

ASP.NET



Visual Studio .NET

Разработка
Web-приложений

Взаимодействие
с базами данных

XML Web-сервисы

*Мощное и гибкое средство создания
многофункциональных Web-проектов*

Игорь Шапошников

САМОУЧИТЕЛЬ

ASP.NET

Санкт-Петербург

«БХВ-Петербург»

2002

УДК 681.3.06

Шапошников И. В.

Самоучитель ASP.NET. — СПб.: БХВ-Петербург, 2002. — 368 с.: ил.

ISBN 5-94157-171-2

Книга представляет собой руководство по созданию мощных полнофункциональных сайтов на основе технологии ASP (Active Server Pages). В книге рассмотрены вопросы установки средств разработки ASP.NET и настройки Web-сервера IIS. Описан процесс создания ASP-приложений, начиная самыми простыми и заканчивая многофункциональными ASP-сценариями. Отдельные главы книги посвящены обзору языков Visual Basic, XML и SOAP. Читатель познакомится с основами создания интернет-сервисов на основе ASP, достаточно новым направлением с многообещающими перспективами.

Для Web-программистов

УДК 681.3.06

Группа подготовки издания:

Главный редактор	<i>Екатерина Кондукова</i>
Зам. главного редактора	<i>Евгений Рыбаков</i>
Зав. редакцией	<i>Анна Кузьмина</i>
Редактор	<i>Владислав Борисов</i>
Компьютерная верстка	<i>Натальи Караваевой</i>
Корректор	<i>Татьяна Звертановская</i>
Дизайн обложки	<i>Игоря Цырульников</i>
Зав. производством	<i>Николай Тверских</i>

Лицензия ИД № 02429 от 24.07.00. Подписано в печать 29.03.02.

Формат 70×100 ¹/₁₆. Печать офсетная. Усл. печ. л. 29,67.

Тираж 3000 экз. Заказ №

"БХВ-Петербург", 198005, Санкт-Петербург, Измайловский пр., 29.

Гигиеническое заключение на продукцию, товар, № 77.99.1.953.П.950.3.99 от 01.03.1999 г. выдано Департаментом ГСЭН Минздрава России.

Отпечатано с готовых диапозитивов
в Академической типографии "Наука" РАН
199034, Санкт-Петербург, 9 линия, 12.

ISBN 5-94157-171-2

© Шапошников И. В., 2002

© Оформление, издательство "БХВ-Петербург", 2002

Содержание

Благодарности	1
Введение.....	3
Глава 1. Инсталляция и настройка	5
Среда разработки	5
Администрирование IIS	7
Глава 2. Основы языка Visual Basic.....	19
Управляющая логика.....	19
Типы данных.....	25
Массивы и коллекции.....	29
Встроенные функции	32
Глава 3. Приложения ASP	45
Здравствуй, мир.....	45
HTML-дизайн	52
Элементы <i>Web Forms</i>	68
Объект <i>HttpResponse</i>	72
Объект <i>HttpRequest</i>	75
Объект <i>HttpApplicationState</i>	79
Объект <i>HttpServerUtility</i>	81
Объект <i>HttpSessionState</i>	83
Формы.....	85
Работа с графикой	111
Работа с файлами.....	119
Работа с объектами <i>HttpRequest</i> и <i>HttpResponse</i>	137
Правила кэширования Web-страниц.....	154
Cookies.....	157
Стилевое оформление Web-страниц.....	165
Сеансы работы пользователей.....	172
Табличные компоненты	191

Аутентификация и авторизация пользователей	201
Использование электронной почты	218
Конфигурирование приложений	226
Отладка приложений	228
Глава 4. Взаимодействие с базами данных	233
Постановка задачи	233
Создание базы данных	234
Ввод данных с Web-страницы	240
Объект <i>SqlConnection</i>	246
Объект <i>SqlCommand</i>	249
Поиск и отображение информации	250
Объект <i>SqlDataAdapter</i>	258
ADO.NET	260
Извлечение данных из XML-файлов	261
Глава 5. XML	267
История создания	267
Корни XML	268
Структура XML-документов	269
Инструкции XML-процессора	270
Объявление типа документа	272
Элементы XML-документа	276
Атрибуты элементов	279
Сущности	283
Комментарии и условные обозначения	287
Тело XML-документа	289
XLink. Умные гиперссылки	292
Создание гиперссылок в XML	293
Ссылки бывают разные	294
Локальный ресурс	299
Внешние ресурсы	300
Правила прохождения ссылок	301
Идентифицирующие элементы ссылок	302
Атрибут типа элемента	303
Атрибут целеуказания	303
Семантические атрибуты	304
Поведенческие атрибуты	304
Атрибуты прохождения ссылки	305
XPointer	306
Основные правила	307
Абсолютные указатели	308
Относительные указатели	309

