

*Создай пользовательский интерфейс
в стиле Web 2.0 с помощью ASP.NET AJAX*

*Рассмотрены
пакеты Control Toolkit,
Core, Futures CTP и другие*

Программирование в

ASP.NET AJAX



O'REILLY®

Кристиан Вениц

Programming
ASP.NET AJAX

Christian Wenz

O'REILLY®

Программирование в ASP.NET AJAX

Кристиан Вениц



*Санкт-Петербург — Москва
2008*

Кристиан Венц
Программирование в ASP.NET AJAX

Перевод А. Киселева

Главный редактор
Зав. редакцией
Научный редактор
Редактор
Корректор
Верстка

А. Галунов
Н. Макарова
Б. Попов
Ю. Бочина
С. Минин
Д. Орлова

Венц К.

Программирование в ASP.NET AJAX. – Пер. с англ. – СПб: Символ-Плюс, 2008. – 512 с., ил.

ISBN-13: 978-5-93286-122-6

ISBN-10: 5-93286-122-3

Книга Кристиана Венца «Программирование в ASP.NET AJAX» – полное практическое введение в новую платформу Microsoft ASP.NET AJAX 1.0, которая предоставляет большие преимущества при разработке Ajax-приложений, схожие с теми, которые дает ASP.NET для разработки серверных сценариев. Вы узнаете, как с помощью технологий Ajax без особых затрат времени создавать профессионально выполненные динамические веб-страницы.

После общего обзора платформы и основ JavaScript и Ajax рассмотрена организация составляющих частей .NET, включая пакеты Extensions, Control Toolkit, Futures CTP и Microsoft AJAX Library. Также читателю предлагается исчерпывающая глава об элементе управления UpdatePanel, который позволяет организовать независимое обновление отдельных частей веб-страницы.

Насыщенная примерами, демонстрирующими ключевые аспекты платформы, эта книга идеально подходит не только разработчикам ASP.NET, стремящимся расширить свои возможности за счет использования Ajax, но и всем специалистам, кто интересуется данной платформой, независимо от того, какие технологии они используют в настоящее время.

ISBN-13: 978-5-93286-122-6

ISBN-10: 5-93286-122-3

ISBN 0-596-51424-7 (англ)

© Издательство Символ-Плюс, 2008

Authorized translation of the English edition © 2007 O'Reilly Media, Inc. This translation is published and sold by permission of O'Reilly Media, Inc., the owner of all rights to publish and sell the same.

Все права на данное издание защищены Законодательством РФ, включая право на полное или частичное воспроизведение в любой форме. Все товарные знаки или зарегистрированные товарные знаки, упоминаемые в настоящем издании, являются собственностью соответствующих фирм.

Издательство «Символ-Плюс». 199034, Санкт-Петербург, 16 линия, 7,
тел. (812) 324-5353, www.symbol.ru. Лицензия ЛП N 000054 от 25.12.98.

Налоговая льгота – общероссийский классификатор продукции
ОК 005-93, том 2; 953000 – книги и брошюры.

Подписано в печать 31.07.2008. Формат 70×100¹/16. Печать офсетная.

Объем 32 печ. л. Тираж 2000 экз. Заказ №

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

Оглавление

Предисловие	11
I. Основы	21
1. ASP.NET AJAX, Ajax и ASP.NET	23
ASP.NET AJAX и Ajax	23
ASP.NET AJAX и ASP.NET	25
Подготовка и установка ASP.NET AJAX	28
Структура и архитектура ASP.NET AJAX	34
Первый пример ASP.NET AJAX: Hello User	35
Элемент управления ScriptManager	39
Подведение итогов	42
Для дополнительного чтения	42
2. JavaScript	43
Язык программирования JavaScript	46
Объектно-ориентированное программирование (ООП)	57
Доступ к элементам страницы	61
Методы DOM	65
Подведение итогов	66
Для дополнительного чтения	67
3. Ajax	68
Объект XMLHttpRequest	69
Объект XMLHttpRequest	81
JSON	86
Подведение итогов	89
Для дополнительного чтения	90
II. Расширения ASP.NET AJAX	91
4. Использование расширений JavaScript в ASP.NET AJAX	93
Псевдонимы и вспомогательные функции ASP.NET AJAX	93
Расширения к существующим объектам JavaScript	96

Объектно-ориентированные возможности JavaScript в ASP.NET AJAX	97
Клиентские версии классов .NET	111
Подведение итогов	115
Для дополнительного чтения	115
5. Веб-службы	116
Обработка ошибок	116
Методы страницы	121
Управление состоянием сеанса	125
Обмен данными со сложной структурой	130
Доступ к веб-службам из JavaScript	134
Подведение итогов	145
Для дополнительного чтения	145
6. UpdatePanel: обновление только части страницы	146
Создание обновляемой области страницы	147
Подведение итогов	162
Для дополнительного чтения	163
7. Использование службы управления профилями в ASP.NET AJAX	164
Подготовка веб-сайта	165
Доступ к данным в профиле	166
Доступ к сгруппированным данным в профиле	171
Подведение итогов	176
Для дополнительного чтения	176
8. Использование службы аутентификации в ASP.NET AJAX	177
Подготовка приложения	177
Вход и выход	180
Подведение итогов	185
Для дополнительного чтения	186
9. Локализация и глобализация приложений	187
Локализация	188
Глобализация и интернационализация	203
Подведение итогов	207
Для дополнительного чтения	207
III. ASP.NET AJAX Control Toolkit	209
10. Использование Control Toolkit	211
Установка пакета Control Toolkit	212
Использование пакета Control Toolkit	216

Подведение итогов	218
Для дополнительного чтения	219
11. Добавление анимационных эффектов в веб-страницу	220
Платформа создания анимационных эффектов	220
Перетащить-и-отпустить	229
Подведение итогов	232
Для дополнительного чтения	232
12. Автодополнение ввода пользователя, борьба со спамом и не только	233
Создание раздвижных панелей	233
Управление относительным положением элемента	236
Добавление функции автодополнения к элементу управления TextBox	237
Присоединение календаря к текстовому полю	246
Динамическая свертка единственной панели	247
Отображение всплывающего окна на странице	248
Борьба со спамом в блогах и других формах ввода информации	252
Создание интерфейса с вкладками	254
Подведение итогов	256
Для дополнительного чтения	257
13. Создание собственных элементов управления и помощь сообществу	258
Создание собственных элементов управления ASP.NET AJAX	258
Содействие проекту Control Toolkit	268
Подведение итогов	277
Для дополнительного чтения	278
IV. ASP.NET AJAX Futures	279
14. Клиентские элементы управления	281
Введение в клиентские элементы управления ASP.NET AJAX	281
Использование элементов управления ASP.NET AJAX	282
Обработка событий в элементах управления	300
Подведение итогов	304
Для дополнительного чтения	304
15. Привязка и проверка данных	305
Привязка данных	305
Проверка данных	323
Подведение итогов	340
Для дополнительного чтения	340

16. Использование клиентских аспектов поведения и компонентов	341
Использование аспектов поведения	341
Использование компонентов	355
Подведение итогов	358
Для дополнительного чтения	358
17. Использование данных, размещенных на стороне сервера	359
Использование элемента управления ListView	359
Создание собственного источника данных	377
Подведение итогов	383
Для дополнительного чтения	383
18. Использование внешних веб-служб	384
Использование веб-службы Google	385
Использование веб-службы Amazon	393
Преобразование результатов веб-службы с помощью XSLT	398
Использование веб-службы Yahoo! (а также REST и XPath)	407
Подведение итогов	416
Для дополнительного чтения	416
19. Использование анимационных эффектов	417
Использование анимационных эффектов	417
Использование анимации для создания эффекта плавного исчезновения	419
Подведение итогов	430
Для дополнительного чтения	430
20. Кнопки Назад/Вперед и закладки	431
Исправление недостатков программным способом	432
Исправление недостатков в поддержке механизма закладок и кнопок Назад/Вперед с помощью элемента управления UpdateHistory	435
Исправление недостатков в поддержке механизма закладок и кнопок Назад/Вперед с помощью элементов управления из пакета ASP.NET AJAX Futures	440
Подведение итогов	447
Для дополнительного чтения	447
21. Веб-части	448
ASP.NET AJAX и ASP.NET Web Parts	448
Подведение итогов	454
Для дополнительного чтения	454

V. Microsoft AJAX Library	455
22. ASP.NET AJAX в комбинации с другими серверными технологиями	457
Использование ASP.NET AJAX вместе с PHP	458
Подведение итогов	462
Для дополнительного чтения	462
VI. Приложения	463
A. Отладка приложений ASP.NET AJAX	465
B. Справочник по XMLHttpRequest	479
C. Справочник DOM	481
D. Справочник по ASP.NET AJAX	485
E. Справочник по ScriptManager, UpdatePanel, UpdateProgress и Timer	489
Алфавитный указатель	493

Предисловие

В Википедии (на странице <http://en.wikipedia.org/wiki/Ajax>) дается более 40 толкований слова Ajax – имена персонажей из поэмы Гомера «Илиада» (Аякс Великий и Аякс Малый), название футбольной команды из Амстердама, пара моделей автомобилей, кличка лошади и даже мое любимое чистящее средство, выпускаемое компанией Colgate. Но кроме всего прочего, под термином Ajax подразумевается целый набор технологий, которые, по мнению многих, стали революционными во Всемирной паутине. Если верить различным публикациям в Интернете, Ajax – это будущее веб-разработки, основа Web 2.0 и действенное средство против множества проблем.

Многие веб-разработчики стремятся обеспечить своих клиентов приложениями, обладающими богатыми возможностями, не создавая при этом обычные приложения Windows (или, по определенным причинам, не имея возможности их создавать). Технология Ajax предоставляет все необходимое для этих целей. Она позволяет наделять их возможностями, которые раньше были характерны только для обычных настольных приложений, такими как горячие комбинации клавиш и техника «перетащить-и-отпустить» (drag-and-drop).

