

# Создание приложений ADOBE® AIR® с помощью Packager for iPhone®

## **Юридическая информация**

Юридическую информацию см. на веб-странице [http://help.adobe.com/ru\\_RU/legalnotices/index.html](http://help.adobe.com/ru_RU/legalnotices/index.html).

# Содержание

## Глава 1. Начало работы по созданию программ AIR для iPhone

Важные понятия .....	1
Получение инструментов разработчика у компании Adobe .....	4
Получение файлов разработчика у компании Apple .....	5
Создание приложения Hello World для iPhone с помощью Flash Professional CS5 .....	9

## Глава 2. Компиляция и отладка приложений iPhone

Значок iPhone и изображения начального экрана .....	14
Настройки приложения iPhone .....	16
Компиляция файла программы установки приложения iPhone (IPA) .....	22
Установка приложения iPhone .....	25
Отладка приложения iPhone .....	26
Отправка приложения для iPhone в магазин App Store .....	29

## Глава 3. Поддержка API-интерфейсов ActionScript 3.0 для мобильных устройств

API-интерфейсы ActionScript 3.0, неподдерживаемые на мобильных устройствах .....	30
Специальные API-интерфейсы ActionScript для мобильных устройств .....	32
API-интерфейсы ActionScript 3.0, которые представляют особый интерес для разработчиков приложений для мобильных устройств .....	37

## Глава 4. Особенности дизайна приложений iPhone

Аппаратное ускорение .....	38
Другие способы повышения производительности экранных объектов .....	40
Плотность информации .....	41
Шрифты и ввод текста .....	42
Сохранение состояния приложения .....	43
Изменение ориентации экрана .....	43
Мишени .....	43
Выделение памяти .....	44
API-интерфейс рисования .....	44
Восхождение событий по цепочке .....	44
Оптимизация производительности видео .....	44
Компоненты Flex и Flash .....	45
Сокращение размера файла приложения .....	45

# Глава 1. Начало работы по созданию программ AIR для iPhone

Инструменты Adobe® Flash® Platform и код ActionScript® 3.0 можно использовать для создания приложений Adobe® AIR® для устройств iPhone и iPod Touch. Эти приложения распространяются, устанавливаются и запускаются так же, как все остальные приложения для iPhone.

*Примечание.* Далее в этом документе под названием «iPhone» следует также понимать iPhone и iPod Touch.

Инструмент Packager for iPhone® включен в пакет программ Adobe® Flash® Professional CS5. Приложение Packager for iPhone компилирует байт-код ActionScript 3.0 в собственный код приложения iPhone. Приложения iPhone распространяются как файлы установщика приложения iPhone (файлы .ipa) через магазин iTunes Store.

С помощью Flash Professional CS5 или Adobe® Flash® Builder™ 4 можно редактировать исходное содержимое ActionScript 3.0 для приложения.

Для разработки приложений iPhone следует использовать Flash Professional CS5.

Также необходимо получить сертификаты разработчика iPhone у компании Apple.

**Важно.** Прежде чем приступить к разработке приложений iPhone, ознакомьтесь с информацией по созданию программного обеспечения для устройства iPhone. См. раздел «[«Особенности дизайна приложений iPhone»](#)» на странице 38». Кроме того, узнайте, какие файлы разработчика необходимы для создания приложений iPhone. См. раздел «[«Получение файлов разработчика у компании Apple»](#)» на странице 5».

## Важные понятия

Важно получить представление о соответствующих понятиях и рабочем процессе, прежде чем приступить к разработке приложений iPhone с помощью ActionScript 3.0.

## Глоссарий

Ниже перечислены важные термины, которые используются в связи с разработкой приложений iPhone.

**Веб-сайт центра разработчиков iPhone** — это веб-сайт Apple (<http://developer.apple.com/iphone/>) на котором доступны следующие возможности:

- регистрация в качестве разработчика iPhone;
- создание сертификатов на разработку для iPhone и управление ими, получение профилей и идентификаторов приложений (см. описание ниже);
- отправка приложений в App Store.

**Сертификат на разработку для iPhone** используется для идентификации разработчика приложений.

Этот файл необходимо получить у компании Apple. Данный сертификат необходимо преобразовать в файл сертификата P12, чтобы подписать приложение iPhone, созданное с помощью ActionScript 3.0. См. раздел «[Файл сертификата P12](#)».

Для простой отладки и тестирования приложений Flash Professional CS5 на компьютере разработчика сертификат на разработку для iPhone не требуется. Однако сертификат на разработку требуется для установки и тестирования приложения на устройстве iPhone.

Сертификат на разработку — не то же самое, что сертификат на распространение, который необходим для сборки окончательной версии приложения. Сертификат на распространение предоставляется компанией Apple после сборки окончательной версии приложения.

**Запрос на подпись сертификата** — это файл, содержащий личные данные, использованные для создания сертификата на разработку. Файл имеет расширение CSR.

**Профиль обеспечения** — это файл, который позволяет тестировать или распространять приложение iPhone. Профиль обеспечения необходимо получить у компании Apple. Профиль обеспечения присваивается определенному сертификату на разработку, идентификатору приложения, а также одному или нескольким идентификаторам устройств. Файлы обеспечения бывают нескольких типов.

- **Профиль обеспечения разработки** используется для установки тестовой версии приложения на устройство iPhone разработчика.
- **Профиль обеспечения тестирования** также называется специальным профилем обеспечения. Используется для распространения тестовой версии приложения нескольким пользователям (и устройствам iPhone). Имея такой профиль обеспечения и тестовое приложение, пользователи могут протестировать приложение, еще не размещенное в магазине App Store. Примечание. С целью распространения тестовых приложений на несколько устройств также можно использовать профиль обеспечения разработки.
- **Профиль обеспечения распространения** используется для создания приложения iPhone с последующей отправкой в App Store.

**Идентификатор приложения** — это уникальная строка, идентифицирующая приложение iPhone (или несколько приложений) определенного разработчика. Создать свой идентификатор приложения можно на веб-сайте центра разработчиков iPhone. С каждым профилем обеспечения связан идентификатор приложения или шаблон идентификатора приложения. Идентификатор приложения (или его шаблон) используется в процессе разработки. Он указывается в диалоговом окне Flash Professional CS5 «Настройки iPhone» (или в файле дескриптора приложения).

Идентификаторы приложений в центре разработчиков iPhone содержат идентификатор начального числа пакета и идентификатор пакета. Идентификатор начального числа пакета — это строка символов, например 5RM86Z4DJM, которую Apple присваивает идентификатору приложения. Идентификатор пакета содержит строку с обратным доменным именем. Идентификатор пакета может заканчиваться звездочкой (\*), указывающей на подстановочный шаблон идентификатора. Примеры:

- 5RM86Z4DJM.com.example.helloWorld
- 96LPVWEASL.com.example.\* (подстановочный шаблон идентификатора приложения)

В центре разработчиков iPhone представлены два типа идентификаторов приложений.

- Подстановочные шаблоны идентификатора приложения, которые заканчиваются звездочкой (\*), например 96LPVWEASL.com.myDomain.\* или 96LPVWEASL.\*. Профиль обеспечения с таким идентификатором позволяет создавать тестовые приложения с использованием идентификатора приложения, соответствующего шаблону. Звездочку можно заменить любой строкой действительных символов. Например, если на веб-сайт центра разработчиков iPhone указан идентификатор приложения 96LPVWEASL.com.example.\*, используйте в качестве идентификатора com.example.foo.

- Специфические идентификаторы приложений — это уникальные идентификаторы конкретных приложений. В центре разработчиков iPhone такие идентификаторы не заканчиваются звездочкой. Пример: 96LPVWEASL.com.myDomain.myApp. Профиль обеспечения с таким идентификатором требует точного соответствия идентификаторов приложений. Например, если на веб-сайте центра разработчиков iPhone указан идентификатор приложения 96LPVWEASL.com.example.helloWorld, используйте идентификатор com.example.foo.

При разработке приложения идентификатор указывается в диалоговом окне Flash Professional CS5 «Настройки iPhone» или в файле дескриптора приложения. Дополнительные сведения об идентификаторе приложения см. в разделе «Вкладка "Развертывание"» статьи «[Настройка свойств приложения iPhone во Flash Professional CS5](#)» на странице 16 или в разделе «[Настройка свойств приложения iPhone в файле дескриптора приложения](#)» на странице 18».

**Важно!** При указании идентификатора приложения пропускайте идентификатор начального числа пакета. Например, если Apple отображает ваш идентификатор приложения как 96LPVWEASL.com.example.bob.myApp, пропустите 96LPVWEASL и укажите только com.example.bob.myApp. Если Apple отображает ваш идентификатор приложения как 5RM86Z4DJM.\*, пропустите 5RM86Z4DJM — это подстановочный шаблон идентификатора приложения.

В центре разработчиков iPhone можно найти идентификатор приложения (или подстановочный шаблон идентификатора приложения), связанный с профилем обеспечения (<http://developer.apple.com/iphone>). Перейдите на портал программы разработчиков iPhone, затем в раздел Provisioning (Обеспечение).

**Файл сертификата P12** — это P12-файл (с расширением .p12), который представляет собой тип файла сертификата (файл обмена личной информацией). Инструмент Packager for iPhone использует этот тип сертификата для создания приложений iPhone. Сертификат разработчика, полученный у компании Apple, необходимо преобразовать в этот тип сертификата.

**Уникальный идентификатор устройства** — это уникальный код, идентифицирующий определенное устройство iPhone. Он также называется UDID или идентификатор устройства.

## Обзор рабочего процесса разработки

В процессе разработки приложения для iPhone необходимо выполнить следующие действия.

- 1 Установите программное обеспечение Flash Professional CS5, полученное у компании Adobe.
- 2 Установите программу iTunes.
- 3 Получите файлы разработчика у компании Apple, включая сертификат разработчика и профили обеспечения. См. раздел «[Получение файлов разработчика у компании Apple](#)» на странице 5».
- 4 Преобразуйте сертификат разработчика в файл сертификата P12. Для Flash CS5 требуются сертификаты типа P12. См. раздел «[Получение файлов разработчика у компании Apple](#)» на странице 5».
- 5 С помощью iTunes свяжите свой профиль обеспечения с устройством iPhone.
- 6 Напишите приложение во Flash Professional CS5.

Важно иметь представление о рекомендуемых методах разработки и оптимизации кода для приложений iPhone. См. раздел «[Особенности дизайна приложений iPhone](#)» на странице 38».

Также следует учитывать, что некоторые API-интерфейсы ActionScript 3.0 имеют ограниченную функциональность или не поддерживаются на устройствах iPhone. См. раздел «[Поддержка API-интерфейсов ActionScript 3.0 для мобильных устройств](#)» на странице 30».

Также можно использовать Flash Builder 4.0 для редактирования кода ActionScript 3.0 в приложении.

С помощью Flash Professional CS5 можно протестировать приложение на компьютере разработчика.

- 7 Создайте изображения для значка и начального экрана приложения. Каждое приложение iPhone имеет набор значков, которые делают его узнаваемым для пользователей. Устройство iPhone отображает изображение начального экрана во время загрузки приложения. См. раздел «[Значок iPhone и изображения начального экрана](#)» на странице 14».
- 8 Отредактируйте настройки iPhone. В их число входят следующие:
- идентифицирующие данные приложения, включая имя файла, название приложения, номер версии и идентификатор приложения;
  - местоположение исходного изображения значка приложения;
  - сертификат P12 и профиль обеспечения, присвоенные приложению;
  - исходное соотношение сторон экрана приложения.

Во Flash Professional CS5 для редактирования этих настроек можно использовать диалоговое окно «Настройки iPhone». Дополнительные сведения см. в разделе «[Настройка свойств приложения iPhone во Flash Professional CS5](#)» на странице 16».

Эти настройки можно также отредактировать непосредственно в файле дескриптора приложения. Дополнительные сведения см. в разделе «[Настройка свойств приложения iPhone в файле дескриптора приложения](#)» на странице 18».

- 9 Скомпилируйте файл IPA с помощью инструмента Packager for iPhone. См. раздел «[Компиляция файла программы установки приложения iPhone \(IPA\)](#)» на странице 22».
- 10 Установите и протестируйте приложение на устройстве iPhone. Установите файл IPA с помощью iTunes. Для специального распространения повторите этот общий процесс, но вместо профиля обеспечения разработки используйте профиль обеспечения тестирования. Для окончательного распространения приложения повторите этот процесс с использованием профиля обеспечения распространения. (Сведения о разных типах профилей обеспечения см. в разделе «[Глоссарий](#)» на странице 1».)
- По завершении создания дистрибутивной версии приложения см. инструкции в разделе «[Отправка приложения для iPhone в магазин App Store](#)» на странице 29».

Краткие инструкции по созданию базового приложения iPhone см. в разделе «[Создание приложения Hello World для iPhone с помощью Flash Professional CS5](#)» на странице 9».

## Получение инструментов разработчика у компании Adobe

Для разработки приложений iPhone с использованием ActionScript 3.0 требуется Flash Professional CS5.

**Важно.** Обновите пробную версию инструмента Packager for iPhone, включенную в комплект Flash Professional CS5, до полнофункциональной. Во Flash Professional CS5 выберите «Справка» > «Обновления».

Приложение Flash Builder 4 можно также использовать для редактирования кода ActionScript. Flash Builder 4 доступен на веб-сайте <http://www.adobe.com/products/flashbuilder/>.

