
Flex 4 Cookbook

*Joshua Noble, Todd Anderson, Garth Braithwaite,
Marco Casario, and Rich Tretola*

O'REILLY®

Beijing • Cambridge • Farnham • Köln • Sebastopol • Taipei • Tokyo

Джошуа Ноубл,
Тодд Андерсон, Гарт Брэйтуэйт,
Марко Казарио, Рич Третола

Flex 4

РЕЦЕПТЫ ПРОГРАММИРОВАНИЯ

«Русская редакция»

«БХВ-Петербург»

2011

УДК 681.3.06
ББК 32.973.26-018.2
Н72

Ноубл, Д.

Н72 Flex 4. Рецепты программирования: Пер. с англ / Д. Ноубл, Т. Андерсон, Г. Брэйтуэйт, М. Казарио, Р. Третола. — М.: Издательство «Русская редакция»; СПб.: БХВ-Петербург, 2011. — 720 с.: ил.

ISBN 978-5-9775-0773-8 («БХВ-Петербург»)
ISBN 978-5-7502-0408-3 («Русская редакция»)

Книга построена как справочник по решению типовых задач, что позволяет лучше понять особенности платформы Flex Framework. Рассказывается о том, как обеспечить взаимодействие различных компонентов и как объединить Flex с другими технологиями для создания полнофункциональных интернет-приложений. Рассматриваются основы работы с Flex и программирование на ActionScript, списки и визуализаторы элементов, связывание и проверка допустимости данных, форматирование и регулярные выражения, работа со службами, взаимодействие с сервером и браузером, работа с данными в среде AIR, интеграция операционной системы со средой AIR, тестирование модулей с помощью FlexUnit и др.

Для веб-дизайнеров и разработчиков сайтов

УДК 681.3.06
ББК 32.973.26-018.2

Подготовлено к изданию по лицензионному договору с O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472.

Nutshell Handbook, логотип Nutshell Handbook, логотип O'Reilly, а также O'Reilly Media, Inc. Flex 4 Cookbook и изображение на обложке являются товарными знаками или охраняемыми товарными знаками O'Reilly Media, Inc. в США и/или других странах. Все другие товарные знаки являются собственностью соответствующих фирм.

Все адреса, названия компаний, организаций и продуктов, а также имена лиц, используемые в примерах, вымышлены и не имеют никакого отношения к реальным компаниям, организациям, продуктам и лицам.

Джошуа Ноубл, Тодд Андерсон, Гарт Брэйтуэйт, Марко Казарио, Рич Третола

Flex 4. Рецепты программирования

Перевод с английского Сергея Иноземцева

Совместный проект издательства «Русская редакция» и издательства «БХВ-Петербург»

 РУССКАЯ РЕДАКЦИЯ

 **bhv**[®]

Лицензия ИД № 02429 от 24.07.00. Подписано в печать 21.07.11.

Формат 70×100^{1/16}. Печать офсетная. Усл. печ. л. 58,05.

Тираж 1200 экз. Заказ №

Санитарно-эпидемиологическое заключение на продукцию
№ 77.99.60.953.Д.005770.05.09 от 26.05.2009 г. выдано Федеральной службой
по надзору в сфере защиты прав потребителей и благополучия человека.

Отпечатано с готовых диапозитивов
в ГУП "Типография "Наука"
199034, Санкт-Петербург, 9 линия, 12.

ISBN 78-0-596-80561-6 (англ.)
ISBN 978-5-9775-0773-8 («БХВ-Петербург»)
ISBN 978-5-7502-0408-3 («Русская редакция»)

© 2010 O'Reilly Media, Inc.
© Перевод на русский язык, издательство «БХВ-Петербург», 2011
© Оформление и подготовка к изданию, издательство «БХВ-Петербург»,
издательство «Русская редакция», 2011

Оглавление

Предисловие	1
Для кого эта книга.....	1
Кому не следует читать эту книгу	2
Как организована эта книга	2
Обозначения, принятые в книге	3
Использование кода	3
Как использовать эту книгу	4
Сборники рецептов от издательства O'Reilly	4
Цифровые библиотеки.....	4
Как с нами связаться	5
Благодарности	5
Об авторах	9
Глава 1. Основы работы с Flex и программирования на ActionScript	11
1.1. Создание Flex-проекта в среде Flash Builder.....	12
1.2. Создание проекта Flex Library в среде Flash Builder.....	15
1.3. Настройка опций компилятора в среде Flash Builder	17
1.4. Компиляция Flex-проекта вне среды Flash Builder.....	20
1.5. Добавление слушателя событий в MXML-файл.....	22
1.6. Создание типизированных векторов.....	24
1.7. Использование всплывания событий	25
1.8. Использование нестандартных событий и передача данных с событиями.....	27
1.9. Прослушивание событий клавиатуры.....	28
1.10. Определение необязательных параметров у методов	29
1.11. Определение и реализация интерфейса	30
1.12. Обращение к родителю Flex-компонента.....	32
Глава 2. Контейнеры	35
2.1. Расположение элементов в контейнере	36
2.2. Динамическое добавление и удаление потомков	38

2.3. Переупорядочивание потомков в контейнере.....	41
2.4. Вывод потомков на экран с учетом элементов данных.....	43
2.5. Использование нестандартного визуализатора элементов в контейнере <i>DataGroup</i>	46
2.6. Использование нескольких визуализаторов элементов в контейнере <i>DataGroup</i>	49
2.7. Прокрутка в контейнере.....	52
2.8. Масштабирование потомков контейнера.....	57
2.9. Применение скинов к контейнеру.....	59
2.10. Установка фонового изображения в контейнере <i>BorderContainer</i>	64
2.11. Использование панели элементов управления.....	66
2.12. Изменение компоновки элементов содержимого в контейнере <i>Panel</i>	68
2.13. Отслеживание положения мыши в контейнере.....	71
2.14. Перетаскивание элементов с одного визуального контейнера на другой.....	73
2.15. Перетаскивание элементов с одного контейнера данных на другой.....	78
2.16. Добавление компоновочного контейнера пакета Spark в навигационный контейнер пакета MX.....	82
2.17. Создание Spark-эквивалента для контейнера <i>ViewStack</i>	84

Глава 3. Компоновка..... 91

3.1. Линейное расположение потомков.....	92
3.2. Переключение способа компоновки на этапе выполнения.....	93
3.3. Выравнивание элементов контейнера в компоновке.....	95
3.4. Размещение потомков по строкам и столбцам.....	98
3.5. Унификация размера потомков.....	100
3.6. Создание по мере необходимости и повторное использование потомков.....	102
3.7. Создание нестандартной компоновки.....	105
3.8. Измерение и изменение размера контейнера.....	107
3.9. Динамическое изменение глубины потомков в компоновке.....	109
3.10. Использование класса <i>Matrix3D</i> для преобразований внутри компоновки.....	111
3.11. Использование класса <i>TransformOffsets</i> для преобразований внутри компоновки.....	113
3.12. Создание нестандартной трехмерной компоновки.....	117
3.13. Программная реализация прокрутки в компоновке.....	120
3.14. Определение видимости элементов в последовательной компоновке.....	123

Глава 4. Графика..... 127

4.1. Управление размером и местоположением графического элемента.....	129
4.2. Использование класса <i>Path</i> для рисования фигуры с заданной заливкой и заданными характеристиками линии.....	130
4.3. Вывод текста в графический элемент.....	134
4.4. Вывод растрового изображения в графическом элементе.....	136

4.5. Вывод текста с градиентом	139
4.6. Наложение растровых данных в качестве маски на графический элемент	140
4.7. Создание нестандартного графического элемента	143
4.8. Создание отдельного нестандартного графического компонента	146
4.9. Определение и многократное использование графических символов	149
Глава 5. Компоненты	153
5.1. Обработка щелчка по кнопке	153
5.2. Создание панели с кнопками	156
5.3. Загрузка внешнего SWF-файла	158
5.4. Ввод даты с помощью календаря	159
5.5. Создание обработчиков событий для элементов управления меню	162
5.6. Вывод сообщения пользователю приложения	163
5.7. Вывод нестандартного всплывающего компонента	165
5.8. Распознавание щелчка мышью за пределами всплывающего компонента и закрытие его	167
5.9. Применение компонента <i>s:Scroller</i> для создания контейнера с прокруткой ...	168
5.10. Обработка событий <i>focusIn</i> и <i>focusOut</i>	169
5.11. Показ раскрывающегося списка в ответ на клавиатурное сокращение	170
5.12. Группирование кнопок-переключателей	172
5.13. Отправка Flex-формы серверному сценарию	174
Глава 6. Скинны и стили	177
6.1. Создание скина для элемента <i>s:ButtonSkin</i>	177
6.2. Накладывание повторяющегося фонового изображения на область приложения	179
6.3. Создание скина для компонентов <i>s:ButtonBar</i> и <i>s:ButtonBarButton</i>	181
6.4. Наложение скина на элемент управления <i>s:DropDownList</i>	183
6.5. Скин для Spark-контейнера	186
6.6. Изменение внешнего вида компонентов с помощью стилей	187
6.7. Наложение скинов и установка свойств Spark- и MX-компонентов с помощью таблиц CSS	189
6.8. Создание кнопки с пиктограммой	191
6.9. Добавление нестандартных стилизованных свойств	193
6.10. Частичное встраивание шрифтов с помощью CSS	195
Глава 7. Работа с текстом	197
7.1. Создание объекта <i>TextFlow</i>	198
7.2. Создание объекта <i>TextFlow</i> из другого источника	199
7.3. Создание ссылок в объекте <i>TextFlow</i>	201
7.4. Добавление графических элементов в объект <i>TextFlow</i>	202
7.5. Связывание значения с элементом управления <i>s:TextInput</i>	203
7.6. Создание нестандартного стиля, действующего избирательно	204

