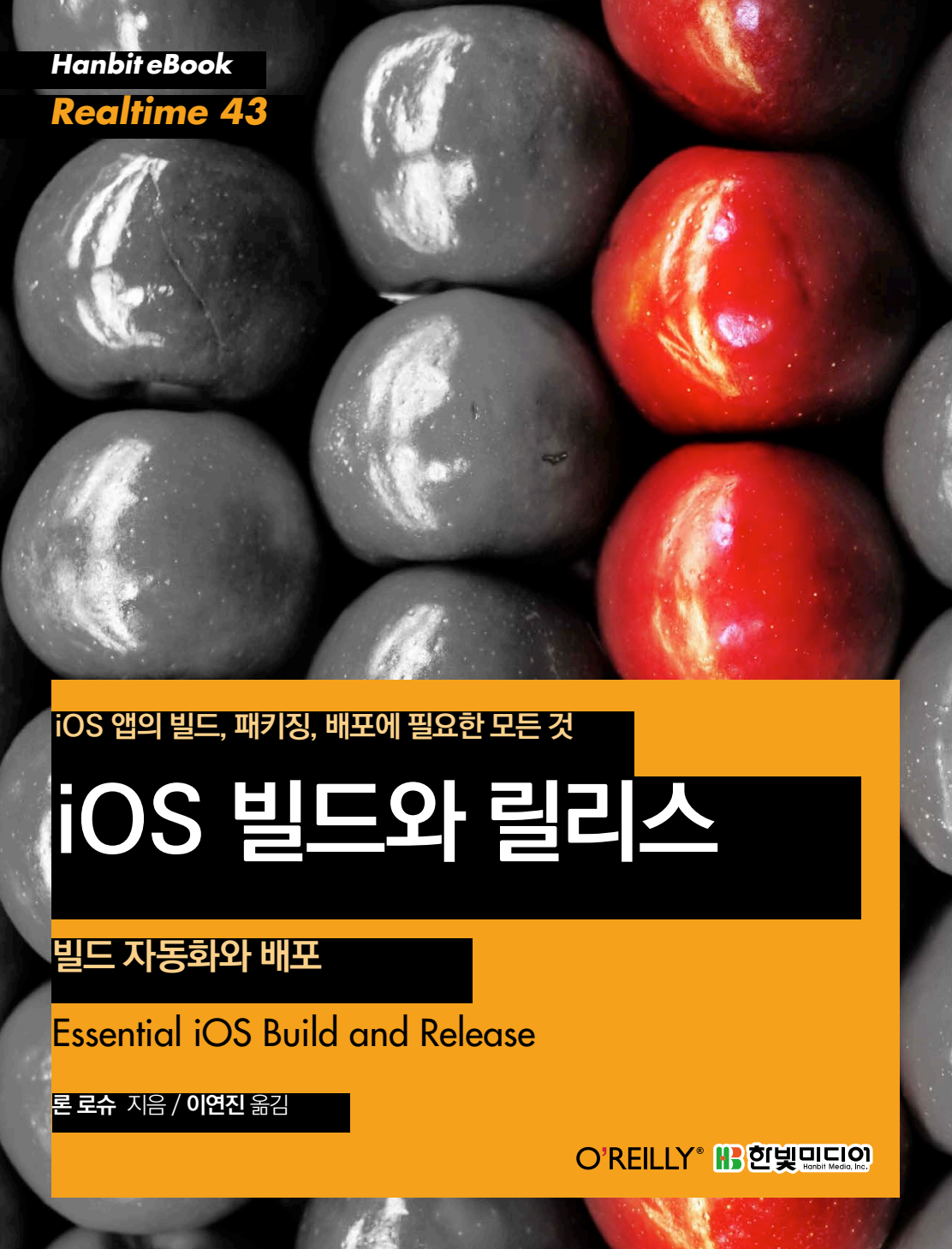


Hanbit eBook

Realtime 43

iOS 앱의 빌드, 패키징, 배포에 필요한 모든 것

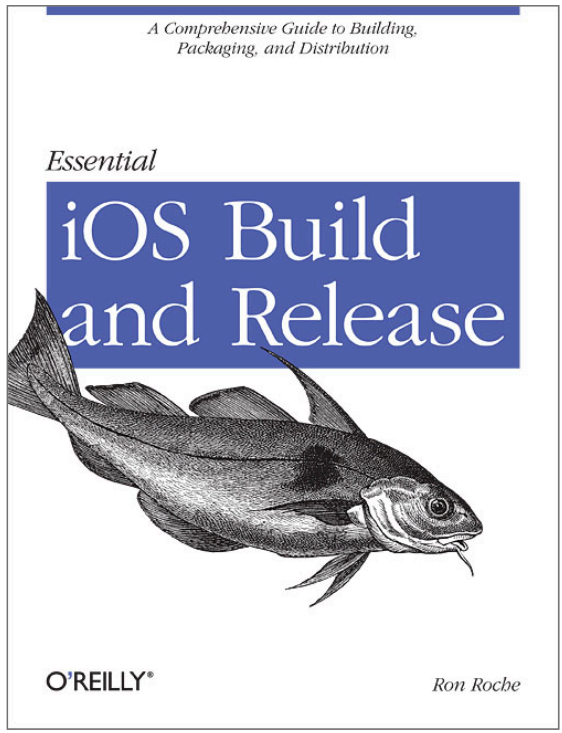
iOS 빌드와 릴리스

빌드 자동화와 배포

Essential iOS Build and Release

론 로슈 지음 / 이연진 옮김

O'REILLY® 한빛미디어 Hanbit Media, Inc.



