

# System.Net.WebRequest Class

```
[ILASM]
.class public abstract serializable WebRequest extends
System.MarshalByRefObject

[C#]
public abstract class WebRequest: MarshalByRefObject
```

## Assembly Info:

- *Name:* System
- *Public Key:* [00 00 00 00 00 00 00 00 04 00 00 00 00 00 00]
- *Version:* 1.0.x.x
- *Attributes:*
  - CLSCompliantAttribute(true)

## Summary

Makes a request to a Uniform Resource Identifier (URI).

## Inherits From: System.MarshalByRefObject

**Library:** Networking

**Thread Safety:** All public static members of this type are safe for multithreaded operations. No instance members are guaranteed to be thread safe.

## Description

**System.Net.WebRequest** is an abstract class that models the request side of transactions used for accessing data from the Internet.

Classes that derive from **System.Net.WebRequest** are required to override the following members of the **System.Net.WebRequest** class in a protocol-specific manner:

- **System.Net.WebRequest.Method** -- Gets or sets the protocol method to use in the current instance.
- **System.Net.WebRequest.RequestUri** -- Gets the **System.Uri** of the resource associated with the current instance.
- **System.Net.WebRequest.Headers** -- Gets or sets the collection of header name/value pairs associated with the request.

- 1           • **System.Net.WebRequest.ContentLength** -- Gets or sets the  
2           content length of the request data being sent.
  
- 3           • **System.Net.WebRequest.ContentType** -- Gets or sets the  
4           content type of the request data being sent.
  
- 5           • **System.Net.WebRequest.Credentials** -- Gets or sets the  
6           credentials used for authenticating the client using the current  
7           instance.
  
- 8           • **System.Net.WebRequest.PreAuthenticate** -- Gets or sets a  
9           value that indicates whether to send authentication information  
10          with a request for resources.
  
- 11          • **System.Net.WebRequest.GetRequestStream** -- Returns a  
12          **System.IO.Stream** for writing data to a resource.
  
- 13          • **System.Net.WebRequest.BeginGetRequestStream** --  
14          Begins an asynchronous request for a stream in which to write  
15          data to be sent in the current request.
  
- 16          • **System.Net.WebRequest.EndGetRequestStream** --  
17          Returns a **System.IO.Stream** for writing data to the resource  
18          accessed by the current instance.
  
- 19          • **System.Net.WebRequest.GetResponse** -- Returns a  
20          response to a request.
  
- 21          • **System.Net.WebRequest.BeginGetResponse** -- Begins an  
22          asynchronous request for a resource.
  
- 23          • **System.Net.WebRequest.EndGetResponse** -- Returns a  
24          **System.Net.WebResponse** that contains a response to a  
25          specified pending request.

26           In addition, derived classes are required to support the  
27           **System.Net.IWebRequestCreate** interface.

28  
29           [*Note:* An application that uses the request/response model can  
30           request data be sent from the Internet in a protocol-agnostic manner,  
31           in which the application works with instances of the  
32           **System.Net.WebRequest** class while classes that derive from  
33           **System.Net.WebRequest** and implement specific protocols perform  
34           the details of the request.

35  
36           Requests are sent from an application to a particular Uniform Resource  
37           Identifier (URI), such as a Web page on a server. Using the URI, the  
38           **System.Net.WebRequest.Create** method creates an instance of a  
39           type derived from **System.Net.WebRequest** to handle the request.  
40           The type is selected from the set of registered types. Types may be  
41           registered to handle a specific protocol, such as HTTP or FTP, or to

1 handle a request to a specific server or path on a server. [Note: For  
2 information on registering types, see  
3 **System.Net.WebRequest.RegisterPrefix.**]

4  
5 The **System.Net.WebRequest** class throws a  
6 **System.Net.WebException** exception when an error occurs while  
7 accessing a resource.

8  
9 Use the **System.Net.WebRequest.Create** method to initialize a new  
10 instance of a class that derives from **System.Net.WebRequest**. Do  
11 not use the **System.Net.WebRequest** constructor.]

## 12 Example

13

14 The following example demonstrates using  
15 **System.Net.WebRequest.Create** to create an instance of  
16 **System.Net.HttpWebRequest**.

17

18

```
[C#]
```

19

```
using System;
```

20

```
using System.Net;
```

21

```
public class WebRequestExample {
```

22

```
    public static void Main() {
```

23

24

```
        // Initialize the WebRequest.
```

25

```
        WebRequest myRequest =
```

26

```
            WebRequest.Create("http://www.contoso.com");
```

27

28

```
        // Print the type of the request.
```

29

```
        Console.WriteLine(myRequest);
```

30

31

```
    }
```

32

```
}
```

33

34

The output is

35

36

System.Net.HttpWebRequest

37

# 1 WebRequest() Constructor

```
2 [ILASM]  
3 family rtspecialname specialname instance void .ctor()  
4 [C#]  
5 protected WebRequest()
```

## 6 Summary

7 Constructs a new instance of the **System.Net.WebRequest** class.

## 8 Description

9 This constructor is called only by classes that derive from  
10 **System.Net.WebRequest**.

11  
12 [Note: Use the **System.Net.WebRequest.Create** method to initialize  
13 a new instance of a class that derives from  
14 **System.Net.WebRequest**. Do not use this constructor.]