7.7. Стилизация ссылок в объекте <i>TextFlow</i>	206
7.8. Поиск элементов в объекте <i>TextFlow</i>	208
7.9. Поиск всех шрифтов, установленных на компьютере пользователя	209
7.10. Вертикальный вывод текста в текстовой области	211
7.11. Выделение текста в элементе <i>TextArea</i>	213
7.12. Управление внешним видом выделенного текста	214
7.13. Копирование символа в виде растрового изображения	215
7.14. Создание связанных контейнеров для объекта <i>TextFlow</i>	218
7.15. Использование нестандартного преобразователя формата	220
7.16. Создание скина для элемента управления <i>TextArea</i>	223
7.17. Вывод текста в несколько столбцов	224
7.18. Выделение последнего символа в объекте <i>TextFlow</i>	225
Глава 8. Списки и визуализаторы элементов	227
8.1. Создание визуализатора элементов для списка <i>Spark</i>	227
8.2. Создание редактируемого списка	229
8.3. Прокрутка к нужному элементу в <i>Spark</i> -списке	231
8.4. Изменение компоновки <i>Spark</i> -списка	231
8.5. Создание вложенного списка	232
8.6. Указание XML-данных для <i>Spark</i> -компонента <i>List</i>	235
8.7. Реализация <i>Spark</i> -компонента <i>List</i> , в котором можно выделить только некоторые элементы	237
8.8. Форматирование и проверка допустимости введенных данных в редакторе элементов <i>Spark</i> -компонента <i>List</i>	239
8.9. Создание контекстного меню для <i>Spark</i> -компонента <i>List</i>	244
8.10. Реализация перетаскивания в <i>Spark</i> -компоненте <i>List</i>	246
8.11. Изменение внешнего вида индикатора операции перетаскивания у <i>Spark</i> -компонента <i>List</i>	249
8.12. Вывод данных, асинхронно загруженных в <i>Spark</i> -компонент <i>List</i>	253
Глава 9. Таблицы	257
9.1. Определение нестандартных столбцов у компонента <i>DataGrid</i>	257
9.2. Определение функций сортировки столбцов компонента <i>DataGrid</i>	261
9.3. Фильтрация элементов в компоненте <i>DataGrid</i>	262
9.4. Создание заголовков для компонента <i>DataGrid</i>	265
9.5. Обработка событий компонента <i>DataGrid</i>	267
9.6. Реализация перетаскивания в компоненте <i>DataGrid</i>	272
9.7. Редактирование элементов в компоненте <i>DataGrid</i>	273
9.8. Поиск внутри компонента <i>DataGrid</i> и автоматическая прокрутка к найденному элементу	275
9.9. Генерирование сводки для плоских данных с помощью объекта <i>GroupingCollection</i>	277
9.10. Асинхронное обновление объекта <i>GroupingCollection</i>	280

Глава 10. Видео	285
10.1. Создание базового видеоплеера	285
10.2. Вывод индикатора хода воспроизведения.....	286
10.3. Создание видеоплеера со скином	288
10.4. Вывод потокового видео	291
10.5. Вывод количества загруженных байтов видеоданных.....	292
10.6. Создание базового видеоплеера с помощью OSMF	293
10.7. Чтение и вывод точек поиска, встроенных в видеофайл	294
10.8. Создание оболочки для OSMF.....	297
10.9. Вывод титров с помощью OSMF.....	298
Глава 11. Анимация и эффекты	303
11.1. Динамическая настройка фильтра для компонента.....	304
11.2. Вызов анимации в MXML-разметке и в коде ActionScript	305
11.3. Создание эффектов, проявляющихся, когда компонент скрыт или показан	307
11.4. Определение ключевых кадров анимации	309
11.5. Создание параллельных эффектов или последовательностей эффектов	311
11.6. Пауза, рестарт и воспроизведение эффекта в обратном направлении.....	312
11.7. Создание эффектов, сопровождающих добавление или удаление компонента.....	313
11.8. Создание нестандартных анимационных эффектов.....	316
11.9. Применение фильтра <i>DisplacementMapFilter</i> во Flex-эффекте	318
11.10. Применение фильтра <i>ConvolutionFilter</i>	322
11.11. Применение фильтра <i>Pixel Bender</i> для создания перехода.....	326
Глава 12. Коллекции	331
12.1. Добавление, удаление и чтение данных из объекта <i>ArrayList</i>	331
12.2. Чтение и сортировка данных из объекта <i>ArrayCollection</i>	333
12.3. Фильтрация объекта <i>ArrayCollection</i>	335
12.4. Распознавание наличия изменений внутри объект <i>ArrayCollection</i>	336
12.5. Создание объекта <i>GroupingCollection2</i>	337
12.6. Создание иерархического провайдера данных для элемента управления	339
12.7. Навигация в коллекции и сохранение позиции.....	344
12.8. Создание объекта <i>HierarchicalViewCollection</i>	346
12.9. Фильтрация и сортировка объекта <i>XMLListCollection</i>	349
12.10. Сортировка по нескольким полям в коллекции.....	351
12.11. Сортировка элементов коллекции по дате	352
12.12. Создание глубокой копии объекта <i>ArrayCollection</i>	353
12.13. Использование объектов данных с уникальными идентификаторами.....	356
Глава 13. Связывание данных	359
13.1. Связывание со свойством.....	360
13.2. Связывание с функцией.....	362

13.3. Создание двунаправленного связывания.....	364
13.4. Связывание свойств с помощью ActionScript	365
13.5. Цепочки связанных свойств.....	369
13.6. Связывание свойств объекта с XML-документом с применением E4X.....	371
13.7. Создание нестандартных связываемых свойств	373
13.8. Связывание с обобщенным объектом.....	377
13.9. Связывание со свойствами динамического класса.....	379

Глава 14. Проверка допустимости данных, форматирование и регулярные выражения..... 385

14.1. Применение валидаторов и форматеров в элементах управления.....	386
14.2. Создание нестандартного форматера.....	389
14.3. Применение регулярных выражений для создания валидатора международных почтовых индексов.....	390
14.4. Проверка корректности для комбинированных списков и групп переключателей.....	393
14.5. Показ ошибок проверки допустимости с помощью объектов <i>ToolTip</i> в форме.....	396
14.6. Применение регулярных выражений для поиска адресов электронной почты	399
14.7. Применение регулярных выражений для поиска номеров кредитных карт.....	399
14.8. Применение регулярных выражений для проверки номеров ISBN.....	400
14.9. Создание регулярных выражений с явно указанными классами символов... ..	400
14.10. Указание типов символов регулярных выражений	401
14.11. Поиск корректных IP-адресов с применением подвыражений	403
14.12. Применение регулярных выражений при поиске различных образцов	404
14.13. Поиск начала или конца строки с помощью регулярных выражений.....	406
14.14. Применение обратных ссылок.....	406
14.15. Просмотр вперед и просмотр назад	407

Глава 15. Работа со службами и взаимодействие с сервером..... 409

15.1. Конфигурирование компонента <i>HTTPService</i>	410
15.2. RESTful-взаимодействие между Flex-приложениями.....	412
15.3. Взаимодействие со службой, возвращающей данные в формате JSON.....	414
15.4. Конфигурирование служб для приложения, использующего сервер BlazeDS....	417
15.5. Конфигурирование компонента <i>RemoteObject</i> и связь с ним	421
15.6. Использование модели "публикация/подписка" для приложений-чатов	424
15.7. Применение интерфейса <i>IExternalizable</i> для нестандартной сериализации	429
15.8. Отслеживание результатов нескольких одновременных обращений к службе	431
15.9. Регистрация типа серверных данных во Flex-приложении	432
15.10. Взаимодействие с объектом <i>WebService</i>	434
15.11. Добавление заголовка SOAP в запрос к веб-службе	436