## Получение файлов разработчика у компании Apple

Как и при разработке любого приложения для iPhone, сначала необходимо получить файлы разработчика iPhone у компании Apple. Необходимо получить сертификат разработчика iPhone и мобильный профиль обеспечения. Также необходимо получить другие профили обеспечения. Определения этих файлов см. в разделе «[Глоссарий](#)» на странице 1».

***Примечание.** Получение этих файлов является важной частью процесса разработки приложения. Обязательно выполните эту процедуру перед разработкой приложения. Получить файлы разработчика не так просто. Внимательно прочитайте эти инструкции и инструкции на веб-сайте Apple iPhone Dev Center.*

### Получение файлов разработчика iPhone и работа с ними

Необходимо получить сертификат разработчика iPhone и профили обеспечения у компании Apple. Предоставленный сертификат необходимо преобразовать в сертификат P12.

#### Установка iTunes

Для установки приложения на iPhone требуется программа iTunes. Кроме того, iTunes служит для определения идентификатора устройства iPhone. Идентификатор устройства необходимо указать в заявке на получение сертификата разработчика iPhone.

#### Отправка заявки на получение сертификата разработчика iPhone и создание профиля обеспечения

Если это еще не сделано, отправьте заявку на регистрацию в качестве разработчика iPhone в центр разработчиков iPhone (<http://developer.apple.com/iphone/>).

***Примечание.** Для разработки приложений AIR для iPhone не требуется пакет iPhone SDK или XCode. Вы должны быть зарегистрированным разработчиком iPhone. И вам необходимо получить сертификат разработчика и профиль обеспечения.*

- 1 Выполните вход в центр разработчиков iPhone, указав идентификатор учетной записи разработчика iPhone.
- 2 В центре разработчиков iPhone приобретите сертификат разработчика для iPhone.  
Вы получите электронное сообщение от компании Apple, содержащее код активации программы разработчиков iPhone.
- 3 Вернитесь в центр разработчиков iPhone. Следуйте инструкциям по активации программы разработчика (когда появится запрос, введите код активации).
- 4 Когда код активации будет принят, перейдите на портал программы разработчиков iPhone в центре разработчиков iPhone.
- 5 Создайте файл запроса на подпись сертификата. Этот файл необходим для получения сертификата на разработку для iPhone. Инструкции см. в разделе «[Генерация запроса на подпись сертификата](#)» на странице 6».
- 6 На следующем этапе будет запрошено предоставление идентификатора устройства (или уникального идентификатора устройства) iPhone. Идентификатор UDID можно получить в программе iTunes:
  - a Подключите iPhone с помощью кабеля USB. Затем в iTunes выберите вкладку «Обзор» для устройства iPhone.
  - b Загрузив файл обеспечения с сайта центра разработчиков iPhone, добавьте его в iTunes.





- 4 В меню программы «Связка ключей» выберите пункт «Ассистент сертификации» > «Запросить сертификат у бюро сертификации».
- 5 Введите адрес электронной почты и имя, указанные в идентификаторе вашего аккаунта разработчика iPhone. Не указывайте адрес бюро сертификации. Нажмите кнопку «Запрос сохранен на диске», затем кнопку «Продолжить».
- 6 Сохраните файл (CertificateSigningRequest.certSigningRequest).
- 7 Отправьте CSR-файл на [веб-сайт разработчиков iPhone](#). (См. раздел «Запрос сертификата разработчика iPhone и создание профиля обеспечения».)

### Формирование запроса на подпись сертификата в ОС Windows

Для тех, кто работает в ОС Windows, будет проще получить сертификат разработчика iPhone, используя компьютер с ОС Mac. Тем не менее, получить сертификат на компьютере с ОС Windows возможно. Сначала нужно создать запрос на подпись сертификата (CSR-файл) с помощью OpenSSL.

- 1 Установите OpenSSL на компьютере Windows. (Перейдите на страницу <http://www.openssl.org/related/binaries.html>.)

Возможно, потребуется также установить файлы Visual C++ 2008 Redistributable, указанные на странице загрузки Open SSL. (Устанавливать Visual C++ на компьютер *не* требуется.)

- 2 Откройте сеанс командной строки Windows и компакт-диск для каталога корзины OpenSSL (например, c:\OpenSSL\bin\).

- 3 Создайте личный ключ, введя следующее в командную строку:

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

Сохраните файл с личным ключом. Он понадобится вам в дальнейшем.

При работе с OpenSSL внимательно читайте сообщения об ошибках. OpenSSL может формировать файлы даже при наличии ошибок, Однако эти файлы могут быть непригодны для использования. Если появляется сообщение об ошибке, проверьте правильность синтаксиса и выполните команду снова.

- 4 Создайте CSR-файл, введя следующее в командную строку:

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

Подставьте свой адрес электронной почты, имя сертификата (CN) и страну (C).

- 5 Отправьте CSR-файл на [веб-сайт разработчиков iPhone](#). (См. раздел «Запрос сертификата разработчика iPhone и создание профиля обеспечения».)

## Преобразование сертификата разработчика в файл P12

Чтобы создавать приложения для iPhone с помощью Flash Professional CS5, необходимо использовать файл сертификата P12. Этот сертификат создается на основе файла сертификата разработчика iPhone, полученного от компании Apple.

### Преобразование сертификата разработчика iPhone в файл P12 в ОС Mac OS

Загрузив сертификат iPhone с сайта Apple, экспортируйте его в виде файла P12. В ОС Mac OS выполните следующие действия.

- 1 Откройте программу «Связка ключей» (каталог Программы/Службные программы).

- 2 Если сертификат еще не добавлен в связку ключей, выберите «Файл» > «Импорт». Найдите файл сертификата (CER-файл), полученный от компании Apple.
- 3 В программе «Связка ключей» выберите категорию «Ключи».
- 4 Выберите личный ключ, связанный с данным сертификатом на разработку iPhone.  
Личный ключ идентифицируется связанным с ним открытым сертификатом «Разработчик iPhone: <имя> <фамилия>».
- 5 Выберите «Файл» > «Экспортировать объекты».
- 6 Сохраните ключ в формате файла обмена личными данными (.p12).
- 7 Появится запрос на создание пароля, который используется при попытке импорта этого ключа на другой компьютер.

### Преобразование сертификата разработчика Apple в файл P12 в ОС Windows

Чтобы создавать приложения iPhone с помощью Flash CS5, необходимо использовать файл сертификата P12. Этот сертификат создается на основе файла сертификата разработчика iPhone, полученного от компании Apple.

- 1 Преобразуйте файл сертификата разработчика, полученный от компании Apple, в файл сертификата PEM. С помощью командной строки запустите следующую операцию из каталога корзины (bin) OpenSSL.

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

- 2 Если используется личный ключ из связки ключей на компьютере с ОС Mac, преобразуйте его в ключ PEM:

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

- 3 Теперь можно создать действительный файл P12 на основе ключа и версии PEM сертификата разработчика iPhone:

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

Если используется ключ из связки ключей в ОС Mac OS, используйте версию PEM, созданную при выполнении предыдущего шага. В противном случае используйте ключ OpenSSL, созданный ранее (в ОС Windows).

## Управление сертификатами, идентификаторами устройств и профилями обеспечения

На веб-сайте центра разработчиков iPhone (<http://developer.apple.com/iphone/>) можно управлять сертификатами, идентификаторами устройств и приложений, а также профилями обеспечения. Перейдите на портал программы разработчиков iPhone на этом веб-сайте.

- По ссылке Certificates (Сертификаты) перейдите к разделу управления своими сертификатами на разработку. Здесь можно создавать, загружать и отзывать сертификаты. Для создания сертификата необходимо сначала создать запрос на его подпись. См. раздел «[Генерация запроса на подпись сертификата](#)» на странице 6».
- По ссылке Devices (Устройства) перейдите к списку устройств, на которые можно устанавливать ваше приложение.
- По ссылке App IDs (Идентификаторы приложений) можно перейти к соответствующему разделу. Каждый создаваемый профиль обеспечения связывается с идентификатором приложения.
- По ссылке Provisioning (Обеспечение) перейдите к разделу управления своими профилями обеспечения. Для создания профилей обеспечения можно воспользоваться Ассистентом обеспечения разработки.

- Для отправки приложения в App Store или создания специальной версии перейдите по ссылке Distribution (Распространение). Здесь вы найдете ссылку на сайт iTunes Connect, с помощью которой можно отправить приложение в App Store.

## Создание приложения Hello World для iPhone с помощью Flash Professional CS5

**Важно.** Прежде чем приступить к созданию приложения, загрузите необходимые для разработки приложения и файлы. См. разделы «[Получение инструментов разработчика у компании Adobe](#)» на странице 4» и «[Получение файлов разработчика у компании Apple](#)» на странице 5».

### Создание проекта Flash Professional CS5

Приложение iPhone можно создавать непосредственно во Flash Professional CS5.

- 1 Откройте Flash CS5.
- 2 Выберите «Файл» > «Создать».
- 3 Выберите iPhone.
- 4 Нажмите кнопку «ОК».

### Добавление содержимого в приложение

После этого добавьте в приложение текст «Здравствуй, мир!», выполнив следующие действия.

- 1 Выберите инструмент «Текст» и щелкните по рабочей области.
- 2 В окне «Свойства» с настройками текстового поля выберите «Классический текст» (а не «Текст TLF»).  
Это базовое приложение, поэтому целесообразно использовать вариант «Классический текст». Чтобы использовать текст TLF, необходимо также настроить некоторые другие параметры. См. раздел «[Шрифты и ввод текста](#)» на странице 42».
- 3 В новом текстовом поле введите «Здравствуй, мир!».
- 4 Выделите текстовое поле при помощи инструмента «Выделение».
- 5 Затем откройте окно «Инспектор свойств» и задайте следующие настройки:
  - «Символ» > «Гарнитура: \_sans»
  - «Символ» > «Размер: 50»
  - «Позиция» > «X: 20»
  - «Позиция» > «Y: 20»
- 6 Сохраните файл.
- 7 Выберите «Управление» > «Тестировать ролик» > «В AIR Debug Launcher (мобильная версия)».

Flash Professional CS5 выполняет компиляцию SWF-содержимого и отображает в AIR Debug Launcher (ADL) версию приложения. Это позволяет быстро просмотреть приложение.

## Создание изображений для значка и начального экрана приложения

Все приложения iPhone имеют значки, которые отображаются в пользовательском интерфейсе приложения iTunes и на экране iPhone.

- 1 Создайте подкаталог в каталоге проекта и присвойте ему имя icons (значки).
- 2 Создайте три PNG-файла в каталоге icons. Сохраните их под именами Icon29.png, Icon57.png и Icon512.png.
- 3 Отредактируйте PNG-файлы, чтобы создать подходящие значки для приложения. Изображения должны иметь размеры 29 x 29, 57 x 57 и 512 x 512 пикселей. В целях тестирования просто используйте в качестве изображений одноцветные квадраты.

Все приложения iPhone отображают начальное изображение в процессе загрузки на iPhone. Определите начальное изображение в PNG-файле.

- 1 В основном каталоге разработки создайте PNG-файл с именем Default.png. (Не помещайте этот файл в подкаталог icons. Файлу нужно обязательно присвоить имя Default.png, с прописной буквой D.)
- 2 Внесите в файл изменения, чтобы его ширина равнялась 320 пикселям, а высота 480 пикселям. Пока в качестве начального изображения можно использовать простой белый прямоугольник. (В дальнейшем его можно будет изменить.)

***Примечание.** При отправке приложения в магазин Apple App Store используется версия JPG (а не PNG) изображения со стороной 512 пикселей. Версия PNG используется при тестировании рабочих версий приложения.*

Дополнительные сведения об этих изображениях см. в разделе «[Значок iPhone и изображения начального экрана](#)» на странице 14».

## Редактирование настроек приложения

**Важно.** Если это еще не сделано, загрузите необходимые инструменты и файлы для разработки приложений iPhone. См. раздел «[Получение файлов разработчика у компании Apple](#)» на странице 5».

Во Flash Professional CS5 откройте диалоговое окно «Настройки iPhone» и задайте многочисленные основные свойства приложения iPhone.

- 1 Выберите «Файл» > «Настройки iPhone OS».
- 2 На вкладке «Общие» задайте следующие настройки.
  - Файл вывода: HelloWorld.ipa  
Это имя создаваемого файла установщика iPhone.
  - Имя приложения: Hello World  
Это имя приложения, отображаемое под значком приложения на экране iPhone.
  - Версия: 1.0  
Версия приложения.
  - Соотношение сторон: книжное
  - Полноэкранный режим: флажок установлен.
  - Автоориентация: флажок снят.
  - Визуализация: ЦП

Другие параметры («Графический процессор» и «Авто») выполняют визуализацию с использованием аппаратного ускорения. Эта функция оптимизирует работу приложений, которым требуется много графических ресурсов (таких как игры), разработанных с учетом использования аппаратного ускорения. Дополнительные сведения см. в разделе «[Аппаратное ускорение](#)» на странице 38».

- Включенные файлы. Добавьте изображение начального экрана (Default.png) в список включенных файлов.

***Примечание.** В этом примере приложения Hello World не изменяйте настройки, указанные в данных инструкциях. Для некоторых настроек, например для настройки «Версия», предусмотрены определенные ограничения. Эти ограничения описаны в разделе «[Настройки приложения iPhone](#)» на странице 16».*

### 3 На вкладке «Развертывание» задайте следующие настройки.

- Сертификат: нажмите кнопку обзора и выберите P12-файл, созданный на основе сертификата разработчика, полученного от компании Apple.

Этот сертификат служит для подписи файла. Сертификат Apple iPhone необходимо преобразовать в P12-формат. Дополнительные сведения см. в разделе «[Получение инструментов разработчика у компании Adobe](#)» на странице 4».