Абсолютная адресация	310
Относительная адресация	312
Адресация интервалов	315
Адресация строчных субресурсов	316
Адресация элементов.....	317
Глава 6. SOAP.....	319
Основы	319
Структура SOAP.....	321
Заголовки сообщений SOAP	323
Тело SOAP-сообщения.....	325
Типы данных SOAP	327
Глава 7. Сервисы ASP.....	335
Постановка задачи	335
Простейший Web-сервис	336
Клиентская часть	344
Послесловие	349
Приложение	351

Даниленко Ольге

*"Если б не было тебя, я б шел по миру, как
слепой,
В гуле сотен чужих голосов узнать пытаюсь
голос твой
И звук твоих шагов"*

Благодарности

Огромное спасибо Вам за то, что Вы держите сейчас книгу в руках и читаете ее. Все книги пишутся именно для их, читателей. На самом деле, мало есть на свете вещей стимулирующих так, как отзывы читателей о книге.

Но автор никогда не пишет книгу один. Всегда есть люди, чей вклад в книгу является не менее весомым, чем авторский. Я, конечно, говорю о редакторском коллективе издательства "БХВ-Петербург". Спасибо нашему главному редактору Кондуковой Екатерине, отдельное спасибо принимающему редактору. На самом деле, упоминать надо всю группу подготовки издания, которая в очередной раз сделала работу просто замечательно.

Спасибо также всем моим друзьям в Сети: Буке, Умке, Лайке, Sweet-dream, Панку, Финисту и многим-многим другим. Ребята, я всех вас помню и люблю.

Огромное спасибо моей Ольге, которая всегда помогала и поддерживала меня во время работы. Спасибо тебе, милая, за терпение и понимание.

Введение

О чем же эта книга? Что такое ASP.NET? По сути дела, в книге рассказывается о создании мощных полнофункциональных сайтов. Давно прошли те времена, когда сайт был просто набором статических Web-страниц. Теперь этого мало. Сейчас сайты все чаще создают с использованием новейших технологий программирования. Большая часть страниц генерируется автоматически, с использованием информации, получаемой из каких-либо баз данных. Естественно, для создания таких страниц одного языка HTML (Hypertext Markup Language, язык описания гипертекстовых документов), который все-таки является основой Web-страниц, явно недостаточно. Даже использование технологии динамического HTML, опирающегося на сценарии, выполняющиеся на стороне пользователя, не поможет в создании подобных сайтов.

Дело в том, что работа приложений, динамически формирующих самые разнообразные Web-страницы по запросу пользователя, может происходить только на самом сервере, который поддерживает функционирование искомого Web-сайта. А если в процессе работы с запросами удаленного пользователя приходится работать с содержимым базы данных, то здесь никак не обойтись без серверных приложений, так как база данных, естественно, может функционировать только на сервере.

Для создания приложений, действующих на стороне Web-сервера, есть достаточно много приемлемых технологий. Но технология ASP (Active Server Pages), о которой и рассказывается в этой книге, стоит особняком в этом ряду. Дело в том, что практически все технологии серверных приложений реализованы в виде дополнительных модулей, подключаемых к web-серверам. А программы ASP не нуждаются в дополнительных интерпретирующих модулях. Поддержка ASP изначально встроена в Web-сервер IIS (Internet Information Services). Этот достаточно мощный сервер в последнее время автоматически включается в состав операционных систем Windows 2000 и Windows NT.

Таким образом, если вы привыкли работать с продуктами корпорации Microsoft, и перед вами стоит задача создать полнофункциональный Web-сайт, то, пожалуй, выбор будет очевидным. Следует использовать связку IIS+ASP. Тем более что ASP-приложения отлично интегрируются с базами данных от Microsoft.

