

Packager for iPhone[®]으로 ADOBE[®] AIR[®] 응용 프로그램 만들기

마지막 업데이트 2011 년 5 월 13 일

법적 고지 사항

법적 고지 사항은 http://help.adobe.com/ko_KR/legalnotices/index.html을 참조하십시오.

목차

1장: iPhone용 AIR 응용 프로그램 만들기 시작

주요 개념	1
Adobe에서 개발자 도구 받기	4
Apple에서 개발자 파일 받기	4
Flash Professional CS5를 사용하여 Hello World iPhone 응용 프로그램 만들기	7

2장: iPhone 응용 프로그램 컴파일 및 디버그

iPhone 아이콘 및 초기 화면 이미지	12
iPhone 응용 프로그램 설정	13
iPhone 응용 프로그램 설치 프로그램(IPA) 파일 컴파일	19
iPhone 응용 프로그램 설치	21
iPhone 응용 프로그램 디버그	22
App Store에 iPhone 응용 프로그램 제출	24

3장: ActionScript 3.0 API의 휴대 장치 지원

휴대 장치에서 지원되지 않는 ActionScript 3.0 API	25
휴대 장치용 AIR 응용 프로그램에만 한정된 ActionScript API	27
모바일 응용 프로그램 개발자가 주목해야 할 ActionScript 3.0 API	30

4장: iPhone 응용 프로그램 설계 고려 사항

하드웨어 가속	32
표시 객체 성능을 향상시키는 다른 방법	34
정보 밀도	34
글꼴 및 텍스트 입력	35
응용 프로그램 상태 저장	36
화면 방향 변경	36
히트 대상	36
메모리 할당	36
드로잉 API	36
이벤트 버블링	37
비디오 성능 최적화	37
Flex 및 Flash 구성 요소	37
응용 프로그램 파일 크기 줄이기	37

1장: iPhone용 AIR 응용 프로그램 만들기 시작

Adobe® Flash® Platform 도구와 ActionScript® 3.0 코드를 사용하여 iPhone 및 iPod Touch용 Adobe® AIR® 응용 프로그램을 만들 수 있습니다. 이러한 응용 프로그램은 다른 iPhone 응용 프로그램과 마찬가지로 배포, 설치 및 실행됩니다.

참고: 본 문서의 나머지 부분에서는 iPhone과 iPod Touch를 모두 "iPhone"으로 지칭합니다.

Packager for iPhone®은 Adobe® Flash® Professional CS5에 포함되어 있습니다. Packager for iPhone은 ActionScript 3.0 바이트코드를 기본 iPhone 응용 프로그램 코드로 컴파일합니다. iPhone 응용 프로그램은 iTunes Store를 통해 iPhone 응용 프로그램 설치 프로그램 파일(.ipa 파일) 형태로 배포됩니다.

Flash Professional CS5 또는 Adobe® Flash® Builder™ 4를 사용하면 응용 프로그램의 소스 ActionScript 3.0 내용을 편집할 수 있습니다.

iPhone 응용 프로그램을 개발하려면 Flash Professional CS5를 사용해야 합니다.

또한 Apple에서 발급하는 iPhone 개발자 인증서도 필요합니다.

중요: iPhone 응용 프로그램 개발을 시작하기 전에 자세한 내용은 32페이지의 “iPhone 응용 프로그램 설계 고려 사항”에서 iPhone용 응용 프로그램 설계 방법에 대한 정보를 참조하십시오. 또한 4페이지의 “Apple에서 개발자 파일 받기”에서 iPhone 응용 프로그램을 만드는 데 필요한 개발자 파일에 대한 자세한 내용을 참조하십시오.

주요 개념

ActionScript 3.0을 사용하여 iPhone 응용 프로그램을 개발하려면 먼저 관련 개념과 작업 과정을 이해해야 합니다.

용어

다음은 iPhone 응용 프로그램을 만들 때 반드시 알아야 할 주요 용어입니다.

iPhone Dev Center 사이트 Apple Computer 웹 사이트(<http://developer.apple.com/iphone/>)에서 다음을 수행할 수 있습니다.

- iPhone 개발자가 되기 위해 지원합니다.
- iPhone 개발 인증서, 프로비저닝 프로파일 및 응용 프로그램 ID(아래에 정의되어 있음)를 관리하고 만듭니다.
- App Store용 응용 프로그램을 제출합니다.

iPhone 개발 인증서 응용 프로그램 개발 용도로 개발자를 식별하기 위해 사용됩니다.

이 파일은 Apple에서 제공합니다. 이 인증서를 P12 인증서로 변환하여 ActionScript 3.0을 사용하여 만든 iPhone 응용 프로그램에 서명합니다. 자세한 내용은 **P12 인증서 파일**을 참조하십시오.

단지 개발 컴퓨터에서 Flash Professional CS5 응용 프로그램을 디버그하고 테스트하기 위해서는 iPhone 개발 인증서가 필요하지 않습니다. 그러나 iPhone에서 응용 프로그램을 설치하고 테스트하려면 개발 인증서가 필요합니다.

개발 인증서는 응용 프로그램의 최종 버전을 만드는 데 사용되는 배포 인증서와는 다릅니다. 응용 프로그램의 최종 버전을 만들 때 Apple로부터 배포 인증서를 받습니다.

인증서 서명 요청 개발 인증서를 생성하는 데 사용되는 개인 정보가 포함되어 있는 파일입니다. CSR 파일이라고도 합니다.

프로비저닝 프로파일 iPhone 응용 프로그램을 테스트하거나 배포할 수 있도록 하는 파일로, Apple에서 제공합니다. 프로비저닝 프로파일은 특정 개발 인증서, 응용 프로그램 ID 및 하나 이상의 장치 ID에 할당됩니다. 프로비저닝 프로파일에는 다음과 같은 여러 가지 유형이 있습니다.

- **개발용 프로비저닝 프로파일** - 개발자의 iPhone에 응용 프로그램의 테스트 버전을 설치하는 데 사용됩니다.
- **테스트용 프로비저닝 프로파일** - 에드혹 프로비저닝 프로파일이라고도 하며 응용 프로그램의 테스트 버전을 여러 사용자 및 iPhone 장치에 배포하는 데 사용됩니다. App Store에 응용 프로그램을 전송하지 않더라도 이 프로비저닝 프로파일과 테스트 응용 프로그램을 사용하면 사용자가 개발된 응용 프로그램을 테스트할 수 있습니다. 참고: 개발용 프로비저닝 프로파일을 사용해도 여러 장치에 테스트 응용 프로그램을 배포할 수 있습니다.
- **배포용 프로비저닝 프로파일** - iPhone 응용 프로그램을 만든 후 App Store에 전송하는 데 사용됩니다.

응용 프로그램 ID 특정 개발자가 개발한 하나 이상의 iPhone 응용 프로그램을 식별하는 고유 문자열입니다. iPhone Dev Center 사이트에서 응용 프로그램 ID를 만듭니다. 각 프로비저닝 프로파일에는 연결된 응용 프로그램 ID 또는 응용 프로그램 ID 패턴이 있습니다. 응용 프로그램을 개발할 때 이 응용 프로그램 ID(또는 패턴)를 사용합니다. 또한 Flash Professional CS5 [iPhone 설정] 대화 상자(또는 응용 프로그램 설명자 파일)에서 응용 프로그램 ID를 사용합니다.

iPhone Dev Center의 응용 프로그램 ID에는 번들 시드 ID와 번들 식별자가 순서대로 포함됩니다. 번들 시드 ID는 5RM86Z4DJM과 같이 Apple에서 응용 프로그램 ID에 할당하는 문자열입니다. 번들 식별자에는 자신이 직접 선택하는 역방향 도메인 이름 문자열이 포함됩니다. 번들 식별자는 와일드카드 응용 프로그램 ID를 나타내는 별표(*)로 끝낼 수 있습니다. 예를 들면 다음과 같습니다.

- 5RM86Z4DJM.com.example.helloWorld
- 96LPVWEASL.com.example.*(와일드카드 응용 프로그램 ID)

iPhone Dev Center에는 다음과 같은 두 가지 유형의 응용 프로그램 ID가 있습니다.

- **와일드카드 응용 프로그램 ID** - iPhone Dev Center에서 이러한 응용 프로그램 ID는 별표(*)로 끝납니다(예: 96LPVWEASL.com.myDomain* 또는 96LPVWEASL.*) 이러한 종류의 응용 프로그램 ID를 사용하는 프로비저닝 프로파일에서는 해당 패턴과 일치하는 응용 프로그램 ID를 사용하는 테스트 응용 프로그램을 생성할 수 있습니다. 응용 프로그램의 응용 프로그램 ID의 경우 별표를 유효한 문자열로 바꿀 수 있습니다. 예를 들어 iPhone Dev Center 사이트에서 96LPVWEASL.com.example.*를 응용 프로그램 ID로 지정하는 경우 com.example.foo 또는 com.example.bar를 응용 프로그램의 응용 프로그램 ID로 사용할 수 있습니다.
- **특정 응용 프로그램 ID** - 이러한 ID는 한 응용 프로그램에서 사용할 고유한 응용 프로그램 ID를 정의합니다. iPhone Dev Center에서 이러한 응용 프로그램 ID는 별표로 끝나지 않습니다. 예를 들면 96LPVWEASL.com.myDomain.myApp 등입니다. 이러한 종류의 응용 프로그램 ID를 사용하는 프로비저닝 프로파일에서는 응용 프로그램이 해당 응용 프로그램 ID와 정확히 일치해야 합니다. 예를 들어 iPhone Dev Center 사이트에서 96LPVWEASL.com.example.helloWorld를 응용 프로그램 ID로 지정하는 경우 응용 프로그램의 응용 프로그램 ID로 com.example.foo를 사용해야 합니다.

응용 프로그램을 개발할 때 응용 프로그램 설명자 파일 또는 Flash Professional CS5의 [iPhone 설정] 대화 상자에서 응용 프로그램 ID를 지정합니다. 응용 프로그램 ID에 대한 자세한 내용은 14페이지의 “Flash Professional CS5에서 iPhone 응용 프로그램 속성 설정”의 “배포 탭” 단원 또는 15페이지의 “응용 프로그램 설명자 파일에서 iPhone 응용 프로그램 속성 설정”을 참조하십시오.

중요: 응용 프로그램 ID를 지정하는 경우 응용 프로그램 ID의 번들 시드 ID 부분은 무시하십시오. 예를 들어 Apple에서 응용 프로그램 ID를 96LPVWEASL.com.example.bob.myApp으로 표시하는 경우 96LPVWEASL을 무시하고 com.example.bob.myApp을 응용 프로그램 ID로 사용하십시오. Apple에서 응용 프로그램 ID를 5RM86Z4DJM.*로 표시하는 경우 5RM86Z4DJM을 무시하십시오. 이는 와일드카드 응용 프로그램 ID입니다.

iPhone Dev Center(<http://developer.apple.com/iphone>)에서 프로비저닝 프로파일과 연결된 응용 프로그램 ID(또는 와일드카드 응용 프로그램 ID 패턴)를 확인할 수 있습니다. iPhone Developer Program Portal로 이동한 다음 [Provisioning] 섹션으로 이동합니다.

P12 인증서 파일 P12 파일(확장자가 .p12인 파일)은 인증서 파일(개인 정보 교환 파일) 유형 중 하나입니다. Packager for iPhone은 이 유형의 인증서를 사용하여 iPhone 응용 프로그램을 만듭니다. Apple에서 받은 개발자 인증서를 이 형태의 인증서로 변환합니다.

고유 장치 ID 특정 iPhone을 식별하는 고유 코드입니다. UDID 또는 장치 ID라고도 합니다.

개발 작업 과정 개요

iPhone용 응용 프로그램은 다음 단계에 따라 개발됩니다.

- 1 Adobe Flash Professional CS5를 설치합니다.
- 2 iTunes를 설치합니다.
- 3 Apple로부터 개발자 파일을 받습니다. 개발자 파일에는 개발자 인증서와 프로비저닝 프로파일도 포함됩니다. 자세한 내용은 4페이지의 [“Apple에서 개발자 파일 받기”](#)를 참조하십시오.
- 4 개발 인증서를 P12 인증서 파일로 변환합니다. Flash CS5에는 P12 인증서를 사용해야 합니다. 자세한 내용은 4페이지의 [“Apple에서 개발자 파일 받기”](#)를 참조하십시오.
- 5 iTunes를 사용하여 프로비저닝 프로파일을 iPhone에 연결합니다.
- 6 Flash Professional CS5에서 응용 프로그램을 작성합니다.

iPhone 응용 프로그램 코드를 설계하고 최적화하는 유용한 방법을 이해해야 합니다. 자세한 내용은 32페이지의 [“iPhone 응용 프로그램 설계 고려 사항”](#)을 참조하십시오.

또한 iPhone에서 기능이 제한되거나 지원되지 않는 ActionScript 3.0 API도 몇 가지 있습니다. 자세한 내용은 25페이지의 [“ActionScript 3.0 API의 휴대 장치 지원”](#)을 참조하십시오.

Flash Builder 4.0을 사용하여 응용 프로그램의 ActionScript 3.0 코드를 편집할 수도 있습니다.

Flash Professional CS5를 사용하여 개발용 컴퓨터에서 응용 프로그램을 테스트할 수 있습니다.

- 7 응용 프로그램의 아이콘 아트와 초기 화면 아트를 제작합니다. 모든 iPhone 응용 프로그램에는 사용자가 응용 프로그램을 식별할 수 있는 일련의 아이콘이 포함되어 있습니다. 초기 화면 이미지는 iPhone에서 프로그램을 로드할 때 표시됩니다. 자세한 내용은 12페이지의 [“iPhone 아이콘 및 초기 화면 이미지”](#)를 참조하십시오.

- 8 iPhone 설정을 편집합니다. 편집 가능한 설정은 다음과 같습니다.

- 응용 프로그램의 식별 정보(파일 이름, 응용 프로그램 이름, 버전 번호 및 응용 프로그램 ID 등)
- 응용 프로그램의 소스 아이콘 아트 위치
- 응용 프로그램에 할당된 P12 인증서 및 프로비저닝 프로파일
- 응용 프로그램의 초기 중형비

Flash Professional CS5를 사용하는 경우 [iPhone 설정] 대화 상자에서 이러한 설정을 편집할 수 있습니다. 자세한 내용은 14페이지의 [“Flash Professional CS5에서 iPhone 응용 프로그램 속성 설정”](#)을 참조하십시오.

또한 응용 프로그램 설명자 파일에서 이러한 설정을 직접 편집할 수도 있습니다. 자세한 내용은 15페이지의 [“응용 프로그램 설명자 파일에서 iPhone 응용 프로그램 속성 설정”](#)을 참조하십시오.

- 9 Packager for iPhone을 사용하여 IPA 파일을 컴파일합니다. 자세한 내용은 19페이지의 [“iPhone 응용 프로그램 설치 프로그램\(IPA\) 파일 컴파일”](#)을 참조하십시오.

- 10 iPhone에 응용 프로그램을 설치하고 테스트합니다. iTunes를 사용하여 IPA 파일을 설치합니다.

에드혹 배포의 경우 이 일반 프로세스를 반복하되 개발용 프로비저닝 프로파일 대신 테스트용 프로비저닝 프로파일을 사용합니다. 최종 응용 프로그램 배포의 경우 배포용 프로비저닝 프로파일을 사용하여 이 프로세스를 반복합니다. 여러 가지 프로비저닝 프로파일에 대한 자세한 내용은 1페이지의 [“용어”](#)를 참조하십시오.

배포 버전의 응용 프로그램을 만든 경우 24페이지의 [“App Store에 iPhone 응용 프로그램 제출”](#)의 지침을 참조하십시오.

기본 iPhone 응용 프로그램을 만드는 방법에 대한 간단한 자습서는 7페이지의 [“Flash Professional CS5를 사용하여 Hello World iPhone 응용 프로그램 만들기”](#)에서 참조할 수 있습니다.

Adobe에서 개발자 도구 받기

ActionScript 3.0을 사용하여 iPhone 응용 프로그램을 개발하려면 Flash Professional CS5가 필요합니다.

중요: Packager for iPhone의 버전을 Flash Professional CS5에 포함되었던 Preview 버전에서 업데이트해야 합니다. Flash Professional CS5에서 [도움말] > [업데이트]를 선택합니다.

Flash Builder 4를 사용하여 ActionScript 코드를 편집할 수도 있습니다. Flash Builder 4는 <http://www.adobe.com/kr/products/flashbuilder/>에서 사용할 수 있습니다.

Apple에서 개발자 파일 받기

모든 iPhone용 응용 프로그램 개발 과정에서는 먼저 Apple로부터 iPhone 개발자 파일을 받아야 합니다. iPhone 개발자 인증서와 모바일용 프로비저닝 프로파일을 받아야 하고 다른 프로비저닝 프로파일도 필요합니다. 이러한 파일의 정의는 1페이지의 “용어”를 참조하십시오.

참고: 이러한 파일을 받는 것은 응용 프로그램 개발 프로세스에서 중요한 부분입니다. 응용 프로그램을 개발하기 전에 파일 받기를 완료하도록 하십시오. 개발자 파일을 받는 것은 간단한 작업이 아닙니다. 여기에 나온 지침과 Apple iPhone Dev Center 사이트에 있는 지침을 주의 깊게 읽으십시오.

iPhone 개발자 파일 받기 및 사용

Apple이 제공하는 iPhone 개발자 인증서와 프로비저닝 프로파일을 받아야 합니다. 또한 받은 인증서를 P12 인증서로 변환해야 합니다.

iTunes 설치

iPhone에 응용 프로그램을 설치하려면 iTunes가 필요합니다. 또한 iPhone의 장치 ID를 확인할 때도 iTunes를 사용합니다. 장치 ID는 iPhone 개발자 인증서를 신청할 때 필요합니다.

iPhone 개발자 인증서 신청 및 프로비저닝 프로파일 만들기

아직 등록하지 않은 경우 Apple iPhone Dev Center 사이트(<http://developer.apple.com/iphone/>)에서 등록된 iPhone 개발자가 되기 위해 신청합니다.

