



Самоучитель

Ларри Уллман

Adobe AIR

Разработка приложений с помощью Ajax

VISUAL QUICKPRO GUIDE

ADOBE
WITH AJAX



Простота и наглядность изложения материала

Четкость и лаконичность
при рассмотрении этапов работы

Автоматическая генерация
кросс-платформенных приложений

Создание приложений
с графическим интерфейсом

VISUAL QUICKPRO GUIDE

ADOBE AIR

(ADOBE INTEGRATED RUNTIME)

WITH AJAX

Larry Ullman

 **Peachpit Press**

Ларри Уллман

Самоучитель

Adobe AIR

**Разработка приложений
с помощью Ajax**

Санкт-Петербург

«БХВ-Петербург»

2009

УДК 681.3.06
ББК 32.973.26-018.2
У49

Уллман Л.

У49 Самоучитель Adobe AIR. Разработка приложений с помощью Ajax:
Пер. с англ. — СПб.: БХВ-Петербург, 2009. — 560 с.: ил.

ISBN 978-5-9775-0329-7

Книга посвящена разработке Web-приложений в кросс-платформенной рабочей среде Adobe Integrated Runtime (AIR) для Microsoft Windows XP (SP2)/Windows Vista и Mac OS X 10.4/10.5 с использованием технологии Ajax (HTML и JavaScript). Рассматривается создание, запуск, отладка и развертывание AIR-приложений, взаимодействие приложений с сетевыми ресурсами, применение средств разработки AIR, создание окон и меню, импорт и экспорт данных, работа с файлами, папками и базами данных, использование медиакomпонентов. Сопровождающий книгу Web-сайт www.DMCInsights.com/air содержит примеры скриптов, весь программный код, приведенный в книге, новые версии программного обеспечения и множество другой полезной информации.

Для разработчиков Web-приложений

УДК 681.3.06
ББК 32.973.26-018.2

Группа подготовки издания:

Главный редактор	<i>Екатерина Кондукова</i>
Зам. главного редактора	<i>Наталья Таркова</i>
Зав. редакцией	<i>Григорий Добин</i>
Редактор	<i>Анна Кузьмина</i>
Перевод с англ.	<i>Елены Михновой</i>
Компьютерная верстка	<i>Натальи Смирновой</i>
Корректор	<i>Наталья Перишакова</i>
Дизайн серии	<i>Инны Тачиной</i>
Оформление обложки	<i>Елены Беляевой</i>
Зав. производством	<i>Николай Тверских</i>

Authorized translation from the English language edition, entitled ADOBE AIR (ADOBE INEGRATED RUNTIME) WITH AJAX: VISUAL QUICKPRO GUIDE, 1st Edition, ISBN 0321524616, by ULLMAN, LARRY, published by Pearson Education, Inc, publishing as Peachpit Press, Copyright © 2008. All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without the prior permission from Pearson Education, Inc. RUSSIAN language edition published by BHV St. Petersburg, Copyright © 2009.

Авторизованный перевод английской редакции, выпущенной Pearson Education, Inc Peachpit Press, импринт Peachpit Press Copyright © 2008. Все права защищены. Никакая часть настоящей книги не может быть воспроизведена или передана в какой бы то ни было форме и какими бы то ни было средствами, будь то электронные или механические, включая фотокопирование, записана на магнитный носитель, размещена в любых базах данных и информационно-поисковых системах без предварительного письменного разрешения Pearson Education, Inc. Перевод на русский язык "БХВ-Петербург" © 2009.

Лицензия ИД № 02429 от 24.07.00. Подписано в печать 30.12.08.

Формат 70×100^{1/16}. Печать офсетная. Усл. печ. л. 45,15.

Тираж 1500 экз. Заказ №

"БХВ-Петербург", 190005, Санкт-Петербург, Измайловский пр., 29.

Отпечатано с готовых диапозитивов

в ГУП "Типография "Наука"

199034, Санкт-Петербург, 9 линия, 12

ISBN 978-0-321-52461-4 (англ.)

ISBN 978-5-9775-0329-7 (рус.)

© 2008 by Larry Ullman

© Оформление, издательство "БХВ-Петербург", 2009

Оглавление

БЛАГОДАРНОСТИ.....	3
ВВЕДЕНИЕ.....	5
Об этой книге	7
Что вам понадобится	8
Получение помощи.....	8
ГЛАВА 1. ЗАПУСК AIR-ПРИЛОЖЕНИЙ.....	11
Установка среды исполнения	11
Установка приложения.....	15
Запуск приложения.....	20
ГЛАВА 2. СОЗДАНИЕ ПРИЛОЖЕНИЯ.....	23
Установка SDK.....	23
Обновление пути в Windows	25
Обновление пути в Mac OS X.....	29
Создание структуры проекта	33
Создание HTML-файла	35
Создание XML-файла.....	37
Тестирование приложения.....	42
Создание сертификата.....	46
Компоновка приложения	48
ГЛАВА 3. СРЕДСТВА РАЗРАБОТКИ AIR.....	51
Использование Aptana Studio	51
Использование Dreamweaver	64
Создание цифровых подписей.....	70

ГЛАВА 4. ОСНОВНЫЕ КОНЦЕПЦИИ И КОД.....	79
Технологическая база.....	79
Использование AIRAliases.js	85
JavaScript Frameworks	88
Библиотеки ActionScript.....	92
Обработка событий.....	94
Объект <i>XMLHttpRequest</i>	99
ГЛАВА 5. ОТЛАДКА	107
Диалоговые окна JavaScript	107
Метод <i>trace()</i>	110
AIR Introspector	113
Другие методы отладки.....	116
ГЛАВА 6. СОЗДАНИЕ ОКОН.....	121
Создание нового окна.....	121
Создание нового "родного" окна.....	126
Настройка окон	130
Получение доступа к новому "родному" окну	135
Создание полноэкранных окон	140
Обработка событий окна.....	144
Создание уникального внешнего вида	148
Перемещение и изменение размеров окна	153
ГЛАВА 7. СОЗДАНИЕ МЕНЮ	159
Терминология меню	159
Создание меню.....	162
Обработка событий меню	168
Меню, ориентированные на операционную систему пользователя	173
Добавление "горячих" клавиш к элементам меню	177
Изменение состояния элемента меню	184
ГЛАВА 8. ИМПОРТ И ЭКСПОРТ ДАННЫХ.....	191
Копирование.....	191
Программная реализация копирования данных в буфер обмена.....	194
Вырезание.....	198

Вставка.....	203
Работа с разными форматами	208
Операции drag and drop, сопровождающиеся переносом данных в приложение.....	214
Операции drag and drop, сопровождающиеся переносом данных во внешние приложения.....	221
ГЛАВА 9. ФАЙЛЫ И ПАПКИ.....	227
Основные понятия	227
Обзор файлов и папок	232
Доступ к информации о файле	238
Считывание содержимого папок.....	243
Удаление файлов и папок	249
Копирование и перемещение.....	255
ГЛАВА 10. РАБОТА С СОДЕРЖИМЫМ ФАЙЛА.....	265
Считывание данных из файлов	266
Запись в файлы.....	273
Пример асинхронного взаимодействия	282
Работа с бинарными данными.....	291
ГЛАВА 11. РАБОТА С БАЗАМИ ДАННЫХ	299
Соединение с базой данных.....	300
Создание базы данных	304
Добавление записей в таблицу	309
Обработка ошибок.....	317
Извлечение записей с помощью запроса <i>SELECT</i>	322
Обновление и удаление записей.....	330
ГЛАВА 12. ТЕХНОЛОГИИ БАЗ ДАННЫХ.....	343
Распространение баз данных	343
Использование подготовленных выражений	349
Извлечение записей группами.....	357
Выполнение транзакций.....	364
Увеличение производительности.....	379
Способы отладки	381

ГЛАВА 13. ВЗАИМОДЕЙСТВИЕ ПРИЛОЖЕНИЙ С СЕТЕВЫМИ РЕСУРСАМИ	385
Класс <i>URLRequest</i>	385
Получение данных	390
Синтаксический разбор данных	394
Передача данных	400
Загрузка файлов с сервера	407
Загрузка больших файлов с сервера	411
Загрузка файлов на сервер	416
ГЛАВА 14. ИСПОЛЬЗОВАНИЕ МЕДИАКОМПОНЕНТОВ	423
Воспроизведение звуковых файлов	423
Воспроизведение длинных песен	427
Потоковое воспроизведение звука	432
Управление воспроизведением звука	439
Отображение PDF-документов	447
Обработка XML-данных	452
ГЛАВА 15. ТЕХНОЛОГИИ БЕЗОПАСНОСТИ	463
Модель безопасности в AIR	464
Работа с содержимым, расположенным вне песочницы приложения	471
Использование моста песочницы безопасности	476
Хранение кодированных данных	496
Проверка данных	503
Лучшие меры безопасности	505
ГЛАВА 16. РАЗВЕРТЫВАНИЕ ПРИЛОЖЕНИЙ	507
Дополнительные опции файла дескриптора приложения	507
Добавление пользовательских пиктограмм для приложения	512
"Бесшовная" установка	516
Еще несколько полезных идей по разворачиванию приложения	520
Обновление приложения	526
ПРЕДМЕТНЫЙ УКАЗАТЕЛЬ	539

Семье Кавенис (Kavounis)

Благодарности

Хотелось бы выразить специальные благодарности первоклассному редактору¹ и чемпиону среди авторов книг, Ребекке Галик (Rebecca Gulick).

Анне Марии Уолкер (Anne Marie Walker) — за ее непревзойденную внимательность к деталям и правку текста, который порой оказывался слишком невнятным.