이 도서는 O'REILLY의
Essential iOS Build and Release의
번역서입니다.

iOS 앱의 빌드, 패키징, 배포에 필요한 모든 것

iOS 빌드와 릴리스

빌드 자동화와 배포

iOS 앱의 빌드, 패키징, 배포에 필요한 모든 것 **iOS 빌드와 릴리스 – 빌드 자동화와 배포**

초판발행 2013년 10월 2일

지은이 론 로슈 / **옮긴이** 이연진 / **펴낸이** 김태헌
펴낸곳 한빛미디어(주) / **주소** 서울시 마포구 양화로 7길 83 한빛미디어(주) IT출판부
전화 02-325-5544 / **팩스** 02-336-7124
등록 1999년 6월 24일 제10-1779호
ISBN 978-89-6848-646-3 15000 / **정가** 9,900원

책임편집 배용석 / **기획** 이중민 / **편집** 김연숙
디자인 표지 여동일, 내지 스튜디오 [밈], 조판 양소회
마케팅 박상용, 박주훈

이 책에 대한 의견이나 오탈자 및 잘못된 내용에 대한 수정 정보는 한빛미디어(주)의 홈페이지나 아래 이메일로 알려주십시오.
한빛미디어 홈페이지 www.hanb.co.kr / **이메일** ask@hanb.co.kr

Published by HANBIT Media, Inc. Printed in Korea

지은이_ **론 로슈**

론 로슈Ron Roche는 실리콘 밸리에서 모바일 앱 빌드/릴리스 전문 엔지니어로 활동 중이다. 복잡한 빌드 프로세스 자동화와 이를 정리한 문서화 작업을 10년 넘게 해 온 전문가다.

옮긴이_ **이연진**

기계와 애플 제품을 좋아하고 영어와 IT에도 관심이 많아 최근 6년간 IT 분야의 하드웨어 및 소프트웨어 제품용 웹사이트 번역에 심혈을 기울였다. 이를 바탕으로 iOS, Mac OS, 관련 개발에도 관심을 두고 공부하고 있다.

저자 서문

필자가 이 책을 집필한 이유는 책상에 머리를 박고 밤늦게까지 작업하는 데도 Xcode는 왜 앱의 코드를 인증해주지 않는지, 왜 아이폰에 앱을 설치해주지 않는지 궁금해하는 사람을 위해서다. 이들은 어느날 갑자기 모든 문제가 해결되어 앱이 iOS 기기로 설치되는 기적을 바라면서, 다른 개발자의 지혜가 녹아있는 마법의 지푸라기를 찾아 웹사이트를 100군데도 넘게 돌아다녔을 것이다.

이 책에서는 iOS 빌드와 릴리스 과정에서 자주 부딪히게 되는 문제에 대한 해답이 될 수 있기를 바라면서, 빌드와 릴리스 과정의 미묘한 차이를 설명하려 했다. 독자 여러분이 개발자라면 이 책은 빌드와 릴리스 과정에 소요되는 시간을 절약하는 데 도움이 되어 개발에 더 많은 시간을 할애할 수 있을 것이다. 또한 회사에서 iOS 앱을 관리하는 릴리스 엔지니어라면 이 책에서 소개한 자료를 활용해 빌드 프로세스를 원활하게 하고, 다수의 iOS 기기를 관리하고, iOS 앱을 훨씬 수월하게 배포할 수 있을 것이다.

iOS 앱 개발을 해본 경험이 있거나 iOS 앱을 빌드해본 사람이라면 실제 iOS 기기에서 작동하는 앱의 빌드와 배포에 얼마나 많은 시간을 투자해야 하는지 잘 알 것이다. 특히 필자는 많은 iOS 앱의 빌드와 릴리스 프로세스를 관리해본 사람으로서 앱의 빌드와 설치 과정에 없어서는 안 될 분명하고 간결한 문서가 없을 경우 개발 전 과정에 얼마나 심각한 문제가 발생하는지 누구보다 잘 알고 있다. 실제로 이러한 빌드와 릴리스 프로세스 때문에 엄청난 시간을 쏟아부으면서 놀랐던 점은 많은 iOS 프로그래밍 도서가 하나같이 빌드와 릴리스 과정을 전혀 다루지 않거나 심지어는 피하기까지 한다는 사실이었다.

그런 의미에서 이 책은 iOS 개발 방법을 전혀 다루지 않는 유일한 iOS 개발 서적일 수도 있다. 즉, 이 책에서는 Objective-C는 단 한 줄도 다루지 않으며, 테스트용 iOS 기기와 앱 스토어에 배포된 앱에 대해 알고 있어야 할 사항을 다룬다.

모쪼록 이 책을 유용한 iOS 빌드 및 릴리스 지침서의 하나로 삼아 이 책을 읽는 모든 독자의 귀중한 시간이 절약될 수 있기를 바란다.

론 로슈Ron Roche

역자 서문

분명히 가이드를 따라 제대로 작업했는데 생각하지 못한 부분에서 오류가 발생해서 일이 진행되지 않을 때가 있다. 그래서 무엇이 잘못되었는지 샅샅이 살펴보고 관련 자료도 찾아보지만, 원하는 자료를 찾기 어려울 때가 많을 뿐만 아니라 어렵사리 찾은 자료를 살펴보면 화가 날 정도로 사소한 부분에서 실수했다는 걸 알게 되는 경우도 허다하다. '이것 하나 찾으려고 그 많은 시간을 허비했나' 싶어 허탈하기도 하고 화도 나지만 어쩌랴, 그래도 해결되어서 다행이지.

저자가 말했듯이 앱 개발도 중요하지만 개발 환경을 구축하는 절차와 주의점도 잘 파악해야 자칫 프로그래밍 작업 시간보다 더 많은 시간을 허비할 수도 있는 오류를 피할 수 있다. 애플 개발자 사이트부터 Xcode의 Organizer와 패스Passes에 이르기까지, 모두들 아는 사실임에도 언급하기는 꺼렸던 내용을 번역하는 동안 저자의 세심함과 개발자를 위한 배려에 감탄했다. 책 내용을 살펴보면 과정이 동일해도 설정하는 항목이 다르면 해당 과정을 다시 한 번 짚어주고, 필요한 부분은 그림으로 확인할 수 있게 해주었기 때문이다. 따라서 이 책을 참고하면서 iOS 앱을 개발한다면 기획 과정이나 프로그래밍에 좀 더 집중할 수 있을 뿐만 아니라 그 이외에 발생할 수 있는 오류를 해결하는 데 들이는 시간도 대폭 줄일 수 있을 것이다.

