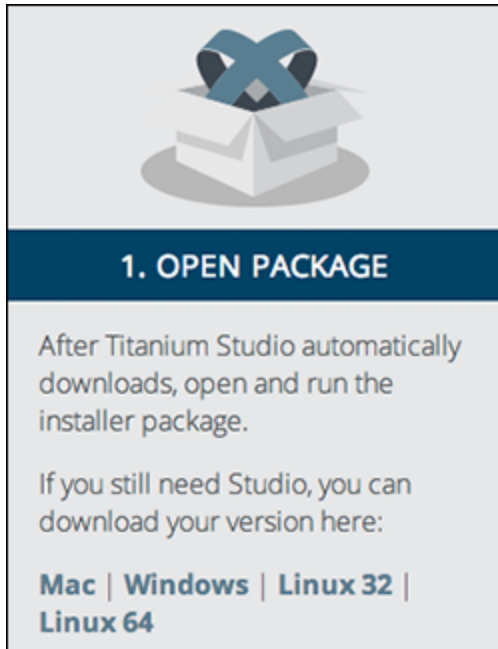


Figure 1-1: Downloading Titanium Studio from Appcelerator.

3. Open the disk image and drag the Titanium Studio folder into Applications, as shown in Figure 1-2.

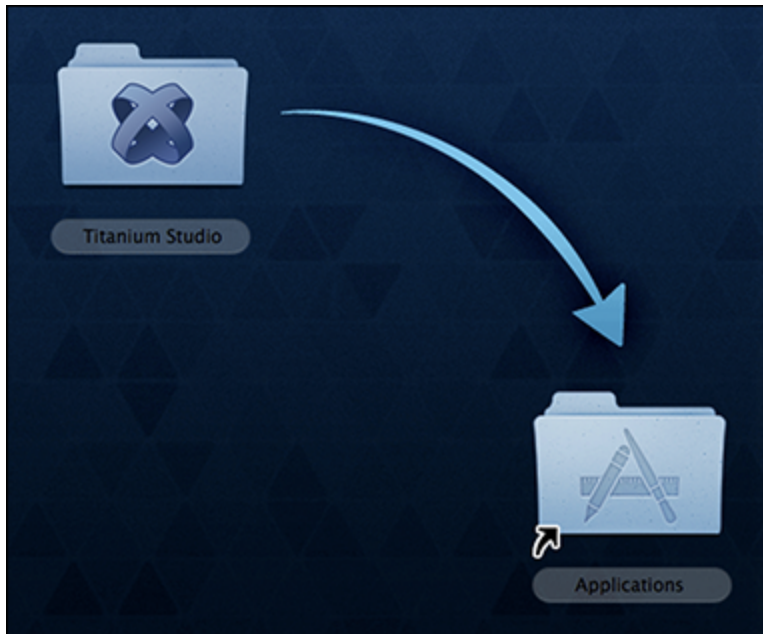


Figure 1-2: Dragging the Titanium Studio folder into the Applications folder.

4. Open Applications ⇒ Titanium Studio ⇒ TitaniumStudio, as shown in Figure [1-3](#).

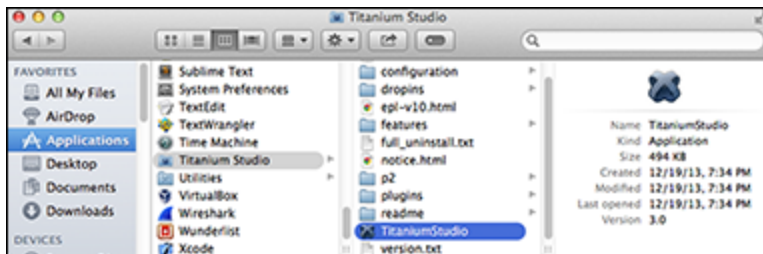


Figure 1-3: Launching Titanium Studio.

