

Введение в Web-продукцию Flash

Если вы новичок во Flash и вообще в создании мультимедийной продукции, вам предоставляется прекрасная возможность это исправить. Для опытных пользователей Flash в этом разделе может быть интересным описание новых возможностей и технологий программы.

В главе 1 пользователю предлагается полный обзор преимуществ и недостатков формата Flash, а также информация о происхождении и развитии этой программы. В главе 2 исследуются различные пути взаимодействия фильмов Flash с другими Web-форматами и рассматриваются проблемы, которые приходится решать при планировании проектов для конкретных аудиторий. В главе 3 описываются инструментальные средства и методы планирования мультимедийного проекта, включая процесс создания блок-схем, карт узла и документов функциональных спецификаций.

ЧАСТЬ



В этой части...

Быстрое начало

Глава 1

Знакомство с Flash MX

Глава 2

Web-технологии

Глава 3

Планирование Flash-проектов

В этой главе...

Обзор возможностей

Примеры для дизайнеров

Примеры для разработчиков

Резюме

Быстрое начало

Выполняя этот экскурс по Flash MX, мы обратим внимание на несколько новых возможностей, реализованных в последнем выпуске от Macromedia, просто чтобы возбудить ваш аппетит.

Проработав множество простых примеров, вы получите возможность самостоятельно ознакомиться с Flash MX. Эти маленькие проекты предназначены для быстрого ознакомления с новым графическим интерфейсом пользователя (GUI — Graphic User Interface) и другими особенностями программы.

Несмотря на то что мы будем отдельно рассматривать новые возможности для дизайнеров и разработчиков, рекомендуем выполнять упражнения в обоих разделах для получения полного представления о Flash MX. Не переживайте: все примеры описываются с помощью пошаговых инструкций и являются максимально простыми для самостоятельного изучения.

Обзор возможностей

Ниже приводится краткое введение в новые возможности Flash MX, описываемые в данной главе.

- **Шаблоны.** Во Flash MX содержится набор стандартных шаблонов различных категорий. Можете использовать шаблон, чтобы ускорить создание следующего проекта.
- **Компоненты.** Новыми во Flash MX являются компоненты, которые вытеснили такое средство, как Smart Clip. Компоненты заменили Smart Clip, поскольку они являются гораздо более мощным и гибким средством. Компоненты представляют собой многократно используемые элементы интерфейса пользователя, такие как полосы прокрутки, кнопки, переключатели, флажки и многие другие, поставляемые в комплекте с Flash MX.
- **Папки слоев.** Несколько усовершенствований временной шкалы Flash значительно расширяют возможности управления документом. Любое количество слоев можно поместить в отдельные папки, что дает возможность быстро заблокировать или скрыть нужную группу слоев в любой момент.

- **Библиотека.** Представьте себе возможность создания символа с помощью простого перетаскивания векторной фигуры с рабочего поля в окно библиотеки. А теперь прекратите мечтать и выполните это! Подобная возможность является одним из многих усовершенствований окна Library (Библиотека).
- **Инспектор свойств.** Повышение практичности и удобства использования Flash MX выразилось в создании инспектора свойств. Инспектор во многом подобен полюбившемуся пользователям инспектору свойств в Macromedia Dreamweaver MX.
- **Модификатор Envelope (Огибающая).** Это небольшое дополнение позволяет произвольным образом искажать на рабочем поле любую фигуру. Хотя данный модификатор ограничен применением только к векторным фигурам, его можно использовать для создания довольно интересных эффектов.
- **Команда Break Apart (Разделить).** Теперь можно разделить текстовое поле на отдельные текстовые поля, содержащие по одной букве. Это упрощает создание простой анимации текста на основе отдельных букв. Повторное применение данной команды к уже разделенному на буквы тексту превратит эти буквы в векторные фигуры (как это происходило при однократном применении к тексту команды Break Apart в предыдущей версии Flash).
- **Команда Distribute to Layers (Распределить по слоям).** Эта возможность очень полезна в сочетании с командой Break Apart. После разделения текстового поля каждую букву можно поместить на отдельный слой. Можно также использовать данную команду для любого слоя, содержащего несколько элементов, — после выбора этой команды элементы будут помещены на отдельные слои.
- **Анимированные маски.** Использование масок представляет собой прекрасную технологию, значительно улучшенную во Flash MX. Теперь анимированную маску создавать так же просто, как применить операцию автоматического заполнения промежуточных кадров с интерполяцией движения или изменения формы, при этом для управления анимацией можно использовать сценарий ActionScript.
- **Настраиваемая панель Actions (Действия).** Панель Actions, используемая для создания кода ActionScript, также подверглась капитальным изменениям в версии Flash MX. Теперь вам предоставляются опции для настройки этой панели по своему усмотрению.
- **Именованные анкеры.** Большим улучшением удобства использования фильмов Flash MX в браузерах является возможность распознавания установленных в этих фильмах маркеров (анкеров), что позволяет использовать кнопки Back (Назад) и Forward (Вперед) для перехода к различным частям фильма Flash.
- **Динамическая загрузка растровых рисунков и звука.** Можно динамически загружать звук и растровые рисунки в фильм Flash во время его выполнения. Например, можно загрузить JPEG-изображение прямо в экземпляр видеоклипа в фильме Flash (.swf-файле) во время его воспроизведения в Web-браузере. Это еще одно огромное усовершенствование, расширяющее динамические возможности Flash MX, для которого были позаимствованы некоторые полезные возможности из Macromedia Generator (эта программа больше не требуется для производственных задач Flash MX).
- **Потоковое видео.** Во Flash MX существует возможность загрузки и просмотра видеоматериалов в виде потока с использованием видекодека Sorenson Spark, встроенного в проигрыватель Flash Player 6.
- **Программируемые текстовые поля.** Одним из основных усовершенствований в объектной модели языка ActionScript является добавление объектов Button и TextField. Далее в этой главе о текстовых полях рассказывается более подробно.

- **Улучшенная модель событий ActionScript.** Теперь можно напрямую устанавливать события для объекта с помощью только ActionScript. События можно применять и удалять по необходимости.
- **Рисование с помощью API (программного интерфейса приложения).** Возможности ActionScript были расширены посредством введения новых методов и свойств объекта MovieClip, которые добавили возможность программного рисования векторных фигур.

Примеры для дизайнеров

Углубляясь в данный раздел, и дизайнеры, и разработчики смогут открыть для себя множество способов усовершенствования макета и управления своими документами Flash. Работники Macromedia серьезно поработали над улучшением практичности и удобства использования пользовательского интерфейса Flash MX, что и рассматривается в данном разделе.

Использование шаблонов

Теперь можно быстро и легко создать новый проект, основываясь на одном из шаблонов. С программой Flash MX поставляется большое количество шаблонов, относящихся к широкому диапазону категорий.

Рассмотрим шаблон из категории PhotoSlideshow (Слайд-шоу фотографий) и определим для себя, насколько легко создать на его основе собственный проект.



Перед началом выполнения данного примера подготовьте несколько фотографий, которые можно будет использовать в шаблоне из категории PhotoSlideshow. Для подготовки изображений можно воспользоваться программой Macromedia Fireworks или любым другим графическим редактором.

Для получения наилучших результатов при подготовке изображений учтите перечисленные ниже замечания.

- Изображения должны иметь размер 640×480 пикселей. Для выполнения данного упражнения понадобится четыре изображения.
- Убедитесь, что выбранные изображения сохранены в формате JPEG или GIF и, таким образом, подходят для использования во Flash MX и в Web.
- Назовите фотографии в соответствии с некоторой последовательностью. Например: photo1.jpg, photo2.jpg, photo3.jpg и photo4.jpg.



Вместо создания собственных изображений можно использовать изображения, содержащиеся в папке chQS\images на компакт-диске, прилагаемом к этой книге. Скопируйте эти файлы на свой жесткий диск.

1. Откройте Flash MX.
2. Создайте новый документ из шаблона, выбрав команду File⇒New From Template (Файл⇒Создать из шаблона). Как следует из рис. БН.1, в открывшемся диалоговом окне New Document (Новый документ) содержится множество шаблонов, которые упрощают создание проектов. В некоторых наиболее завершенных шаблонах содержатся собственные компоненты для обеспечения интерактивности.

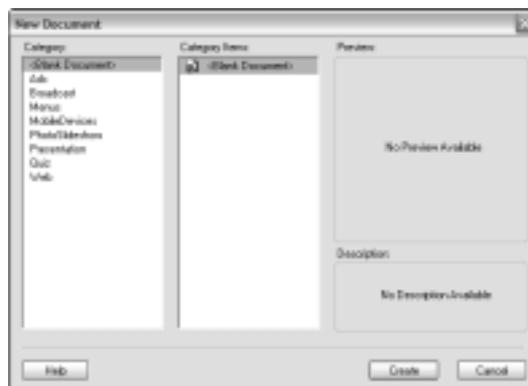


Рис. БН.1. Поиск в диалоговом окне *New Document* шаблонов, относящихся к категории *Photo Slideshow*

3. В списке *Category* (Категория) выберите пункт *Photo Slideshow*. Обратите внимание и на другие категории. При выборе элемента в списке *Category* в соседнем списке *Category Items* (Элементы категории) отображаются все шаблоны данной категории.
4. В списке *Category Items* выберите шаблон *Photo Slideshow_style1* и щелкните на кнопке *Create* (Создать). Во Flash MX откроется новый документ, отображающий на рабочем поле инструкции (рис. БН.2.). Многие панели инструкций из других шаблонов содержат URL-ссылки на страницы Web-узла Macromedia, где можно найти подробную информацию о выбранном шаблоне или категории.



Рис. БН.2. В каждом шаблоне используется собственный набор инструкций

5. Сохраните документ Flash (.fla-файл) на жестком диске вместе с уже подготовленными изображениями. Назовите его **slide_slow fla**. Верхний слой шаблона имеет название **_instructions**. На этом слое содержатся инструкции по использованию шаблона. Зачастую используется два, три или большее количество кадров с инструкциями, которые можно просматривать при помощи кнопок **CONTINUE** (далее) и **BACK** (назад). Эти кнопки должны работать в среде создания и редактирования документов Flash. Если кнопки не работают, проверьте выбрана ли команда **Control⇒Enable Simple Buttons** (Управление⇒Активизировать простые кнопки).

Инструкции в шаблоне могут показаться слегка неопределенными. Пример представляет собой полное руководство, гарантирующее успех при работе с шаблоном.

Слой с инструкциями можно скрыть, щелкнув на маленькой точке, расположенной в строке слоя `_instructions` под пиктограммой глаза. Однако рекомендуется все-таки просмотреть эти советы.

6. Выберите слой `Old Photos` (Старые фотографии). Этот слой следует удалить, поскольку вы собираетесь наполнить шаблон собственными фотографиями.
7. При выделенном слое `Old Photos` щелкните на кнопке `Delete Layer` (Удалить слой) — пиктограмме мусорной корзины в окне `Timeline` (Временная шкала).
8. Создайте новый слой, выбрав команду `Insert⇒Layer` (Вставить⇒Слой).
9. Щелкните дважды на названии нового слоя и введите слово **Photos**. Этот слой будет использоваться для импортированных изображений.
10. Выполните команду `File⇒Import` (Файл⇒Импорт), чтобы выбрать изображения, которые хотите использовать для собственного слайд-шоу. Перейдите в папку, где хранятся эти изображения, выделите первое из них и щелкните на кнопке `Open` (Открыть).
11. Если изображения названы последовательно, на экран будет выведено диалоговое окно с вопросом, следует ли импортировать все изображения последовательности. Щелкните на кнопке `Yes` (Да).
12. Если изображения названы `image1.jpg`, `image2.jpg` и так далее, отобразится диалоговое окно с сообщением о конфликте библиотек, показанное на рис. БН.3. Причина конфликта в том, что изображения, хранящиеся в библиотеке шаблона, имеют те же названия, что и импортируемые. Выберите опцию `Replace existing items` (Заменить существующие элементы) и щелкните на кнопке `OK`. Это новая возможность `Flash MX`, которая обеспечивает автоматическую замену уже существующих в библиотеке элементов импортируемыми в случае совпадения их названий (в предыдущих версиях `Flash` новые элементы были бы добавлены одновременно с сохранением в библиотеке документа уже существующих элементов). Если не хотите заменять элементы библиотеки, выберите в диалоговом окне `Resolve Library Conflict` опцию `Don't replace existing items` (Не заменять существующие элементы).

Все изображения последовательности будут импортированы, и для каждого из них будет создан ключевой кадр в слое `Photos`. Все, что осталось выполнить, — ввести для каждого изображения подпись, дату и заголовок.