Платформа ASP.NET под кодовым названием «Atlas», разрабатываемая компанией Microsoft, обеспечивает набор средств, реализующих функциональность Ajax, для разработчиков ASP.NET. Она предлагает большую часть функциональности, доступной для разработки ASP.NET-приложений на стороне сервера. Осенью 2006 года было объявлено о выходе окончательной версии продукта под названием ASP.NET AJAX. (Впрочем, название Atlas более удобно для произношения.)

Я долго воздерживался от написания книги об Ajax. В течение многих лет я использовал на практике и писал о технологиях, являющихся составными частями Ajax, но сам термин появился лишь в начале 2005 года, еще до того, как данная технология стала популярной. На мой взгляд, лучше всего по этому поводу выразился Клеменс Вастерс (Clemens Vasters) в своем блоге, озаглавив эту запись «Web 2.0 бла-бла-бла AJAX бла-бла-бла преимущества(!?)» (<http://vasters.com/clemensv/PermaLink,guid,d88c1112-d8da-496e-9fd0-8cf03cf55c32.aspx>).

Крикливая реклама, которая сейчас развернулась вокруг Ajax, напоминает мне шумиху вокруг XML и веб-служб, поднятую несколько лет тому назад: все говорили о них, но мало кто читал соответствующие

спецификации. После того как шумиха улеглась и появились реальные приложения, оказалось, что объединение этих технологий дало существенный эффект.

Я уверен, что развитие Ajax пойдет тем же путем, только более высокими темпами. Во Всемирной паутине уже сейчас можно обнаружить удобные Ajax-приложения.

Но вернемся к моему нежеланию писать об Ajax.

Я всегда говорил, что описание самой технологии Ajax займет от силы страниц 20–30. Добавив немного вводной информации и примеров, этот объем можно довести до 75 страниц, может быть, даже до 100. Но чем можно было бы заполнить остальную часть книги? Авторам многих книг, посвященных Ajax, которые сейчас можно найти в продаже, пришлось пойти на определенные ухищрения, чтобы достигнуть желаемого количества страниц.

Мое мнение изменилось, когда в сентябре 2005 года я посетил конференцию профессиональных разработчиков Microsoft (Microsoft Professional Developers Conference) в Лос-Анджелесе и впервые увидел Atlas. На этой конференции компания Microsoft объявила о платформе, которая не просто обеспечивает функциональные возможности Ajax, но и предоставляет инструментальные средства, облегчающие разработку современных веб-приложений. Тогда я подумал, что это действительно стоит того, чтобы написать книгу. Я приступил к работе над рукописью, основываясь на предварительной версии Atlas. Мне пришлось несколько раз переписывать ее по мере появления промежуточных версий платформы Atlas, которые попадали мне в руки. Нехватка документации к промежуточным версиям вынуждала меня выполнять самостоятельные исследования внутренних механизмов Atlas. Как результат этих усилий, в этой книге можно найти несколько неофициальных способов решения некоторых задач.

В сентябре 2006 года вышла книга «Programming Atlas». Как одна из первых книг, посвященных этой теме, она содержала подробные сведения о платформе, которая еще продолжала изменяться. В конце января 2007 года вышла окончательная версия ASP.NET AJAX 1.0. Кроме изменения названия, существенные изменения претерпела и внутренняя организация платформы, что потребовало создания новой редакции книги, так как все существующие приложения необходимо было адаптировать под новую версию платформы.

В данной книге рассказывается о том, как создавать профессиональные динамические веб-страницы на основе платформы ASP.NET AJAX. В самом начале книги приводятся некоторые вводные сведения о JavaScript и ASP.NET, поскольку они будут необходимы для понимания темы.

Я стараюсь придерживаться принципа «больше дела, меньше слов». Поэтому книга содержит большое количество примеров, иллюстрирующих ключевые аспекты использования платформы ASP.NET AJAX.

Кроме того, стараясь фокусировать внимание на конкретных фактах, я стремился создавать небольшие примеры, объясняющие один или два вопроса, благодаря чему мне удалось избежать в книге длинных листингов, каждый из которых объяснял бы сразу множество вопросов. На своем опыте автора книг и преподавателя я уже убедился, что более короткие примеры приводят к лучшим результатам и существенно упрощают изучение материала.

Кроме того, следует отметить, что все примеры являются достаточно универсальными. Это позволит вам добавлять их непосредственно в свои собственные проекты, видоизменять и извлекать из них отдельные фрагменты по мере потребностей. Все примеры являются совершенно самостоятельными, что облегчает их использование.

Для кого написана книга

Данная книга адресована двум группам веб-разработчиков: тем, кто уже использует ASP.NET и хотел бы привнести в свои приложения возможности, которые предоставляет технология Ajax, и тем, кто в настоящее время использует в своей работе другие технологии, но заинтересован в освоении платформы ASP.NET AJAX. Кроме того, книга будет интересна программистам на JavaScript, которые хотели бы избежать головной боли при разработке программного кода, совместимого с разными типами браузеров. В этой книге используются языки программирования C# и JavaScript. Если вам необходимы начальные сведения об этих языках, можно порекомендовать обратиться к книгам «Learning C# 2005» (авторы: Джесс Либерти (Jesse Liberty) и Брайан Макдональд (Brian MacDonald)) и «Learning JavaScript» (автор Шелли Пауэрс (Shelley Powers)).

Как устроена книга

Глава 1 «ASP.NET AJAX, Ajax и ASP.NET» содержит обзор технологии Ajax и платформы ASP.NET AJAX, а также охватывает процедуру установки ASP.NET AJAX, описывает ее структуру и содержит первый пример.

Глава 2 «JavaScript» представляет собой краткое введение в JavaScript. Несмотря на то, что платформа ASP.NET AJAX стремится избавить программиста ASP.NET от ненужных функциональных подробностей, тем не менее некоторое знание JavaScript будет действительно необходимо настоящему специалисту в ASP.NET AJAX.

Глава 3 «Ajax» описывает технологии без лишней рекламы. Здесь вы узнаете о том, как работают внутренние механизмы Ajax; и все, что действительно необходимо знать, описывается менее чем на 20 страницах.

Глава 4 «Использование расширений JavaScript в ASP.NET AJAX» рассказывает о том, как ASP.NET AJAX обогащает функциональность

клиентского JavaScript за счет добавления объектно-ориентированных возможностей и переопределения некоторых классов платформы .NET для обеспечения возможности их использования на стороне клиента.

Глава 5 «Веб-службы» рассказывает о веб-службах, использующих XML. Хотя в платформе ASP.NET AJAX основной упор сделан на разработке клиентских приложений, в ней также имеются средства разработки веб-служб, исполняемых на стороне сервера. Сюда включаются возможности обработки ошибок и поддержка сеансов.

Глава 6 «UpdatePanel: обновление только части страницы» представит вашему вниманию элемент управления UpdatePanel, который позволяет обновлять отдельные части страницы без необходимости выполнять обновление всей страницы целиком. Это один из наиболее важных элементов платформы ASP.NET AJAX.

Глава 7 «Использование службы управления профилями в ASP.NET AJAX» описывает порядок организации доступа из JavaScript к прикладному интерфейсу ASP.NET 2.0 Profile API в ASP.NET AJAX.

Глава 8 «Использование службы аутентификации в ASP.NET AJAX» описывает функции JavaScript доступа к прикладному интерфейсу ASP.NET 2.0 Forms Authentication API.

Глава 9 «Локализация и глобализация приложений» охватывает тему создания веб-приложений, которые могут работать с различными языками и региональными настройками.

Глава 10 «Использование Control Toolkit» представляет собой введение в ASP.NET Control Toolkit – комплект серверных элементов управления, дополненных функциональными возможностями Ajax.

Глава 11 «Добавление анимационных эффектов в веб-страницу» описывает платформу для создания анимационных эффектов, входящую в состав ASP.NET AJAX Control Toolkit.

Глава 12 «Автодополнение ввода пользователя, борьба со спамом и не только» описывает особенности (возможно, спорные) ASP.NET AJAX Control Toolkit, демонстрирует разнообразные инструментальные средства, содержит некоторые советы и рассказывает о наиболее удачных приемах использования.

Глава 13 «Создание собственных элементов управления и помощь сообществу» рассказывает о том, как с помощью платформы Control Toolkit писать собственные элементы управления и как интегрировать их с существующими элементами в проекте.

Глава 14 «Клиентские элементы управления» описывает элементы управления, используемые на стороне клиента, которые поставляются в комплекте с ASP.NET AJAX Futures CTP. Они позволяют упростить обращение к HTML-элементам из сценариев JavaScript с использованием непротиворечивого прикладного интерфейса.

Глава 15 «Привязка и проверка данных» демонстрирует, как реализовать привязку данных к элементам управления на стороне клиента с помощью ASP.NET AJAX Futures CTP.

Глава 16 «Использование аспектов клиентского поведения и компонентов» демонстрирует аспекты поведения (behaviors), встроенные в ASP.NET AJAX, и порядок связывания их функциональности с клиентскими компонентами и элементами управления.

Глава 17 «Использование данных, размещенных на стороне сервера» объясняет, как организовать работу с базами данных. Платформа ASP.NET AJAX может связываться с источниками данных посредством специализированных веб-служб, облегчая доступ к данным без необходимости обновлять всю страницу. Кроме того, в ASP.NET AJAX имеются специализированные элементы управления, предназначенные для отображения данных на стороне клиента.

Глава 18 «Использование внешних веб-служб» поможет преодолеть ограничения общности происхождения, которые накладывает политика безопасности JavaScript, и обеспечить возможность обращения к удаленным веб-службам посредством моста на стороне сервера.

Глава 19 «Использование анимационных эффектов» демонстрирует некоторые возможности ASP.NET AJAX Futures CTP по созданию анимационных эффектов.

Глава 20 «Кнопки Назад/Вперед и закладки» рассказывает о возможных решениях двух наиболее раздражающих проблем, которые присущи Ajax-приложениям (связанных с нарушением стандартного поведения броузера).