15.12. Разбор SOAP-ответа от веб-службы.....	437
15.13. Безопасное взаимодействие с AMF при помощи объекта <i>SecureAMFChannel</i>	439
15.14. Отправка и получение двоичных данных через сокетное соединение.....	440
15.15. Взаимодействие с сервером при помощи объекта <i>XMLSocket</i>	441
15.16. Навигация по XML-документу с помощью E4X	442
15.17. Использование регулярных выражений в запросах E4X	444
15.18. Добавление объекта <i>XMLElement</i> в объект <i>XML</i>	445
15.19. Обработка пространств имен в XML-документе, возвращенном службой	447
15.20. Преобразование объекта данных ActionScript в XML-документ	448
15.21. Преобразование XML-документа, возвращенного веб-службой, в строго типизированные объекты	449

Глава 16. Взаимодействие с браузером..... 453

16.1. Переход по внешнему URL-адресу	453
16.2. Работа с классом <i>FlashVars</i>	454
16.3. Вызов функций JavaScript из Flex-приложения	456
16.4. Вызов функций ActionScript из кода JavaScript	457
16.5. Изменение заголовка HTML-страницы с помощью объекта <i>BrowserManager</i>	459
16.6. Разбор URL-адреса с помощью объекта <i>BrowserManager</i>	460
16.7. Внешние ссылки на данные с помощью объекта <i>BrowserManager</i>	462
16.8. Внешние ссылки на контейнеры с помощью объекта <i>BrowserManager</i>	464

Глава 17. Модули и совместно используемые библиотеки этапа выполнения..... 467

17.1. Создание RSL-библиотеки	468
17.2. Междоменные совместно используемые библиотеки этапа выполнения.....	471
17.3. Оптимизация совместно используемой библиотеки этапа выполнения	473
17.4. Создание модуля на основе MXML	474
17.5. Создание модуля на языке ActionScript.....	476
17.6. Загрузка модулей с помощью класса <i>ModuleLoader</i>	478
17.7. Загрузка модулей с помощью класса <i>ModuleManager</i>	480
17.8. Загрузка модулей с разных серверов	482
17.9. Взаимодействие с модулем	484
17.10. Отправка данных модулю через строки запросов	488
17.11. Использование отчетов редактора связей для оптимизации модулей.....	490

Глава 18. Основы работы с AIR..... 493

18.1. Создание и запуск AIR-приложения в Flash Builder 4.....	493
18.2. Подписывание и экспорт AIR-приложения.....	496
18.3. Подписывание AIR-файла сертификатом, заслуживающим доверия.....	498

18.4. Разработка под конкретную версию AIR.....	499
18.5. Установка идентификатора приложения.....	500
18.6. Указание имени приложения и имени его файла	501
18.7. Указание версии приложения.....	502
18.8. Редактирование описания приложения и информации об авторских правах.....	503
18.9. Редактирование параметров стартового окна	505
18.10. Указание установочного каталога для приложения	507
18.11. Указание папки для меню <i>Программы</i>	508
18.12. Установка нестандартной пиктограммы для приложения.....	509
18.13. Взаимодействие AIR-приложения с браузером	511
18.14. Настройка приложения на корректное обновление	512
18.15. Выяснение версии приложения на этапе выполнения	513
18.16. Создание многоязычных установок AIR-приложения	514
18.17. Создание приложений со способностью к обновлению	515
18.18. Создание приложений со способностью к обновлению нестандартным интерфейсом.....	520
18.19. Упаковка приложения в стандартный установщик	527
18.20. Включение низкоуровневого кода в AIR-приложение.....	529

Глава 19. Работа с данными в среде AIR 533

19.1. Защита файлов с помощью зашифрованного локального хранилища	533
19.2. Отображение обновлений в сериализованной версии.....	536
19.3. Создание базы данных в оперативной памяти	538
19.4. Защита базы данных с помощью пароля	538
19.5. Использование параметров в запросах	542
19.6. Встраивание базы данных в приложение	547
19.7. Сохранение простых отношений с помощью ORM	548

Глава 20. Интеграция операционной системы со средой AIR..... 553

20.1. Одновременное закрытие всех открытых окон.....	554
20.2. Добавление тени к нестандартному окну	555
20.3. Отложенный вывод на экран данных из буфера обмена	560
20.4. Создание нестандартных форматов данных для буфера обмена	561
20.5. Закрепление "горячих" клавиш за пунктами меню	565
20.6. Уведомление пользователя через панель <i>Dock</i> (в Mac OS) или панель задач (в Windows).....	566
20.7. Регистрация нестандартных типов файлов	569
20.8. Открытие файла в приложении, ассоциированном с ним по умолчанию	572
20.9. Проверка смонтированных и размонтированных устройств.....	576
20.10. Получение списка доступных внешних дисков	577
20.11. Сообщить операционной системе, что файл был загружен из Интернета.....	583

20.12. Распространение AIR-приложения в виде собственного установщика.....	584
20.13. Создание HTTP-прокси с помощью класса <i>ServerSocket</i>	590
Глава 21. Диаграммы.....	595
21.1. Создание диаграммы	595
21.2. Применение эффектов	598
21.3. Выделение областей диаграммы	601
21.4. Форматирование надписей на осях диаграммы	603
21.5. Создание нестандартной метки на диаграмме	605
21.6. Применение эффекта детализации данных для гистограммы.....	606
21.7. Смена скинов у элементов диаграммы	609
21.8. Динамическое добавление и удаление столбиков диаграммы.....	612
21.9. Перекрытие нескольких рядов данных.....	615
21.10. Перетаскивание элементов диаграммы мышью	616
21.11. Создание редактируемой диаграммы.....	619
Глава 22. Тестирование модулей с помощью FlexUnit.....	623
22.1. Создание приложения, работающего на платформе FlexUnit.....	624
22.2. Создание приложения для запуска тестов FlexUnit.....	624
22.3. Создание тестового случая FlexUnit	627
22.4. Выполнение кода до и после каждого теста	630
22.5. Использование одних данных в разных тестовых случаях.....	632
22.6. Обработка событий в тестовом случае	634
22.7. Тестирование визуальных компонентов с помощью FlexUnit.....	636
22.8. Создание макетов для тестирования	645
22.9. Применение сложных утверждений в тестовом случае.....	648
Глава 23. Компиляция, отладка и развертывание.....	651
23.1. Применение операторов <i>trace</i> вне среды Flash Builder	651
23.2. Работа с компилятором компонентов	653
23.3. Установка библиотеки Flex Ant tasks.....	654
23.4. Компиляция Flex-приложений с помощью программ mxmhc и Ant.....	656
23.5. Применение программы Ant для компиляции и развертывания Flex-приложений, использующих RSL-библиотеки.....	658
23.6. Компиляция Flex-приложений с помощью программы Rake.....	661
23.7. Создание и просмотр выражений в отладчике Flash Builder Debugger	662
23.8. Установка Ant в автономной версии Flash Builder	664
23.9. Генерирование документации с помощью ASDoc и Ant	665
23.10. Опция экспресс-установки	666
23.11. Профилирование памяти с помощью Flash Builder.....	668
23.12. Проверка производительности конкретных методов	670

Глава 24. Локализация, доступность и вывод на печать	673
24.1. Добавление международного набора символов.....	673
24.2. Локализация приложения с помощью пакетов ресурсов	676
24.3. Использование класса <i>ResourceManager</i> для локализации.....	680
24.4. Применение модулей ресурсов для локализации	682
24.5. Поддержка редактора метода ввода.....	685
24.6. Распознавание программ озвучивания текста.....	687
24.7. Определение порядка перехода фокуса.....	687
24.8. Печать выделенных элементов приложения	689
24.9. Форматирование содержимого приложения перед печатью	690
24.10. Управление многостраничной печатью содержимого неизвестного размера.....	692
24.11. Добавление колонтитулов в распечатку	693
Предметный указатель	697

Предисловие

Flex 4 — это мощная платформа, предоставляющая пригодные для коммерческого применения компоненты для работы в среде Flash Player, написанные на языке разметки, понятном каждому, кто знаком с языками HTML или XML. Платформа Flex предоставляет компоненты для размещения элементов на экране, воспроизведения визуальных эффектов, организации данных в таблицы, взаимодействия с сервером, вывода диаграмм и т. д.