13. Если при импортировании вы выбрали в диалоговом окне `Resolve Library Conflict` опцию `Don't replace existing items`, то после успешного импортирования новых изображений, можно удалить из библиотеки старые. Для этого откройте окно `Library` (Библиотека) с помощью команды `Window⇒Library` (Окно⇒Библиотека), а затем выделите все старые изображения (их имена останутся без изменения, а к совпадающим с ними именам новых изображений будет добавлено слово `copy`) и щелкните на пиктограмме мусорной корзины в нижней части окна библиотеки.
14. Выберите кадр 1 на слое `Title, Date` (Заголовок, дата).
15. Замените заголовок `Visiting the aquarium` заголовком, соответствующим используемому изображению.
16. Замените дату `February 2, 2002` на дату, соответствующую вашему изображению.
17. Выберите кадр 1 на слое `Captions` (Подписи). В левой нижней части сцены документа, должен находиться текст `The elegant seahorse`, окруженный синим прямоугольни-

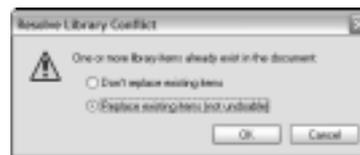


Рис. БН.3. Диалоговое окно `Resolve Library Conflict` (Разрешение конфликта библиотек)

ком; дважды щелкните на этом тексте для его редактирования. Введите соответствующую подпись для первого изображения. Повторите эти действия для остальных изображений. Для этого придется выбирать каждый кадр по очереди и вводить необходимый текст.

18. Протестируйте полученное слайд-шоу, для этого выберите команду **Control⇒Test Movie** (Управление⇒Тестирование фильма). Для смены изображений используйте кнопки контроллера фото-шоу (Photoshow Controller). При щелчке на кнопке **Autoplay** (Автоспроизведение) изображения будут сменяться автоматически; при достижении конца последовательности процесс начнется заново.

Как можно заметить, некоторые из шаблонов являются более полезными, чем другие. Поэтому можно по желанию выбирать любой шаблон, наилучшим образом подходящий для поставленных целей.



Не все шаблоны являются настолько завершенными, как шаблон, используемый в данном упражнении. Некоторые из них представляют собой просто фильмы с предустановленными размерами рабочего поля и общими инструкциями для его использования.

Компоненты и инспектор свойств

Капитальное обновление средства Smart Clip привело к появлению так называемых *компонент* (*components*). При помощи компонент можно устанавливать различные параметры в рабочей среде Flash MX, подобно тому, как это делалось с помощью Smart Clip во Flash 5. Однако, используя код ActionScript, можно изменять функциональность компонента прямо в процессе воспроизведения фильма Flash.

На узле Macromedia Exchange можно найти множество компонент, созданных разработчиками Flash MX. Для получения дополнительной информации обратитесь по адресу www.macromedia.com/exchange.

Вместе с Flash MX поставляется семь стандартных компонент интерфейса пользователя: **CheckBox** (Флажок), **ComboBox** (Поле со списком), **Listbox** (Список), **PushButton** (Кнопка), **RadioButton** (Переключатель), **ScrollBar** (Полоса прокрутки) и **ScrollPane** (Окно с прокруткой). Все эти элементы можно использовать для быстрого создания простого интерфейса, обеспечивающего взаимодействие с пользователем.

Рассмотрим пример создания полосы прокрутки для текстового поля. Преимущество этого компонента состоит в том, что для него не нужно использовать никакого кода ActionScript. Все уже выполнено за пользователя. Теперь посмотрим, как создать данный компонент и насколько просто с ним работать.

1. Откройте Flash MX и создайте новый файл, выбрав команду **File⇒New** (Файл⇒Создать).
2. Убедитесь в том, что отображается инспектор свойств. Если это не так, то откройте его, выбрав из меню команду **Window⇒Properties** (Окно⇒Свойства). Этот появившийся в версии Flash MX инспектор используется для изменения свойств объектов в среде редактирования и создания проектов Flash. С его помощью можно изменять свойства текстовых полей, экземпляров символов, видеоклипов, графических элементов, кнопок, звука, видео, кадров, а также векторных фигур.
3. Выберите инструмент **Text** (Текст). В раскрывающемся списке **Text type** (Тип текста) инспектора свойств выберите элемент **Input Text** (Вводимый текст).
4. Щелкните на рабочем поле и перетащите курсор, чтобы создать квадрат размерами 100×100 пикселей. Точный размер квадрата соблюдать необязательно.

5. В инспекторе свойств выберите из раскрывающегося списка Line type (Тип строки) элемент Multiline (Многострочный) и щелкните на кнопке Show Border Around Text (Показать границу вокруг текста) для отображения границ текстового поля. В поле Instance Name инспектора свойств введите **myText**.
6. Вы можете внедрить шрифт в фильм Flash, чтобы текст отображался на любом компьютере заданным шрифтом, независимо от того, установлен ли данный шрифт на компьютере пользователя. Для этого щелкните на кнопке Character (Знак) в инспекторе свойств. В появившемся диалоговом окне Character Options (Параметры знаков) выберите опцию All Characters (Все знаки) и закройте это окно, щелкнув на кнопке Done (Готово). Работа над текстовым полем закончена. Сравните установки своего текстового поля с приведенными на рис. БН.4. Теперь можно применять стандартный компонент ScrollBar (Полоса прокрутки).



Рис. БН.4. Инспектор свойств отображает правильные установки для текстового поля

7. Откройте панель Components (Компоненты) выбрав из меню команду Window⇒Components (Окно⇒Компоненты).
8. Выберите компонент ScrollBar и перетащите его экземпляр на рабочее поле документа. Отпустите кнопку мыши, когда поместите компонент внутрь текстового поля возле его правой границы. Другими словами, перетащите компонент и опустите его поверх текстового поля.
9. После этого полоса прокрутки должна сама установиться в правильное положение и динамически изменить свои размеры. Самое сложное позади. Теперь все, что необходимо сделать, это протестировать фильм и ввести текст в текстовое поле.
10. Сохраните документ с помощью команды File⇒Save (Файл⇒Сохранить) и протестируйте фильм, выбрав команду Control⇒Test Movie (Управление⇒Тестирование фильма).
11. Установите курсор внутри текстового поля и введите в него несколько строк текста.
12. Выделите только что введенный текст, нажав комбинацию клавиш <Ctrl+A> (для Windows) или <⌘+A> (для Macintosh), а затем скопируйте его в буфер обмена, нажав комбинацию клавиш <Ctrl+C> (для Windows) или <⌘+C> (для Macintosh).
13. Вставьте текст из буфера обмена в текстовое поле, нажав комбинацию клавиш <Ctrl+V> (для Windows) или <⌘+V> (для Macintosh). Повторите это действие несколько раз, чтобы заполнить все текстовое поле. После этого вставьте еще несколько таких фрагментов, чтобы убедиться, что текста достаточно много для использования полосы прокрутки.

Как только текстовое поле переполнится, полоса прокрутки активизируется — на ней появится бегунок, размеры которого обратно пропорциональны количеству текста в поле. Для просмотра текста можно перетаскивать бегунок полосы прокрутки вверх или вниз либо щелкать на навигационных кнопках со значками стрелок.

Теперь для читателя в использовании компонента ScrollBar нет ничего сложного. При помощи введенных в версии Flash MX компонентов пользовательского интерфейса очень просто добавить в фильм интерактивность.



Компонент ScrollBar также описывается в главе 20, "Создание первого проекта во Flash MX". Остальные компоненты описываются в главе 29, "Использование компонентов".

Быстрое создание анимации

В этом примере мы создадим небольшую анимацию. Благодаря этому вы освоите ряд новых возможностей Flash MX. Мы обсудим папки слоев, окно Library (Библиотека), модификатор Envelope (Огибающая), команды Break Apart (Разделить) и Distribute to Layers (Распределить по слоям).

Исследуемые здесь особенности являются предметом особого внимания компании Macromedia. Версия Flash MX была усовершенствована в соответствии с требованиями и интересами дизайнеров и аниматоров.



Файл quick_animation_start fla можно найти в папке ch08 на компакт-диске, прилагаемом к этой книге. Скопируйте файл на жесткий диск. Основной макет уже подготовлен, поэтому вы можете сосредоточиться на использовании новых возможностей Flash MX.

1. Выберите первый кадр слоя Flash text. На этом слое содержится текстовое поле со словом Flash.
2. Щелкните на кадре 20 этого же слоя и вставьте в него ключевой кадр, выбрав команду Insert⇒Keyframe (Вставка⇒Ключевой кадр).
3. Убедитесь в том, что 20-й кадр выделен именно в слое Flash text, после чего разделите текст при помощи команды Modify⇒Break Apart (Изменить⇒Разделить). Заметьте, что после этого каждая буква все еще является текстовым полем (но уже отдельным), что упрощает изменение букв в случае необходимости. Это одно из основных усовершенствований, касающихся работы с текстом во Flash MX.
4. Разделите текст еще раз, так чтобы превратить его в векторные фигуры.
5. Выберите из меню команду Modify⇒Transform⇒Envelope (Изменить⇒Преобразование⇒Огибающая). Модификатор Envelope (Огибающая) инструмента Free Transform (Свободное преобразование) является новым добавлением к инструментам рисования во Flash MX. Используйте модификатор Envelope для искажения векторных фигур на рабочем поле. Активизировать это средство также можно, выделив фигуру, выбрав инструмент Free Transform на панели инструментов рисования и щелкнув на кнопке Envelope в разделе Options этой же панели.
6. Используя инструмент Free Transform с активизированным модификатором Envelope, перетащите некоторые маркеры, окружающие преобразованный в фигуры текст, для искажения его формы (рис. БН.5).
7. После выполнения модификаций выделите первый ключевой кадр слоя Flash text.
8. Разделите текст, выбрав команду Modify⇒Break Apart. Примените эту команду еще раз. Поскольку мы собираемся применить операцию автоматического заполнения промежуточных кадров с интерполяцией изменения формы, в обоих ключевых кадрах должны содержаться примитивные фигуры.
9. Откройте инспектор свойств при помощи команды Window⇒Properties, если он еще не отображен. Примените операцию автоматического заполнения промежуточных

кадров с интерполяцией изменения формы, выбрав в раскрывающемся списке **Interpolate** (Интерполяция) (расположенном справа от надписи **Tween** (Заполнение кадров)) инспектора свойств элемент **Shape** (Форма).



*Рис. БН.5. Модификатор **Envelope** инструмента **Free Transform** можно использовать для изменения векторных фигур на рабочем поле*

10. Щелкните правой кнопкой мыши (для Windows) или воспользуйтесь комбинацией <Control+щелчок> (для Macintosh) на пиктограмме листа бумаги в строке слоя **Flash text** и выберите команду **Mask** (Маска). Слой преобразуется в слой маски, при этом находящийся под ним слой **green bar** становится подчиненным маскируемым слоем. Просматриваемая область подчиненных слоев определяется содержимым слоя маски.
11. Выберите первый кадр слоя **MXText** и разделите его содержимое при помощи команды **Modify⇒Brake Apart** (применив ее только один раз).
12. Распределите текст по слоям с помощью команды **Modify⇒Distribute to Layers** (Изменить⇒Распределить по слоям). Команда **Distribute to Layers** является новой возможностью Flash MX, используя которую можно распределить по отдельным слоям несколько элементов, расположенных на одном слое. Обратите внимание, что новым слоям автоматически присваиваются названия, соответствующие содержащимся в них буквам.
13. Откройте окно **Library** (Библиотека), выбрав команду **Window⇒Library** (Окно⇒Библиотека).
14. Выделите текстовое поле с буквой **M** на рабочем поле слоя **M**. Перетащите его в окно **Library** и отпустите кнопку мыши. На экране отобразится диалоговое окно **Convert to Symbol** (Преобразовать в символ). Эта операция аналогична выделению элемента на рабочем поле и последующему выбору команды **Insert⇒Convert to Symbol** (Вставка⇒Преобразовать в символ). Возможность перетаскивания в окно **Library** является еще одним усовершенствованием, появившемся во Flash MX и экономящим рабочее время пользователя.
15. В диалоговом окне **Convert to Symbol** выберите в группе **Behavior** (Поведение) переключатель **Graphic** (Графика), введите в поле **Name** (Имя) букву **M** и щелкните на кнопке **OK**.
16. Щелкните на кадре 20 в слое **M**. Вставьте в него ключевой кадр при помощи команды **Insert⇒Keyframe**.
17. Выделите кадр 1 слоя **M** и примените операцию автоматического заполнения промежуточных кадров с интерполяцией движения, выбрав из меню команду **Insert⇒Create Motion Tween** (Вставка⇒Создать заполнение кадров с интерполяцией движения).