Дебби Роберти (Debbie Roberti) и Мирне Владик (Myrna Vladic) — за сборку воедино отдельных частей моей книги.

Ребекке Планкет (Rebecca Plunkett) — за компоновку индекса.

Всем сотрудникам Peachpit Press — за огромную работу, которую они выполняют!

Прейенк Суарупу (Prayank Swaroop) — за точные и очень полезные технические обзоры.

Николь (Nicole) — за то, что она присматривала за детьми, а я в свою очередь мог больше времени уделять этой книге.

Джессике (Jessica), Зоу (Zoe) и Сэму (Sam) — за то, что они не занимали мой компьютер и давали возможность поработать.

¹ В оригинальном издании. — *Прим. ред.*

Введение

Adobe AIR (Adobe Integrated Runtime, интегрированная среда исполнения Adobe) — это потрясающая новая технология, которая расширяет возможности разработчиков по созданию настольных приложений. Всем известные программы Microsoft Word, Web-браузер Firefox и Adobe Reader написаны на таких языках программирования, как C или C++. Изучение этих технологий программирования (хотя и не слишком сложных) требует определенных усилий, а создание графических интерфейсов и кроссплатформенных приложений — еще большая проблема. Теперь, с появлением Adobe AIR разработчик вправе использовать любую известную ему Web-технологю — будь то Adobe Flash, Adobe Flex или стандартный Ajax (HTML и JavaScript) — для создания полнофункциональных настольных приложений, которые успешно функционируют и под Windows, и под Mac OS X, и под Linux.

Данная книга специализируется исключительно на подходе к разработке с использованием Ajax (HTML и JavaScript). С ее помощью читатель непременно научится создавать полезные Adobe AIR-приложения. С минимальным количеством технического жаргона и множеством практических примеров, этот четко структурированный текст является отличным путеводителем для тех, кто хочет применить свои знания по Web-разработке в новом качестве. Adobe AIR — превосходное инструментальное средство разработки программного обеспечения как для собственного, так и для корпоративного пользования. Adobe AIR можно использовать для многих целей, в том числе и для реорганизации уже готового Web-сайта, что позволит расширить целевую аудиторию пользователей.

На сегодняшний день существует огромное число средств разработки. В связи с этим возникает вполне резонный вопрос: "Почему я должен использовать именно Adobe AIR?" Одним из наиболее явных преимуществ Adobe AIR является *простота в использовании*. К тому же, разработчик применяет уже имеющиеся у него навыки программирования и ему не нужно переучиваться. В худшем случае, при отсутствии базовых знаний HTML и JavaScript, Adobe AIR по праву считается наиболее легкой в освоении технологией. Поэтому достаточно быстрое обучение сразу даст положительные результаты.

Вторым преимуществом Adobe AIR является автоматическая генерация *кроссплатформенных приложений*. Написанные программы будут одинаково хорошо работать во всех операционных системах: Windows, Mac OS X и Linux (независимо от того, на какой платформе разрабатывалось приложение). Концепция кроссплатформенных приложений была ключевой при создании Adobe AIR. Поэтому специфические действия по проектированию под разные платформы в Adobe AIR нужны крайне редко.

Третье достоинство Adobe AIR заключается в создании *приложений с графическим интерфейсом*. В результате получаются наглядные программы, которые выполняются не в командной строке или терминальном приложении, а используют все преимущества управления интерфейсом с помощью мыши и клавиатуры. Для вас это может оказаться открытием, но на самом деле при работе с другими технологиями программирования (например, C или C++) порой очень сложно создать графический интерфейс, в отличие от утилиты командной строки.

Adobe AIR-приложение может выполнять следующие функции:

- получать доступ к файлам и папкам на компьютере пользователя;
- интегрировать базу данных, расположенную на стороне клиента;
- обеспечивать безопасность данных путем их хранения в зашифрованном формате;
- содержать пользовательские окна и меню;
- взаимодействовать с сетевыми ресурсами (Web-сайтами, серверами и т. д.);
- получать доступ к буферу обмена и поддерживать операции копирования, вставки, вырезания, перетаскивания и растягивания;
- воспроизводить аудио- и видеофайлы;
- отображать PDF-файлы.

Короче говоря, AIR-приложение может обладать практически безграничной функциональностью, которую только можно себе представить!

Если вам все еще нужны доказательства уникальности этого продукта, то в его пользу также говорит свободно распространяемое программное обеспечение, требуемое для разработки и выполнения AIR-приложений. Причем поддержка этих программ осуществляется самой лучшей компанией в мире — Adobe. Если посмотреть на то, как Adobe создавала, продвигала и распространяла PDF (Portable Document Format, портативный формат документов), можно не сомневаться в светлом будущем Adobe AIR.

Об этой книге

В этой книге раскрыты все тонкости, которые должен знать разработчик для создания настольных приложений в Adobe AIR. Несмотря на то, что существуют три базовые технологии, которые могут использоваться в AIR (Ajax, Flash и Flex), в этой книге рассказывается только об Ajax (по сути, это HTML и JavaScript). Технология Ajax была выбрана мной исходя из нескольких соображений.

- по-моему, HTML и JavaScript проще в освоении, чем Flash и Flex;
- больше людей владеют HTML и JavaScript;
- существует гораздо больше программ автоматической генерации кода на HTML и JavaScript.

Книга представлена шестнадцатью главами, которые можно читать в произвольной последовательности. Однако для начала я настоятельно рекомендую ознакомиться с первыми четырьмя главами, т. к. они содержат основную информацию, которая необходима для понимания всего последующего материала.

Судя по моему подходу к любому делу, очень надеюсь, что вы найдете содержимое этой книги корректным и соответствующим тематике, она не покажется вам слишком сложной и перегруженной техническим жаргоном. Также я попытался сделать акцент на использовании практических примеров из реальной жизни. За некоторым исключением все рисунки книги демонстрируют последовательность действий, выполняемых настольными приложениями.

Из-за ограниченного объема книги здесь раскрыты далеко не все возможности AIR. Как уже было сказано ранее, в книге не содержится информация по написанию AIR-приложений во Flash или Flex (кстати, не думаю, что хорошая книга может покрыть вопросы разработки с использованием всех существующих технологий). Кроме этого было опущено еще несколько тем, например, использование аргументов командной строки или системы DRM (digital rights management, управление цифровыми правами) для медиафайлов. Смеею вас заверить, что я опустил лишь те темы, которые не нужны большинству читателей. Со всеми этими темами, при желании, можно ознакомиться в online-документации (на самом деле это можно сказать о любой теме).

Что вам понадобится

К счастью, для успешной работы по созданию Adobe AIR-приложений с помощью Ajax требования достаточно просты. Фактически вам даже не придется платить никаких денег! Для запуска и выполнения AIR-приложений вам понадобится скачать и установить среду исполнения Adobe AIR (все подробности раскрыты в *главе 1*). Среда исполнения работает под операционными системами Microsoft Windows 2000, XP или Vista, а также под Mac OS X 10.4.9 или более поздними версиями. Во время написания этой книги (апрель 2008 г.) была только-только выпущена альфа-версия среды исполнения Adobe AIR под Linux.

Для создания AIR-приложения необходимо скачать и установить SDK (Software Development Kit, набор инструментальных средств разработки). Последовательность установки подробно описана в *главе 2*. SDK предполагает использование либо среды исполнения Java (JRE, Java Runtime Environment), либо набора средств разработки Java (JDK, Java Development Kit). Поэтому одно из двух должно быть установлено на вашем компьютере. И JDK, и JRE свободно распространяются на сайте <http://java.sun.com> и являются составной частью так называемой стандартной версии Java (Java SE, Java Standard Edition).

Для разработки Adobe AIR-приложения, т. е. для написания кода на HTML и JavaScript, вам понадобится текстовый редактор или IDE (Integrated Development Environment, интегрированная среда разработки). Если у вас уже есть излюбленный текстовый редактор, это просто замечательно. В *главе 2* описана последовательность действий, которую необходимо предпринять для создания AIR-приложения с помощью текстового редактора. *Глава 3* предназначена для тех читателей, которые предпочитают IDE для разработки AIR-приложений (например, Aptana Studio или Adobe Dreamweaver).

Кроме изложенных выше требований по программному обеспечению, предполагается, что читатель обладает базовыми знаниями HTML. В противном случае, советую прочитать несколько книг на эту тему. Наличие некоторого представления о JavaScript (или знание какого-либо другого языка программирования) позволит быстрее воспринимать приведенные в книге листинги.

Получение помощи

Если вдруг у вас возникнут какие-либо проблемы с Adobe AIR или вопросы по содержанию этой книги, в Интернете представлено множество ресурсов

по оказанию посильной помощи. Первый из них — это собственно Web-сайт с поддержкой по Adobe AIR — www.adobe.com/go/air/. На этом сайте вы сможете скачать среду исполнения AIR и SDK, а также множество примеров приложений с открытым программным кодом. Здесь вы найдете статьи и руководства для разработчиков AIR-приложений на Ajax (Adobe Developer Center — www.adobe.com/devnet/air/ajax/).

Еще я рекомендую сделать закладки в двух областях сайта Adobe. Первая — www.adobe.com/go/learn_air_html — это online-документация компании Adobe для разработки AIR-приложений с помощью Ajax. На этих Web-страницах повествуется о том, как выполнять различные задачи.

Вторая ссылка довольно часто используется — это JavaScript Language Reference (справочная информация по JavaScript), www.adobe.com/go/learn_air_html_jslr. Здесь представлена более серьезная техническая поддержка, в наличии подробное описание всех функций JavaScript, которые нужны для разработки AIR-приложений. Короче говоря, если вы забудете назначение какого-либо widget-элемента (виджета), эти страницы оперативно предоставят необходимую информацию.

