

Введение

Платформа .NET принесла с собой небольшую лавину новых технологий. Появился совершенно новый способ написания веб-приложений (ASP.NET), совершенно новый способ подключения к базам данных (ADO.NET), новые безопасные к типам языки (C# и VB.NET) и управляемая исполняющая среда (CLR). Не последнее место занимала и технология Windows Forms — библиотека классов для построения Windows-приложений.

Хотя Windows Forms — зрелый и полнофункциональный набор инструментов, он был тесно привязан к некоторым частям внутреннего устройства Windows, которые не слишком изменились за последние 10 лет. Что более существенно, при создании визуального представления стандартных пользовательских интерфейсных элементов, таких как кнопки, текстовые поля, флажки и т.п., Windows Forms полагается на Windows API. В результате его ингредиенты мало поддаются настройке и изменениям.

Например, чтобы построить стилизованную блестящую кнопку, придется создать специальный элемент управления и нарисовать каждый аспект этой новой кнопки (во всех разных состояниях), используя низкоуровневую модель рисования. Хуже того, обычные окна разрезаются на отдельные области, и каждому элементу управления отводится собственная такая область. В результате не существует такого хорошего способа рисования в одном элементе управления (например, эффекта свечения под кнопки), чтобы он распространялся на область, принадлежащую другому элементу. И не стоит даже думать о создании анимационных эффектов вроде вращающегося текста, мерцающих кнопок, сворачивающихся окон или активных предварительных просмотров, потому что каждая деталь должна быть нарисована вручную.

В Windows Presentation Foundation (WPF) эта ситуация изменилась за счет ввода модели с совершенно другим устройством. Хотя платформа WPF включает знакомые стандартные элементы управления, она рисует каждый текст, контур и фон *самостоятельно*. В результате WPF может предоставить намного более мощные средства, которые позволяют изменить визуализацию любой части экранного содержимого. С помощью этих средств можно изменить стиль обычных элементов управления, таких как кнопки, часто даже без написания кода. Кроме того, можно применять трансформации объектов для вращения, растяжения, масштабирования и сдвига любой части пользовательского интерфейса, и даже использовать встроенную систему анимации WPF, чтобы делать все это прямо на глазах пользователя. И поскольку механизм WPF визуализирует содержимое окна как часть одной операции, он может обрабатывать неограниченное количество слоев перекрывающихся элементов, даже имеющих нерегулярную форму и частичную прозрачность.

В основе WPF лежит мощная инфраструктура, основанная на DirectX — API-интерфейсе графики с аппаратным ускорением, который обычно используется в современных компьютерных играх. Это означает возможность применения развитых графических эффектов, не платя за это производительностью, как это было в Windows Forms. Фактически даже становятся доступными такие расширенные средства, как поддержка видеофайлов и трехмерное содержимое. Используя эти средства (при наличии хорошего инструмента графического дизайна), можно создавать бросающиеся в глаза пользовательские интерфейсы и визуальные эффекты, которые были просто невозможны в Windows Forms.

Хотя новейшие средства видео, анимации и 3-D часто привлекают максимум внимания в WPF, важно отметить, что WPF можно применять для построения обычных Windows-приложений со стандартными элементами управления и привычным внешним видом. Фактически использовать стандартные элементы управления в WPF так же легко, как и в Windows Forms. Более того, WPF расширяет средства, адресованные именно бизнес-разработчикам, включая значительно усовершенствованную модель привязки данных, набор классов для печати содержимого и управления очередями печати, а также средства работы с документами для отображения огромных объемов форматированного текста. Доступна даже модель для построения приложений на основе страниц, которые гладко работают в Internet Explorer и могут запускаться с веб-сайта — и все это без привычных предупреждений о безопасности или надоедливых приглашений к установке.

Вообще говоря, WPF комбинирует лучшее из мира Windows-разработки с новейшими технологиями для построения современных, графически развитых пользовательских интерфейсов. Хотя приложения Windows Forms будут еще жить долгие годы, разработчикам, приступающим к новым проектам Windows, сначала стоит обратить внимание на WPF.

Об этой книге

Эта книга представляет собой углубленное руководство по WPF для профессиональных разработчиков, знакомых с платформой .NET, языком C# и средой разработки Visual Studio. Опыт работы с предыдущими версиями WPF не обязателен, хотя новые средства в книге специально выделены во врезках “Что нового?” в начале каждой главы.

Книга предлагает полное описание каждого из основных средств WPF — от XAML (языка разметки, используемого для определения пользовательских интерфейсов WPF) до трехмерного рисования и анимации. По ходу чтения вы ознакомитесь с кодом, который включает работу с другими средствами .NET Framework, такими как классы ADO.NET, которые служат для запросов к базе данных. Эти средства здесь не рассматриваются. За дополнительной информацией о средствах .NET, которые не являются специфичными для WPF, обращайтесь к соответствующим книгам.