- Пароль: введите пароль для сертификата.
- Файл обеспечения: нажмите кнопку обзора и выберите файл обеспечения разработчика, полученный от компании Apple. См. раздел «[Получение файлов разработчика у компании Apple](#)» на странице 5».
- Идентификатор приложения: если это поле доступно для выделения, в него можно ввести идентификатор приложения, соответствующий тому, который был заявлен при получении файлов разработчика от Apple (например, com.example.as3.HelloWorld).

Этот идентификатор уникален для данного приложения.

Если поле не доступно для выделения, профиль обеспечения связан с определенным идентификатором приложения. Этот идентификатор отображается в поле.

Сведения об указании идентификатора приложения см. в разделе «Вкладка "Развертывание"» статьи «[Настройка свойств приложения iPhone во Flash Professional CS5](#)» на странице 16».

### 4 На вкладке «Значки» выберите «Значок 29 x 29» в списке «Значки». Затем укажите местоположение созданного ранее PNG-файла с размером 29 x 29 пикселей (см. раздел «[Создание изображений для значка и начального экрана приложения](#)» на странице 10»). После этого укажите PNG-файлы для значков с размерами 57 x 57 пикселей и 512 x 512 пикселей.

### 5 Нажмите кнопку «ОК».

### 6 Сохраните файл.

Дополнительные сведения о настройках приложения см. в разделе «[Настройки приложения iPhone](#)» на странице 16».

## Компиляция файла IPA

Теперь можно скомпилировать файл установщика IPA.

### 1 Выберите «Файл» > «Опубликовать».

### 2 В диалоговом окне «Настройки iPhone» нажмите кнопку «ОК».

Инструмент Packager for iPhone создает файл установщика приложения iPhone с именем HelloWorld.ipa в каталоге приложения. Компиляция файла IPA может занять несколько минут.

## Установка приложения на iPhone

Чтобы установить приложение iPhone для тестирования на устройстве iPhone, выполните следующие действия.

- 1 Откройте приложение iTunes.
- 2 Если это еще не сделано, добавьте профиль обеспечения данного приложения в iTunes. В iTunes выберите «Файл» > «Добавить в медиатеку». Выберите файл профиля обеспечения (с типом файла mobileprovision)  
На данном этапе для тестирования приложения на рабочем устройстве iPhone можно использовать профиль обеспечения разработки.  
Позднее, при отправке приложения в iTunes Store, потребуются использовать профиль распространения. Для специального распространения приложения (на несколько устройств без отправки в iTunes Store) нужно использовать специальный профиль обеспечения.  
Дополнительные сведения о файлах обеспечения см. в разделе «[Получение файлов разработчика у компании Apple](#)» на странице 5».
- 3 В некоторых версиях iTunes не выполняется замена, если уже установлена та же версия приложения. В таком случае удалите приложение с устройства и из списка приложений в iTunes.
- 4 Дважды щелкните IPA-файл для приложения. Он должен быть указан в списке приложений.
- 5 Подключите iPhone к порту USB компьютера.
- 6 В iTunes перейдите на вкладку «Программа» для устройства и проверьте, что программа выбрана в списке устанавливаемых.
- 7 Выберите устройство в списке на левой панели программы iTunes. Затем нажмите кнопку «Синхронизировать». По окончании синхронизации приложение Hello World появится на устройстве iPhone.

Если новая версия не установилась, удалите приложение с iPhone и из списка приложений в iTunes, а затем повторите процедуру. Такое происходит, если текущая установленная версия использует тот же идентификатор приложения и номер версии.

Если при установке приложения в iTunes возникают ошибки, см. «Устранение неполадок при установке приложения» в разделе «[Установка приложения iPhone](#)» на странице 25».

## Редактирование изображения начального экрана

Перед компиляцией приложения был создан файл Default.png (см. раздел «[Создание изображений для значка и начального экрана приложения](#)» на странице 10»). Этот PNG-файл служит начальным экраном, который отображается во время загрузки приложения. При тестировании приложения на iPhone вы, возможно, заметили этот пустой экран при загрузке.

Это изображение необходимо изменить, чтобы оно соответствовало начальному экрану приложения («Здравствуй, мир!»).

- 1 Откройте приложение на устройстве. Когда появится первый текст «Здравствуй, мир!», нажмите и удерживайте кнопку «Домой» (под экраном). Удерживая кнопку «Домой», нажмите кнопку «Питание/Режим сна» (вверху на iPhone). В результате будет сделан снимок экрана и отправлен в приложение «Фотопленка».
- 2 Перенесите изображение на рабочий компьютер с помощью функции переноса фотографий из iPhoto или другого приложения для перемещения фотографий. (В ОС Mac OS также можно использовать программу «Захват изображений».)

Также фотографию можно отправить на рабочий компьютер по электронной почте.

- Откройте приложение «Фотографии».
- Откройте инструмент «Фото пленка».
- Откройте сделанный снимок экрана.
- Нажмите изображение, а затем нажмите кнопку со стрелкой («переслать») в нижнем левом углу. Нажмите кнопку «Отправить по e-mail» и отправьте изображение на свой адрес.

**3** Замените файл Default.png (в каталоге проекта) на PNG-версию снимка экрана.

**4** Выполните повторную компиляцию приложения (см. раздел «[Компиляция файла IPA](#)» на странице 11») и переустановите его на iPhone.

Теперь при загрузке приложения отображается новый начальный экран.

***Примечание.** В файле Default.png может использоваться любая графика при условии сохранения правильных размеров (320 x 480 пикселей). Однако в большинстве случаев рекомендуется, чтобы изображение Default.png соответствовало начальному экрану приложения.*



## Глава 2. Компиляция и отладка приложений iPhone

Приложение для iPhone можно скомпилировать с помощью приложения Packager for iPhone. Инструмент Packager for iPhone включен в пакет программ Flash Professional CS5.

Отладку приложения можно выполнять на рабочем компьютере. Также можно установить отладочную версию на iPhone и получать вывод `trace()` во Flash Professional CS5.

Инструкции по созданию приложения iPhone от начала до конца см. в разделе «[Создание приложения Hello World для iPhone с помощью Flash Professional CS5](#)» на странице 9.

### Значок iPhone и изображения начального экрана

Все приложения iPhone имеют значки, которые отображаются в пользовательском интерфейсе приложения iTunes и на экране iPhone.

#### Значки приложения iPhone

Для приложения iPhone необходимо определить следующие значки:

- значок 29 x 29 пикселей используется в результатах поиска Spotlight на iPhone и iPod touch;
- значок 48 x 48 пикселей используется в результатах поиска Spotlight на iPad;
- значок 57 x 57 пикселей используется на экране «Домой» iPhone и iPod touch;
- значок 72 x 72 пикселей — используется на экране «Домой» iPad;
- значок 512 x 512 пикселей отображается в iTunes. PNG-файл изображения 512 x 512 пикселей используется только для тестирования отладочных версий приложения. При отправке окончательной версии приложения в Apple App Store, изображение 512 x 512 пикселей передается отдельно в формате JPG. Оно не включается в IPA-файл.

Во Flash Professional CS5 эти значки добавляются на вкладку «Значки» диалогового окна «Настройки iPhone». См. раздел «[Настройка свойств приложения iPhone во Flash Professional CS5](#)» на странице 16».

Местоположение значков также можно указать в файле дескриптора приложения.

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

На устройстве iPhone значок приобретает эффект блеска. Его не следует включать в исходное изображение. Чтобы убрать эффект блеска, используемый по умолчанию, добавьте следующий код в элемент `InfoAdditions` в файле дескриптора приложения:

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

См. раздел «[Настройка свойств приложения iPhone в файле дескриптора приложения](#)» на странице 18».

### Изображение начального экрана (Default.png)

Все приложения iPhone отображают начальное изображение в процессе загрузки на iPhone. Определите начальное изображение в PNG-файле и сохраните его под именем Default.png. В основном каталоге разработки создайте PNG-файл с именем Default.png. (Не помещайте этот файл в подкаталог. Файлу нужно обязательно присвоить имя Default.png, с прописной буквой D.)

Файл Default.png имеет ширину 320 пикселей и высоту 480 пикселей вне зависимости от исходной ориентации приложения и от того, является ли оно полноэкранным.

Если приложение имеет альбомную ориентацию, используйте те же размеры, что и для книжной: 320 пикселей в ширину и 480 в высоту. Просто поверните само изображение в PNG-файле на 90° против часовой стрелки. Таким образом, левая сторона PNG-изображения отвечает верхнему краю экрана iPhone при альбомной ориентации. (Сведения о настройке исходной ориентации приложения см. в разделе «[Настройки приложения iPhone](#)» на странице 16».)

Если приложение не является полноэкранным, верхние 20 пикселей изображения отсекаются. Вместо них в верхней части отображается полоса состояния iPhone. В приложении с альбомной ориентацией эта область соответствует прямоугольнику шириной 20 пикселей в левой части изображения в файле Default.png (и отображается в верхней части в альбомной ориентации). В приложении с книжной ориентацией эта область представляет собой прямоугольник шириной 20 пикселей в верхней части изображения в файле Default.png.

Для большинства приложений изображение Default.png должно соответствовать экрану запуска приложения. Чтобы сделать снимок начального экрана приложения, выполните следующие действия.

- 1 Откройте приложение на iPhone. Когда появится первый экран пользовательского интерфейса, нажмите и удерживайте клавишу «Домой» (под экраном). Удерживая кнопку «Домой», нажмите кнопку «Питание/Режим сна» (вверху на устройстве). В результате будет сделан снимок экрана и отправлен в приложение «Фотопленка».
- 2 Перенесите изображение на рабочий компьютер с помощью функции переноса фотографий из iPhoto или другого приложения для перемещения фотографий. (В ОС Mac OS также можно использовать программу «Захват изображений».)

Также фотографию можно отправить на рабочий компьютер по электронной почте.

- Откройте приложение «Фотографии».
- Откройте инструмент «Фотопленка».
- Откройте сделанный снимок экрана.
- Нажмите изображение, а затем нажмите кнопку со стрелкой («переслать») в нижнем левом углу. Нажмите кнопку «Отправить по e-mail» и отправьте изображение на свой адрес.

**Примечание.** В файле Default.png может использоваться любая графика при условии сохранения правильных размеров. Однако в большинстве случаев рекомендуется, чтобы изображение Default.png соответствовало начальному экрану приложения.

Не добавляйте текст в изображение Default.png, если приложение локализовано на несколько языков. Изображение Default.png является статическим, и текст не будет соответствовать другим языкам.

Во Flash Professional CS5 обязательно добавьте файл Default.png в список включенных файлов в диалоговом окне «Настройки iPhone». См. раздел «[Настройка свойств приложения iPhone во Flash Professional CS5](#)» на странице 16».

Выполняя компиляцию с использованием приложения PFI в командной строке, обязательно укажите этот файл в списке включенных ресурсов. См. раздел «[Создание установщика приложения iPhone с помощью командной строки](#)» на странице 23».

## Настройки приложения iPhone

Настройки приложения включают:

- Имя приложения
- Имя файла IPA
- Версия приложения
- Ориентация начального экрана приложения и необходимость автоматического изменения ориентации экрана при повороте устройства iPhone
- Необходимость отображения начального окна во весь экран
- Информация о значках приложения
- Информация об аппаратном ускорении

Во Flash Professional CS5 можно отредактировать настройки приложения.

Можно также отредактировать их в файле дескриптора приложения. Файл дескриптора приложения представляет собой файл XML, содержащий настройки для приложения.

### Настройка свойств приложения iPhone во Flash Professional CS5

Во Flash Professional CS5 имеется диалоговое окно «Настройки iPhone», где можно задать многочисленные основные свойства приложения iPhone.

Чтобы открыть диалоговое окно «Настройки iPhone», выполните следующие действия.

- ❖ Выберите «Файл» > «Настройки iPhone».

#### Вкладка «Общие»

На вкладке «Общие» перечислены следующие настройки, связанные с iPhone.

- Файл вывода: название приложения, отображаемое под значком приложения на экране iPhone. Не используйте знак плюса (+) в имени файла вывода.
- Имя приложения: название приложения, отображаемое под значком приложения на экране iPhone. Не используйте знак плюса (+) в имени приложения.

- **Версия:** помогает пользователям определить, какую версию приложения они устанавливают. Она используется как версия пакета CFBundleVersion приложения iPhone. Она должна иметь такой формат, как nnnnn[.nn[.nn]], где n — это цифра 0-9, а в квадратные скобки заключены дополнительные компоненты, такие как 1, 1.0 или 1.0.1. Версии iPhone должны содержать только цифры и десятичные точки. Версии iPhone могут содержать не более двух десятичных точек.
- **Соотношение сторон:** исходное соотношение сторон приложения (книжное или альбомное).
- **Полноэкранный режим:** указывает на то, занимает приложение весь экран или допускает отображение строки состояния iPhone.
- **Автоориентация:** установите этот флажок, чтобы отображаемое содержимое приложения меняло ориентацию вместе с устройством iPhone.

При использовании автоматической установки ориентации для получения наилучших результатов добавьте код ActionScript, чтобы задать для свойства align объекта Stage следующие значения:

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

- **Визуализация:** определяет способ визуализации экранных объектов на iPhone.
  - **ЦП:** приложение использует центральный процессор для визуализации всех экранных объектов. Аппаратное ускорение не используется.
  - **Графический процессор:** приложение использует графический процессор iPhone для совмещения растровых изображений.
  - **Авто:** эта функция не реализована.