번역이 끝나고 원고가 다듬어지는 동안 애플 개발자 웹사이트가 일부 바뀌기도 하고 해킹 시도로 사이트에서 제공하는 서비스가 중단되었다가 다시 열리는 우여곡절을 겪기도 했다. 그러면서 원서와 달라진 부분까지도 역서에는 모두 반영하려고 노력했다. 원서와 다른 부분이 간혹 보일 텐데, 이는 이러한 애플 개발자 웹사이트의 변화에 따라 내용을 다듬었기 때문임을 미리 알리는 바다.

항상 느끼는 바지만 책은 혼자서 만들 수 없다. 역자가 번역에 중점을 두느라 자칫 소홀해질 수 있는 부분을 확인하는 기획편집자와 관련 부서의 여러 직원이 꼼꼼하게 검토해야 제대로 된 한 권의 책이 세상에 나올 수 있다. 이는 이번 전자책도 마찬가지라 생각한다. 다년간의 출판 경력과 애플에 대한 애정을 바탕으로 번역을 의뢰하고 원고를 다듬어준 이중민 대리에게 특히 감사드린다. 그리고 번역을 온전히 끝낼 수 있도록 옆에서 독려해준 남편에게도 고맙다는 말을 전하고 싶다.

번역을 마치며

역자 **이연진**

대상 독자와 테스트 환경

초급 **초중급** 중급 중고급 고급

Mac 컴퓨터

최소 OS X 10.7라이언 또는 OS X 10.8마운틴 라이언으로 구동되는 Mac이 필요하다. 그리고 이 책에서 참고하는 애플 개발자 웹사이트에 접속할 때는 Safari 웹 브라우저를 사용하자. 애플이 애플 개발자 웹사이트에 등록하는 콘텐츠는 다른 웹 브라우저에서는 정확하게 보이지 않을 수 있기 때문이다.

Xcode

Mac에 Xcode 5를 설치해야 한다. Xcode 5는 앱 스토어에서 무료로 다운로드하거나 iOS 개발자 센터 웹사이트에서 iOS 개발자 프로그램iOS Developer Program의 회원이 되면 다운로드할 수 있다.

iOS 기기

앱을 로딩할 iOS 기기가 최소한 한 대는 있어야 한다. 이 책을 집필하는 시점에서 해당되는 iOS 기기는 아이패드, 아이패드 미니, 아이폰, 아이팟 터치다.

iOS 개발자 프로그램 회원 가입

앱을 테스트용 기기에 설치하려면 iOS 개발자 프로그램 또는 iOS 개발자 기업 프로그램iOS Developer Enterprise Program에 등록해야 한다. iOS 개발자 프로그램은 Individual개인 또는 Company/Organization회사/조직 자격으로 가입할 수 있다.

이 책의 내용

이 책은 각종 빌드 방법과 배포 방법에 초점을 맞추었다. 그리고 앱을 개발할 때 당면하는 코드 서명 또는 빌드 오류의 원인을 일일이 다룰 수 없으므로 개발과 빌드 환경을 적절하게 설정해 오류 발생을 최소화하는 방향으로 안내하려고 했다. 또한 독자가 익숙해져야 하는 내용이라면 반복해서 언급했으므로 여러분은 필요한 부분만 정확하게 찾아서 보면 된다.

1장. 빌드와 릴리스

『iOS 빌드와 릴리스 - 환경 설정』에서 iOS 개발에 필요한 환경 설정을 다루었으므로 이제는 빌드를 해볼 차례다. 1장에서는 여러 가지 빌드 방법과 개발한 앱의 내부 및 App Store 릴리스를 자세히 알아본다.

2장. 빌드 자동화

회사나 조직에서 iOS 앱을 개발할 때 필요한 빌드 자동화에 초점을 맞춰야 하는 빌드 및 릴리스 엔지니어를 위한 내용이다.

3장. 패스

패스를 위한 개발 프레임워크 설정에 필요한 사항을 설명한다. 패스 타입 아이디 Pass Type ID와 인증서가 올바르게 설정되었는지 확인한 다음, 개발 환경에서 패스 서명 앱을 실행하는 데 필요한 정보를 확인할 수 있다.

참고로 이 책은 애플 개발자 사이트와 iOS 7에서 변경 사항이 있을 경우 원서 내용과 상관없이 바뀐 부분의 글과 그림을 수정했다. 원서와 내용이 크게 다른 부분은 역자주로 알릴 것이며, 기본적으로는 2013년 10월 애플 개발자 사이트, iOS 7, Xcode 5 기준으로 이 책을 읽는 것이 가장 좋음을 알린다.

감사의 말

이 책의 편집을 맡아주었을 뿐만 아니라 전문가로서 귀중한 의견을 아끼지 않고 제시해주면서 필자를 이끌어준 앤디 오람Andy Oram에게 특히 감사를 표한다. 또한 필자의 원고를 읽고 기술 검토를 맡아준 밴디드 나하밴디푸어Vandad Nahavandipoor에게도 진심으로 감사하다는 말을 전하고 싶다. 검토 과정을 통해 그가 전해준 의견, 제안, 서신이 필자에게 얼마나 도움이 되었는지 말로 다 할 수 없다. 이 프로젝트를 시작할 때 필자를 도와준 크리스 번Chirs Byrne에게도 감사를 표한다.

끝으로 아내 패트리샤Patricia에게 가장 깊은 감사를 표하고 싶다. 항상 필자를 격려해주고, 시간을 들여 이 책을 쓸 수 있게 배려해주었으며, 원고의 초안을 수없이 검토해주었다. 아내의 통찰력과 사려 깊은 조언 덕분에 이 책을 훨씬 완성도 있게 만들 수 있었다.