5. Check the box to use this folder as the default, and then click OK. See Figure [1-4](#).

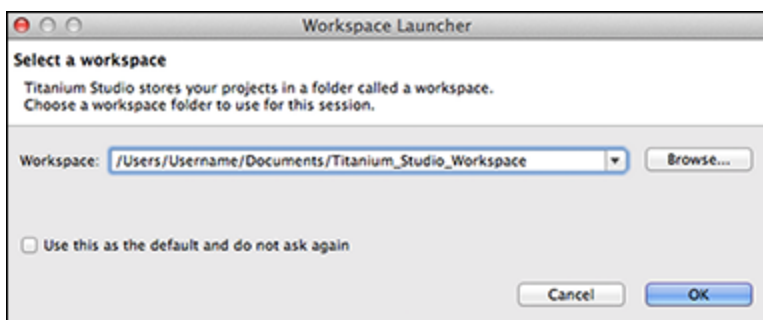


Figure 1-4: Selecting a workspace.

NOTE

After completing Step 5, your projects will be saved in Documents/Titanium_Studio_Workspace by default.

6. Log in using the account you created earlier. See [Figure 1-5](#).

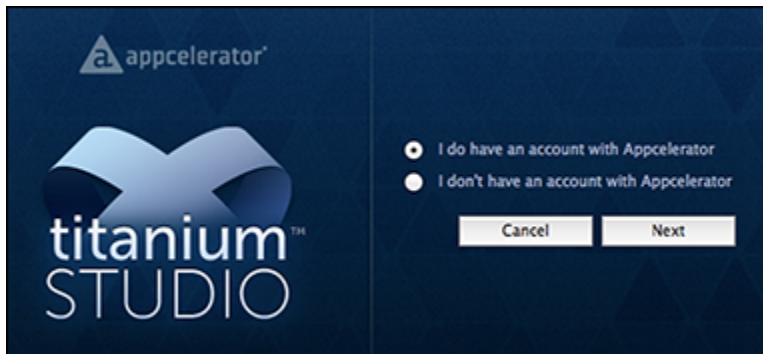


Figure 1-5: Logging in to use Titanium Studio.

Installing Xcode

Titanium Studio opens the Dashboard by default. You can reach the Dashboard again by clicking on the red home icon shown in [Figure 1-6](#).

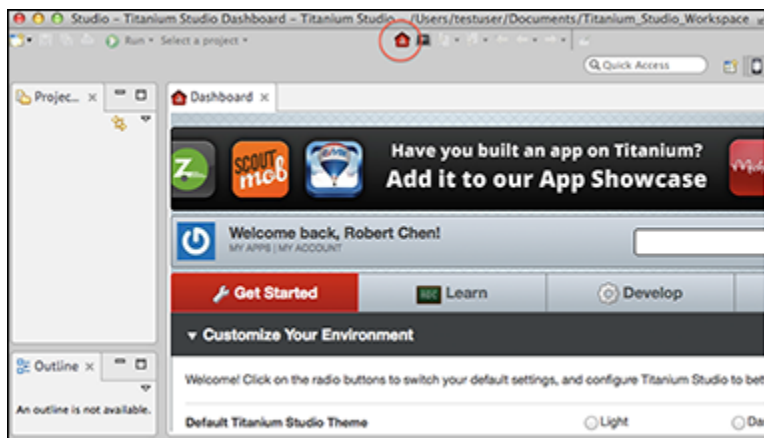


Figure 1-6: The Titanium Studio Dashboard is always accessible by clicking the red home icon.

1. Click the Get Started tab.

2. Scroll down to the Configure Native SDKs section and select iOS SDK. Click the Install or Update iOS SDK button on the left. See Figure [1-7](#).

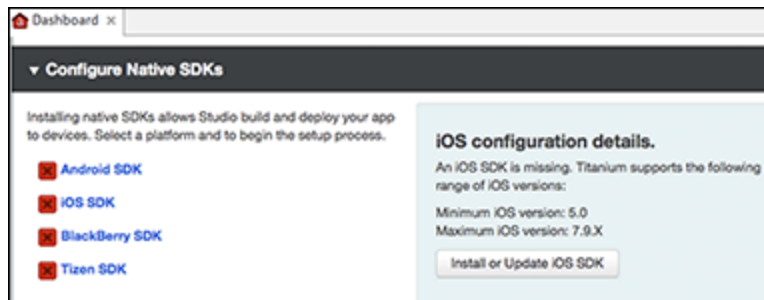


Figure 1-7: Selecting iOS SDK from the Configure Native SDKs section.