Дополнительные сведения см. в разделе «[Аппаратное ускорение](#)» на странице 38».

- **Включенные файлы:** добавьте все файлы и каталоги, которые необходимо упаковать в приложение iPhone. Основной SWF-файл и файл дескриптора приложения включены по умолчанию. Остальные активы нужно добавить в список самостоятельно. Обязательно добавьте изображение начального экрана (Default.png) в список включенных файлов.

### Вкладка «Развертывание»

На вкладке «Развертывание» перечислены настройки подписи и компиляции приложения.

- **Цифровая подпись iPhone:** укажите файл сертификата P12 и пароль для сертификата. Сертификат Apple iPhone необходимо преобразовать в P12-формат. Дополнительные сведения см. в разделе «[Получение файлов разработчика у компании Apple](#)» на странице 5».
- **Файл обеспечения:** укажите файл обеспечения для данного приложения, полученный от компании Apple. Дополнительные сведения см. в разделе «[Получение файлов разработчика у компании Apple](#)» на странице 5».
- **Идентификатор приложения:** уникальный идентификатор вашего приложения. Если файл обеспечения связан с каким-либо идентификатором приложения, Flash Professional CS5 автоматически заполняет это поле, и изменить его вручную нельзя. В противном случае для профиля обеспечения можно использовать несколько идентификаторов приложения. Укажите идентификатор приложения, соответствующий подстановочному шаблону, указанному в заявке на получение файлов у компании Apple:
  - Если ваш идентификатор приложения Apple — com.myDomain.\*, идентификатор приложения в диалоговом окне «Настройки iPhone» должен начинаться с com.myDomain. (например, com.myDomain.myApp или com.myDomain.app22).

- Если идентификатор приложения Apple — \*, идентификатором приложения в диалоговом окне «Настройки iPhone» может быть любая строка действительных символов.

В центре разработчиков iPhone можно найти идентификатор приложения Apple (или подстановочный шаблон идентификатора приложения), связанный с профилем обеспечения (<http://developer.apple.com/iphone>). Перейдите на портал программы разработчиков iPhone, затем в раздел Provisioning (Обеспечение).

**Важно!** Пропустите символы в начале идентификатора приложения Apple. В терминологии Apple эта часть строки называется идентификатором начального числа пакета. Например, если Apple отображает ваш идентификатор приложения как 96LPVWEASL.com.example.bob.myApp, пропустите 96LPVWEASL и укажите только com.example.bob.myApp. Если Apple отображает ваш идентификатор приложения как 5RM86Z4DJM.\*, пропустите 5RM86Z4DJM — это подстановочный шаблон идентификатора приложения.

- Тип развертывания iPhone:
  - Быстрая публикация для проверки устройства: выберите этот вариант, чтобы быстро скомпилировать версию приложения, предназначенную для тестирования на рабочем устройстве iPhone.
  - Быстрая публикация для отладки устройства: выберите этот вариант, чтобы быстро скомпилировать отладочную версию приложения, предназначенную для тестирования на рабочем устройстве iPhone. При выборе этого варианта отладчик Flash Professional CS5 может получать вывод `trace()` от приложения iPhone. (См. раздел «[Отладка приложения iPhone](#)» на странице 26.)
  - Развертывание — специальное: выберите этот вариант, чтобы создать приложение для специального развертывания. Сведения см. в центре разработчиков Apple iPhone
  - Развертывание — Apple App Store: выберите этот вариант, чтобы создать окончательную версию IPA-файла для развертывания в Apple App Store.

#### Вкладка «Значки»

На вкладке «Значки» укажите расположения изображений значков размером 29 x 29, 48 x 48, 57 x 57, 72 x 72 и 512 x 512. См. раздел «[Значок iPhone и изображения начального экрана](#)» на странице 14».

*Примечание.* Параметры значков 48 x 48 и 72 x 72 не включены в пробную версию инструмента Packager for iPhone, поставляемую с Flash Professional CS5. Для их добавления выберите во Flash Professional CS5 меню «Справка» > «Обновления».

## Настройка свойств приложения iPhone в файле дескриптора приложения

Файл дескриптора приложения представляет собой файл XML, содержащий свойства для всего приложения, такие как его имя, версия, авторские права и другие настройки.

Flash Professional CS5 генерирует файл дескриптора приложения на базе настроек, заданных в диалоговом окне «Настройки iPhone». Но файл дескриптора приложения можно отредактировать в текстовом редакторе. Flash Professional генерирует имя для файла дескриптора приложения, добавляя «-app.xml» к имени проекта. Например, файл дескриптора приложения для проекта HelloWorld получает имя HelloWorld-app.xml. Внесите изменения в файл дескриптора приложения, если требуется определить настройки, которых нет в диалоговом окне «Настройки iPhone» во Flash Professional CS5. Например, можно определить элемент `InfoAdditions`, чтобы задать настройки `info.plist` для приложения.

**Важно!** Не редактируйте файл дескриптора приложения, когда открыто диалоговое окно Flash Professional CS5. Прежде чем открывать диалоговое окно «Настройки iPhone», сохраните изменения в файле дескриптора приложения.

Далее приводится образец файла дескриптора приложения.

```
<?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>
```

Ниже приводятся подробные сведения о настройках в этом файле дескриптора приложения.

- Для создания приложений iPhone в элементе `<application>` необходимо указать пространство имен AIR 2.0 `<application xmlns="http://ns.adobe.com/air/application/2.0">`
- Элемент `<id>`:

`<id>com.example.as3.HelloWorld</id>` Идентификатор приложения уникален для данного приложения. Рекомендуется использовать строку, разделенную точками, с использованием обратного стиля DNS, например "com.company.AppName". Компилятор использует это значение в качестве идентификатора пакета для приложения iPhone.

Если файл обеспечения привязан к какому-либо идентификатору приложения, укажите идентификатор в этом элементе. Не указывайте символы, стоящие в начале идентификатора приложения Apple (идентификатор начального числа пакета). Например, если идентификатор приложения профиля обеспечения — 96LPVWEASL.com.example.bob.myApp, укажите в файле дескриптора только com.example.bob.myApp.

Если в файле обеспечения разрешено использовать подстановочные шаблоны, идентификатор приложения заканчивается звездочкой (например, 5RM86Z4DJM.\*). Укажите идентификатор приложения, соответствующий подстановочному шаблону, указанному в заявке на получение файлов у компании Apple.

- Если ваш идентификатор приложения Apple — com.myDomain.\*, идентификатор приложения в файле дескриптора должен начинаться с com.myDomain. Идентификатором приложения может быть, например, com.myDomain.myApp или com.myDomain.app22.

- Если идентификатор приложения Apple — \*, идентификатором приложения в файле дескриптора может быть любая строка действительных символов.

В центре разработчиков iPhone можно найти идентификатор приложения Apple (или подстановочный шаблон идентификатора приложения), связанный с профилем обеспечения (<http://developer.apple.com/iphone>). Перейдите на портал программы разработчиков iPhone, затем в раздел Provisioning (Обеспечение).

**Важно!** Пропустите символы в начале идентификатора приложения Apple. В терминологии Apple эта часть строки называется идентификатором начального числа пакета. Например, если Apple отображает ваш идентификатор приложения как 5RM86Z4DJM.\*, пропустите 5RM86Z4DJM — это подстановочный шаблон идентификатора приложения. Если Apple отображает ваш идентификатор приложения как 96LPVWEASL.com.example.bob.myApp, пропустите 96LPVWEASL и укажите только com.example.bob.myApp.

- Элемент `<filename>`:  
`<filename>HelloWorld</filename>` Это имя, используемое для файла программы установки iPhone. Не используйте знак плюса (+) в имени файла.
- Элемент `<name>`:  
`<name>Hello World</name>` Имя приложения, отображаемое в iTunes и на iPhone. Не используйте знак плюса (+) в имени.
- Элемент `<version>`:  
`<version>1.0</version>` Помогает пользователям определить, какую версию приложения они устанавливают. Она используется как версия пакета `CFBundleVersion` приложения iPhone. Она должна иметь такой формат, как `nnnnn[.nn[.nn]]`, где `n` — это цифра 0-9, а в квадратные скобки заключены дополнительные компоненты, такие как 1, 1.0 или 1.0.1. Версии iPhone должны содержать только цифры и десятичные точки. Версии iPhone могут содержать не более двух десятичных точек.
- Элемент `<initialWindow>` содержит следующие дочерние элементы для определения свойств начального экрана приложения.  
`<content>HelloWorld.swf</content>` Определяет корневой SWF-файл для компиляции приложения iPhone.  
`<visible>true</visible>` Это обязательная настройка.  
`<fullScreen>true</fullScreen>` Указывает, что приложение использует весь экран iPhone.  
`<aspectRatio>portrait</aspectRatio>` Указывает, что в качестве исходного соотношения сторон используется книжная ориентация (а не альбомная). Обратите внимание, что файл `Default.png`, используемый для определения начального экрана должен иметь ширину 320 пикселей и высоту 480 пикселей независимо от этой настройки. (См. раздел «[Значок iPhone и изображения начального экрана](#)» на странице 14».)  
`<autoOrients>true</autoOrients>` (Необязательно) Указывает, изменяется ли автоматически ориентация содержимого в приложении при изменении физического положения самого устройства. Значение по умолчанию — `true`. Чтобы отменить автоматическую ориентацию, вызовите метод `preventDefault()` для события `orientationChanging`, отправляемое объектом Stage. Дополнительные сведения см. в разделе [Установка и обнаружение ориентации экрана](#).

При использовании автоматической установки ориентации для получения наилучших результатов задайте для свойства `align` объекта Stage следующие значения:

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

**<renderMode>gpu</renderMode>** (Необязательно) Режим визуализации, используемый приложением. Может использоваться одно из трех значений.

- `cpu`: приложение использует центральный процессор для визуализации всех экранных объектов. Аппаратное ускорение не используется.
- `gpu`: приложение использует графический процессор iPhone для совмещения растровых изображений.
- `auto`: этот вариант не поддерживается.

Дополнительные сведения см. в разделе «[Аппаратное ускорение](#)» на странице 38».

- Элемент `<profiles>`:

**<profiles>mobileDevice</profiles>** Устанавливает ограничения для приложения, чтобы оно компилировалось в профиль мобильного приложения. Этот профиль в настоящее время поддерживает только приложения iPhone. Имеется три поддерживаемых профиля:

- `desktop`: приложение AIR для компьютера;
- `extendedDesktop`: приложение AIR с поддержкой API собственных процессов;
- `mobileDevice`: приложение AIR для мобильных устройств. В настоящее время единственным поддерживаемым устройством является iPhone.

Назначение для приложения определенного профиля предотвращает его компиляцию с использованием остальных профилей. Если профиль не задан, то компиляция приложения может выполняться для любого профиля. Можно задать несколько профилей, перечислив их в элементе `<profiles>` и разделяя пробелами.

Обязательно включите `mobileDevice` в качестве поддерживаемого профиля (или оставьте элемент `<profiles>` пустым).

- Элемент `<icon>` содержит следующие дочерние элементы для определения значков, используемых приложением:

**<image29x29>icons/icon29.png</image29x29>** Это изображение, используемое в результатах поиска Spotlight.

**<image48x48>icons/icon48.png</image48x48>** Это изображение, используемое в результатах поиска Spotlight на iPad.

**<image57x57>icons/icon57.png</image57x57>** Это изображение используется на экране «Домой» iPhone и iPod touch.

**<image72x72>icons/icon72.png</image72x72>** Это изображение используется на экране «Домой» iPad.

**<image512x512>icons/icon512.png</image512x512>** Это изображение используется в приложении iTunes.

Инструмент Packager for iPhone использует значки размером 29, 57 и 512 пикселей, ссылки на которые содержатся в файле дескриптора приложения. Этот инструмент копирует их в файлы `Icon-Small.png`, `Icon.png` и `iTunesArtwork` соответственно. Чтобы эти файлы не копировались, их можно упаковать непосредственно. В таком случае их нужно упаковать, поместив в каталог, содержащий файл дескриптора приложения, и указать правильные имена и пути.

Изображение со стороной 512 пикселей предназначено только для внутреннего тестирования. При отправке приложения в Apple изображение со стороной 512 пикселей передается отдельно. Оно не включается в IPA-файл. Укажите его, чтобы перед окончательной отправкой убедиться, что оно будет хорошо выглядеть в приложении iTunes.



- Элемент `<iPhone>` содержит следующие дочерние элементы для установки настроек, связанных с iPhone.

`<InfoAdditions></InfoAdditions>` содержит дочерние элементы, которые определяют пары «ключ-значение», чтобы использовать настройки Info.plist для приложения.

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

В этом примере значения задают стиль строки состояния приложения и указывают, что приложению не требуется постоянный доступ к Wi-Fi.

Настройки InfoAdditions заключены в тег CDATA.

Для поддержки iPad добавьте пары «ключ-значение» в `UIDeviceFamily`. Параметр `UIDeviceFamily` является массивом строк. Каждая строка определяет поддерживаемые устройства. Параметр `<string>1</string>` определяет поддержку iPhone и iPod Touch. Параметр `<string>2</string>` определяет поддержку iPad. Если указать только одну из этих строк, будет поддерживаться только одно семейство устройств. Например, следующий параметр означает, что поддерживается только iPad:

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

Следующий параметр означает, что поддерживаются оба семейства устройств (iPhone/iPod Touch и iPad):

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

Сведения о других настройках Info.plist см. в документации для разработчиков Apple.

## Компиляция файла программы установки приложения iPhone (IPA)

Используйте инструмент Packager for iPhone для компиляции приложения ActionScript-3.0 в файл программы установки IPA.