참고: iPhone용 AIR 응용 프로그램을 개발하는 데는 iPhone SDK 또는 XCode가 필요하지 않습니다. 하지만 등록된 iPhone 개발자여야 하며 개발자 인증서와 프로비저닝 프로파일을 받아야 합니다.

- 1 iPhone 개발자 계정 ID를 사용하여 iPhone Dev Center에 로그인합니다.
- 2 iPhone Dev Center에서 iPhone 개발자 인증서를 신청(및 구입)합니다.
iPhone 개발자 프로그램 활성화 코드가 들어 있는 전자 메일 메시지를 Apple로부터 받게 됩니다.
- 3 iPhone Dev Center로 돌아옵니다. 개발자 프로그램 활성화에 대한 지침을 따릅니다(해당 메시지가 나타나면 활성화 코드 입력).
- 4 활성화 코드가 승인되면 iPhone Dev Center의 [iPhone Developer Program Portal] 섹션으로 이동합니다.
- 5 인증서 서명 요청 파일을 만듭니다. 이 파일을 사용하여 iPhone 개발 인증서를 받습니다. 자세한 지침은 5페이지의 “인증서 서명 요청 생성”을 참조하십시오.
- 6 이후 단계에서 iPhone의 장치 ID나 고유 장치 ID(UDID)를 입력하라는 메시지가 표시됩니다. UDID는 iTunes에서 얻을 수 있습니다.
 - a iPhone에 USB 케이블을 연결하고 iTunes에서 iPhone에 대한 요약 탭을 선택합니다.

- b** iPhone 개발자 센터 사이트에서 프로비저닝 프로파일을 다운로드한 후 iTunes에 추가합니다.
 - c** 그런 다음 표시되는 일련 번호를 클릭합니다. 그러면 UDID가 표시됩니다. Command+C(Mac) 또는 Ctrl+C(Windows)를 눌러 UDID를 클립보드에 복사합니다.
- 7** 프로비저닝 프로파일과 iPhone 개발 인증서를 만들어 설치합니다.
iPhone Dev Center의 지침을 따릅니다. [iPhone Developer Program Portal] 섹션에서 지침을 확인합니다. Development Provisioning Assistant를 사용하여 개발 인증서를 받고 프로비저닝 프로파일을 만들 수 있습니다.
XCode 관련 단계는 무시하십시오. Flash Professional CS5를 사용하여 iPhone 응용 프로그램을 개발하는 데는 XCode가 필요 없습니다.
- 8** iTunes에서 [파일] > [보관함에 추가]를 선택합니다. 파일 이름 확장명이 mobileprovision인 프로비저닝 프로파일 파일을 선택합니다. 그런 다음 iPhone을 iTunes와 동기화합니다.
그러면 이 프로비저닝 프로파일이 연결된 응용 프로그램을 iPhone에서 테스트할 수 있습니다.
특정 프로비저닝 프로파일이 iTunes에 추가되어 있는지 여부는 보관함에 해당 프로파일을 추가해 보면 알 수 있습니다. 추가할 때 기존 프로비저닝 프로파일을 바꿀지 묻는 메시지가 나타나면 [취소] 버튼을 누릅니다. 이 경우 해당 프로파일이 이미 설치되어 있는 것입니다. 또한 iPhone에서 프로비저닝 프로파일이 설치되어 있는지 확인할 수도 있습니다.
 - a** iPhone에서 설정 응용 프로그램을 엽니다.
 - b** [일반] 범주를 엽니다.
 - c** [프로파일]을 누릅니다. [프로파일] 페이지에 설치된 프로비저닝 프로파일 목록이 표시됩니다.
- 9** 아직 다운로드하지 않은 경우 iPhone 개발 인증서 파일(.cer 파일)을 다운로드합니다.
이 파일을 다운로드할 수 있도록 Development Provisioning Assistant에서 링크를 제공할 수 있습니다. 또한 Apple iPhone Dev Center 사이트(<http://developer.apple.com/iphone/>)에 있는 Provisioning Portal의 Certificates 섹션에서도 이 파일을 찾을 수 있습니다.
- 10** 그런 다음 iPhone 개발자 인증서를 P12 파일로 변환합니다. 자세한 지침은 6페이지의 “개발자 인증서를 P12 파일로 변환”을 참조하십시오.
이제 간단한 Hello World 응용 프로그램을 만들 수 있습니다. 자세한 내용은 7페이지의 “Flash Professional CS5를 사용하여 Hello World iPhone 응용 프로그램 만들기”를 참조하십시오.

인증서 서명 요청 생성

개발자 인증서를 받으려면 Apple iPhone Dev Center 사이트에서 제출하는 인증서 서명 요청 파일을 생성합니다.

Mac OS에서 인증서 서명 요청 생성

Mac OS에서 키체인 접근 응용 프로그램을 사용하여 코드 서명 요청을 생성할 수 있습니다. 키체인 접근 응용 프로그램은 [응용 프로그램] 디렉토리의 [유틸리티] 하위 디렉토리에 있습니다. [키체인 접근] 메뉴에서 [인증 지원] > [인증 기관에서 인증서 요청]을 선택합니다.

- 1** 키체인 접근을 엽니다.
- 2** [키체인 접근] 메뉴에서 [환경설정] 선택합니다.
- 3** [환경설정] 대화 상자에서 [인증서]를 클릭합니다. 그런 다음 [OCSP(Online Certificate Status Protocol)] 및 [CRL(Certificate Revocation List)]을 [Off]로 설정합니다. 대화 상자를 닫습니다.
- 4** [키체인 접근] 메뉴에서 [인증 지원] > [인증 기관에서 인증서 요청]을 선택합니다.
- 5** iPhone 개발자 계정 ID와 일치하는 전자 메일 주소 및 이름을 입력합니다. CA 전자 메일 주소는 입력하지 마십시오. [디스크에 저장됨]을 선택한 다음 [계속] 버튼을 클릭합니다.
- 6** 파일(CertificateSigningRequest.certSigningRequest)을 저장합니다.

- 7 CSR 파일을 [iPhone 개발자 사이트](#)에서 Apple에 업로드합니다. "iPhone 개발자 인증서 신청 및 프로비저닝 프로파일 만들기"를 참조하십시오.

Windows에서 인증서 서명 요청 생성

Windows 개발자의 경우에도 Mac 컴퓨터에서 iPhone 개발자 인증서를 받는 것이 가장 쉽지만 Windows 컴퓨터에서 인증서를 받을 수도 있습니다. 먼저 OpenSSL을 사용하여 인증서 서명 요청(CSR 파일)을 만듭니다.

- 1 Windows 컴퓨터에 OpenSSL을 설치합니다. <http://www.openssl.org/related/binaries.html>로 이동합니다.

또한 Open SSL 다운로드 페이지에 나열된 Visual C++ 2008 재배포 가능 파일을 설치해야 할 수도 있습니다. 컴퓨터에 Visual C++를 설치할 필요는 없습니다.

- 2 Windows 명령 세션을 열고 CD 명령을 사용하여 OpenSSL bin 디렉토리(예: c:\OpenSSL\bin\)로 이동합니다.

- 3 명령줄에 다음을 입력하여 개인 키를 만듭니다.

```
openssl genrsa -out mykey.key 2048
```

이 개인 키 파일을 저장합니다. 이 파일은 나중에 사용합니다.

OpenSSL을 사용하는 경우 오류 메시지를 무시하지 마십시오. OpenSSL에서 오류 메시지를 생성하는 경우에도 파일을 출력할 수 있습니다. 그러나 해당 파일이 사용 가능한 상태가 아닐 수 있습니다. 오류가 표시되는 경우 구문을 확인하고 명령을 다시 실행합니다.

- 4 명령줄에 다음을 입력하여 CSR 파일을 만듭니다.

```
openssl req -new -key mykey.key -out CertificateSigningRequest.certSigningRequest -subj  
"/emailAddress=yourAddress@example.com, CN=John Doe, C=US"
```

전자 메일 주소, CN(인증서 이름) 및 C(국가) 값을 적절하게 바꿉니다.

- 5 [iPhone 개발자 사이트](#)에서 Apple에 CSR 파일을 업로드합니다. "iPhone 개발자 인증서 신청 및 프로비저닝 프로파일 만들기"를 참조하십시오.

개발자 인증서를 P12 파일로 변환

Flash Professional CS5를 사용하여 iPhone 응용 프로그램을 개발하려면 P12 인증서 파일을 사용해야 합니다. 이 인증서는 Apple로부터 받은 Apple iPhone 개발자 인증서 파일을 기반으로 생성합니다.

Mac OS에서 iPhone 개발자 인증서를 P12 파일로 변환

Apple에서 Apple iPhone 인증서를 다운로드한 후에는 인증서를 P12 인증서 포맷으로 내보냅니다. Mac® OS에서 이 작업을 수행하는 방법은 다음과 같습니다.

- 1 키체인 접근 응용 프로그램(응용 프로그램/유틸리티 폴더에 있음)을 엽니다.
- 2 아직 키체인에 인증서를 추가하지 않은 경우 [파일] > [가져오기]를 선택합니다. 그런 다음 Apple에서 받은 인증서 파일(.cer 파일)로 이동합니다.
- 3 [키체인 접근]에서 [키] 카테고리를 선택합니다.
- 4 iPhone 개발 인증서에 연결된 개인 키를 선택합니다.
개인 키는 iPhone 개발자별로 식별됩니다. 즉, <이름> <성> 공용 인증서가 쌍을 이루고 있습니다.
- 5 [파일] > [항목 보내기]를 선택합니다.
- 6 개인 정보 교환(.p12) 파일 포맷으로 키를 저장합니다.
- 7 이 키를 다른 컴퓨터로 가져올 때 사용할 암호를 만들라는 메시지가 나타납니다.

Windows에서 Apple 개발자 인증서를 P12 인증서로 변환

Flash CS5를 사용하여 iPhone 응용 프로그램을 개발하려면 P12 인증서 파일을 사용해야 합니다. 이 인증서는 Apple로부터 받은 Apple iPhone 개발자 인증서 파일을 기반으로 생성합니다.

- 1 Apple로부터 받은 개발자 인증서 파일을 PEM 인증서 파일로 변환합니다. OpenSSL bin 디렉토리에서 다음 명령줄 명령문을 실행합니다.

```
openssl x509 -in developer_identity.cer -inform DER -out developer_identity.pem -outform PEM
```

- 2 Mac 컴퓨터의 키체인에서 가져온 개인 키를 사용하는 경우 키를 PEM 키로 변환합니다.

```
openssl pkcs12 -nocerts -in mykey.p12 -out mykey.pem
```

- 3 이제 iPhone 개발자 인증서의 PEM 버전과 키를 사용하여 유효한 P12 파일을 생성할 수 있습니다.

```
openssl pkcs12 -export -inkey mykey.key -in developer_identity.pem -out iphone_dev.p12
```

Mac OS 키체인에서 가져온 키를 사용하는 경우 이전 단계에서 생성한 PEM 버전을 사용합니다. 그렇지 않으면 앞서 Windows에서 생성한 OpenSSL 키를 사용합니다.

인증서, 장치 ID, 응용 프로그램 ID 및 프로비저닝 프로파일 관리

Apple iPhone Dev Center 사이트(<http://developer.apple.com/iphone/>)에서 인증서, 장치 ID, 응용 프로그램 ID 및 프로비저닝 프로파일을 관리할 수 있습니다. 사이트의 [iPhone Developer Program Portal] 섹션으로 이동합니다.

- 개발 인증서를 관리하려면 [Certificates] 링크를 클릭합니다. 인증서를 만들거나, 다운로드하거나, 해제할 수 있습니다. 인증서를 만들려면 먼저 인증서 서명 요청을 만들어야 합니다. 5페이지의 “인증서 서명 요청 생성”을 참조하십시오.
- 개발 응용 프로그램을 설치할 수 있는 장치 목록을 관리하려면 [Devices] 링크를 클릭합니다.
- 응용 프로그램 ID를 관리하려면 [App IDs] 링크를 클릭합니다. 프로비저닝 프로파일을 만드는 경우 응용 프로그램 ID에 바인딩됩니다.
- 프로비저닝 프로파일을 관리하려면 [Provisioning] 링크를 클릭합니다. Development Provisioning Assistant를 사용하여 개발 프로비저닝 프로파일을 만들 수도 있습니다.
- App Store에 응용 프로그램을 제출하거나 Ad Hoc 버전의 응용 프로그램을 만들려는 경우 [Distribution] 링크를 클릭합니다. 이 섹션에는 App Store에 응용 프로그램을 제출하는 데 사용하는 iTunes Connect 사이트에 대한 링크가 있습니다.

Flash Professional CS5를 사용하여 Hello World iPhone 응용 프로그램 만들기

중요: 응용 프로그램을 만들기 전에 먼저 필요한 개발자 응용 프로그램과 파일을 다운로드하십시오. 자세한 내용은 4페이지의 “Adobe에서 개발자 도구 받기”와 4페이지의 “Apple에서 개발자 파일 받기”를 참조하십시오.

Flash Professional CS5 프로젝트 만들기

Flash Professional CS5에서 직접 iPhone 응용 프로그램을 생성할 수 있습니다.

- 1 Flash CS5를 엽니다.
- 2 [파일] > [새로 만들기]를 선택합니다.
- 3 [iPhone]을 선택합니다.
- 4 [확인] 버튼을 클릭합니다.

응용 프로그램에 내용 추가

다음으로 "Hello world!"라는 텍스트를 응용 프로그램에 추가합니다.

- 1 [텍스트 도구]를 선택하고 스테이지를 클릭합니다.
- 2 텍스트 필드에 대한 [속성] 설정에서 [클래식 텍스트]([TLF 텍스트] 아님)를 선택합니다.
기본 응용 프로그램이므로 클래식 텍스트가 적합합니다. TLF 텍스트를 사용하려면 몇 가지 다른 설정을 적용해야 합니다.
자세한 내용은 35페이지의 “글꼴 및 텍스트 입력”을 참조하십시오.
- 3 새 텍스트 필드에 "Hello World!"를 입력합니다.
- 4 [선택 도구]로 텍스트 필드를 선택합니다.
- 5 그런 다음 속성 관리자 열고 다음 설정을 적용합니다.
 - 문자 > 집합: `_sans`
 - 문자 > 크기: 50
 - 위치 > X: 20
 - 위치 > Y: 20
- 6 파일을 저장합니다.
- 7 [컨트롤] > [동영상 테스트] > [AIR Debug Launcher(모바일)]를 선택합니다.
Flash Professional CS5가 SWF 내용을 컴파일하고 응용 프로그램의 버전을 ADL(AIR Debug Launcher)에 표시합니다.
. 따라서 응용 프로그램을 간편하게 미리 볼 수 있습니다.

응용 프로그램의 아이콘 아트 및 초기 화면 아트 제작

모든 iPhone 응용 프로그램에는 iTunes 응용 프로그램 사용자 인터페이스와 iPhone 화면에 표시되는 아이콘이 있습니다.

- 1 프로젝트 디렉토리에 새 디렉토리를 만들고 이름을 `icons`로 지정합니다.
- 2 `icons` 디렉토리에 PNG 파일을 3개 만들고 각각 `Icon29.png`, `Icon57.png`, `Icon512.png`로 이름을 지정합니다.
- 3 PNG 파일을 편집하여 응용 프로그램에 적합한 아트를 만듭니다. 이 파일은 29 x 29 픽셀, 57 x 57 픽셀, 512 x 512 픽셀 크기여야 합니다. 이 테스트에는 단색 사각형을 아트로 사용하면 됩니다.

모든 iPhone 응용 프로그램은 iPhone에서 로드되는 동안 초기 이미지를 표시합니다. 이 초기 이미지는 PNG 파일로 정의합니다.

- 1 기본 개발 디렉토리에 `Default.png`라는 PNG 파일을 만듭니다. 이 파일을 `icons` 하위 디렉토리에 넣지 마십시오. 파일 이름은 반드시 대문자 D로 시작하는 `Default.png`이어야 합니다.
- 2 폭 320픽셀, 높이 480 픽셀로 파일을 편집합니다. 지금은 내용을 흰색 사각형으로 두면 됩니다. 내용은 나중에 변경합니다.

참고: Apple App Store로 응용 프로그램을 전송할 때는 PNG 버전이 아니라 JPG 버전의 512픽셀 파일을 사용합니다. PNG 버전은 응용 프로그램의 개발 버전을 테스트하는 동안에만 사용합니다.

이러한 그래픽에 대한 자세한 내용은 12페이지의 “iPhone 아이콘 및 초기 화면 이미지”를 참조하십시오.

응용 프로그램 설정 편집

중요: 아직 다운로드하지 않은 경우 iPhone 개발에 필요한 개발자 응용 프로그램 및 파일을 다운로드하십시오. 자세한 내용은 4페이지의 “Apple에서 개발자 파일 받기”를 참조하십시오.

Flash Professional CS5 [iPhone 설정] 대화 상자에서 iPhone 응용 프로그램의 여러 가지 기본 속성을 정의합니다.

- 1 [파일] > [iPhone OS 설정]을 선택합니다.

2 [일반] 탭에서 다음 설정을 적용합니다.

- 출력 파일: **HelloWorld.ipa**
생성할 iPhone 설치 프로그램 파일의 파일 이름입니다.
- 응용 프로그램 이름: **Hello World**
iPhone에서 응용 프로그램 아이콘 아래에 표시되는 응용 프로그램 이름입니다.
- 버전: 1.0
응용 프로그램의 버전입니다.
- 중첩비: 세로
- 전체 화면: 선택
- 자동 방향: 선택 안 함
- 렌더링: CPU

다른 렌더링 옵션(GPU 및 자동)은 렌더링에 하드웨어 가속을 사용하는 옵션입니다. 이 기능은 하드웨어 가속을 활용하도록 디자인된, 그래픽을 많이 사용하는 응용 프로그램(예: 게임)의 성능을 향상시키는 데 도움이 될 수 있습니다. 자세한 내용은 32페이지의 “[하드웨어 가속](#)”을 참조하십시오.

- 포함된 파일: [포함된 파일] 목록에 초기 화면 아트 파일(Default.png)을 추가합니다.