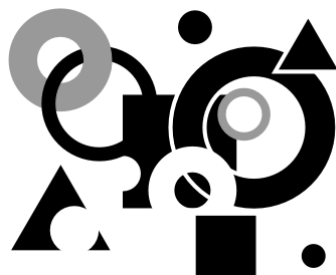
Технология ASP известна достаточно давно. В этой книге рассматривается ее последняя на данный момент модификация — ASP.NET. Разработка приложений ASP.NET производится с помощью пакета Visual Studio .NET.

Структура книги достаточно проста. В первой главе мы рассмотрим вопросы установки средств разработки программ ASP.NET и настройки web-сервера IIS. Во второй главе будет приведен обзор языка Visual Basic, на котором и разрабатываются приложения ASP. Третья глава является, пожалуй, основной в этой книге. Именно в ней будет рассказано о создании ASP-приложений. Начнем мы с самых простых вариантов и закончим созданием многофункциональных ASP-сценариев. В четвертой главе будут освещены вопросы связывания Web-приложений с системами управления базами данных. Затем будут рассмотрены языки XML и SOAP, с которыми очень тесно связана технология Microsoft .NET. А в последней главе мы рассмотрим создание интернет-сервисов на основе ASP. Это достаточно новое направление, но уже с многообещающими перспективами. Собственно говоря, сама концепция ".NET" строится на создании сервисов, которые будут доступны для работы пользователя с любой компьютерной платформой.

Разумеется, сами сервисы будут функционировать на Windows-серверах, но удаленные пользователи могут использовать их вне зависимости от того, на какой платформе они работают. Для связи клиентской системы и .NET-сервиса обычно используется язык XML с его потрясающей адаптивностью и расширяемостью. Естественно, для того чтобы строить подобные сервисы, следует знать основные правила работы с языком XML. Именно поэтому целая глава будет посвящена рассмотрению языка XML.

А теперь перейдем к работе.

Глава 1



Инсталляция и настройка

Среда разработки

Создание приложений ASP.NET производится при помощи средства разработки Visual Studio .NET. Это огромный пакет средств и технологий для создания самых разнообразных приложений. У меня просто не хватает слов, чтобы вкратце описать все его возможности. Разработчик может использовать один из трех языков программирования, входящих в состав пакета: Visual Basic .NET, Visual C++ .NET и Visual C# .NET. Все они объединены общим интерфейсом, поэтому разработчик для реализации каждой задачи может выбирать тот язык, который максимально хорошо подойдет к данной ситуации. При этом внешний вид среды разработки, ее основные функции и механизмы не будут изменяться, и в любом случае, разработчик будет действовать в рамках знакомой среды программирования. Подобное единообразие значит очень много, так как удобство разработки является одним из ключевых факторов скорости создания и качества программного продукта.

Дистрибутив Visual Studio .NET занимает семь дисков CD-ROM. Естественно, на жесткий диск переносится не все их содержимое, но так или иначе, придется выделить не менее 3 Гбайт на основном жестком диске и около 300 Мбайт на системном логическом диске.

Как и все системное программное обеспечение, производимое корпорацией Microsoft, среда разработки Visual Studio .NET является достаточно требовательным приложением. В качестве минимальных системных требований, помимо уже упоминавшегося необходимого объема свободного места на дисках системы, необходимо наличие процессора не ниже уровня Pentium II — 450 и оперативная память от 96 Мбайт. На практике, для более приемлемой скорости работы, следует нарастить объем оперативной памяти до 128 Мбайт. Да и более мощный процессор явно не помешает.

Также стоит обратить внимание на используемую операционную систему. Рекомендуется использовать операционные системы семейства Windows, основанные на технологии NT. Это сама Windows NT, Windows 2000 (желательно оригинальная версия) или Windows XP. Однако последняя в настоящий момент имеет не слишком долгую историю эксплуатации, поэтому, теоретически, может быть несколько нестабильна. На первые две перечисленные системы надо установить самые новые пакеты обновлений. Иначе говоря, машину, на которой будет производиться разработка, необходимо поднять на максимально высокий уровень.