Простая поисковая функция позволяет найти множество менее официальных ресурсов по Adobe AIR. При использовании подобного рода сайтов обращайтесь внимание, действительно ли на сайте речь идет именно об AIR-приложениях, написанных на Ajax, Flash или Flex. Большинство сайтов оснащено форумами, которые могут оказаться очень полезными для получения быстрых ответов на интересующие вопросы (кстати, на сайте Adobe тоже есть форум).

Безусловно, вы должны (даже обязаны) пользоваться сайтом, который я создал специально для читателей этой книги. Сайт находится по адресу www.DMCInsights.com/air/. На этом сайте посетители могут скачать весь программный код, приведенный в книге, получить доступ к форуму, задать свой вопрос, найти исправленные ошибки, если таковые закрались в книгу, а также связаться с искренне вашим автором, т. е. со мной.

ГЛАВА 1



Запуск AIR-приложений

Вместо того чтобы начинать с нуля разработку своего собственного Adobe AIR-приложения (Adobe Integrated Runtime, интегрированная среда исполнения Adobe), следует прочитать данную главу, в которой рассказано о том, как запустить и выполнить уже готовое AIR-приложение. Эти знания, вместе с некоторыми исходными настройками, пригодятся при тестировании собственных приложений, когда вы их создадите. Еще один важный момент — приведенная последовательность действий понадобится каждому потенциальному пользователю, чтобы увидеть свою великолепную программу.

Сначала описан порядок установки интегрированной среды разработки Adobe под Windows и под Mac OS X (техническую документацию по установке AIR под Linux см. на сайте Adobe). После того как среда исполнения будет успешно инсталлирована, вы убедитесь в простоте установки любого AIR-приложения. И наконец, вы сможете выполнить приложение на своем компьютере. Некоторые настройки, сделанные в процессе разработки вашего приложения, повлияют на его установку и выполнение, поэтому обратите внимание на детали, которые также обсуждаются в этой главе.

Установка среды исполнения

Установку AIR можно выполнить на перечисленные здесь платформы:

- Windows XP с Service Pack 2;
- Windows Vista Ultimate;
- Mac OS X 10.4.8 или более поздней версии (PowerPC);
- Mac OS X 10.4.8 или более поздней версии (Intel).

Все это касается бета-версии AIR, которая даже может быть установлена под Linux, если дата ее выпуска более поздняя, чем выпуск первой официальной версии.

Далее описан процесс установки AIR для операционной системы Windows XP Professional и Mac OS X 10.4.9 (Intel) с последними версиями патчей и обновлений. Рисунки этого раздела являются комбинацией копий экранов, выполненных в двух разных операционных системах, однако, невзирая на это, последовательность этапов сохраняется.

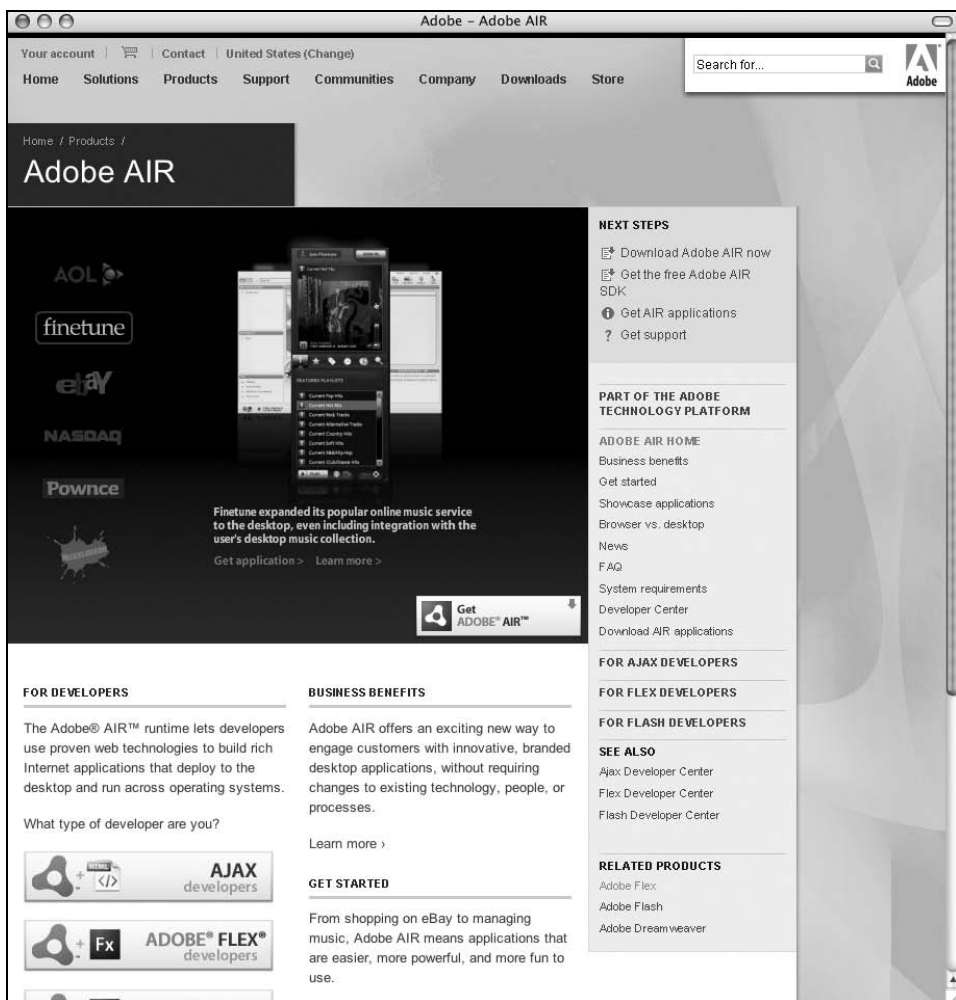


Рис. 1.1. Главная страница Adobe AIR

Этапы установки среды исполнения:

1. Скачайте установочную программу Adobe AIR. Для этого перейдите на Web-страницу www.adobe.com/go/air/ (рис. 1.1). Щелкните на ссылке **Download Adobe AIR now** для перехода на страницу загрузки (рис. 1.2).
2. Запустите скачанный установочный файл (рис. 1.3). В Windows имя файла будет AdobeAIRInstaller.exe. Для запуска установки сделайте двойной щелчок мышью на пиктограмме этого файла.

В Mac OS X имя файла — AdobeAIR.dmg. Дважды щелкните на DMG-файле, чтобы смонтировать образ диска (если он не смонтировался автоматически). Затем выполните двойной щелчок мышью на пиктограмме установочного файла (рис. 1.3), которая находится в образе диска.



Рис. 1.2. Страница загрузки Adobe AIR.
На ней также находятся ссылки на примеры AIR-приложений



Рис.1.3. Пиктограмма установочного файла среды исполнения Adobe

3. Прочитайте и примите лицензионное соглашение (рис. 1.4). Лицензия — это правовой документ, который следует прочитать глазами юриста и подумать: не противоречит ли эта лицензия каким-либо принятым вами соглашениям и т. п. Можете просто щелкнуть на кнопке **I Agree** (Я принимаю), не особенно вдаваясь в подробности — это уже на ваше усмотрение.

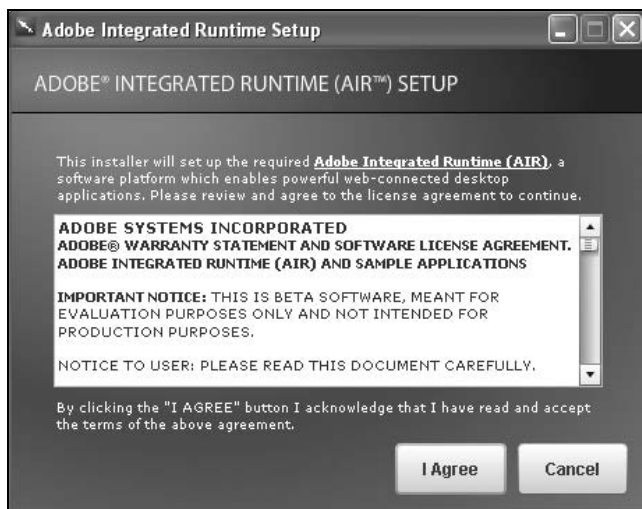


Рис. 1.4. Для установки среды исполнения Adobe нужно принять лицензионное соглашение

4. Когда установка будет завершена (рис. 1.5), щелкните на кнопке **Finish** (Готово).

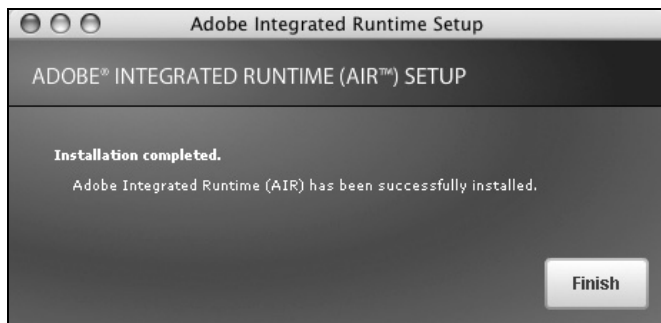


Рис. 1.5. Ура! Теперь смело можете устанавливать AIR-приложения

СОВЕТ

- Компания Adobe постоянно выпускает обновленные версии AIR. Вы можете скачать их на официальном сайте Adobe и обновить старую версию AIR, установленную ранее на вашем компьютере. Установка новых версий выполняется аналогично.
- Одним из явных преимуществ AIR-приложений является возможность их выполнения на разных платформах. При этом в самом приложении не нужно делать никаких изменений. Поэтому техническая поддержка Adobe для разных операционных систем постоянно расширяется и совершенствуется.
- После успешной установки среды исполнения AIR она функционирует как бы за кулисами. AIR не является программой, в том смысле, что она не запускается отдельным приложением, и пользователь не увидит ее рабочее окно. Для подтверждения успешной установки среды исполнения необходимо установить AIR-приложение.