Обзор глав

Эта книга включает в себя 33 главы. Если вы только начинаете знакомство с WPF, лучше читайте их по порядку, поскольку более поздние главы опираются на приемы, продемонстрированные в ранних.

Ниже приводятся краткие описания всех глав.

Глава 1. Введение в WPF. Описана архитектура WPF, внутренние механизмы DirectX, а также новая, независимая от устройства система измерения, которая автоматически изменяет размеры пользовательских интерфейсов.

Глава 2. XAML. Рассматривается стандарт XAML, который используется для определения пользовательских интерфейсов. Будет показано, зачем он был создан и как работает, а также, как создавать базовые окна WPF с помощью различных подходов к кодированию.

Глава 3. Компоновка. Дается углубленное представление панелей компоновки, которые позволяют организовать элементы в окне WPF. Будут рассматриваться различные стратегии компоновки и строится некоторые распространенные типы окон.

Глава 4. Свойства зависимости. Описано использование в WPF свойств зависимости для обеспечения поддержки таких ключевых средств, как привязка данных и анимация.

Глава 5. Маршрутизируемые события. Рассматривается использование в WPF маршрутизации событий с пузырьковым распространением и туннелированием через элементы пользовательского интерфейса. В главе также содержится описание базового набора событий мыши, клавиатуры и сенсорных панелей, поддерживаемых всеми элементами WPF.

Глава 6. Элементы управления. Описаны элементы управления, знакомые каждому разработчику Windows, такие как кнопки, текстовые поля и метки, и их воплощение в WPF.

Глава 7. Класс Application. Рассматривается модель приложений WPF. Будет показано, как создавать приложения одного экземпляра и приложения WPF, основанные на документах.

Глава 8. Привязка элементов. Объясняется привязка данных в WPF. Будет показано, как привязать объекты любого типа к пользовательскому интерфейсу.

Глава 9. Команды. Описана модель команд WPF, которая позволяет связывать несколько элементов управления с одинаковым логическим действием.

Глава 10. Ресурсы. Показано, как с помощью ресурсов встраивать двоичные файлы в сборку и многократно использовать важные объекты по всему пользовательскому интерфейсу.

Глава 11. Стили и поведения. Рассматривается система стилей WPF, которая позволяет применять набор общих значений свойств к целой группе элементов управления.

Глава 12. Фигуры, кисти и трансформации. Описана модель двухмерного рисования в WPF. Будет показано, как создавать фигуры, изменять элементы с помощью трансформаций и получать экзотические эффекты с помощью градиентов, укладки плиткой и изображениями.

Глава 13. Классы Geometry и Drawing. Более глубоко рассматривается двухмерное рисование. Будет показано, как создавать сложные пути, включающие дуги и кривые, а также эффективно использовать сложную графику.

Глава 14. Эффекты и класс Visual. Рассматривается программирование низкоуровневой графики. Будет показано, как применять эффекты в стиле Photoshop с помощью построителей текстур, вручную создавать растровые изображения, а также использовать визуальный уровень WPF для оптимизации рисования.

Глава 15. Основы анимации. Описана платформа анимации WPF, которая позволяет интегрировать динамические эффекты в приложение, используя прямолинейную декларативную разметку.

Глава 16. Расширенная анимация. Рассматриваются более сложные приемы анимации, такие как анимация ключевого кадра, анимация, основанная на пути, и анимация на основе кадров. Также предлагается детальный пример, демонстрирующий создание и управление динамической анимацией в коде.

Глава 17. Шаблоны элементов управления. Показано, как придать совершенно новый вид (и новое поведение) любому элементу управления WPF, подключая специализированный шаблон. Также описано, как с помощью шаблонов создавать приложения со сменными обложками.

Глава 18. Пользовательские элементы. Посвящена расширению существующих элементов управления WPF и созданию собственных. Будет предложено несколько примеров, включая основанный на шаблоне селектор цвета, переворачиваемую па-

нель, специальный контейнер компоновки и декоратор, который выполняет специальное рисование.

Глава 19. Привязка данных. Рассматривается, как извлекать информацию из базы данных, вставлять в специальные объекты данных и привязывать эти объекты к элементам управления WPF. Также описаны приемы повышения производительности больших связанных с данными списков посредством виртуализации, а также перехват ошибок редактирования за счет проверки достоверности.

Глава 20. Форматирование привязанных данных. Описаны некоторые трюки для превращения неформатированных данных в развитое экранное представление, включающее изображения, элементы управления и эффекты выбора.

Глава 21. Представления данных. Объясняется использование представления в окне, привязанном к данным, для навигации по списку элементов данных и применения фильтрации, сортировки и группировки.