참고: 이 Hello World 예의 경우 이 지침에 나온 설정과 다르게 수정하지 마십시오. 버전 설정을 비롯한 일부 설정에는 특정 제한 사항이 있습니다. 이러한 제한 사항은 13페이지의 “[iPhone 응용 프로그램 설정](#)”에 설명되어 있습니다.

3 [배포] 탭에서 다음 설정을 적용합니다.

- 인증서: **Apple**에서 받은 개발자 인증서를 기반으로 만든 .p12 인증서를 찾아 선택합니다.
이 인증서는 파일에 서명하는 데 사용됩니다. **Apple iPhone** 인증서는 .p12 포맷으로 변환해야 합니다. 자세한 내용은 4페이지의 “[Adobe에서 개발자 도구 받기](#)”를 참조하십시오.
- 암호: 인증서의 암호를 입력합니다.
- 프로비저닝 파일: **Apple**에서 받은 개발자 프로비저닝 파일을 찾아 선택합니다. 자세한 내용은 4페이지의 “[Apple에서 개발자 파일 받기](#)”를 참조하십시오.
- 응용 프로그램 ID: 이 필드를 선택 가능한 경우 **Apple**에 제공한 응용 프로그램 ID와 일치하는 응용 프로그램 ID를 입력할 수 있습니다(예: com.example.as3.HelloWorld).
이 응용 프로그램 ID는 응용 프로그램을 고유하게 식별합니다.
프로비저닝 프로파일이 특정 응용 프로그램 ID에 바인딩된 경우에는 이 필드를 선택할 수 없습니다. 이 경우 필드에 해당 응용 프로그램 ID가 표시됩니다.
응용 프로그램 ID 지정에 대한 자세한 내용은 14페이지의 “[Flash Professional CS5에서 iPhone 응용 프로그램 속성 설정](#)”의 “배포 탭” 단원을 참조하십시오.

4 [아이콘] 탭의 아이콘 목록에서 29 x 29 아이콘을 클릭합니다. 그런 다음 앞서 만든 29 x 29 픽셀 PNG 파일의 위치를 지정합니다(8페이지의 “[응용 프로그램의 아이콘 아트 및 초기 화면 아트 제작](#)” 참조). 그리고 57 x 57 픽셀 아이콘과 512 x 512 픽셀 아이콘의 PNG 파일을 지정합니다.

5 [확인] 버튼을 클릭합니다.

6 파일을 저장합니다.

응용 프로그램 설정에 대한 자세한 내용은 13페이지의 “[iPhone 응용 프로그램 설정](#)”을 참조하십시오.

IPA 파일 컴파일

이제 IPA 설치 프로그램 파일을 컴파일할 수 있습니다.

- 1 [파일] > [제작]을 선택합니다.
- 2 iPhone 설정 대화 상자에서 [확인] 버튼을 클릭합니다.

Packager for iPhone이 프로젝트 디렉토리에 iPhone 응용 프로그램 설치 프로그램 파일(HelloWorld.ipa 파일)을 생성합니다. IPA 파일을 컴파일하는 데는 몇 분 정도 걸릴 수 있습니다.

iPhone에 응용 프로그램 설치

테스트를 위해 iPhone에 iPhone 응용 프로그램을 설치하려면

- 1 iTunes 응용 프로그램을 엽니다.
- 2 이 응용 프로그램의 프로비저닝 프로파일을 iTunes에 추가합니다. iTunes에서 [파일] > [보관함에 추가]를 선택합니다. 프로비저닝 프로파일 파일(파일 유형이 mobileprovision인 파일)을 선택합니다.
지금은 개발자 iPhone에서 응용 프로그램을 테스트하는 것이므로 개발용 프로비저닝 프로파일을 사용합니다.
나중에 iTunes Store에 응용 프로그램을 배포할 때는 배포용 프로파일을 사용해야 합니다. 에드혹 방식으로 응용 프로그램을 배포(iTunes Store를 통하지 않고 여러 장치에 배포)하려면 에드혹용 프로비저닝 프로파일을 사용합니다.
프로비저닝 프로파일에 대한 자세한 내용은 4페이지의 “[Apple에서 개발자 파일 받기](#)”를 참조하십시오.
- 3 일부 iTunes 버전은 같은 버전의 응용 프로그램이 이미 설치되어 있으면 응용 프로그램을 바꾸지 않습니다. 이 경우 장치와 iTunes의 응용 프로그램 목록에서 응용 프로그램을 삭제하십시오.
- 4 응용 프로그램의 IPA 파일을 두 번 클릭합니다. 해당 파일은 응용 프로그램 목록에 표시됩니다.
- 5 iPhone을 컴퓨터의 USB 포트에 연결합니다.
- 6 iTunes에서 장치의 [응용 프로그램] 탭에 있는 설치할 응용 프로그램 목록에 응용 프로그램이 선택되어 있는지 확인합니다.
- 7 iTunes 응용 프로그램의 왼쪽 목록에서 장치를 선택합니다. 그런 다음 [동기화] 버튼을 클릭합니다. 동기화가 완료되면 iPhone에 Hello World 응용 프로그램이 나타납니다.

새 버전이 설치되지 않은 경우 iPhone과 iTunes의 응용 프로그램 목록에서 응용 프로그램을 삭제하고 이 절차를 다시 실행합니다. 현재 설치된 버전의 응용 프로그램 ID와 버전이 동일하기 때문일 수 있습니다.

응용 프로그램을 설치하려고 할 때 iTunes에서 오류를 표시하는 경우 21페이지의 “[iPhone 응용 프로그램 설치](#)”의 “응용 프로그램 설치 문제 해결”을 참조하십시오.

초기 화면 그래픽 편집

앞서 응용 프로그램을 컴파일하기 전에 Default.png 파일(8페이지의 “[응용 프로그램의 아이콘 아트 및 초기 화면 아트 제작](#)” 참조)을 만들었습니다. 이 PNG 파일은 응용 프로그램이 로드되는 동안 시작 이미지로 사용됩니다. iPhone에서 응용 프로그램을 테스트하는 경우 시작 시에 이 빈 화면이 표시되는 것을 볼 수 있습니다.

이 이미지를 응용 프로그램의 시작 화면(“Hello World!”)과 일치하도록 변경해야 합니다.

- 1 장치에서 응용 프로그램을 엽니다. 첫 번째 “Hello World” 텍스트가 나타나면 화면 아래쪽에 있는 [홈] 버튼을 누르고 있습니다. [홈] 버튼을 누른 상태에서 iPhone 맨 위에 있는 [잠자기/깨우기] 버튼을 누릅니다. 이렇게 하면 스크린 샷이 생성되어 카메라 롤로 보내집니다.
- 2 iPhoto 또는 기타 사진 전송 응용 프로그램에서 개발 컴퓨터로 이미지를 전송합니다. Mac OS에서는 이미지 캡처 응용 프로그램을 사용할 수도 있습니다.

전자 메일을 통해 사진을 개발 컴퓨터로 보낼 수도 있습니다.

- Photos 응용 프로그램을 엽니다.

- 카메라 롤을 엽니다.
- 캡처한 스크린 샷 이미지를 엽니다.
- 이미지를 누르고 왼쪽 아래에 있는 "전달"(화살표) 버튼을 누릅니다. 그런 다음 [이메일 메시지에 사진 보내기] 버튼을 눌러 이미지를 자신의 전자 메일 주소로 보냅니다.

3 개발 디렉토리에 있는 **Default.png** 파일을 화면 캡처 이미지의 PNG 버전으로 바꿉니다.

4 응용 프로그램을 다시 컴파일(10페이지의 “IPA 파일 컴파일” 참조)하고 iPhone에 다시 설치합니다.

이제 응용 프로그램을 로드할 때 새 시작 화면이 사용됩니다.

참고: 크기(320 x 480 픽셀)만 맞으면 어떤 아트든 **Default.png** 파일로 만들 수 있습니다. 그러나 일반적으로 **Default.png** 이미지는 응용 프로그램의 초기 상태와 일치하는 것이 좋습니다.

2장: iPhone 응용 프로그램 컴파일 및 디버그

Packager for iPhone을 사용하여 iPhone 응용 프로그램을 컴파일할 수 있습니다. Packager for iPhone은 Flash Professional CS5에 포함되어 있습니다.

개발용 컴퓨터에서 응용 프로그램을 디버그할 수도 있으며, iPhone에 디버그 버전을 설치하고 Flash Professional CS5에서 trace() 출력을 수신할 수도 있습니다.

iPhone 응용 프로그램을 만드는 전체 과정에 대한 자습서는 7페이지의 [“Flash Professional CS5를 사용하여 Hello World iPhone 응용 프로그램 만들기”](#)에서 참조할 수 있습니다.

iPhone 아이콘 및 초기 화면 이미지

모든 iPhone 응용 프로그램에는 iTunes 응용 프로그램 사용자 인터페이스와 iPhone에 표시되는 아이콘이 포함되어 있습니다.

iPhone 응용 프로그램 아이콘

iPhone 응용 프로그램에 사용할 다음 아이콘을 정의합니다.

- 29 x 29 픽셀 아이콘 - iPhone 및 iPod Touch의 Spotlight 검색 결과에서 이 아이콘을 사용합니다.
- 48 x 48 픽셀 아이콘 - iPad의 Spotlight 검색 결과에서 이 아이콘을 사용합니다.
- 57 x 57 픽셀 아이콘 - iPhone 및 iPod Touch 홈 화면에서 이 아이콘을 표시합니다.
- 72 x 72 픽셀 아이콘(선택 사항) - iPad 홈 화면에서 이 아이콘을 표시합니다.
- 512 x 512 픽셀 아이콘 - iTunes에서 이 아이콘을 표시합니다. 512픽셀 PNG 파일은 개발 버전의 응용 프로그램을 테스트 하는 데에만 사용됩니다. Apple App Store에 최종 응용 프로그램을 제출하는 경우 JPG 파일 형식의 512픽셀 이미지를 별도로 제출해야 합니다. IPA에 포함되지 않습니다.

Flash Professional CS5에서 [iPhone 설정] 대화 상자의 [아이콘] 탭에 이들 아이콘을 추가합니다. 자세한 내용은 14페이지의 [“Flash Professional CS5에서 iPhone 응용 프로그램 속성 설정”](#)을 참조하십시오.

응용 프로그램 설명자 파일에 아이콘의 위치를 추가할 수도 있습니다.

```
<icon>
  <image29x29>icons/icon29.png</image29x29>
  <image57x57>icons/icon57.png</image57x57>
  <image72x72>icons/icon72.png</image72x72>
  <image512x512>icons/icon512.png</image512x512>
</icon>
```

iPhone에서 자동으로 아이콘에 후광 효과가 추가되므로 원본 이미지에 후광 효과를 추가할 필요는 없습니다. 이 기본 후광 효과를 제거하려면 응용 프로그램 설명자 파일의 InfoAdditions 요소에 다음을 추가하십시오.

```
<InfoAdditions>
  <![CDATA[
    <key>UIPrerenderedIcon</key>
    <true/>
  ]]>
</InfoAdditions>
```

자세한 내용은 15페이지의 [“응용 프로그램 설명자 파일에서 iPhone 응용 프로그램 속성 설정”](#)을 참조하십시오.

초기 화면 아트(Default.png)

모든 iPhone 응용 프로그램은 iPhone에서 로드되는 동안 초기 이미지를 표시합니다. 이 초기 이미지는 Default.png라는 PNG 파일로 정의됩니다. 기본 개발 디렉토리에 Default.png라는 PNG 파일을 만듭니다. 이 파일을 하위 디렉토리에 넣지 마십시오. 파일 이름은 반드시 대문자 D로 시작하는 Default.png이어야 합니다.

응용 프로그램의 초기 방향이나 전체 화면인지 여부와 상관없이 Default.png 파일의 폭은 320픽셀이고 높이는 480픽셀입니다.

응용 프로그램의 초기 방향이 가로인 경우 세로 응용 프로그램에서 사용하는 것과 동일한 크기, 320픽셀(폭) x 480픽셀(높이)을 사용합니다. 그러나 PNG 파일에서는 아트워크를 시계 반대 방향으로 90° 회전합니다. PNG 아트의 왼쪽은 가로 모드의 iPhone 화면 위쪽에 해당합니다. 초기 응용 프로그램 방향을 설정하는 방법에 대한 자세한 내용은 13페이지의 “[iPhone 응용 프로그램 설정](#)”을 참조하십시오.

전체 화면이 아닌 응용 프로그램의 경우 기본 이미지 아트의 맨 위 20픽셀은 무시됩니다. iPhone에서는 기본 이미지 맨 위에 있는 폭 20픽셀의 사각형 위에 상태 표시줄이 표시됩니다. 가로 방향 응용 프로그램의 경우 이 영역은 Default.png 파일 왼쪽에 있는 폭 20픽셀의 사각형에 해당하며 가로 모드의 맨 위에 표시됩니다. 세로 방향 응용 프로그램의 경우 이 영역은 Default.png 파일 맨 위에 있는 폭 20픽셀의 사각형입니다.

대부분의 응용 프로그램에서 Default.png 이미지는 응용 프로그램의 시작 화면과 일치해야 합니다. 응용 프로그램 시작 화면의 스크린 샷을 생성하려면

1 iPhone에서 응용 프로그램을 엽니다. 사용자 인터페이스의 첫 화면이 나타나면 화면 아래에 있는 [홈] 버튼을 누르고 있습니다. [홈] 버튼을 누른 상태에서 장치의 맨 위에 있는 [잠자기/깨우기] 버튼을 누릅니다. 이렇게 하면 스크린 샷이 생성되어 카메라 롤로 보내집니다.

2 iPhoto 또는 기타 사진 전송 응용 프로그램에서 개발 컴퓨터로 이미지를 전송합니다. Mac OS에서는 이미지 캡처 응용 프로그램을 사용할 수도 있습니다.

전자 메일을 통해 사진을 개발 컴퓨터로 보낼 수도 있습니다.

- Photos 응용 프로그램을 엽니다.
- 카메라 롤을 엽니다.
- 캡처한 스크린 샷 이미지를 엽니다.
- 이미지를 누르고 왼쪽 아래에 있는 “전달”(화살표) 버튼을 누릅니다. 그런 다음 [이메일 메시지에 사진 보내기] 버튼을 눌러 이미지를 자신의 전자 메일 주소로 보냅니다.

참고: 크기만 올바르게 어떤 아트든 Default.png 파일로 만들 수 있습니다. 그러나 일반적으로 Default.png 이미지는 응용 프로그램의 초기 상태와 일치하는 것이 좋습니다.

응용 프로그램이 여러 가지 언어로 지역화되는 경우 Default.png 이미지에 텍스트를 포함하지 마십시오. Default.png는 불변하므로 텍스트가 다른 언어와 맞지 않게 됩니다.

Flash Professional CS5에서 [iPhone 설정] 대화 상자의 [포함된 파일] 목록에 Default.png 파일을 추가해야 합니다. 자세한 내용은 14페이지의 “[Flash Professional CS5에서 iPhone 응용 프로그램 속성 설정](#)”을 참조하십시오.

PFI 응용 프로그램을 사용하여 명령줄에서 컴파일하는 경우 포함된 예셋 목록에 있는 이 파일을 참조하는지 확인하십시오. 자세한 내용은 19페이지의 “[명령줄에서 iPhone 응용 프로그램 설치 프로그램 파일 만들기](#)”를 참조하십시오.

iPhone 응용 프로그램 설정

응용 프로그램 설정에는 다음이 포함됩니다.

- 응용 프로그램 이름
- IPA 파일 이름
- 응용 프로그램 버전

- 응용 프로그램의 초기 화면 방향 및 iPhone이 회전할 때 화면 방향이 자동으로 회전하는지 여부
- 초기 보기가 전체 화면인지 여부
- 응용 프로그램 아이콘에 대한 정보
- 하드웨어 가속에 대한 정보

Flash Professional CS5에서 응용 프로그램 설정을 편집할 수 있습니다.

또한 응용 프로그램 설명자 파일에서도 응용 프로그램 설정을 편집할 수 있습니다. 응용 프로그램 설명자 파일은 응용 프로그램에 대한 설정이 포함된 XML 파일입니다.

Flash Professional CS5에서 iPhone 응용 프로그램 속성 설정

Flash Professional CS5 [iPhone 설정] 대화 상자에서 iPhone 응용 프로그램의 여러 가지 기본 속성을 정의할 수 있습니다.

iPhone 설정 대화 상자를 열려면

- ❖ [파일] > [iPhone 설정]을 선택합니다.

일반 탭

[일반] 탭에는 다음 iPhone 관련 설정이 포함되어 있습니다.

- 출력 파일 - iPhone에서 응용 프로그램 아이콘 아래에 표시되는 응용 프로그램 이름입니다. 출력 파일 이름에 더하기 기호 (+)를 포함해서는 안 됩니다.
- 응용 프로그램 이름 - iPhone에서 응용 프로그램 아이콘 아래에 표시되는 응용 프로그램 이름입니다. 응용 프로그램 이름에 더하기 기호(+)를 포함해서는 안 됩니다.
- 버전 - 사용자에게 설치하는 응용 프로그램의 버전을 알려줍니다. 이 버전은 iPhone 응용 프로그램의 CFBundleVersion으로 사용되며, nnnnn[.nn[.nn]]과 유사한 형식이어야 합니다. 여기서 n은 0-9 사이의 숫자이고 대괄호는 1, 1.0 또는 1.0.1과 같이 선택적 구성 요소를 나타냅니다. iPhone 버전에는 숫자와 소수점만 사용해야 합니다. iPhone 버전에는 소수점 이하 두 자리 숫자까지 포함할 수 있습니다.
- 중횡비 - 응용 프로그램의 초기 중횡비(세로 또는 가로)입니다.
- 전체 화면 - 응용 프로그램이 전체 화면으로 표시되는지 아니면 iPhone 상태 표시줄로 표시되는지 여부를 설정합니다.
- 자동 방향 - iPhone의 방향을 조정할 때 응용 프로그램 표시 내용의 방향도 조정되도록 하려면 이 응용 프로그램을 선택합니다.