УДАЛЕНИЕ СРЕДЫ ИСПОЛНЕНИЯ

Многим не понадобится удалять среду исполнения, но на самом деле это делается очень просто. Для удаления среды исполнения в Windows откройте Control panel (Панель управления) и выберите в ней опцию **Add or Remove Programs** (Установка и удаление программ). Для удаления среды исполнения в Mac OS X дважды щелкните мышью на значке программы деинсталляции Adobe AIR, который находится в папке установок Applications.

Установка приложения

Среда исполнения AIR устанавливается на компьютер единожды и позволяет в дальнейшем устанавливать любое количество AIR-приложений. Прочтение данной книги предполагает создание нескольких AIR-приложений (как-никак, это и есть цель книги), однако в Интернете выложено достаточно много уже готовых приложений. Кроме различных менее официальных Web-ресурсов, на сайте Adobe можно скачать примеры AIR-приложений. Один из них будет использован далее для описания этапов установки.

Этапы установки AIR-приложения:

1. Скачайте приложение из Интернета. Немного приложений есть на официальном сайте компании Adobe (www.adobe.com/devnet/air/flex/samples.html), а вообще во всей Глобальной сети их огромное множество.

Для данного примера я выбрал веселенькое приложение от Adobe, которое называется ScreenBoard. Помните, что загружаемый файл приложения зачастую не чувствителен к операционной системе, где он впоследствии будет выполняться.

2. Для того чтобы начать процесс установки, дважды щелкните на загруженном файле (рис. 1.6). AIR-приложения используют расширение `air`, а значок файла напоминает картонную коробку.



Рис. 1.6. Приложение ScreenBoard с расширением `air` использует интегрированную среду исполнения Adobe

При установке первого AIR-приложения на экране появится окно с предупреждением о первом открытии установщика Adobe AIR-приложений (рис. 1.7). Щелкните на кнопке **Open** (Открыть) для продолжения процесса установки. При последующих установках AIR-приложений это окно появляться не будет.



Рис. 1.7. Инсталляция AIR-приложений осуществляется с помощью установщика Adobe AIR-приложений. В большинстве систем при установке первого AIR-приложения на экране появится это окно

3. Ознакомьтесь с информацией, представленной в исходном окне установки (рис. 1.8). Для установки приложения щелкните на кнопке **Install** (Установить).

В исходном окне установки содержится очень важная информация, например, название приложения и его издатель (юридическое или физическое лицо, создавшее приложение). В последующих главах вы узнаете о том, как присваивать эти значения своим приложениям.

Также в исходном окне установки отображается информация о доступе устанавливаемого приложения к системе. Ведь очень важно, чтобы разработчики и пользователи AIR-приложений сознательно относились к вопросам безопасности. Adobe AIR позволяет Web-разработчикам создавать настольные приложения. Это означает, что AIR-приложение может нанести ущерб компьютеру пользователя. Обычный Web-сайт, например, никогда не причинит вреда посетителю (за некоторым исключением). В данной книге уделяется достаточно большое внимание проблемам безопасности. Непосредственно этой теме посвящена *глава 15*.



Рис. 1.8. Процесс установки начинается с детальной информации о приложении: названия, автора и параметров безопасности

4. На следующем шаге (рис. 1.9) надо указать местоположение установки и необходимость автоматического запуска приложения сразу после установки. Щелкните на кнопке **Continue** (Продолжить).

В операционной системе Windows AIR-приложение устанавливается по умолчанию в папку Program Files, как и любая другая программа. пользо-

вателям Windows также предоставляется возможность добавить ярлык приложения на рабочий стол (рис. 1.10).

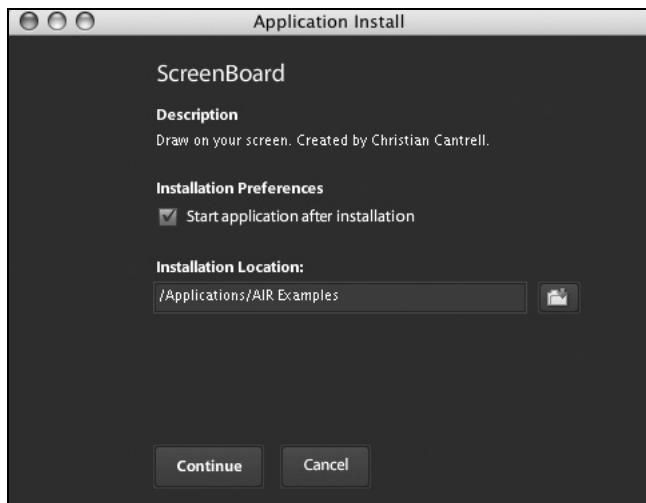


Рис. 1.9. Установщик позволяет определить месторасположение программы

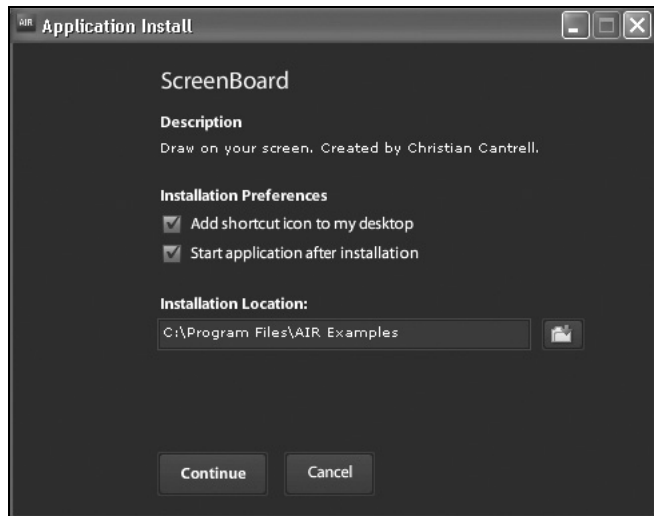


Рис. 1.10. Установщик Windows предоставляет возможность добавить ярлык для нового приложения на рабочий стол

В операционной системе Mac OS X программы AIR устанавливаются по умолчанию в папку Applications, как и все остальные приложения.

Некоторые приложения отображают путь к папке, куда они будут установлены. Приложение ScreenBoard, к примеру, по умолчанию будет установлено в папку AIR Examples (которая находится в каталоге Applications или Program Files).

Та-дам! Все готово. Не так уж и сложно. Если были установлены соответствующие флажки (рис. 1.9 и 1.10), по окончании установки приложение запустится автоматически.

СОВЕТ

Если вы попытаетесь установить ранее установленное AIR-приложение, на экране появится уведомление о том, что приложение уже существует (рис. 1.11). В случае, если устанавливается более новая версия, вам будет предложено обновить установленное приложение.

УДАЛЕНИЕ ПРИЛОЖЕНИЯ

Для удаления приложения в Windows откройте Control panel (Панель управления) и выберите в ней опцию **Add or Remove Programs** (Установка и удаление программ). Для деинсталляции приложения в Mac OS X удалите соответствующий файл, расположенный по умолчанию в папке Applications. Альтернативный вариант удаления приложения в обеих платформах — повторный запуск установочного файла AIR-приложения. В этом случае на экране появится окно с кнопкой **Uninstall** (Удалить) (см. рис. 1.11).

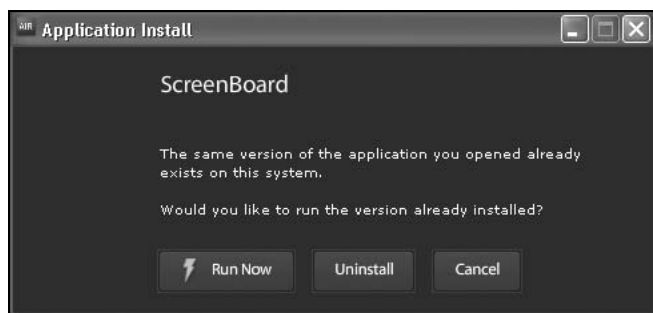


Рис. 1.11. Такой результат получается при инсталляции ранее установленного приложения той же версии

Запуск приложения

После успешной установки AIR-приложения появляется несметное число способов его выполнения. Все эти варианты вы вполне сможете обнаружить и сами, т. к. вскоре убедитесь, что, после одного раза установки среды исполнения и приложения любое AIR-приложение функционирует точно так же, как и все остальные программы на вашем компьютере.

Варианты выполнения приложения:

- ❑ во время установки поставьте флажок **Start application after installation** (Запустить приложение после установки) (см. рис. 1.9 и 1.10);
- ❑ в операционной системе Windows воспользуйтесь ярлыком на рабочем столе, если таковой был создан (рис. 1.12);



Рис. 1.12. Ярлык для приложения ScreenBoard, расположенный на рабочем столе Windows

- ❑ в операционной системе Windows воспользуйтесь меню **Start** (Пуск) (рис. 1.13);
- ❑ дважды щелкните мышью на значке приложения в Mac OS X. Большинство приложений по умолчанию будет установлено в папку Applications (рис. 1.14);
- ❑ в Mac OS X добавьте приложение на панель, чтобы можно было его запустить оттуда (рис. 1.15).