15

# 1 WebRequest.Abort() Method

```
2 [ILASM]  
3 .method public hidebysig virtual void Abort()  
4 [C#]  
5 public virtual void Abort()
```

## 6 Summary

7 Attempts to cancel an asynchronous request made by the current  
8 instance to access a resource.

## 9 Behaviors

10 As described above.

## 11 Default

12 The **System.Net.WebRequest** class is abstract and does not provide  
13 an implementation for this method. This method throws  
14 **System.NotSupportedException**.

## 15 How and When to Override

16 This method must be overridden by classes that inherit from  
17 **System.Net.WebRequest** to provide this functionality.

## 18 Usage

19 Use this method to cancel an asynchronous operation started with the  
20 **System.Net.WebRequest.BeginGetResponse** method.

## 21 Exceptions

22  
23

Exception	Condition
<b>System.NotSupportedException</b>	This method is not overridden in the derived class.

24  
25  
26

# 1 WebRequest.BeginGetRequestStream(System.AsyncCallback, System.Object)

## 3 Method

```
4 [ILASM]
5 .method public hidebysig virtual class System.IAsyncResult
6 BeginGetRequestStream(class System.AsyncCallback callback,
7 object state)
8
9 [C#]
10 public virtual IAsyncResult
BeginGetRequestStream(AsyncCallback callback, object state)
```

### 11 Summary

12 Begins an asynchronous request for a stream in which to write data to  
13 be sent in the current request.

### 14 Parameters

15  
16

Parameter	Description
<i>callback</i>	A <b>System.AsyncCallback</b> delegate to be called when the stream is available. Can be <b>null</b> .
<i>state</i>	A <b>System.Object</b> containing state information for the asynchronous request.

17  
18  
19

### Return Value

20 A **System.IAsyncResult** object that contains information about the  
21 asynchronous operation.

### 22 Description

23 The *state* parameter can be any object that the caller wishes to have  
24 available for the duration of the asynchronous operation. This object is  
25 available via the **System.IAsyncResult.AsyncState** property of the  
26 object returned by this method.

### 27 Behaviors

28 This method starts an asynchronous operation to obtain a stream used  
29 to write data to be sent in the current request. To get the request  
30 stream, call the **System.Net.WebRequest.EndGetRequestStream**  
31 method and specify the **System.IAsyncResult** object returned by this  
32 method.

1  
2 If the *callback* parameter is not **null**, the method referenced by  
3 *callback* is invoked when the asynchronous operation completes. The  
4 **System.IAsyncResult** object returned by this method is passed as  
5 the argument to the method referenced by *callback*.

## 6 **Default**

7 The **System.Net.WebRequest** class is abstract and does not provide  
8 an implementation for this method. This method throws  
9 **System.NotSupportedException**.

## 10 **How and When to Override**

11 This method must be overridden by classes that inherit from  
12 **System.Net.WebRequest** to provide this functionality.

## 13 **Usage**

14 Use this method to start an asynchronous request for a stream used to  
15 send data to a resource. The *callback* delegate can call the  
16 **System.Net.WebRequest.EndGetRequestStream** method to obtain  
17 the request stream.

## 18 **Exceptions**

19  
20

Exception	Condition
<b>System.NotSupportedException</b>	This method is not overridden in the derived class.

21  
22  
23

# 1 WebRequest.BeginGetResponse(System.A 2 syncCallback, System.Object) Method

```
3 [ILASM]  
4 .method public hidebysig virtual class System.IAsyncResult  
5 BeginGetResponse(class System.AsyncCallback callback,  
6 object state)  
  
7 [C#]  
8 public virtual IAsyncResult BeginGetResponse(AsyncCallback  
9 callback, object state)
```

## 10 Summary

11 Begins sending the current request asynchronously.

## 12 Parameters

13  
14

Parameter	Description
<i>callback</i>	A <b>System.AsyncCallback</b> delegate to be called when the response from the server is available.
<i>state</i>	A <b>System.Object</b> containing state information for the asynchronous request.

15  
16  
17

## 16 Return Value

18 A **System.IAsyncResult** object that contains information about the  
19 asynchronous operation.

## 20 Description

21 The *state* parameter can be any object that the caller wishes to have  
22 available for the duration of the asynchronous operation. This object is  
23 available via the **System.IAsyncResult.AsyncState** property of the  
24 object returned by this method.

## 25 Behaviors

26 This method starts an asynchronous operation to send the current  
27 request and receive the response from the server that processed the  
28 request. To get the response, call the  
29 **System.Net.WebRequest.EndGetResponse** method and specify the  
30 **System.IAsyncResult** object returned by this method.

31  
32 If the *callback* parameter is not **null**, the method referenced by  
33 *callback* is invoked when the asynchronous operation completes. The

1 **System.IAsyncResult** object returned by this method is passed as  
2 the argument to the method referenced by *callback*.