한빛 eBook 리얼타임

한빛 eBook 리얼타임은 IT 개발자를 위한 eBook입니다.

요즘 IT 업계에는 하루가 멀다 하고 수많은 기술이 나타나고 사라져 갑니다. 인터넷을 아무리 뒤져도 조금이나마 정리된 정보를 찾는 것도 쉽지 않습니다. 또한 잘 정리되어 책으로 나오기까지는 오랜 시간이 걸립니다. 어떻게 하면 조금이라도 더 유용한 정보를 빠르게 얻을 수 있을까요? 어떻게 하면 남보다 조금 더 빨리 경험하고 습득한 지식을 공유하고 발전시켜 나갈 수 있을까요? 세상에는 수많은 종이책이 있습니다. 그리고 그 종이책을 그대로 옮긴 전자책도 많습니다. 전자책에는 전자책에 적합한 콘텐츠와 전자책의 특성을 살린 형식이 있다고 생각합니다.

한빛이 지금 생각하고 추구하는, 개발자를 위한 리얼타임 전자책은 이렇습니다.

1. **eBook Only - 빠르게 변화하는 IT 기술에 대해 핵심적인 정보를 신속하게 제공합니다.**

   500페이지 가까운 분량의 잘 정리된 도서(종이책)가 아니라, 핵심적인 내용을 빠르게 전달하기 위해 조금은 거칠지만 100페이지 내외의 전자책 전용으로 개발한 서비스입니다. 독자에게는 새로운 정보를 빨리 얻을 수 있는 기회가 되고, 자신이 먼저 경험한 지식과 정보를 책으로 펴내고 싶지만 너무 바빠서 엄두를 못 내는 선배, 전문가, 고수 분에게는 보다 쉽게 집필할 수 있는 기회가 될 수 있으리라 생각합니다. 또한 새로운 정보와 지식을 빠르게 전달하기 위해 O'Reilly의 전자책 번역 서비스도 하고 있습니다.

2. **무료로 업데이트되는, 전자책 전용 서비스입니다.**

   종이책으로는 기술의 변화 속도를 따라잡기가 쉽지 않습니다. 책이 일정 분량 이상으로 집필되고 정리되어 나오는 동안 기술은 이미 변해 있습니다. 전자책으로 출간된 이후에도 버전 업을 통해 중요한 기술적 변화가 있거나 저자(역자)와 독자가 소통하면서 보완하여 발전된 노하우가 정리되면 구매하신 분께 무료로 업데이트해 드립니다.

3. 독자의 편의를 위하여 DRM-Free로 제공합니다.

구매한 전자책을 다양한 IT 기기에서 자유롭게 활용할 수 있도록 DRM-Free PDF 포맷으로 제공합니다. 이는 독자 여러분과 한빛이 생각하고 추구하는 전자책을 만들어 나가기 위해 독자 여러분이 언제 어디서 어떤 기기를 사용하더라도 편리하게 전자책을 볼 수 있도록 하기 위함입니다.

4. 전자책 환경을 고려한 최적의 형태와 디자인에 담고자 노력했습니다.

종이책을 그대로 옮겨 놓아 가독성이 떨어지고 읽기 힘든 전자책이 아니라, 전자책의 환경에 가능한 한 최적화하여 쾌적한 경험을 드리고자 합니다. 링크 등의 기능을 적극적으로 이용할 수 있음은 물론이고 글자 크기나 행간, 여백 등을 전자책에 가장 최적화된 형태로 새롭게 디자인하였습니다.

앞으로도 독자 여러분의 충고에 귀 기울이며 지속해서 발전시켜 나가도록 하겠습니다.

차례

1 | 빌드와 릴리스

1장에서는 개발한 앱을 배포하기 위한 빌드와 준비 방법을 다룬다. 『iOS 빌드와 릴리스 - 환경 설정』을 읽었다면 앱 아이디App ID, 인증서Certificates, 프로비저닝 프로파일Provisioning Profiles을 만들어보았을 것이다. 이제 이를 종합해 빌드해보자.

먼저 특정 빌드 상황에 한정되어 있는 Xcode의 설정 몇 가지를 알아본 다음, iOS Simulator를 이용한 빌드와 Ad Hoc 및 앱 스토어 배포를 위한 빌드 같은 일반적인 빌드 시나리오를 살펴본다. 마지막으로 아이튠즈 커넥트iTunes Connect 웹사이트에서 앱 '레코드record'[01]를 설정하고 이 앱 레코드로 앱을 업로드해 앱 스토어에 등록할 수 있도록 안내해줄 것이다.

1.1 Xcode 빌드 설정

Xcode는 개발하는 앱에 필요한 사용자 환경을 설정하거나 추가할 수 있는 다양한 옵션을 제공한다. 이 책에서 모든 옵션을 다루기는 불가능하므로, 필자와 여러분의 주된 관심사인 앱 개발 및 배포 준비와 관련 있는 가장 일반적인 설정만 살펴본다. 다양한 사용자 환경을 필요에 맞게 설정하려면 Xcode Overview의 'Run Your App'를 참고하자.[02]

1.1.1 Xcode 설정 이해하기

Xcode를 실행한 다음 프로젝트 에디터를 살펴보면 프로젝트 레벨(프로젝트 에디터의 PROJECT 부분)과 타깃 레벨(프로젝트 에디터의 TARGETS 부분) 설정 구성이 매우 다양하다는 점을 알 수 있다.

---

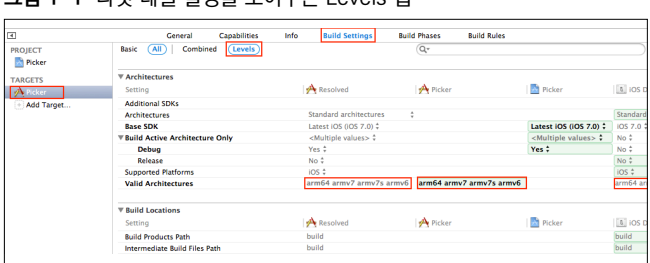
01 역자주_개발하는 앱에 대한 기록이다.