Глава 21 «Веб-части» демонстрирует, что можно реализовать с помощью веб-частей (web parts) ASP.NET AJAX и что нельзя, включая, к примеру, реализацию механизма «перетащить-и-отпустить» (drag-and-drop), который будет работать в любом броузере.

Глава 22 «ASP.NET AJAX в комбинации с другими серверными технологиями» наглядно демонстрирует, что некоторые части Microsoft Ajax Library никак не связаны с ASP.NET 2.0. Приводится пример приложения на языке PHP, которое демонстрирует, как можно соединить эти два мира.

Приложение А «Отладка приложений ASP.NET AJAX» рассказывает, как выполняется поиск ошибок в приложениях ASP.NET AJAX и описывает некоторые инструментальные средства браузеров, которые желательно иметь каждому разработчику.

Приложение В «Справочник по XMLHttpRequest» содержит перечень методов и свойств объекта XMLHttpRequest.

Приложение С «Справочник DOM» описывает наиболее важные методы JavaScript объектной модели документа (DOM).

Приложение D «Справочник по ASP.NET AJAX» описывает наиболее важные методы, предоставляемые платформой ASP.NET AJAX.

Приложение E «Справочник по ScriptManager, UpdatePanel, UpdateProgress и Timer» документирует свойства этих четырех ключевых серверных элементов управления платформы ASP.NET AJAX.

Технические требования

Для опробования примеров из этой книги необходимо и достаточно иметь платформу ASP.NET 2.0, которая включена в состав свободно распространяемой платформы .NET. Однако, чтобы получить максимум от использования ASP.NET и ASP.NET AJAX, вам необходимо иметь какую-либо интегрированную среду разработки из предлагаемых компанией Microsoft. Среда Visual Web Developer 2005 Express Edition (VWD) распространяется бесплатно. Visual Studio 2005 (любые ее редакции) – пакет программного обеспечения, который обладает богатыми возможностями, распространяется на коммерческой основе. Любой из упомянутых инструментов прекрасно подойдет для опробования примеров из этой книги.

Типографские соглашения

В этой книге приняты следующие соглашения:

Обычный шрифт

Применяется для обозначения пунктов меню, кнопок и комбинаций клавиш (таких как Alt и Ctrl).

Курсив

Обозначает новые термины, адреса электронной почты и веб-сайтов, имена файлов и каталогов, расширений имен файлов, а также названия утилит Unix.

Моноширинный шрифт

Применяется для выделения команд, параметров, ключей, переменных, атрибутов, имен функций, типов, классов, пространств имен, методов, модулей, свойств, аргументов, значений, объектов, событий, обработчиков событий, тегов XML, тегов HTML, макроопределений, содержимого файлов и результатов работы команд.

Моноширинный жирный

Используется для выделения участков программного кода.

Моноширинный курсив

Используется для выделения элементов, которые необходимо заметить на реальные значения.



Этим значком обозначаются советы, предложения и примечания общего характера.



Этим значком обозначаются предупреждения и предостережения.

Использование программного кода примеров

Данная книга призвана оказать вам помощь в решении ваших задач. Вообще вы можете свободно использовать примеры программного кода из этой книги в своих приложениях и в документации. Вам не нужно обращаться в издательство за разрешением, если вы не собираетесь воспроизводить существенные части программного кода. Например, если вы разрабатываете программу и используете в ней несколько отрывков программного кода из книги, вам не нужно обращаться за разрешением. Однако в случае продажи или распространения компакт-дисков с примерами из этой книги вам *необходимо* получить разрешение от издательства O'Reilly. Для цитирования данной книги или примеров из нее, при ответе на вопросы не требуется получение разрешения. При включении существенных объемов программного кода примеров из этой книги в вашу документацию вам *необходимо* будет получить разрешение издательства.

Мы приветствуем, но не требуем добавлять ссылку на первоисточник при цитировании. Под ссылкой на первоисточник мы подразумеваем указание авторов, издательства и ISBN. Например: «Programming ASP.NET AJAX, by Christian Wenz. Copyright 2007 Christian Wenz, 978-0-596-51424-2».

При необходимости использовать значительный объем программного кода примеров из этой книги, за получением разрешения обращайтесь по адресу permissions@oreilly.com.

Отзывы и предложения

С вопросами и предложениями, касающимися этой книги, обращайтесь в издательство:

O'Reilly Media, Inc.
1005 Gravenstein Highway North
Sebastopol, CA 95472
800-998-9938 (в Соединенных Штатах Америки или в Канаде)
707-829-0515 (международный)
707-829-0104 (факс)

Список опечаток, файлы с примерами и другую дополнительную информацию вы найдете на сайте книги:

<http://www.oreilly.com/catalog/9780596514242>

Свои комментарии и вопросы технического характера отправляйте по адресу:

bookquestions@oreilly.com

Дополнительную информацию о книгах, обсуждения, Центр ресурсов издательства O'Reilly вы найдете на сайте:

<http://www.oreilly.com>

Safari® Books Online



Если на обложке технической книги есть пиктограмма «Safari® Books Online», это означает, что книга доступна в Сети через O'Reilly Network Safari Bookshelf.

Safari предлагает намного лучшее решение, чем электронные книги. Это виртуальная библиотека, позволяющая без труда находить тысячи лучших технических книг, вырезать и вставлять примеры кода, загружать главы и находить быстрые ответы, когда требуется наиболее верная и свежая информация. Она свободно доступна по адресу <http://safari.oreilly.com>.

Благодарности (к книге «Programming Atlas»)

Работа над этой книгой оказалась непростой задачей. Нехватка документации, которая описывала бы изменения от версии к версии, и сложности в отладке программного кода JavaScript обусловили появление большого числа ошибок и необходимость проведения экспериментов. При том, что я работал с ASP.NET и JavaScript в течение длительного времени, Atlas мне пришлось изучать с самого начала. К счастью, разработчики Atlas оказались готовы оказать поддержку и открыты для контактов, в частности, на общедоступном форуме по адресу: <http://forums.asp.net/default.aspx?GroupID=34>.

Я благодарен большому коллективу технических редакторов, которые помогли мне оформить эту книгу и с которыми мы работали в режиме обратной связи. Далее в алфавитном порядке перечислены имена тех, кто помог мне спасти свою репутацию в нескольких случаях: Адонис Битар (Adonis Bitar), Арсен Еремин (Arsen Yeremin), Бертран Ле Рой (Bertrand Le Roy), Кристоф Уилл (Christoph Wille), Майк Поуп (Mike Pope) и Тобиас Хаузер (Tobias Hauser).

Кроме того, я весьма признателен своему редактору Джону Осборну (John Osborn), который руководил этим проектом. Он – единственный известный мне редактор, который постоянно проявлял неудовольствие, когда я отправлял подготовленный материал до условленного

крайнего срока. Но именно его превосходное руководство проектом позволило мне сосредоточиться на работе и оставаться при этом в графике (а иногда и опережать его).

Должен признать, что я не большой приверженец выражения личной благодарности членам семьи, мужьям/женам/невестам/партнерам, а также кошкам/собакам. (Единственное исключение – Ричард Хандхаузен (Richard Hundhausen), который когда-то выразил благодарность за то, что в районе, где он жил, не оказалось круглосуточной службы, выполняющей расторжение брака.) Тем не менее я хотел бы воспользоваться такой возможностью и поблагодарить моих родителей. Они очень поддерживали меня в период, когда я работал над своей первой книгой, и теперь, когда у меня за плечами уже более 50 книг, я хочу выразить им свою признательность. Мне неловко признаться, но они иногда умудряются находить ошибки, даже не зная описываемых технологий: однажды мой отец заметил, что в листинге содержится неравное число открывающих и закрывающих скобок. Спасибо вам, Мама и Папа. И еще – спасибо моим друзьям и семье, которые, кажется, относились ко мне с терпением, когда я подолгу работал над книгами или уезжал на еще какую-нибудь конференцию.

Благодарности (к книге «Programming ASP.NET AJAX»)

Иногда оказывается так, что вы просто неудачно выбрали время. Спустя приблизительно две недели после выхода в свет первого издания книги под названием «Programming Atlas» компания Microsoft изменила название платформы на ASP.NET AJAX. Неудачность выбора времени обусловлена не только изменением названия – в Microsoft также весьма радикально изменили внутреннюю архитектуру платформы. Как следствие этого – все примеры программного кода для Atlas оказались неработоспособными в ASP.NET AJAX. Следует признать, что переписать большую часть примеров не составило большого труда, но некоторые функциональные возможности платформы были утрачены или изменены до неузнаваемости.

Вследствие этого данное издание книги полностью отличается от предыдущего. Структура книги претерпела существенные изменения: в нее было добавлено большое число новых глав, какой-то материал был добавлен, какой-то пришлось выбросить, а некоторые главы были полностью переписаны. Таким образом, хотя технически это второе издание – это более или менее новая книга. Однако если у вас имеется программный код, основанный на особенностях платформы Atlas, не волнуйтесь: в нескольких главах этой книги вы найдете рекомендации по переносу устаревшего программного кода на новую версию платформы.

Я выражаю благодарность моему редактору Джону Осборну (John Osborn), который руководил проектом, постоянно высказывая новые

идеи, касающиеся книги. Главным техническим редактором был Майк Поуп (Mike Pope) – эту роль он взял на себя еще при подготовке первого издания книги. Он не только удалял мои остроты в адрес Microsoft (вздых), но и поставлял бесконечное число предложений, комментариев и идей для этого издания. Это был тяжелый труд как для него (поиск ошибок), так и для меня (их исправление), но, на мой взгляд, получившийся результат стоил затраченных усилий. Спасибо вам обоим за ваши усилия, направленные на то, чтобы сделать второе издание книги еще лучше.

Кроме того, спасибо всем читателям первого издания, приславшим огромное число отзывов и предложений. Спасибо всем разработчикам, выучившимся с помощью этой книги, кто также высказывал мне свои замечания.