3. On the next window, click the Configure button. See Figure [1-8](#). This will launch the Mac App Store and take you to the Xcode download page, as shown in Figure [1-9](#).

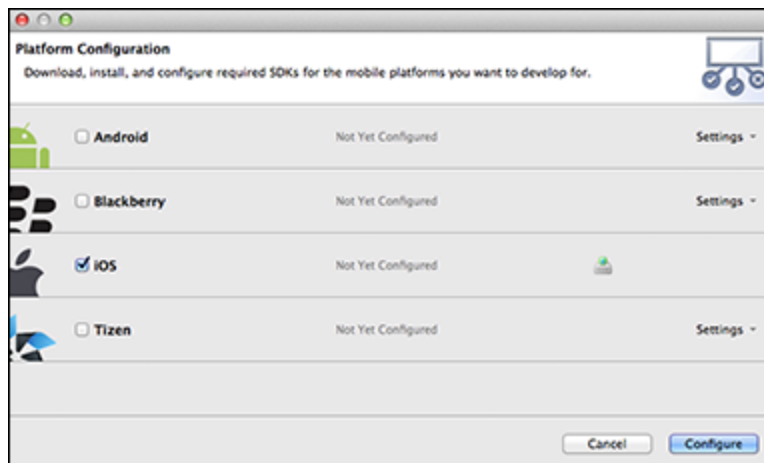


Figure 1-8: Platform Configuration window (iOS).

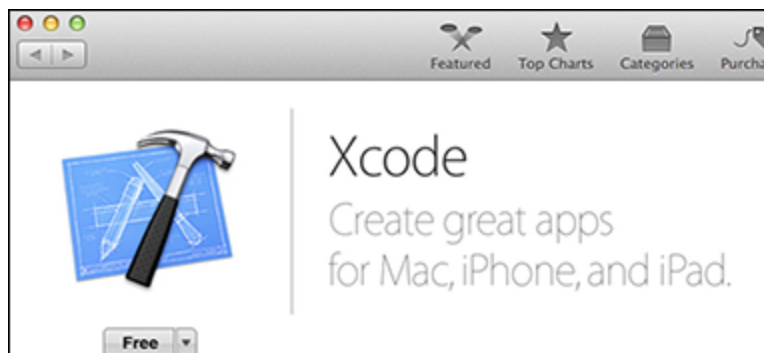


Figure 1-9: Installing Xcode from the Mac App store.

4. When it's finished, there should be a green checkmark next to iOS SDK, as shown in Figure [1-10](#).

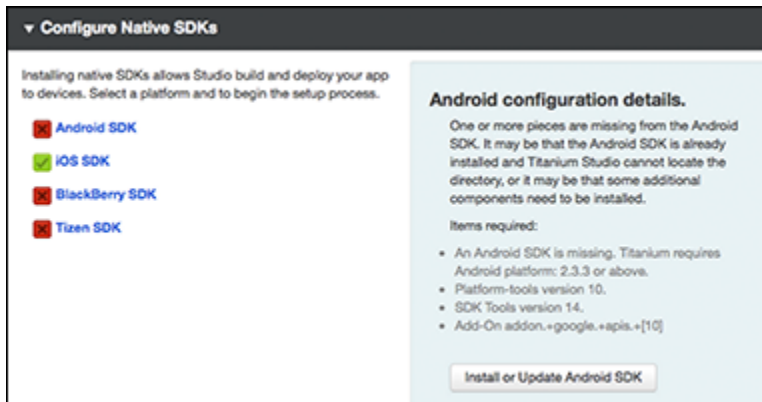


Figure 1-10: Configuring of the Native SDK section is complete.