### 3 **Default**

4 The **System.Net.WebRequest** class is abstract and does not provide  
5 an implementation for this method. This method throws  
6 **System.NotSupportedException**.

### 7 **How and When to Override**

8 This method must be overridden by classes that inherit from  
9 **System.Net.WebRequest** to provide this functionality.

### 10 **Usage**

11 The **System.Net.WebRequest.BeginGetResponse** method starts an  
12 asynchronous request for a response. The callback delegate can call  
13 the **System.Net.WebRequest.EndGetResponse** method to return  
14 the **System.Net.WebResponse** received from the resource.

### 15 **Exceptions**

16  
17

Exception	Condition
<b>System.NotSupportedException</b>	This method is not overridden in the derived class.

18  
19  
20

# 1 WebRequest.Create(System.String)

## 2 Method

```
3 [ILASM]
4 .method public hidebysig static class System.Net.WebRequest
5 Create(string requestUriString)
6
7 [C#]
8 public static WebRequest Create(string requestUriString)
```

### 8 Summary

9 Constructs a new instance of a class derived from  
10 **System.Net.WebRequest**. The new instance is of the type registered  
11 for the scheme of the specified URI.

### 12 Parameters

13  
14

Parameter	Description
<i>requestUriString</i>	A <b>System.String</b> that contains a URI.

15  
16  
17

### 16 Return Value

18 A new instance of a class that derived from  
19 **System.Net.WebRequest** and is registered to handle the scheme of  
20 *requestUriString*.

### 21 Description

22 [Note: This method returns a new instance of a class that derived from  
23 **System.Net.WebRequest**. The **System.Type** of this new instance is  
24 determined at run time by the scheme of the URI in *requestUriString*.  
25 For example, when a URI beginning with `http://` is passed in  
26 *requestUriString*, a **System.Net.HttpWebRequest** instance is  
27 returned.

28  
29  
30  
31

Classes that derive from **System.Net.WebRequest** that are created  
to handle other requests are registered with the  
**System.Net.WebRequest.RegisterPrefix** method.]

### 32 Exceptions

33  
34

Exception	Condition
<b>System.ArgumentNullException</b>	<i>requestUriString</i> is <b>null</b> .

<b>System.NotSupportedException</b>	The request scheme specified in <i>requestUri</i> is not registered.
<b>System.UriFormatException</b>	The URI specified in <i>requestUriString</i> is not a valid URI.
<b>System.Security.SecurityException</b>	The caller does not have permission to connect to the requested URI or a URI that the request is redirected to.

1  
2  
3  
4

## Permissions

Permission	Description
<b>System.Security.Permissions.WebPermission</b>	Requires permission to connect to the requested URI. See <b>System.Net.NetworkAccess.Connect</b> .

5  
6  
7

# 1 WebRequest.Create(System.Uri) Method

```
2 [ILASM]  
3 .method public hidebysig static class System.Net.WebRequest  
4 Create(class System.Uri requestUri)  
5  
6 [C#]  
7 public static WebRequest Create(Uri requestUri)
```

## 7 Summary

8 Constructs a new instance of a class derived from  
9 **System.Net.WebRequest**.

## 10 Parameters

11  
12

Parameter	Description
<i>requestUri</i>	A <b>System.Uri</b> containing the URI of the requested resource.

13  
14  
15

## 14 Return Value

16 A new instance of a class derived from **System.Net.WebRequest**  
17 that is registered to handle the closest registered match for *requestUri*.

## 18 Description

19 To determine the closest match, this method checks the registered  
20 URIs for the longest URI prefix that matches *requestUri*.

21  
22 [Note: For an example that demonstrates this method, see  
23 **System.Net.WebRequest.CreateDefault**.]

## 24 Exceptions

25  
26

Exception	Condition
<b>System.ArgumentNullException</b>	<i>requestUri</i> is <b>null</b> .
<b>System.NotSupportedException</b>	The request scheme specified in <i>requestUri</i> is not registered.
<b>System.Security.SecurityException</b>	The caller does not have permission to connect to the requested URI or a URI that the request is redirected to.

27  
28

## 28 Permissions

1  
2

Permission	Description
<b>System.Security.Permissions.WebPermission</b>	Requires permission to connect to the requested URI. See <b>System.Net.NetworkAccess.Connect</b> .

3  
4  
5

# 1 WebRequest.CreateDefault(System.Uri)

## 2 Method

```
3 [ILASM]  
4 .method public hidebysig static class System.Net.WebRequest  
5 CreateDefault(class System.Uri requestUri)  
  
6 [C#]  
7 public static WebRequest CreateDefault(Uri requestUri)
```

### 8 Summary

9 Constructs a new instance of a class derived from  
10 **System.Net.WebRequest**. The new instance is of the type registered  
11 for the scheme of the specified URI.

### 12 Parameters

13  
14

Parameter	Description
<i>requestUri</i>	A <b>System.Uri</b> containing the URI of the requested resource.

15  
16  
17

### 16 Return Value

18 A new instance of the type derived from **System.Net.WebRequest**  
19 that is registered for the scheme of the specified **System.Uri**.