Впрочем, на пятом диске дистрибутива Visual Studio .NET находятся все дополнительные компоненты, необходимые для функционирования среды разработки и среды выполнения программ. И первым действием программы установки будет проверка наличия необходимых дополнительных компонентов в составе операционной системы. Кстати, одним из таких дополнительных компонентов является набор серверных расширений FrontPage, который отказывается корректно устанавливаться на русскую версию Windows 2000. Поэтому я и упоминал, что следует использовать именно оригинальную версию. Впрочем, данный компонент не является критичным для функционирования Visual Studio .NET. И уж тем более, он не нужен для разработки ASP-приложений. Данные расширения позволяют обеспечивать работу некоторых активных элементов, которые FrontPage позволяет встраивать в разрабатываемые HTML-документы. Мы же будем сами разрабатывать подобные активные элементы, поэтому наличие специализированных расширений нам абсолютно не нужно.

Изначально скрипты ASP создавались при помощи языка Visual Basic. Однако концепция MSIL (Microsoft Intermediate Language), являющаяся частью Microsoft .NET, позволяет писать ASP-приложения на любом языке, поддерживаемом средой разработки Visual Studio .NET. В ее состав входят языки Visual Basic .NET, Visual C++ .NET и Visual C# .NET. Приложения ASP можно разрабатывать на любом из этих языков, но традиционно они разрабатывались при помощи языка Visual Basic, и мы тоже будем использовать именно этот язык.

Но вернемся к рассмотрению концепции MSIL. Для того чтобы разработчик мог выбирать язык программирования в зависимости от своих предпочтений или поставленной задачи, необходимо нечто большее, чем единая среда разработки. Необходима общая основа для всех приложений. И она была создана. Как мы помним, при установке Visual Studio .NET были установлены дополнительные элементы. Одним из таких элементов была среда .NET Framework. Этот компонент можно рассматривать как некоторое дополнение к операционной системе, которое несколько расширяет ее возможности. Вместе с .NET Framework технология разработки приложений разбивается на два этапа. Сначала разработчик пишет исходный код на выбранном языке программирования. После этого компилятор Visual

Studio .NET переводит его на язык MSIL. А на втором этапе приложение, записанное на языке MSIL при помощи виртуальной машины CLR (Common Language Runtime), переводится в код, присущий той компьютерной платформе, на которой приложение будет функционировать. При этом приложение получает возможность использовать всю функциональность, описанную в Microsoft .NET Framework.

Естественно, подобная структура компиляции полностью копирует идеологию Java. Как мы помним, Java-компиляторы переводили исходный текст программы не в машинные коды, а в некий байт-код, который затем компилировался виртуальной Java-машиной. Конечно виртуальную Java-машину надо было писать для каждой специфичной платформы. Точно также обстоит дело и с Microsoft .NET Framework. Точнее, с ее основной частью — так называемым *"компилятором по требованию"*, JIT-компилятором, чаще называемым *джиттером*, который и отвечает за перевод кода MSIL в код, специфичный для конкретной платформы. На момент написания книги джиттер был разработан только для операционной системы Microsoft Windows 2000 и Windows XP.

Администрирование IIS

Но вернемся к ASP. Данная технология позволяет создавать приложения, обеспечивающие интерактивность и иные дополнительные возможности для сайтов. ASP-приложения действуют не на стороне удаленного пользователя, а на самом WWW-сервере. Поддержка технологии ASP встроена в WWW-сервер IIS (Internet Information Services), который входит в состав дистрибутива Windows NT 4 и Windows 2000. Впрочем, на диске с дополнительными компонентами, входящими в состав дистрибутива Visual Studio.NET, также находится этот компонент, и если он не установлен в операционной системе, то инсталлятор Visual Studio.NET попытается установить IIS самостоятельно. Но все-таки стоит установить его с самого начала еще перед установкой Visual Studio.NET, чтобы правильно его настроить.

Для установки IIS следует открыть окно **Control Panel** (Панель управления) операционной системы и запустить утилиту **Add/Remove Programs** (Установка и удаление программ). Нам потребуется утилита **Add/Remove Windows Components** (Установка компонентов Windows). Эта утилита активизирует диалоговое окно **Windows Components Wizard** (Мастер установки компонентов Windows), показанное на рис. 1.1.