I

ОСНОВЫ

Глава 1. ASP.NET AJAX, Ajax и ASP.NET

Глава 2. JavaScript

Глава 3. Ajax

1

ASP.NET AJAX, Ajax и ASP.NET

В этой книге рассказывается о платформе ASP.NET AJAX (в предварительных версиях известной как «Atlas») – комплекте новых технологий компании Microsoft, которые дают возможность веб-разработчикам, и разработчикам на ASP.NET 2.0 в частности, упростить создание веб-сайтов со страницами, использующими технологии Ajax. Страницы, написанные в стиле Ajax, обеспечивают более богатый пользовательский интерфейс. Такие страницы более отзывчивы, потому что в состоянии немедленно откликаться на действия пользователя и достаточно быстро взаимодействовать с сервером. Кроме того, платформа ASP.NET AJAX включает в себя инструментальные средства для создания *гибридных (mashups)* веб-приложений, которые могут отображать содержимое, собранное с нескольких сайтов, – как правило, с использованием прикладных интерфейсов веб-служб, реализованных третьими фирмами. На протяжении всей книги мы будем исследовать эти возможности. Данная глава представляет введение в платформу ASP.NET AJAX, дает краткий обзор технологии, лежащей в ее основе, и архитектурное представление о том, как она функционирует.

ASP.NET AJAX и Ajax

Платформа ASP.NET AJAX расширяет набор технологий, поддерживаемых браузером, включая асинхронный JavaScript и XML (Asynchronous JavaScript and XML – Ajax). Технология Ajax произвела довольно много шума в последнее время (в предисловии приводятся некоторые соображения по этому поводу), поскольку она позволяет приблизить веб-приложения по функциональности и пользовательскому интерфейсу к обычным настольным приложениям.

Главная концепция, которая лежит в основе Ajax, заключается в том, чтобы дать веб-страницам возможность производить HTTP-запросы

в фоновом режиме, или *асинхронно*, без необходимости обновлять содержимое страницы целиком (или, в терминологии ASP.NET, без отправки заполненной формы и получения в ответ новой страницы). Кроме того, применение технологий Ajax позволяет создавать более отзывчивый пользовательский интерфейс с привлечением всей мощи функциональных возможностей, поддерживаемых браузером, таких как JavaScript, объектная модель документа (Document Object Model – DOM) и каскадные таблицы стилей (Cascading Style Sheets – CSS). Приложение Google Suggest (<http://www.google.com/webhp?complete=1&hl=en>) наглядно демонстрирует, как страница с поддержкой Ajax может предоставлять пользователю подсказки по мере ввода текста (этот прием известен также как *автодополнение*). Еще один пример подобных приложений – Microsoft Virtual Earth (<http://www.virtualearth.com/>).

Платформа ASP.NET AJAX поможет вам в создании подобных приложений, обладающих поддержкой Ajax, исполняющихся браузером (клиентом). Для работы с технологиями Ajax и ASP.NET AJAX на стороне клиента вам потребуется четкое понимание базовых технологий Ajax. Для создания веб-страниц с поддержкой Ajax, в которых используются сценарии, исполняемые браузером, необходимо знание и понимание языка программирования JavaScript, DOM и объекта XMLHttpRequest, который выполняет запросы от клиента к серверу. Знание языка разметки XML и языка описания преобразований XSLT желательно, но не обязательно. (В книге они подробно не рассматриваются.)

Введение в основы языка программирования JavaScript дается в главе 2. В главе 3 более подробно обсуждаются прочие технологии Ajax. Для понимания примера, который приводится в конце этой главы (раздел «Первый пример ASP.NET AJAX: Hello User»), достаточно обладать базовыми представлениями о технологии Ajax. Ваши знания будут углубляться по мере продвижения вперед.

Создание приложений на основе технологии Ajax без использования какой-либо оболочки, такой как ASP.NET AJAX, может оказаться делом достаточно сложным. В какой-то момент вы вдруг обнаружите, что вам снова и снова приходится писать один и тот же программный код, выполняющий такие действия, как отображение данных, полученных от сервера по запросу, привязка данных к элементам управления или взаимодействие с веб-службами. Вам также придется столкнуться с необходимостью писать код, который учитывал бы различия реализации DOM в разных браузерах. Одна из целей, которые преследует ASP.NET AJAX, состоит в том, чтобы уменьшить или вообще ликвидировать необходимость написания избыточного программного кода и предоставить в распоряжение разработчика, занимающегося разработкой клиентских приложений, весь опыт, накопленный разработчиками платформы ASP.NET 2.0. Еще одна цель состоит в том, чтобы усилить JavaScript некоторыми преимуществами, которые дают объектно-ориентированное программирование (ООП) и использование таких платформ, как .NET. Платформа ASP.NET AJAX включает в себя

библиотеки клиентских сценариев, которые дают программисту JavaScript/DOM/CSS следующие преимущества:

Совместимость с разными типами браузеров

Обеспечивает возможность исполнения сценариев ASP.NET AJAX в большинстве браузеров и ликвидирует необходимость создавать вручную сценарии для каждого браузера, поддерживаемого приложением. (Тем не менее создание сценариев, учитывающих особенности отдельных браузеров, неизбежно, в чем вы убедитесь в главе 3.)

Основные службы

Предоставляют расширения JavaScript, которые позволяют использовать объектно-ориентированный стиль программирования, включая поддержку классов, пространств имен, обработку событий, наследование и сериализацию (serialization) объектов с учетом форматов JSON (JavaScript Object Notation – формат представления объектов JavaScript) и XML. Наиболее важные из этих расширений будут рассматриваться в главе 4.

Библиотека базовых классов

Эта библиотека содержит ряд .NET-подобных компонентов, таких как строители строк и таймеры. О них будет говориться в главе 4.

Сценарные компоненты и элементы управления

Представляют собой версии ASP.NET AJAX стандартных элементов управления HTML, которые дополнены такими возможностями, как привязка данных, предопределенное поведение (например, поддержка функциональности «перетащить-и-отпустить») и тесная интеграция с клиентскими библиотеками ASP.NET AJAX. Вы можете использовать эти элементы управления и компоненты непосредственно или воспользоваться новой декларативной разметкой, получившей название *xml-script*, которая будет рассматриваться в нескольких главах этой книги. Если вы уже знакомы с синтаксисом разметки ASP.NET, значит, вы понимаете (в общих чертах) взаимосвязь между элементами управления HTML, абстрактными программируемыми версиями этих элементов управления и декларативным синтаксисом.

ASP.NET AJAX и ASP.NET

Платформа ASP.NET AJAX оказывает неоценимую помощь разработчику клиентских сценариев при создании Ajax-приложений, но ее возможности не ограничиваются одним только JavaScript и способностью производить асинхронные обращения к серверу. Эта платформа создавалась командой разработчиков ASP.NET, поэтому совершенно неудивительно, что одна из отличительных ее особенностей – это серверная инфраструктура, интегрированная с (и требующая наличия) ASP.NET 2.0.

Как и ASP.NET, платформа ASP.NET AJAX проектировалась для предоставления функциональных возможностей (в данном случае – в сочетании с преимуществами Ajax) без необходимости владения технологиями Ajax. Платформа ASP.NET AJAX в состоянии обеспечивать функциональность Ajax без участия программиста, точно так же, как платформа ASP.NET обеспечивает такие функциональные возможности HTTP, как отправка данных формы на сервер (postback), управление состоянием, а клиентский сценарий только требует, чтобы в ASP.NET это «просто работало».

Кроме всего прочего на стороне сервера платформа ASP.NET AJAX работает как часть ASP.NET, получая в свое распоряжение преимущества от наследования функциональных возможностей. Элементы управления ASP.NET AJAX могут взаимодействовать с элементами управления и компонентами ASP.NET и участвовать в жизненном цикле страницы. Она имеет возможность обращаться к таким службам ASP.NET 2.0, как механизмы управления сессиями, аутентификацией и профилями, позволяя вам использовать эти функциональные возможности на стороне клиента. Наконец, с помощью ASP.NET AJAX и ASP.NET можно выполнять обращения к специализированным веб-службам.

Ниже перечислены некоторые из ключевых элементов серверной платформы ASP.NET AJAX:

Серверные элементы управления ASP.NET AJAX

ASP.NET AJAX предоставляет серверные элементы управления, напоминающие аналогичные элементы, имеющиеся в ASP.NET 2.0, но в отличие от последних их функциональнее доступна из клиентской платформы ASP.NET AJAX. Наиболее фундаментальными для приложений ASP.NET AJAX являются следующие два элемента управления: `ScriptManager`, который будет обсуждаться далее в этой главе (раздел «Элемент управления `ScriptManager`»), и `UpdatePanel`, который рассматривается в главе 6.

Службы ASP.NET AJAX и ASP.NET

Предоставляет некоторые прикладные службы ASP.NET 2.0, доступные непосредственно клиентским сценариям ASP.NET AJAX, включая службы управления профилями, персонализацией, аутентификацией, членством в группах и региональными настройками по умолчанию. Вполне оправданно ожидать, что число служб ASP.NET, доступных приложениям ASP.NET AJAX, будет увеличиваться с выходом новых версий ASP.NET AJAX.

Библиотека Microsoft Ajax Library

Эта библиотека содержит исключительно программный код JavaScript, который никак не связан с ASP.NET. Благодаря этому данная библиотека может использоваться и без ASP.NET, о чем мы поговорим в главе 22.

Несомненно, ASP.NET AJAX займет свое законное место ключевого компонента в следующих версиях ASP.NET и получит полную поддержку дизайнеров, технологии IntelliSense и инструментальных средств отладки в будущих версиях Visual Studio.

Пакеты ASP.NET AJAX

На домашней странице проекта ASP.NET AJAX (<http://ajax.asp.net/>) представлен перечень пакетов, каждый из которых предназначен для решения определенных задач:

ASP.NET AJAX Extensions

Известен также под названием «ASP.NET AJAX Core». Это «главный» пакет ASP.NET AJAX. Он целиком поддерживается компанией Microsoft и содержит инфраструктуру ASP.NET AJAX (обсуждается во второй части книги).