자동 방향 조정을 사용할 때 최적의 결과를 얻으려면 ActionScript 코드를 추가하여 Stage의 align 속성을 다음과 같이 설정합니다.

```
stage.align = StageAlign.TOP_LEFT;  
stage.scaleMode = StageScaleMode.NO_SCALE;
```

- 렌더링 - iPhone에서 객체를 렌더링하는 방법입니다.
 - CPU - 응용 프로그램에서 CPU를 사용하여 모든 객체를 렌더링합니다. 하드웨어 가속은 사용되지 않습니다.
 - GPU - 응용 프로그램에서 iPhone GPU를 사용하여 비트맵을 합성합니다.
 - 자동 - 이 기능은 아직 구현되지 않았습니다.

자세한 내용은 32페이지의 “하드웨어 가속”을 참조하십시오.

- 포함된 파일 - 모든 파일과 디렉토리를 iPhone 응용 프로그램의 패키지에 추가합니다. 기본 SWF 파일 및 응용 프로그램 설명자 파일은 기본적으로 포함됩니다. [포함된 파일] 목록에 필요한 다른 모든 에셋을 추가합니다. 초기 화면 아트 파일 (Default.png)도 [포함된 파일] 목록에 추가해야 합니다.

배포 탭

[배포] 탭에는 응용 프로그램에 대한 서명 및 컴파일 설정이 포함되어 있습니다.

- iPhone 디지털 서명 - 인증서의 P12 인증서 파일과 암호를 지정합니다. Apple iPhone 인증서는 .p12 포맷으로 변환해야 합니다. 자세한 내용은 4페이지의 “[Apple에서 개발자 파일 받기](#)”를 참조하십시오.
- 프로비저닝 파일 - Apple에서 받은 응용 프로그램의 프로비저닝 파일을 가리킵니다. 자세한 내용은 4페이지의 “[Apple에서 개발자 파일 받기](#)”를 참조하십시오.
- 응용 프로그램 ID - 응용 프로그램 ID는 응용 프로그램을 고유하게 식별합니다. 프로비저닝 파일이 특정 응용 프로그램 ID와 연결되어 있는 경우 Flash Professional CS5에서 이 필드를 설정하며 사용자는 이 필드를 편집할 수 없습니다. 그렇지 않은 경우에는 프로비저닝 프로파일에 여러(와일드카드) 응용 프로그램 ID를 사용할 수 있습니다. Apple에 제공한 응용 프로그램 ID 와일드카드 패턴과 일치하는 응용 프로그램 ID를 입력합니다.
- Apple 응용 프로그램 ID가 com.myDomain.*인 경우 [iPhone 설정] 대화 상자의 응용 프로그램 ID는 com.myDomain으로 시작해야 합니다(예: com.myDomain.myApp 또는 com.myDomain.app22).
- Apple 응용 프로그램 ID가 *인 경우 [iPhone 설정] 대화 상자의 응용 프로그램 ID는 유효한 문자로 구성된 임의의 문자열이 될 수 있습니다.

iPhone Dev Center(<http://developer.apple.com/iphone>)에서 프로비저닝 프로파일과 연결된 Apple 응용 프로그램 ID(또는 와일드카드 응용 프로그램 ID 패턴)를 확인할 수 있습니다. iPhone Developer Program Portal로 이동한 다음 [Provisioning] 섹션으로 이동합니다.

중요: Apple 응용 프로그램 ID 앞쪽에 있는 문자는 무시하십시오. Apple에서는 이 문자열을 번들 시드 ID라고 합니다. 예를 들어 Apple에서 응용 프로그램 ID를 96LPVWEASL.com.example.bob.myApp으로 표시하는 경우 96LPVWEASL을 무시하고 com.example.bob.myApp을 응용 프로그램 ID로 사용하십시오. Apple에서 응용 프로그램 ID를 5RM86Z4DJM.*로 표시하는 경우 5RM86Z4DJM을 무시하십시오. 이는 와일드카드 응용 프로그램 ID입니다.

- iPhone 배포 유형:
 - 장치 테스트용 빠른 제작 - 개발자 iPhone에서 특정 응용 프로그램 버전을 테스트용으로 빠르게 컴파일하려면 이 옵션을 선택합니다.
 - 장치 디버그용 빠른 제작 - 개발자 iPhone에서 테스트를 위해 응용 프로그램의 디버그 버전을 빠르게 컴파일하려면 이 옵션을 선택합니다. 이 옵션을 사용하면 Flash Professional CS5 디버거가 iPhone 응용 프로그램으로부터 trace() 출력을 수신할 수 있습니다. 자세한 내용은 22페이지의 “[iPhone 응용 프로그램 디버그](#)”를 참조하십시오.
 - 배포 - 애드혹 - 애드혹 배포용으로 응용 프로그램을 만들려면 이 옵션을 선택합니다. 자세한 내용은 Apple iPhone 개발자 센터를 참조하십시오.
 - 배포 - Apple App Store - Apple App Store에 배포할 IPA 파일의 최종 버전을 만들려면 이 옵션을 선택합니다.

아이콘 탭

[아이콘] 탭에서 29 x 29 픽셀 아이콘 이미지, 48 x 48 픽셀 아이콘 이미지, 57 x 57 픽셀 아이콘 이미지, 72 x 72 픽셀 아이콘 이미지 및 512 x 512 픽셀 아이콘 이미지의 위치를 지정합니다. 자세한 내용은 12페이지의 “[iPhone 아이콘 및 초기 화면 이미지](#)”를 참조하십시오.

참고: 48 x 48 픽셀 및 72 x 72 픽셀에 대한 옵션은 Flash Professional CS5에 포함된 Packager for iPhone Preview 버전에는 포함되어 있지 않습니다. Flash Professional CS5에서 [도움말] > [업데이트]를 선택하여 이러한 옵션을 추가합니다.

응용 프로그램 설명자 파일에서 iPhone 응용 프로그램 속성 설정

응용 프로그램 설명자 파일은 이름, 버전, 저작권 및 기타 설정을 비롯한 전체 응용 프로그램에 대한 속성이 포함된 XML 파일입니다.

Flash Professional CS5는 [iPhone 설정] 대화 상자의 설정에 따라 응용 프로그램 설명자 파일을 생성합니다. 또한 텍스트 편집기에서 응용 프로그램 설명자 파일을 편집할 수도 있습니다. Flash Professional은 프로젝트 이름에 "-app.xml"을 추가하여 응용 프로그램 설명자 파일의 이름을 지정합니다. 예를 들어 HelloWorld 프로젝트의 응용 프로그램 설명자 파일 이름은 HelloWorld-app.xml이 됩니다. Flash Professional CS5 iPhone 설정 대화 상자에서 지원되지 않는 설정을 정의하려면 응용 프로그램 설명자 파일을 편집하면 됩니다. 예를 들어 InfoAdditions 요소를 정의하여 응용 프로그램의 info.Plist 설정을 정의할 수 있습니다.

중요: Flash Professional CS5 대화 상자가 열려 있는 동안 응용 프로그램 설명자 파일을 편집하지 마십시오. [iPhone 설정] 대화 상자를 열기 전에 응용 프로그램 설명자 파일에 대한 변경 내용을 저장하십시오.

다음은 응용 프로그램 설명자 파일의 예제입니다.

```
<?xml version="1.0" encoding="UTF-8"?>
<application xmlns="http://ns.adobe.com/air/application/2.0">
  <id>com.example.HelloWorld</id>
  <filename>HelloWorld</filename>
  <name>Hello World</name>
  <version>v1</version>
  <initialWindow>
    <renderMode>gpu</renderMode>
    <content>HelloWorld.swf</content>
    <fullScreen>true</fullScreen>
    <aspectRatio>portrait</aspectRatio>
    <autoOrients>true</autoOrients>
  </initialWindow>
  <supportedProfiles>mobileDevice desktop</supportedProfiles>
  <icon>
    <image29x29>icons/icon29.png</image29x29>
    <image57x57>icons/icon57.png</image57x57>
    <image512x512>icons/icon512.png</image512x512>
  </icon>
  <iPhone>
    <InfoAdditions>
      <![CDATA[
        <key>UIStatusBarStyle</key>
        <string>UIStatusBarStyleBlackOpaque</string>
        <key>UIRequiresPersistentWiFi</key>
        <string>NO</string>
      ]]>
    </InfoAdditions>
  </iPhone>
</application>
```

다음은 이 응용 프로그램 설명자 파일에 대한 자세한 설명입니다.

- iPhone 응용 프로그램을 만들려면 <application> 요소에 AIR 2.0 네임스페이스가 필요합니다.

<application xmlns="http://ns.adobe.com/air/application/2.0">

- <id> 요소:

<id>com.example.as3.HelloWorld</id> 응용 프로그램 ID는 응용 프로그램을 고유하게 식별합니다.

"com.company.AppName"과 같이 점으로 구분된 반전된 DNS-스타일 문자열을 사용할 수 있습니다. 컴파일러는 이 값을 iPhone 응용 프로그램의 번들 ID로 사용합니다.

프로비저닝 파일이 특정 응용 프로그램 ID와 연결되어 있는 경우 이 요소에서 해당 응용 프로그램 ID를 사용합니다. Apple에서 Apple 응용 프로그램 ID의 시작 부분에 할당하는 문자(번들 시드 ID라고 함)는 무시하십시오. 예를 들어 프로비저닝 프 로파일의 응용 프로그램 ID가 96LPVWEASL.com.example.bob.myApp인 경우 com.example.bob.myApp을 응용 프 로그램 설명자 파일의 응용 프로그램 ID로 사용합니다.

프로비저닝 프로파일에서 여러(와일드카드) 응용 프로그램 ID를 허용하는 경우 응용 프로그램 ID는 별표로 끝납니다(예: 5RM86Z4DJM.*). Apple에 제공한 응용 프로그램 ID 와일드카드 패턴과 일치하는 응용 프로그램 ID를 입력합니다.

- Apple 응용 프로그램 ID가 com.myDomain.*인 경우 응용 프로그램 설명자 파일의 응용 프로그램 ID는 com.myDomain으로 시작해야 합니다. com.myDomain.myApp 또는 com.myDomain.app22와 같은 응용 프로그램 ID를 지정할 수 있습니다.
- Apple 응용 프로그램 ID가 *인 경우 응용 프로그램 설명자 파일의 응용 프로그램 ID는 유효한 문자로 구성된 임의의 문자열이 될 수 있습니다.

iPhone Dev Center(<http://developer.apple.com/iphone>)에서 프로비저닝 프로파일과 연결된 Apple 응용 프로그램 ID(또는 와일드카드 응용 프로그램 ID 패턴)를 확인할 수 있습니다. iPhone Developer Program Portal로 이동한 다음 [Provisioning] 섹션으로 이동합니다.

중요: Apple 응용 프로그램 ID 앞쪽에 있는 문자는 무시하십시오. Apple에서는 이 문자열을 번들 시드 ID라고 합니다. 예를 들어 Apple에서 응용 프로그램 ID를 5RM86Z4DJM.*로 표시하는 경우 5RM86Z4DJM을 무시하십시오. 이는 와일드카드 응용 프로그램 ID입니다. Apple에서 응용 프로그램 ID를 96LPVWEASL.com.example.bob.myApp으로 표시하는 경우 96LPVWEASL을 무시하고 com.example.bob.myApp을 응용 프로그램 ID로 사용합니다.

- <filename> 요소:

<filename>HelloWorld</filename> iPhone 설치 프로그램 파일에 사용되는 이름입니다. 파일 이름에 더하기 기호(+)를 포함해서는 안 됩니다.

- <name> 요소:

<name>Hello World</name> iTunes 응용 프로그램과 iPhone에 표시되는 응용 프로그램 이름입니다. 이름에 더하기 기호(+)를 포함해서는 안 됩니다.

- <version> 요소:

<version>1.0</version> 사용자에게 설치하는 응용 프로그램의 버전을 알려줍니다. 이 버전은 iPhone 응용 프로그램의 CFBundleVersion으로 사용되며, nnnnn[.nn[.nn]]과 유사한 형식이어야 합니다. 여기서 n은 0-9 사이의 숫자이고 대괄호는 1, 1.0 또는 1.0.1과 같이 선택적 구성 요소를 나타냅니다. iPhone 버전에는 숫자와 소수점만 사용해야 합니다. iPhone 버전에는 소수점 이하 두 자리 숫자까지 포함할 수 있습니다.

- <initialWindow> 요소에는 응용 프로그램의 초기 모양에 대한 속성을 지정하는 다음 자식 요소가 포함되어 있습니다.

<content>HelloWorld.swf</content> iPhone 응용 프로그램으로 컴파일할 루트 SWF 파일을 식별합니다.

<visible>true</visible> 필수 설정입니다.

<fullScreen>true</fullScreen> 응용 프로그램이 iPhone의 전체 화면을 사용하도록 지정합니다.

<aspectRatio>portrait</aspectRatio> 응용 프로그램의 초기 종횡비를 가로 모드가 아닌 세로 모드로 지정합니다. 응용 프로그램의 초기 윈도우를 정의하는 데 사용되는 Default.png 파일은 이 설정에 관계없이 폭 320픽셀, 높이 480픽셀이어야 합니다. 자세한 내용은 12페이지의 “iPhone 아이콘 및 초기 화면 이미지”를 참조하십시오.

<autoOrients>true</autoOrients> (선택 사항) 장치 자체의 실제 방향이 변경되는 경우 응용 프로그램 내 내용의 방향이 자동으로 조정되도록 할지 여부를 지정합니다. 기본값은 true입니다. 자동 방향은 orientationChanging 이벤트의 preventDefault() 메서드를 호출하여 취소할 수 있습니다. 이 메서드는 Stage 객체에서 전달됩니다. 자세한 내용은 [화면 방향 설정 및 감지](#)를 참조하십시오.

자동 방향 조정을 사용할 때 최적의 결과를 얻으려면 Stage의 align 속성을 다음과 같이 설정합니다.

```
stage.align = StageAlign.TOP_LEFT;
stage.scaleMode = StageScaleMode.NO_SCALE;
```

<renderMode>gpu</renderMode> (선택 사항) 응용 프로그램에 사용되는 렌더링 모드입니다. 세 가지 설정 중에서 선택할 수 있습니다.

- cpu - CPU를 사용하여 응용 프로그램의 모든 객체를 렌더링합니다. 하드웨어 가속은 사용되지 않습니다.

- gpu - 응용 프로그램에서 iPhone GPU를 사용하여 비트맵을 합성합니다.
- auto - 이 기능은 아직 구현되지 않았습니다.

자세한 내용은 32페이지의 “하드웨어 가속”을 참조하십시오.

- <profiles> 요소:

<profiles>mobileDevice</profiles> 응용 프로그램이 휴대 장치 프로파일로 컴파일되도록 제한합니다. 이 프로파일은 현재 iPhone 응용 프로그램만 지원합니다. 다음 세 가지 프로파일이 지원됩니다.

- desktop - 데스크톱 AIR 응용 프로그램
- extendedDesktop - 기본 프로세스 API를 지원하는 데스크톱 AIR 응용 프로그램
- mobileDevice - 휴대 장치용 AIR 응용 프로그램. 현재 휴대 장치는 iPhone만 지원됩니다.

응용 프로그램을 특정 프로파일로 제한하면 다른 프로파일로는 컴파일되지 않습니다. 프로파일을 지정하지 않은 경우에는 응용 프로그램을 세 가지 프로파일 중 원하는 것으로 컴파일할 수 있습니다. 둘 이상의 프로파일을 각각 공백으로 구분하여 <profiles> 요소에 나열하는 방법으로 복수의 프로파일을 지정할 수 있습니다.

mobileDevice를 지원되는 프로파일에 반드시 포함해야 합니다(또는 <profiles> 요소를 비워둠).

- <icon> 요소에는 응용 프로그램에 사용되는 아이콘을 지정하는 다음 자식 요소가 포함되어 있습니다.

<image29x29>icons/icon29.png</image29x29> Spotlight 검색 결과에 사용되는 이미지입니다.

<image48x48>icons/icon48.png</image48x48> iPad의 Spotlight 검색 결과에 사용되는 이미지입니다.

<image57x57>icons/icon57.png</image57x57> iPhone 및 iPod Touch 홈 화면에 사용되는 이미지입니다.

<image72x72>icons/icon72.png</image72x72> iPad 홈 화면에 사용되는 이미지입니다.

<image512x512>icons/icon512.png</image512x512> iTunes 응용 프로그램에 사용되는 이미지입니다.

Packager for iPhone 도구는 응용 프로그램 설명자 파일에서 참조되는 29, 57 및 512 아이콘을 사용합니다. 해당 아이콘을 각각 Icon-Small.png, Icon.png 및 iTunesArtwork라는 파일에 복사합니다. 파일을 직접 패키지화하면 파일을 각각 복사하는 번거로움을 덜 수 있습니다. 응용 프로그램 설명자 파일이 들어 있는 디렉토리에 파일을 넣고 올바른 이름과 경로를 나열하여 패키지화합니다.

512 이미지는 내부 테스트용으로만 사용됩니다. 응용 프로그램을 Apple로 전송할 때 512 이미지는 별도로 전송합니다. 이 이미지는 IPA에 포함되지 않습니다. 512 이미지가 iTunes에서 잘 표시되도록 전송하기 전에 이 이미지를 지정하십시오.

- <iPhone> 요소에는 iPhone 고유 설정을 지정하는 다음 자식 요소가 포함되어 있습니다.

<InfoAdditions></InfoAdditions>에는 응용 프로그램의 Info.plist 설정으로 사용할 키-값 쌍을 지정하는 자식 요소가 포함됩니다.

```
<![CDATA[
  <key>UIStatusBarStyle</key>
  <string>UIStatusBarStyleBlackOpaque</string>
  <key>UIRequiresPersistentWiFi</key>
  <string>NO</string>
]]>
```

이 예제에서 값은 응용 프로그램의 상태 표시줄 스타일을 설정하며, 응용 프로그램에 지속적인 Wi-Fi 액세스가 필요하지 않음을 나타냅니다.