Короче говоря, Flex является достаточно развитой платформой, и попытка охватить все ее стороны в одной книге, без сомнения, обречена на провал. Понимая это, мы постарались затронуть темы, которые больше всего интересуют разработчиков, имеющих дело с Flex 4. По ходу изложения мы обсуждаем структуру платформы и помогаем разработчикам, знакомым с предыдущими версиями Flex, освоиться с новыми компонентами и функциональными возможностями Flex 4. Официальная документация по Flex достаточно хорошо разъясняет поведение конкретных методов и классов, поэтому мы сосредоточились на том, как решать типичные задачи на платформе Flex, как обеспечить взаимодействие различных компонентов и как объединить Flex с другими технологиями для создания насыщенных интернет-приложений. Например, с помощью интегрированной среды выполнения Adobe AIR вы можете использовать инструментальные возможности Flex и Flash Player для создания развертываемых настольных приложений. Эта среда дополняет многочисленные инструменты (как коммерческие, так и имеющие открытый код), предназначенные для разработки программного обеспечения на языках Java и PHP, а также на платформе .NET. В результате платформа Flex оказывается мощным средством решения программистских задач самого широкого диапазона и самой высокой сложности.

Для кого эта книга

Книга предназначена для разработчиков, стремящихся к более глубокому пониманию платформы Flex Framework, нуждающихся в справочном материале или консультации для решения конкретных задач либо пытающихся разобраться в нововведениях, которые появились в четвертой версии. Поэтому в книге предполагается, что у вас есть некоторый опыт работы с Flex и ActionScript 3. Примеры кода и разъяснения предназначены для разработчиков, уже не являющихся новичками,

понимающих связь между языками MXML, знакомых хотя бы с некоторыми компонентами платформы Flex и с основными приемами разработки Flex-приложений.

Мы сознательно приняли решение предоставлять во всех рецептах готовые к использованию компоненты и работоспособные, протестированные реализации этих компонентов. Это было сделано не для того, чтобы искусственно "раздуть" объем книги, а для того, чтобы она была полезной как для опытных разработчиков, которым достаточно взглянуть на небольшой фрагмент кода, чтобы понять суть применяемого подхода, так и для читателей, которым еще предстоит научиться оптимальным приемам работы с платформой Flex Framework.

Кому не следует читать эту книгу

Если вы хотите изучить Flex Framework с нуля, прочитайте книгу "Programming Flex 3" Джои Лотта (Joey Lott) и Чафика Кейзоуна (Chafic Kazoun)¹ или книгу "Hello! Flex" (Здравствуй, Flex!) Питера Армстронга (Peter Armstrong)², выпущенную издательством Manning. Эти книги дадут вам базовое представление о технике программирования под Flex, после чего вы сможете читать нашу книгу. Имея опыт работы с Flex и программирования на ActionScript, вы лучше поймете технические приемы, изложенные в этой книге. Если вам необходимо освежить в памяти то, что вы знаете об ActionScript, или вы хотите сосредоточиться на программировании под Flash на этом языке, вам поможет книга "ActionScript 3.0: Cookbook" Джои Лотта (Joey Lott), Дэррона Шалла (Darron Schall) и Кейта Питерса (Keith Peters).³ Хотя в нашей книге обсуждаются некоторые области пересечения между Flex Framework и основными классами Flash ActionScript, она посвящена разработке Flex-приложений.

Как организована эта книга

Книга содержит рецепты, предназначенные для того, чтобы обучить вас техническим приемам, позволяющим эффективнее программировать Flex-приложения. Для облегчения поиска нужного рецепта мы организовали их по темам. Вообще говоря, внутри каждой главы рецепты расположены в порядке возрастания сложности.

Эта книга не предназначена для чтения от начала и до конца. Ее следует использовать как справочник по решению конкретных задач, позволяющий лучше понять конкретные особенности платформы Flex Framework. В рецепты входят законченные реализации компонентов, демонстрирующие, как применять на практике обсуждаемые теоретические понятия. Вы сможете использовать приведенный код в своих приложениях или, по крайней мере, адаптировать его фрагменты под свои задачи.

¹ Кейзоун Ч., Лотт Дж. Программирование с использованием Adobe Flex. — СПб.: Питер, 2009.

² См. <http://www.manning.com/armstrong3/>.

³ Лотт Дж., Шалл Д., Питерс К. ActionScript 3.0. Сборник рецептов. — СПб.: Символ-плюс, 2007.

Обозначения, принятые в книге

Мы придерживаемся следующих обозначений.

- *Курсив* используется для выделения новых терминов и для усиления высказывания.
- Моноширинным шрифтом напечатаны листинги программ и программные элементы в тексте: имена переменных и функций, теги и компоненты, типы данных, переменные окружения, операторы и ключевые слова.
- **Моноширинным полужирным шрифтом** напечатаны команды и текст, которые должны быть введены пользователем именно в таком виде. Шрифт также используется для выделения участков кода в листингах.
- *Моноширинным курсивным шрифтом* напечатан текст, который должен быть заменен пользователем на конкретные значения или значения, определяемые контекстом.

ПРИМЕЧАНИЕ

Так напечатаны советы и примечания общего характера.

ПРЕДУПРЕЖДЕНИЕ

Так напечатано предупреждение или предостережение.

Использование кода

Эта книга написана специально для того, чтобы вы лучше справлялись со своей работой. Вообще говоря, вы можете пользоваться кодом из этой книги в своих программах и документации. Если вы не воспроизводите наш код в значительном объеме, то не обязаны связываться с нами для получения разрешения. Например, при написании программы, в которой использовано несколько фрагментов кода из этой книги, разрешение не требуется. Однако для продажи или распространения дисков с примерами кода из книг издательства O'Reilly разрешение необходимо. Если, отвечая на чей-то вопрос, вы процитируете эту книгу и приведете пример кода из нее, вам не нужно получать разрешение. Если же вы захотите включить большой объем кода из этой книги в документацию к своему продукту, то должны будете получить разрешение на это.

Мы будем благодарны за указание источника кода, но не настаиваем на этом. Сведения об источнике, как правило, включают в себя название книги, автора, название издательства и стандартный международный номер книги (ISBN). Например, "Flex 4 Cookbook by Joshua Noble, Todd Anderson, Garth Braithwaite, Marco Casario, and Rich Tretola. Copyright 2010 O'Reilly Media, Inc., 978-0-596-80561-6". Если вам кажется, что при использовании нашего кода вы вышли за рамки, обрисованные выше, не стесняйтесь написать нам по адресу permissions@oreilly.com.

Как использовать эту книгу

Относитесь к этой книге как к другу и советчику. Не ставьте ее на полку, а держите на рабочем столе, чтобы проще было получить справку. Когда у вас возникнут вопросы по тому, как работают компоненты или как решить конкретную программистскую задачу, откройте книгу и найдите в ней нужный рецепт (или несколько рецептов). Книга написана так, что вы сможете быстро найти ответы на свои вопросы. А поскольку это книга, вы можете не бояться, что она поднимет вас на смех за наивный вопрос. Не бывает вопросов незначительных, как не бывает вопросов, на которые нельзя ответить.

Хотя вы можете прочитать эту книгу от начала до конца, мы советуем обращаться к ней, только когда вам нужен ответ на вопрос. Цель этой книги состоит не в изложении теории, а в том, чтобы помочь вам решать задачи, возникающие при написании программ. Эта книга предназначена для использования в условиях практической работы, а не в исследовательской лаборатории.

Сборники рецептов от издательства O'Reilly

Вы ищете решение проблемы, возникшей при написании программы? Вам не требуется ничего, кроме сборника рецептов от издательства O'Reilly. Каждый сборник содержит сотни программных рецептов и сотни сценариев, программ и команд, которыми можно воспользоваться при решении конкретных задач.

Рецепты в сборниках от издательства O'Reilly имеют простую структуру:

Задача. Каждая задача в сборнике рецептов от издательства O'Reilly четко сформулирована, конкретна и связана с практикой.

Решение. Решение, приведенное в сборнике, легко понять и реализовать.

Обсуждение. Обсуждение рецепта приводится для пояснения контекста задачи и решения. Оно также содержит примерный код, демонстрирующий решение задачи. Примечательно, что код любого примера, приведенного в сборнике рецептов от издательства O'Reilly, может быть загружен с веб-сайта книги, например, <http://www.oreilly.com/catalog/9780596805616>.¹

См. также. В таких абзацах представлены ссылки на другие источники, в которых вы найдете информацию, связанную с темой рецепта.

Подробную информацию о серии сборников рецептов от издательства O'Reilly и о конкретных сборниках на интересующие вас темы вы найдете на сайте <http://cookbooks.oreilly.com>.

Цифровые библиотеки

Safari Books Online — это цифровая библиотека, дающая вам более чем 7500 книг и видеороликов, содержащих справочные материалы по информационным технологиям. В ней вы сможете быстро найти ответы на свои вопросы.

¹ На указанной странице следует перейти по ссылке **Download Example Code**. Или см. <http://examples.oreilly.com/9780596805623/>.