5. Launch Xcode and accept the license agreement.

Installing the iOS Simulator

You will use the iOS Simulator regularly, so it's important to install it next. Open Xcode and navigate to Xcode ⇒ Preferences ⇒ Downloads. Select each available version of the iOS Simulator, as shown in Figure [1-11](#). Click the Check and Install Now button.

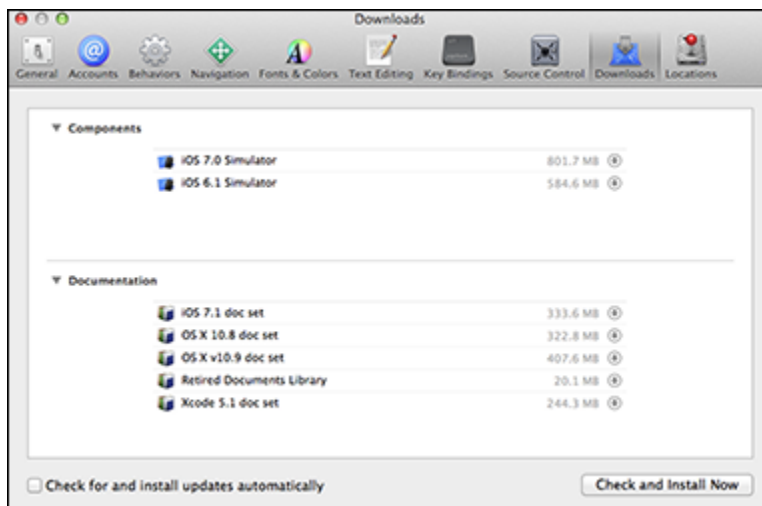


Figure 1-11: Downloading the iOS Simulator in the Xcode Preferences section.

Installing the Titanium Command-Line Interface to Use an Alternate IDE

If you choose not to use the Titanium Studio IDE, you will need to set up Titanium on the command line. To do so, open Terminal and run the following two commands.

NOTE

Node comes with Titanium Studio, so npm should work.

```
sudo npm install -g alloy
```

and

```
sudo npm install -g titanium
```

Installing the Android SDK

In the Titanium Studio Dashboard, select Android SDK and click Install or Update Android SDK. Then expand the Settings drop-down and select the Android API levels you want to support. Then click Configure. Note the Android SDK location: `/Users/<username>/Library/android-sdk-macosx/`. See Figure [1-12](#).



Figure 1-12: Installing Android SDKs in the platform configuration.

TIP

Library is a hidden folder, but you can reach it using Finder ⇒ Go and then holding down the Option key to reveal its location.

Installing Titanium Studio on Windows

To install Titanium Studio on Windows, download Titanium Studio from Appcelerator. Then install the Android SDK (Xcode requires a Mac, so you will not be able to deploy to iOS using Windows). The following sections cover this process in detail.

Installing Titanium Studio

Register for an account at www.appcelerator.com and then download Titanium Studio at www.appcelerator.com/titanium/download. See Figure [1-13](#).

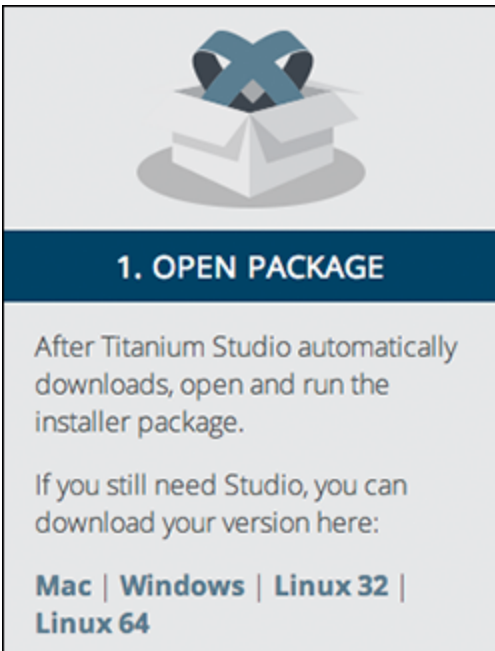


Figure 1-13: Downloading Titanium Studio from Appcelerator.