02 역자주_원서에는 Xcode User Guide의 'Build and Run Your App'이라고 소개하였으나 iOS 7의 정식 발표와 함께 바뀌었다.

그럼 다음 사항을 참고해 빌드 과정에서 근본적으로 어떤 Xcode 설정을 사용해야 할지 알아두자.

- 프로젝트 레벨에서 설정한 사항은 전역 설정이며 모든 타깃 레벨 설정에 적용된다. 그리고 타깃 레벨의 설정은 프로젝트 레벨에서 설정한 사항보다 '더 낮은 레벨'의 설정으로 간주되므로 해당 타깃에만 적용할 수 있다. 단, 해당 프로젝트의 타깃 레벨 설정은 프로젝트 레벨 설정보다 중요하다.
- 명령 줄에서 빌드(xcodebuild를 이용해)할 때 전달한 혹은 사용한 매개변수는 모두 타깃 레벨 설정보다 우선한다.
- 'Build Settings' 탭 아래의 'Levels'를 누르면 환경 설정 항목 각각에 'resolved 결정된'라는 설정이 표시된다. 해당 설정을 결정할 때 사용한 옵션 항목은 왼쪽에서 오른쪽으로 우선순위를 가지며 회색 글씨로 표시된다. 설정 사항을 결정한 타깃 레벨은 녹색으로 표시된다.
- 수정한 설정 사항은 프로젝트 에디터에 진한 글씨로 표시된다. 그리고 기본 설정으로 되돌려도 변경되었음을 알려주려고 계속 진한 글씨로 표시된다. 이 프로젝트를 위한 'Valid Architectures유효 아키텍처'의 기본 설정은 'arm64 armv7 armv7s'이다. 또한 armv6 아키텍처를 타깃 레벨에 추가했으므로 Resolved 설정 사항은 그림 1-1처럼 'arm64 armv7 armv7s armv6'으로 대체한다는 점을 알 수 있다.

그림 1-1 타깃 레벨 설정을 보여주는 Levels 탭



1.1.2 앱 아이디 설정하기

이번에는 Certificates, Identifiers & Profiles에서 만든 앱 아이디로 Xcode 프로젝트의 'Bundle identifier번들 식별자' 값을 정확하게 설정하는 방법을 자세히 설명한다. 빌드한 앱을 배포하기 전에는 반드시 프로비저닝 프로파일을 이용한 개발 또는 배포 인증서로 서명해야 한다. 또한 여러분도 알다시피 (개발 또는 배포를 위한) 프로비저닝 프로파일은 특정 앱 아이디와 연계해 만들어진다. 따라서 앱을 서명하려면 Xcode의 앱 타깃 설정에 앱 아이디의 번들 아이디 부분을 정확하게 입력해야 한다. 즉, Xcode 프로젝트에 원하는 프로비저닝 프로파일을 지정해 개발하는 앱을 서명하게 하여 동작한다는 뜻이다.

iOS 개발 센터iOS Dev Center에 로그인한 다음 Certificates, Identifiers & Profiles로 이동하자. 그리고 개발 중인 앱에 연결된 개발 및 배포 프로비저닝 프로파일을 만들기 위해 이 프로파일이 사용하는 앱 아이디를 기록해두자.

개발과 배포를 위한 앱 아이디는 반드시 같은 아이디를 사용해야 한다. 따라서 iOS Apps → Identifiers → App IDs에서 앱 아이디의 번들 아이디(메뉴의 ID) 부분을 기록해둔다. 이 번들 아이디를 Xcode에 입력할 것이다. 번들 아이디가 와일드카드 문자(*)로 끝난다면 이 문자를 원하는 문자열로 대체해야 한다. 표 1-1을 참고해 앱 아이디의 번들 아이디 부분을 결정하자. 간단히 말하면 번들 시드 아이디Bundle Seed ID(표 1-1 첫 번째 줄의 3H569L9349)의 오른쪽 소수점 뒤에 오는 모든 문자열(com.acme.FinancialPlanner)은 Xcode의 'Bundle identifier' 값을 설정하는 데 사용한다.

표 1-1 Xcode용 'Bundle identifier' 설정

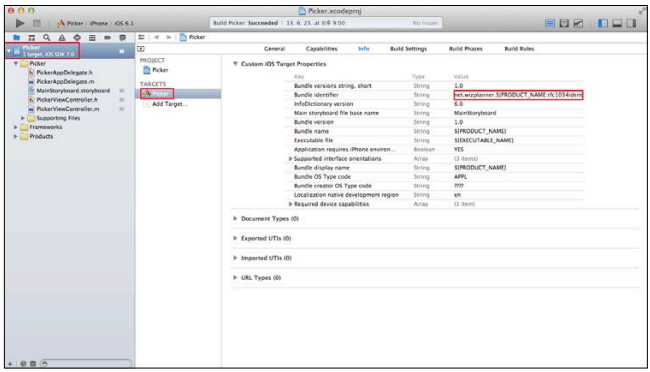
앱 아이디 예	번들 아이디 부분	Xcode용 번들 아이디 설정
3H569L9349.com.acme.FinancialPlanner	com.acme.FinancialPlanner	com.acme.FinancialPlanner
7L2O9A2384.pokertournament	pokertournament	pokertournament
4B587C2146.*	*	pokertournament
AW4MH6TPFX.com.acme.*	com.acme.*	com.acme.mortgageCalc
PSQV8VS4PW.com.tomdeveloper.*	com.tomdeveloper.*	com.tomdeveloper.fastcars

번들 아이디 설정하기

다음 단계를 따라 개발하는 앱에 사용할 'Bundle identifier'의 값을 Xcode에 설정한다(그림 1-2 참고).