### 20 Description

21 [Note: When this method is invoked, only the scheme portion of  
22 *requestUri* is checked against the list of URIs registered for the current  
23 instance. Conversely, when **System.Net.WebRequest.Create** is  
24 invoked, the entire URI is checked against the list of registered URIs.]

### 25 Exceptions

26  
27

Exception	Condition
<b>System.ArgumentNullException</b>	<i>requestUri</i> is <b>null</b> .
<b>System.NotSupportedException</b>	The request scheme specified in <i>requestUri</i> is not registered.
<b>System.Security.SecurityException</b>	The caller does not have permission to connect to the requested URI or a URI that the request is redirected to.

1  
2 **Example**  
3

4 This example demonstrates the use of the  
5 **System.Net.WebRequest.Create** and  
6 **System.Net.WebRequest.CreateDefault** methods.  
7

```
8 [C#]
9 using System;
10 using System.Net;
11
12 public class ContosoTextRequest: WebRequest,
13 IWebRequestCreate
14 {
15     public new WebRequest Create(Uri uri)
16     {
17         return new ContosoTextRequest();
18     }
19 }
20
21 public class CreateDefaultExample
22 {
23     public static void Main()
24     {
25         ContosoTextRequest contoso = new
26 ContosoTextRequest();
27         Uri contosoUri = new
28 Uri("http://www.contoso.com/text");
29
30 WebRequest.RegisterPrefix("http://www.contoso.com/text",
31 contoso);
32
33         WebRequest httpContoso =
34 WebRequest.CreateDefault(contosoUri);
35         Console.WriteLine("CreateDefault --> {0}",
36 httpContoso);
37
38         WebRequest textContoso =
39 WebRequest.Create(contosoUri);
40         Console.WriteLine("Create --> {0}", textContoso);
41     }
42 }
```

43 The output is  
44  
45 CreateDefault --> System.Net.HttpWebRequest  
46  
47

1           Create --> ContosoTextRequest

2

3 **Permissions**

4

5

Permission	Description
<b>System.Security.Permissions.WebPermission</b>	Requires permission to connect to the requested URI. See <b>System.Net.NetworkAccess.Connect</b> .

6

7

8

# 1 WebRequest.EndGetRequestStream(System.IAsyncResult) Method

```
3 [ILASM]
4 .method public hidebysig virtual class System.IO.Stream
5 EndGetRequestStream(class System.IAsyncResult asyncResult)
6
7 [C#]
8 public virtual Stream EndGetRequestStream(IAsyncResult
9 asyncResult)
```

## 9 Summary

10 Returns a **System.IO.Stream** for writing data to the resource  
11 identified by the **System.Net.WebRequest.RequestUri** property of  
12 the current instance.

## 13 Parameters

Parameter	Description
<i>asyncResult</i>	A <b>System.IAsyncResult</b> object that references a request for a <b>System.IO.Stream</b> started with <b>System.Net.WebRequest.BeginGetRequestStream</b> .

## 16 Return Value

17 A **System.IO.Stream** to write data to.

## 20 Description

21 This method completes an asynchronous request for a stream that was  
22 started by the **System.Net.WebRequest.BeginGetRequestStream**  
23 method.

## 24 Behaviors

25 As described above.

## 26 Default

27 The **System.Net.WebRequest** class is abstract and does not provide  
28 an implementation for this method. This method throws  
29 **System.NotSupportedException**.

## 30 How and When to Override

1 This method must be overridden by classes that inherit from  
2 **System.Net.WebRequest** to provide this functionality.

### 3 Usage

4 Use this method to complete an asynchronous request for a stream  
5 that was started with the  
6 **System.Net.WebRequest.BeginGetRequestStream** method.

### 7 Exceptions

8  
9

Exception	Condition
<b>System.NotSupportedException</b>	This method is not overridden in the derived class.
<b>System.ArgumentException</b>	<i>asyncResult</i> was not returned by a call to <b>System.Net.WebRequest.BeginGetRequestStream</b> .
<b>System.ArgumentNullException</b>	<i>asyncResult</i> is a null reference.
<b>System.InvalidOperationException</b>	This method was called previously using <i>asyncResult</i> . -or- No stream is available.
<b>System.Net.WebException</b>	An error occurred while processing the request.

10  
11  
12

# 1 WebRequest.EndGetResponse(System.IAsyncResult) Method

```
3 [ILASM]
4 .method public hidebysig virtual class
5 System.Net.WebResponse EndGetResponse(class
6 System.IAsyncResult asyncResult)
7
8 [C#]
9 public virtual WebResponse EndGetResponse(IAsyncResult
  asyncResult)
```

## 10 Summary

11 Returns a **System.Net.WebResponse** that contains a response to a  
12 specified pending request.

## 13 Parameters

Parameter	Description
<i>asyncResult</i>	A <b>System.IAsyncResult</b> object that references a pending request that was started with <b>System.Net.WebRequest.BeginGetResponse</b> .