### Создание программы установки приложения iPhone с помощью инструмента Packager for iPhone, включенного в пакет Flash Professional CS5

Для использования инструмента Packager for iPhone, включенного в пакет Flash Professional CS5, выполните следующее:

- 1 Выберите «Файл» > «Опубликовать».

- 2 В диалоговом окне «Настройки iPhone» убедитесь, что заданы значения для всех настроек. Проверьте, что на вкладке «Развертывание» выбраны правильные параметры. См. раздел «[Настройка свойств приложения iPhone во Flash Professional CS5](#)» на странице 16».
- 3 Нажмите кнопку «Опубликовать».

Инструмент Packager for iPhone генерирует файл установщика приложения iPhone (IPA). Компиляция файла IPA может занять несколько минут.

Инструмент Packager for iPhone можно также запустить из командной строки. См. раздел «[Создание установщика приложения iPhone с помощью командной строки](#)» на странице 23».

## Создание установщика приложения iPhone с помощью командной строки

Запустить инструмент Packager for iPhone можно с помощью командной строки. Packager for iPhone преобразует байт-код SWF-файла и других исходных файлов в приложение формата iPhone.

- 1 Откройте командную строку или терминал и перейдите к папке проекта приложения iPhone.
- 2 Затем с помощью инструмента pfi создайте IPA-файл, используя следующий синтаксис:

```
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
```

Измените ссылку на приложение pfi, указав полный путь к нему. Приложение pfi устанавливается в подкаталог pfi/bin каталога установки Flash Professional CS5.

Выберите параметр `-target` в соответствии с типом приложения iPhone, которое требуется создать.

- `-target ipa-test`: выберите этот вариант, чтобы быстро скомпилировать версию приложения для тестирования на рабочем устройстве iPhone.
- `-target ipa-debug`: выберите этот вариант, чтобы скомпилировать отладочную версию приложения для тестирования на рабочем устройстве iPhone. При выборе этого варианта можно использовать отладочный сеанс для получения вывода `trace()` от приложения iPhone.

Для указания IP-адреса компьютера, на котором выполняется отладка, добавьте один из следующих параметров `-connect` (CONNECT\_OPTIONS):

- `-connect` — программа будет пытаться подключиться к сеансу отладки на компьютере, используемом для разработки, для компиляции программы.
- `-connect IP_ADDRESS` — программа будет пытаться подключиться к сеансу отладки на компьютере с указанным IP-адресом. Например:  

```
-target ipa-debug -connect 192.0.32.10
```
- `-connect HOST_NAME` — программа будет пытаться подключиться к сеансу отладки на компьютере с указанным именем хоста. Например:  

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

**Примечание.** Параметр `-connect` не включен в пробную версию инструмента Packager for iPhone, входящую в комплект Flash Professional CS5. Обновите Packager for iPhone. Для этого выберите во Flash Professional CS5 меню «Справка» > «Обновления».

Параметр `-connect` не является обязательным. Если он не задан, программа отладки не будет пытаться подключиться к отладчику на хосте.

Если не удастся подключиться к отладчику, программа отображает диалоговое окно, в котором пользователь должен ввести IP-адрес хоста, на котором производится отладка. Если устройство не подключено к сети Wi-Fi, установить подключение может быть невозможно. Это возможно и в случае, если устройство подключено, но не защищено брандмауэром хоста, на котором производится отладка.

Дополнительные сведения см. в разделе «[Отладка приложения iPhone](#)» на странице 26».

Для использования функции диагностики визуализации с использованием графического процессора можно включить параметр `-renderingdiagnostics`. Дополнительные сведения см. в подразделе «Отладка с помощью функции диагностики визуализации с использованием графического процессора» раздела «[Отладка приложения iPhone](#)» на странице 26».

- `-target ipa-ad-hoc`: выберите этот вариант, чтобы создать приложения для специального развертывания. Сведения см. в центре разработчиков Apple iPhone
- `-target ipa-app-store`: выберите этот вариант, чтобы создать окончательную версию IPA-файла для развертывания в Apple App Store.

Замените строку `PROFILE_PATH` на путь к файлу профиля обеспечения своего приложения.

Дополнительные сведения о профилях обеспечения см. в разделе «[Получение файлов разработчика у компании Apple](#)» на странице 5».

Вместо `SIGNING_OPTIONS` укажите ссылку на сертификат и пароль разработчика iPhone. Пользуйтесь следующим синтаксисом:

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

Вместо `P12_FILE_PATH` укажите путь к файлу сертификата P12. Вместо `PASSWORD` укажите пароль сертификата. (См. пример ниже.) Дополнительные сведения о файле сертификата P12 см. в разделе «[Преобразование сертификата разработчика в файл P12](#)» на странице 7».

Вместо `APP_DESCRIPTOR` укажите файл дескриптора приложения.

Вместо `SOURCE_FILES` укажите основной SWF-файл проекта, а затем остальные ресурсы, которые требуется включить. Укажите пути ко всем файлам значков, определенным в диалоговом окне «Настройки iPhone» во Flash CS5 или в пользовательском файле дескриптора приложения. Также добавьте изображение начального экрана (Default.png).

Рассмотрим следующий пример.

```
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"
```

Данный код компилирует файл HelloWorld.ipa с использованием следующих ресурсов:

- специальный сертификат PKCS#12, использующий пароль сертификата dfb7VKL19;
- файл дескриптора приложения HelloWorld-app.xml;
- исходный файл HelloWorld.swf;
- созданный файл Default.png и файлы значков.

Упаковщик pfi компилирует приложение в IPA-файл на основе файла дескриптора, SWF-файла и других активов.

В ОС Mac OS можно использовать сертификат, сохраненный в программе «Цепочка ключей». Для этого добавьте в команду pfi следующие параметры.

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

Вместо *ALIAS\_NAME* укажите псевдоним сертификата, который необходимо использовать. Когда вы указываете на сертификат, сохраненный программе «Цепочка ключей» Mac, вместо указания расположения сертификата необходимо задать псевдоним.

## Установка приложения iPhone

Для установки приложения разработки на iPhone необходимо добавить на iPhone профиль обеспечения и только потом можно установить приложение.

### Добавление на iPhone профиля обеспечения

Для добавления на iPhone профиля обеспечения выполните следующие действия.

- 1 В iTunes выберите «Файл» > «Добавить в медиатеку». Выберите файл профиля обеспечения (с расширением mobileprovision).

Убедитесь, что iPhone добавлен в профиль обеспечения. На веб-сайте центра разработчиков iPhone (<http://developer.apple.com/iphone/>) можно управлять профилями обеспечения. Перейдите на портал программы разработчиков iPhone на этом веб-сайте. По ссылке Devices (Устройства) перейдите к списку устройств, на которые можно устанавливать ваше приложение. По ссылке Provisioning (Обеспечение) перейдите к разделу управления своими профилями обеспечения.

- 2 Подключите iPhone к компьютеру и выполните синхронизацию.

Сведения о получении профиля обеспечения см. в разделе «[Получение файлов разработчика у компании Apple](#)» на странице 5».

### Установка приложения

Приложение разработки устанавливается так же, как любой другой IPA-файл.

- 1 Если ранее какая-либо версия приложения уже была установлена, удалите ее с устройства и из списка приложений в iTunes.
- 2 Добавьте приложение в iTunes одним из следующих способов.
  - В меню «Файл» (в iTunes) выберите команду «Добавить в медиатеку». Затем выберите IPA-файл и нажмите кнопку «Открыть».
  - Дважды щелкните по IPA-файлу.
  - Перетащите IPA-файл в медиатеку iTunes.
- 3 Подключите iPhone к порту USB компьютера.
- 4 В iTunes перейдите на вкладку «Программа» для устройства и проверьте, что программа выбрана в списке устанавливаемых.
- 5 Выполните синхронизацию iPhone.

### Устранение неполадок при установке приложения

Если при установке приложения в iTunes возникают ошибки, проверьте следующие условия.

- Убедитесь, что идентификатор устройства добавлен в профиль обеспечения.
- Чтобы убедиться, что профиль обеспечения установлен, можно перетащить его в программу iTunes или использовать команду «Добавить в медиатеку» в меню «Файл».

Также проверьте, совпадает ли идентификатор приложения с идентификатором приложения Apple.

- Если ваш идентификатор приложения Apple — `com.myDomain.*`, идентификатор приложения в файле дескриптора или в диалоговом окне «Настройки iPhone» должен начинаться с `com.myDomain` (например, `com.myDomain.anythinghere`).
- Если идентификатор приложения Apple — `com.myDomain.myApp`, идентификатором приложения в файле дескриптора или пользовательском интерфейсе Flash Profession CS5 должен быть `com.myDomain.myApp`.
- Если идентификатор приложения Apple — `.*`, идентификатор приложения в файле дескриптора или пользовательском интерфейсе Flash Profession CS5 может быть любым.

Идентификатор приложения задается в диалоговом окне «Настройки iPhone» во Flash Professional CS5 или в файле дескриптора приложения.

## Отладка приложения iPhone

Приложение, запущенное в ADL, можно отладить на компьютере, используемом для разработки. Можно также отладить приложение в устройстве iPhone.

Некоторые функциональные возможности AIR, которые не поддерживаются в устройстве iPhone, по-прежнему доступны при тестировании приложения с помощью ADL (на компьютере, используемом для разработки). Помните об этих различиях при тестировании содержимого на настольном компьютере. Дополнительные сведения см. в разделе «[API-интерфейсы ActionScript 3.0, неподдерживаемые на мобильных устройствах](#)» на странице 30».

### Отладка приложения на компьютере, используемом для разработки

Чтобы отладить приложение на компьютере, используемом для разработки, в среде Flash Professional CS5, выполните следующие действия:

- ❖ Выберите «Отладка» > «Отладка ролика» > «В AIR Debug Launcher (мобильная версия)».

Можно также отладить приложение, вызвав ADL из командной строки. Синтаксис:

```
adl -profile mobileDevice appDescriptorFile
```

Вместо `appDescriptorFile` укажите путь к файлу дескриптора приложения.

Обязательно выберите параметр `-profile mobileDevice`.

### Отладка приложения в устройстве iPhone

Чтобы отладить приложение на iPhone, выполните следующие действия.

- 1 Выполните компиляцию приложения, предусмотрев возможность отладки.
  - Во Flash Professional CS5 выполните компиляцию, выбрав вариант «Быстрая компиляция для отладки устройства». (См. раздел «[Создание программы установки приложения iPhone с помощью инструмента Packager for iPhone, включенного в пакет Flash Professional CS5](#)» на странице 22».)
  - С помощью приложения командной строки PFI выполните компиляцию программы, используя параметр `target ipa-debug`. (См. раздел «[Создание установщика приложения iPhone с помощью командной строки](#)» на странице 23».)
- 2 Установите приложение на iPhone.
- 3 Включите модуль Wi-Fi на iPhone подключитесь к сети, в которой находится компьютер разработчика.

- 4 Начните сеанс отладки на рабочем компьютере. Во Flash Professional CS5 выберите «Отладка» > «Начать сеанс удаленной отладки» > «Файл ActionScript 3.0».
- 5 Запустите приложение на iPhone.

Отладочная версия приложения запрашивает IP-адрес рабочего компьютера. Введите IP-адрес и нажмите кнопку «ОК». Чтобы получить IP-адрес рабочего компьютера, выполните следующие действия.

- В ОС Mac OS откройте меню Apple и выберите «Системные настройки». В окне «Системные настройки» нажмите на значок «Сеть». В окне «Настройки сети» указан IP-адрес.
- В ОС Windows откройте командную строку и выполните команду `ipconfig`.

Сеанс отладки отображает весь вывод `trace()`, получаемый от приложения.

При отладке приложения, установленного на iPhone, Flash Professional CS5 поддерживает все функции отладки, в том числе проверку точек останова, пошаговый просмотр кода и мониторинг переменных.

#### **Отладка с помощью функции диагностики визуализации с использованием графического процессора**

Функция диагностики визуализации с использованием графического процессора позволяет проверить, как программа работает с аппаратным ускорением (это касается программ с режимом визуализации с использованием графического процессора). Для использования этой функции скомпилируйте программу с помощью инструмента PFI в командой строке и включите параметр `-renderingdiagnostics`:

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

Флаг `-renderingdiagnostics` должен стоять непосредственно за флагом `-package`.

Функция диагностики визуализации с использованием графического процессора отмечает все экранные объекты цветными прямоугольниками со следующими значениями:

- Синий — экранный объект не является растровым или кэшированным как растровый. Он визуализируется.

Если экранный объект, который не изменяется, постоянно отмечается синим цветом, возможно, он пересекается с подвижными экранными объектами. Например, этот объект может использоваться в фоне движущихся экранных объектов. Попробуйте кэшировать этот экранный объект как растровый.

Если синим отмечается объект, который уже кэширован, возможно, используется эффект, не поддерживаемый графическим процессором. К таким эффектам относятся некоторые режимы наложения, преобразования цветов, свойство `scrollRect` и маски.

Программа также отмечает синим экранные объекты, загруженные в графический процессор, размер которых превышает свободную память.

Программа регистрирует в журнале сообщения для каждого синего прямоугольника. Они выводятся вместе с остальными сообщениями `trace()` и отладочными сообщениями.

- Зеленый — экранный объект является растровым или кэшированным как растровый. Он загружается в графический процессор впервые.