1. Xcode의 메뉴에서 [View] → [Navigators] → [Show Project Navigator]를 선택한다.
2. 프로젝트의 루트 폴더를 선택한다.
3. 프로젝트 에디터의 TARGETS 섹션에서 대상(현재 실행한 앱 프로젝트)을 선택한다.
4. 'Info' 탭을 누른다.
5. Xcode는 기본적으로 번들 식별자를 com.appname.${PRODUCT_NAME:-rfc1034identifier}와 같은 형태로 설정한다. 따라서 Xcode의 'Bundle identifier' 설정을 사용하려는 번들 아이디와 일치하도록 변경한다. 이 설정은 대소문자가 구별된다는 점에 주의하고, 번들 식별자 다음에 빈칸이 추가되지 않도록 조심한다.

그림 1-2 앱에서 사용할 번들 식별자 설정



번들 아이디 확인하기

다음 단계를 따라 앱의 타깃 레벨에 설정한 번들 식별자가 프로비저닝 프로파일과 정확하게 연결되어 있는지 확인한다(그림 1-3 참고).

1. TARGETS 아래에 있는 해당 프로젝트 이름을 선택한 다음 'Build Settings' 탭을 누른다.
2. 탭 아래 설정이 'All'과 'Combined'로 선택되어 있는지 확인한다.
3. Code Signing 아래 Code Signing Identity 왼쪽의 회색 삼각형(▶)을 눌러 모든 빌드 설정 사항이 나타나게 한다. 번들 식별자가 프로비저닝 프로파일에 연결된 앱 아이디와 일치한다면 각각의 프로비저닝 프로파일에 대한 'Any iOS SDK'라는 항목을 설정할 수 있다. 기본적으로 Xcode는 개발 프로비저닝 프로파일에 대해 기본 Debug와 Release디버그 및 릴리스 빌드 환경 설정 모두에 'Any iOS SDK'를 설정한다.

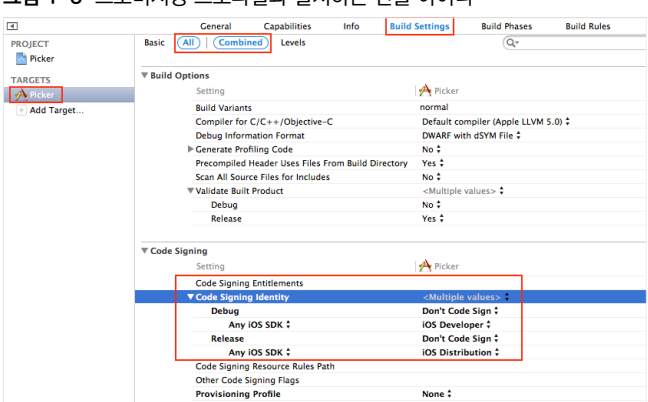
    - Debug 빌드 환경 설정의 Any iOS SDK 설정 값은 'iPhone Developer(현재 'iPhone Developer: Development_Certificate_Name in Developm-ent_Provisioning_Profile_Name'와 일치)'로 설정된다. 이

설정은 앱에 지정한 'Bundle identifier' 값이 설치된 개발 프로비저닝 프로파일과 일치한다는 점을 보여준다.

- Code Signing Identity → Release → Any iOS SDK 설정에서 이를 Ad Hoc이나 앱 스토어 배포 프로비저닝 프로파일로 변경하면 해당 Relase 구성에 대한 배포 프로비저닝 프로파일이 앱의 번들 식별자와 연계해서 동작하는지 확인할 수 있다. 이 설정은 'iPhone Distribution: Distribution_Certificate_Name' 형식으로 되어있다. 또한 Automatic에서 iPhone Distribution 프로파일을 선택해도 배포 프로비저닝 프로파일을 선택하고 해당 설정 사항을 iPhone Distribution(현재 'iPhone Distribution...'과 일치)으로 변경할 수 있다.

NOTE_Xcode 프로젝트가 열려있을 때 프로비저닝 프로파일을 설치하면 Xcode 프로젝트 에디터는 새로운 프로파일을 보여주지 않지만 오거나이저는 보여준다는 점을 알 수 있다. Xcode 프로젝트 에디터에서도 프로비저닝 프로파일을 확인하려면 PROJECT와 TARGETS를 여러 번 누르거나 Xcode를 다시 시작해야 한다.

그림 1-3 프로비저닝 프로파일과 일치하는 번들 아이디



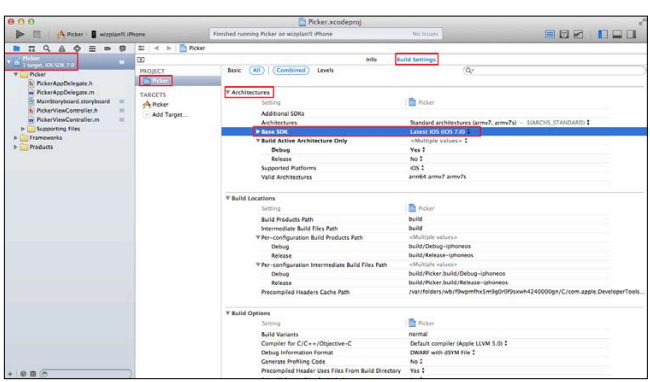
1.1.3 Base SDK 설정하기

Xcode에서는 앱을 컴파일할 때 사용할 SDK 버전을 설정할 수 있다. 최적의 성능을 발휘하도록 개발하려면 Latest최신 SDK 설정을 사용하는 것이 가장 좋은 방법이다. Latest SDK에는 가장 최근에 업데이트한 라이브러리, 프레임워크, 통합 컴파일러 등이 포함되어 있기 때문이다. 따라서 지금부터 SDK 설정을 앱의 프로젝트 레벨과 타깃 레벨에 적용했는지 확인해보자.