Имея подписку, вы сможете прочитать любую страницу и просмотреть любой ролик из нашей библиотеки. Читайте наши книги с помощью мобильного телефона или портативного компьютера. Читайте новые книги до того, как они будут напечатаны в типографии, получите эксклюзивный доступ к еще не законченным рукописям и отправляйте свои комментарии авторам. Копируйте примеры кода в свои приложения, организуйте избранные страницы, загружайте целые главы и ставьте закладки на важных разделах, пишите заметки, распечатывайте страницы и пользуйтесь массой других функциональных возможностей, позволяющих экономить время.

Служба O'Reilly Media выложила эту книгу в электронную библиотеку Safari Books Online. Чтобы иметь неограниченный доступ к цифровой версии этой книги и других книг по той же тематике, выпущенных издательством O'Reilly и другими издательствами, пройдите процедуру бесплатной регистрации на сайте <http://my.safaribooksonline.com>.

Как с нами связаться

Для этой книги создана специальная веб-страница со списком замеченных опечаток, кодом примеров и дополнительной информацией:

<http://www.oreilly.com/catalog/9780596805616>

Комментарии и технические вопросы можно отправлять по электронной почте на адрес: bookquestions@oreilly.com.

Более подробную информацию о наших книгах, конференциях, центрах ресурсов (Resource Centers) и службе O'Reilly Network можно найти на сайте издательства: <http://www.oreilly.com>.

Благодарности

Эта книга в значительной степени является продуктом всего Flex-сообщества. Мы хотим поблагодарить многих разработчиков и менеджеров из компании Adobe, в частности, Мэтта Чотина (Matt Chotin), Илая Гринфилда (Ely Greenfield) и Алекса Харуи (Alex Harui), а также программистов, работающих с продуктами Adobe, которые сделали ценные замечания на сайте этой книги или в своих блогах. Без них написание этой книги было бы невозможно.

Мы приносим огромную благодарность коллективу издательства O'Reilly. Большую работу проделали Стив Вейс (Steve Weiss), Мэри Треслер (Mary Treseler), Линда Лафлам (Linda LaFlamme) и Мишель Филши (Michele Filshie). Благодарим их за гибкий подход и терпение, проявленные при подготовке рукописи и ее редактировании.

Качество технической информации в этой книге обеспечено не только знаниями ее авторов. Технические редакторы, — Джоди О'Рурк (Jodie O'Rourke), Эд Мансури (Ed Mansouri), Кевин Саттл (Kevin Suttle), Микола Билоконски (Mikola Bilokon-sky), Чак Фридман (Chuck Freedman), Расс Фергюсон (Russ Ferguson) и Шон Мур (Sean Moore), — не только оказали помощь в отладке, исправлении и улучшении

кода примеров в этой книге, но и дали нам множество советов о том, как сделать текст более ясным, как структурировать главы и как отредактировать рецепты, чтобы читатель лучше понял их.

От Джошуа

В первую очередь я хочу поблагодарить Джои Лотта (Joey Lott) и Стива Вейса (Steve Weiss) за то, что они любезно предоставили мне возможность написать мою первую книгу много лет назад и за другие замечательные возможности, которые они предоставляли мне с тех пор. Я благодарен своим соавторам, Тодду Андерсону (Todd Anderson), Гарту Брэйтуэйту (Garth Braithwaite), а также авторам книги "Adobe AIR 1.5 Cookbook" (Adobe AIR 1.5. Сборник рецептов) Ричу Третоле (Rich Tretola) и Марко Казарио (Marco Casario).¹ Без вас эта книга не появилась бы на свет! То же самое я хочу сказать тем, кто участвовал в обсуждениях на сайте Adobe Cookbook и на таких форумах, как FlexCoders, образуя живое дружественное сообщество, оказавшее всем нам большую помощь.

Я хочу поблагодарить также моих друзей по работе и друзей по жизни за их помощь, советы и добрый юмор. И еще я благодарю моего брата, отца и, в особенности, мать за веру в мои силы, поддержку, мудрость и юмор.

От Тодда

В первую очередь я хочу поблагодарить Джошуа Ноубла (Joshua Noble) за то, что предложил участвовать в написании этой книги, за его знания, терпение и неистощимый юмор. Я благодарю также Джои Лотта (Joey Lott) за его глубокую веру в людей и их способности. Я благодарю своих друзей и все Flash-сообщество за компетентные советы и даже за насмешки. Наконец, я благодарю свою семью. Никаких слов не хватит, чтобы поблагодарить вас за безграничную любовь и поддержку.

От Рича

Я хочу поблагодарить жену и лучшего друга Ким, которая всегда находится рядом и поддерживает меня во всех моих многочисленных начинаниях. Я хочу поблагодарить также своих дочерей, Скай, Корал и Тринити. Я всех вас очень люблю! И еще благодарю всех соавторов за большую работу.

От Гарта

Я не стал бы участником этого проекта, если бы не Стив Вейс (Steve Weiss), Джошуа Ноубл (Josh Noble) и Тодд Андерсон (Todd Anderson), и я благодарю их за предоставленную возможность. Кроме того, я благодарю отца за то, что привил мне любовь к программированию, и доктора наук Пола Меррилла (Paul Merrill) за то, что многому научил меня, когда я еще осваивал эту профессию. Я благодарю коллег

¹ См. <http://oreilly.com/catalog/9780596522513/>.

по RIA Radio — Лейфа Уэллса (Leif Wells), Заха Штепека (Zach Stepek) и Стейси Малкахи (Stacey Mulcahy). Я должен поблагодарить все сообщество Adobe и в особенности Рейчел Люксембург (Rachel Luxemburg), Эдварда Салливана (Edward Sullivan) и Грега Хеймера (Greg Hamer) за веру в мои силы. И, наконец, я ничего не добился бы без поддержки жены, дочерей, мамы и всей семьи.

От Марко

Приношу особую благодарность моим замечательным соавторам за качество и количество труда, вложенного ими в эту книгу. Я хочу поблагодарить коллег по Comtaste, без упорного труда которых над нашими внутренними проектами я бы не смог достичь того, что мне удалось. Выражаю искреннюю и глубокую благодарность коллективу издательства O'Reilly за терпение, постоянную помощь и профессионализм на всем протяжении проекта.

Я приветствую диалог и жду комментариев к этой книге. Пишите мне по адресу m.casario@comtaste.com или оставляйте комментарии в моих блогах <http://blog.comtaste.com> или <http://casario.blogs.com>.

Об авторах

Джошуа Ноубл (Joshua Noble) — консультант, автор и преподаватель. Живет в Нью-Йорке и в Портленде, штат Орегон. Является соавтором книг "Programming Interactivity" (Программирование интерактивности)¹, "Flex 3 Cookbook"², обе издательства O'Reilly, и "ActionScript 3.0 Bible" (Библия по ActionScript 3.0)³, издательства Wiley. Последние шесть лет он работал с Flex и Flash, создав большое количество приложений на самых разных платформах. Кроме того, имеет богатый опыт работы на PHP, Ruby и Erlang. Разрабатывал архитектурные инсталляции и интерактивные среды с использованием таких технологий, как Processing, C++, OpenCV, а также микроконтроллеров. Имеет сайт <http://thefactoryfactory.com>.

Тодд Андерсон (Todd Anderson) — старший программист в компании Infrared5. Имеет восьмилетний опыт разработок для платформы Flash в таких областях, как насыщенные интернет-приложения и компьютерные игры. Тодд создавал настольные, мобильные и веб-приложения для издательского дела и индустрии развлечений, сотрудничая с такими компаниями, как Conde Nast Publications, Adobe, THQ и Motorola. Является соавтором книг "Adobe AIR: Create — Modify — Reuse" (AIR — создание, модификация, многократное использование)⁴ издательства Wiley Wrox и "Flex 3 Cookbook" издательства O'Reilly. В настоящее время живет около Бостона. Когда не сидит за компьютером, проводит время с женой, читает вслух сыну, бегаёт наперегонки с собакой или пытается вспомнить свои прежние занятия изобразительным искусством. Андерсон ведет блог <http://www.custardbelly.com/blog>, посвященный разработке настольных, мобильных и веб-приложений.

Гарт Брэйтуэйт (Garth Braithwaite) — старший программист в компании Rain в штате Юта. Он специализируется в разработке интерактивных модулей с использованием HTML/CSS/JS и платформы Flash. Владеет RIA Radio, InsideRIA и еженедельным подкастом для O'Reilly, регулярно выступает на конференциях 360|Flex, MAX и Flash and the City, а также преподаёт. Гарт имеет звания ACP (Adobe Community Professional, профессионал в сообществе Adobe), ACI (Adobe Certified Instructor, сертифицированный преподаватель Adobe), ACE (Adobe Certified Expert, сертифицированный эксперт Adobe) и Eagle Scout (Скаут-орел).

¹ См. <http://oreilly.com/catalog/9780596154158/>.

² Ноубл Дж., Андерсон Т. Flex 3. Сборник рецептов. — СПб.: Символ-Плюс, 2009.