18. Выделите символ М в первом ключевом кадре слоя М. Выберите в раскрывающемся списке Color Styles (Стили цвета), расположенном справа от надписи Color (Цвет) в инспекторе свойств, элемент Alpha. В инспекторе свойств появится поле Alpha Amount, и в нем автоматически установится значение, которое вы использовали при последнем применении данного цветового эффекта, — это еще одна особенность Flash MX, позволяющая экономить время при создании проекта. Для данного примера установите с помощью ползунка или введите вручную значение 0%.
19. Повторите пункты с 14 по 17 для текста X на слое X.
20. Во Flash 5 на этом работу над проектом можно было бы закончить. Однако во Flash MX добавлено несколько новых возможностей, при помощи которых можно усовершенствовать фильм. Например, создать папки слоев и разбить все слои на разделы анимации. Подобная методика чаще всего используется в больших проектах, где на временной шкале могут содержаться 40–90 слоев.
21. В окне Timeline щелкните на кнопке Insert Layer Folder (Вставить папку слоев) (расположенной в нижней левой части этого окна, под списком слоев). Переименуйте папку, дважды щелкнув на ее старом названии и введите новое название **Motion Tween**.
22. Перетащите слои М и X на папку слоев Motion Tween и отпустите кнопку мыши — слои переместятся в папку. Щелкнув на маленьком треугольнике (слева от пиктограммы папки), можно закрыть папку, скрыв в стеке слоев все содержащиеся в ней слои. Теперь удалите слой MXText, поскольку он больше не пригодится. Это можно осуществить, выделив слой и щелкнув на кнопке Delete Layer (Удалить слой) (с пиктограммой мусорной корзины), расположенной под стеком слоев в окне Timeline.
23. Создайте еще одну папку слоев, щелкнув на кнопке Insert Layer Folder в окне Timeline. Переименуйте ее в **Animated Mask**.
24. Перетащите слой маски с названием Flash text в папку Animated Mask. Все маскированные слои также будут перемещены в эту папку. Теперь закройте папку.
25. Сохраните документ с помощью команды File⇒Save и протестируйте его, выбрав из меню команду Control⇒Test Movie. Просмотрите анимацию меняющих форму букв.



Завершенный файл quick_animation_complete fla можно найти в папке chQS на компакт-диске, прилагаемом к этой книге.

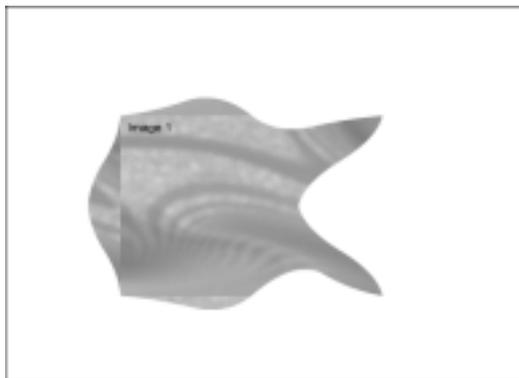
При создании этого примера анимации использовались новые возможности организации и управления Flash MX. Благодаря этим возможностям (в особенности улучшенной команде Break Apart и новой команде Distribute to Layers) можно быстрее создавать анимацию, на которую в версии Flash 5 пришлось бы потратить гораздо больше времени.

Растровые изображения и модификатор Envelope

Используя модификатор Envelope (Огибающая), можно получить действительно интересные эффекты. Вы можете расширить область его применения до работы с растровыми рисунками. Для того чтобы применить данный инструмент к растровым рисункам, выполните следующее.

1. Откройте Flash и создайте новый файл, выбрав команду File⇒New.
2. Импортируйте любой растровый файл, хранящийся на жестком диске, выбрав команду File⇒Import (Файл⇒Импорт). Откройте соответствующую папку, выделите необходимый рисунок, и щелкните на кнопке Open (Открыть).
3. Выделите рисунок на рабочем поле и разделите его с помощью команды Modify⇒Break Apart.

4. Убедитесь в том, что растровый рисунок выделен, и выберите команду `Modify⇒Transform⇒Envelope` (Изменить⇒Преобразование⇒Огибающая). Теперь можете перетаскивать окружающие объект узлы огибающей, подобно работе с векторной фигурой. Однако имеются и небольшие отличия — результат выглядит подобно маске, примененной к слою, заполненному мозаикой из растровых рисунков, — соответствующий пример приводится на следующем рисунке. И тем не менее, используя эту возможность можно создавать прекрасные эффекты.



Пример применения модификатора Envelope к растровому рисунку

Примеры для разработчиков

Кроме усовершенствований для дизайнеров, во Flash MX были добавлены усовершенствования, полезные и для разработчиков. В частности, в новой версии программы используется значительно измененный и переработанный язык сценариев ActionScript, дополненный более широкими возможностями. Некоторые из этих усовершенствований рассматриваются в данном разделе.

Компонент ComboBox

Компонент ComboBox (Поле со списком) сходен с раскрывающимся меню в HTML-документе. Используя данный компонент, можно создать раскрывающийся список, например, имен людей. При выборе определенного имени откроется окно Output, с отображенным в нем возрастом этого человека.



Начальный файл `components_start fla` можно найти в папке `ch05` на компакт-диске, прилагаемом к этой книге. Скопируйте этот файл на жесткий диск.

1. Откройте документ `components_start fla`, скопированный на жесткий диск. Поскольку графика уже создана, можете сосредоточиться на применении компонента ComboBox. Обратите внимание на то, что слева на рабочем поле расположены два текстовых поля (рис. БН.6). Первое представляет переменную `name`, а второе — переменную `age`.
2. Выделите первый кадр слоя Component, а затем откройте панель Components с помощью команды `Window⇒Components` и перетащите экземпляр компонента ComboBox на рабочее поле.



Рис. БН.6. Рабочее поле исходного документа

3. Откройте инспектор свойств (если он еще не отображен) выбрав команду **Window**⇒**Properties**.
4. В инспекторе свойств вы увидите параметры экземпляра компонента **ComboBox**. Укажите имя для этого экземпляра, заменив в поле **Instance Name** (Имя экземпляра) инспектора свойств текст `<Instance Name>` на **dropdown**. Это имя теперь можно использовать для обращения к данному экземпляру компонента **ComboBox** в коде **ActionScript**.
5. По умолчанию для экземпляра компонента в инспекторе свойств будет отображена вкладка **Parameters** (Параметры). Выберите в списке параметр **Labels** (Надписи). Щелкните на белом прямоугольнике, появившемся вокруг знаков `[]` справа от названия параметра, после чего откроется диалоговое окно **Values** (Значения). Здесь следует ввести значения текущего параметра. Параметр **Labels** представляет собой массив, используемый для хранения надписей, отображаемых в компоненте **ComboBox**.
6. Щелкните на кнопке **+** в верхнем левом углу диалогового окна **Values**. После появления в таблице значений нового элемента, выделите его и введите текст на свое усмотрение (например, любое имя и фамилию). Этот текст отобразится в качестве надписи в компоненте **ComboBox** при экспорте документа в фильм **Flash** (`.swf`-файл). Повторите этот процесс для ввода необходимого количества имен. Щелкните на кнопке **OK**.
7. В инспекторе свойств выберите параметр **Data** (Данные) и откройте диалоговое окно **Values**, щелкнув на белом прямоугольнике, появившемся вокруг знаков `[]` справа от названия параметра.
8. Добавьте новый элемент с помощью кнопки **+** и введите возраст воображаемого человека. Если вы ввели несколько имен в п. 6, повторите процесс для добавления информации о возрасте каждого из них. Эти значения следует вводить в том же порядке, что и значения параметра **Labels**.
9. Теперь выберите параметр **Change Handler** (Обработчик изменения) и введите в поле, появившееся справа от него, текст **dropDownChanged**. `dropDownChanged` — это функция, выполняемая при каждом выборе, сделанном в компоненте **ComboBox**. После всех этих операций установки вашего экземпляра компонента должны соответствовать показанным на рис. БН.7.
10. Выберите кадр 1 слоя **Actions** и с помощью команды **Window**⇒**Actions** (Окно⇒Действия) откройте панель **Actions** (Действия). В меню параметров данной панели (оно открывается после щелчка на пиктограмме в правом верхнем углу панели)

выберите команду **Expert Mode** (Экспертный режим). Теперь в окне сценария панели **Actions** введите следующий код.

```
// данная функция будет вызвана при выборе элемента в
// компоненте ComboBox.
function dropDownChanged(obj) {
// Получение надписи и значения выбранного элемента
var name = obj.getSelectedItem().label;
var age = obj.getSelectedItem().data;
// Передача имен и возраста людей в окно Output
trace("You have selected "+name+", who is "+age+" years old.");
}
```

Функция `dropDownChanged` представляет собой метод обратного вызова экземпляра компонента `ComboBox` и выполняется при каждом выборе в этом экземпляре.



Рис. БН.7. Установки во вкладке *Parameters* инспектора свойств должны соответствовать установкам на данном рисунке



Действие `trace()` работает только в режиме тестирования фильма (команда `Control⇒Test Movie`). Оно просто получает из переменных имя и возраст выбранного человека и отправляет эти значения в окно `Output`.

11. Выберите на рабочем поле кнопку с надписью `add person` (добавление человека) и откройте панель **Actions** при помощи команды `Window⇒Actions`. Теперь введите следующие строки кода:

```
on (press) {
// Добавление имени и возраста человека в экземпляр
// компонента ComboBox.
dropDown.addItem(name, age);
}
```

При выполнении этого фрагмента кода имя и возраст человека будут добавлены в экземпляр `dropDown` компонента `ComboBox`. Здесь используются значения переменных `name` и `age` из текстовых полей, размещенных на рабочем поле.

12. Теперь сохраните документ `Flash` и протестируйте его, выбрав команду `Control⇒Test Movie`. Щелкните на направленной вниз стрелке экземпляра компонента `ComboBox` и выберите любое имя. На экране отобразится окно `Output`, в котором будут выведены имя и возраст выбранного человека.
13. Введите в соответствующих текстовых полях свое собственное имя и возраст. Щелкните на кнопке `add person`, после чего ваше имя и возраст будут добавлены в экземпляр компонента `ComboBox`.
14. Щелкните на направленной вниз стрелке в экземпляре компонента `ComboBox` и выберите свое имя, после этого ваше имя и возраст отобразятся в окне `Output`.

Данный пример можно совершенствовать, а также находить другие применения экземпляра компонента `ComboBox`. Для того чтобы заставить текстовое поле, над кото-

рым расположена надпись `current selection`, отображать текущий выбор в экземпляре компонента `ComboBox`, выполните п.15 и п. 16.

15. Выйдите из режима тестирования фильма, закрыв окно `.swf`-файла (но не самой программы Flash MX).
16. В конце кода функции `dropDownChanged` добавьте следующий фрагмент:

```
currentSelection = name;
```

Этот код задает для переменной `currentSelection` значение, которое содержится в переменной `name`, а переменная `currentSelection` назначена для текстового поля помеченного надписью `current selection`. Для тестирования модифицированной версии повторите пп. 12–14.



Законченный документ `components_complete.fla` содержится в папке `ch05` на компакт-диске, прилагаемом к этой книге.

В рассмотренном примере использовался инспектор свойств для изменения компонента в среде редактирования и создания документа Flash, а также простые команды `ActionScript` для изменения компонента `ComboBox` во время выполнения фильма.



В данном упражнении в полях вводимого текста (т.е. таких, для которых тип текста в раскрывающемся списке `Text type` инспектора свойств задан как `Input Text`) для извлечения содержащейся в них информации используются переменные, имена которых задаются в поле `Variable` (Переменная), расположенном справа от надписи `Var` в инспекторе свойств. Во Flash MX можно также присваивать имена экземплярам текстовых полей. В последующих главах мы будем использовать имя экземпляра и свойства объекта `TextField` для работы с компонентами и другими элементами интерфейса. Несмотря на то что при помощи переменных, задаваемых в поле `Variable` инспектора свойств, можно просто обращаться к содержимому текстового поля, мы не рекомендуем использовать эти переменные в проектах, предназначенных для Flash Player 6.

Использование компонента `ListBox`

Компоненты `ListBox` (Список) и `ComboBox` (Поле со списком) весьма схожи. Следуйте приведенным ниже инструкциям, чтобы заменить компонент `ComboBox` на компонент `ListBox`. Для этого необходимо сначала выполнить все пункты в примере с компонентом `ComboBox`.