프로젝트 레벨에서 Base SDK를 설정하는 방법은 다음과 같다(그림 1-4 참고).

1. Xcode의 메뉴에서 [View] → [Navigators] → [Show Project Navigator]를 선택한다.
2. 프로젝트의 루트 폴더를 선택한다.
3. 프로젝트 에디터에서 PROJECT 아래의 해당 프로젝트 이름를 선택한다.
4. 'Build Settings' 탭을 누른 다음(아래에 'All'과 'Combined' 탭을 선택했는지 확인) Architectures에서 Base SDK를 'Latest iOS 버전'으로 설정한다.

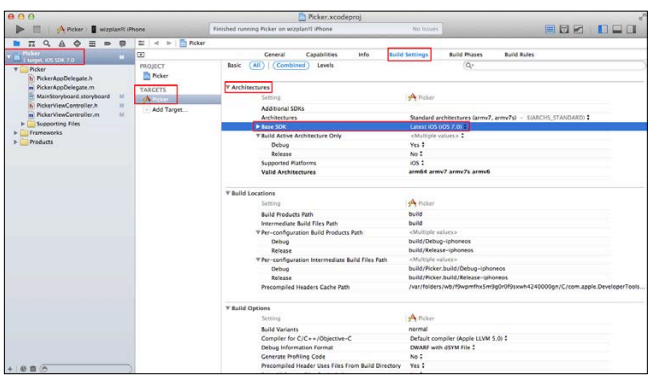
그림 1-4 프로젝트 레벨에서의 Base SDK 설정



타깃 레벨에서 Base SDK를 설정하는 방법은 다음과 같다(그림 1-5 참고).

1. Xcode의 메뉴에서 [View] → [Navigators] → [Show Project Navigator]를 선택한다.
2. 프로젝트의 루트 폴더를 선택한다.
3. 프로젝트 에디터에서 TARGETS 아래의 해당 프로젝트 이름을 선택한다.
4. 'Build Settings' 탭을 누른 다음(아래에 'All'과 'Combined' 탭을 선택했는지 확인) Architectures에서 Base SDK를 'Latest iOS 버전'으로 설정한다.

그림 1-5 타깃 레벨에서의 Base SDK 설정



1.1.4 배치 타깃 설정하기

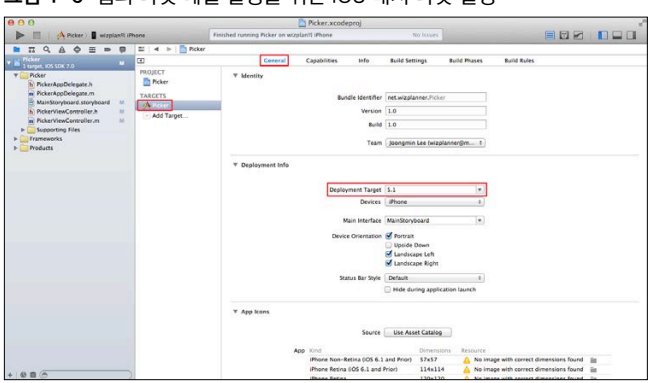
개발 중 어느 시점이 되면 개발하는 앱을 실행할 기기의 가장 낮은 iOS 버전을 결정해야 한다. 이때 보통은 개발하는 앱이 사용하는 API에 따라 버전을 결정하게 된다. 그보다 더 낮은 버전의 iOS 또는 개발하는 앱이 요구하는 하드웨어 요구 사항을 맞출 수 없는 낮은 버전의 iOS로 구동되는 기기에서는 앱을 사용할 수 없기 때문이다. 배치 타깃Deployment Target 설정은 앱 스토어 고객들이 앱을 사용하는 데 필요한 최소한의 iOS 버전을 반영한다. 예를 들어 배치 타깃 설정을 5.1로 지정하면 앱이 앱 스토어에서 'iOS 5.1 또는 그 이상'으로 표시된다. 사용자가 5.1보다 낮은

버전으로 구동되는 iOS를 사용하는 기기에 여러분의 앱을 설치하려고 하면 '애플리케이션이 호환되지 않음'이라는 오류 메시지가 나타난다. 즉, 개발하는 앱의 배치 타깃에서 설정하는 iOS 버전을 높게 지정할수록 앱을 사용할 수 있는 고객의 수는 줄어든다는 점을 명심해야 한다. 더 낮은 버전의 iOS를 사용하는(혹은 더 이상 높은 버전의 iOS를 사용할 수 없는 기기를 보유한) 고객들은 이 앱을 사용할 수 없기 때문이다.

다음 단계를 따라 앱의 타깃 레벨 설정을 위한 iOS 배치 타깃을 설정한다(그림 1-6 참고).

1. Xcode의 메뉴에서 [View] → [Navigators] → [Show Project Navigator]를 선택한다.
2. 프로젝트의 루트 폴더를 선택한다.
3. 프로젝트 에디터에서 TARGETS 아래의 해당 프로젝트 이름을 선택한다.
4. 'General일반' 탭을 누르고 아래 Deployment Info의 Deployment Target에서 드롭다운 목록을 눌러 원하는 iOS 버전을 설정한다.

그림 1-6 앱의 타깃 레벨 설정을 위한 iOS 배치 타깃 설정



1.1.5 특정 하드웨어 아키텍처 대상 정하기

앱의 배치 타깃으로 지정하는 iOS 버전은 하드웨어 아키텍처 설정에 직접적인 영향을 미친다. Xcode 5에서 기본 하드웨어 아키텍처는 'arm64 armv7 armv7s'다.[03] 그런데 아이폰 3G와 아이팟 터치 2세대처럼 오래된 기기는 하드웨어 아키텍처가 armv6다. 즉, 앱의 iOS 배치 타깃이 iOS 4.2 이하라면 armv6를 추가해 armv6 아키텍처를 지원하는 기기도 고려해야 한다.

