**Класс Uri**

Как следует из табл. 26.1, метод WebRequest.Create() существует в двух вариантах.

В одном варианте он принимает идентификатор URI в виде строки. Именно этот

вариант и был использован в предыдущих примерах программ. А во втором варианте

этот метод принимает идентификатор URI в виде экземпляра объекта класса Uri,

определенного в пространстве имен System. Класс Uri инкапсулирует идентификатор

URL Используя класс Uri, можно сформировать URI, чтобы затем передать этот

идентификатор методу Create(). Кроме того, идентификатор URI можно разделить

на части. Для выполнения многих простых операций в Интернете класс Uri малопригоден.

Тем не менее он может оказаться весьма полезным в более сложных ситуациях

сетевого подключения к Интернету.

В классе Uri определяется несколько конструкторов. Ниже приведены наиболее

часто используемые конструкторы этого класса.

public Uri(string uriString)

public Uri(Uri baseUri, string relativeUri)

В первой форме конструктора объект класса Uri создается по идентификатору URI,

заданному в виде строки *uriString.* А во второй форме конструктора он создается по

относительному URI, заданному в виде строки *relativeUri* относительно абсолютного

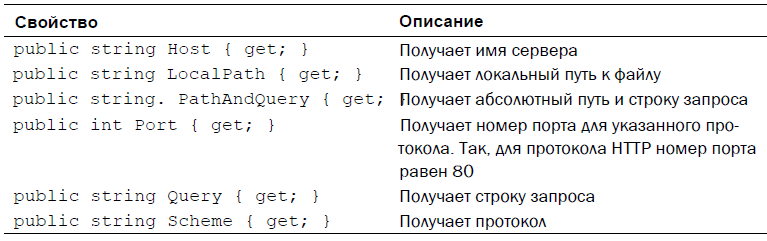
URI, обозначаемого в виде объекта *baseUri* типа Uri. Абсолютный URI определяет

полный адрес URI, а относительный URI — только путь к искомому ресурсу.

В классе Uri определяются многие поля, свойства и методы, оказывающие помощь

в управлении идентификаторами URI или в получении доступа к различным частям

URI. Особый интерес представляют приведенные ниже свойства.



Перечисленные выше свойства полезны для разделения URI на составные части. Применение

этих свойств демонстрируется в приведенном ниже примере программы.

(файла нет, ибо ссылка в коде не работает)

using System.Net;

class UriDemo

{

static void Main()

{

Uri sample = new

Uri("http://HerbSchildt.com/somefile.txt?SomeQuery");

Console.WriteLine("Хост: " + sample.Host);

Console.WriteLine("Порт: " + sample.Port);

Console.WriteLine("Протокол: " + sample.Scheme);

Console.WriteLine("Локальный путь: " + sample.LocalPath);

Console.WriteLine("Запрос: " + sample.Query);

Console.WriteLine("Путь и запрос: " + sample.PathAndQuery);

}

}

Эта программа дает следующий результат.

Хост: HerbSchildt.com

Порт: 80

Протокол: http

Локальный путь: /somefile.txt

Запрос: ?ScmeQuery

Путь и запрос: /somefile.txt?SomeQuery