1. Выделите экземпляр компонента `ComboBox` на рабочем поле и удалите его, нажав клавишу `<Delete>`.
2. Если панель `Components` еще не открыта, откройте ее, выбрав команду `Window⇒Components`.
3. Перетащите на рабочее поле экземпляр компонента `ListBox`.
4. Если после этого на экране отобразится диалоговое окно `Resolve Component Conflict` (Разрешение конфликта компонент), выберите в нем опцию `Replace existing component` (Заменить существующий компонент) и щелкните на кнопке `OK`. (Похоже, что в окончательной версии программы Flash MX разработчики компании `Macromedia` решили проблему конфликта компонент, поскольку в ней диалоговое окно `Resolve Component Conflict` не появляется при добавлении на рабочее поле нового компонента. — *Прим. ред.*)
5. Выполните пп. 4–9 примера с `ComboBox`. Единственное отличие проектов состоит в использовании компонента `ListBox` вместо `ComboBox`.

6. Протестируйте фильм при помощи команды **Control⇒Test Movie**. Как можно убедиться, здесь предоставлены те же самые функциональные возможности, что и в примере с использованием компонента **ComboBox**.

Этот пример демонстрирует легкость использования и изменения компонент **Flash MX**.

Настраиваемая панель **Actions**

Компания **Macromedia** всерьез сконцентрировала свои усилия на усовершенствовании программы **Flash MX** для разработчиков. Одним из основных недостатков во **Flash 5** была ограниченная панель **Actions**, которую неудобно было использовать при введении большого количества кода **ActionScript**. Зато теперь во **Flash MX** содержится усовершенствованная и настраиваемая панель **Actions**, гораздо более удобная для работы с текстом сценария **ActionScript**.

Рассмотрим конфигурирование панели **Actions** для ваших нужд.

1. Откройте **Flash MX** и создайте новый файл, выбрав команду **File⇒New**.
2. Выбрав из меню команду **Edit⇒Preferences** (**Правка⇒Настройка**), откройте диалоговое окно **Preferences** (**Настройка**), а затем выберите в нем вкладку **ActionScript Editor** (**Редактор ActionScript**) (рис. БН.8).
3. В разделе **Editing Options** (**Параметры редактирования**) рекомендуется использовать устанавливаемые по умолчанию опции **Automatic Indentation** (**Автоматический отступ**) и **Code Hints** (**Подсказки кода**). Можете устанавливать с помощью ползунка **Delay** (**Задержка**) любую (от 0 до 4 секунд) задержку для отображения подсказки; лучше использовать 0, поскольку мало толку от подсказки, когда вы уже ввели имя функции.
4. В поле **Tab Size** (**Размер табуляции**) можно устанавливать любое значение размера табуляции; довольно удобно использовать одни и те же настройки в различных программах редактирования кода.
5. В группе **Text** (**Текст**) выберите любой наиболее удобный для себя шрифт. Выбирайте моноширинные шрифты (т.е. шрифты, в которых для любого знака используется одинаковая ширина), наиболее часто используемые в языках программирования, поскольку они облегчают работу с кодом.
6. Цветной синтаксис, обеспечиваемый опциями группы **Syntax Coloring**, представляет собой огромное преимущество панели **Actions** в **Flash MX**. Можно выбрать цвета, используемые в остальных программах редактирования кода. Если вы новичок в написании кода, следует немного поэкспериментировать и найти наиболее удобный для себя вариант.
7. Закройте окно **Preferences** и откройте панель **Actions** при помощи команды **Window⇒Actions**. Как можно убедиться, панель подверглась радикальным изменениям. Большинство параметров для настройки панели содержится в меню параметров, расположенном в верхнем правом углу панели. Многие из них предназначены для работы с кодом. Например, опция **Check Syntax** (**Проверка синтаксиса**) служит для нахождения синтаксических ошибок в коде. При выборе опции **View Line Numbers** (**Отобразить номера строк**) перед строками кода добавляются номера. Вы также можете из обычного режима редактирования кода (**Normal Mode**) в экспертный (**Expert Mode**).



Рис. БН.8. Диалоговое окно **Preferences** можно использовать для настройки большинства параметров **Flash MX**



Более опытные пользователи могут изменить редактор ActionScript в файле `ActionsPanel.xml`, который содержится в папке `First Run\ActionsPanel` в программной папке Flash MX. Однако эту операцию следует выполнять крайне осторожно. Настоятельно рекомендуется создать резервную копию файла перед его изменением. Более подробно об этих “продвинутых” функциональных возможностях рассказывается в книге *Macromedia Flash MX ActionScript. Библия пользователя* Роберта Рейнхардта (Robert Reinhardt) и Джоя Лотта (Joey Lott).

Именованные анкеры

Улучшая функциональность Flash Player, компания Macromedia разработала методы, позволяющие использовать кнопки **Back** (Назад) и **Forward** (Вперед) в Web-браузерах для управления Flash-фильмом, причем это не представляет особой сложности. Теперь нет никаких оправданий для создателей таких Web-узлов, основанных на Flash, которые не отвечают определенным требованиям удобства и простоты использования.



Именованные анкеры (named anchors) работают только в Web-браузерах, поддерживающих JavaScript и действия `fscommand`. Это означает, что Internet Explorer для Macintosh, а также Netscape 6.0 (или более новая версия) не будут работать с именованными анкерами.

Далее приводится пример для демонстрации использования именованных анкеров во Flash; это простейшее слайд-шоу, в котором установлено взаимодействие с кнопками браузера **Back** (Назад) и **Forward** (Вперед).



Файл `named_anchors_start.fla` содержится в папке `ch05` на компакт-диске, прилагаемом к этой книге. Скопируйте этот файл на жесткий диск.

1. Откройте Flash MX.
2. Теперь откройте файл `named_anchors_start.fla`, выбрав команду `File⇒Open` (Файл⇒Открыть). Найдите сохраненный на жестком диске файл, выделите его и щелкните на кнопке `Open` (Открыть). Здесь уже создана основная графика и последовательность кадров. Все, что следует выполнить, — это создать именованные анкеры и протестировать фильм в браузере.
3. Выберите первый ключевой кадр на слое с названием `Labels` (Метки).
4. При помощи команды `Window⇒Properties` откройте инспектор свойств, если он еще не отображен на экране.
5. Из поля `Frame Label` (Метка кадра) в инспекторе свойств удалите `// intro` и введите просто `intro`.
6. Установите флажок `Named Anchor` (Именованный анкер). После установки этого флажка в кадре на временной шкале появится маленький значок якоря, обозначающий, что кадр является именованным анкером. На рис. БН.9 отображены значки именованных анкеров.
7. Повторите п. 5 и п. 6 для каждого содержащего комментарий кадра на слое `Labels`, включая последний. Назовите их следующим образом: `intro` (его вы уже обработали), `sec1`, `sec2` и `end`. Сценарий ActionScript для кнопок перехода к каждой странице, использующий приведенные выше названия кадров, уже написан, поэтому убедитесь в правильности их написания. Теперь осталось только опубликовать документ.

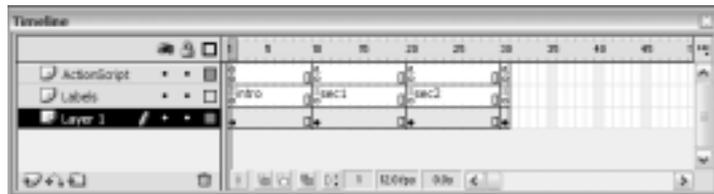


Рис. БН.9. На временной шкале документа *named_anchors_start fla* показаны значки, используемые для обозначения именованных анкером

8. Откройте окно **Publish Settings** (Параметры публикации), выбрав из меню команду **File⇒Publish Settings** (Файл⇒Параметры публикации).
9. Убедитесь в том, что на вкладке **Formats** (Форматы) этого диалогового окна установлены флажки **Flash** и **HTML**.
10. Выберите вкладку **HTML**. Заданные по умолчанию настройки на вкладке **Flash** не нужно изменять.
11. В раскрывающемся списке **Template** (Шаблон) выберите **Flash with Named Anchors** (Flash с именованными анкерами). Выбирайте при публикации эту опцию всякий раз, когда хотите использовать именованные анкеры.
12. После изменения других настроек щелкните на кнопке **Publish** (Опубликовать), а затем на кнопке **OK**.
13. И, наконец, осталось только открыть созданный HTML-файл в браузере. Выполните это, используя выбранный вами Web-браузер. Щелкните несколько раз на кнопке в нижней части страницы для последовательного перехода по разделам фильма. После этого вы сможете использовать кнопки браузера **Back** (Назад) и **Forward** (Далее) для перехода к уже просмотренным разделам фильма.



Чтобы просмотреть полную версию данного примера, откройте документ *named_anchors_complete fla* в папке *ch05* на компакт-диске, прилагаемом к этой книге.

Динамическая загрузка растровых рисунков и звука

На протяжении долгого времени программа Macromedia Flash представляла собой прекрасный инструмент для динамического обновления Web-узла. Можно было загружать XML, текстовые файлы и переменные для быстрого обновления узла. В настоящее время появилась возможность динамической загрузки растровых рисунков и звуковых файлов. Во Flash 5 это можно было осуществить только при помощи Generator или подобной программы. Теперь Macromedia предоставила пользователям опцию для осуществления такой возможности непосредственно во Flash MX.

В данном упражнении создается простой пример, в котором загружаются файлы форматов MP3 и JPEG.



Файл *atmospheres_1.mp3* содержится в папке *ch28*, файл *image1.jpg* — в папке *ch05\images* на компакт-диске, прилагаемом к этой книге. Скопируйте эти файлы в одну папку на жесткий диск. Новый файл Flash MX нужно будет сохранить в этой же папке.

1. Откройте Flash MX и создайте новый файл при помощи команды **File⇒New**. В данном примере не придется ничего рисовать, здесь используется только ActionScript.
2. Сохраните этот файл в той же папке на жестком диске, в которой были сохранены .mp3- и .jpg-файл, и назовите его **dynamic_loading fla**.
3. Щелкните дважды на имени слоя **Layer 1** и назовите его **actions**.
4. Выберите первый кадр слоя **actions**.
5. Откройте панель **Actions**, выбрав команду **Window⇒Actions**, и введите приведенный ниже код. Убедитесь в том, что в меню параметров панели выбрана опция **Expert Mode** (Экспертный режим).

```
// создание пустого видеоклипа для хранения звукового объекта
_root.createEmptyMovieClip("bgSoundHolder", 1);
```

В данном фрагменте кода создается пустой видеоклип. Используйте этот видеоклип для хранения звука. Таким образом можно одновременно проигрывать несколько звуковых файлов, каждому из которых будет соответствовать индивидуальные средства управления громкостью и настройками.

6. В том же кадре вставьте следующий фрагмент кода:

```
// создание нового объекта Sound в bgSoundHolder
bgSound = new Sound("bgSoundHolder");
// загрузка указанного mp3-файла
BgSound.loadSound("atmospheres_1.mp3", true);
```

При выполнении кода создается новый экземпляр объекта **Sound** под названием **bgSound**. Затем используется новый метод **loadSound()** для динамической загрузки и потокового воспроизведения .mp3-файла.

7. Протестируйте фильм при помощи команды **Control⇒Test Movie**. Воспроизведение .mp3-файла начнется сразу после того, как будет сгенерирован фильм Flash (.swf-файл).



Убедитесь в том, что файл **atmospheres_1.mp3** скопирован в ту же папку, где сохранен фильм Flash. Если эти два файла находятся в различных папках, фильм не будет "работать". В данном примере можно использовать свой собственный файл формата MP3, следует только отредактировать его имя в методе **loadSound()**.

8. Теперь, после импортирования звука, в этот же фильм Flash можно динамически загрузить JPEG-изображение.
9. Выберите первый кадр слоя **actions**.
10. Если панель **Actions** до сих пор не отображена, откройте ее с помощью команды **Window⇒Actions** и вставьте следующий код:

```
// создание пустого видеоклипа для хранения в нем
// растрового рисунка
_root.createEmptyMovieClip("bitmapHolder", 2);
// загрузка растрового рисунка
bitmapHolder.loadMovie("image1.jpg");
```

Сначала этот код создает новый, пустой видеоклип для хранения в нем растрового рисунка, динамически загружаемого во время воспроизведения фильма. Затем используется усовершенствованная функция **loadMovie**, при помощи которой производится непосредственное обращение к .jpg-файлу для его загрузки.

11. Протестируйте фильм, выбрав команду **Control⇒Test Movie**. После того как фильм Flash (.swf-файл) будет сгенерирован, в его верхнем правом углу отобразится растровый рисунок и начнет проигрываться звук.

На данном примере можно легко убедиться в простоте использования динамических элементов во Flash MX. Это новая мощная возможность, которая открывает новые горизонты для разработчиков Flash MX.