InfoAdditions 설정은 CDATA 태그 안에 있습니다.

iPad 지원을 위해 UIDeviceFamily에 대한 키 값 설정을 포함합니다. UIDeviceFamily 설정은 문자열 배열입니다. 각 문자열은 지원되는 장치를 정의합니다. <string>1</string> 설정은 iPhone 및 iPod Touch에 대한 지원을 정의합니다.

<string>2</string> 설정은 iPad에 대한 지원을 정의합니다. 이러한 문자열 중 하나만 지정하는 경우 해당 장치 제품군만 지원됩니다. 예를 들어 다음 설정은 iPad로 지원을 제한합니다.

```
<key>UIDeviceFamily</key>
  <array>
    <string>2</string>
  </array>>
```

다음은 두 가지 장치 제품군(iPhone/iPod Touch 및 iPad) 모두에 대해 지원을 설정합니다.

```
<key>UIDeviceFamily</key>
<array>
  <string>1</string>
  <string>2</string>
</array>
```

다른 Info.plist 설정에 대한 자세한 내용은 Apple 개발자 설명서를 참조하십시오.

iPhone 응용 프로그램 설치 프로그램(IPA) 파일 컴파일

ActionScript 3.0 응용 프로그램을 IPA 설치 프로그램 파일로 컴파일하려면 Packager for iPhone을 사용합니다.

Flash Professional CS5에 포함된 Packager for iPhone을 사용하여 iPhone 응용 프로그램 설치 프로그램 파일 만들기

Flash Professional CS5에 포함된 Packager for iPhone을 사용하려면 다음 작업을 수행합니다.

- 1 [파일] > [제작]을 선택합니다.
- 2 [iPhone 설정] 대화 상자에서 모든 설정 값을 지정했는지 확인합니다. [배포] 탭에서 올바른 옵션을 선택했는지 확인합니다. 자세한 내용은 14페이지의 “Flash Professional CS5에서 iPhone 응용 프로그램 속성 설정”을 참조하십시오.
- 3 [제작] 버튼을 클릭합니다.

Packager for iPhone에서 IPA(iPhone 응용 프로그램 설치 프로그램) 파일이 생성됩니다. IPA 파일을 컴파일하는 데는 몇 분 정도 걸릴 수 있습니다.

명령줄에서 Packager for iPhone을 실행할 수도 있습니다. 자세한 내용은 19페이지의 “명령줄에서 iPhone 응용 프로그램 설치 프로그램 파일 만들기”를 참조하십시오.

명령줄에서 iPhone 응용 프로그램 설치 프로그램 파일 만들기

명령줄에서 Packager for iPhone을 실행할 수 있습니다. Packager for iPhone은 SWF 파일 바이트 코드와 기타 소스 파일을 기본 iPhone 응용 프로그램으로 변환합니다.

- 1 명령줄 셸 또는 터미널을 열고 iPhone 응용 프로그램의 프로젝트 폴더로 이동합니다.
- 2 다음으로, PFI 도구에서 다음 구문을 사용하여 IPA 파일을 만듭니다.

```
pfi -package -target [ipa-test ipa-debug ipa-app-store ipa-ad-hoc] -provisioning-profile PROFILE_PATH  
SIGNING_OPTIONS TARGET_IPA_FILE APP_DESCRIPTOR SOURCE_FILES
```

PFI 응용 프로그램의 전체 경로를 포함하도록 참조 pfi를 변경합니다. PFI 응용 프로그램은 Flash Professional CS5 설치 디렉토리의 pfi/bin 하위 디렉토리에 설치됩니다.

만들려는 iPhone 응용 프로그램 유형에 해당하는 -target 옵션을 선택합니다.

- -target ipa-test - 응용 프로그램의 특정 버전을 개발자 iPhone에서 테스트용으로 빠르게 컴파일하려면 이 옵션을 선택합니다.
- -target ipa-debug - 응용 프로그램의 디버그 버전을 개발자 iPhone에서 테스트용으로 컴파일하려면 이 옵션을 선택합니다. 이 옵션을 선택하면 디버그 세션을 사용하여 iPhone 응용 프로그램에서 trace() 출력을 수신할 수 있습니다.

다음 -connect 옵션(CONNECT_OPTIONS) 중 하나를 포함하여 디버거를 실행하는 개발 컴퓨터의 IP 주소를 지정할 수 있습니다.

- -connect - 응용 프로그램에서 응용 프로그램을 컴파일하는 데 사용되는 개발 컴퓨터의 디버그 세션에 연결하려고 합니다.
- -connect IP_ADDRESS - 응용 프로그램에서 지정된 IP 주소를 사용하는 컴퓨터의 디버그 세션에 연결하려고 합니다. 예를 들면 다음과 같습니다.

```
-target ipa-debug -connect 192.0.32.10
```

- -connect HOST_NAME - 응용 프로그램에서 지정된 호스트 이름을 사용하는 컴퓨터의 디버그 세션에 연결하려고 합니다. 예를 들면 다음과 같습니다.

```
-target ipa-debug -connect bobroberts-mac.example.com
```

참고: -connect 옵션은 Flash Professional CS5에 포함된 Packager for iPhone Preview에 포함되어 있지 않습니다. Flash Professional CS5에서 [도움말] > [업데이트]를 선택하여 Packager for iPhone을 업데이트합니다.

-connect 옵션은 선택 사항입니다. 지정되지 않은 경우, 결과 디버그 응용 프로그램에서 호스팅되는 디버거에 연결하려고 하지 않습니다.

디버그 연결이 실패하는 경우 디버깅 호스트 컴퓨터의 IP 주소를 입력하라는 메시지가 사용자에게 표시됩니다. 장치가 WiFi에 연결되어 있지 않은 경우 연결이 실패할 수 있습니다. 장치가 연결되어 있지만 디버깅 호스트 컴퓨터의 방화벽 내에 있지 않은 경우에도 이러한 문제가 발생할 수 있습니다.

자세한 내용은 22페이지의 “iPhone 응용 프로그램 디버그”를 참조하십시오.

-renderingdiagnostics 옵션을 포함하여 GPU 렌더링 진단 기능을 활성화할 수도 있습니다. 자세한 내용은 22페이지의 “iPhone 응용 프로그램 디버그”의 “GPU 렌더링 진단을 사용하여 디버그”를 참조하십시오.

- -target ipa-ad-hoc - 애플리케이션 배포용 응용 프로그램을 만들려면 이 옵션을 선택합니다. 자세한 내용은 Apple iPhone 개발자 센터를 참조하십시오.
- -target ipa-app-store - Apple App Store에 배포할 IPA 파일의 최종 버전을 만들려면 이 옵션을 선택합니다.

PROFILE_PATH를 응용 프로그램 프로비저닝 프로파일 파일의 경로로 바꿉니다. 프로비저닝 프로파일에 대한 자세한 내용은 4페이지의 “Apple에서 개발자 파일 받기”를 참조하십시오.

iPhone 개발자 인증서와 암호를 참조하도록 SIGNING_OPTIONS를 바꿉니다. 다음 구문을 사용합니다.

```
-storetype pkcs12 -keystore P12_FILE_PATH -storepass PASSWORD
```

P12_FILE_PATH를 P12 인증서 파일의 경로로 바꿉니다. PASSWORD를 인증서 암호로 바꿉니다. (다음 예제를 참조하십시오.) P12 인증서 파일에 대한 자세한 내용은 6페이지의 “개발자 인증서를 P12 파일로 변환”을 참조하십시오.

응용 프로그램 설명자 파일을 참조하도록 APP_DESCRIPTOR를 바꿉니다.

프로젝트의 기본 SWF 파일과 포함할 다른 예제를 참조하도록 SOURCE_FILES를 바꿉니다. 사용자 정의 응용 프로그램 설명자 파일 또는 Flash CS5의 응용 프로그램 설정 대화 상자에서 정의한 모든 아이콘 파일의 경로를 포함합니다. 또한 초기 화면 아트 파일 Default.png도 추가합니다.

다음 예를 참조하십시오.

```
pfi -package -target ipa-test -storetype pkcs12 -keystore  
"/Users/Jeff/iPhoneCerts/iPhoneDeveloper_Jeff.p12" -storepass dfb7VKL19 "HelloWorld.ipa" "HelloWorld-  
app.xml" "HelloWorld.swf" "Default.png" "icons/icon29.png" "icons/icon57.png" "icons/icon512.png"
```

이 명령문은 다음을 사용하여 HelloWorld.ipa 파일을 컴파일합니다.

- dfb7VKL19라는 인증서 암호를 사용하는 특정 PKCS#12 인증서
- HelloWorld-app.xml 응용 프로그램 설명자 파일
- 소스 HelloWorld.swf 파일

- 특정 Default.png 및 아이콘 파일

pfi 응용 프로그램이 응용 프로그램 설명자 파일, SWF 파일 및 기타 에셋을 기반으로 응용 프로그램을 IPA 파일로 컴파일합니다.

Mac OS에서는 pfi 명령에 다음 옵션을 추가하여 키체인에 저장된 인증서를 사용할 수 있습니다.

```
-alias ALIAS_NAME -storetype KeychainStore -providerName Apple
```

사용할 인증서의 별칭으로 ALIAS_NAME을 바꿉니다. Mac 키체인에 저장된 인증서를 가리키는 경우 인증서 파일 위치를 가리키는 대신에 이 별칭을 지정하면 됩니다.

iPhone 응용 프로그램 설치

iPhone에 개발 응용 프로그램을 설치하려면 iPhone에 프로비저닝 프로파일을 추가한 다음 iPhone에 응용 프로그램을 설치합니다.

iPhone에 프로비저닝 프로파일 추가

iPhone에 프로비저닝 프로파일을 추가하려면

- 1 iTunes에서 [파일] > [보관함에 추가]를 선택합니다. 프로비저닝 프로파일 파일(파일 이름 확장명이 mobileprovision인 파일)을 선택합니다.

iPhone이 프로비저닝 프로파일에 추가되었는지 확인합니다. Apple iPhone Dev Center 사이트

(<http://developer.apple.com/iphone/>)에서 프로비저닝 프로파일을 관리할 수 있습니다. 사이트의 [iPhone Developer Program Portal] 섹션으로 이동합니다. 개발 응용 프로그램을 설치할 수 있는 장치 목록을 관리하려면 [Devices] 링크를 클릭합니다. 프로비저닝 프로파일을 관리하려면 [Provisioning] 링크를 클릭합니다.

- 2 iPhone을 컴퓨터에 연결하여 동기화합니다.

프로비저닝 프로파일을 받는 방법에 대한 자세한 내용은 4페이지의 “Apple에서 개발자 파일 받기”를 참조하십시오.

응용 프로그램 설치

다른 모든 IPA 파일을 설치하는 것과 동일한 방식으로 개발 응용 프로그램을 설치합니다.

- 1 해당 응용 프로그램을 이전에 설치한 경우 장치 및 iTunes의 응용 프로그램 목록에서 해당 응용 프로그램을 삭제합니다.
- 2 다음과 같은 방법 중 하나로 iTunes에 응용 프로그램을 추가합니다.
 - iTunes의 [파일] 메뉴에서 [보관함에 추가] 명령을 선택합니다. 그런 다음 IPA 파일을 선택하고 [열기] 버튼을 클릭합니다.
 - IPA 파일을 두 번 클릭합니다.
 - IPA 파일을 iTunes 보관함에 드래그합니다.
- 3 iPhone을 컴퓨터의 USB 포트에 연결합니다.
- 4 iTunes에서 장치의 [응용 프로그램] 탭에 있는 설치할 응용 프로그램 목록에 응용 프로그램이 선택되어 있는지 확인합니다.
- 5 iPhone을 동기화합니다.

응용 프로그램 설치 문제 해결

응용 프로그램을 설치하려고 할 때 iTunes에서 오류 메시지가 표시되는 경우 다음 사항을 확인하십시오.

- 장치 ID가 프로비저닝 프로파일에 추가되었는지 확인합니다.
- 프로비저닝 프로파일을 설치하려면 프로비저닝 프로파일을 iTunes로 드래그하거나 [파일] > [보관함에 추가] 명령을 사용합니다.

또한 응용 프로그램의 응용 프로그램 ID가 Apple 응용 프로그램 ID와 일치하는지 확인하십시오.

- Apple 응용 프로그램 ID가 com.myDomain.*인 경우 응용 프로그램 설명자 파일의 응용 프로그램 ID 또는 [iPhone 설정] 대화 상자의 응용 프로그램 ID가 com.myDomain으로 시작해야 합니다(예: com.myDomain.anythinghere).
- Apple 응용 프로그램 ID가 com.myDomain.myApp인 경우 응용 프로그램 설명자 파일 또는 Flash Professional CS5 사용자 인터페이스의 응용 프로그램 ID는 com.myDomain.myApp이어야 합니다.
- Apple 응용 프로그램 ID가 *인 경우 응용 프로그램 설명자 파일 또는 Flash Professional CS5 사용자 인터페이스의 응용 프로그램 ID는 무엇이든 될 수 있습니다.

응용 프로그램 설명자 파일 또는 Flash Professional CS5의 [iPhone 설정] 대화 상자에서 응용 프로그램의 응용 프로그램 ID를 설정합니다.

iPhone 응용 프로그램 디버그

개발 컴퓨터에서 ADL로 실행되는 응용 프로그램을 디버그할 수 있습니다. 또한 iPhone에서도 응용 프로그램을 디버그할 수 있습니다.

일부 AIR 기능은 iPhone에서 지원되지 않지만 개발 컴퓨터에서 ADL로 응용 프로그램을 테스트할 때는 사용할 수 있습니다. 데스크톱에서 내용을 테스트할 때 이러한 차이에 유의해야 합니다. 자세한 내용은 25페이지의 “[휴대 장치에서 지원되지 않는 ActionScript 3.0 API](#)”를 참조하십시오.

개발 컴퓨터에서 응용 프로그램 디버그

Flash Professional CS5를 사용하여 개발 컴퓨터에서 응용 프로그램을 디버그하려면 다음 작업을 수행합니다.

- ❖ [디버그] > [동영상 디버그] > [AIR Debug Launcher(모바일)]를 선택합니다.

또한 명령줄에서 ADL을 호출하여 응용 프로그램을 디버그할 수도 있습니다. 다음 구문을 사용합니다.

```
adl -profile mobileDevice appDescriptorFile
```

appDescriptorFile을 응용 프로그램 설명자 파일 경로로 바꿉니다.

-profile mobileDevice 옵션을 포함했는지 확인하십시오.

iPhone에서 응용 프로그램 디버그

iPhone에서 응용 프로그램을 디버그하려면

1 디버그 지원을 받아 응용 프로그램을 컴파일합니다.

- Flash Professional CS5에서 "장치 디버깅용 빠른 제작" 설정을 사용하여 컴파일합니다. 자세한 내용은 19페이지의 “[Flash Professional CS5에 포함된 Packager for iPhone을 사용하여 iPhone 응용 프로그램 설치 프로그램 파일 만들기](#)”를 참조하십시오.
- PFI 명령줄 응용 프로그램에서 target ipa-debug 옵션을 사용하여 응용 프로그램을 컴파일합니다. 자세한 내용은 19페이지의 “[명령줄에서 iPhone 응용 프로그램 설치 프로그램 파일 만들기](#)”를 참조하십시오.

2 iPhone에 응용 프로그램을 설치합니다.

3 iPhone에서 Wi-Fi를 켜고 개발 컴퓨터와 동일한 네트워크에 연결합니다.

4 개발용 컴퓨터에서 디버그 세션을 시작합니다. Flash Professional CS5에서 [디버그] > [원격 디버그 세션 시작] > [ActionScript 3.0]을 선택합니다.

5 iPhone에서 응용 프로그램을 실행합니다.

응용 프로그램의 디버그 버전에서 개발자 컴퓨터의 IP 주소를 묻는 메시지가 나타납니다. IP 주소를 입력하고 [확인] 버튼을 누릅니다. 개발용 컴퓨터의 IP 주소를 확인하려면 다음과 같이 합니다.

- Mac OS의 경우 [Apple] 메뉴에서 [시스템 환경 설정]을 선택합니다. [시스템 환경 설정] 윈도우에서 [네트워크] 아이콘을 클릭합니다. [네트워크] 환경 설정 윈도우에 IP 주소 목록이 표시됩니다.
- Windows의 경우 명령줄 세션을 시작하고 ipconfig 명령을 실행합니다.

디버그 세션에 응용 프로그램의 모든 trace() 출력이 표시됩니다.

iPhone에 설치된 응용 프로그램을 디버그할 때 Flash Professional CS5에서는 중단점 제어, 코드 단계별 확인, 변수 모니터링 등 모든 디버그 기능을 지원합니다.

GPU 렌더링 진단을 사용하여 디버그

GPU 렌더링 진단 기능을 사용하면 응용 프로그램에서 하드웨어 가속을 사용하는 방식을 확인할 수 있습니다(GPU 렌더링 모드를 사용하는 응용 프로그램의 경우). 이 기능을 사용하려면 명령줄에서 PFI 도구를 사용하여 응용 프로그램을 컴파일하고 -renderingdiagnostics 옵션을 포함합니다.

```
pfi -package -renderingdiagnostics -target ipa-debug -connect ...
```

-renderingdiagnostics 플래그는 -package 플래그 바로 뒤에 와야 합니다.

GPU 렌더링 진단 기능은 모든 표시 객체에 대해 색상이 지정된 사각형을 표시합니다.

- 파랑 - 표시 객체가 비트맵이거나 비트맵으로 캐시되지 않으며, 렌더링되고 있습니다.

변경되지 않는 표시 객체에 대해 파랑이 반복적으로 표시되는 경우 해당 표시 객체가 이동하는 표시 객체와 교차하기 때문일 수 있습니다. 예를 들어 해당 표시 객체가 이동하는 표시 객체의 배경일 수 있습니다. 해당 표시 객체를 비트맵으로 캐시하는 것을 고려하십시오.

캐시되어야 한다고 생각되는 객체에 대해 파랑이 표시되는 경우 해당 객체에서 GPU가 적용할 수 없는 효과를 사용하기 때문일 수 있습니다. 이러한 효과에는 일부 블렌드 모드, 색상 변형, scrollRect 속성 및 마스크가 포함됩니다.