Компонент Internet Information Services (IIS) состоит из нескольких разнородных частей. Список их можно увидеть, если выделить в диалоговом окне **Windows Component Wizard** (Мастер установки компонентов Windows) строку **Internet Information Services (IIS)** и нажать кнопку **Details** (Состав). В результате будет отображено диалоговое окно **Internet Information Services (IIS)**, показанное на рис. 1.2.

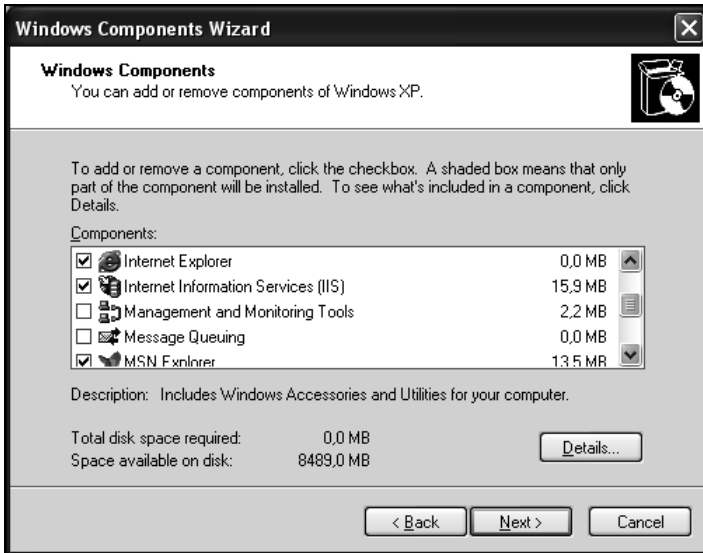


Рис. 1.1. Диалоговое окно **Windows Components Wizard**

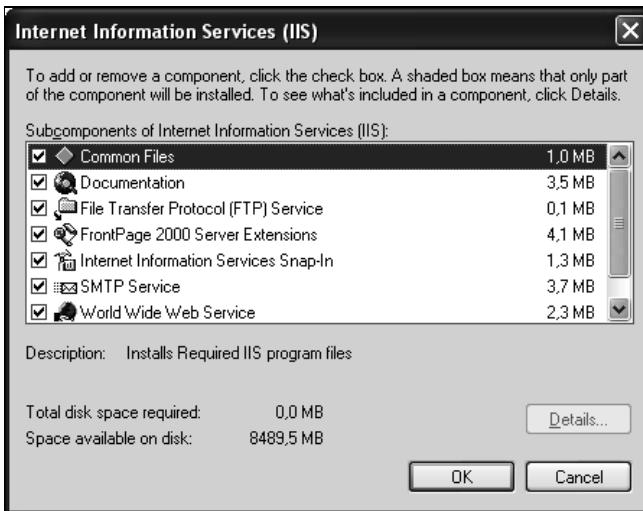


Рис. 1.2. Диалоговое окно **Internet Information Services (IIS)**

В диалоговом окне **Internet Information Services (IIS)** мы можем указать, какие именно функциональные части элемента IIS требуется установить в операционную систему. Для создания полноценного сервера нам потребуются следующие компоненты:

- Common Files** (Общие файлы). Общие файлы, необходимые для функционирования практически всех элементов IIS;

- ❑ **File Transfer Protocol (FTP) Server** (Служба FTP). Сервер приема и передачи файлов по протоколу FTP;
- ❑ **Internet Information Services Snap-In** (Оснастка IIS). Система администрирования сервера при помощи утилиты MMC (Microsoft Management Console);
- ❑ **Personal Web Manager** (Управление личным Web-сервером). Графический интерфейс администрирования сервера;
- ❑ **SMTP service** (Служба SMTP). Служба отправки исходящих сообщений электронной почты по протоколу SMTP;
- ❑ **World Wide Web Server** (Служба WWW). Собственно, сам WWW-сервер, который и поддерживает функционирование сайтов.

Для того чтобы установить все эти элементы, необходимо поставить флажки в независимых переключателях, находящихся рядом с наименованиями компонентов, и нажать кнопку **ОК**. После этого компонент IIS будет автоматически установлен. Использование остальных элементов не является критичным для создания полноценных сайтов с использованием ASP-приложений, поэтому их установка необязательна.