## 16 Return Value

19 A **System.Net.WebResponse** that contains a response to the request  
20 referenced by *asyncResult*.

## 21 Behaviors

22 As described above.

## 23 Default

24 The **System.Net.WebRequest** class is abstract and does not provide  
25 an implementation for this method. This method throws  
26 **System.NotSupportedException**.

## 27 How and When to Override

28 This method must be overridden by classes that inherit from  
29 **System.Net.WebRequest** to provide this functionality.

## 30 Usage

1 Use this method to complete an asynchronous request for an Internet  
2 resource that was started with the  
3 **System.Net.WebRequest.BeginGetResponse** method.

4 **Exceptions**

5  
6

Exception	Condition
<b>System.NotSupportedException</b>	This method is not overridden in the derived class.
<b>System.ArgumentException</b>	<i>asyncResult</i> was not returned by a call to <b>System.Net.WebRequest.BeginGetResponse</b> .
<b>System.ArgumentNullException</b>	<i>asyncResult</i> is a null reference.
<b>System.InvalidOperationException</b>	The <b>System.Net.WebRequest.ContentLength</b> property of the current instance is greater than zero but no data has been written to the request stream. -or- This method was called previously using <i>asyncResult</i> .
<b>System.Net.WebException</b>	An error occurred while processing the request.

7  
8  
9

# 1 WebRequest.GetRequestStream() Method

```
2 [ILASM]  
3 .method public hidebysig virtual class System.IO.Stream  
4 GetRequestStream()  
5 [C#]  
6 public virtual Stream GetRequestStream()
```

## 7 Summary

8 Returns a **System.IO.Stream** for writing data to a resource.

## 9 Return Value

11 A **System.IO.Stream** for writing data to a resource.

## 12 Behaviors

13 As described above.

## 14 Default

15 The **System.Net.WebRequest** class is abstract and does not provide  
16 an implementation for this method. This method throws  
17 **System.NotSupportedException**.

## 18 How and When to Override

19 This method is required to be overridden by classes that inherit from  
20 **System.Net.WebRequest**.

## 21 Usage

22 Use this method to initiate a request to send data to a resource and  
23 obtain a **System.IO.Stream** instance for sending data to that  
24 resource.

25  
26 The **System.Net.WebRequest.GetRequestStream** method provides  
27 synchronous access to the **System.IO.Stream**. For asynchronous  
28 access, use the  
29 **System.Net.WebRequest.BeginGetRequestStream** and  
30 **System.Net.WebRequest.EndGetRequestStream** methods.

## 31 Exceptions

Exception	Condition
-----------	-----------

1  
2  
3

<b>System.NotSupportedException</b>	This method is not overridden in the derived class.
-------------------------------------	---

# 1 `WebRequest.GetResponse()` Method

```
2 [ILASM]  
3 .method public hidebysig virtual class  
4 System.Net.WebResponse GetResponse()  
  
5 [C#]  
6 public virtual WebResponse GetResponse()
```

## 7 **Summary**

8 Returns a response to a request.

## 9 **Return Value**

10

11 A **System.Net.WebResponse** containing the response to the request.

## 12 **Behaviors**

13 This method returns an instance of a type derived from  
14 **System.Net.WebResponse** that is registered for the  
15 **System.Net.WebRequest.RequestUri** property of the current  
16 instance. This new instance is required to contain a response from the  
17 resource to the current request.

18  
19 If the timeout period for the request expires, or an error occurs while  
20 processing the request, this method is required to throw a  
21 **System.Net.WebException** exception.

## 22 **Default**

23 The **System.Net.WebRequest** class is abstract and does not provide  
24 an implementation for this method. This method throws  
25 **System.NotSupportedException**.

## 26 **How and When to Override**

27 This method must be overridden by classes that inherit from  
28 **System.Net.WebRequest** to provide this functionality.

## 29 **Usage**

30 Use this method for synchronous access to a resource. For  
31 asynchronous access, use the  
32 **System.Net.WebRequest.BeginGetResponse** and  
33 **System.Net.WebRequest.EndGetResponse** methods.

1 **Exceptions**  
2  
3

<b>Exception</b>	<b>Condition</b>
<b>System.NotSupportedException</b>	This method is not overridden in the derived class.
<b>System.Net.WebException</b>	The request timed out.  -or-  An error occurred while processing the request.

4  
5  
6

# 1 `WebRequest.RegisterPrefix(System.String` 2 `, System.Net.IWebRequestCreate) Method`

```
3 [ILASM]  
4 .method public hidebysig static bool RegisterPrefix(string  
5 prefix, class System.Net.IWebRequestCreate creator)  
  
6 [C#]  
7 public static bool RegisterPrefix(string prefix,  
8 IWebRequestCreate creator)
```

## 9 Summary

10 Registers a type derived from **System.Net.WebRequest**, and  
11 associates the type with the specified URI.

## 12 Parameters

13  
14

Parameter	Description
<i>prefix</i>	A <b>System.String</b> containing the URI that the derived type services. Can specify a scheme or a complete URI.
<i>creator</i>	An instance of a type that implements the <b>System.Net.IWebRequestCreate</b> interface.