GPU에 업로드된 표시 객체가 메모리 제한을 초과하는 경우에도 응용 프로그램에서 파랑을 표시합니다.

응용 프로그램에서 각 파랑 사각형에 대해 메시지를 기록합니다. 응용 프로그램에서 이러한 메시지를 다른 trace() 및 디버그 출력 메시지와 함께 출력합니다.

- 녹색 - 표시 객체가 비트맵이거나 비트맵으로 캐시되며, 처음으로 GPU에 업로드되고 있습니다.

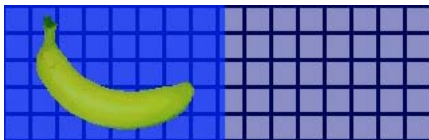
표시 객체에 대해 녹색이 반복적으로 표시되면 응용 프로그램의 코드에서 해당 표시 객체를 다시 만들고 있는 것입니다. 예를 들어 타임라인이 표시 객체를 만드는 프레임으로 돌아가는 경우 이러한 상황이 발생할 수 있습니다. 동일한 객체를 다시 만들지 않도록 내용을 수정하는 것을 고려하십시오.

- 빨강 - 표시 객체가 비트맵이거나 비트맵으로 캐시되며, GPU에 다시 업로드되고 있습니다.

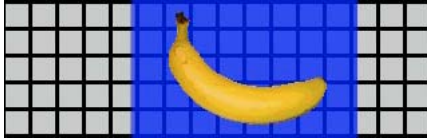
이러한 표시 객체가 응용 프로그램에서 비트맵 표현을 다시 렌더링해야 하는 방식으로 변경될 때마다 빨강이 표시됩니다. 예를 들어 cacheAsBitmapMatrix 속성이 설정되지 않은 2D 객체인 경우 크기 조절 또는 회전 시 다시 렌더링됩니다. 자식 표시 객체가 이동하거나 변경되는 경우에도 다시 렌더링됩니다.

해당 주기 동안 색상 지정의 이유가 다시 발생하지 않는 경우, 색상이 지정된 각 사각형은 4가지 화면 다시 그리기 주기 후 페이드됩니다. 그러나 화면에서 변경 사항이 없는 경우 진단 색상 지정이 변경되지 않습니다.

예를 들어 비트맵으로 캐시되지 않는 벡터 배경 앞의 비트맵 표시 객체(바나나)를 생각해 보겠습니다. 바나나가 먼저 렌더링되면 녹색으로 색상이 지정됩니다. 배경이 먼저 렌더링되면 파랑으로 색상이 지정됩니다.

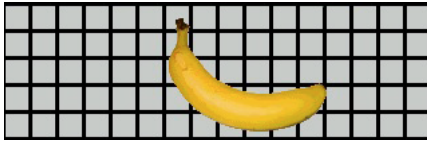


바나나가 이동함에 따라 CPU에서 배경을 다시 렌더링해야 하며, 이로 인해 배경 위에 파랑 음영이 나타납니다.



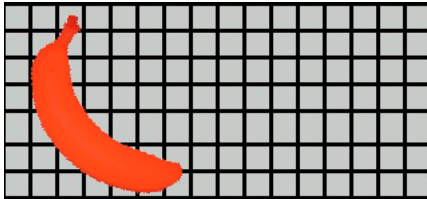
배경 위의 파랑 음영은 GPU에 전송해야 할 다시 그려진 영역을 반영합니다.

그러나 배경이 비트맵으로 캐시되는 경우 바나나가 이동하면 렌더링 진단 기능이 색상이 지정된 농도를 표시하지 않습니다.



GPU에서 배경 비트맵이 유지되므로 진단 기능이 색상이 지정된 농도를 표시하지 않습니다. GPU는 CPU를 사용하지 않고 배경과 바나나를 합성할 수 있습니다.

바나나가 `cacheAsBitmapMatrix` 속성이 설정되지 않은 2D 표시 객체라고 가정해 보겠습니다. 표시 객체가 회전(또는 크기 조절)될 때마다 렌더링 진단 기능이 빨간색을 표시합니다. 이는 응용 프로그램이 표시 객체의 새 버전을 GPU에 업로드해야 함을 나타냅니다.



App Store에 iPhone 응용 프로그램 제출

App Store에 응용 프로그램을 제출하려면 다음 작업을 수행합니다.

- 1 iPhone Dev Center 웹 사이트(<http://developer.apple.com/iphone/>)에서 배포 인증서 및 프로비저닝 프로파일을 받습니다.
배포 인증서 이름은 "iPhone Developer: XXX.cer"로 지정됩니다. 여기서 XXX는 개발자의 이름입니다.
자세한 내용은 4페이지의 "[Apple에서 개발자 파일 받기](#)"를 참조하십시오.
- 2 배포 인증서를 P12 파일로 변환합니다.
자세한 내용은 6페이지의 "[개발자 인증서를 P12 파일로 변환](#)"을 참조하십시오.
- 3 P12 파일 및 프로비저닝 프로파일을 사용하여 응용 프로그램을 컴파일합니다.
배포 인증서를 기반으로 만든 P12 파일을 사용하고, 배포 프로비저닝 프로파일과 연결된 응용 프로그램 ID를 사용합니다.
자세한 내용은 19페이지의 "[iPhone 응용 프로그램 설치 프로그램\(IPA\) 파일 컴파일](#)"을 참조하십시오.
- 4 설치 프로그램 파일의 파일 유형을 .ipa에서 .zip으로 변경합니다. 예를 들어 MyApp.ipa를 MyApp.zip으로 변경합니다.
- 5 iPhone Dev Center 웹 사이트(<http://developer.apple.com/iphone/>)에서 응용 프로그램을 제출합니다.

3장: ActionScript 3.0 API의 휴대 장치 지원

Flash Player 10.1 및 AIR 2 데스크톱 응용 프로그램 개발에 사용되는 것과 같은 ActionScript 3.0 API를 사용하여 휴대 장치용 AIR 응용 프로그램을 만들 수 있습니다. 단, 몇 가지 예외 및 추가 사항이 있습니다.

휴대 장치에서 지원되지 않는 ActionScript 3.0 API

일부 ActionScript 3.0 API는 휴대 장치 프로파일로 실행되는 응용 프로그램(예: iPhone에서 실행되는 응용 프로그램)에 사용할 수 없습니다.

동일한 ActionScript 코드를 여러 프로파일(예: 데스크톱 및 모바일) 개발에 사용하는 경우 API 지원 여부를 테스트하는 코드를 사용하십시오. 예를 들어 NativeWindow 클래스는 iPhone 응용 프로그램에서 지원되지 않습니다. iPhone 응용 프로그램은 기본 윈도우를 사용하거나 만들 수 없습니다. 특정 프로파일(예: 데스크톱 프로파일)로 실행되는 응용 프로그램에서 기본 윈도우가 지원되는지 여부를 테스트하려면 NativeWindow.isSupported 속성을 확인하면 됩니다.

다음 표에는 휴대 장치 프로파일에서 지원되지 않는 API가 나와 있습니다. 또한 플랫폼에서 실행 중인 응용 프로그램에서 API가 지원되는지 여부를 확인할 수 있는 속성도 나열되어 있습니다.

API	지원 테스트
Accessibility	Capabilities.hasAccessibility
Camera	Camera.isSupported
DatagramSocket	DatagramSocket.isSupported
DNSResolver	DNSResolver.isSupported
DockIcon	NativeApplication.supportsDockIcon
DRMManager	DRMManager.isSupported
EncryptedLocalStore	EncryptedLocalStore.isSupported
HTMLLoader	HTMLLoader.isSupported
LocalConnection	LocalConnection.isSupported
Microphone	Microphone.isSupported
NativeApplication.exit()	—
NativeApplication.menu	NativeApplication.supportsMenu
NativeApplication.isSetAsDefaultApplication()	NativeApplication.supportsDefaultApplication
NativeApplication.startAtLogin	NativeApplication.supportsStartAtLogin
NativeMenu	NativeMenu.isSupported
NativeProcess	NativeProcess.isSupported
NativeWindow	NativeWindow.isSupported
NativeWindow.notifyUser()	NativeWindow.supportsNotification
NetworkInfo	NetworkInfo.isSupported

API	지원 테스트
PDF support	HTMLLoader.pdfCapability
PrintJob	PrintJob.isSupported
SecureSocket	SecureSocket.isSupported
ServerSocket	ServerSocket.isSupported
Shader	—
ShaderFilter	—
StorageVolumeInfo	StorageVolumeInfo.isSupported
XMLSignatureValidator	XMLSignatureValidator.isSupported

HTML이나 JavaScript 기반의 AIR 응용 프로그램을 휴대 장치 프로파일로 작성할 수 없습니다.

다음과 같이 일부 ActionScript 3.0 클래스는 부분적으로만 지원됩니다.

File

iPhone 응용 프로그램은 응용 프로그램 디렉토리와 응용 프로그램 저장소 디렉토리에만 액세스할 수 있습니다. 또한 File.createTempFile()과 File.createTempDirectory()를 호출할 수 있습니다. 다른 디렉토리에 액세스하는 작업(예: FileStream read 또는 write 메서드)을 호출하면 IOError 예외가 발생합니다.

iPhone 응용 프로그램은 File.browseForOpen() 메서드가 제공하는 것과 같은 기본 파일 브라우저 대화 상자를 지원하지 않습니다.

Loader

iPhone 응용 프로그램에서는 Loader.load() 메서드를 사용할 수 없습니다. 또한 Loader.load() 메서드로 로드된 SWF 내용에 있는 모든 ActionScript를 실행할 수 없습니다. 그러나 SWF 파일의 에셋(라이브러리에 있는 동영상 클립, 이미지, 글꼴, 사운드 등)은 사용할 수 있으며, Loader.load() 메서드로 이미지 파일을 로드할 수도 있습니다.

비디오

iPhone의 AIR 응용 프로그램에서는 Sorensen 비디오와 ON2 VP6 비디오만 지원됩니다.

navigateToURL() 메서드를 사용하면 응용 프로그램 외부에서 H.264 비디오를 열 수 있습니다. URL이 비디오를 가리키는 URLRequest 객체를 request 매개 변수로 전달하면 됩니다. 그러면 iPhone의 비디오 플레이어에서 비디오가 실행됩니다.

텍스트 필드

iPhone에서는 텍스트 필드의 글꼴 및 기타 설정에 제한 사항이 있습니다. 자세한 내용은 35페이지의 “글꼴 및 텍스트 입력”을 참조하십시오.

지원되지 않는 API 및 ADL을 사용한 디버그

일부 AIR 기능은 iPhone에서 지원되지 않지만 개발 컴퓨터에서 ADL로 응용 프로그램을 테스트할 때는 사용할 수 있습니다. 따라서 ADL로 내용을 테스트할 때 이러한 차이에 유의해야 합니다.

Speex(오디오), H.264/AVC(비디오), AAC(오디오)와 같은 오디오/비디오 코덱은 iPhone에서 실행되는 AIR 응용 프로그램에서는 사용할 수 없지만 데스크톱에서는 정상적으로 작동합니다.

액세스 가능성 및 화면 판독기 지원 API는 Windows에서 ADL로 사용할 때는 작동하지만 iPhone에서는 작동하지 않습니다.

RTMPE 프로토콜은 데스크톱에서 ADL로 사용할 때는 정상적으로 작동하지만 RTMPE 프로토콜을 사용한 NetConnection 연결 시도는 iPhone에서 실패합니다.

Loader 클래스는 ADL로 내용을 실행할 때는 별도 제한 사항 없이 작동하지만 iPhone에서 실행할 때는 ActionScript 바이트 코드가 포함된 SWF 내용을 로드하려는 시도가 수행되면 오류 메시지가 표시됩니다.

Shader 인스턴스는 ADL에서 실행되지만 iPhone에서는 Pixel Bender 바이트코드가 해석되지 않아 셰이더가 그래픽 효과를 내지 못합니다.

자세한 내용은 22페이지의 “[iPhone 응용 프로그램 디버그](#)”를 참조하십시오.

휴대 장치용 AIR 응용 프로그램에만 한정된 ActionScript API

다음 API는 휴대 장치의 AIR 응용 프로그램에서만 사용할 수 있습니다. 이러한 API는 현재 Flash Player나 데스크톱 버전의 AIR에서 작동하지 않습니다.

화면 방향 API

화면 방향 API를 사용하면 스테이지와 iPhone의 방향을 조정할 수 있습니다.

- Stage.autoOrients - 장치가 회전할 때 응용 프로그램이 스테이지 방향을 자동으로 조정하도록 설정할지 여부를 나타냅니다. Flash Professional CS5 [iPhone 설정] 대화 상자에서 [자동 방향] 옵션을 선택하면 이 속성이 true로 설정됩니다. 응용 프로그램 설명자 파일에서 autoOrients 요소를 true로 설정할 수도 있습니다. 자세한 내용은 13페이지의 “[iPhone 응용 프로그램 설정](#)”을 참조하십시오. Stage 객체에 대해 orientationChanging 이벤트 리스너를 추가하여 자동 방향 조정을 취소할 수 있습니다. 이 이벤트 객체의 preventDefault() 메서드를 호출하면 자동 방향 조정이 취소됩니다.

자동 방향 조정을 사용할 때 최적의 결과를 얻으려면 Stage의 align 속성을 다음과 같이 설정합니다.

```
stage.align = StageAlign.TOP_LEFT;  
stage.scaleMode = StageScaleMode.NO_SCALE;
```

- Stage.deviceOrientation - 장치의 실제 방향입니다. StageOrientation 클래스는 이 속성 값을 정의합니다.
- Stage.orientation - 스테이지의 현재 방향입니다. StageOrientation 클래스는 이 속성 값을 정의합니다.
- Stage.supportsOrientationChange - iPhone에서는 true로, AIR 응용 프로그램에서는 false로 설정됩니다.
- Stage.setOrientation() - 스테이지의 방향을 설정합니다. 이 메서드의 매개 변수는 하나로, 새 스테이지 방향을 정의하는 문자열입니다. StageOrientation 클래스의 상수는 매개 변수에 사용할 수 있는 값을 정의합니다.
- StageOrientation - 스테이지 방향 값을 정의합니다. 예를 들어 StageOrientation.ROTATED_RIGHT는 장치의 기본 방향을 기준으로 오른쪽으로 회전된 스테이지를 나타냅니다.
- StageOrientationEvent - 화면의 방향이 바뀔 때 Stage에서 전달하는 이벤트를 정의합니다. 이 이벤트는 사용자가 iPhone을 회전할 때 발생합니다. 두 가지 유형의 이벤트가 있습니다. Stage는 장치가 회전되면 orientationChanging 이벤트를 전달합니다. 스테이지의 방향이 조정되지 않도록 하려면 orientationChanging 이벤트 객체의 preventDefault() 메서드를 호출하십시오. 스테이지 방향 조정이 완료되면 Stage가 orientationChange 이벤트를 전달합니다.

현재, 화면 방향 API는 휴대 장치의 AIR 응용 프로그램에서만 유용합니다. 모바일 AIR 응용 프로그램과 데스크톱 AIR 응용 프로그램이 소스 코드를 공유하는 경우 API 지원 여부를 확인하려면 Stage.supportsOrientationChange 속성을 사용하십시오.

다음 예에서는 사용자의 장치 회전에 응답하는 방법을 보여 줍니다.

```
stage.addEventListener(StageOrientationEvent.ORIENTATION_CHANGE,
    onOrientationChange);

function onOrientationChange(event:StageOrientationEvent):void
{
    switch (event.afterOrientation) {
        case StageOrientation.DEFAULT:
            // re-orient display objects based on
            // the default (right-side up) orientation.
            break;
        case StageOrientation.ROTATED_RIGHT:
            // Re-orient display objects based on
            // right-hand orientation.
            break;
        case StageOrientation.ROTATED_LEFT:
            // Re-orient display objects based on
            // left-hand orientation.
            break;
        case StageOrientation.UPSIDE_DOWN:
            // Re-orient display objects based on
            // upside-down orientation.
            break;
    }
}
```

이 예에서 다른 스테이지 방향에 대해서는 실제 코드 대신 주석을 달았습니다.

Stage 객체의 `setOrientation()` 메서드를 호출하여 스테이지의 방향을 변경할 수 있습니다. 방향 설정은 비동기식 작업입니다. `orientationChange` 이벤트를 수신하여 방향 변경이 완료되었는지 확인할 수 있습니다. 다음 코드에서는 스테이지를 오른쪽 방향으로 설정하는 방법을 보여 줍니다.

```
stage.addEventListener(StageOrientationEvent.ORIENTATION_CHANGE,
    onOrientationChange);
stage.setOrientation(StageOrientation.ROTATED_RIGHT);

function onOrientationChange(event:StageOrientationEvent):void
{
    // Code to handle the new Stage orientation
}
```

스테이지를 회전하면 크기가 조정되고 **Stage** 객체는 `resize` 이벤트를 전달합니다. `resize` 이벤트에 대한 응답으로 스테이지의 표시 객체를 크기 조정하고 위치 변경할 수 있습니다.

NativeApplication.systemIdleMode 및 SystemIdleMode

`NativeApplication.systemIdleMode` 속성을 사용하여 iPhone이 대기 모드로 전환되지 않도록 할 수 있습니다. 기본적으로 일정 기간 동안 터치 스크린 상호 작용이 없는 경우 iPhone은 대기 모드로 전환됩니다. 대기 모드로 전환될 경우 화면이 어두워질 수 있습니다. 또한 iPhone이 잠금 모드가 될 수도 있습니다. 이 속성은 다음 두 가지 값 중 하나로 설정할 수 있습니다.

- `SystemIdleMode.NORMAL` - iPhone이 정상 대기 모드 비헤이비어를 따릅니다.
- `SystemIdleMode.KEEP_AWAKE` - 응용 프로그램에서 iPhone이 대기 모드로 전환되지 않도록 합니다.

이 기능은 휴대 장치에서만 지원됩니다. 데스크톱 운영 체제에서 실행되는 AIR 응용 프로그램에서는 지원되지 않습니다. 데스크톱에서 실행되는 응용 프로그램에서는 `NativeApplication.systemIdleMode` 속성을 설정해도 아무런 영향을 미치지 않습니다.