СОВЕТ

- Изображение значка ScreenBoard используется по умолчанию для любого AIR-приложения. Вы можете сами создать значок и использовать его вместо стандартного для своего AIR-приложения.
- Приложение ScreenBoard позволяет рисовать указателем мыши поверх открытых на экране окон других программ (рис. 1.16). Конечно, это приложение не является показателем всех возможностей применения технологии AIR, однако оно наглядно демонстрирует удобство использования этой не наносящей никому вреда граффити.

- Если вы создали или изменили файл `mms.cfg`, который используется программой Flash Player, могут возникнуть проблемы при выполнении AIR-приложений. Чтобы устранить ошибку, временно удалите этот файл из его исходной папки (например, в Windows XP — это `C:\winnt\system32\macromedia\flash\mms.cfg`, а в Mac OS X — `/Library/Application Support/Macromedia/mms.cfg`). Однако зачастую этот конфликт не возникает.



Рис. 1.13. В меню **Start** операционной системы Windows все AIR-приложения находятся в пункте **All Programs**



Рис. 1.14. Приложение ScreenBoard в Mac OS X



Рис. 1.15. Приложение ScreenBoard на панели

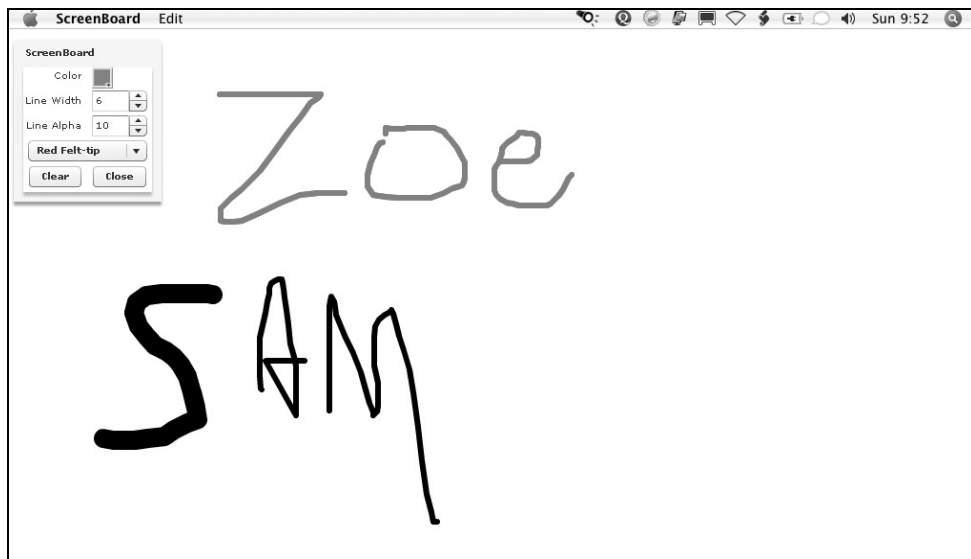


Рис. 1.16. Программа ScreenBoard в действии (на Mac OS X)

ГЛАВА 2



Создание приложения

При создании Adobe AIR-приложения у разработчика всегда есть выбор технологий и инструментальных средств. Существуют три варианта технологий: Ajax (HTML и JavaScript), Adobe Flex и Adobe Flash. Данная книга посвящена исключительно разработке на Ajax. Перечень инструментальных средств практически безграничен. В этой главе описана последовательность действий по созданию AIR-приложения в текстовом редакторе с использованием Adobe AIR SDK (Software Development Kit, набор инструментальных средств разработки). А в следующей главе я расскажу о применении интегрированных сред разработки (IDE, Integrated Development Environment) Dreamweaver CS3 и Aptana Studio вместо текстового редактора.

Разработка AIR-приложения начинается с создания рабочей папки проекта. Затем вы создаете два текстовых файла: HTML-файл и XML-файл. На заключительном этапе выполняется тестирование и компоновка обоих файлов посредством Adobe AIR SDK. Так как для разработки приложения необходимы AIR-утилиты, первые несколько разделов главы посвящены установке SDK и настройке операционной системы. Эти действия являются исходными и выполняются всего один раз.

Установка SDK

Adobe AIR SDK — это не то же самое, что среда исполнения AIR, используемая для выполнения приложений. SDK содержит множество элементов:

- два инструмента командной строки для тестирования и упаковки приложений;
- framework (библиотеки с полезным кодом);

- образцы и примеры (например, пиктограммы для приложений);
- шаблон XML-файла.

Для успешной работы SDK необходимо установить либо среду исполнения Java (JRE, Java Runtime Environment), либо набор средств разработки Java (JDK, Java Development Kit) на персональный компьютер. Обе программы являются свободно распространяемыми и доступны на сайте <http://java.sun.com>, как составная часть Java SE (Java Standard Edition, стандартная версия Java). После того как вы установите одну из программ (пользователям Mac повезло, т. к. у них JRE уже установлена; пользователям Windows предстоит выполнить установку самостоятельно), можете приступить к выполнению последующих этапов.

Этапы установки SDK

Выполните следующие шаги:

1. На сайте компании Adobe скачайте SDK для той операционной системы, которая установлена на вашем компьютере (рис. 2.1).

SDK доступен по адресу <http://www.adobe.com/products/air/tools/sdk/>. На момент написания книги на сайте был представлен SDK для Windows и Mac OS X.



Рис. 2.1. Скачайте SDK под свою операционную систему (вам также предстоит принять лицензионное соглашение)

2. Откройте и распакуйте скачанный архивный файл (рис. 2.2).

Пользователям Windows нужно распаковать ZIP-архив, а пользователям Mac — смонтировать DMG-файл (disc image, образ диска).

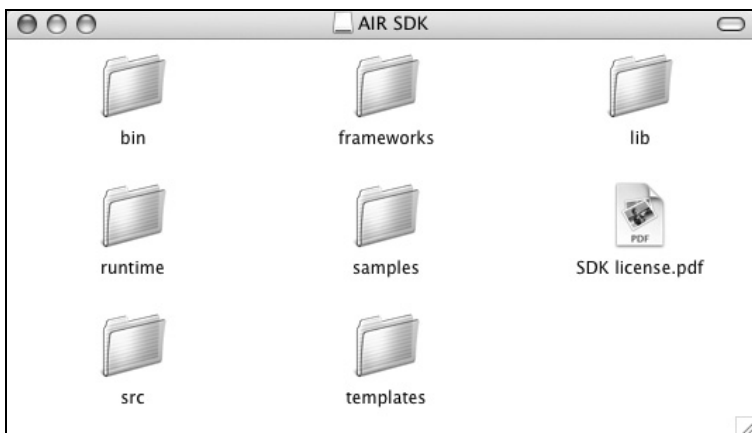


Рис. 2.2. Содержимое архивного файла Adobe AIR SDK

3. Скопируйте все содержимое скачанного файла в отдельную папку на компьютере.

Куда вы поместите SDK — дело ваше. Можете создать новую папку на рабочем столе или в корневом каталоге на жестком диске. Какой бы каталог вы ни выбрали, запомните его месторасположение. Оно понадобится вам в дальнейшем.

4. Обновите системный путь, чтобы он содержал SDK-папку bin.

О том, что это значит и как это осуществить, рассказывается в следующих двух разделах данной главы.

СОВЕТ

- JRE и JDK — это требования сугубо AIR SDK. Конечным пользователям нужно будет установить лишь среду исполнения Adobe AIR (см. главу 1).
- Вместе с SDK на сайте Adobe можно скачать документацию по разработке AIR-приложений, а также рабочие примеры с исходным кодом.

Обновление пути в Windows

При инсталляции SDK на компьютере устанавливаются два инструментальных средства — средство отладки (ADL, AIR Debug Launcher) и средство

разработки (ADT, AIR Development Tool). Оба инструмента используются из командной строки. Это значит, что вы будете выполнять их в Windows из командной строки, а не в графическом интерфейсе. Синтаксис командной строки очень прост, но для успешной работы нужно, чтобы ваш компьютер "знал" обе программы. Для этого необходимо добавить SDK-папку bin в системный путь.

Путь — это просто перечень мест, где компьютер будет искать программы при наличии соответствующего запроса. Скорее всего, вы не сталкивались с этим термином прежде, потому что просто не работали с приложениями, которые запускаются из командной строки. Достаточно выполнить несложную последовательность действий и все будет в порядке.

Изменение системного пути

Выполните действия для изменения системного пути:

1. Закройте все консольные окна, если таковые были открыты.

Новый путь вступит в силу для всех консольных окон (известных также под названием *командных окон DOS*), открытых после внесения изменений. Чтобы избежать проблем в будущем, закройте все открытые консольные окна перед изменением пути.

2. Щелкните правой кнопкой мыши на значке **My Computer** (Мой компьютер). В контекстном меню выберите пункт **Properties** (Свойства) (рис. 2.3), чтобы получить доступ к свойствам системы (окно **System Properties**).



Рис. 2.3. Получение доступа к свойствам системы

3. В диалоговом окне **System Properties** (Свойства системы) откройте вкладку **Advanced** (Дополнительно) (рис. 2.4).