ASP.NET AJAX Control Toolkit

Этот пакет содержит обширную коллекцию серверных компонентов, предоставляющих легкий и удобный доступ к потрясающим функциональным возможностям Ajax. Пакет Control Toolkit – проект с открытыми исходными текстами, хотя компания Microsoft и остается руководителем проекта, что гарантирует его высокое качество. Тем не менее официально Microsoft не осуществляет поддержку элементов пакета. Возможности пакета Control Toolkit будут исследованы третьей части книги.

ASP.NET AJAX Futures Release

Этот пакет содержит экспериментальные реализации функциональных возможностей, которые могут войти (или не войти) в состав ASP.NET и ASP.NET AJAX. Кроме того, в пакет Futures Release помещаются редко используемые элементы, которые первоначально входили в состав предварительных версий ASP.NET AJAX. Версия CTP (Community Technology Preview – предварительная версия, доступная для загрузки) обновляется чаще, чем основной пакет. Но эта версия пакета официально не поддерживается, поэтому вы можете использовать его только на свой страх и риск. В четвертой части книги рассматривается пакет Futures Release, наиболее свежий на момент работы над книгой, а именно версия ASP.NET Futures от июля 2007 года. Кроме того, в мае 2007 года ASP.NET AJAX Futures CTP входил в состав пакета ASP.NET Futures CTP, который включает в себя новые интересные (и не поддерживаемые) возможности для работы с классической версией ASP.NET.

Microsoft Ajax Library

Упомянутая ранее библиотека JavaScript, которая будет рассматриваться в пятой части книги.



Что самое удивительное – компания Microsoft предоставляет полный комплект исходных текстов ASP.NET AJAX. Загрузить их можно по адресу: <http://ajax.asp.net/>.

В этой главе будет рассказано и показано, как установить основной пакет ASP.NET AJAX Extensions (а также мы вкратце коснемся пакета Futures CTP), а затем во второй части книги этот пакет будет рассмотрен более подробно. Остальные пакеты будут рассматриваться далее в соответствующих частях.

Подготовка и установка ASP.NET AJAX

Лучший способ осознать мощь ASP.NET AJAX – начать использовать его. Все, что вам потребуется для разработки приложений, – это браузер с поддержкой JavaScript на стороне клиента и веб-сервер с поддержкой ASP.NET 2.0. Для начала вполне будет достаточно обычного текстового редактора. Однако, по мере роста сложности приложений, наличие интегрированной среды разработки с такими функциональными возможностями, как IntelliSense, автодополнение кода, управление проектами, отладчик и WYSIWYG (What You See Is What You Get – что видишь, то и получишь) поможет сэкономить массу времени. В мире ASP.NET 2.0 для этих целей обычно используется редактор, входящий в состав Visual Studio 2005.

Установка интегрированной среды разработки

Следует иметь в виду, что хотя в общем случае наилучшим выбором является приобретение полной версии Visual Studio 2005, тем не менее бесплатно распространяемая Express-версия Visual Studio 2005, предназначенная для разработки веб-приложений – Microsoft Visual Web Developer 2005 Express Edition, – также полностью поддерживает ASP.NET AJAX.



Для упрощения в этой книге иногда вместо полного названия Visual Web Developer будет использоваться аббревиатура VWD. Под этой аббревиатурой будут подразумеваться как Visual Web Developer 2005 Express Edition, так и полная версия Visual Studio 2005. Компонент разработки веб-приложений, входящий в состав Visual Studio 2005 также называется Visual Web Developer (вы можете увидеть его в процессе установки Visual Studio), поэтому название VWD можно считать универсальным обозначением инструментальных средств разработки приложений для ASP.NET 2.0 в интегрированной среде разработки компании Microsoft.

Если у вас еще нет интегрированной среды разработки, установите Visual Studio 2005 или Visual Web Developer Express Edition. Если вы выбрали второй вариант, откройте в браузере страницу <http://msdn.microsoft.com/vstudio/express/vwd/download>, где вы найдете мастер установ-

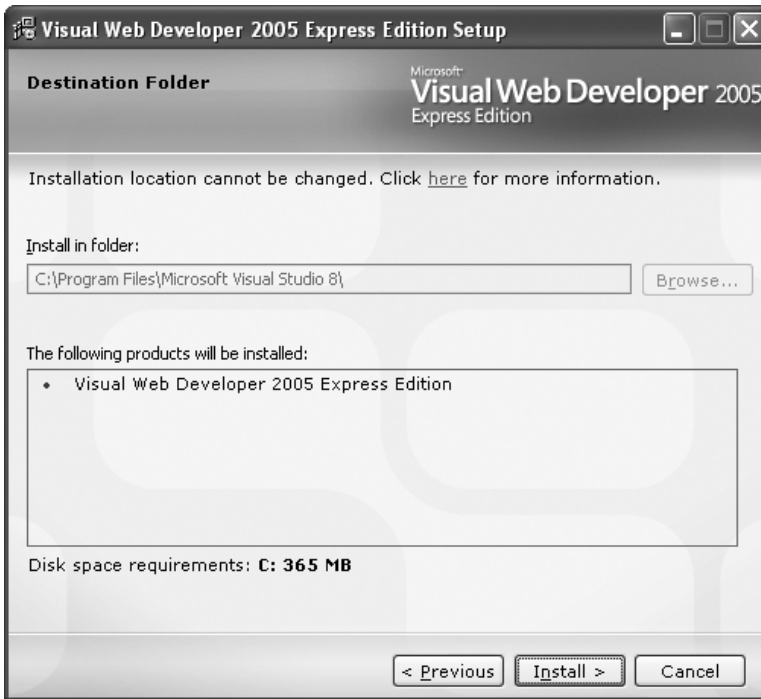


Рис. 1.1. Установка Visual Web Developer Express Edition

ки по сети, который не только загрузит и установит VWD (рис. 1.1), но и установит при необходимости .NET Framework 2.0.



Если мастер установки по сети не сможет работать на вашей машине (например, из-за ограничений на соединение с узлами Интернета из корпоративной сети или из-за очень медленного соединения), на этом же сайте вы найдете образы компакт-дисков в формате ISO и IMG, содержащие все необходимое для установки Visual Web Developer (<http://msdn.microsoft.com/vstudio/express/support/install>), которые можно загрузить на любой другой компьютер, на котором имеется лучшее соединение с Интернетом, а затем записать их на компакт-диски.

Установка ASP.NET AJAX

Независимо от того, какая версия VWD используется, ASP.NET AJAX интегрируется непосредственно в среду разработки. На домашней странице ASP.NET AJAX (<http://ajax.asp.net>) вы найдете ссылку на дистрибутив ASP.NET AJAX в формате MSI с именем *ASPAJAXExtSetup.msi*. Ищите ссылку с текстом «Microsoft ASP.NET 2.0 AJAX Extensions 1.0».

Перед запуском пакета установки необходимо удалить все предыдущие версии ASP.NET AJAX, которые могли быть установлены в системе.

Мастер установки *.msi* задаст всего несколько вопросов. После того как будут приняты условия лицензионного соглашения ASP.NET AJAX, начнется процесс установки, как показано на рис. 1.2.

После установки в VWD или в Visual Studio появится новый шаблон приложений – ASP.NET AJAX Web Site. Описанный способ установки ASP.NET AJAX является предпочтительным, так как в этом случае все необходимые файлы копируются в соответствующие каталоги (рис. 1.3). Кроме того, при таком способе установки сборка ASP.NET

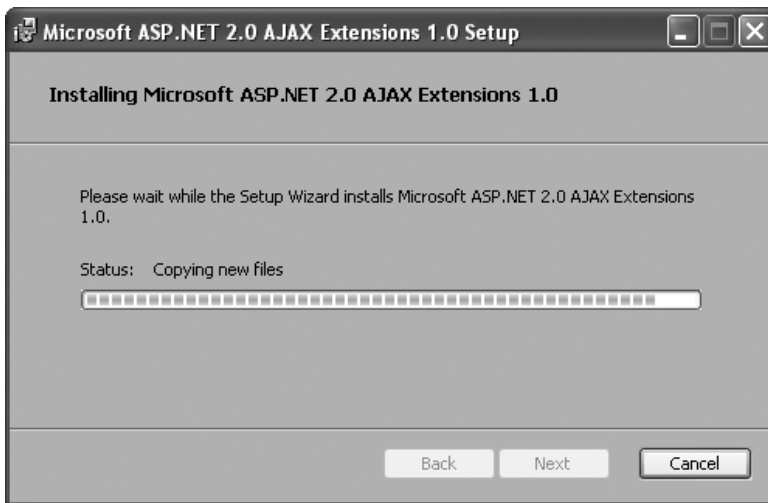


Рис. 1.2. Установка шаблона

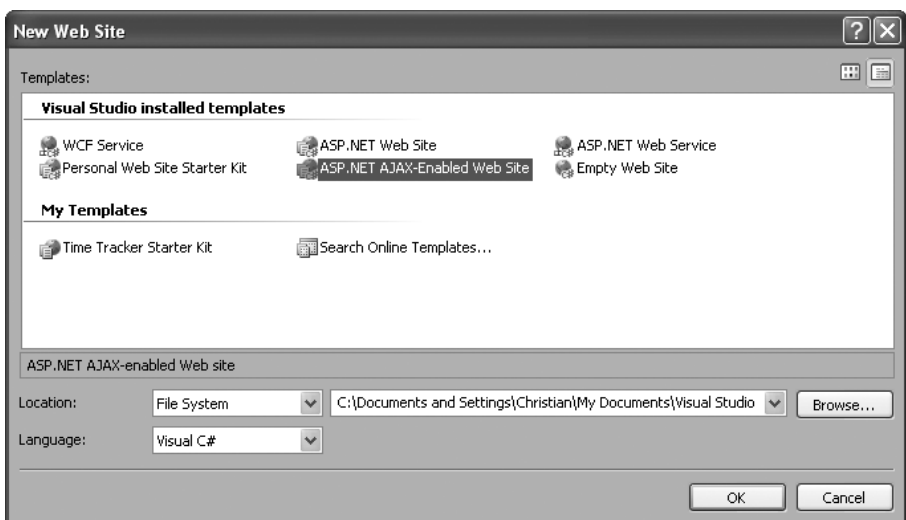


Рис. 1.3. После установки появится новый шаблон создания веб-сайта

AJAX будет помещена в глобальный кэш сборки (Global Assembly Cache – GAC), что автоматически делает ее доступной для всех приложений ASP.NET.