다음 코드는 iPhone 대기 모드를 비활성화하는 방법을 보여 줍니다.

```
NativeApplication.nativeApplication.systemIdleMode = SystemIdleMode.KEEP_AWAKE;
```

CameraRoll

CameraRoll 클래스를 사용하여 iPhone 카메라 롤에 이미지를 추가할 수 있습니다. addBitmapData() 메서드는 iPhone 카메라 롤에 이미지를 추가합니다. 이 메서드의 매개 변수는 bitmapData 하나입니다. 이 매개 변수는 카메라 롤에 추가할 이미지가 포함된 BitmapData 객체입니다.

CameraRoll 기능은 휴대 장치에서만 지원됩니다. 데스크톱 운영 체제에서 실행되는 AIR 응용 프로그램에서는 지원되지 않습니다. 응용 프로그램에서 CameraRoll 기능을 지원하는지 여부를 런타임에 확인하려면 정적 CameraRoll.supportsAddBitmapData 속성을 확인하십시오.

addBitmapData() 메서드를 호출한 후에는 CameraRoll 객체에서 다음 두 가지 이벤트 중 하나를 전달합니다.

- complete - 작업이 완료되었습니다.
- error - 오류가 발생했습니다. 예를 들어 iPhone에 이미지를 저장할 충분한 공간이 없습니다.

다음 코드는 카메라 롤에 스테이지 이미지(화면 캡처)를 추가합니다.

```
if (CameraRoll.supportsAddBitmapData)
{
    var cameraRoll:CameraRoll = new CameraRoll();
    cameraRoll.addEventListener(ErrorEvent.ERROR, onCrError);
    cameraRoll.addEventListener(Event.COMPLETE, onCrComplete);
    var bitmapData:BitmapData = new BitmapData(stage.stageWidth, stage.stageHeight);
    bitmapData.draw(stage);
    cameraRoll.addBitmapData(bitmapData);
}
else
{
    trace("not supported.");
}

function onCrError(event:ErrorEvent):void
{
    // Notify user.
}

function onCrComplete(event:Event):void
{
    // Notify user.
}
```

DisplayObject.cacheAsBitmapMatrix

cacheAsBitmapMatrix 속성은 cacheAsBitmap이 true로 설정된 경우 표시 객체를 렌더링하는 방법을 정의하는 Matrix 객체입니다. 응용 프로그램에서 비트맵 버전의 표시 객체를 렌더링할 때 이 행렬을 변형 행렬로 사용합니다.

cacheAsBitmapMatrix를 설정한 경우 응용 프로그램은 표시 행렬 대신에 해당 행렬을 사용하여 렌더링되는 캐시된 비트맵 이미지를 유지합니다. 표시 행렬은 표시 객체의 transform.concatenatedMatrix 값입니다. 이 행렬이 표시 행렬과 일치하지 않는 경우 필요에 따라 비트맵이 크기 조절되고 회전됩니다.

cacheAsBitmapMatrix가 설정된 표시 객체는 cacheAsBitmapMatrix 값이 변경되는 경우에만 렌더링됩니다. 비트맵은 표시 행렬에 맞도록 적절히 크기 조절되거나 회전됩니다.

CPU 및 GPU 기반 렌더링은 모두 cacheAsBitmapMatrix 속성을 사용할 경우 장점이 있습니다. 그러나 일반적으로 GPU 렌더링의 경우에 장점이 더 큽니다.

참고: 하드웨어 가속을 사용하려면 Flash Professional CS5의 [iPhone 설정] 대화 상자에 있는 [일반] 탭에서 [렌더링]을 [GPU]로 설정하십시오. 또는 응용 프로그램 설명자 파일에서 renderMode 속성을 gpu로 설정하십시오.

예를 들어 다음 코드에서는 표시 객체의 변형되지 않은 비트맵 표현을 사용합니다.


```
matrix:Matrix = new Matrix(); // creates an identity matrix
mySprite.cacheAsBitmapMatrix = matrix;
mySprite.cacheAsBitmap = true;
```

다음 코드에서는 현재 렌더링과 일치하는 비트맵 표현을 사용합니다.

```
mySprite.cacheAsBitmapMatrix = mySprite.transform.concatenatedMatrix;
mySprite.cacheAsBitmap = true;
```

일반적으로 ID 행렬 (`new Matrix()`) 또는 `transform.concatenatedMatrix`로 충분합니다. 그러나 다른 행렬(예: 축소된 행렬)을 사용하여 GPU에 다른 비트맵을 업로드할 수 있습니다. 예를 들어 다음 예제에서는 x축 및 y축에서 0.5만큼 크기 조절된 `cacheAsBitmapMatrix` 행렬을 적용합니다. GPU가 사용하는 비트맵 객체는 더 작지만 GPU에서 표시 객체의 `transform.matrix` 속성과 일치하도록 크기를 조정합니다.

```
matrix:Matrix = new Matrix(); // creates an identity matrix
matrix.scale(0.5, 0.5); // scales the matrix
mySprite.cacheAsBitmapMatrix = matrix;
mySprite.cacheAsBitmap = true;
```

일반적으로 응용 프로그램에 표시되는 크기로 표시 객체를 변형하는 행렬을 선택합니다. 예를 들어 응용 프로그램에서 반으로 축소된 비트맵 버전의 스프라이트를 표시하는 경우 반으로 축소되는 행렬을 사용합니다. 응용 프로그램에서 현재 크기보다 큰 스프라이트를 표시하는 경우에는 해당 요인만큼 확장되는 행렬을 사용합니다.

`cacheAsBitmapMatrix` 속성이 설정된 표시 객체의 크기에 대한 실제 제한이 있습니다. 이 제한은 1020 x 1020 픽셀입니다. `cacheAsBitmapMatrix` 속성이 설정된 모든 표시 객체에 대해 총 픽셀 수에 대한 실제 제한이 있습니다. 이 제한은 약 4백만 픽셀입니다.

`cacheAsBitmapMatrix` 및 하드웨어 가속을 사용하는 경우 고려해야 할 사항이 많이 있습니다. 해당 속성을 설정해야 하는 표시 객체와 설정하지 않아야 하는 표시 객체를 아는 것이 중요합니다. 이 속성 사용에 대한 중요한 정보는 32페이지의 “[하드웨어 가속](#)”을 참조하십시오.

GPU 렌더링 진단 기능을 사용하여 응용 프로그램의 디버그 빌드에서 GPU 사용을 진단할 수 있습니다. 자세한 내용은 22페이지의 “[iPhone 응용 프로그램 디버그](#)”를 참조하십시오.

네트워킹 참고 사항

`navigateToURL()` 함수에 다음 URL 스킴을 사용하면 외부 응용 프로그램에서 문서가 열립니다.

URL 스킴	nativeToURL() 호출 결과	예제
mailto:	메일 응용 프로그램에서 새 메시지를 엽니다.	<pre>str = "mailto:test@example.com"; var urlReq:URLRequest = new URLRequest(str); navigateToURL(urlReq);</pre>
sms:	텍스트 메시지 응용 프로그램에서 메시지를 엽니다.	<pre>str = "sms:1-415-555-1212"; var urlReq:URLRequest = new URLRequest(str); navigateToURL(urlReq);</pre>
tel:	휴대폰에서 특정 전화 번호에 전화를 겁니다(사용자 승인 시).	<pre>str = "tel:1-415-555-1212"; var urlReq:URLRequest = new URLRequest(str); navigateToURL(urlReq);</pre>

iPhone 응용 프로그램은 `https` 요청과 같은 보안 트랜잭션 시 서버 인증을 위해 기존에 설치된 자체 서명 루트 인증서를 사용할 수 있습니다. 서버에서 리프 인증서뿐만 아니라 루트 인증서에 체인으로 연결된 모든 중간 인증서도 보내야 합니다.

모바일 응용 프로그램 개발자가 주목해야 할 ActionScript 3.0 API

휴대 장치에서 유용하게 사용되는 기능을 정의하는 ActionScript 3.0 API는 다음과 같습니다.

가속도 센서 API

다음은 응용 프로그램이 장치의 가속도 센서에서 이벤트를 받을 수 있도록 하는 클래스입니다.

- Accelerometer
- AccelerometerEvent

자세한 내용은 [가속도 센서 입력](#)을 참조하십시오.

지리적 위치 API

다음은 응용 프로그램이 장치의 위치 센서에서 이벤트를 받을 수 있도록 하는 클래스입니다.

- Geolocation
- GeolocationEvent

자세한 내용은 [지리적 위치](#)를 참조하십시오.

터치, 다중 터치 및 동작 API

다음은 응용 프로그램이 터치 및 동작 이벤트를 받을 수 있도록 하는 클래스입니다.

- GestureEvent
- GesturePhase
- MultiTouch
- MultitouchInputMode
- TouchEvent
- TransformGestureEvent

자세한 내용은 [터치, 다중 터치 및 동작 입력](#)을 참조하십시오.

4장: iPhone 응용 프로그램 설계 고려 사항

iPhone의 처리 속도와 화면 크기 때문에 설계 및 코딩과 관련하여 특별하게 고려해야 할 사항이 있습니다. 그러나 설계 고려 사항은 모든 응용 프로그램 또는 모바일 응용 프로그램의 개발에 공통적으로 적용되는 경우가 많습니다.

응용 프로그램을 최적화하는 방법에 대한 자세한 내용은 [Flash 플랫폼용 모바일 내용 최적화](#)를 참조하십시오. 이 문서에는 모바일 내용, Flash Player 내용, AIR 내용, ActionScript 기반 내용의 성능을 최적화할 수 있는 일반적인 제안 사항이 많이 수록되어 있습니다. 대부분의 제안 사항은 iPhone용 AIR 응용 프로그램에도 적용됩니다.

중요: 이러한 디자인 고려 사항 및 최적화 기법 중 다수는 iPhone에서 원활하게 실행되는 응용 프로그램을 개발하는 데 있어 필수적입니다.

하드웨어 가속

OpenGL ES 1.1 하드웨어 가속을 사용하여 일부 응용 프로그램의 그래픽 성능을 높일 수 있습니다. 표시 객체가 애니메이션으로 제작되는 게임 등의 응용 프로그램에서 특히 유용합니다. 하드웨어 가속을 사용하는 응용 프로그램은 일부 그래픽 프로세스를 CPU에서 iPhone GPU로 오프로드하여 성능을 크게 향상시킬 수 있습니다.

GPU를 사용하도록 응용 프로그램을 디자인하는 경우 내용이 효과적으로 GPU 가속되도록 해 주는 규칙을 따르는 것이 중요합니다.

하드웨어 가속을 사용하려면 Flash Professional CS5의 [iPhone 설정] 대화 상자에 있는 [일반] 탭에서 [렌더링]을 [GPU]로 설정하십시오. 또한 응용 프로그램 설명자 파일에서 `renderMode` 속성을 `gpu`로 설정할 수도 있습니다.

```
<initialWindow>
  <renderMode>gpu</renderMode>
  ...
```

14페이지의 “[Flash Professional CS5에서 iPhone 응용 프로그램 속성 설정](#)” 및 15페이지의 “[응용 프로그램 설명자 파일에서 iPhone 응용 프로그램 속성 설정](#)”을 참조하십시오.

해당 내용이 자주 변경되지 않는 경우 하드웨어 가속으로 빠르게 렌더링될 수 있는 다음과 같은 4가지 표시 객체 클래스가 있습니다.

- 비트맵 객체
- `cacheAsBitmap` 속성이 `true`로 설정되고 선택적으로 `cacheAsBitmapMatrix` 속성이 설정된 2D 표시 객체(아래 참조)
- 3D 표시 객체(즉, 해당 `z` 속성이 설정됨)
- 단색 사각형 채우기가 하나 있고 가장자리가 화면의 픽셀에 맞춰 정렬되는 표시 객체

해당 객체의 위 또는 아래에서 다른 스프라이트에 애니메이션이 적용될 때마다 백터 기반 객체가 다시 렌더링됩니다. 따라서 애니메이션의 배경 또는 전경으로 사용되는 모든 객체도 이러한 범주 중 하나에 속해야 합니다.

`cacheAsBitmap`이 `true`로 설정된 표시 객체의 경우 `cacheAsBitmapMatrix`를 설정하면 GPU가 행렬 변환으로 만들어진 비트맵을 사용하게 됩니다. GPU는 객체가 회전되거나 크기 조절되는 경우에도 비트맵 표현을 사용합니다. GPU는 CPU가 백터 렌더링 객체를 다시 그리는 것보다 훨씬 더 빠르게 이 비트맵을 합성하고 애니메이션을 적용할 수 있습니다.

`cacheAsBitmap`을 `true`로 설정하는 것만으로 표시 객체(및 모든 자식)가 캐시됩니다. 새 영역이 표시될 때 표시 객체가 다시 그려지지 않거나, 결합된 전체 그래픽이 변환됩니다.

표시 객체의 `cacheAsBitmapMatrix` 속성을 설정하면 표시 객체가 보이지 않을 때에도 응용 프로그램에서 해당 객체의 표현을 만들 수 있습니다. 응용 프로그램은 다음 프레임의 시작 부분에 표시 객체의 캐시된 표현을 만듭니다. 그러면 스테이지에 표시 객체를 추가하는 경우 응용 프로그램에서 해당 객체를 빠르게 렌더링합니다. 또한 응용 프로그램은 객체 회전 또는 크기 조절에 애니메이션을 빠르게 적용할 수 있습니다. 회전 또는 크기 조절되는 객체의 경우 `cacheAsBitmap` 속성을 설정할 때 `cacheAsBitmapMatrix` 속성도 함께 설정해야 합니다.

또한 응용 프로그램은 비트맵으로 캐시되는 객체에 대해 알파 변형을 빠르게 수행할 수도 있습니다. 그러나 하드웨어 가속 알파 변형에 대해서는 0에서 1.0 사이의 `alpha` 값만 지원됩니다. 이는 0에서 256 사이의 `colorTransform.alphaMultiplier` 설정에 해당합니다.

텍스트 필드와 같이 자주 업데이트되는 객체의 경우 `cacheAsBitmap` 속성을 `true`로 설정하지 마십시오.

자주 변경되는 그래픽 내용이 포함된 표시 객체는 일반적으로 GPU 렌더링을 수행하기에 적절한 대상이 아닙니다. 이는 GPU 성능이 낮은 오래된 장치의 경우 특히 그렇습니다. 그래픽을 GPU에 업로드하는 오버헤드 때문에 CPU 렌더링이 더 나은 선택이 될 수 있습니다.

부모를 기준으로 이동하는 자식 표시 객체가 포함된 표시 객체의 구조를 변경합니다. 자식 표시 객체가 부모의 형제가 되도록 변경합니다. 그러면 각각 고유의 비트맵 표현을 갖게 됩니다. 또한 새 그래픽을 GPU에 업로드할 필요 없이 각 표시 객체가 다른 표시 객체를 기준으로 이동할 수 있게 됩니다.

자식 표시 객체에 애니메이션이 적용되지 않는 표시 목록의 최상위 수준에서 `cacheAsBitmap` 속성을 `true`로 설정합니다. 즉, 움직이는 객체가 없는 표시 객체 컨테이너에 대해 설정합니다. 자식 표시 객체에 대해서는 이 속성을 설정하지 마십시오. 애니메이션이 적용되는 다른 표시 객체가 포함된 스프라이트에 대해서는 이 속성을 설정하지 마십시오.

표시 객체의 `z` 속성을 설정한 경우 응용 프로그램에서는 항상 캐시된 비트맵 표현을 사용합니다. 또한 표시 객체의 `z` 속성을 설정한 후에는 객체를 회전하거나 크기를 조절하더라도 응용 프로그램에서 캐시된 비트맵 표현을 사용합니다. 응용 프로그램에서 `z` 속성이 설정된 표시 객체에 대해서는 `cacheAsBitmapMatrix` 속성을 사용하지 않습니다. `rotationX`, `rotationY`, `rotationZ`, `transform.matrix3D` 속성 등의 3차원 표시 객체 속성을 설정하는 경우에도 동일한 규칙이 적용됩니다.

하드웨어 가속을 사용할 내용이 포함된 표시 객체 컨테이너에 대해서는 `scrollRect` 또는 `mask` 속성을 설정하지 마십시오. 이러한 속성을 설정하면 해당 표시 객체 컨테이너 및 자식 객체에서 하드웨어 가속이 비활성화됩니다. `mask` 속성을 설정하는 대신, 마스크가 적용되는 표시 객체 위에 마스크 표시 객체의 레이어를 두는 방법이 있습니다.

하드웨어 가속에 사용할 수 있는 표시 객체의 크기에는 제한이 있습니다. 오래된 장치에서는 폭과 높이 모두 1024픽셀 이하여야 합니다. 최신 장치에서는 2048픽셀 이하여야 합니다. GPU 렌더링 진단 도구를 사용하여 장치에서 성능을 테스트할 수 있습니다.

GPU는 또한 비트맵 이미지를 저장하는 데 iPhone RAM을 사용합니다. 이때 비트맵 이미지에 필요한 메모리 용량 이상을 사용합니다.

GPU는 각 비트맵 이미지 크기에 대해 2의 거듭제곱 형식으로 메모리를 할당합니다. 예를 들어 GPU는 512 x 1024 또는 8 x 32 크기로 메모리를 예약할 수 있습니다. 또한 9 x 15 픽셀 이미지는 16 x 16 픽셀 이미지와 동일한 메모리 용량을 차지합니다. 캐시된 표시 객체의 경우 각 방향에서 2의 거듭제곱에 가깝지만 이를 초과하지 않는 크기를 사용할 수 있습니다. 예를 들어 33 x 17 픽셀 표시 객체보다는 32 x 16 픽셀 표시 객체를 사용하는 것이 더 효율적입니다.