Глава 22. Элементы управления ListView, TreeView и DataGrid. Дается экскурс по многофункциональным элементам управления WPF, включая ListView, TreeView и DataGrid.

Глава 23. Окна. Рассматривается работа окон в WPF. Будет показано, как создавать окна неправильной формы и использовать “стеклянные” эффекты Windows Vista. Кроме того, будет реализовано большинства средств Windows 7 за счет настройки списков часто используемых элементов в панели задач, миниатюр и налагаемых значков.

Глава 24. Страницы и навигация. Описано построение страниц в WPF и отслеживание хронологии навигации. Также будет показано, как строить основанные на браузере приложения WPF, которые могут быть запущены с веб-сайта.

Глава 25. Меню, панели инструментов и ленты. Посвящена командно-ориентированным элементам управления, таким как меню и панели инструментов. Также демонстрируется более современный интерфейс на основе свободного загружаемого элемента управления Ribbon.

Глава 26. Звук и видео. Описана поддержка мультимедиа в WPF. Будет показано, как управлять воспроизведением звука и видео, и как реализовать синхронизированные анимации и живые эффекты.

Глава 27. Трехмерная графика. Рассматривается поддержка рисования трехмерных фигур в WPF. Будет показано, как создавать, трансформировать и анимировать трехмерные объекты, а также, как помещать интерактивные двухмерные элементы управления на трехмерные поверхности.

Глава 28. Документы. Описана поддержка форматированных документов в WPF. Будет показано, как использовать потоковые документы для представления больших объемов текста в наиболее читабельном виде и фиксированные документы для отображения страниц, готовых к печати. Кроме того, рассматривается применение элемента управления RichTextBox для редактирования документа.

Глава 29. Печать. Представлена модель печати WPF, которая позволяет выводить текст и фигуры в печатный документ. Будет также показано, как управлять настройками страниц и очередями печати.

Глава 30. Взаимодействие с Windows Forms. Описаны способы комбинирования содержимого WPF и Windows Forms в пределах одного приложения и даже одного окна.

Глава 31. Многопоточность. Рассматривается создание отзывчивых приложений WPF, которые выполняют длительные задачи в фоновом режиме.

Глава 32. Модель дополнений. Показано, как создавать расширяемое приложение, которое может динамически обнаруживать и подгружать отдельные компоненты.

Глава 33. Развертывание ClickOnce. Рассматриваются вопросы, связанные с развертыванием приложений WPF с помощью технологии ClickOnce.

Что необходимо для чтения этой книги

Для того чтобы *запустить* приложение WPF, на компьютере должна быть установлена система Windows 7, Windows Vista или Windows XP с Service Pack 2. Также понадобится .NET Framework 4. Чтобы *создавать* приложения WPF (и открывать примеры проектов, включенные в эту книгу), необходима среда Visual Studio 2010, которая включает .NET Framework 4.

Существует еще одна возможность. Вместо использования любой версии Visual Studio строить и тестировать приложения WPF можно с помощью инструмента графического дизайна Expression Blend. В целом, Expression Blend предназначен для дизайнеров графики, которые большую часть времени занимаются созданием внешнего вида, в то время как Visual Studio — идеальная среда для работы программистов, пишущих код приложений. В книге предполагается применение Visual Studio. За дополнительными сведениями об Expression Blend следует обращаться к одной из специализированных книг. (Кстати, для создания приложений с помощью WPF 4 понадобится Expression Blend 4.)

Исходный код примеров

Исходный код всех рассматриваемых в настоящей книге примеров доступен на веб-сайте издательства по адресу <http://www.williamspublishing.com/>.

От издательства

Вы, читатель этой книги, и есть главный ее критик и комментатор. Мы ценим ваше мнение и хотим знать, что было сделано нами правильно, что можно было сделать лучше и что еще вы хотели бы увидеть изданным нами. Нам интересно услышать и любые другие замечания, которые вам хотелось бы высказать в наш адрес.

Мы ждем ваших комментариев и надеемся на них. Вы можете прислать нам бумажное или электронное письмо, либо просто посетить наш Web-сервер и оставить свои замечания там. Одним словом, любым удобным для вас способом дайте нам знать, нравится или нет вам эта книга, а также выскажите свое мнение о том, как сделать наши книги более интересными для вас.

Посылая письмо или сообщение, не забудьте указать название книги и ее авторов, а также ваш обратный адрес. Мы внимательно ознакомимся с вашим мнением и обязательно учтем его при отборе и подготовке к изданию последующих книг.

Наши координаты:

E-mail: info@williamspublishing.com

WWW: <http://www.williamspublishing.com>

Информация для писем из:

России: 127055, г. Москва, ул. Лесная, д. 43, стр. 1

Украины: 03150, Киев, а/я 152