В процессе установки IIS будет создана структура каталогов, в которых размещается содержимое серверов. По умолчанию, корневой каталог элемента носит наименование `Inetpub`. В каталоге `FTPROOT` размещается содержимое FTP-сервера, каталог `MAILROOT` предназначен для почтового SMTP-сервера, а в каталоге `WWWROOT` находится содержимое WWW-сервера.

Нас будет интересовать администрирование именно WWW-сервера. При установке, автоматически создается некий локальный сайт, в терминологии Microsoft часто называемый Web-узлом. Доступ к нему можно получить, если в строке **Адрес** браузера набрать адрес **`http://localhost`**.

Необходимо сделать некоторое техническое отступление. В подавляющем большинстве, Web-сайты обладают доменным именем, как, например, **`www.bhv.ru`**. В том случае, если сайт не имеет подобного имени, доступ к нему осуществляется по IP-адресу того сервера, на котором он установлен. Однако, использование IP-адресов вместо доменных имен явно невыгодно, так как запомнить IP-адрес достаточно трудно, чего нельзя сказать о правильно подобранном доменном имени. Сопоставление доменных имен и IP-адресов серверов, на которых размещены сайты с этими именами производится DNS-серверами при помощи DNS-таблиц. Служба DNS (Domain Name Service) как раз и существует для того, чтобы пользователь WWW мог набрать обычное символьное доменное имя сайта, вместо IP-адреса сервера.

IP-адрес, как известно, представляет собой комбинацию из четырех положительных целых чисел, находящихся в промежутке от 0 до 255, и разделенных точками. При этом все адреса, начинающиеся с числа 127

(обозначается, как 127.*.*), являются локальными. Эти адреса нельзя назначить серверу, реально находящемуся в Интернете. Они могут быть назначены только локальным системам. Проще говоря, если необходимо на локальной машине разработать сервер, используя при этом некое доменное имя, например, **www.mysite.com**, то надо сопоставить это имя одному из таких локальных IP-адресов. Казалось бы, для этого необходимо запускать на локальной машине еще и DNS-сервер, чтобы браузер мог в поисках сервера, на котором содержится искомый сайт, обратиться к этому серверу без выхода в Интернет, но на самом деле все проще. Все Windows-системы перед тем, как обратиться к DNS-серверу, проверяют наличие файла `hosts`, находящегося в системном каталоге Windows. Это текстовый файл, в котором указывается список доменных имен и соответствующих им сайтов. Поэтому, если мы хотим, чтобы на нашем сервере действовал локальный сайт с доменным именем **www.mysite.com**, необходимо в этот файл добавить следующую строку:

```
127.0.0.1 www.mysite.com
```

Теперь, для того чтобы связаться с этим сайтом, браузер не будет обращаться к DNS-серверам, а всего лишь воспользуется файлом `hosts`. Впрочем, справедливости ради надо сказать, что сайт с доменным именем `localhost` система всегда ищет в корневом каталоге функционирующего WWW-сервера даже без обращения к файлу `hosts`.

Мы уже говорили, что основной сайт размещается в каталоге `Inetpub/WWWROOT`. Но в этом каталоге находятся не только HTML-документы и скрипты. Любой сайт, это еще и структура дополнительных каталогов. Всегда удобнее структурировать данные по различным каталогам. Отдельно держать графические файлы, отдельно информацию для различных разделов, и так далее. Поэтому URL графического файла с наименованием `picture1.gif` может выглядеть следующим образом:

<http://www.mysite.com/image/picture1.gif>

При этом подразумевается, что графический файл `picture1.gif` находится в каталоге `images`, который в свою очередь размещен в корневом каталоге WWW-сервера. На самом деле, в URL не обязательно указывается точное наименование каталога. В нем указывается наименование так называемого "виртуального каталога", чье наименование не обязательно будет совпадать с именем настоящего физического каталога. Так, каталог, в котором находится графический файл `picture1.gif`, мог бы на самом деле называться `img`, а не `image`, как указано в URL. То есть, виртуальные каталоги это всего лишь названия реальных, физических каталогов.

Процедура создания виртуальных каталогов достаточно проста. В группе органов управления, располагающихся в **Control Panel** (Панель управления), существует инструмент настройки основных сервисов системы —