³ См. <http://eu.wiley.com/WileyCDA/WileyTitle/productCd-0470135603.html>.

⁴ См. <http://eu.wiley.com/WileyCDA/WileyTitle/productCd-0470182075.html>.

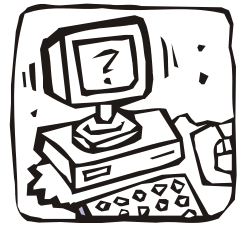
Марко Казарио (Marco Casario) — основатель компании Comtaste (<http://www.comtaste.com>), занимающейся исследованием последних достижений в области создания насыщенных интернет-приложений и конвергенции между Всемирной паутиной и миром мобильных устройств. Он является автором книг "Flex Solutions: Essential Techniques for Flex 2 and Flex 3 Developers" (Решения Flex: технические приемы для разработчиков под Flex 2 и Flex 3)¹ и "Advanced AIR Applications" (Сложные AIR-приложения) вышедших в издательстве Friends of ED. Марко часто выступает на таких конференциях, как Adobe MAX, The O'Reilly Web 2.0 Summit, FITC, The AJAXWorld Conference & Expo, 360Flex, From A to Web, AdobeLive и многих других. Подробности можно найти в его блоге (<http://casario.blogs.com>).

Рич Третола (Rich Tretola) в настоящее время занимает пост менеджера по разработке приложений в компании Herff Jones, Inc. Он имеет знаки отличия как Flex-разработчик, а также является ведущим автором книги "Professional Flex 2" (Профессиональная работа в Flex 2)² и единственным автором книги "Beginning Adobe AIR: Building Applications for the Adobe Integrated Runtime" (AIR для начинающих. Разработка приложений для Adobe Integrated Runtime), вышедших в издательстве Wiley Wrox. Кроме того, он написал некоторые разделы для книги "Adobe AIR 1.5 Cookbook" (Сборник рецептов Adobe AIR 1.5)³ издательства O'Reilly. Рич разрабатывает интернет-приложения свыше 10 лет и работает с Flex с 2003 года, когда появилась бета-версия Royale. Помимо Flex он пишет приложения на базе таких технологий, как ColdFusion, Flash и Java. Рич пользуется уважением во Flex-сообществе как эксперт в области создания насыщенных интернет-приложений и носит почетное звание ACP (Adobe Community Professional, профессионал в сообществе Adobe). Он ведет популярный блог, посвященный Flex и AIR (<http://blog.EverythingFlex.com>), а также на сайте <http://InsideRIA.com>. Рич делал доклады по просьбе организаторов конференции Adobe MAX три года подряд. Он занимается не только программными технологиями, но и является совладельцем шоколадной фабрики на острове Мауи, носящей название "WowWee Maui".

¹ См. <http://www.friendsofed.com/book.html?isbn=1590598768>.

² См. <http://eu.wiley.com/WileyCDA/WileyTitle/productCd-0470102675.html>.

³ См. <http://eu.wiley.com/WileyCDA/WileyTitle/productCd-0470229047.html>.



Глава 1

Основы работы с Flex и программирования на ActionScript

Всякое приложение под Flex состоит, в основном, из кода, написанного на двух очень разных языках — ActionScript и MXML. Язык *ActionScript* к своей третьей версии прошел путь от языка создания сценариев, основанного на прототипах, до полноценного объектно-ориентированного языка со строгой типизацией. Что касается MXML, это язык разметки, понятный каждому, кто работал с HTML, XML или с любым из множества более новых языков разметки.

Новички, лишь недавно приступившие к работе с Flex, обычно интересуются, как MXML и ActionScript связаны друг с другом. Компилятор MXML (mxmclc) транслирует идиомы языка в объекты, так что следующие два фрагмента:

```
<s:Button id="btn" label="My Button" height="100"/>
```

и

```
var btn:Button = new Button();  
btn.label = "My Button";  
btn.height = 100;
```

создают один и тот же объект.

Основная разница между двумя языками заключается в том, что на языке ActionScript (второй пример) просто создается кнопка, а при создании объекта на MXML кнопка добавляется к компоненту, содержащему этот код. Платформа Flex Framework обрабатывает код, вызывая конструктор объекта, описанного на MXML, и либо добавляет объект к его родителю, либо задает его в качестве свойства объекта-родителя.

Файлы с MXML-кодом могут содержать код ActionScript внутри тегов `<fx:Script>`, но файлы с кодом на языке ActionScript не могут содержать код MXML. Хотя у программиста может возникнуть искушение считать MXML языком описания внешнего вида приложения и образующих его компонентов, а язык ActionScript — языком описания обработчиков событий и логики конкретного приложения, это не всегда правильно. Гораздо более точным будет представление, согласно которому оба языка, в конечном счете, позволяют описывать одни и те же объекты, но имеют разный синтаксис. Некоторые аспекты платформы Flash недостижимы без применения таких конструкций языка ActionScript, как циклы, объявления функций, условные операторы и др. Поэтому использование ActionScript и интеграция языков ActionScript и MXML совершенно необходимы при создании приложений, за исключением самых простых.

В этой главе рассматриваются различные вопросы интеграции MXML и ActionScript, в частности создание компонентов на MXML, создание классов с помощью ActionScript, добавление слушателей событий в код приложения, создание файлов фонового кода с помощью ActionScript и MXML, а также объявление функций. Хотя глава и не дает ответы на все мыслимые вопросы, она может послужить вам хорошей стартовой площадкой, т. к. содержит изложение основ ActionScript и MXML.

1.1. Создание Flex-проекта в среде Flash Builder

Задача

Нужно создать проект в среде Flash Builder.

Решение

Воспользуйтесь мастером **Create New Project** (Создать новый проект).

Обсуждение

Среда *Flash Builder* является надстройкой над Eclipse, давно существующей и снискавшей всеобщее одобрение интегрированной средой разработки, которая, в первую очередь, ассоциируется с программированием на Java. Хотя среда Flash Builder, строго говоря, не является необходимой для создания Flex-приложений, она была первым инструментом, разработанным для этой цели, и, как таковая, обладает большим количеством функциональных возможностей, позволяющих эффективно проектировать и разрабатывать приложения. Вы можете использовать среду Flash Builder либо как отдельное приложение, либо как модуль, добавляемый в заранее установленную среду разработки Eclipse.

Первое, что вы должны сделать, — это создать новый Flex-проект. Проект этого типа отличается от других, создаваемых в Flash Builder, поскольку он включает в себя библиотеку SWC (отсутствующую у проектов ActionScript) и компилируется в SWF-файл, который можно открыть с помощью приложения Flash Player (чего нельзя сделать с проектом Flex Library). Чтобы создать проект, выполните двойной щелчок правой кнопкой мыши (на компьютерах с Mac OS — щелчок с нажатой клавишей <Ctrl>) в навигаторе проектов Flash Builder (рис. 1.1), чтобы открылось контекстное меню. В качестве альтернативы можно воспользоваться меню **File** (Файл) в верхней части окна приложения. Далее в любом случае выберите **New | Flex Project** (Создать | Проект Flex). Появится диалоговое окно, и вы получите возможность пройти через все этапы создания проекта.

Когда появится запрос указать, каким образом проект будет получать данные, выберите опцию **Basic** (Основной), в результате чего откроется диалоговое окно **New Flex Project** (Новый проект Flex) (рис. 1.2).

Введите имя приложения и, чуть ниже, каталог для файлов приложения в вашей системе. По умолчанию файлы располагаются в папке `C:\Documents and Settings\Username\Documents\workspace\Projectname` в Windows и `Users/Username/Documents/workspace/Projectname` в Mac OS. Конечно, вы можете снять флажок **Use default location** (Использовать каталог по умолчанию) и хранить файлы в любом удобном для вас каталоге. Имя проекта должно быть уникальным.