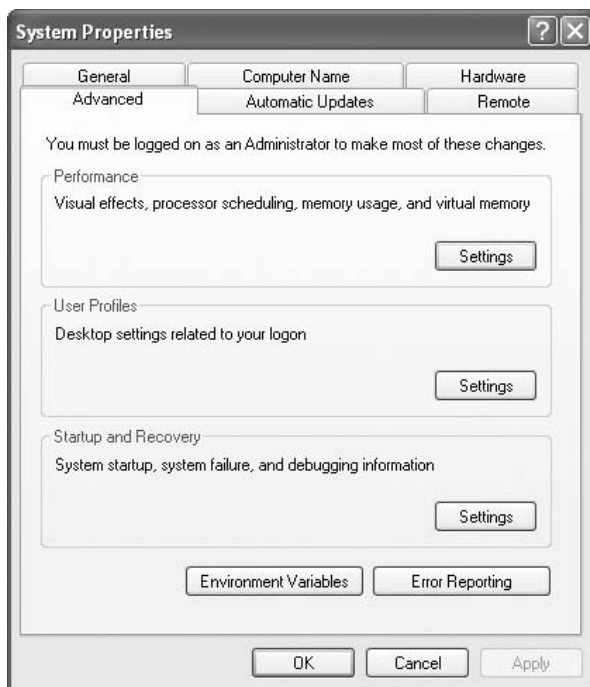


Рис. 2.4. Доступ к системному пути можно получить с помощью кнопки **Environment Variables**, расположенной на вкладке **Advanced**

4. Щелкните на кнопке **Environment Variables** (Переменные среды). Эта кнопка расположена внизу диалогового окна на рис. 2.4.
5. В диалоговом окне **Environment Variables** (Переменные среды) щелкните на системной переменной `Path` в области **System variables** (Системные переменные), чтобы выделить ее (рис. 2.5).
6. Щелкните на кнопке **Edit** (Изменить), чтобы открыть диалоговое окно **Edit System Variable** (Изменение системной переменной).
7. В конце строки **Variable value** (Значение переменной) добавьте точку с запятой и введите полный путь к SDK-папке `bin` (рис. 2.6).

Обратите внимание, что путь к SDK-папке `bin` нужно *дописать*, а не *заменить* имеющееся значение переменной.

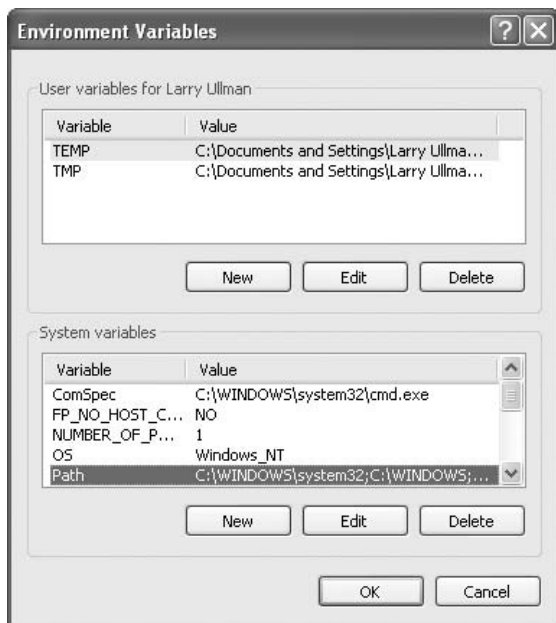


Рис. 2.5. Перечень редактируемых системных переменных расположен в нижней части окна **Environment Variables**



Рис. 2.6. В поле **Variable value** переменной `Path` перечислены все папки, где компьютер будет искать программы на выполнение. Путь к каждой папке отделен точкой с запятой

Можете скопировать путь к SDK-папке `bin` из адресной строки файлового менеджера Explorer (Проводник) (рис. 2.7), предварительно открыв SDK-папку `bin` с помощью графического интерфейса и мыши. Убедитесь, что вы не забыли добавить название самой папки `\bin` в текстовое поле **Variable value**, т. к. это очень важно.

- Щелкните на кнопке **OK** во всех диалоговых окнах, чтобы закрыть их.

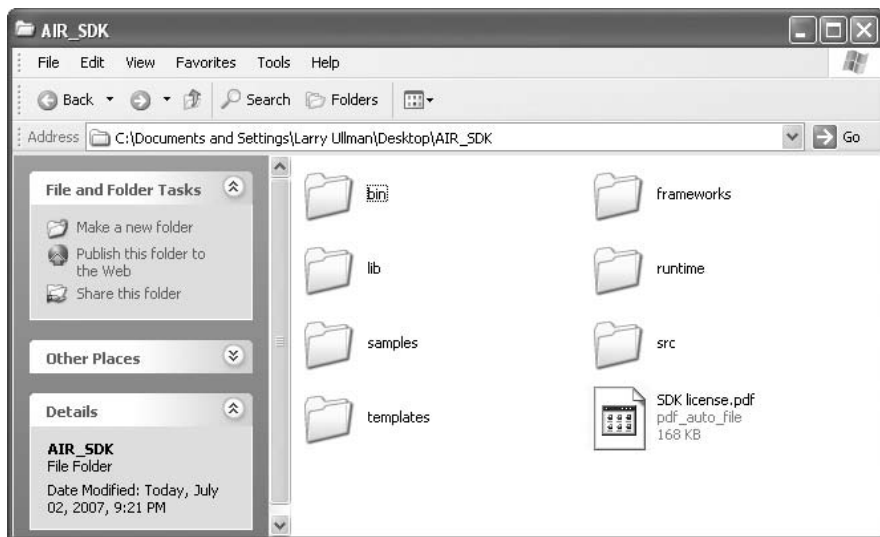


Рис. 2.7. В данном случае папка SDK расположена на рабочем столе. Полный путь к папке отображается в адресной строке файлового менеджера Explorer

СОВЕТ

На самом деле, вовсе не обязательно менять системный путь для работы с инструментами командной строки. Если вы не сделаете этого, то при использовании этих инструментов вам каждый раз придется вводить в консольном окне что-то вроде `C:\\"Documents and Settings\" \"Larry Ullman\" \Desktop\SDK\bin\adt` вместо того, чтобы просто вызвать команду, как `adt`. Поэтому изменение пути сэкономит ваше время в дальнейшем.

Обновление пути в Mac OS X

Два инструментальных средства, установленных SDK — средство отладки (ADL, AIR Debug Launcher) и средство разработки (ADT, AIR Development Tool), — функционируют из командной строки. Это означает, что вы сможете запускать эти программы на выполнение в терминальном приложении, а не через графический интерфейс. Синтаксис командной строки предельно прост, но для успешной работы нужно, чтобы ваш компьютер "знал" обе программы. Для этого необходимо добавить SDK-папку `bin` в системный *путь*.

Изменение системного пути

Выполните действия для изменения системного пути:

1. Закройте все терминальные окна, если таковые были открыты.

Новый путь вступит в силу для всех терминальных окон, открытых после внесения изменений. Чтобы избежать проблем в будущем, перед изменением пути закройте все открытые терминальные окна.

2. Узнайте, какую оболочку (shell) вы используете. Для этого откройте окно **Terminal Inspector** (Центр настроек консоли) (рис. 2.8), выбрав **Terminal | Window Settings** (Терминал | Настройки окна).

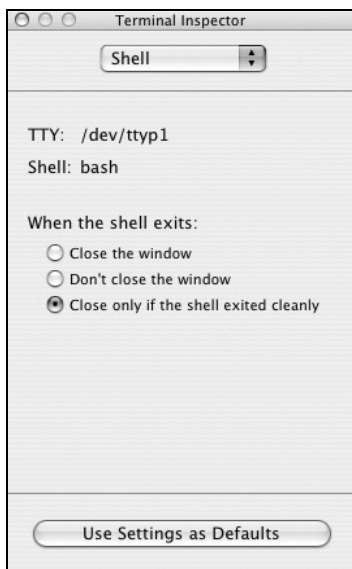


Рис. 2.8. В системах UNIX, в том числе в Mac OS X, чтобы изменить путь, вам необходимо знать разновидность используемой оболочки. Узнать текущую оболочку можно в окне **Terminal Inspector**

Порядок изменения пути зависит от используемой оболочки (если вас интересует значение термина "оболочка", задайте поисковый запрос *оболочка unix* или *unix shell*). В окне **Terminal Inspector** отобразится текущая оболочка. Далее перечислены наиболее часто используемые оболочки в Mac OS (названия программ указаны в скобках):

- Bourne (sh);

- Bourne Again Shell (bash);
- оболочка C (csh);
- оболочка T или TC (tcsh);
- оболочка Korn (ksh).

Последние версии Mac OS X снабжены оболочкой bash (как на рис. 2.8). Поэтому я предполагаю, что у вас установлена именно она. Если в вашем окне **Terminal Inspector** отображается другая оболочка, придется поискать информацию по смене пути для вашей оболочки в Интернете.

3. Откройте терминальное окно (**File | New Shell** (Файл | Новая оболочка) или комбинация клавиш <Command>+<N>), если не сделали этого ранее (рис. 2.9).

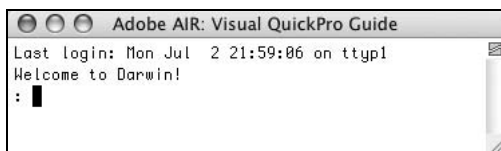


Рис. 2.9. Новое терминальное окно

4. Перейдите в домашнюю папку. Для этого введите команду `cd` и нажмите клавишу <Return>.

Эти действия не требуются, если при открытии нового терминального окна вы уже находитесь в домашнем каталоге. Но для того чтобы наверняка все было в порядке, все равно стоит перейти в домашний каталог. Команда `cd` применяется для смены текущей папки. Отсутствие каких-либо параметров этой команды (например, отсутствие названия каталога) приведет к переходу в домашнюю папку.