1. Launch the downloaded executable and accept all the defaults in the install wizard. Titanium will install the Java Development Environment, Git, and Node. See Figure [1-14](#).

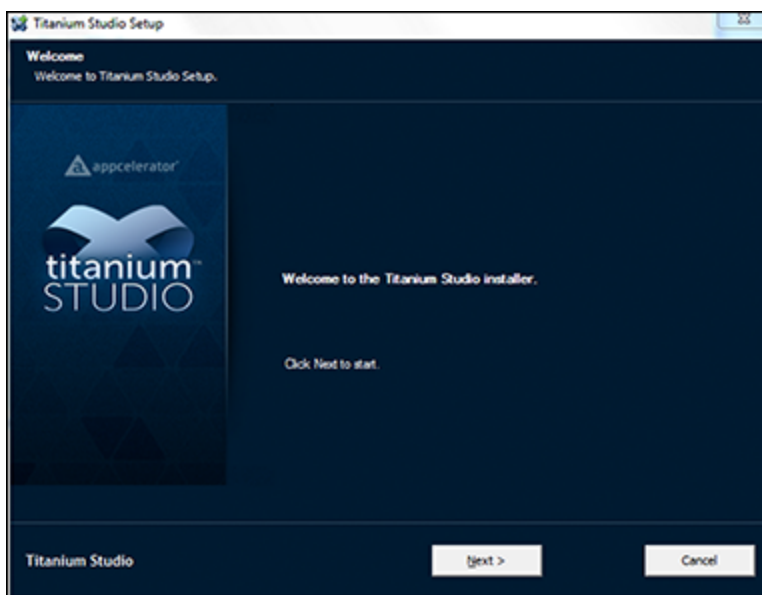


Figure 1-14: Titanium Studio Setup wizard.

2. After installation, launch Titanium by choosing Start ⇒ All Programs ⇒ Appcelerator ⇒ Titanium Studio. Check the box to accept the default folder location and click OK. See Figure [1-15](#).

NOTE

After completing Step 2, your projects will be saved in your Documents\Titanium_Studio_Workspace\ folder by default.

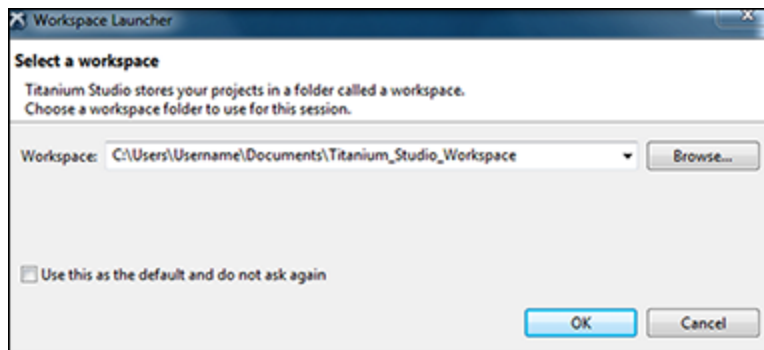


Figure 1-15: Selecting a workspace location.

3. Log in using the account you created earlier. See Figure [1-16](#).



Figure 1-16: Logging in to use Titanium Studio.

Installing Android SDK

Dashboard opens by default and you can come back to it later by clicking on the red home icon.

1. Scroll to the bottom of Dashboard to the Configure Native SDKs section.
2. Click Android SDK.
3. Click the Install or Update Android SDK button, as shown in Figure [1-17](#).