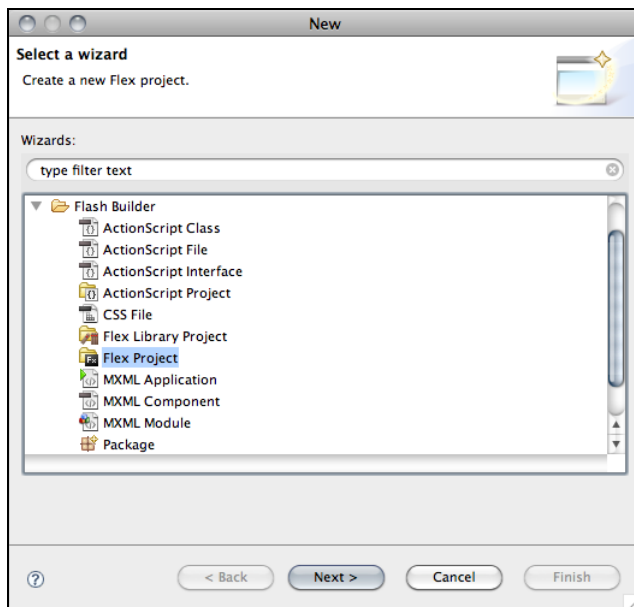


Рис. 1.1. Создание Flex-проекта

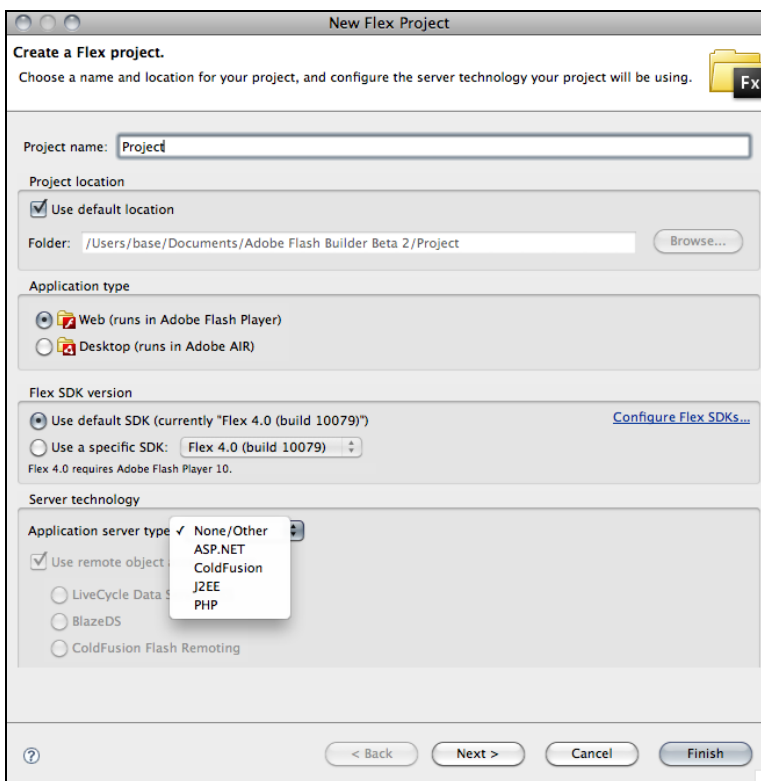


Рис. 1.2. Создание нового проекта в среде Flash Builder

Раздел **Application type** (Тип приложения) позволяет выбрать между приложением, выполняемым в среде AIR (Adobe Integrated Runtime, интегрированная среда выполнения от Adobe), и приложением, которое будет работать в браузере, точнее, в добавляемом модуле Flash Player. Настройки в разделе **Server technology** (Серверная технология) позволяют указать, будет ли приложение соединяться с сервером и, если будет, какой тип сервера и какая конфигурация для этого потребуются.

Если вы больше ничего не хотите добавить, нажмите кнопку **Finish** (Готово). Чтобы изменить каталог, в который будет помещен откомпилированный SWF-файл, нажмите кнопку **Next** (Далее), и тогда откроется окно, изображенное на рис. 1.3.

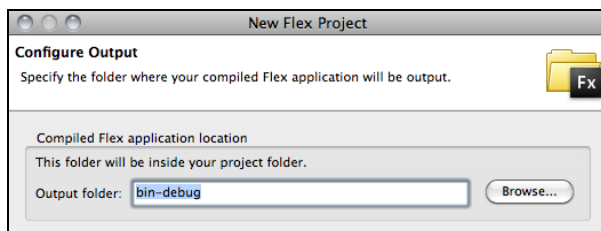


Рис. 1.3. Указание каталога для откомпилированного SWF-файла

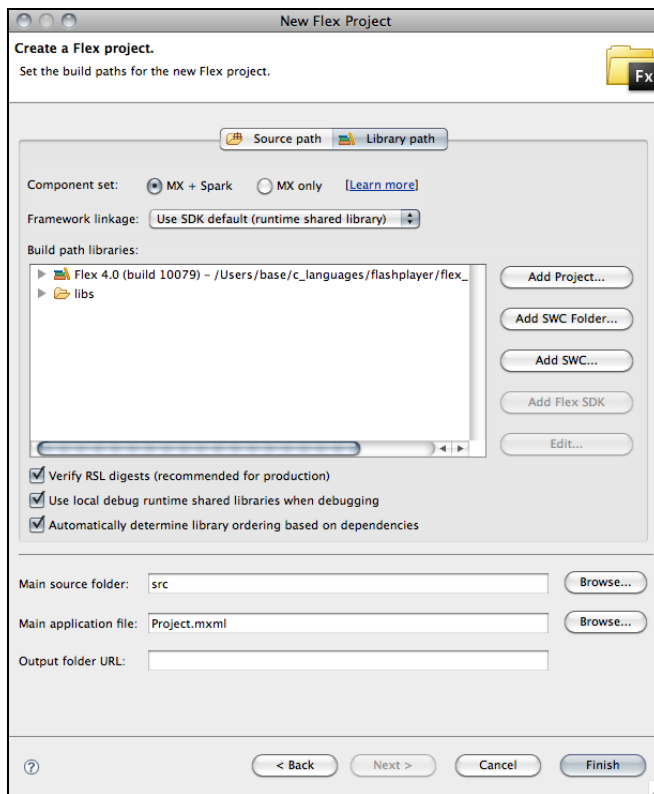


Рис. 1.4. Указание папки для исходного кода и имени главного файла приложения

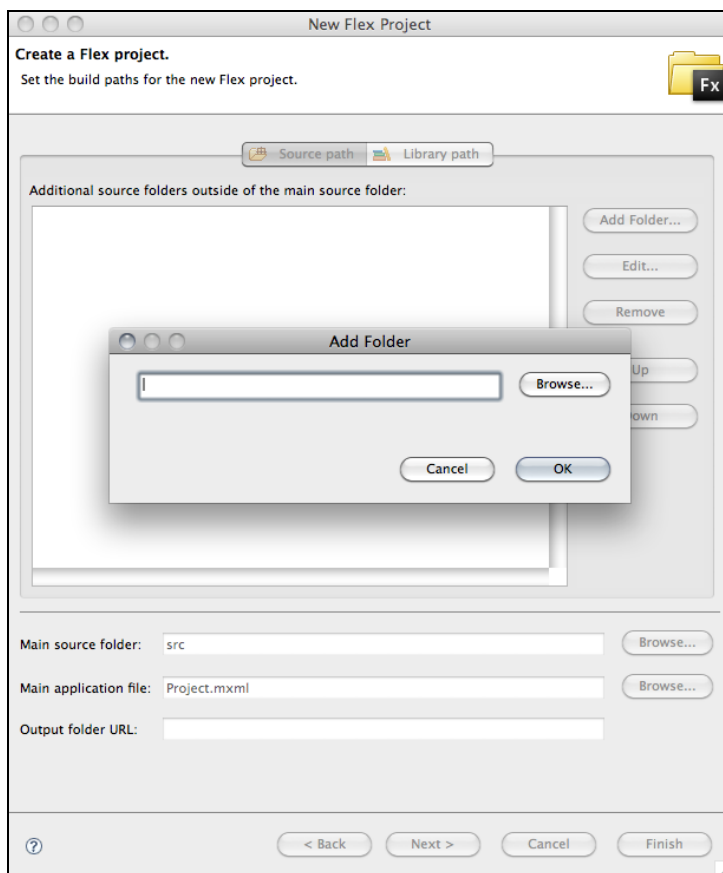


Рис. 1.5. Указание дополнительных путей к файлам Flex-проекта

Указав местоположение для сгенерированного SWF-файла, вы можете либо щелкнуть по кнопке **Finish** (Готово), либо добавить в проект папки для исходного кода или SWC-файлов. Чтобы добавить одну или несколько папок, щелкните по вкладке **Source path** (Путь к исходным файлам) (рис. 1.4). Чтобы добавить SWC-файлы, щелкните по вкладке **Library path** (Путь к библиотеке) (рис. 1.5). В этом окне вы также сможете указать другой главный MXML-файл приложения, который по умолчанию носит то же имя, что и проект.

Разобравшись с путями и именами файлов, нажмите кнопку **Finish** (Готово). Конфигурирование проекта завершено, и вы можете приступить к разработке.

1.2. Создание проекта Flex Library в среде Flash Builder

Задача

Нужно создать проект типа Flex Library.

Решение

В окне Flex Navigator выберите опцию **Flex Library Project** (Создать проект Flex Library). Появится окно мастера **Create New Project** (Создать новый проект).