Потоковое видео

Flash MX также предоставляет возможность импорта и потокового воспроизведения видеофайлов. Для декодирования видеоданных в версии Flash MX используется кодек Sorenson Spark. Вы можете осуществлять полный контроль над воспроизведением видеофайла — запускать его, останавливать, а также перематывать, используя собственные кнопки.

Мы создали простой пример импортирования видеофайла Windows (.avi-файла) и его потокового воспроизведения в фильме Flash.



Файл `robertSpin.avi` содержится в папке `ch17\source` на компакт-диске, прилагаемом к этой книге. Скопируйте этот файл на жесткий диск.

Импортирование видеофайла

Прежде всего необходимо импортировать цифровой видеофайл в документ Flash MX. Для этого можно использовать .avi-файл, содержащийся на компакт-диске, или какой-либо собственный файл.

1. Откройте Flash MX и создайте новый файл, выбрав команду меню `File⇒New`.
2. Сохраните этот файл на жестком диске под названием **video_import fla**; убедитесь в том, что сохраняете его в той же папке, в которой был сохранен файл `robertSpin.avi`.
3. Создайте новый символ видеоклипа для “вложения” в него видеоматериала, для этого выберите команду `Insert⇒New Symbol` (Вставка⇒Новый символ).
4. В диалоговом окне `Create New Symbol` (Создание нового символа) в поле `Name` (Имя) введите **video**.
5. Убедитесь в том, что в группе `Behavior` (Поведение) выбран переключатель `Movie Clip` (Видеоклип), и щелкните на кнопке `OK`. Вы окажетесь в режиме редактирования символа, при этом отобразится рабочее поле нового видеоклипа.
6. Импортируйте .avi-файл при помощи команды `File⇒Import`. В открывшемся диалоговом окне `Import` (Импорт) найдите файл `robertSpin.avi`, выделите его и щелкните на кнопке `Open` (Открыть).
7. На экране отобразится диалоговое окно `Import Video Settings` (Параметры импортированного видео), показанное на рис. БН.10. Здесь можно изменять свойства импортированного видеофайла, например размеры и качество. Для воспроизведения видеоданных во Flash используется кодек Sorenson Spark.
8. Установите с помощью бегунка `Quality` (Качество) значение 80%, с помощью бе-

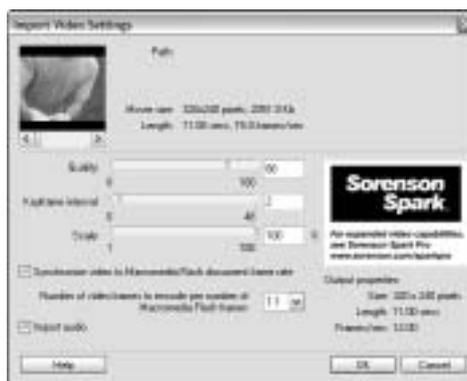


Рис. БН.10. Диалоговое окно `Import Video Settings`

гунка **Keyframe interval** (Интервал между ключевыми кадрами) — 2 и с помощью бегунка **Scale** (Масштаб) — 80%. Можно изменять эти параметры как угодно; использование ползунка **Quality** позволяет выбрать установку, наиболее соответствующую вашим нуждам, но не забывайте о размерах файла.

9. Установите флажок **Import Audio** (Импорт аудио) для импортирования звука вместе с видеоданными. Щелкните на кнопке **OK**. После этого видеоданные будут сжаты в кодеке Sorenson Spark; это может занять некоторое время, в зависимости от быстродействия используемой машины.
10. На экране отобразится диалоговое окно с вопросом о том, следует ли автоматически создать во Flash кадры в соответствии с длиной импортируемого видеофайла, щелкните на кнопке **Yes** (Да). После этого видеоданные будут вставлены на временную шкалу документа.
11. Сохраните документ Flash как **embeddedVideo fla**.

Размещение видеоданных

В этом разделе описывается процесс помещения видеоданных в первую сцену документа.

1. Вернитесь к основной временной шкале (то есть к сцене **Scene 1**) документа `embeddedVideo fla`, щелкнув на ссылке **Scene 1** в строке сцены и символа, расположенной сразу под окном **Timeline**, или выберите сцену **Scene 1** на панели **Scene** (Сцена). Панель **Scene** можно открыть с помощью команды **Window⇒Scene** (Окно⇒Сцена).
2. Откройте окно **Library** (Библиотека), выбрав команду **Window⇒Library**.
3. Поместите экземпляр видеоклипа `video` на рабочее поле, для этого щелкните в окне библиотеки на символе `video` и перетащите его на рабочее поле.
4. Если инспектор свойств не отображен на экране, откройте его при помощи команды **Window⇒Properties**.
5. Выберите экземпляр видеоклипа `video` на рабочем поле. В поле **Instance Name** (Имя экземпляра) инспектора свойств введите `video`.
6. Установите в инспекторе свойств координаты для видеоклипа `video`. В поле **X** введите **120**, а в поле **Y** — **90**.

Можно (и даже предпочтительнее) ввести эти координаты на панели **Info** (Информация), поскольку на ней, в отличие от инспектора свойств, имеется средство управления, позволяющее выбрать, относительно какой точки (верхнего левого угла или точки регистрации, которая обычно совпадает с центром) экземпляра видеоклипа задаются координаты. По умолчанию в инспекторе свойств координаты задаются относительно точки, которая была задана при последнем использовании панели **Info**. В данном примере указанные координаты необходимо задавать относительно верхнего левого угла экземпляра видеоклипа. Поэтому откройте панель **Info** с помощью команды **Window⇒Info** и убедитесь, что на сетке выравнивания, расположенной слева от полей **X** и **Y**, выбран (выделен черным цветом) квадратик в верхнем левом углу. — *Прим. ред.*

7. Снова сохраните документ Flash (**File⇒Save**).

Управление импортированным видеоматериалом

Теперь, когда экземпляр видеоклипа находится на главной временной шкале, к нему можно обратиться с помощью **ActionScript** для управления воспроизведением импортированных видеоданных.

1. Теперь видеоматериал готов к использованию. Однако необходимо некое средство управления для взаимодействия с ним. Выберите команду **Window**⇒**Common Libraries**⇒**Buttons** (Окно⇒Общие библиотеки⇒Кнопки). Можно воспользоваться библиотекой кнопок для создания кнопок остановки, проигрывания и перемотки.
2. В библиотеке кнопок откройте папку **Playback** (Воспроизведение).
3. Перетащите на рабочее поле экземпляр символа кнопки **gel Rewind**.
4. Перетащите на рабочее поле также экземпляры кнопок **gel Stop**, **gel Right** и **gel Fast Forward**.
5. Если инспектор свойств не отображен на экране, откройте его при помощи команды **Window**⇒**Properties** и используйте для точного позиционирования кнопок в следующем порядке (предлагаемое размещение показано на рис. БН.11):
 - **gel Rewind**; X: 210 и Y: 300
 - **gel Stop**; X: 244 и Y: 300
 - **gel Right**; X: 279 и Y: 300
 - **gel Fast Forward**; X: 313 и Y: 300



Рис. БН.11. После выполнения п. 5 ваш фильм должен выглядеть, как показано здесь

6. Выберите кнопку **gel Rewind** и откройте панель **Actions** (если она не открыта) с помощью команды **Window**⇒**Actions**.
7. Перейдите в экспертный режим, выбрав команду **Expert Mode** в меню параметров данной панели, расположенном в ее верхнем правом углу.
8. Введите следующий код **ActionScript**:


```
on (release) {
  video.gotoAndStop(1);
}
```

Этот код возвращает фильм к первому кадру видеofilьма. Взаимодействие с видеоклипом **video** происходит так же, как и с любым другим видеоклипом, таким образом, организация управления видеofilьмом является чрезвычайно простой задачей.
9. Выберите кнопку **gel Stop** и введите в панели **Actions** следующий код **ActionScript**:


```
on (release) {
  video.stop();
}
```

Этот код останавливает воспроизведение видеofilьма в том кадре, который являлся текущим в момент щелчка на кнопке, что подобно действию традиционной кнопки “Пауза”.

10. Выберите кнопку **gel Right** и введите в панели **Actions** следующий код ActionScript:

```
on (release) {  
    video.play();  
}
```

При выполнении этого кода видеofilьм начинает воспроизводиться с текущей позиции.

11. И наконец введите код ActionScript для последней кнопки **gel Fast Forward**:

```
on (release) {  
    video.gotoAndStop(video._totalframes);  
}
```

Данный код перемещает видеofilьм к последнему кадру. Свойство `_totalframes` объекта `MovieClip` используется для определения количества кадров в экземпляре `video`.

12. Весь код ActionScript, необходимый для управления фильмом, введен.
13. Протестируйте свой фильм, выбрав команду **Control**⇒**Test Movie**. Теперь для управления воспроизведением видеofilьма можете воспользоваться кнопками воспроизведения, остановки и быстрой перемотки назад или вперед.

Обратите внимание на размер экспортируемого `.swf`-файла, в котором содержится видеofilьм. Нетрудно заметить, что он довольно хорошо оптимизирован во Flash MX.



Дополнительная информация о кодеке Sorenson Spark Pro и использовании видеоматериалов в фильмах Flash MX содержится в главе 17, “Внедрение видеоматериалов”.

Программируемые текстовые поля

Во Flash MX можно динамически создавать текстовое поле при помощи ActionScript. Это одно из усовершенствований и добавлений к языку ActionScript. В данном примере динамически создается текстовое поле, которое заполняется текстом, загружаемым из `.txt`-файла.

1. Откройте Flash MX и создайте новый файл (**File**⇒**New**). Графика в данном примере не используется; здесь все создается с помощью ActionScript.
2. Сохраните файл на жестком диске и назовите его **text_field fla**.
3. Используя любой текстовый редактор, создайте новый `.txt`-файл и сохраните его на диске в той же папке, где содержится документ Flash (`.fla`-файл). Назовите этот текстовый файл **text.txt**.

4. Вставьте в новый текстовый файл следующую строку:

```
textMessage=this+is+the+information+in+the+text+field
```

Здесь представлен текст, который будет отображен в текстовом поле после его создания во Flash.

5. Вернитесь назад во Flash MX и выберите первый кадр на слое **Layer 1**.
6. Откройте, если нужно, панель **Actions** и вставьте следующий фрагмент кода:

```
// создание пустого видеоклипа для хранения в нем  
// текстового поля  
_root.createEmptyMovieClip("textFieldHolder", 1);
```

Этот код создает пустой видеоклип для хранения в нем текстового поля, создаваемого в следующем пункте.

7. Под строками только что введенного кода вставьте следующий фрагмент:

```
// создание нового объекта LoadVars
info = new LoadVars();

// вызов функции writeTextField при загрузке данных
info.onLoad = writeTextField;

// загрузка текстового файла во Flash
info.load("text.txt");
```

Сначала данный код создает объект `LoadVars`, который является новинкой Flash MX. Далее функция `writeTextField` назначается для события `onLoad`. Событие `onLoad` выполняется при окончании загрузки во Flash текстового файла. И, наконец, в третьем блоке информация из указанного текстового файла загружается во Flash.

8. Вставьте следующий код непосредственно после предыдущего блока. Не вводите символ `↵`, он обозначает продолжение той же строки кода.

```
// ФУНКЦИЯ: используется для создания текстового поля
function writeTextField (success) {

    // проверяется, была ли информации корректно загружена
    if (success == true) {

        // назначение текстового поля новому видеоклипу
        textFieldHolder.createTextField ("myTextField", ↵
            2, 0, 0, 200, 20);

        // отображение границ поля
        textFieldHolder.myTextField.border = true;

        // присвоение текстовому полю значения переменной
        // textMessage из загруженного текстового файла
        textFieldHolder.myTextField.text = ↵
        this.textMessage;
    }
}
```

Функция `writeTextField` запускается сразу после загрузки информации. В ней определяется правильность загрузки содержимого файла, после чего создается текстовое поле и для его свойства `text` присваивается значение переменной `textMessage`, загруженной во Flash из текстового файла (`.txt`).

9. Протестируйте фильм, выбрав команду `Control⇒Test Movie`. В верхнем левом углу Flash-фильма должно отобразиться текстовое поле с текстом из `.txt`-файла.

Подобная доработка объектной модели Flash-документа является большим вкладом в возможности Flash MX.

Модель событий ActionScript и API-рисование

Основным преимуществом языка ActionScript во Flash MX является доработанная модель событий. Теперь можно применять событие к объекту динамически посредством ActionScript. Данная возможность является довольно полезной для разработчиков приложений во Flash.

Вторым важным усовершенствованием является возможность взаимодействия с API (программным интерфейсом приложения) для рисования. Теперь можно создавать векторную графику динамически во время воспроизведения фильма.