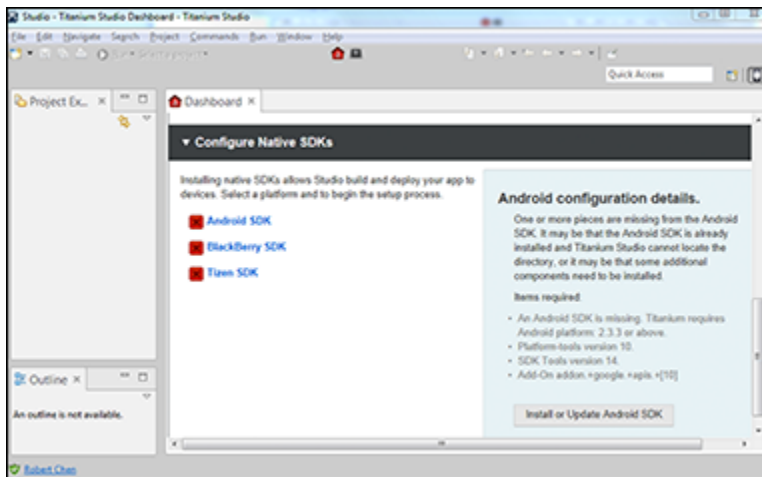


Figure 1-17: The Configure Native SDKs section.

4. Expand Settings and check the boxes for each Android API level you want to support.
5. Click the Configure button. See Figure [1-18](#).

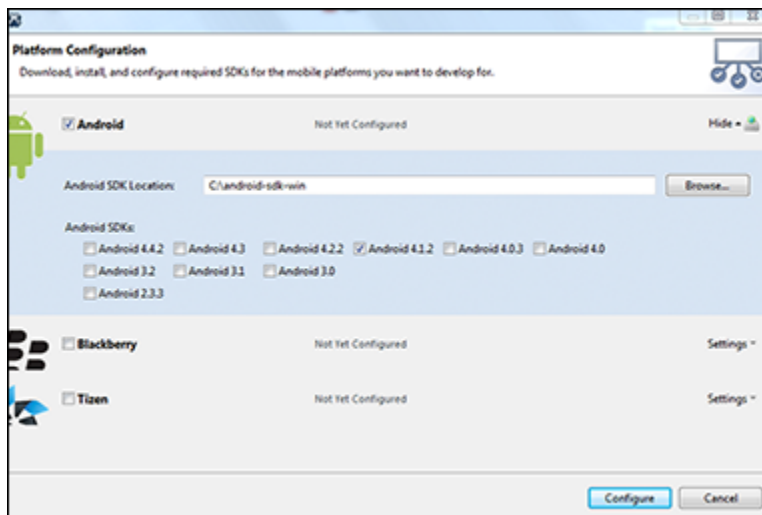


Figure 1-18: Installing Android SDKs in the platforms you want to support.

Note the Android SDK default location of `C:\android-sdk-win`. You may need to reopen Titanium Studio to refresh the Android SDK status. Look for the green checkmark.

Summary

To set up your environment, download and install Titanium Studio. On the Get Started tab in the Dashboard, you can configure native SDKs. The native SDKs enable you to deploy your app to platforms such as Android and iOS. The Android SDK link allows you to download and install different Android SDK versions. And on Mac, the iOS SDK link opens the Xcode download page on the Mac App Store. Use the Dashboard to verify whether each SDK was installed properly. When you're all set up properly, you're ready to move on to [Chapter 2](#), where you learn about all that Appcelerator Cloud Services has to offer.

Chapter 2

Introducing Appcelerator Cloud Services

ONE OF THE bigger challenges in building a complex mobile application comes from an unexpected source, building the supporting backend systems for the application. The majority of applications out there interact with web services or databases to save or retrieve information for presentation in a mobile application. As a mobile developer, you need to find a solution since your goal is to develop mobile solutions, not build and maintain backend services and perform IT management and support.

The traditional approach is to build this backend system, find a place to host it, and then provide the appropriate resources to support it. Taking that approach in the mobile solutions world is cost-prohibitive, is an ongoing maintenance challenge, and is a financial burden on the mobile solution even before the application is launched.

Appcelerator Cloud Services provides a complete framework for integrating the backend services into your mobile application. These services are hosted and maintained by Appcelerator, the APIs are tested and supported by Appcelerator, and handling of the appropriate scaling as needed is also their responsibility. These Appcelerator-provided services enable you to create rich immersive mobile applications. You can extend the application's services with the Node.ACS product and most importantly leverage the infrastructure for the backend provided by Appcelerator.