Обсуждение

У проекта Flex Library нет главного MXML-файла, который компилировался бы в SWF-файл. Вместо этого файлы проекта компилируются в SWC-файл, который может быть использован в других приложениях. Он также может служить исходным материалом для *общей библиотеки этапа выполнения*. Классы из этой библиотеки применяются для создания групп ресурсов, которые можно многократно использовать в разных проектах на этапе компиляции или на стадии выполнения. Чтобы создать проект Flex Library, выполните двойной щелчок правой кнопкой мыши (на компьютерах с Mac OS — щелчок с нажатой клавишей <Ctrl>) в навигаторе проектов Flash Builder (рис. 1.6), чтобы открылось контекстное меню. В качестве альтернативы можно воспользоваться меню **File** (Файл). Далее в любом случае выберите **New | Flex Library Project** (Создать | Проект Flex Library).

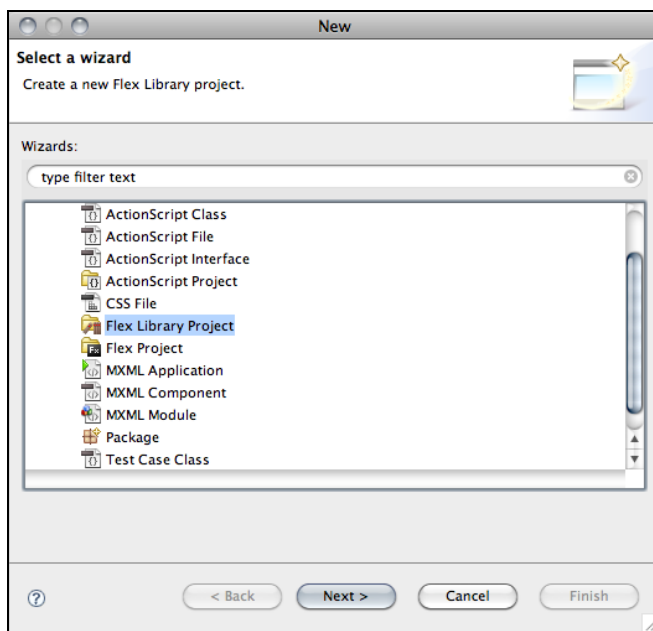


Рис. 1.6. Создание проекта Flex Library

В открывшемся диалоговом окне (рис. 1.7) укажите имя проекта и его местоположение.

Если вы больше ничего не хотите добавить, нажмите кнопку **Finish** (Готово). Если вам нужно включить в проект файлы, ресурсы или дополнительные SWC-файлы (в том числе, библиотеки AIR), нажмите кнопку **Next** (Далее) и выберите, что требуется, в появившемся диалоговом окне. Для определения путей к ресурсам или классам, кото-

рые должны быть добавлены в создаваемую библиотеку, найдите сначала их в своей системе, а затем укажите, какие классы или графические ресурсы нужно откомпилировать в библиотеку. Чтобы создать проект, нажмите кнопку **Finish** (Готово).

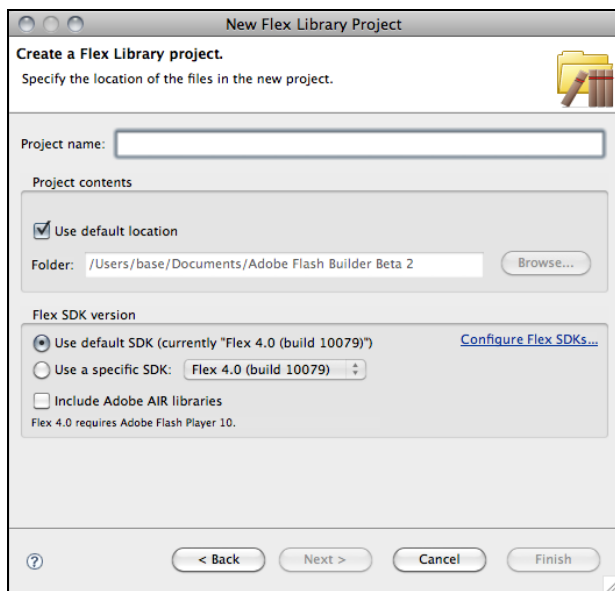


Рис. 1.7. Указание местоположения проекта и выбор комплекта SDK для компилятора

1.3. Настройка опций компилятора в среде Flash Builder

Задача

Нужно установить конкретные опции компилятора MXML.

Решение

Аргументы для компилятора настраиваются в разделе **Flex Compiler** (Компилятор Flex) диалогового окна **Properties for** (Свойства проекта).

Обсуждение

Компилятор MXML (программа mxmcl) транслирует файлы ActionScript и MXML-файлы в SWF-файл, который может быть воспроизведен в плеере Flash Player. Когда вы выполняете или отлаживаете Flex-приложение в среде Flash Builder, вызывается компилятор MXML, и файлы передаются компилятору в виде аргумента приложения. Когда вы отлаживаете плеер, аргумент на создание SWF-файла передается MXML-компилятору. Среда Flash Builder позволяет программисту передавать компилятору и другие аргументы. Например, вы можете передать аргументы, определяющие местоположение внешней библиотеки, позволяющие SWF-файлу обращаться к локальным файлам или устанавливающие цвет фона.

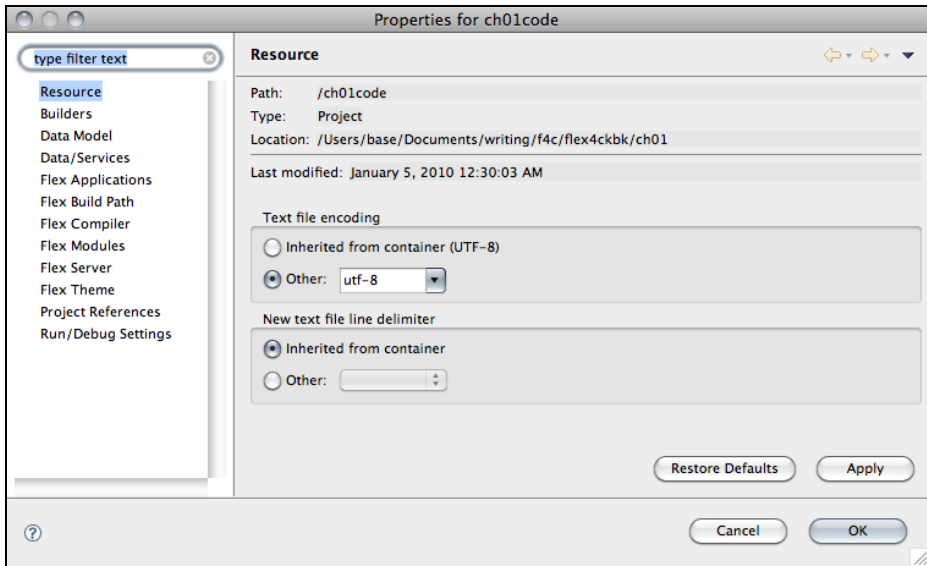


Рис. 1.8. Изменение свойств проекта

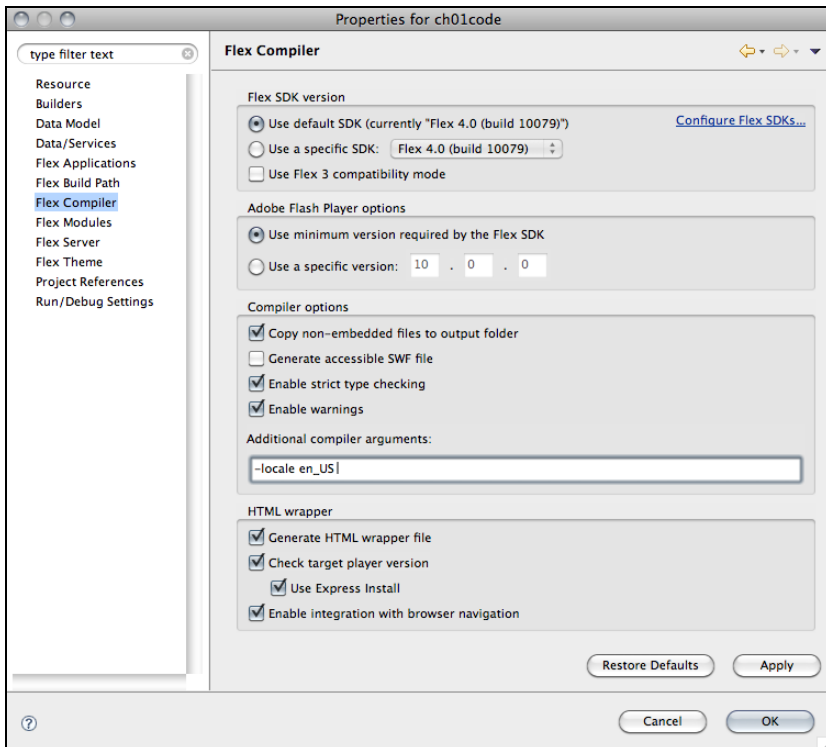


Рис. 1.9. Настройка опций компилятора