В данном примере с помощью ActionScript создается пустой видеоклип, в котором выводится черный квадрат размером 50×50 пикселей и выполняется анимация.

1. Откройте Flash MX и создайте новый файл (File⇒New).
2. Сохраните файл на жестком диске и назовите его **drawing_api fla**.
3. Выделите первый кадр слоя Layer 1 и в панели Actions введите следующий код:

```
// создание пустого видеоклипа для рисования в нем квадрата  
_root.createEmptyMovieClip("square", 1);
```

Этот код просто создает новый видеоклип, который будет использоваться для “рисования” векторной графики (список методов API-рисования приводится на рис. БН.12).

4. В следующих строках кода для вывода черного квадрата со стороны 50 пикселей используется новый метод API-рисования.

```
// вывод черного квадрата размерами 50×50 пикселей  
square.beginFill(0x000000, 100);  
square.lineTo(0, 50);  
square.lineTo(50, 50);  
square.lineTo(50, 0);  
square.endFill();
```

Метод `beginFill` используется для начала рисования векторной графики. Методы `beginFill` и `endFill` необходимы при рисовании с помощью ActionScript. Однако имеются и другие методы, например `lineTo`, который здесь фактически используется для “рисования”.

5. Теперь назначьте событие `enterFrame` видеоклипу `square` при помощи следующего кода:

```
// присвоение события enterFrame для перемещения квадрата  
// по горизонтали вдоль рабочего поля и возвращения его  
// к началу при достижении правой границы рабочего поля  
square.onEnterFrame = function () {  
  
    if (this._x <= Stage.width) this._x += 10;  
    if (this._x > Stage.width) this._x = 0;  
  
}
```

Данный фрагмент является примером использования ActionScript для динамического назначения события `enterFrame` новому видеоклипу, также созданному посредством ActionScript. Эта методика является новинкой Flash MX.

6. Протестируйте фильм, выбрав команду Control⇒Test Movie, чтобы увидеть конечный результат. На экране отобразится черный квадрат размером 50×50 пикселей, перемещающийся поперек экрана и возвращающийся на исходную позицию после достижения границы рабочего поля (иллюстрация работы этой программы приведена на рис. БН.13).

В примере используется несколько новых методов и объектов, включая объект `Stage`, API-рисование и обработчики динамических событий.

Это простой пример, который демонстрирует, какой мощью теперь обладает язык ActionScript. Теперь, используя его новые возможности, можно легко и быстро выполнить задачи, которые невозможно было осуществить в предыдущей версии Flash.

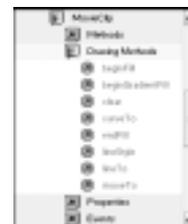


Рис. БН.12. Методы рисования во Flash MX являются частью объекта `MovieClip`

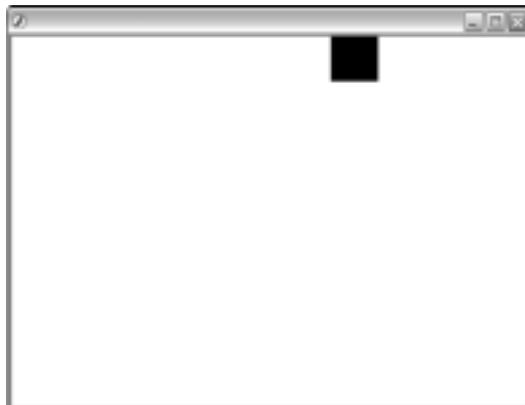


Рис. БН.13. Законченный фильм, воспроизводящийся в режиме тестирования

Резюме

- Основной целью Macromedia было создание более динамичной версии Flash. Action-Script является мощным инструментом, прекрасно подходящим для создания усовершенствованных Web-приложений.
- Во Flash MX появилось много удобных возможностей для дизайнеров. Благодаря доработкам временной шкалы Flash MX фильмы стали намного более управляемыми и презентабельными. Особенно стоит отметить папки слоев, новую команду **Distribute to Layers** (Распределить по слоям) и усовершенствованную команду **Break Apart** (Разделить).
- Благодаря расширению динамических возможностей и усовершенствованию многих объектов во Flash MX, разработчики смогут воспользоваться преимуществом данных нововведений для создания мощных приложений.
- В этой вступительной главе приводятся простые примеры для демонстрации лишь некоторых новых возможностей Flash. Однако во Flash MX появилось также множество других возможностей, таких как взаимодействие с Web-камерами, использование микрофонов, а также возможность создания материала, подобного файлам cookie. Это позволяет использовать Flash для создания Web-приложений.
- В остальных главах этой книги подробно описаны упомянутые здесь (а также другие) технологии Flash.

Знакомство с Flash MX

История развития Macromedia Flash началась в 1997 году с появления таких простейших программ, как, например, FutureSplash, которая впоследствии превратилась в мощный инструмент для создания широкого диапазона медиакомпонентов. С каждой новой выпущенной версией этой программы возможности создания графического и динамического содержимого, предназначенного для использования на Web-узлах и за их пределами, постоянно увеличивались. Открывая дополнительные возможности, расширяя набор инструментальных средств и привлекая новых пользователей, Macromedia способствовала развитию Flash и расширению масштабов ее использования. Никогда еще прежде приложение Flash не включало так много новых функциональных особенностей в одном выпуске.

В этой главе мы постараемся рассказать, что представляет собой приложение Flash MX, и рассмотрим множество функциональных возможностей этой программы, которые разработчик может использовать в своей работе. Кроме того, мы рассмотрим, насколько Flash может заменить или расширить возможности других программ, с которыми вы, должно быть, уже знакомы.

Flash-фильмы можно просматривать несколькими способами. Наиболее распространенным является просмотр с помощью Web-браузера. В этом случае Flash-фильм может загружаться либо как часть содержимого HTML-страницы, либо как Web-узел в целом, представляющий собой один основной Flash-фильм (.swf-файл), использующий несколько небольших Flash-фильмов (.swf-файлов) в виде загружаемых элементов. Кроме того, для просмотра фильмов может быть использована программа Flash Player. Это автономное приложение (известное также как *проектор*), которое может использоваться для просмотра фильмов, не требуя использования Web-браузера или подключения внешних аппаратных средств. Этот метод, как правило, используется для просмотра Flash-фильмов, записанных на компакт-дисках, дискетах или других носителях автономного формата.

ГЛАВА

1

В этой главе...

Мир Flash MX

Многогранность
возможностей Flash MX

Резюме



Для получения дополнительной информации относительно проекторов и автономных приложений см. главу 23, “Работа с проигрывателем Flash и проектором”.

Мир Flash MX

За последние годы возможности и дизайн Flash получили существенное развитие. Разработчики Flash совершенствуют технологию работы программы с каждым новым ее выпуском, осваивая новые области ее применения, о чем свидетельствуют тенденции развития программы. Современная версия программы Flash MX представляет собой достаточно развитый и весьма популярный среди пользователей инструмент разработки авторских проектов.

Flash MX — не единственный продукт Macromedia. Компания выпустила целое семейство программ MX: Flash MX, Dreamweaver MX, Fireworks MX и ColdFusion MX. Наряду с FreeHand 10, эти приложения составляют Macromedia Studio MX. Интерфейсы пользователя программ Flash MX, Dreamweaver MX и Fireworks MX практически идентичны, каждое из приложений имеет инспектор свойств, закрепляемые наборы панелей и специальные инструментальные средства для интегрирования одного программного продукта в другой.



Для получения дополнительной информации о каждой из этих программ и их возможностях по расширению работы с Flash см. часть IX, “Расширение возможностей Flash”, этой книги.



В момент написания этой книги, компания Macromedia объявила о выпуске новой технологии передачи данных, обеспечивающей двустороннюю передачу аудио- и видеoinформации, а также передачу данных, поступающих в режиме реального времени с помощью приложения Flash Player 6. Но эта технология уже не является частью Studio MX.

Несмотря на то что обширный диапазон использования Flash уже и так говорит сам за себя, безупречный интерфейс и мощные функциональные возможности Flash MX, несомненно, принесут с собой новую волну содержательных, интересных, ну и, наконец, просто забавных и очаровательных новшеств.

Конечно, методов использования Flash может быть гораздо больше, нежели прилагательных, которыми их можно описать, однако здесь мы рассмотрим только несколько наиболее распространенных примеров.

- Формы для сбора информации от пользователя и динамической загрузки информации, выбранной пользователем в процессе интерактивной работы.
- Видео-портфолио (авторские коллекции видеоматериалов), использующее встроенные в Flash MX возможности импорта видеoinформации и динамической загрузки содержимого.
- Заставки со встроенной функцией определения загрузки.
- Полезные Web-утилиты, такие как средства поиска
- Устойчиво работающие виртуальные комнаты для бесед (chat room), основанные на XML и технологии двунаправленных каналов сервера.
- Аудиоинтерфейс, динамически предоставляющий пользователю запрашиваемые музыкальные файлы и использующий средства MX для загрузки формата MP3.

- Интерактивные концептуальные художественные эксперименты, включая работу с несколькими пользователями, 3D-графику или запись и воспроизведение взаимодействия между пользователем и системой.
- Реализация решений в области электронной коммерции, полностью базирующихся на использовании Flash и серверной технологии.
- Загрузка альтернативного содержимого или изменение атрибутов фильма на основе тестирования системных возможностей с использованием Flash MX.
- Проекторы, используемые для создания слайдов презентаций в стиле PowerPoint, размещаемых на компакт-диске или другом альтернативном устройстве хранения информации.
- Передача качественной мультимедиа, рекламы или титров.
- Оптимизированная мультимедиа для Web-узлов или портативных устройств, таких как, например, сотовые телефоны или карманные ПК (Pocket PC).
- Интерфейс, обеспечивающий доступность информации для людей с ограниченными физическими возможностями с помощью специальной программы чтения экрана.
- Flash-фильмы, экспортируемые исключительно для использования в цифровых видео-проектах, требующих спецэффектов и композиции.

Безусловно, это неполный список направлений использования программы Flash. С каждым новым выпуском программы он постоянно расширяется. Однако уже из этого перечня очевидна безграничность возможностей Flash.

Топография Flash MX

Прежде чем приступить к созданию в приложении Flash интерактивного проекта, ознакомьтесь со структурой рабочей среды Flash. Даже если вы уже знакомы Flash 5, это не будет лишним. Дело в том, что в новый выпуск программы — Flash MX, компанией Macromedia было добавлено множество новых функциональных возможностей интерфейса. Поэтому мы убедительно рекомендуем изучить новый интерфейс программы и прочитать всю книгу от начала до конца.



Для получения дополнительной информации относительно нового интерфейса Flash MX, настройки рабочего пространства пользователя и оптимизации производственного процесса см. главу 4, “Основы интерфейса”.

Кроме того, прежде, чем приступить к созданию интерактивного проекта во Flash, его необходимо спланировать. Игнорирование процесса предварительного планирования значительно удлинит и усложнит ваш производственный процесс. Чем лучше будет выглядеть проект фильма на бумаге, тем лучше он окажется в заключительном исполнении.



Более подробно основы планирования интерактивных проектов Flash будут обсуждаться в главе 3, “Планирование Flash-проектов”. Кроме того, расширенные описания концепций планирования будут рассматриваться в главах, посвященных производственному процессу при создании конкретных проектов Flash. Более подробно практическое применение стратегий планирования проектов Flash будет обсуждаться в главе 20, “Создание первого проекта во Flash MX”.

В этой книге общая информация о принципах интерактивного планирования представлена в первых главах. Последующие главы содержат пошаговые описания реальных проектов, которые позволят читателю объединить изученную теорию и советы и приступить к разработке конкретных проектов.



В главе 32, “Создание во Flash MX узла с коллекцией художественных работ”, описан процесс создания Web-узла, включающего различные источники изображений. В главе 33, “Создание игры”, будет рассмотрена логическая схема, необходимая для разработки и написания сценария занимательной игры.

В процессе создания любого Flash-проекта вы создаете два основных типа файлов: документы Flash (.fla-файлы) и фильмы Flash (.swf-файлы). Эти форматы файлов будут рассмотрены несколько позже.

Типы файлов во Flash MX

К первому, основному типу файлов Flash относятся документы Flash (.fla-файлы), архитектура которых обеспечивает авторскую среду разработки для создания проектов любых размеров. В дальнейшем в пределах этой среды, содержимое проекта может быть организовано в сцены, а сцены упорядочены в выполняющийся определенным образом цикл. Слои документа обеспечивают разделение графики в пределах каждой сцены. Более того, направляющие (Guide) или маскирующие (Mask) слои могут помочь при рисовании или даже создании спецэффектов. На временной шкале (в окне Timeline) отображаются ключевые кадры, последовательности кадров, полученные с помощью операции автоматического заполнения промежуточных кадров с интерполяцией движения и с изменения формы, метки и комментарии. Библиотека (окно Library) позволяет хранить все созданные во Flash символы, например графику, шрифты, звук и видео, анимированные элементы и другие компоненты Flash-проекта. Библиотека может совместно использоваться несколькими фильмами как во время разработки проекта, так и во время его выполнения.