ASP.NET AJAX также работает в готовящихся к выходу Visual Studio 2008 и Visual Web Developer 2008 Express Edition. В этих средах разработки при создании нового веб-сайта просто укажите, что должна использоваться платформа .NET Framework 3.5 (рис. 1.4); вам не придется устанавливать ASP.NET AJAX, поскольку она уже включена в состав .NET Framework 3.5. Для версий .NET ниже 3.5 придется отдельно устанавливать ASP.NET AJAX. Если вы работаете и с Visual Studio 2005, и с Visual Studio 2008 или вы хотите открыть в Visual Studio 2008 проект ASP.NET AJAX, который был создан в Visual Studio 2005, перейдите к разделу «Для дополнительного чтения» в конце этой главы; там вы найдете несколько важных советов по данному вопросу.



На сайте проекта ASP.NET AJAX можно найти дополнительную информацию и программное обеспечение. Ниже перечислена лишь часть того, что можно найти:

- Документация, которая ознакомит вас с некоторыми аспектами ASP.NET AJAX (<http://ajax.asp.net/docs>). Кроме того, имеется возможность загрузить и установить документацию на локальном компьютере (*AspNet_AJAX_Documentation.zip*).
- Примеры ASP.NET AJAX
- Ссылки на другие пакеты ASP.NET AJAX: ASP.NET AJAX Control Toolkit, Microsoft Ajax Library, ASP.NET AJAX Futures Release и исходные тексты платформы ASP.NET AJAX.

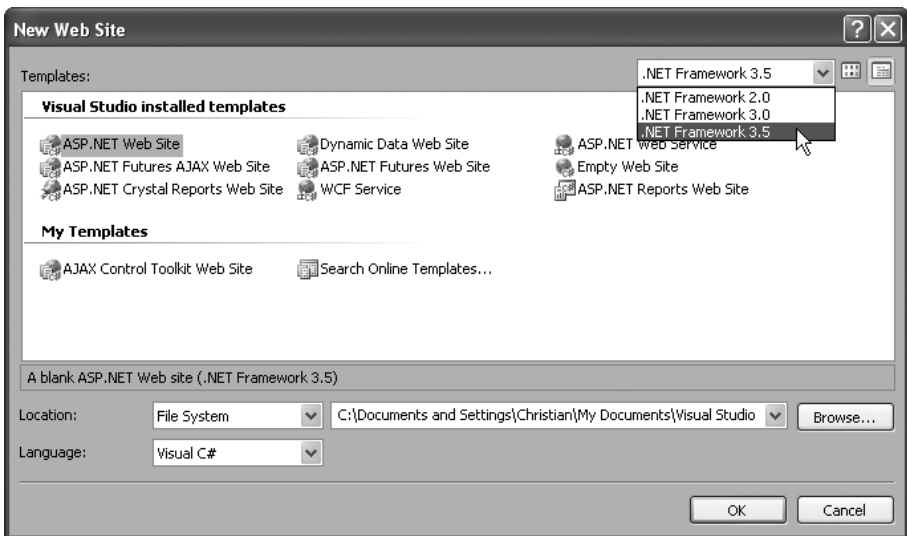


Рис. 1.4. В .NET Framework 3.5 для проектов по созданию новых веб-сайтов автоматически используется ASP.NET AJAX

Установка испытательной базы данных

Примеры в этой книге написаны в предположении, что в качестве сервера баз данных используется SQL Server 2005 Express Edition (хотя примеры легко могут быть адаптированы для использования любых других источников данных). Если у вас на компьютере установлен Microsoft SQL Server, то он также может быть использован. В противном случае загрузите и установите SQL Server Express (его можно загрузить бесплатно).

Чтобы максимально упростить процесс установки и развертывания, во всех примерах из этой книги, работающих с базой данных, используется типовая база данных *AdventureWorks* компании Microsoft. Также предполагается, что *AdventureWorks* была установлена в локальный сервер SQL Server 2005 Express Edition и доступна с параметрами аутентификации Windows в *(local)\SQLEXPRESS*.



Возможно, вам придется изменить в примерах путь к SQL Express, если он отличается от указанного.

В зависимости от используемой версии SQL Server требуемая версия базы данных *AdventureWorks* доступна для загрузки по следующему адресу:

SQL Server 2005

<http://www.microsoft.com/downloads/details.aspx?familyid=E719ECF7-9F46-4312-AF89-6AD8702E4E6E&displaylang=en>

SQL Server 2005 Express Edition

<http://www.microsoft.com/downloads/details.aspx?familyid=9697AAAA-AD4B-416E-87A4-A8B154F92787&displaylang=en>

Выберите нужную версию, загрузите дистрибутив и запустите мастер установки. По окончании установки вам будет необходимо подключить файл *AdventureWorks_Data.mdf* (в подкаталоге *Data*) к вашему серверу баз данных SQL Server 2005. Проще всего это сделать с помощью Microsoft SQL Server Management Studio Express (SSMSE) – бесплатного инструмента с графическим интерфейсом для администрирования SQL Server 2005 Express Edition. SSMSE доступен для загрузки в двух версиях, 32-битной и 64-битной, по адресу: <http://www.microsoft.com/downloads/details.aspx?FamilyID=c243a5ae-4bd1-4e3d-94b8-5a0f62bf7796&DisplayLang=en>.

По окончании установки и настройки базы данных *AdventureWorks* откройте меню Пуск (Start) и запустите SQL Server. Введите в диалоговом окне информацию для вашего сервера SQL Server 2005 Express Edition, как показано на рис. 1.5. По умолчанию сервер доступен по имени *(local)\SQLEXPRESS* или *ИмяКомпьютера\SQLEXPRESS* с использованием типа аутентификации Windows Authentication.



Рис. 1.5. Окно регистрации SSMSE

Далее, в окне SSMSE щелкните правой кнопкой мыши на папке с базами данных и выберите пункт контекстного меню *Attach* (присоединить). В открывшемся диалоге (рис. 1.6) щелкните на кнопке *Add* (добавить) и выберите файл *AdventureWorks_Data.mdf*. Дважды нажмите *OK*. После этого база данных *AdventureWorks* будет подключена к вашему серверу SQL Server 2005 Express Edition.

Установка Futures CTP

Процесс установки ASP.NET AJAX Futures CTP (часть более обширного пакета ASP.NET Futures CTP) мало чем отличается от установки ASP.NET AJAX Extensions. При установке в Visual Studio и Visual Web Developer также будет установлен шаблон веб-сайта. Шаблон имеет название *ASP.NET Futures AJAX Web Site* (не следует путать его с *ASP.NET Futures Web Site*, который служит для создания сайтов без поддержки Ajax). Прежде чем устанавливать пакет CTP, необходимо сначала установить пакет ASP.NET AJAX Extensions.

В примерах, представленных в этой книге, приложения ASP.NET AJAX Extensions получают стандартное имя *AJAXEnabledWebSite1*, а приложения CTP получают имя *AJAXFuturesEnabledWebSite1*. Приложения первого типа по умолчанию работают с портом 1234, а второго – с портом 1236. Порт с номером 1235 зарезервирован для примеров приложений ASP.NET AJAX Control Toolkit. Предполагается, вы можете определить иной номер порта.

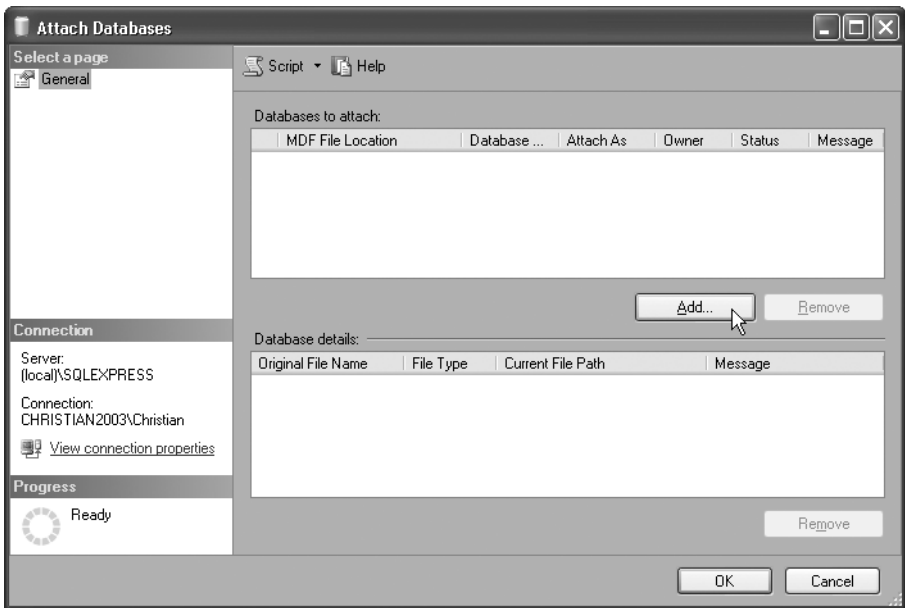


Рис. 1.6. Подключение MDF-файла к серверу SQL Server 2005 Express Edition

Структура и архитектура ASP.NET AJAX

Теперь можно приступить к работе с ASP.NET AJAX. Запустите VWD и создайте с помощью шаблона ASP.NET AJAX новый проект веб-сайта ASP.NET. Если теперь посмотреть на окно Solution Explorer, вы увидите проект обычного веб-сайта ASP.NET. Не удивляйтесь, это мастер установки поместил сборку ASP.NET AJAX непосредственно в GAC. Единственное, что его отличает, – это наличие файла *Web.config*, который содержит настройки, необходимые для работы ASP.NET AJAX.