실제 경우를 살펴보면, iOS 배치 타깃 설정은 4.0이지만 armv6 아키텍처를 추가하지 않으면 iOS 4.2.1 버전이 설치된 아이폰 3G나 아이팟 터치 2세대에는 개발한 앱을 설치할 수 없다. 반대로 프로젝트가 iOS 4.3과 같거나 낮은 버전을 더 이상 지원하지 않는다면 armv6 하드웨어 아키텍처 설정을 제거해야 할 수도 있다.

따라서 이번에는 앱의 타깃 레벨에 부가적인 하드웨어 아키텍처를 추가하는 방법을 설명한다. 먼저 Xcode 5의 TARGETS에 armv6 하드웨어 아키텍처를 추가하는 방법을 살펴본다. 또한 앱이 사용하는 각각의 서드파티 라이브러리에 대한 타깃 레벨 설정에도 armv6 하드웨어 아키텍처를 추가해본다.

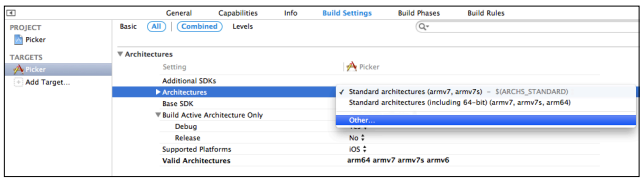
1. Xcode의 메뉴에서 [View] → [Navigators] → [Show Project Navigator]를 선택한다.
2. 프로젝트의 루트 폴더를 선택한다.
3. 프로젝트 에디터에서 TARGETS 아래의 해당 프로젝트 이름을 선택한다.
4. 'Build Settings' 탭을 누른다(아래에 'All'과 'Combined' 탭을 선택했는지 확인).
5. Xcode 5의 기본 아키텍처Architectures 설정(Architectures 아래 Architectures 확인)은 'Standard (armv7 armv7s) - $(ARCHS_STANDARD)'다.

---

03 역자주_ iOS 7에서는 Xcode 5를 사용하며 기본 하드웨어 아키텍처에 'arm64'가 추가되었다.

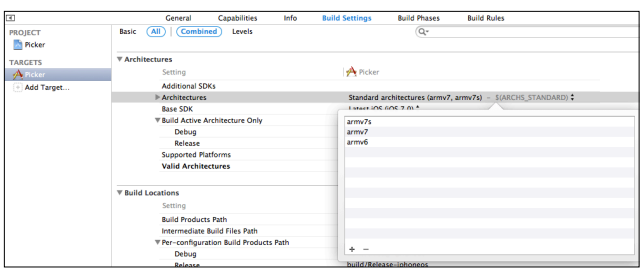
6. 오른쪽 아키텍처 설정을 선택한 다음, 이 설정을 수정하기 위해 'Other'를 선택한다(그림 1-7 참고).

그림 1-7 기타 하드웨어 아키텍처 추가



7. $(ARCHS_STANDARD)를 선택하고 [-]를 눌러 삭제한다. 그리고 [+]를 눌러 armv7s, armv7, armv6를 각각 추가한다(그림 1-8 참고).

그림 1-8 추가 하드웨어 아키텍처 추가



8. 'Build Active Architecture Only' 설정이 'No'로 되어있는지 확인한다.

> **NOTE_** 타깃 레벨 설정에 armv6 아키텍처를 추가하지 않고 배치 타깃이 iOS 4.3 버전 미만인 상태에서 아이튠즈 커넥트에 앱을 업로드하려고 하면 다음과 같은 오류 메시지가 나타난다.
>
> "iPhone/iPod Touch: application executable is missing a required architecture. At least one of the following architecture(s) must be present: armv6."[02]

04 역자주_아이폰/아이팟 터치: 애플리케이션 실행에 필요한 아키텍처가 없습니다. 최소한 다음 아키텍처 중 하나라도 있어야 합니다: armv6

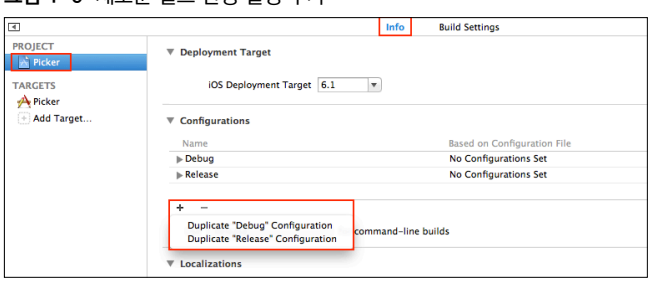
1.1.6 빌드 환경 설정하기

기본적으로 새로운 Xcode 프로젝트는 모두 Default and Release[기본 및 릴리스] 빌드 환경 설정과 함께 만들어진다. 그런데 경우에 따라 프로젝트에 다른 빌드 환경 설정을 추가해야 하는 경우도 있다. 이번에는 이런 경우에 대한 예로 앱을 다른 번들 아이디로 빌드하거나 다른 프로비저닝 프로파일을 사용해 키체인에 접근하는 경우 등을 살펴본다.

다음 단계를 따라 Xcode 프로젝트에 부가적인 빌드 환경 설정을 추가한다.

1. Xcode의 메뉴에서 [View] → [Navigators] → [Show Project Navigator]를 선택한다.
2. 프로젝트의 루트 폴더를 선택한다.
3. 프로젝트 에디터에서 PROJECT 아래의 해당 프로젝트 이름을 선택한다.
4. 'Info' 탭의 Configurations에서 [+]를 눌러 기존 빌드 환경 설정을 복사해 시작점으로 사용한다(그림 1-9 참고).

그림 1-9 새로운 빌드 환경 설정 추가



5. 빌드 환경 설정에 이름을 붙인다. 프로젝트 레벨과 타깃 레벨 모두에 있는 'Build Settings' 탭의 환경 설정 옵션에 방금 이름 붙인 부가적인 빌드 환경 설정을 포함하게 된다.