Компоненты Flash MX представляют собой эволюционировавшее средство Smart Clip, которое многие разработчики использовали для повторного использования кода ActionScript. Более подробно поставляемые с Flash MX компоненты будут рассматриваться в главе 29, “Использование компонентов”. В других частях книги мы также рассмотрим некоторые пользовательские компоненты, используемые в различных проектах Flash.

Документы Flash



Обратите внимание: во всех примерах, приведенных в этой книге, используются Flash-документы, которые представляют собой .fla-файлы, созданные во Flash MX при выборе команды меню File⇒Save (Файл⇒Сохранить) или File⇒Save as (Файл⇒Сохранить как). В отличие от некоторых графических приложений, таких как, например, Macromedia FreeHand или Adobe Illustrator, расширение файла Flash-документов не отражает версию программы, используемую для создания файла. К примеру, как в версии приложения Flash 5, так и в версии Flash MX Flash-документы будут сохранены как файлы с расширением .fla. Однако вы не сможете открыть документы, созданные в более поздней версии приложения, используя более раннюю версию. Документы Flash не могут быть просмотрены в приложении Flash Player, поэтому нет смысла помещать эти файлы на свой Web-сервер. Просто храните версию (и резервную копию!) вашего Flash-документа.



Flash MX позволяет сохранять документы, созданные в версии приложения Flash MX (.fla-файлы) в формате документов версии Flash 5 (.fla-файлов). Для этого выберите команду меню File⇒Save As, а затем в раскрывающемся списке Save as type (Тип файла) выберите параметр Flash 5 Document (Документ Flash 5), после чего вы сможете открывать этот Flash документ в приложении Flash 5.

На рис. 1.1 приведен пример Flash-документа, составленного из отдельных сцен, которые содержат ключевые кадры, описывающие все изменения на *рабочем поле* (*Stage*). На рисунке вы не увидите эффективности совместного использования Flash-библиотек несколькими документами, а также загрузки других Flash-фильмов в основной (“родительский”) Flash-фильм с помощью действия `LoadMovie()` или создания интерактивных элементов с помощью методов написания сценариев.

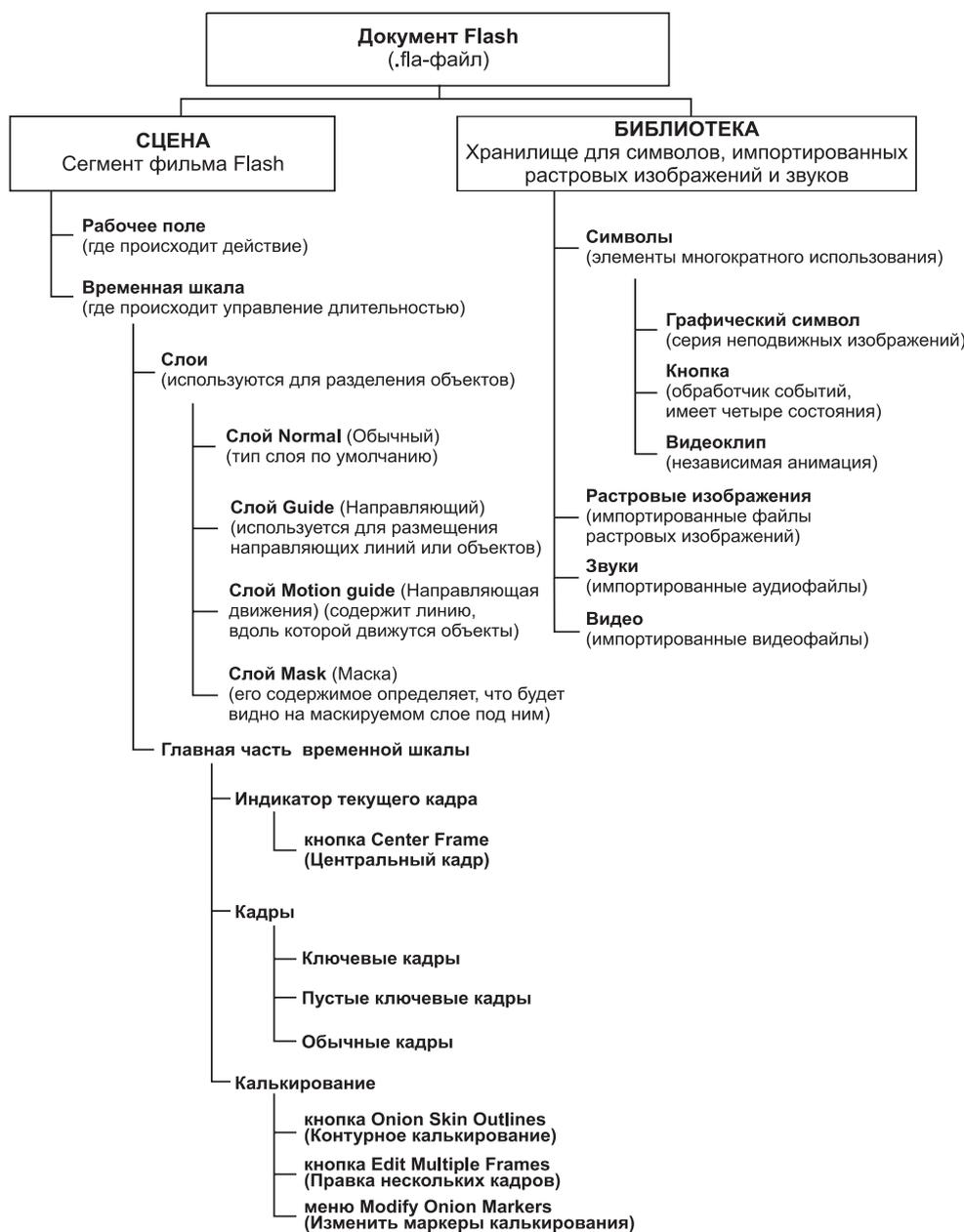


Рис. 1.1. Элементы документа Flash (.fla-файла) в среде разработки проекта



Для получения более подробной информации относительно совместного использования и загрузки Flash-фильмов, см. часть VII, "Освоение ActionScript" этой книги. В этой части приведена подробная информация о разработке сценариев с помощью ActionScript.

Flash-фильмы



Когда вы публикуете или тестируете Flash-документ, приложение Flash MX создает файл Flash-фильма с расширением `.swf`. Этот формат файла представляет собой оптимизированную версию Flash документа, сохраняющего только те элементы файла проекта, которые фактически используются. После этого Flash-фильмы загружаются на Web-сервер разработчика, где они, как правило, интегрируются в HTML-документы, предназначенные для просмотра другими пользователями Web-страниц. Вы можете защитить законченные Flash-фильмы от возможности простого импортирования или редактирования в среде разработки другими пользователями.



Имейте в виду, что опция `Protect from import` (Запретить импорт) диалогового окна `Publish Settings` (Параметры публикации) не предотвращает возможности извлечения из вашего Flash-фильма художественных работ, символов, звуков и кода ActionScript с помощью специальных утилит других производителей. Для получения дополнительной информации читайте главу 21, "Публикация Flash-фильмов".

Большая часть исходной информации, содержащейся во Flash-документе (`.fla`-файле), отбрасывается при экспорте его в виде Flash-фильма (`.swf`-файла), поскольку программа пытается сделать файл Flash-фильма как можно более маленьким. При экспорте фильма все задействованные исходные элементы остаются, однако слои документа сливаются в один, и фильм выполняется на одной временной шкале в порядке, который был установлен в документе Flash. Фактически вся информация, первоначально присутствующая в файле, так или иначе будет оптимизирована, однако любые элементы библиотеки, которые не были использованы во Flash-фильме, экспортироваться не будут. Содержимое библиотеки загружается и хранится в первом из кадров, в котором оно используется. Для оптимизации Flash-фильма многократно используемые в фильме элементы сохраняются в файле только один раз. В дальнейшем программа просто использует ссылки на место их хранения. Растровые изображения и звуковые файлы также могут быть сжаты, для чего используются различные настройки параметров качества этих файлов.



Сегодня Flash-фильмы, созданные с помощью шестой версии программы Flash, могут быть оптимизированы с помощью специализированной опции `Compress Movie` (Сжать фильм), которая находится во вкладке *Flash* диалогового окна `Publish Settings` (его можно открыть с помощью команды `File⇒Publish Settings`). Применяв эту опцию, вы увидите, насколько принципиально изменится размер файла, содержащего фильм, использующий значительное количество программного кода ActionScript.

На рис. 1.2 графически представлены характеристики формата Flash-фильма (`.swf`-файла).

Знакомство с Flash Player 6

Различие между условными обозначениями, принятыми во встраиваемом модуле Flash Player и непосредственно в самой программе Flash, предназначенной для создания Flash-фильмов, может показаться пользователю несколько непонятным. Дело в том, что в последнем выпуске проигрывателя Flash Player 6 компания Macromedia использовала в конце имени программы номер ее версии, отказавшись от принятого в предыдущих версиях со-

глашения об именовании. Одна из причин, по которой вместо символов "MX" (которые используются в названии программного обеспечения, предназначенного для создания файлов) в названии проигрывателя был использован номер версии, заключается в том, что стандартный последовательный номер требуется для работы программ, автоматически определяющих наличие встраиваемого модуля проигрывателя Flash на компьютере пользователя.

Сегодня Flash Player позволяет сделать содержимое Flash-фильмов более доступным для людей с ограниченными физическими возможностями. Для этого в приложение была включена технология MSAA для поддержки вспомогательных технологий, таких как, например, программы чтения экрана. Разработчики компании Macromedia поместили XML-технологии непосредственно в проигрыватель, что существенно ускорило прохождение данных. Значительно улучшено воспроизведение на компьютерах типа Macintosh, в то же время размер загружаемого файла проигрывателя, несмотря на дополнения, остался небольшим.



Рис. 1.2. Общий обзор формата Flash-фильма (.swf)



Более подробно вопросы определения наличия приложения Flash Player будут рассмотрены позже в главе 22, "Интеграция Flash-содержимого с HTML".



Руководства по созданию содержимого для Flash Player 6 приведены в главе 20, "Создание первого проекта во Flash MX".

Существует также несколько других способов, с помощью которых вы можете просмотреть или отобразить Flash-фильм или его отдельные части. Начиная с версии Flash 4 средство публикации предлагает опции, позволяющие экспортировать Flash-фильмы или отдельные части фильмов либо в формат цифрового видео QuickTime, либо в векторный формат QuickTime Flash, либо в формат Animated GIF (анимированный GIF). Части фильмов могут также экспортироваться как последовательности индивидуальных растровых рисунков или как векторные файлы. В этих форматах могут также экспортироваться и отдельные кадры фильма.

Многогранность возможностей Flash MX

Приложение Flash — это в некоторой степени гибридное приложение, которое не похоже ни на одну другую программу. На первый взгляд может показаться, что это сочетание Web-ориентированной программы обработки растровых изображений и программы создания векторной графики типа Macromedia FreeHand или Adobe Illustrator. Однако, несмотря на то что Flash и представляет собой гибрид, это приложение способно на нечто гораздо большее. Flash — это также среда создания интерактивных мультимедийных продуктов и сложная программа анимации, предназначенная для создания широкого диапазона мультипликаций — от простых орнаментов для Web-страниц до оптимизированных для ускорения загрузки мультипликаций. Но и это еще не все. Это также базовый узел для мощного и легко адаптируемого языка написания сценариев.

С течением времени из ограниченного метода перетаскивания разблокированной анимации ActionScript превратился в достаточно развитый объектно-ориентированный язык программирования, весьма похожий на JavaScript. Язык ActionScript программы Flash может работать совместно с XML (Extensible Markup Language — расширяемый язык гипертекстовой разметки), HTML и многими другими приложениями и составляющими Web-страниц. Сегодня содержимое Flash может быть интегрировано во многие сервер-ориентированные технологии. К примеру, Flash Player обеспечивает встроенную поддержку динамически загружающихся изображений, файлов формата MP3, Flash-фильмов и других данных. Flash может свободно работать с серверами ColdFusion MX и XML, что позволяет распространять оптимизированное, динамически обновляемое интерактивное содержимое.