데스크톱과 같이 다른 플랫폼에 맞춰진 크기의 에셋을 크기 조절된 스테이지에 맞춰 축소하지 마십시오. 대신 `cacheAsBitmapMatrix` 속성을 사용하거나, iPhone용으로 제작하기 전에 에셋의 크기를 조절하십시오. 3D 객체는 표면 이미지를 캐싱할 때 `cacheAsBitmapMatrix` 속성을 무시합니다. 따라서 3D 표면에 렌더링될 표시 객체는 제작 전에 크기를 조절하는 것이 좋습니다.

하드웨어 가속을 사용하는 것과 RAM을 사용하는 것에는 서로 절충되는 이점이 있습니다. 메모리 용량이 가득 차면 iPhone OS는 실행 중인 다른 기본 iPhone 응용 프로그램에 알려 메모리를 비우도록 합니다. 해당 응용 프로그램들이 이 알림을 처리하여 메모리를 비우는 작업을 할 때 사용자의 응용 프로그램과 CPU 사이클을 놓고 결합할 수 있습니다. 이에 따라 응용 프로그램의 성능이 일시적으로 저하될 수 있습니다. 실행 중인 프로세스에 사용할 수 있는 메모리가 훨씬 적은 오래된 장치에서 응용 프로그램을 테스트해보는 것이 좋습니다.

iPhone에서 응용 프로그램을 디버그하는 경우 GPU 렌더링 진단 기능을 활성화할 수 있습니다. 이 기능은 응용 프로그램에서 GPU 렌더링을 사용하는 방식을 확인하는 데 도움이 됩니다. 자세한 내용은 22페이지의 “iPhone 응용 프로그램 디버그”의 “GPU 렌더링 진단을 사용하여 디버그”를 참조하십시오.

cacheAsBitmapMatrix 속성을 사용하는 방법에 대한 자세한 내용은 27페이지의 “휴대 장치용 AIR 응용 프로그램에만 한정된 ActionScript API”의 "DisplayObject.cacheAsBitmapMatrix" 단원을 참조하십시오.

표시 객체 성능을 향상시키는 다른 방법

하드웨어 가속을 통해 일부 표시 객체 클래스에서 그래픽 성능을 높일 수 있습니다. 다음은 그래픽 성능을 극대화할 수 있는 몇 가지 팁입니다.

- 스테이지에 표시되는 항목 수를 최대한 줄입니다. 항목마다 렌더링하고 주위의 다른 항목과 합성하는 데 시간이 걸립니다. 더 이상 표시할 필요가 없는 표시 객체는 visible 속성을 false로 설정하거나 스테이지에서 제거(removeChild())합니다. alpha 속성을 0으로 설정해서는 안 됩니다.
- 가급적 블렌드 모드를 사용하지 않도록 하고 특히 레이어 블렌드 모드 사용을 피합니다. 가능하면 보통 블렌드 모드를 사용하십시오.
- 표시 객체 필터는 계산하는 데 리소스가 많이 소모되므로 사용하지 않는 것이 좋습니다. 예를 들어 소개 화면에 필터를 몇 개 사용하는 것은 괜찮지만 많은 수의 객체나 애니메이션이 적용되는 객체, 또는 높은 프레임 속도가 요구되는 경우에 필터를 사용하는 것은 바람직하지 않습니다.
- 변형 모양을 가급적 사용하지 않습니다.
- 클리핑을 가급적 사용하지 않습니다.
- Graphic.beginBitmapFill() 메서드를 호출할 때 가능하면 repeat 매개 변수를 false로 설정합니다.
- 오버드로우하지 않습니다. 배경색을 배경으로 사용합니다. 큰 모양을 서로 겹치지 않습니다. 그려야 하는 모든 픽셀마다 리소스가 소모됩니다. 이는 하드웨어 가속되지 않는 표시 객체의 경우 특히 그렇습니다.
- 길고 가늘게 튀어나온 부분이 있거나, 가장자리가 서로 교차하거나, 가장자리를 따라 세부 요소가 많은 모양을 사용하지 않습니다. 이러한 모양은 가장자리가 매끄러운 표시 객체보다 렌더링하는 데 오래 걸립니다. 이는 하드웨어 가속되지 않는 표시 객체의 경우 특히 그렇습니다.
- 비트맵 크기를 $2^n \times 2^m$ 비트보다 작지만 이 크기에 가깝게 만듭니다. 크기가 반드시 2의 거듭제곱일 필요는 없지만, 2의 거듭제곱을 초과하지 않으면서 2의 거듭제곱에 가까워야 합니다. 예를 들어 31 x 15 픽셀 이미지는 33 x 17 픽셀 이미지보다 빠르게 렌더링됩니다. 31과 15는 2: 32 및 16보다 조금 작습니다. 또한 이러한 이미지는 메모리를 보다 효율적으로 사용합니다.
- 표시 객체의 크기를 1024 x 1024 픽셀(또는 최신 장치의 경우 2048 x 2048)로 제한합니다.

정보 밀도

휴대 장치 화면은 데스크톱보다 픽셀 밀도는 높고 물리적 크기는 더 작습니다. 텍스트가 선명할수록 보기는 좋지만 글리프의 물리적 크기가 최소화되어야 가독성이 향상됩니다.

휴대 장치는 이동 중이나 조명 상태가 나쁜 곳에서도 자주 사용됩니다. 따라서 현실적으로 가독성을 확보하면서 화면에 표시할 수 있는 정보의 양을 고려해야 합니다. 픽셀 크기가 같은 데스크톱 화면에 표시할 수 있는 정보의 양보다 적을 수밖에 없습니다.

입력 체계 계층을 사용하여 중요한 정보를 강조하십시오. 글꼴 크기, 두께, 배치, 간격 등을 사용하면 사용자 인터페이스 요소의 상대적 중요도를 나타낼 수 있습니다. 계층 구조의 각 수준에 큐를 하나 이상 사용할 수 있습니다. 이러한 큐는 응용 프로그램 전체에서 일관되게 적용합니다. 이러한 큐는 공간적(들여쓰기, 줄 간격, 배치) 또는 그래픽적(크기, 스타일, 글꼴 색상)으로 만들 수 있습니다. 다중 큐를 적용하면 효과적으로 계층을 명확하게 표현할 수 있습니다. 단, 각 그룹화 수준마다 큐를 3개 이상 사용하지 않도록 합니다.

필요한 레이블과 설명을 가능한 한 줄입니다. 예를 들어 텍스트 필드에서 내용을 보여 주는 샘플 입력을 사용하고 별도의 레이블은 사용하지 않습니다.

글꼴 및 텍스트 입력

장치 글꼴을 사용할 때 모양이 가장 좋습니다. 예를 들어 iPhone에는 다음과 같은 장치 글꼴이 있습니다.

- Serif: Times New Roman, Georgia 및 _serif
- Sans-serif: Helvetica, Arial, Verdana, Trebuchet, Tahoma 및 _sans
- Fixed-width: Courier New, Courier 및 _typewriter

크기가 14픽셀 이상인 글꼴을 사용합니다.

편집 가능한 텍스트 필드에는 장치 글꼴을 사용합니다. 텍스트 필드의 장치 글꼴도 포함된 글꼴보다 빠르게 렌더링됩니다.

입력 텍스트 필드에 밑줄이 그어진 텍스트를 사용하지 마십시오. 또한 텍스트 필드의 정렬도 설정해서는 안 됩니다. iPhone의 입력 텍스트 필드는 왼쪽 정렬(기본값)만 지원합니다.

Flash Professional CS5에서 텍스트 필드에 TLF 텍스트 설정을 사용하는 경우 ActionScript 3.0 설정에서 런타임 공유 라이브러리의 기본 링크를 해제하십시오. 그렇지 않으면 응용 프로그램이 런타임 공유 라이브러리 SWF 파일을 사용하려 하기 때문에 iPhone에서 작동하지 않게 됩니다.

- 1 [파일] > [제작 설정]을 선택합니다.
- 2 [제작 설정] 대화 상자에서 [Flash] 탭을 클릭합니다.
- 3 [스크립트(ActionScript 3.0)] 드롭다운 목록의 오른쪽에 있는 [스크립트] 버튼을 클릭합니다.
- 4 [라이브러리 경로] 탭을 클릭합니다.
- 5 [기본 링크] 드롭다운 목록에서 [코드로 병합]을 선택합니다.

입력 텍스트 필드를 사용하는 대체 방법을 구현해보십시오. 예를 들어 사용자가 숫자 값을 입력하도록 할 때는 텍스트 필드를 사용하지 않아도 됩니다. 대신 값을 높이거나 낮추는 버튼 2개만 제공하면 됩니다.

이때는 가상 키보드에 사용되는 공간을 고려해야 합니다. 가상 키보드가 활성화되면(예: 사용자가 텍스트 필드를 누른 경우) 응용 프로그램은 스테이지의 위치를 조정합니다. 이러한 자동 위치 조정 기능은 선택한 입력 텍스트 필드가 가려지지 않도록 합니다.

- 스테이지 맨 위에 있는 텍스트 필드는 표시되는 스테이지 영역의 맨 위로 이동합니다. 이때 가상 키보드를 표시하기 위해 스테이지 영역이 좁게 나타납니다.
- 스테이지 맨 아래에 있는 텍스트 필드는 새 스테이지 영역의 맨 아래에 그대로 있습니다.
- 스테이지의 다른 부분에 있는 텍스트 필드는 스테이지의 수직 가운데로 이동합니다.

사용자가 편집하기 위해 텍스트 필드를 클릭하면(이때 가상 키보드가 표시됨) TextField 객체가 focusIn 이벤트를 발송합니다. 텍스트 필드의 위치를 조정하도록 이 이벤트에 대한 이벤트 리스너를 추가할 수 있습니다.

한 줄 텍스트 필드에서는 사용자가 텍스트를 편집할 때 텍스트 오른쪽에 지우기 버튼이 포함됩니다. 그러나 텍스트 필드가 너무 좁으면 이 지우기 버튼이 표시되지 않습니다.

한 줄 텍스트 필드에서 텍스트를 편집한 후 키보드의 Done 키를 누르면 가상 키보드가 숨겨집니다.

한편 여러 줄 텍스트 필드에서 텍스트를 편집한 후에는 텍스트 필드 외부로 눌러야 가상 키보드가 숨겨집니다. 텍스트 필드 외부로 누르면 텍스트 필드에서 포커스가 없어집니다. 가상 키보드가 표시되어도 텍스트 필드의 외부 영역이 나타나도록 응용 프로그램을 설계해야 합니다. 텍스트 필드가 너무 크면 다른 영역이 표시되지 않을 수 있습니다.

일부 Flash Professional CS5 구성 요소를 사용할 경우 텍스트 필드에서 포커스를 없애지 못할 수 있습니다. 이러한 구성 요소의 포커스 비헤이비어는 데스크톱 시스템에서 사용하기 알맞게 설계되었기 때문입니다. 한 예로 TextArea 구성 요소를 들 수 있습니다. 편집과 같은 작업으로 인해 이 구성 요소에 포커스가 있으면 다른 표시 객체를 클릭하여 포커스를 없애지 못합니다. 또한 스테이지에 일부 다른 Flash Professional CS5 구성 요소를 배치해도 편집 중인 텍스트에서 포커스를 변경하지 못할 수 있습니다.

키보드 이벤트에 의존하는 것은 좋지 않습니다. 예를 들어 웹용으로 디자인된 일부 SWF 내용은 사용자가 키보드를 사용하여 응용 프로그램을 제어하도록 합니다. 그러나 iPhone에서는 사용자가 텍스트 필드를 편집할 때만 가상 키보드가 표시됩니다.

iPhone 응용 프로그램은 가상 키보드가 표시될 때만 키보드 이벤트를 전달합니다.

응용 프로그램 상태 저장

응용 프로그램은 언제든지 종료될 수 있습니다(예: 전화가 오는 경우). 따라서 응용 프로그램 상태가 변경될 때마다 상태를 저장하는 것이 좋습니다. 예를 들어 파일 또는 데이터베이스에 대한 설정을 응용 프로그램 저장소 디렉토리에 저장한 다음 또는 로컬 공유 객체에 데이터를 저장할 수 있습니다. 그런 다음 다시 시작할 때 응용 프로그램 상태를 복원할 수 있습니다. 전화 통화로 인해 응용 프로그램이 중단되는 경우 통화가 끝나면 응용 프로그램이 다시 시작됩니다.

응용 프로그램이 종료될 때 `exiting` 이벤트를 전달하는 `NativeApplication` 객체에 의존하지 마십시오. 이벤트가 전달되지 않을 수도 있습니다.

화면 방향 변경

iPhone 내용은 세로 또는 가로 방향으로 볼 수 있습니다. 따라서 응용 프로그램에서 화면 방향 변화를 어떻게 처리할지 고려해야 합니다. 자세한 내용은 [장치 방향 감지](#)를 참조하십시오.

히트 대상

사용자가 누르는 버튼과 기타 사용자 인터페이스 요소를 디자인할 때는 히트 대상의 크기를 고려하십시오. 이러한 요소는 터치 스크린에서 손가락으로 쉽게 활성화할 수 있을 만큼 충분히 커야 합니다. 또한 대상 간의 공간도 충분해야 합니다. 히트 대상의 크기는 44픽셀에서 57픽셀 사이가 좋습니다.

메모리 할당

메모리의 새 블록을 할당하면 그만큼 메모리 리소스가 소모됩니다. 애니메이션 또는 조작을 실행하는 동안 가비지 수집이 트리거되어 응용 프로그램이 느려지거나 성능이 저하될 수 있습니다.

객체를 제거하고 새로 만드는 대신 가능한 한 재사용하십시오.

배열보다 벡터 객체가 메모리를 적게 소모할 수 있다는 점을 염두에 두십시오. 자세한 내용은 [Vector 클래스와 Array 클래스 비교](#)를 참조하십시오.

메모리 사용에 대한 자세한 내용은 [메모리 절약](#)을 참조하십시오.

드로잉 API

`ActionScript` 드로잉 API(`Graphics` 클래스)를 사용하여 그래픽을 제작하지 않는 것이 좋습니다. 드로잉 API는 스테이지에서 객체를 동적으로 만든 다음 래스터라이저로 렌더링합니다. 제작 시에는 가능하면 해당 객체를 정적인 상태로 만드십시오.

드로잉 API를 사용하여 만든 객체는 반복적으로 호출될 경우 `ActionScript`를 실행할 때마다 삭제된 후 다시 생성됩니다. 반면 정적 객체는 여러 타임라인을 진행하는 내내 메모리에 유지됩니다.

이벤트 버블링

여러 단계로 중첩된 표시 객체 컨테이너의 경우 이벤트 버블링에 리소스가 많이 소모될 수 있습니다. 이벤트를 전적으로 대상 객체에서 처리하고 이벤트 객체의 `stopPropagation()` 메서드를 호출함으로써 이러한 리소스 소모를 줄이십시오. 이 메서드를 호출하면 이벤트가 버블링되지 않습니다. 또한 이 메서드를 호출할 경우 부모 객체에 이벤트가 전달되지 않습니다.

중첩된 표시 객체를 병합하면 긴 이벤트 체인이 발생하지 않는 데 따른 여러 가지 이점이 있습니다.

가급적 `TouchEvent` 대신 `MouseEvent` 이벤트에 대해 등록하십시오. `MouseEvent` 이벤트가 `TouchEvent` 이벤트보다 프로세서 오버헤드를 적게 사용합니다.

가급적 `mouseEnabled` 및 `mouseChildren` 속성을 `false`로 설정합니다.

비디오 성능 최적화

모바일 비디오 재생을 최적화하려면 비디오가 재생되는 동안 응용 프로그램에서 수행되는 다른 작업을 최소화하십시오. 이렇게 해야 비디오 디코딩 및 렌더링 프로세스에 가능한 한 많은 CPU가 사용될 수 있습니다.

비디오가 재생되는 동안 실행되는 `ActionScript` 코드가 거의 없거나 아예 없도록 합니다. 잦은 간격 타이머나 타임라인에서 실행되는 코드를 실행하지 않도록 하십시오.

비디오가 아닌 표시 객체를 다시 그리는 작업을 최소화합니다. 특히 비디오 영역과 교차하는 표시 객체를 다시 그리지 않도록 합니다. 비디오 밑에 숨겨진 경우에도 마찬가지입니다. 이 경우에도 다시 그려지고 처리 리소스를 사용합니다. 예를 들어 위치 표시기에 대해 단순한 모양을 사용하고 모든 프레임에서가 아니라 1초에 2회만 위치 표시기를 업데이트할 수 있습니다. 비디오 컨트롤이 비디오 영역에 겹치지 않도록 하십시오. 비디오 영역 바로 아래에 비디오 컨트롤을 배치합니다. 비디오 버퍼링 애니메이션이 있는 경우 사용되지 않을 때 비디오 뒤에 숨기지 말고 표시되지 않도록 설정합니다.

Flex 및 Flash 구성 요소

많은 Flex 구성 요소와 Flash 구성 요소는 데스크톱 응용 프로그램에서 사용하도록 디자인됩니다. 이러한 구성 요소, 특히 표시 구성 요소는 휴대 장치에서 느리게 수행될 수 있습니다. 느리게 수행될 뿐 아니라, 데스크톱 구성 요소는 휴대 장치용으로 적절하지 않은 상호 작용 모델을 포함할 수 있습니다.

Adobe에서는 모바일용으로 최적화된 버전의 Flex 프레임워크를 개발 중입니다. 자세한 내용은 <http://labs.adobe.com/technologies/flex/mobile/>을 참조하십시오.

응용 프로그램 파일 크기 줄이기

다음은 IPA 파일의 파일 크기를 줄이기 위한 몇 가지 팁입니다.

- 배경 비트맵을 검토하여 올바른 크기인지(필요한 것보다 크지 않은지) 확인합니다.
- 포함되는 추가 글꼴이 있는지 확인합니다.
- 알파 채널의 PNG 에셋을 확인하여 필요하지 않은 경우 제거합니다. PNG 크런치와 같은 유틸리티를 사용하여 PNG 에셋의 크기를 줄입니다.
- 가능한 경우 PNG 에셋을 JPG 에셋으로 변환합니다.
- 낮은 비트율을 사용하여 사운드 파일을 압축하는 것을 고려합니다.
- 사용되지 않는 모든 에셋을 제거합니다.