5. Для того чтобы в терминальном окне отобразить перечень файлов домашнего каталога, введите команду `ls -a` и нажмите клавишу <Return> (рис. 2.10).

Команда `ls` возвращает содержимое папки, а параметр `-a` свидетельствует о том, что должны отображаться все файлы, в том числе и скрытые.

6. Если файл с именем `.bash_profile` отсутствует в перечне (см. рис. 2.10), создайте новый. Для этого введите команду `touch .bash_profile`.

Специальные или скрытые файлы в Mac OS X начинаются с точки. Файл `.bash_profile` предназначен для описания поведения оболочки. Если файл отсутствует, команда `touch` создаст его.

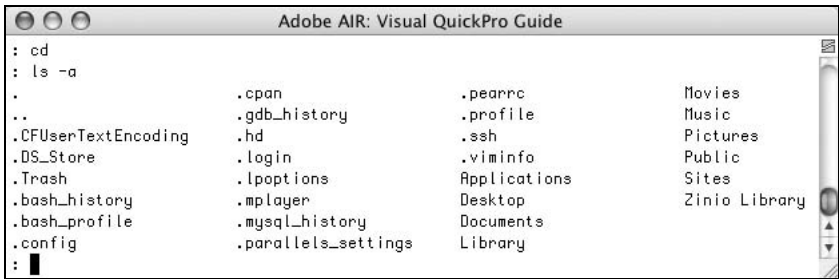


Рис. 2.10. Полный перечень файлов в моем домашнем каталоге

7. С помощью любого текстового редактора откройте файл `.bash_profile`. Лично я использую популярный (и самый лучший) текстовый редактор `BBEdit`. Для того чтобы открыть файл из командной строки, мне нужно ввести `bbedit .bash_profile`. С тем же успехом можно использовать свободно распространяемый текстовый редактор `TextWrangler` от [Bare Bones \(www.barebones.com\)](http://www.barebones.com) или один из многочисленных редакторов командной строки: `vi`, `vim`, `emacs`, `pico` и т. д.
8. Добавьте строку `export PATH="$PATH:/path/to/AIR/SDK/bin/"` в файл `.bash_profile`.

Команда `export PATH` позволяет изменить путь оболочки `bash`. В качестве значения указывается текущий путь (представленный символами `$PATH`) плюс полный путь к SDK-папке `bin` (вы введете реальный путь вместо `/path/to/`, см. первый совет далее). В пути каждая папка отделена двоеточием. Если ваш документ `.bash_profile` уже содержит строку `export PATH...`, просто добавьте двоеточие и полный путь к SDK-папке `bin`.

9. Сохраните и закройте файл.
10. Закройте терминальное окно.

Изменения пути вступят в силу при следующем открытии терминального окна.

СОВЕТ

- В программах `Mac OS X` вы можете вставить полный путь к папке в текст файла, просто перетащив нужную папку в файл. Например, если вы перетащите SDK-каталог `bin` из файлового менеджера `Finder` в файл `.bash_profile`, открытый в `BBEdit`, полный путь к каталогу `bin` автоматически вставится в то место файла `.bash_profile`, где вы отпустите кнопку мыши.

- На самом деле вовсе не обязательно менять путь для работы с инструментами командной строки. Если вы не сделаете этого, просто при использовании этих инструментов вам каждый раз придется вводить в консольном окне что-то вроде `/Users/larryullman/Desktop/AIR/SDK/bin/adt` вместо того, чтобы просто вызвать команду, как `adt`. Поэтому изменение пути сэкономит ваше время в дальнейшем.

Создание структуры проекта

Хотя вовсе не обязательно придерживаться какой-то определенной структуры папки для AIR-приложений, я считаю, что лучше это все-таки сделать. Любой опытный Web-разработчик и профессиональный программист привык логически организовывать файлы Web-сайта. Далеко не все проекты имеют одну и ту же структуру. К тому же, вы можете избрать разные схемы именования файлов. Однако есть несколько общих принципов, которых стоит придерживаться.

Обратите внимание на те базовые подходы, которые помогут удачно организовать файлы проекта. В данной главе читателям представлена возможность создать свой первый проект и структурировать файлы в нем. В рамках этого проекта не понадобятся папки для CSS- и JavaScript-файлов. Я о них упоминаю лишь для того, чтобы у вас была исчерпывающая информация.

Этапы создания структуры приложения

Выполните следующие шаги:

1. Создайте новую папку для AIR-приложений в любом месте на вашем компьютере.

Вы наверняка знаете, как это сделать (в операционной системе Windows щелкните правой кнопкой мыши и выберите пункт контекстного меню **New | Folder** (Создать | Папка); в Mac OS X используйте меню **File | New Folder** (Файл | Новая папка) или нажмите комбинацию клавиш `<Command>+<Shift>+<N>`). Например, вы можете создать папку AIR Development на рабочем столе или в каталоге со своими документами. Важно, чтобы все папки и файлы AIR-приложений, с которыми вы работаете, хранились в одном месте. Тогда будет быстрее и проще разрабатывать и компоновать приложения.

2. Внутри папки, сформированной на предыдущем этапе, создайте новый каталог для своего проекта.

Создавайте отдельную папку для каждого нового приложения. Все эти папки разместите внутри одного родительского каталога AIR Development.

3. Задайте то же имя новой папке проекта, что и приложению (рис. 2.11).

Для первого примера я выбрал старое доброе имя — "Hello, World!" С него начитается любое руководство по программированию. И еще я создал одноименную папку HelloWorld, в которой будут расположены все файлы проекта.

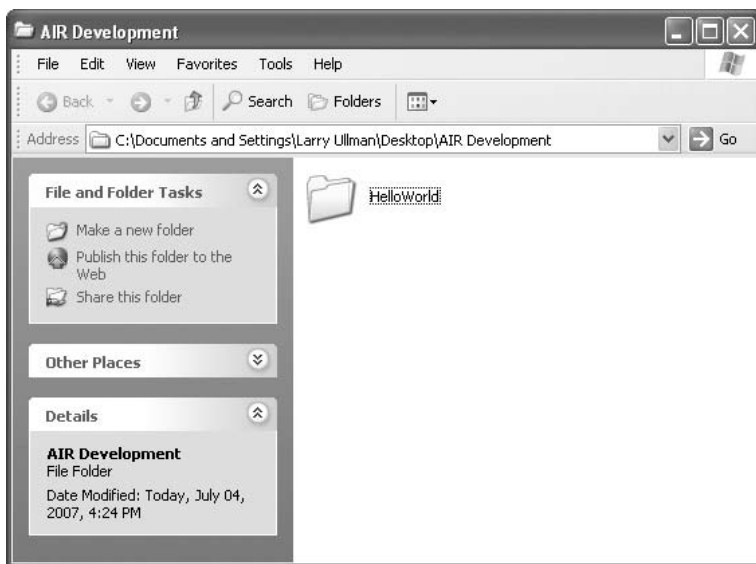


Рис. 2.11. Папка HelloWorld будет содержать файлы вашего первого AIR-приложения

4. Внутри папки HelloWorld (или папки с другим именем, которое было задано на этапе 3) создайте каталог для таблиц каскадных стилей (CSS, Cascading Style Sheets).

По очевидным причинам я назвал эту папку css. В ней будут храниться все CSS-файлы, используемые приложением. Основой AIR-приложений является HTML, поэтому следует логически организовать папку проекта, как если бы это был Web-сайт.

5. Внутри папки проекта создайте каталог для файлов JavaScript. Не удивительно, что я назвал эту папку `js`. Все файлы сценариев будут храниться здесь.

6. Создайте каталог для изображений внутри папки проекта.

Назовем эту папку `images`. Конечно, вы можете дать ей любое другое имя, например, `assets` или `imgs`. По большому счету совершенно не важно, как будет называться папка с графикой для AIR-приложения.

7. Создайте каталог для пиктограмм в папке вашего проекта (рис. 2.12).

Я решил назвать эту папку `icons`. В *главе 16* вы научитесь устанавливать уникальные значки для своих приложений. Пиктограммы — это не то же самое, что изображения и графика, используемые программой.

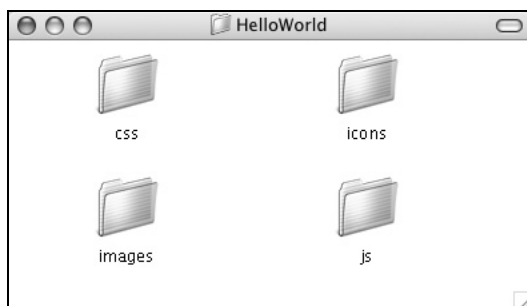


Рис. 2.12. Структура основной папки проекта

СОВЕТ

Вы можете создать и другие каталоги внутри папки проекта. Например, создать отдельную папку для хранения аудиозаписей и назвать ее `audio`; другую папку — для документов — назвать `docs`; создать еще одну папку, где хранить все остальные элементы, необходимые для корректной работы приложения, и назвать ее `resources`. Опять же, эти имена не являются догмой, я лишь предлагаю оптимальный, с моей точки зрения, вариант.

Создание HTML-файла

Первый файл приложения, который я создам, будет файл HTML (так сказать, главная страница). При разработке Adobe AIR-приложений с помо-

стью HTML и JavaScript этот HTML-файл по существу и будет вашим приложением.

Одно из основных преимуществ AIR заключается в том, что вы можете применять уже имеющиеся знания по Web-разработке для создания настольных приложений. Иными словами, разработчик использует умения и технологии проектирования Web-интерфейса для проработки внешнего вида своего AIR-приложения. К тому же, вы можете запустить HTML-файл приложения в браузере и посмотреть, как будет выглядеть ваша программа.