15  
16  
17

## 16 Return Value

18 **true** if registration is successful; **false**, if *prefix* is already registered.

## 19 Description

20 **System.Net.HttpWebRequest** is registered to service requests for  
21 HTTP and HTTPS schemes. Attempts to register a different type for  
22 these schemes will fail.

23  
24 [Note: This method registers types that derive from  
25 **System.Net.WebRequest** to service requests. These derived types  
26 are typically registered to handle a specific protocol, such HTTP or FTP,  
27 but can be registered to handle a request to a specific server or path  
28 on a server. Therefore, *prefix* can be either a scheme or a complete  
29 URI.

30  
31 The **System.Net.WebRequest** class calls the  
32 **System.Net.IWebRequestCreate.Create** method to create  
33 additional instances of the same type as *creator*.]

1 **Exceptions**

2

3

Exception	Condition
<b>System.ArgumentNullException</b>	<i>prefix is null or creator is null.</i>

4

5 **Example**

6

7 The following example demonstrates how to register a new scheme.

8

9

[C#]

10

```
using System;
```

11

```
using System.Net;
```

12

```
public class ftpWebRequest: WebRequest {
```

13

```
    //implement ftp-specific protocol methods and properties
```

14

```
}
```

15

16

```
public class ftpCreator: IWebRequestCreate
```

17

```
{
```

18

```
    public WebRequest Create(Uri uri)
```

19

```
{
```

20

```
    return new ftpWebRequest();
```

21

```
}
```

22

```
}
```

23

24

```
public class RegisterPrefixExample
```

25

```
{
```

26

27

```
    public static void Main()
```

28

```
{
```

29

```
        ftpCreator creator = new ftpCreator();
```

30

```
        WebRequest.RegisterPrefix("ftp://", creator);
```

31

```
        WebRequest wr = WebRequest.Create("ftp://testFile");
```

32

```
        Console.WriteLine(wr);
```

33

```
}
```

34

```
}
```

35

36

37

38

The output is

39

40

ftpWebRequest

41

# 1 **WebRequest.ConnectionGroupName**

## 2 **Property**

```
3 [ILASM]
4 .property string ConnectionGroupName { public hidebysig
5 virtual specialname string get_ConnectionGroupName() public
6 hidebysig virtual specialname void
7 set_ConnectionGroupName(string value) }
8
9 [C#]
10 public virtual string ConnectionGroupName { get; set; }
```

### 10 **Summary**

11 Gets or sets the name of the connection group for the current  
12 instance.

### 13 **Property Value**

15 A **System.String** that contains the name of the connection group for  
16 the current instance.

### 17 **Description**

18 This property associates specific requests within an application with a  
19 **System.Net.ServicePoint**.

### 20 **Behaviors**

21 As described above.

### 22 **Default**

23 This property throws **System.NotSupportedException**.

### 24 **How and When to Override**

25 This property is required to be overridden by classes that inherit from  
26 **System.Net.WebRequest**. The  
27 **System.Net.WebRequest.ConnectionGroupName** property  
28 typically associates a group of requests that share a set of credentials  
29 with a connection to an Internet resource to avoid potential security  
30 failures.

### 31 **Usage**

32 Use this property to get or set the name of the connection group for  
33 the current instance.

1 **Exceptions**

2

3

<b>Exception</b>	<b>Condition</b>
<b>System.NotSupportedException</b>	This property is not implemented in the derived class.

4

5

6

# WebRequest.ContentLength Property

```
[ILASM]
.property int64 ContentLength { public hidebysig virtual
specialname int64 get_ContentLength() public hidebysig
virtual specialname void set_ContentLength(int64 value) }

[C#]
public virtual long ContentLength { get; set; }
```

## Summary

Gets or sets the content length of the request data being sent.

## Property Value

A **System.Int64** containing the number of bytes of request data being sent.

## Behaviors

This property is required to throw a **System.InvalidOperationException** exception if data has already been written to the request stream, and a **System.ArgumentOutOfRangeException** exception if the property is being set to a value less than zero.

## Default

This property throws **System.NotSupportedException**.

## How and When to Override

This property is required to be overridden by classes that inherit from **System.Net.WebRequest**.

## Usage

Use this property to get the number of bytes sent to the resource.

## Exceptions

Exception	Condition
<b>System.NotSupportedException</b>	This property is not implemented in the derived class.
<b>System.InvalidOperationException</b>	Data has already been written to the

1  
2  
3

	request stream.
<b>System.ArgumentOutOfRangeException</b>	This property is being set to a value less than zero.

# 1 WebRequest.ContentType Property

```
2 [ILASM]
3 .property string ContentType { public hidebysig virtual
4 specialname string get_ContentType() public hidebysig
5 virtual specialname void set_ContentType(string value) }
6
7 [C#]
8 public virtual string ContentType { get; set; }
```

## 8 Summary

9 Gets or sets the content type of the request data being sent.

## 10 Property Value

11  
12 A **System.String** that represents the content type of the request  
13 data.