ASP.NET AJAX состоит из серверных и клиентских компонентов. Есть возможность использовать только серверные или только клиентские компоненты. Единственное исключение: любое приложение ASP.NET AJAX требует наличие серверного элемента управления *ScriptManager*, который будет рассматриваться ниже, в этой же главе. Как правило, в приложениях используются и серверные, и клиентские компоненты.

Назначение клиентских и серверных компонентов в проекте ASP.NET AJAX станет понятнее, когда мы ближе познакомимся с тем, как Ajax-приложения используют в своей работе объект *XMLHttpRequest*.

На рис. 1.7 показана базовая структура ASP.NET AJAX. При стандартном подходе в ответ на каждый запрос со стороны клиента сервер отправляет одну веб-страницу, но при наличии поддержки Ajax веб-страницы могут производить непрерывный обмен данными с сервером. Платформа ASP.NET AJAX осуществляет поддержку такого обмена

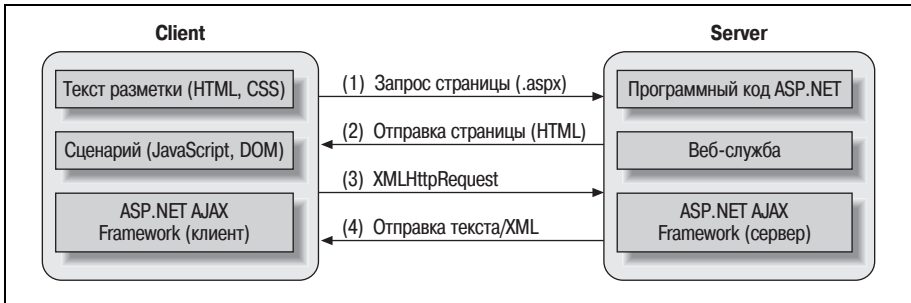


Рис. 1.7. Жизненный цикл веб-страницы ASP.NET AJAX

с обеих сторон соединения. Библиотеки с программным кодом сценариев (которые динамически загружаются компонентом `ScriptControl`, как вы вскоре увидите) облегчают взаимодействие браузера и веб-сервера и упрощают программирование клиентской части. Программный код, реализованный в сборке ASP.NET AJAX на стороне сервера, выполняет прием и обработку запросов от объекта `XMLHttpRequest` и, кроме того, обеспечивает некоторые дополнительные удобные серверные элементы управления, которые будут описаны далее в книге. В результате компоненты на стороне клиента и на стороне сервера получают возможность обмениваться данными, почти не требуя усилий программиста.

Платформа ASP.NET AJAX на стороне клиента (нижний уровень клиентского компонента на рис. 1.7) передается клиентскому браузеру при первом обращении к странице с поддержкой ASP.NET AJAX (шаги 1 и 2 на рис. 1.7). После этого из той же страницы Ajax-приложения можно производить запросы к серверу по протоколу HTTP и получать в ответ текст или XML (шаги 3 и 4 на рис. 1.7). Веб-страницы ASP.NET для выполнения различных задач могут производить как полноценную отправку форм с данными, так и асинхронные запросы к серверу.

Отдельные компоненты ASP.NET AJAX, работающие на стороне сервера и на стороне клиента, подробно будут рассматриваться на протяжении всей книги. Однако вы должны постоянно держать в уме эту базовую структуру, включающую в себя обмен данными между клиентом и сервером. Чем реже выполняется запрос полных страниц, тем лучше, по крайней мере, потому, что за счет этого можно избежать обновления страницы целиком.

Первый пример ASP.NET AJAX: Hello User

Чтобы проверить, насколько успешно прошла установка ASP.NET AJAX, и увидеть платформу в действии, закончим эту главу примером создания небольшого приложения. В этом примере страница ожидает ввода имени пользователя, отправляет его серверу (в фоновом режиме, с помощью объекта `XMLHttpRequest`), получает обратно имя, дополненное некоторым текстом, и выводит его перед пользователем. Данный

пример наглядно демонстрирует, насколько просто создаются приложения, обладающие функциональными возможностями ASP.NET AJAX. В последующих главах принцип действия внутренних механизмов будет описан более подробно.

В VWD создайте новый проект веб-сайта из шаблона ASP.NET AJAX. Затем в корневом каталоге веб-сайта создайте новую веб-службу (с помощью файла-шаблона веб-службы) с именем *WebService.asmx*. В файле *.asmx* реализуйте простой веб-метод, который принимает единственный строковый параметр, скопировав в файл программный код из примера 1.1. Обратите внимание на обязательный атрибут `[ScriptServices]`, который фактически определяется платформой ASP.NET AJAX (в пространстве имен `System.Web.Script.Services`).

Пример 1.1. Веб-служба

```
WebService.asmx

<%@ WebService Language="C#" Class="WebService" %>

using System;
using System.Web;
using System.Web.Services;
using System.Web.Services.Protocols;

[WebService(Namespace = "http://hauser-wenz.de/AspNetAJAX/")]
[WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)]
[System.Web.Script.Services.ScriptService]
public class WebService : System.Web.Services.WebService {
    [WebMethod]
    public string sayHello(string name) {
        return "Hello " + name + ", says the server!";
    }
}
```

Теперь попробуйте обратиться к этой веб-службе из веб-браузера, добавив символы */js* в конец адреса URL. При обращении по такому URL, как показано на рис. 1.8, возвращается программный код JavaScript, который фактически представляет собой реализацию класса-посредника JavaScript (подробное описание см. в главе 5). Самое важное здесь заключается в том, что этот программный код создает переменную *WebService*, которая обеспечивает взаимодействие с веб-службой.

Можно заметить, что шаблон ASP.NET AJAX уже создал файл *Default.aspx*, содержимое которого мы рассмотрим ниже. Следующий фрагмент – это текст, содержащийся в данном файле:

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Default.aspx.cs"
Inherits="_Default" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
```

```

    <title>Untitled Page</title>
  </head>
  <body>
    <form id="form1" runat="server">
      <asp:ScriptManager ID="ScriptManager1" runat="server" />
      <div>
      </div>
    </form>
  </body>
</html>

```

Первое, на что следует обратить внимание, – это новый элемент управления: `<asp:ScriptManager>`. Этот элемент управления является основой любой веб-страницы с поддержкой ASP.NET AJAX. Элемент `ScriptManager` будет рассмотрен гораздо подробнее ниже, в этой же главе (раздел «Элемент управления `ScriptManager`»).

Откройте файл `Default.aspx` в текстовом редакторе и добавьте в элемент `ScriptManager` ссылку на веб-службу, как показано ниже. При запуске страницы благодаря этой ссылке будет сгенерирован объект-посредник JavaScript, с помощью которого страница будет способна использовать программный код, динамически сгенерированный программным кодом веб-службы:

```

<asp:ScriptManager ID="ScriptManager1" runat="server">
  <Services>
    <asp:ServiceReference Path="WebService.asmx" />
  </Services>
</asp:ScriptManager>

```

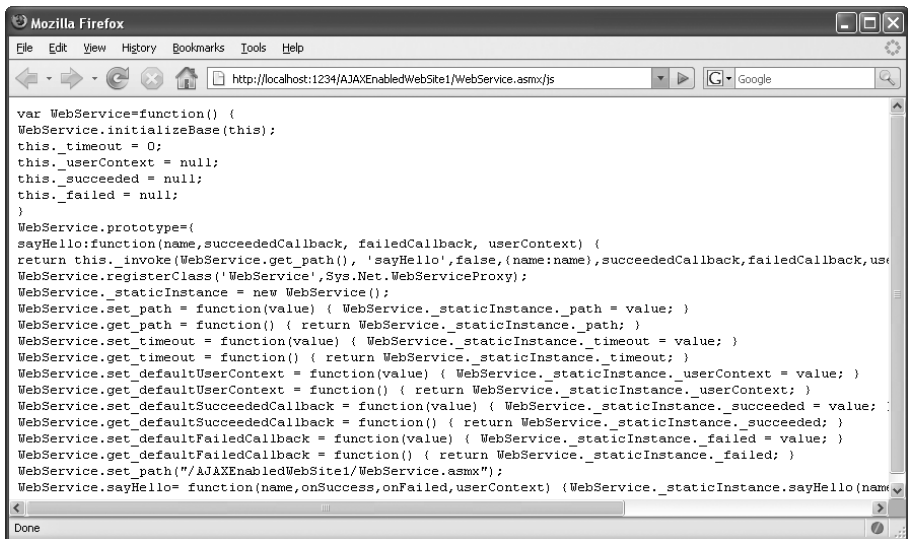


Рис. 1.8. Платформа ASP.NET AJAX создала этот программный код JavaScript автоматически

Теперь необходимо вставить в страницу несколько HTML-элементов. Добавьте текстовое поле ввода и кнопку в существующий элемент `<form>` (внутри элемента `<div>`, если вы собираетесь придерживаться требований стандарта XHTML):

```
<input type="text" id="name" name="name" />
<input type="button" value="Call Service" onclick="callService(this.form);" />
```

Обработчик события кнопки вызывает функцию JavaScript с именем `callService()` и передает ей ссылку на текущую форму. Метод `callService()` – это та самая точка в сценарии, где происходит обращение к веб-службе. Чтобы вызвать метод веб-службы `sayHello()`, сценарий должен воспользоваться объектом-посредником, который доступен через автоматически созданную переменную с именем `WebService`. (Имя `WebService` соответствует имени класса веб-службы, созданного ранее.)

Метод `sayHello()` ожидает получить не только строку, но и три ссылки на функции-обработчики: первая вызывается в случае успешного исполнения веб-службы (`callComplete`), вторая при появлении ошибки (`callError`) и третья по истечении предельного времени ожидания ответа (в данном примере мы используем только первые две функции).

Затем поместите следующий программный код в клиентский элемент `<script>` на странице:

```
function callService(f) {
    WebService.sayHello(
        f.elements["name"].value,
        callComplete,
        callError);
}
```

В заключение необходимо написать две функции, которые будут обрабатывать события `callComplete` и `callError`. Для этого добавьте следующий фрагмент в клиентский блок сценария, который только что был создан:

```
function callComplete(result) {
    window.alert(result);
}
function callError(result) {
    window.alert("Error! " + result);
}
```

В примере 1.2 приводится содержимое файла *Default.aspx* целиком.

Пример 1.2. Простая страница ASP.NET AJAX, которая обращается к веб-службе

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Default.aspx.cs"
Inherits="_Default" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
```

```
<title>ASP.NET AJAX</title>
<script language="Javascript" type="text/javascript">
function callService(f) {
    WebService.sayHello(
        f.elements["name"].value,
        callComplete,
        callError);
}

function callComplete(result) {
    window.alert(result);
}

function callError(result) {
    window.alert("Error! " + result);
}
</script>
</head>
<body>
    <form id="form1" runat="server">
        <asp:ScriptManager
            ID="ScriptManager1"
            runat="server">
            <Services>
                <asp:ServiceReference Path="WebService.asmx" />
            </Services>
        </asp:ScriptManager>
        <div>
            <input type="text" id="name" name="name" />
            <input type="button" value="Call Service"
                onclick="callService(this.form);" />
        </div>
    </form>
</body>
</html>
```

На рис. 1.9 показан внешний вид страницы после того, как она будет загружена и пользователь щелкнет на кнопке Call Service.

Запустите приложение (клавишей F5 или Ctrl+F5 в VWD). В окне браузера можно увидеть вполне предсказуемый результат, причем не только в Internet Explorer, но и в любом другом браузере, отвечающем нашим требованиям. Щелкните на кнопке несколько раз и убедитесь, что обновления страницы не происходит, хотя страница выполняет обмен данными с веб-службой на стороне сервера.

Элемент управления ScriptManager

Теперь, после рассмотрения первого примера, можно поговорить о том, как он работает и как работают остальные примеры ASP.NET AJAX из этой книги.

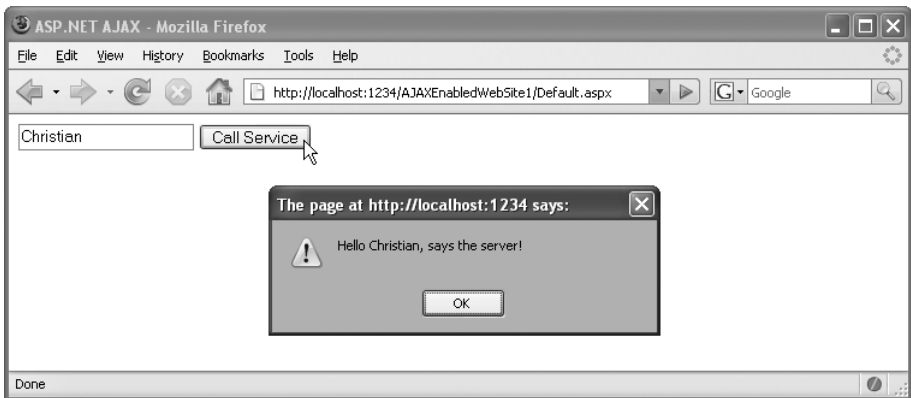


Рис. 1.9. Приложение работает именно так, как и ожидалось

Центральным элементом любой страницы с поддержкой ASP.NET AJAX является элемент управления `ScriptManager`. Он выполняет загрузку необходимых библиотек JavaScript платформы ASP.NET AJAX. Если вы запустите приложение ASP.NET AJAX и затем просмотрите полученный исходный текст страницы в браузере, то увидите, что программный код немного изменился по сравнению с изначально введенным. Элемент `<asp:ScriptManager>` окажется замещен следующим программным кодом (учтите, что в вашей системе малопонятные значения в строках URL будут отличаться от тех, что приведены здесь):

```
<script src="/AJAXEnabledWebSite1/WebResource.axd?d=Jd4j-
uCaCWzJ5gY8Rtbjnw2&am;t=632962478475625000" type="text/javascript"></script>
<script src="/AJAXEnabledWebSite1/ScriptResource.axd?d=4vKPTV3rK3vcGz1fNEcIXI-
FjnEGgHGMpqfU1mBk4NA5KxnrqcWXFT6h1n9QkTug1OUzzonzRPSF5F3_-
0aWhdWOb3FCqEciv4AZjgqdk5us1&am;t=633074690770156250" type="text/javascript">
</script>
<script src="/AJAXEnabledWebSite1/ScriptResource.axd?d=4vKPTV3rK3vcGz1fNEcIXI-
FjnEGgHGMpqfU1mBk4NA5KxnrqcWXFT6h1n9QkTug1OUzzonzRPSF5F3_-
0aWhdU1qTQPzcChFvbHT6FrI-81&am;t=633074690770156250"
type="text/javascript"></script>
```

Кроме того, платформа ASP.NET AJAX сгенерирует программный код JavaScript, выполняющий инициализацию отдельных частей инфраструктуры Ajax.



При сборке приложения в режиме отладки программный код JavaScript, создаваемый ASP.NET, будет отформатирован, и в его состав будет включена обработка некоторых ошибок. Это удобно в процессе разработки, но не требуется (более того, рекомендуется обратное) при развертывании веб-сайта, поскольку это увеличивает объем программного кода JavaScript, а следовательно, и размер страницы.

Загрузка дополнительных файлов JavaScript

Элемент `ScriptManager` может также использоваться для загрузки дополнительных библиотек JavaScript: как тех, что входят в состав ASP.NET AJAX, так и ваших собственных:

```
<asp:ScriptManager ID="ScriptManager1" runat="server">
  <Scripts>
    <asp:ScriptReference Path="MyScript.js" />
  </Scripts>
</asp:ScriptManager>
```

Обратите внимание: программный код JavaScript, сгенерированный ASP.NET AJAX, выполняет проверку, что все внешние файлы JavaScript загружены без ошибок. Если вы пишете свои собственные сценарии JavaScript, не забудьте добавить следующий фрагмент кода в конец каждого файла `.js`:

```
if (typeof(Sys) != "undefined") {
    Sys.Application.notifyScriptLoaded();
}
```

Этот фрагмент извещает ASP.NET AJAX, что был достигнут конец файла, и сигнализирует платформе Ajax, что все внешние сценарии были полностью загружены.

Элемент `ScriptManager` должен присутствовать на любой странице, которая использует функциональные возможности ASP.NET AJAX.

Если вы используете шаблонные страницы ASP.NET 2.0 и большая часть ваших страниц использует функциональные возможности платформы ASP.NET AJAX, есть смысл разместить элемент управления `ScriptManager` не в отдельных страницах, а в странице шаблона. Однако такой подход может вызвать сложности при необходимости обратиться из какой-то страницы к дополнительным файлам JavaScript или веб-службам (таким как в примере «Hello User», рассмотренном ранее). Поскольку на странице может находиться только один элемент `ScriptManager`, то вы будете вынуждены мириться со ссылками на файл JavaScript или веб-службу в каждой странице, использующей функциональность ASP.NET AJAX, даже если на этой странице внешние ресурсы будут неостребованы.

Для решения этой проблемы платформа ASP.NET AJAX предоставляет элемент управления `ScriptManagerProxy`. Данный элемент обеспечивает необходимую функциональность `ScriptManager` даже при наличии другого элемента `ScriptManager`.

```
<asp:ScriptManagerProxy ID="ScriptManagerProxy1" runat="server">
  <Scripts>
    <asp:ScriptReference Path="MyScript.js" />
  </Scripts>
</asp:ScriptManagerProxy>
```

```
</Scripts>  
</asp:ScriptManagerProxy>
```

Элемент `ScriptManagerProxy` удобно использовать при создании пользовательских элементов управления, обладающих функциональностью AJAX, когда на главной странице уже присутствует элемент `ScriptManager`.

Это был лишь первый шаг. А сколько всего откроется в следующих главах!

Подведение итогов

Эта глава – введение в платформу ASP.NET AJAX; в ней мы рассказали о том, какое отношение эта платформа имеет к Ajax и ASP.NET 2.0, и провели вас через процесс установки ASP.NET AJAX и другого программного обеспечения, необходимого для опробования примеров из этой книги, включая базу данных *AdventureWorks*. Здесь, кроме того, был представлен первый работающий пример ASP.NET AJAX и элемент управления `ScriptManager` – один из двух ключевых серверных элементов управления, которые входят в состав ASP.NET AJAX. В следующей главе будет рассматриваться язык программирования JavaScript, знание которого потребуется вам при работе с платформой ASP.NET AJAX.

Для дополнительного чтения

<http://ajax.asp.net/>

Домашняя страница проекта ASP.NET AJAX

<http://ajax.asp.net/downloads/default.aspx?tabid=47>

Страница, откуда можно загрузить ASP.NET AJAX

<http://ajax.asp.net/docs/InstallingASPNETAJAX.aspx>

Документация Microsoft с инструкциями по установке ASP.NET AJAX

<http://blogs.msdn.com/webdevtools/archive/2007/07/28/upgrading-asp-net-ajax-1-0-websites-and-web-applications-to-net-framework-3-5.aspx>

Важная информация по обновлению веб-сайтов с поддержкой ASP.NET AJAX, созданных в Visual Studio 2005, при переходе на Visual Studio 2008/.NET Framework 3.5

<http://weblogs.asp.net/scottgu/archive/2007/08/04/fixes-for-common-vs-2008-and-net-3-5-beta2-issues.aspx>

Перечень известных проблем в Visual Studio 2008 Beta 2 и способы их решения.