Для этого демонстрационного примера я выбрал классический вариант приложения "Hello, World!" В сущности, никаких полезных операций программа выполнять не будет. Она приводится просто в качестве примера этапов разработки, упаковки и выполнения AIR-приложений. Эта последовательность действий пригодится вам до самого конца книги.

Этапы создания HTML-файла

Выполните следующие шаги:

1. С помощью текстового редактора или IDE создайте новый HTML-документ (листинг 2.1).

```
<html>
  <head>
    <title>Hello, World!</title>
  </head>
```

Так как реальный Web-сайт не разрабатывается, вполне можете опустить лишние HTML-элементы, например, теги DOCTYPE и META, и т. п.

Листинг 2.1. Этот HTML-файл является основой AIR-приложения

```
<html>
  <head>
    <title>Hello, World!</title>
  </head>
  <body>
    <h1>Hello, World!</h1>
  </body>
</html>
```

2. Добавьте контейнерный тег `<body>`.

```
<body>
  <h1>Hello, World!</h1>
</body>
```

Ранее уже упоминалось, что это приложение не будет чем-то сверхъестественным, но оно даст читателю возможность увидеть, насколько просто создавать кроссплатформенные настольные приложения.

3. Закройте HTML-тег.

```
</html>
```

4. Сохраните файл в папке проекта с именем `HelloWorld.html`.

Этот файл должен быть сохранен в самой папке проекта — в ее корне, а не в созданных подкаталогах.

СОВЕТ

- HTML rendering engine (движок рендеринга HTML) — это интерпретатор HTML-кода в графическое представление — используется AIR и называется WebKit (www.webkit.org). На этом же движке работает Web-браузер Safari фирмы Apple. Поэтому HTML-документ, открытый в Safari, будет иметь максимальное сходство по внешнему виду и функциям с AIR-приложением. Третья версия браузера Safari совместима с двумя операционными системами: Mac и Windows.
- AIR-приложения могут быть написаны на Ajax, Flash или Flex. Основным файлом AIR-приложения на Ajax является файл с расширением `html`. Для Flash и Flex AIR-приложений основной файл — `swf` (Shockwave format).

Создание XML-файла

Кроме HTML-файла, ваши AIR-приложения должны также включать XML-файл, который играет роль дескриптора. Этот файл содержит *метаданные* приложения (информацию о программе). В состав метаданных могут входить следующие компоненты:

- название;
- версия;
- автор;

- описание;
- авторское право;
- пиктограммы;
- установочная папка по умолчанию;
- внешний вид окна и его поведение;
- многое другое.

Большая часть этих данных отображается в меню приложения **About** (О программе) (рис. 2.13), а также в процессе установки (рис. 2.14).

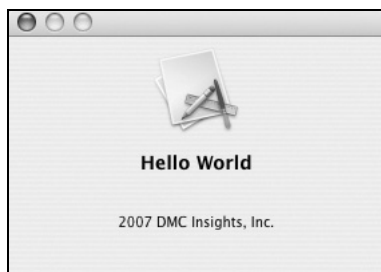


Рис. 2.13. В окне **About** содержатся сведения о названии, авторе и авторском праве на приложение

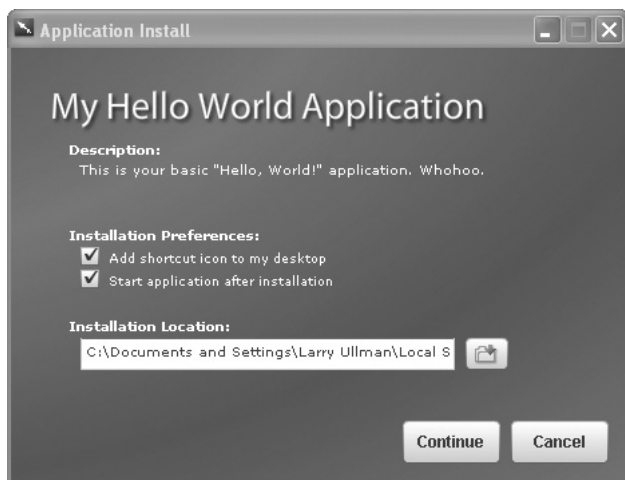


Рис. 2.14. В процессе установки отображается некоторая информация файла-дескриптора

Если вы никогда не работали с XML, не переживайте: этот язык не слишком отличается от HTML. Я детально расскажу обо всем, что вам нужно знать. Здесь я сделаю акцент на *обязательные* XML-данные. В других же главах более подробно остановлюсь на дополнительных настройках, которые могут быть заданы в XML-файле, а в *главе 16* описано еще несколько полезных опций.

Этапы создания XML-файла

Выполните следующие шаги:

1. В текстовом редакторе или IDE создайте новый XML-документ (листинг 2.2).

XML-файл — это обычный текстовый документ, который создается в любом текстовом редакторе. XML-файл начинается с объявления версии XML (в данном случае 1.0) и кодировки (см. первый совет далее).

Листинг 2.2. XML-файл — это дескриптор, необходимый для каждого AIR-приложения

```
<?xml version="1.0" encoding="utf-8" ?>
<application xmlns="http://ns.adobe.com/air/application/1.0">
  <id>com.dmci.air.HelloWorld</id>
  <filename>Hello World</filename>
  <version>1.0</version>
  <initialWindow>
    <content>HelloWorld.html</content>
    <visible>true</visible>
  </initialWindow>
</application>
```

2. Добавьте контейнерный тег `<application>`.

```
<application>
</application>
```

Все XML-файлы содержат этот базовый тег (как и во всех HTML-файлах есть контейнерный тег `<HTML>`). В файле-дескрипторе AIR-приложения обязательно должен присутствовать элемент `application`, т. к. остальные данные XML расположены между его открывающим и закрывающим тегами.

3. Добавьте атрибут `xmlns` в открывающий тег `<application>`.

```
<application xmlns="http://ns.adobe.com/air/application/1.0">
```

Этот атрибут предназначен для определения *пространства имен XML* (XML namespace). Пространство имен — это специальная концепция программирования, в которую можно особенно не вникать. Достаточно знать, что значение атрибута указывает на самую раннюю версию среды исполнения AIR, которую поддерживает приложение. Значение данного атрибута соответствует первой официальной версии AIR. Все программы, в которых указано это пространство имен, должны корректно работать с любой более поздней версией среды исполнения AIR.

4. Между открывающим и закрывающим тегами `<application>` добавьте вложенный тег `<id>`:

```
<id>com.dmci.air.HelloWorld</id>
```

Значение элемента `id` является уникальной ссылкой на AIR-приложение. Рекомендуется использовать синтаксис `com.название_компании.имя_приложения`. Это название вы придумываете сами, но оно должно быть уникальным и значащим. К примеру, `id` у Adobe AIR-приложения будет `com.adobe.air.имя_приложения`. Для приложения, созданного моей компанией (DMC Insights, Inc.), я выбрал `id` — `com.dmci.air.HelloWorld`. Поменяйте это название на что-то близкое вам.

Допускается применять латинские буквы A–Z, цифры 0–9, точки и тире. В `id` не допускается использование пробелов. Максимальная длина `id` составляет 255 символов.

5. Между открывающим и закрывающим тегами `<application>` добавьте вложенный тег `<filename>`:

```
<filename>Hello World</filename>
```



Рис. 2.15. Надпись под ярлыком соответствует имени приложения, указанного во вложенном теге `<filename>` XML-файла

Данное значение будет видно всем пользователям, как имя приложения. Оно отображается в меню **About** (О программе) (см. рис. 2.13), в надписях

к ярлыкам (рис. 2.15), в меню **Start** операционной системы Windows и т. д. А значение элемента `id`, наоборот, находится, как бы, за кулисами. Значение `id` доступно только для программ, в то время как конечный пользователь его не видит.

6. Между открывающим и закрывающим тегами `<application>` добавьте вложенный тег `<version>`.

```
<version>1.0</version>
```

Значение данного элемента указывает на версию приложения. Это вымышленное значение, тем не менее, должно быть значимым. Бета-версиям приложений обычно присваивается номер, больший 1. При значительных изменениях приложения последующей версии принято присваивать ближайшее целое число, идущее по порядку (например, обновление от 1 ко 2 версии, от 2 — к 3 и т. д.). При менее серьезных изменениях обычно меняют только десятые доли версии, например, обновление от версии 1.1 к 1.2. Самое главное, что каждой последующей версии приложения присваивается номер, следующий по возрастанию. Таким образом, пользователю становится сразу ясно, когда была обновлена та или иная версия продукта.

Из листинга 2.2 видно, что все элементы (или вложенные контейнерные теги) расположены между открывающим и закрывающим тегами `<application>`. Причем совершенно неважен порядок, в котором они записаны.

7. Между открывающим и закрывающим тегами `<application>` добавьте вложенный тег `<initialWindow>`.

```
<initialWindow>  
</initialWindow>
```

Контейнерный тег `<initialWindow>` содержит элементы, значения которых обуславливают содержимое и внешний вид окна исходного приложения.

8. Между открывающим и закрывающим тегами `<initialWindow>` добавьте вложенный тег `<content>`.

```
<content>HelloWorld.html</content>
```

Элементу `content` соответствует значение основного HTML-файла (см. листинг 2.1). Лучше, когда оба файла (и HTML, и XML) хранятся в одной и той же папке, тогда в качестве значения элемента `content` достаточно одного имени HTML-файла. В противном случае придется указывать относительный путь от XML- к HTML-файлу (к примеру, `../HelloWorld.html` или `content/HelloWorld.html`).