## 14 Description

15 The **System.Net.WebRequest.ContentType** property contains the  
16 media type of the request.

17  
18 [*Note:* This is typically the MIME encoding of the content.]

## 19 Behaviors

20 As described above.

## 21 Default

22 This property throws **System.NotSupportedException**.

## 23 How and When to Override

24 This property is required to be overridden by classes that inherit from  
25 **System.Net.WebRequest**.

## 26 Usage

27 Use this property to get the media type of request.

## 28 Exceptions

Exception	Condition
-----------	-----------

1  
2  
3

<b>System.NotSupportedException</b>	This property is not implemented in the derived class.
-------------------------------------	--

# 1 WebRequest.Credentials Property

```
2 [ILASM]
3 .property class System.Net.ICredentials Credentials {
4 public hidebysig virtual specialname class
5 System.Net.ICredentials get_Credentials() public hidebysig
6 virtual specialname void set_Credentials(class
7 System.Net.ICredentials value) }
8
9 [C#]
public virtual ICredentials Credentials { get; set; }
```

## 10 Summary

11 Gets or sets the credentials used for authenticating the client using the  
12 current instance.

## 13 Property Value

14

15 A **System.Net.ICredentials** object containing the authentication  
16 credentials associated with the request. The default is **null**.

## 17 Behaviors

18 As described above.

## 19 Default

20 This property throws **System.NotSupportedException**.

## 21 How and When to Override

22 This property is required to be overridden by classes that inherit from  
23 **System.Net.WebRequest**.

## 24 Usage

25 Use this property to store or access the user, password, and domain  
26 information of the current instance.

## 27 Exceptions

28

29

Exception	Condition
<b>System.NotSupportedException</b>	This property is not implemented in the derived class.

1  
2  
3

# 1 WebRequest.Headers Property

```
2 [ILASM]
3 .property class System.Net.WebHeaderCollection Headers {
4 public hidebysig virtual specialname class
5 System.Net.WebHeaderCollection get_Headers() public
6 hidebysig virtual specialname void set_Headers(class
7 System.Net.WebHeaderCollection value) }
8
9 [C#]
public virtual WebHeaderCollection Headers { get; set; }
```

## 10 Summary

11 Gets or sets the collection of header name/value pairs associated with  
12 the request.

## 13 Property Value

14

15 A **System.Net.WebHeaderCollection** containing the header  
16 name/value pairs associated with the current instance.

## 17 Description

18 This property contains a **System.Net.WebHeaderCollection** instance  
19 containing the header information to send to resources.

## 20 Behaviors

21 As described above.

## 22 Default

23 This property throws a **System.NotSupportedException** exception.

## 24 How and When to Override

25 This property must be overridden by classes that inherit from  
26 **System.Net.WebRequest**.

## 27 Usage

28 Use this property to determine the header information of a request.

## 29 Exceptions

30

31

Exception	Condition
-----------	-----------

1  
2  
3

<b>System.NotSupportedException</b>	This property is not implemented in the derived class.
-------------------------------------	--

# 1 WebRequest.Method Property

```
2 [ILASM]
3 .property string Method { public hidebysig virtual
4 specialname string get_Method() public hidebysig virtual
5 specialname void set_Method(string value) }
6
7 [C#]
8 public virtual string Method { get; set; }
```

## 8 Summary

9 Gets or sets the protocol method to use in the current instance.

## 10 Property Value

11

12 A **System.String** containing the protocol method to use in the current  
13 instance.

## 14 Behaviors

15 The default value of this property is required to be a protocol method  
16 that does not require protocol-specific properties to be set. For the  
17 HTTP protocol, this value is GET.

## 18 Default

19 This property throws **System.NotSupportedException**.

## 20 How and When to Override

21 This property must be overridden by classes that inherit from  
22 **System.Net.WebRequest** to provide this functionality.

## 23 Usage

24 Use this property to set the protocol-specific method that will be used  
25 to make a request.

## 26 Exceptions

27

28

Exception	Condition
<b>System.NotSupportedException</b>	This property is not implemented in the derived class.

29

30

31

# 1 WebRequest.PreAuthenticate Property

```
2 [ILASM]
3 .property bool PreAuthenticate { public hidebysig virtual
4 specialname bool get_PreAuthenticate() public hidebysig
5 virtual specialname void set_PreAuthenticate(bool value) }
6 [C#]
7 public virtual bool PreAuthenticate { get; set; }
```

## 8 Summary

9 Gets or sets a **System.Boolean** value that determines whether to  
10 send authentication information with the current request instead of  
11 waiting for an authentication challenge from the requested resource.

## 12 Property Value

14 **true** if authentication information will be sent with the current request  
15 without waiting for an authentication challenge from the requested  
16 resource; otherwise, **false**.

## 17 Behaviors

18 If **System.Net.WebRequest.PreAuthenticate** is **true**, the current  
19 instance sends authentication credentials without waiting to be  
20 challenged by the server specified by the  
21 **System.Net.WebRequest.RequestUri** property of the current  
22 instance. When this property is **false**, the current instance waits for a  
23 challenge from the server before sending credentials.