И все-таки, почему этот гибрид под названием Flash представляет собой такое ценное программное обеспечение? Это тот вопрос, на который придется ответить вам как разработчику. На самом деле мы надеемся, что вы научитесь работать с этим приложением и порадуете нас своими произведениями. Ведь именно для этого мы и написали эту книгу: рассказать о возможностях этого инструмента в надежде, что вы возьмете его в руки и начнете творить.

Таким образом, если Flash — это приложение-гибрид и если это приложение способно почти на все, то работу с этим мощным инструментом разумно начать с вопроса: из каких элементов состоит этот гибрид? И если они независимы друг от друга, то как описать их возможности? На эти вопросы мы постараемся ответить в этой главе.

Программа обработки растровых изображений

Программа Flash обладает ограниченными возможностями редактирования растровых изображений. Эту часть приложения Flash лучше описать как программу для обработки растровых изображений. Растровые рисунки состоят из множества точек — сетки отдельных пикселей. Положение (и цвет) каждой точки должно храниться в памяти, а это приводит к интенсивному использованию ресурсов компьютера и большим размерам файлов. Еще одним недостатком растровых изображений является то, что их нельзя масштабировать без потери качества (четкости). Причем при увеличении размера изображения этот недостаток проявляется сильнее, чем при уменьшении. Из-за этих двух недостатков — большого размера файла и ограничения по изменению размера — растровые изображения не являются идеальным решением для Web. Однако растровый формат совершенно необходим для получения изображений фотографического качества, поскольку он обеспечивает лучшее качество изображения и меньший размер файла, чем векторное изображение такой же сложности.

Программа обработки векторной графики

Приложение Flash главным образом является программой для создания векторной графики, возможности которой можно сравнить с программами Macromedia Freehand или Adobe Illustrator. Программа векторной графики при создании графического изображения полагает-

ся не на отдельные пиксели, а определяет точки фигуры, которые описываются координатами. Линии, которые соединяют эти точки, называются *траекториями*, а векторы в каждой точке описывают изменение направления траектории. Поскольку эта схема использует математическое представление, здесь существует два основных преимущества: содержимое в векторном виде имеет более компактный вид, и масштаб изображения можно изменять без ущерба для качества. Эти преимущества имеют большое значение для Web.

Программа для векторной анимации

Компонент программы Flash, служащий для создания векторной анимации, отличается от всех существовавших ранее программ. Хотя Flash может управлять растровыми изображениями, ее “родной” формат — векторный. Поэтому, в отличие от других анимационных и медиапрограмм, при создании фильма Flash полагается на добротный и надежный векторный формат. Вместо того чтобы хранить в каждом кадре мегабайты информации о пикселях, программа Flash хранит компактное векторное описание каждого кадра. В то время как программы для анимации растровых изображений (например, QuickTime от Apple) борются за отображение каждого растрового рисунка с нужной скоростью, Flash быстро формирует векторное изображение, предъявляя меньше требований к пропускной способности Internet-соединения и затрачивая меньше ресурсов компьютера пользователя. Это имеет огромное преимущество при передаче Flash-анимации и другого графического содержимого в Web.

Видеокompрессор

Последняя версия приложения Flash содержит встроенный видеокompрессор — кодек Sorenson Spark. Благодаря этому вы можете импортировать видеофайлы из других источников непосредственно в документы Flash MX (.fla-файлы). Но наиболее существенно то, что Flash Player 6 также включает кодек Sorenson Spark! Это означает, что теперь встраиваемый модуль проигрывателя Flash Player 6 может рассматриваться как один из самых маленьких в мире модулей для просмотра видеоданных. Сегодня для того чтобы просмотреть Flash-фильм, пользователю уже не обязательно устанавливать QuickTime от Apple, RealOne от RealSystems или Microsoft Windows Media Player. В отличие от приложения Director Shockwave, которое воспринимает видеоматериалы, но при этом для их воспроизведения требует инсталляции Apple QuickTime, Flash Player 6 воспроизводит видеофайлы, не прибегая к использованию других специальных программ.

Вы можете управлять видеосодержимым и/или включать его в ваши продукты. Кроме того, вы можете использовать приложение Flash MX для прямого экспорта высококачественного видеосодержимого в предназначенную для телевещания видеопродукцию.



Для получения дополнительной информации относительно этой новой интересной особенности Flash MX обратитесь к главе 17, “Внедрение видеоматериалов”. Там же будет обсуждено совершенно новое приложение — Sorenson Squeeze, разработанное для создания видеосодержимого самого высокого качества.



Более подробно переход от Flash-анимации к высококачественному видео будет обсуждаться в главе 14, “Экспортирование анимации”.

Программа создания мультимедиа

Если программа обработки векторной графики — сердце Flash, то программа создания мультимедиа (или *среда создания*) — ее тело. Документы Flash (.fla-файлы) могут содержать множество медиакомпонентов, включая звук, фотографии, анимацию и видео. Кроме

того, Flash представляет собой мощный инструмент, предназначенный для создания истинно интерактивного содержимого, поскольку позволяет разработчику добавлять команды (ActionScript) для динамического управления воспроизведением фильма (.swf-файла). Независимо от того, создаете ли вы простые системы меню или экспериментируете над интуитивно понятными пользовательскими интерфейсами, Flash-содержимое может быть разработано так, что оно будет распознавать команды пользователя и реагировать на его действия.

Программа упорядочения анимации

Многие мультимедийные программы обладают компонентом для создания упорядоченных последовательностей содержимого, например анимации, и Flash — не исключение. Но в программе Flash контроллер анимации является ключевым элементом программы. Управление содержимым Flash-проекта, как статическим, так и анимированным, осуществляется с помощью окна Timeline (Временная шкала). Окно временной шкалы содержит две области, которые предназначены для организации содержимого фильма во времени (с помощью кадров) и в визуальном пространстве (с помощью слоев).

Слои и папки слоев позволяют разработчику следить за содержимым, помещенным в документ Flash. Вы можете управлять видимостью каждого слоя независимо друг от друга, что позволяет легко изолировать каждый созданный вами элемент в отдельности. Все слои документа, начиная с переднего плана и заканчивая задним планом, можно просмотреть в каждом кадре временной шкалы — элементы, расположенные на верхних слоях, перекрывают элементы в нижних слоях. Количество элементов, размещаемых в каждом отдельном слое, не ограничено, однако чем больше их количество, тем сложнее управлять ими в пределах слоя. Имейте в виду, что в пределах одного слоя документа все несгруппированные векторные линии и фигуры всегда будут помещаться на задний план, в то время как растровые изображения, текст, сгруппированные элементы, а также экземпляры символов будут расположены на переднем плане самого слоя.



В версию приложения Flash MX была включена возможность создания папок слоев, что весьма полезно для организации проектов, состоящих из множества отдельных элементов.



Для получения более подробной информации относительно среды приложения Flash MX обратитесь к главе 4, «Основы интерфейса». Процессы создания художественных работ и управления группами и символами более подробно рассматриваются в главе 5, «Рисование в приложении Flash», и в главе 6, «Символы, экземпляры и библиотека».

Основными компонентами структуры, создающей иллюзию движения в любом Flash-фильме, является серия кадров. Каждый кадр документа отображается в определенный момент времени. Управляя перемещением через эти кадры воспроизводящей головки (или указателя текущего кадра), разработчик может задавать скорость, продолжительность и порядок воспроизведения анимационной последовательности.

Flash предоставляет в распоряжение разработчика два способа создания анимации: покадрово (в этом случае все изменения в слоях каждого кадра задаются вручную) и с помощью операции автоматического заполнения промежуточных кадров (последний способ и сделал приложение Flash таким популярным инструментом создания анимации). При втором способе вручную задаются только длительность анимации (заданием числа кадров), а также начальные и конечные формы и взаиморасположение объектов (в ключевых кадрах), а все промежуточные значения генерируются (интерполируются) самой программой Flash. Используя два вида автоматического заполнения кадров: с *интерполяцией движения (Motion tween)* и с *интерполяцией изменения формы (Shape tween)*, вы легко можете преобразовывать векторные фигуры или изменять координаты объектов на *рабочем поле (Stage)*.



Более подробно различные способы создания Flash-анимации будут рассматриваться в части III, “Создание анимации и эффектов”, этой книги.

Кроме того, в пределах одного Flash-документа, разработчик может устанавливать целую серию отдельных сцен, каждая из которых является продолжением одной и той же основной временной шкалы; однако сцены могут быть переименованы или переупорядочены в любое время. Сцены проигрываются от первой до последней (если интерактивные команды Flash, известные под названием “действия”, не указывают на другое).



Этапы использования программного кода ActionScript с целью простого управления воспроизведением фильма будут рассматриваться в части V, “Добавление основных элементов интерактивности в фильмы Flash”, этой книги.

Программирование и клиентская часть базы данных

Две последние версии приложения Flash продемонстрировали обширный рост возможностей интеграции интерфейса Flash с ориентированными на сервер технологиями и динамическим загрузкой содержимого с использованием XML, ColdFusion и Generator. Этим улучшением Flash в значительной мере обязана разработке и развитию такого жизнеспособного языка программирования, как ActionScript. Эти возможности позволили Flash работать в качестве клиентской части серверной базы данных для сложных интерактивных приложений (предназначенных, например, для обеспечения торговли в интерактивном режиме), проигрывателя MP3-файлов, канала, передающего сводку погоды, и других возможностей, которые обычно не ассоциируются с анимационной программой.

С появлением Flash MX в руках разработчика появилось гораздо больше возможностей. Одна из самых потрясающих новых функциональных возможностей этой версии — включение компонентов с возможностями, которые обеспечивались ранее только программой Generator или другими приложениями от независимых производителей. Теперь во время выполнения вы можете свободно загружать во Flash-фильмы файлы формата JPEG и MP3 и нет необходимости использовать для этого Generator. Кроме того, благодаря программным улучшениям, внесенным во Flash Player 6, существенно усовершенствована поддержка XML-документов. Теперь передача любой информации занимает гораздо меньше времени, чем это требовалось в версии приложения Flash 5.

Итак, новая версия программы — Flash MX — претерпела существенные изменения. Надеемся, что множество улучшений, внесенных как в среду программирования, так и в функциональные возможности, будут оценены по достоинству не только новичками, но и опытными пользователями Flash. Ниже приведено лишь несколько новых возможностей программирования, которые вы найдете во Flash MX.

- Несмотря на то что улучшения языка программирования ActionScript, внесенные в версию программы Flash MX, по сравнению с версией Flash 5 не столь очевидны, как прыжок из Flash 4 во Flash 5, несколько радикальных изменений все же существует. Эти изменения обеспечивают продвижение частного языка ActionScript в направлении к принятию его в качестве стандартного объектно-ориентированного языка программирования. Одна из новых возможностей, которая обязательно понравится как начинающим, так и продвинутым программистам, — это понятный программный интер-

фейс приложения, который позволяет разработчикам создавать сложную графику непосредственно с помощью программного кода.

- Flash Player 6 обеспечивает более совершенную поддержку XML-документов. Более полно интегрирована с проигрывателем и языковая поддержка, что позволяет улучшить качество выполнения синтаксического анализа, избежать выполнения множества ненужной работы, что часто требовалось в версии приложения Flash 5 Player.
- Проверка системных возможностей теперь встроена непосредственно в ActionScript.
- Новый объект `LoadVars` делает выполнение действия `loadVariables()` более совершенным, расширяя и упрощая процесс отправления и получения данных с сервера.
- Кнопки теперь могут быть целевым объектом действия.
- Совместно используемые объекты могут применяться подобно файлам `cookie`, позволяя разработчику сохранять данные на компьютере конечного пользователя.
- В язык ActionScript версии Flash MX добавлена новая модель объекта и событий. Объект отслеживает и реагирует на определенные события, например, такие как ввод пользователем данных.
- Редактор языка ActionScript, известный также как панель **Actions** (Действия), дополнен новыми функциональными возможностями, включая подсказки кода, точки прерывания и управление форматированием.

Резюме

- Flash представляет собой гибкую программу, которая объединяет множество ключевых инструментальных средств, предназначенных для создания элементов мультимедиа. Использование возможностей Flash в комплексе с другими программами и языками позволяет создавать улучшенное Web-содержимое и более сложные приложения.
- Содержимое Flash можно найти не только на Web-страницах. Оно может использоваться для создания компакт-дисков, графики, предназначенной для передачи по телевидению, автономно работающих интерфейсов и бизнес-презентаций.
- Flash MX — это многогранное приложение, с помощью которого разработчик может создавать широкий диапазон интерактивных продуктов, предназначенных для поддержки всего многообразия устройств доступа к Web-страницам, с помощью которых пользователи могут обращаться к сети Internet.
- И, наконец, внимательное и точное планирование Flash-проектов, несомненно, экономит не только время разработчика, но и потребует меньших усилий для его создания.