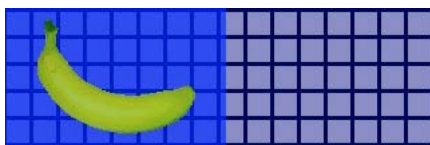
Если экранный объект постоянно отмечается зеленым цветом, значит, код программы воссоздает его. Это происходит, например, если временная шкала возвращается к кадру, в котором создается экранный объект. Попробуйте изменить содержимое так, чтобы одинаковые объекты не создавались повторно.

- Красный — экранный объект является растровым или кэширован как растровый. Он загружается в графический процессор повторно.

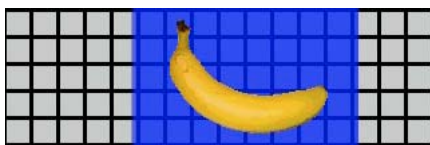
Экранный объект отмечается красным каждый раз, когда он изменяется настолько, что программе необходимо выполнять повторную визуализацию растрового представления. Например, если 2D-объекту не задано свойство `cacheAsBitmapMatrix`, он повторно визуализируется при масштабировании или повороте. Повторная визуализация также выполняется при движении или изменении дочернего экранного объекта.

Каждый цветной прямоугольник бледнеет после четырех циклов перерисовки экрана (если в течение этих четырех циклов не возникают новые условия для окрашивания). Однако, если на экране ничего не меняется, диагностическая окраска тоже не меняется.

Представим, что у нас есть экранный объект — банан, размещенный на векторном фоне, который не кэширован как растровое изображение. При первой визуализации банан окрашивается в зеленый. Фон при первой визуализации окрашивается в синий:

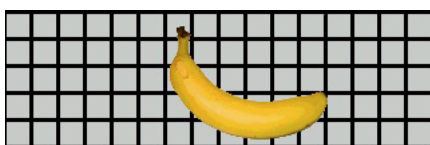


При движении банана ЦП приходится повторно визуализировать фон, поэтому он отмечается синим цветом:



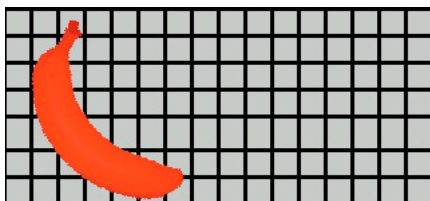
Синяя окраска фона указывает на перерисованные области, которые необходимо отправлять для обработки в графический процессор.

Тем не менее, если фон кэширован как растровое изображение, то при движении банана функция диагностики не будет окрашивать его:



Окраска не появляется, потому что растровое изображение фона сохраняется в графическом процессоре. Графический процессор может наложить банан на фон без обращения к ЦП.

Предположим, банан — это экранный 2D-объект, для которого не задано свойство `cacheAsBitmapMatrix`. При вращении или масштабировании экранного объекта функция диагностики визуализации окрашивает его в красный. Это означает, что программе приходится загружать в графический процессор новую версию экранного объекта:



## Отправка приложения для iPhone в магазин App Store

Чтобы отправить приложение в магазин App Store, выполните следующие действия:

- 1 Получите сертификат на распространение и профиль обеспечения на веб-сайте iPhone Dev Center (<http://developer.apple.com/iphone/>).

Сертификат на распространение называется «iPhone Developer: XXX.cer», где XXX — ваше имя.

Дополнительные сведения см. в разделе «[Получение файлов разработчика у компании Apple](#)» на странице 5».

- 2 Преобразуйте сертификат на распространение в файл P12.

Дополнительные сведения см. в разделе «[Преобразование сертификата разработчика в файл P12](#)» на странице 7».

- 3 Скомпилируйте приложение с использованием файла P12 и профиля обеспечения.

Используйте файл P12, созданный на основе сертификата на распространение. Используйте идентификатор приложения, связанный с профилем обеспечения распространения.

Дополнительные сведения см. в разделе «[Компиляция файла программы установки приложения iPhone \(IPA\)](#)» на странице 22».

- 4 Измените тип файла программы установки с .ipa на .zip. (Например, измените его с MyApp.ipa на MyApp.zip.)

- 5 Отправьте приложение на веб-сайт iPhone Dev Center (<http://developer.apple.com/iphone/>).



# Глава 3. Поддержка API-интерфейсов ActionScript 3.0 для мобильных устройств

При создании приложений AIR для мобильных устройств можно использовать те же API-интерфейсы ActionScript 3.0, что и для разработки компьютерных приложений Adobe Flash Player 10.1 и AIR 2. Тем не менее, существует несколько исключений и дополнений.

## API-интерфейсы ActionScript 3.0, неподдерживаемые на мобильных устройствах

Некоторые API-интерфейсы ActionScript 3.0 недоступны для приложений с профилем мобильного устройства, таких как приложения iPhone.

Когда один и тот же код ActionScript используется с целью создания приложений для разных профилей (например, для профиля рабочего стола и мобильного устройства), необходимо проверить с помощью кода, поддерживается ли тот или иной API-интерфейс. Например, класс `NativeWindow` не поддерживается для приложений iPhone. (Приложения iPhone не могут использовать или создавать собственные окна.) Чтобы проверить, поддерживает используемый профиль приложения собственные окна (например, профиль рабочего стола), проверьте свойство `NativeWindow.isSupported`.

В следующей таблице перечислены API-интерфейсы, не поддерживаемые в профиле мобильных устройств. В ней также приводятся свойства, проверив которые можно узнать, поддерживает ли текущая платформа тот или иной API-интерфейс.

API	Проверка поддержки
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

API	Проверка поддержки
NativeApplication.isSetAsDefaultApplication()	NativeApplication.supportsDefaultApplication
NativeApplication.startAtLogin	NativeApplication.supportsStartAtLogin
NativeMenu	NativeMenu.isSupported
NativeProcess	NativeProcess.isSupported
NativeWindow	NativeWindow.isSupported
NativeWindow.notifyUser()	NativeWindow.supportsNotification
NetworkInfo	NetworkInfo.isSupported
Поддержка PDF	HTMLLoader.pdfCapability
PrintJob	PrintJob.isSupported
SecureSocket	SecureSocket.isSupported
ServerSocket	ServerSocket.isSupported
Shader	—
ShaderFilter	—
StorageVolumeInfo	StorageVolumeInfo.isSupported
XMLSignatureValidator	XMLSignatureValidator.isSupported

Для мобильных устройств нельзя создавать приложения AIR с использованием кода HTML и JavaScript.

Некоторые классы ActionScript 3.0 поддерживаются только частично:

### File

Приложения iPhone имеют доступ только к каталогу приложения и каталогу хранения приложений. Также можно вызывать методы `File.createTempFile()` и `File.createTempDirectory()`. Вызов операции для доступа к другому каталогу (например, метод чтения или записи `FileStream`) вызывает исключение `IOError`.

Приложения iPhone не поддерживают собственные диалоговые окна обозревателя файлов, например создаваемые методом `File.browseForOpen()`.

### Loader

В приложении iPhone нельзя использовать метод `Loader.load()`. Однако нельзя выполнять какой-либо код ActionScript в содержимом SWF, загруженном с помощью метода `Loader.load()`. Но можно использовать активы в SWF-файле (такие как фрагменты роликов, изображения, шрифты и звуки в библиотеке). Для загрузки файлов изображений можно также использовать метод `Loader.load()`.

### Видео

В приложениях AIR на iPhone поддерживаются только видекодеки Sorensen и ON2 VP6.

С помощью метода `navigateToURL()` можно открыть видео в формате H.264 за пределами приложения. В качестве параметра `request` необходимо передать объект `URLRequest`, содержащий URL-ссылку на видео. Воспроизведение видео запускается в видеопроигрывателе iPhone.

### Текстовые поля

В устройстве iPhone на использование шрифтов и других настроек текстовых полей накладываются ограничения. См. раздел «[Шрифты и ввод текста](#)» на странице 42».

### Неподдерживаемые API-интерфейсы и отладка с использованием ADL

Некоторые функциональные возможности AIR, которые не поддерживаются в устройстве iPhone, по-прежнему доступны при тестировании приложения с помощью ADL (на компьютере, используемом для разработки). Помните об этих различиях при тестировании содержимого с использованием ADL.

Эти функциональные возможности включают следующие видео- и аудиокодеки: Speex (аудио), H.264/AVC (видео) и AAC (аудио). Эти кодеки недоступны для приложений AIR, выполняющихся в устройстве iPhone. Однако они работают как обычно на настольном компьютере.

Поддержка расширенного доступа и средства чтения с экрана работает в ADL на платформе Windows. Однако эти API-интерфейсы не поддерживаются в устройстве iPhone.

Протокол RTMPE работает как обычно при использовании из ADL на настольном компьютере. Однако подключение NetConnection с использованием протокола RTMPE в устройстве iPhone выполняется неудачно.

Класс Loader работает без дополнительных ограничений, когда содержимое выполняется с использованием ADL. Однако при выполнении в устройстве iPhone попытки загрузить содержимое SWF с байт-кодом ActionScript заканчиваются появлением сообщения об ошибке.

Экземпляры шейдера выполняются в ADL. Однако в устройстве iPhone байт-код Pixel Bender не интерпретируется и шейдеры не имеют графического эффекта.

Дополнительные сведения см. в разделе «[Отладка приложения iPhone](#)» на странице 26».

## Специальные API-интерфейсы ActionScript для мобильных устройств

Следующие API-интерфейсы доступны только в приложениях AIR на мобильных устройствах. В данный момент они не поддерживаются в Flash Player и в компьютерных версиях AIR.

### API-интерфейс ориентации экрана

API-интерфейс ориентации экрана позволяет манипулировать ориентацией рабочей области и устройства iPhone:

- `Stage.autoOrients`: указывает, настроено ли в программе автоматическое изменение ориентации при повороте устройства. Это свойство имеет значение `true`, когда в диалоговом окне «Настройки iPhone» во Flash Professional CS5 установлен флажок «Автоориентация». (Также можно задать элементу `autoOrients` значение `true` в файле дескриптора приложения.) См. раздел «[Настройки приложения iPhone](#)» на странице 16». Чтобы отменить автоматическое изменение ориентации, нужно добавить прослушатель события `orientationChanging` для объекта `Stage`. Вызов метода `preventDefault()` для этого объекта события отменяет автоматическое изменение ориентации.

При использовании автоматической установки ориентации для получения наилучших результатов задайте для свойства `align` объекта `Stage` следующие значения:

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

- `Stage.deviceOrientation`: физическая ориентация устройства. Класс `StageOrientation` определяет значения для этого свойства.
- `Stage.orientation`: текущая ориентация рабочей области. Класс `StageOrientation` определяет значения для этого свойства.
- `Stage.supportsOrientationChange`: имеет значение `true` в приложениях iPhone и `false` в приложениях AIR.
- `Stage.setOrientation()`: устанавливает ориентацию рабочей области. У этого метода один параметр — строка, в которой определена ориентация рабочей области. Возможные значения параметра определяются постоянными класса `StageOrientation`.
- `StageOrientation`: определяет значения для ориентации рабочей области. Например, `StageOrientation.ROTATED_RIGHT` означает, что рабочая область повернута вправо относительно ориентации по умолчанию.
- `StageOrientationEvent`: определяет события, которые объект `Stage` отправляет при изменении ориентации экрана. Это событие происходит, когда пользователь поворачивает iPhone. Существует два типа событий. Объект `Stage` отправляет событие `orientationChanging` в процессе поворота устройства. Чтобы предотвратить изменение ориентации рабочей области, необходимо вызвать метод `preventDefault()` объекта события `orientationChanging`. Объект `Stage` отправляет событие `orientationChange` после завершения изменения ориентации.

В настоящее время API-интерфейс ориентации экрана применим только в приложениях AIR на мобильных устройствах. Если в программе AIR для мобильных устройств и в программе AIR для настольного компьютера используется один и тот же исходный код, задайте свойство `Stage.supportsOrientationChange`, чтобы проверить, поддерживается ли API-интерфейс.

В следующем примере показана реакция на поворот устройства:

```
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;
    }
}
```

В этом примере при изменении ориентации рабочей области вместо функционального кода использованы комментарии.

Изменить ориентацию рабочей области можно, вызвав метод `setOrientation()` объекта `Stage`. Установка ориентации является асинхронной операцией. Проверить завершение установки ориентации можно, прослушав событие `orientationChange`. Следующий код показывает способ установки для рабочей области правосторонней ориентации:

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

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

При повороте рабочей области изменяются ее размеры, и объект `Stage` отправляет событие `resize`. В ответ на событие `resize` можно изменять размер или местоположение экранных объектов на рабочей области.

### NativeApplication.systemIdleMode и SystemIdleMode

Свойство `NativeApplication.systemIdleMode` запрещает iPhone переходить в режим ожидания. По умолчанию iPhone переходит в режим ожидания, если пользователь какое-то время не прикасается к сенсорному экрану. В режиме ожидания экран гаснет. Кроме того, iPhone при этом может перейти в режим блокировки. Для данного свойства можно установить одно из двух значений:

- `SystemIdleMode.NORMAL` — iPhone следует обычной процедуре перехода в режим ожидания.
- `SystemIdleMode.KEEP_AWAKE` — приложение запрещает iPhone переходить в режим ожидания.

Эта функциональность поддерживается только на мобильных устройствах. Она не поддерживается в программах AIR, выполняемых в настольных операционных системах. Если приложение предназначено для настольных платформ, свойство `NativeApplication.systemIdleMode` не играет роли.

Следующий код позволяет отключить режим ожидания на iPhone:

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

### CameraRoll

Класс `CameraRoll` позволяет добавлять изображения в программу «Фотопленка» iPhone. Метод `addBitmapData()` добавляет изображение в программу «Фотопленка» устройства iPhone. У этого метода один параметр — `bitmapData`. Этот параметр представляет собой объект `BitmapData`, содержащий изображение, которое добавляется в программу «Фотопленка».