## 24 Default

25 This property throws **System.NotSupportedException**.

## 26 How and When to Override

27 This property must be overridden by classes that inherit from  
28 **System.Net.WebRequest** to provide this functionality.

## 29 Usage

30 Use this property to ensure that authentication information is sent with  
31 every request. Setting this property to **true** allows clients to improve  
32 server efficiency by avoiding extra round trips caused by  
33 authentication challenges.

1 **Exceptions**

2

3

<b>Exception</b>	<b>Condition</b>
<b>System.NotSupportedException</b>	This property is not implemented in the derived class.

4

5

6

# 1 WebRequest.Proxy Property

```
2 [ILASM]
3 .property class System.Net.IWebProxy Proxy { public
4 hidebysig virtual specialname class System.Net.IWebProxy
5 get_Proxy() public hidebysig virtual specialname void
6 set_Proxy(class System.Net.IWebProxy value) }
7
8 [C#]
9 public virtual IWebProxy Proxy { get; set; }
```

## 9 Summary

10 Gets or sets the network proxy to use to access resources.

## 11 Property Value

13 A **System.Net.IWebProxy** to use to access resources.

## 14 Description

15 The **System.Net.WebRequest.Proxy** property identifies the network  
16 proxy that the request uses to access resources. The request is made  
17 through the proxy server rather than directly to the server hosting the  
18 resource.

## 19 Behaviors

20 If the **System.Net.WebRequest.Proxy** property of the current  
21 instance has not been set, the value of this property is required to be  
22 **null**.

24 If the property is being set to **null**, it is required to throw a  
25 **System.ArgumentNullException** exception.

## 26 Default

27 This property throws **System.NotSupportedException**.

## 28 How and When to Override

29 This property must be overridden by classes that inherit from  
30 **System.Net.WebRequest** to provide this functionality.

## 31 Usage

32 Use this method to obtain a **System.Net.IWebProxy** instance that  
33 represents the proxy server used by the current instance.

1 **Exceptions**

2

3

<b>Exception</b>	<b>Condition</b>
<b>System.NotSupportedException</b>	This property is not implemented in the derived class.

4

5

6

# 1 WebRequest.RequestUri Property

```
2 [ILASM]
3 .property class System.Uri RequestUri { public hidebysig
4 virtual specialname class System.Uri get_RequestUri() }
5 [C#]
6 public virtual Uri RequestUri { get; }
```

## 7 Summary

8 Gets the **System.Uri** of the resource associated with the current  
9 instance.

## 10 Property Value

11

12 A **System.Uri** containing the URI of the resource associated with the  
13 current instance

## 14 Description

15 This property is read-only.

## 16 Behaviors

17 **System.Net.WebRequest.RequestUri** is required to contain the URI  
18 passed to the **System.Net.WebRequest.Create** methods. If the  
19 protocol implemented by a derived class supports redirection, the  
20 derived class is required to provide a property to contain the URI that  
21 actually services the request.

## 22 Default

23 This property throws a **System.NotSupportedException** exception.

## 24 How and When to Override

25 This property must be overridden by classes that inherit from  
26 **System.Net.WebRequest** to provide this functionality.]

## 27 Usage

28 Use this property to determine the URI that the request was addressed  
29 to. For information about the URI that actually serviced the request,  
30 see **System.Net.WebResponse.ResponseUri**.

1 **Exceptions**

2

3

<b>Exception</b>	<b>Condition</b>
<b>System.NotSupportedException</b>	This property is not implemented in the derived class.

4

5

6

# 1 WebRequest.Timeout Property

```
2 [ILASM]
3 .property int32 Timeout { public hidebysig virtual
4 specialname int32 get_Timeout() public hidebysig virtual
5 specialname void set_Timeout(int32 value) }
6
7 [C#]
8 public virtual int Timeout { get; set; }
```

## 8 Summary

9 Gets or sets the length of time before requests for resources time out.

## 10 Property Value

11  
12 A **System.Int32** containing the length of time, in milliseconds, before  
13 the current request will time out, or  
14 **System.Threading.Timeout.Infinite** to indicate that the request  
15 does not time out.

## 16 Behaviors

17 Classes that derive from **System.Net.WebRequest** are required to  
18 indicate a timeout by throwing a **System.Net.WebException** with the  
19 **System.Net.WebException.Status** field set to  
20 **System.Net.WebExceptionStatus.Timeout** if a request times out.

## 21 Default

22 This property throws a **System.NotSupportedException** exception.

## 23 How and When to Override

24 This property must be overridden by classes that inherit from  
25 **System.Net.WebRequest** to provide this functionality.

## 26 Usage

27 Use this property to set the timeout period for requests for resources.

28  
29 The **System.Net.WebRequest.Timeout** property affects only  
30 synchronous requests made with the  
31 **System.Net.WebRequest.GetResponse** method. To time out  
32 asynchronous requests, use the **System.Net.WebRequest.Abort**  
33 method.

1 **Exceptions**

2

3

Exception	Condition
<b>System.NotSupportedException</b>	This property is not implemented in the derived class.

4

5