Поддержка сетевого подключения осуществляется через несколько пространств имен, определенных в среде .NET Framework, и главным среди них является пространство

имен System.Net. В нем определяется целый ряд высокоуровневых, но простых в использовании классов, поддерживающих различные виды операций, характерных для работы с Интернетом. Для этих целей доступен также ряд пространств, вложенных в пространство имен System.Net.

Например, средства низкоуровневого сетевого управления через сокеты находятся в пространстве имен System.Net. Sockets, поддержка электронной почты — в пространстве имен System.Net.Mail, а поддержка защищенных сетевых потоков — в пространстве имен System.Net. Security. Дополнительные функциональные возможности предоставляются в ряде других вложенных пространств имен. К числу других не менее важных пространств имен, связанных с сетевым подключением к Интернету, относится пространство System. Web. Это и вложенные в него пространства имен поддерживают сетевые приложения на основе технологии ASP.NET.

В среде .NET Framework имеется достаточно гибких средств и возможностей для сетевого

подключения к Интернету. Тем не менее для разработки многих приложений

более предпочтительными оказываются функциональные возможности, доступные в

пространстве имен System.Net. Они и удобны, и просты в использовании. Именно поэтому

пространству имен System.Net будет уделено основное внимание в этой главе.

**Члены пространства имен System.Net**

Пространство имен System.Net довольно обширно и состоит из многих членов.

Полное их описание и обсуждение всех аспектов программирования для Интернета

выходит далеко за рамки этой главы. (На самом деле для подробного рассмотрения

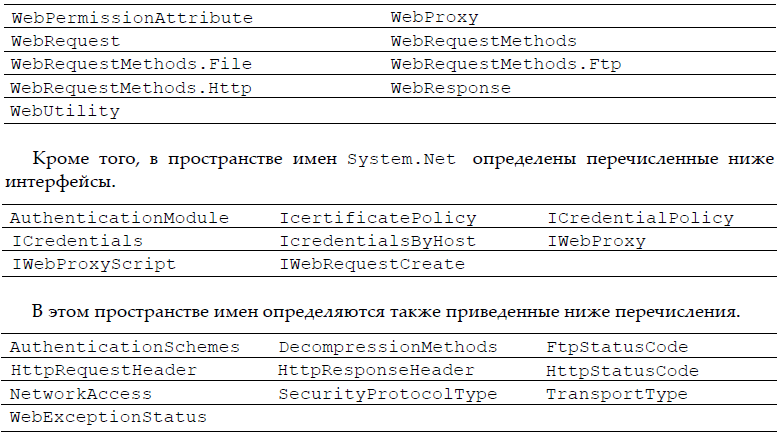
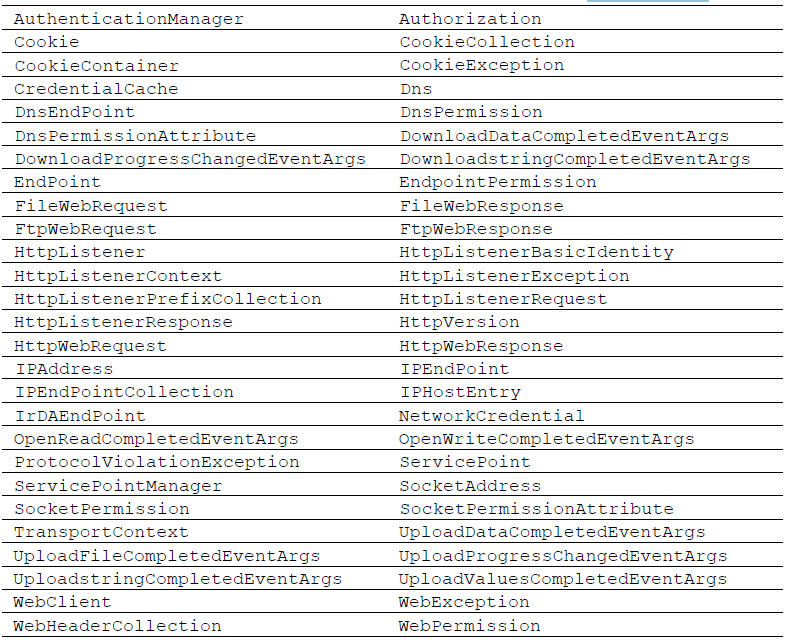
всех вопросов, связанных с сетевым подключением к Интернету и его поддержкой в

С#, потребуется отдельная книга.) Однако целесообразно хотя бы перечислить члены

пространства имен System.Net, чтобы дать какое-то представление о том, что именно

доступно для использования в этом пространстве.

Ниже перечислены классы, определенные в пространстве имен System.Net.



Помимо этого, в пространстве имен System.Net определен ряд делегатов.

Несмотря на то что в пространстве имен System.Net определено немало членов,

лишь немногие из них на самом деле требуются при решении наиболее типичных задач

программирования для Интернета. Основу сетевых программных средств составляют абстрактные

классы WebRequest и WebResponse. От этих классов наследуют все классы,

поддерживающие конкретные сетевые протоколы. (*Протокол* определяет правила передачи

данных по сети.) Например, к производным классам, поддерживающим стандартный

сетевой протокол HTTP, относятся классы HttpWebRequest и HttpWebResponse.

Классы HttpWebRequest и HttpWebResponse довольно просты в использовании.