Функциональность `CameraRoll` поддерживается только на мобильных устройствах. Она не поддерживается в программах AIR, выполняемых в настольных операционных системах. Чтобы узнать во время выполнения, поддерживает ли ваше приложение функции класса `CameraRoll`, проверьте статическое свойство `CameraRoll.supportsAddBitmapData`.

После вызова метода `addBitmapData()` объект `CameraRoll` отправляет одно из следующих двух событий:

- `complete` — операция успешно завершена.
- `error` — возникла ошибка. Например, на iPhone оказалось недостаточно места для сохранения изображения.

Следующий код добавляет в программу «Фотопленка» изображение рабочей области (снимок экрана):

```
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

Свойство `cacheAsBitmapMatrix` — это объект `Matrix`, определяющий способ визуализации экранного объекта, если свойство `cacheAsBitmap` имеет значение `true`. Эта матрица используется приложением в качестве матрицы преобразования для визуализации растровой версии экранного объекта.

Когда задано свойство `cacheAsBitmapMatrix`, приложение сохраняет не экранную матрицу, а кэшированное растровое изображение, полученное с помощью этой матрицы. (Экранная матрица — это значение `transform.concatenatedMatrix` экранного объекта.) Если матрица не совпадает с экранной, растровое изображение масштабируется и вращается по мере необходимости.

Визуализация экранного объекта, для которого задано свойство `cacheAsBitmapMatrix`, выполняется только при изменении значения `cacheAsBitmapMatrix`. Растровое изображение масштабируется и вращается согласно размеру и ориентации экранной матрицы.

Свойство `cacheAsBitmapMatrix` ускоряет прежде всего визуализация, выполняя за счет ресурсов графического процессора, однако и визуализация с использованием ЦП выполняется быстрее.

**Примечание.** Для применения аппаратного ускорения задайте для визуализации значение «Графический процессор» (во *Flash Professional CS5* откройте диалоговое окно «Настройки iPhone» и перейдите на вкладку «Общие»). (Также можно задать свойству `renderMode` значение `gpu` в файле дескриптора приложения.)

Например, следующий код оперирует растровым представлением экранного объекта в неизменном виде.

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

В этом коде растровое представление соответствует текущей визуализации.

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

Как правило, используется единичная матрица (`new Matrix()`) или суффиксы `transform.concatenatedMatrix`. Однако можно использовать и другую матрицу. Например, для загрузки в графический процессор другого растрового изображения можно применить уменьшенную матрицу. В показанном ниже примере используется матрица `cacheAsBitmapMatrix`, уменьшенная в 2 раза по осям *x* и *y*. Растровый объект, используемый графическим процессором, меньше, однако графический процессор адаптирует его размер согласно свойству `transform.matrix` экранного объекта:

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

В большинстве случаев следует выбирать матрицу, преобразующую экранный объект до размера, который будет использоваться в приложении. Например, если в приложении используется растровая версия спрайта, уменьшенная в половину, матрицу следует тоже сделать в половину меньше. Если спрайт в приложении будет больше своего исходного размера, используйте матрицу, увеличенную в соответствующее количество раз.

Размеры экранных объектов, для которых задано свойство `cacheAsBitmapMatrix`, не должны превышать 1020 x 1020 пикселей. Кроме того, суммарное количество пикселей всех экранных объектов, для которых задано свойство `cacheAsBitmapMatrix`, не должно превышать четыре миллиона.

При совместном использовании свойства `cacheAsBitmapMatrix` и аппаратного ускорения следует брать в расчет многие факторы. Важно понимать, для каких экранных объектов следует, а для каких не следует задавать это свойство. Важные сведения об использовании этого свойства см. в разделе «[Аппаратное ускорение](#)» на странице 38».

Функция диагностики визуализации с использованием графического процессора позволяет проверить работу в отладочных версиях программы. Дополнительные сведения см. в разделе «[Отладка приложения iPhone](#)» на странице 26».

### Замечания о взаимодействии по сети

При использовании следующих схем URL-адресов с функцией `navigateToURL()` документ открывается во внешнем приложении:

Схема URL-адресов	Результат вызова функции <code>navigateToURL()</code>	Пример
Адрес получателя <code>mailto:</code>	Открытие нового сообщения в приложении почтового клиента.	<pre>str = "mailto:test@example.com"; var urlReq:URLRequest = new URLRequest(str); navigateToURL(urlReq);</pre>
<code>sms:</code>	Открытие сообщения в приложении для обмена текстовыми сообщениями.	<pre>str = "sms:1-415-555-1212"; var urlReq:URLRequest = new URLRequest(str); navigateToURL(urlReq);</pre>
<code>tel:</code>	Набор номера телефона (после одобрения пользователем).	<pre>str = "tel:1-415-555-1212"; var urlReq:URLRequest = new URLRequest(str); navigateToURL(urlReq);</pre>

Приложение iPhone может полагаться на установленные самозаверенные корневые сертификаты для проверки подлинности сервера во время защищенной транзакции, например запроса `https`. Сервер должен отправлять не только сертификат листа, но и все промежуточные сертификаты, образующие связку с корневым сертификатом.

## API-интерфейсы ActionScript 3.0, которые представляют особый интерес для разработчиков приложений для мобильных устройств

Следующие API-интерфейсы ActionScript 3.0 определяют функциональную возможность, полезную на мобильных устройствах.

### API-интерфейс акселерометра

Следующие классы позволяют приложению получать события от акселерометра устройства:

- Accelerometer
- AccelerometerEvent

Дополнительные сведения см. в разделе [Ввод акселерометра](#).

### API-интерфейс географического местоположения

Следующие классы позволяют приложению получать события от датчика местоположения устройства:

- Geolocation
- GeolocationEvent

Дополнительные сведения см. в разделе [Геопозиционирование](#).

### API-интерфейс ввода прикосновением, несколькими прикосновениями или жестами

Следующие классы позволяют приложению получать события прикосновения и жестов:

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

Дополнительные сведения см. на в разделе [Ввод прикосновением, несколькими прикосновениями или жестами](#).



# Глава 4. Особенности дизайна приложений iPhone

Скорость обработки и размер экрана iPhone требуют особого подхода к разработке дизайна и кода. Однако многие вопросы, связанные с дизайном, можно отнести ко всем приложениям для любых мобильных устройств.

Дополнительные сведения об оптимизации приложений см. в разделе [Оптимизация содержимого для платформы Flash Platform](#). В это документе приводится много рекомендаций по оптимизации производительности мобильного содержимого, содержимого Flash Player и AIR, а также содержимого на базе ActionScript в целом. Большинство из этих рекомендаций также применимы к приложениям на базе AIR, разрабатываемым для iPhone.

**Важно!** Многие из этих рекомендаций и техник оптимизации имеют критическое значение при разработке приложений для iPhone.

## Аппаратное ускорение

Для повышения производительности графики в некоторых приложениях можно использовать аппаратное ускорение OpenGL ES 1.1. Аппаратное ускорение позволяет улучшить работу игр и других приложений с анимированными экранными объектами. Приложения, использующие аппаратное ускорение, могут передавать обработку некоторых графических процессов с центрального процессора на графический процессор iPhone, что позволяет значительно улучшить производительность.

При разработке программы, использующей графический процессор, необходимо следовать правилам, обеспечивающим эффективное аппаратное ускорение содержимого.

Для применения аппаратного ускорения задайте для визуализации значение «Графический процессор» (во Flash Professional CS5 откройте диалоговое окно «Настройки iPhone» и перейдите на вкладку «Общие»). Также можно задать свойству `renderMode` значение `gpu` в файле дескриптора приложения.

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

См. разделы «[Настройка свойств приложения iPhone во Flash Professional CS5](#)» на странице 16 и «[Настройка свойств приложения iPhone в файле дескриптора приложения](#)» на странице 18».

Существует четыре класса экранных объектов, которые быстро обрабатываются с помощью аппаратного ускорения, если их содержимое изменяется не слишком часто:

- растровые изображения;
- экранные 2D-объекты, у которых свойство `cacheAsBitmap` имеет значение `true` и задано необязательное свойство `cacheAsBitmapMatrix` (см. ниже);
- экранные 3D-объекты (т. е. объекты, у которых задано свойство `z`);
- экранные объекты, представляющие собой прямоугольник с одноцветной заливкой, края которого выровнены по пикселям экрана.

При анимации спрайта, лежащего за или перед векторным объектом, выполняется повторная визуализация этого векторного объекта. Таким образом, объекты, которые выполняют роль заднего или переднего плана анимации, также должны относиться к одной из этих категорий.

Если у экранного объекта свойство `cacheAsBitmap` имеет значение `true`, свойство `cacheAsBitmapMatrix` указывает графическому процессору, что следует использовать растровое изображение, полученное в результате преобразования матрицы. Растровое представление используется, даже если объект повернут, растянут или сжат. Графический процессор способен совместить и анимировать растровое изображение гораздо быстрее, чем ЦП перерисует векторный объект.

Если параметр `cacheAsBitmap` имеет значение `true`, экранный объект и все его дочерние объекты кэшируются. Экранный объект не перерисовывается при отображении новых областей или переводе комбинированной графики.

Когда задано свойство `cacheAsBitmapMatrix` экранного объекта, приложение может создать представление этого объекта, даже если он невидимый. Приложение создает кэшированное представление экранного объекта в начале следующего кадра. Когда экранный объект добавляется в рабочую область, приложение быстро выполняет визуализацию. Кроме того, анимация, вращение и масштабирование объекта также занимает гораздо меньше времени. Если объект вращается или масштабируется, вместе со свойством `cacheAsBitmap` обязательно задайте свойство `cacheAsBitmapMatrix`.

Программа может также быстро выполнять трансформации альфа-канала объекта, кэшированного как растровое изображение. Тем не менее, для трансформаций альфа-канала с аппаратным ускорением значение `alpha` должно быть от 0 до 1.0. Это соответствует значению параметра `colorTransform.alphaMultiplier` от 0 до 256.

Не задавайте свойству `cacheAsBitmap` значение `true` в часто обновляемых объектах, таких как текстовые поля.

Экранные объекты с часто изменяющимся графическим содержимым, как правило, не подходят для визуализации с использованием графического процессора. Прежде всего это касается старших устройств с менее мощными графическими процессорами. Если для загрузки графики в графический процессор требуется слишком много ресурсов, лучше выбрать визуализацию с использованием ЦП.

Пересмотрите структуру экранных объектов, содержащих дочерние объекты, которые двигаются относительно родительских. Замените их таким образом, чтобы дочерние объекты стали родственными. Тогда у каждого из них будет собственное растровое представление. Кроме того, каждый экранный объект будет двигаться относительно других без загрузки новой графики в графический процессор.

Задавайте свойству `cacheAsBitmap` значение `true` на самом высоком уровне списка отображения, ниже которого дочерние экранные объекты не анимируются. Другими словами, его необходимо задать для контейнеров экранных объектов, которые не содержат подвижных частей. На задавайте его для дочерних экранных объектов. *Не* задавайте его для спрайтов, которые содержат другие анимируемые экранные объекты.

Когда для экранного объекта задано свойство `z`, в приложении всегда используется кэшированное растровое представление. Кроме того, если для экранного объекта задано свойство `z`, даже при его вращении или масштабировании используется растровое представление. Свойство `cacheAsBitmapMatrix` не используется, если для экранного объекта не задано свойство `z`. Такие же правила действуют при задании любых свойств трехмерного экранного объекта, включая свойства `rotationX`, `rotationY`, `rotationZ` и `transform.matrix3D`.

Не задавайте свойство `scrollRect` или `mask` контейнера экранного объекта с содержимым, для которого необходимо использовать аппаратное ускорение. При задании этих свойств отключается аппаратное ускорение контейнера экранного объекта и его дочерних объектов. Вместо использования свойства `mask` можно наложить маскировочный экранный объект над маскируемым.

Размеры экранных объектов, для которых используется аппаратное ускорение, подвержены определенным ограничениям. На более старых устройствах это не более 1024 пикселей в ширину и в высоту. На новых устройствах это не более 2048 пикселей. Для тестирования производительности на устройстве можно применять инструмент диагностики визуализации с использованием графического процессора.

Графический процессор также использует оперативную память iPhone для хранения растровых изображений. Он использует по меньшей мере столько памяти, сколько необходимо для растровых изображений.

Графический процессор использует выделения памяти, равные степени числа 2 для каждого размера растрового изображения. Например, графический процессор может резервировать память размером 512 x 1024 или 8 x 32. Поэтому для изображения размером 9 x 15 пикселей используется такой же объем памяти, что и для изображения размером 16 x 16 пикселей. У кэшированных экранных объектов стороны могут быть не более чем в 2 раза длиннее. Например, использование экранного объекта размером 32 x 16 пикселей является более эффективным, чем использование экранного объекта 33 x 17 пикселей.

Не полагайтесь на измененную по размеру рабочую область для уменьшения масштаба активов, которые были изменены по размеру для других платформ (таких как рабочий стол). Вместо этого используйте свойство `cacheAsBitmapMatrix` или измените размер активов перед публикацией для iPhone. 3D-объекты пропускают свойство `cacheAsBitmapMatrix` при кэшировании изображения поверхности. По этой причине лучше изменять размер экранных объектов перед публикацией, если они будут визуализироваться на 3D-поверхности.