Тем не менее решение некоторых задач можно еще больше упростить, применяя подход,

основанный на классе WebClient. Так, если требуется только загрузить или выгрузить

файл, то для этой цели лучше всего подойдет класс WebClient.

**Универсальные идентификаторы ресурсов**

В основу программирования для Интернета положено понятие *универсального идентификатора*

*ресурса* (URI), иногда еще называемого *унифицированным указателем информационного*

*ресурса* (URL). Этот идентификатор описывает местоположение ресурса в

сети. В корпорации Microsoft принято пользоваться сокращением URI при описании

членов пространства имен System.Net, и поэтому в данной книге выбрано именно

это сокращение для обозначения универсального идентификатора ресурса. Идентификаторы

URI, без сомнения, известны каждому, кто хотя бы раз пользовался браузером

для поиска информации в Интернете. По существу, это адрес информационного

ресурса, который указывается в соответствующем поле окна браузера.

**Протокол://Идентификационный\_номер\_сервера/Путь\_к\_файлу?Запрос**

где *Протокол* — это применяемый протокол, например HTTP; *Идентификационный\_*

*номер\_сервера* — конкретный сервер, например mhprofessional.com или

HerbSchildt.com; *Путь\_к\_файлу* — путь к конкретному файлу. Если же *Путь\_к\_*

*файлу* не указан, то получается страница, доступная на указанном сервере по умолчанию.

И наконец, Запрос обозначает информацию, отправляемую на сервер. Указывать

Запрос необязательно. В C# идентификаторы URI инкапсулированы в класс Uri, рассматриваемый

далее в этой главе.

**Основы организации доступа к Интернету**

В классах, находящихся в пространстве имен System.Net, поддерживается модель

взаимодействия с Интернетом по принципу запроса и ответа. При таком подходе

пользовательская программа, являющаяся клиентом, запрашивает информацию у

сервера, а затем переходит в состояние ожидания ответа. Например, в качестве запроса

программа может отправить на сервер идентификатор URI некоторого веб-сайта. В ответ

она получит гипертекстовую страницу, соответствующую указанному идентификатору

URI. Такой принцип запроса и ответа удобен и прост в применении, поскольку

большинство деталей сетевого взаимодействия реализуются автоматически.

На вершине иерархии сетевых классов находятся классы WebRequest и

WebResponse, реализующие так называемые *подключаемые протоколы.* Как должно

быть известно большинству читателей, для передачи данных в сети применяется несколько

разнотипных протоколов. К числу наиболее распространенных в Интернете

относятся протокол передачи гипертекстовых файлов (HTTP), а также протокол передачи

файлов (FTP). При создании идентификатора URI его префикс обозначает применяемый

сетевой протокол. Например, в идентификаторе http://www.HerbSchildt.

com используется префикс http, обозначающий протокол передачи гипертекстовых

файлов (HTTP).

Как упоминалось выше, классы WebRequest и WebResponse являются абстрактными,

а следовательно, в них определенны в самом общем виде операции запроса и

ответа, типичные для всех протоколов. От этих классов наследуют более конкретные

производные классы, в которых реализуются отдельные протоколы. Эти производные

классы регистрируются самостоятельно, используя для этой цели статический метод

RegisterPrefix(), определенный в классе WebRequest. При создании объекта типа

WebRequest автоматически используется протокол, указываемый в префиксе URI,

если, конечно, он доступен. Преимущество такого принципа "подключения" протоколов

заключается в том, что большая часть кода пользовательской программы остается

без изменения независимо от типа применяемого протокола.

В среде выполнения .NET Runtime протоколы HTTP, HTTPS и FTP определяются автоматически.

Так, если указать идентификатор URI с префиксом HTTP, то будет автоматически

получен HTTP-совместимый класс, который поддерживает протокол HTTP.

А если указать идентификатор URI с префиксом FTP, то будет автоматически получен

FTP-совместимый класс, поддерживающий протокол FTP.

При сетевом подключении к Интернету чаще всего применяется протокол HTTP,

поэтому именно он и рассматривается главным образом в этой главе. (Тем не менее

аналогичные приемы распространяются и на все остальные поддерживаемые протоколы.)

Протокол HTTP поддерживается в классах HttpWebRequest и HttpWebResponse.

Эти классы наследуют от классов WebRequest и WebResponse, а кроме того, имеют

собственные дополнительные члены, применимые непосредственно к протоколу

HTTP.

В пространстве имен System.Net поддерживается как синхронная, так и асинхронная

передача данных. В Интернете предпочтение чаще всего отдается синхронным

транзакциям, поскольку ими легче пользоваться. При синхронной передаче данных

пользовательская программа посылает запрос и затем ожидает ответа от сервера. Но

для некоторых разновидностей высокопроизводительных приложений более подходящей

оказывается асинхронная передача данных. При таком способе передачи данных

пользовательская программа продолжает обработку данных, ожидая ответа на

переданный запрос. Но организовать асинхронную передачу данных труднее. Кроме

того, не во всех программах можно извлечь выгоды из асинхронной передачи данных.

Например, когда требуется получить информацию из Интернета, то зачастую ничего

другого не остается, как ожидать ее. В подобных случаях потенциал асинхронной

передачи данных используется не полностью. Вследствие того что синхронный доступ

к Интернету реализуется проще и намного чаще, именно он и будет рассматриваться

в этой главе.

Далее речь пойдет прежде всего о классах WebRequest и WebResponse, поскольку

именно они положены в основу сетевых программных средств, доступных в пространстве

имен System.Net.