Между преимуществами аппаратного ускорения и использованием оперативной памяти существует компромисс. При заполнении памяти ОС iPhone уведомляет другие запущенные собственные приложения iPhone об освобождении памяти. По мере того, как приложения обрабатывают это уведомление и работают над освобождением памяти, они могут вести конкурентную борьбу с вашим приложением за использование циклов ЦП. Это может мгновенно привести к снижению производительности приложения. Обязательно протестируйте приложение в более старых устройствах, поскольку в них может быть предусмотрен гораздо меньший объем памяти, доступной для запущенных процессов.

При отладке программы на iPhone можно применять функцию диагностики визуализации с использованием графического процессора. Эта функция позволяет проверить, как программа выполняет визуализацию с использованием графического процессора. Дополнительные сведения см. в подразделе «Отладка с помощью функции диагностики визуализации с использованием графического процессора» раздела «[Отладка приложения iPhone](#)» на странице 26».

Дополнительные сведения о работе со свойством `cacheAsBitmapMatrix` см. в разделе «[DisplayObject.cacheAsBitmapMatrix](#)» статьи «[Специальные API-интерфейсы ActionScript для мобильных устройств](#)» на странице 32».

## Другие способы повышения производительности экранных объектов

Аппаратное ускорение может повысить производительность обработки графики для некоторых классов экранных объектов. Ниже приводятся несколько советов по увеличению производительности:

- Старайтесь ограничить число видимых объектов в рабочей области. На визуализацию каждого элемента требуется время, как и на его совмещение с окружающими объектами.

Когда отображать экранный объект больше не требуется, задайте его свойству `visible` значение `false` или уберите его из рабочей области (`removeChild()`). Для этого недостаточно просто задать свойству `alpha` значение 0.

- Старайтесь не использовать режимы наложения в общем и режим наложения слоев в частности. По возможности используйте обычный режим наложения.
- На фильтрацию экранных объектов требуется много вычислительных ресурсов. Не следует прибегать к ней слишком часто. Например, вполне допустимо использовать немного фильтров на начальном экране. Однако старайтесь не использовать их сразу для многих объектов, для анимированных объектов или при использовании высокой частоты кадров.
- Старайтесь не трансформировать фигуры.
- Старайтесь не использовать обтравку.
- По возможности, задайте параметру `repeat` значение `false` при вызове метода `Graphic.beginBitmapFill()`.
- Не применяйте перерисовку. Используйте для заднего плана цвет фона. Не накладывайте друг на друга большие фигуры. На отрисовку каждого пиксела требуются системные ресурсы. Это касается, прежде всего, экранных объектов без аппаратного ускорения.
- Старайтесь не использовать такие фигуры, как тонкие длинные зубцы, пересекающиеся сами с собой края или много мелких деталей по краям. На визуализацию таких фигур требуется больше времени, чем на визуализацию экранных объектов с ровными краями. Это касается, прежде всего, экранных объектов без аппаратного ускорения.
- Используйте растровые изображения, размеры которых приближены, но меньше  $2^n \times 2^m$  бит. Размеры не должны быть равны степени числа 2, но должны быть приближены к этому значению, не превышая его. Например, изображение с размером 31 x 15 пикселей визуализируется быстрее, чем с размером 33 x 17 пикселей. (31 и 15 чуть меньше степеней числа 2: 32 и 16.) Такие программы также более эффективно используют память.
- Высота и ширина экранных объектов должны быть не более 1024 x 1024 пикселей (2048 x 2048 на новых устройствах).

## Плотность информации

Физический размер экрана мобильных устройств меньше, чем у настольных, хотя и пиксельная плотность выше. Более четкий текст удобнее читать, но глифы должны иметь минимальный физический размер, иначе они будут нечитабельны.

Мобильные устройства часто используются на ходу и в условиях плохой освещенности. Учитывайте также то, какой объем информации реально можно отобразить на экране без ущерба удобочитаемости. Информации может поместиться меньше, чем на экране настольного компьютера с тем же разрешением.

Используйте типографическую иерархию для выделения важной информации. Используйте размер, толщину шрифта, размещение и интервалы для выражения относительной важности элементов пользовательского интерфейса. На каждом уровне иерархии можно использовать один или несколько способов выделения. Способы выделения должны быть согласованы в рамках всего приложения. Уровни иерархии можно выделять пространственно (отступ, интервал между строками, расположение) или графически (размер, стиль и цвет шрифта). Одновременное применение нескольких способов выделения позволяет эффективно отразить иерархию. Однако старайтесь не использовать более трех способов выделения для каждого уровня группировки.

Старайтесь упростить метки и пояснения, если они требуются. Например, добавьте образец ввода в текстовое поле, чтобы подсказать требуемое содержимое без добавления метки.

## Шрифты и ввод текста

Чтобы текст выглядел лучше, используйте шрифты устройства. Например, на iPhone доступны следующие шрифты устройства:

- Serif: Times New Roman, Georgia и \_serif
- Sans-serif: Helvetica, Arial, Verdana, Trebuchet, Tahoma и \_sans
- Моноширинные: Courier New, Courier и \_typewriter

Используйте шрифты размером 14 пикселей и больше.

Используйте шрифты устройства для редактируемых текстовых полей. Шрифты устройства в текстовых полях визуализируются быстрее, чем встроенные.

Не используйте текст с подчеркиванием для полей вводимого текста. Кроме того, не задавайте выравнивание текстового поля. В полях вводимого текста в устройстве iPhone поддерживается только выравнивание по левому краю (по умолчанию).

Если в среде Flash Professional CS5 для текстового поля используется параметр «Текст TLF», отключите общую библиотеку во время выполнения в связывании по умолчанию в окне «Параметры ActionScript 3.0». В противном случае приложение не будет работать на iPhone, так как оно будет пытаться использовать SWF-файл из общей библиотеки времени выполнения:

- 1 Выберите «Файл» > «Параметры публикации».
- 2 В диалоговом окне «Параметры публикации» перейдите на вкладку «Flash».
- 3 Нажмите кнопку «Сценарий» справа от раскрывающегося списка «Сценарий (ActionScript 3.0)».
- 4 Перейдите на вкладку «Путь к библиотеке».
- 5 В списке «Связывание по умолчанию» выберите «Объединено с кодом».

Попробуйте найти альтернативы использованию текстовых полей для ввода. Например, чтобы пользователь мог ввести число, не требуется текстовое поле. Можно добавить две кнопки для увеличения или уменьшения значения.

Учитывайте место, которое занимает виртуальная клавиатура. При активации виртуальной клавиатуры (например, когда пользователь прикасается к текстовому полю), приложение корректирует свое местоположение на рабочей области. Положение автоматически изменяется так, чтобы выбранное текстовое поле оставалось видимым:

- Текстовое поле вверху рабочей области перемещается в верхнюю часть видимой области. (Видимая область меньше, так как на экране отображается виртуальная клавиатура.)
- Текстовое поле внизу рабочей области остается в нижней части видимой области.
- Текстовое поле в другой части рабочей области перемещается в центр видимой области.

Когда пользователь щелкает текстовое поле, чтобы отредактировать его (и отображается виртуальная клавиатура), объект TextField отправляет событие `focusIn`. Можно добавить прослушиватель этого события для изменения положения текстового поля.

Однострочное текстовое поле включает кнопку очистки (справа от текста), когда пользователь редактирует текст. Однако эта кнопка очистки не отображается, если текстовое поле является слишком узким.

После редактирования текста в однострочном текстовом поле пользователь закрывает виртуальную клавиатуру путем нажатия клавиши «Готово» на клавиатуре.

После редактирования текста в многострочном текстовом поле пользователь закрывает виртуальную клавиатуру прикосновением за пределами текстового поля. Это позволяет удалить фокус из текстового поля. Убедитесь, что при отображении виртуальной клавиатуры за пределами текстового поля предусмотрена свободная область. Если текстовое поле является слишком большим, остальная область может быть скрыта.

Использование некоторых компонентов Flash Professional CS5 позволяет предотвратить удаление фокуса из текстового поля. Эти компоненты предназначены для использования на настольных компьютерах, в которых такое поведение фокуса является предпочтительным. Одним из таких компонентов является TextArea. Когда он находится в фокусе (и редактируется), удаление фокуса путем выбора другого экранного объекта невозможно. Добавление некоторых других компонентов Flash Professional CS5 в рабочую область также позволяет предотвратить переключение фокуса редактируемого текстового поля.

Не полагайтесь на события клавиатуры. Например, SWF-содержимое, разработанное для Интернета, предоставляет пользователю возможность управления с помощью клавиатуры. Однако на устройстве iPhone виртуальная клавиатура активна, только когда пользователь редактирует текстовое поле. Приложение iPhone отправляет события клавиатуры, только когда она активна.

## Сохранение состояния приложения

Приложение может завершить работу в любой момент (например, при получении вызова). Следует предусмотреть сохранение состояния приложения после каждого изменения. Например, можно сохранить настройки в файле или в базе данных в каталоге хранения приложения. Также можно сохранить данные в локальном общем объекте. При следующем запуске приложения можно будет восстановить сохраненное состояние. Если при телефонном вызове работа приложения прерывается, после окончания разговора она возобновится.

Не полагайтесь на событие `exitting` объекта `NativeApplication` при закрытии приложения, так как он может не отправить его.

## Изменение ориентации экрана

Содержимое iPhone можно просматривать с использованием книжной или альбомной ориентации. Предусмотрите поведение приложения при изменении ориентации экрана. Дополнительные сведения см. в разделе [Обнаружение ориентации устройства](#).

## Мишени

Учитывайте размер мишени при разработке кнопок и других элементов интерфейса, которые должны нажимать пользователи. Эти элементы должны быть достаточно большими, чтобы их можно было удобно активировать с помощью пальца на сенсорном экране. Кроме того, между мишенями должно быть достаточно места. Приблизительный размер мишени составляет 44 – 57 пикселей.

## Выделение памяти

Выделение новых блоков памяти требует больших ресурсов. Это может снизить быстродействие приложения или вызвать задержки при анимации или взаимодействии, когда активируется процесс сбора мусора.

По возможности старайтесь повторно использовать объекты, вместо того чтобы удалять их и потом создавать новые.

Помните о том, что векторы потребляют меньше памяти, чем массивы. См. раздел [Сравнение классов Vector и Array](#).

Дополнительные сведения об использовании памяти см. в разделе [Экономия памяти](#).

## API-интерфейс рисования

Старайтесь не использовать API-интерфейс рисования `ActionScript` (класс `Graphics`) для создания графики. При использовании API-интерфейса рисования объекты создаются динамически в рабочей области, а затем визуализируются для растривания. По возможности создавайте эти объекты статически во время разработки.

При повторных вызовах объекты, созданные с помощью API-интерфейса рисования, удаляются и повторно создаются при каждом выполнении кода `ActionScript`. А статические объекты остаются в памяти в течение всего времени.

## Восхождение событий по цепочке

Для контейнера экранных объектов с множеством вложенных объектов восхождение событий по цепочке может быть ресурсоемким. Во избежание этого событие следует полностью обрабатывать в целевом объекте, а затем вызывать метод `stopPropagation()` для объекта события. Вызов этого метода предотвращает дальнейшее восхождение события. Кроме того, в результате этого родительские объекты не получают события.

В качестве альтернативного решения можно сократить число дочерних объектов, чтобы избежать создания длинных цепочек событий.

По возможности регистрируйте события `MouseEvent` вместо событий `TouchEvent`. Для событий `MouseEvent` используется меньше ресурсов процессора, чем для событий `TouchEvent`.

По возможности задавайте для свойств `mouseEnabled` и `mouseChildren` значение `false`.

## Оптимизация производительности видео

Для оптимизации воспроизведения видео на мобильных устройствах избегайте в приложении ресурсоемких процессов, выполняемых во время воспроизведения. Тогда для декодирования и визуализации видео ЦП будет использоваться как можно интенсивнее.

Во время воспроизведения видео не должен выполняться код `ActionScript`. Старайтесь не использовать код, выполняемый автоматически через короткие интервалы или по расписанию.

Сведите к минимуму перерисовку экранных объектов, не связанных с видео. Прежде всего это касается экранных объектов, пересекающихся с областью видео, даже если они скрыты за видео. Эти объекты все равно будут перерисовываться и занимать ресурсы процессора. Например, для индикатора положения следует выбрать простую фигуру и обновлять его не в каждом кадре, а всего пару раз в секунду. Не накладывайте элементы управления видео на область воспроизведения, лучше разместите их под ней. Если используется буферизированная анимация, не помещайте ее за видео, когда она неактивна; лучше сделать ее невидимой.

## Компоненты Flex и Flash

Многие компоненты Flex и Flash были разработаны для настольных программ. Эти компоненты, особенно экранные, могут медленно работать на мобильных устройствах. Кроме того, модели взаимодействия компонентов, предназначенных для настольных платформ, могут не подходить для мобильных устройств.

Компания Adobe в настоящее время работает над оптимизацией инфраструктуры Flex для мобильных платформ. Дополнительные сведения см. по адресу <http://labs.adobe.com/technologies/flex/mobile/>.

## Сокращение размера файла приложения

Предлагаем несколько советов по уменьшению размера IPA-файла:

- Проверьте размер растровых изображений на заднем плане (он не должен быть больше, чем необходимо).
- Удалите ненужные встроенные шрифты, если имеются.
- Проверьте PNG-активы на предмет альфа-каналов и удалите ненужные. Для сокращения размера PNG-активов можно воспользоваться, например, служебной программой PNG Crunch.
- Преобразуйте PNG-активы в JPG-активы, если это возможно.
- Уменьшите размер звуковых файлов (за счет более низкой скорости потока).
- Удалите неиспользуемые активы.