

# 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 [C#]  
18  
19 using System;  
20 using System.Net;  
21  
22 public class WebRequestExample {  
23  
24     public static void Main() {  
25  
26         // Initialize the WebRequest.  
27         WebRequest myRequest =  
28             WebRequest.Create("http://www.contoso.com");  
29  
30         // Print the type of the request.  
31         Console.WriteLine(myRequest);  
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  
5 [C#]  
6 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

# WebRequest.Abort() Method

```
[ILASM]
.method public hidebysig virtual void Abort()

[C#]
public virtual void Abort()
```

## Summary

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

## Behaviors

As described above.

## Default

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

## How and When to Override

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

## Usage

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

## Exceptions

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

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

## Method

```
[ILASM]
.method public hidebysig virtual class System.IAsyncResult
BeginGetRequestStream(class System.AsyncCallback callback,
object state)

[C#]
public virtual IAsyncResult
BeginGetRequestStream(AsyncCallback callback, object state)
```

## Summary

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

## Parameters

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.

## Return Value

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

## Description

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

## Behaviors

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

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

# WebRequest.BeginGetResponse(System.AsyncCallback, System.Object) Method

```
[ILASM]
.method public hidebysig virtual class System.IAsyncResult
BeginGetResponse(class System.AsyncCallback callback,
object state)

[C#]
public virtual IAsyncResult BeginGetResponse(AsyncCallback
callback, object state)
```

## Summary

Begins sending the current request asynchronously.

## Parameters

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.

## Return Value

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

## Description

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

## Behaviors

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

If the *callback* parameter is not **null**, the method referenced by *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**

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

# WebRequest.Create(System.String)

## Method

```
[ILASM]
.method public hidebysig static class System.Net.WebRequest
Create(string requestUriString)

[C#]
public static WebRequest Create(string requestUriString)
```

### Summary

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

### Parameters

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

### Return Value

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

### Description

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

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

### Exceptions

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

1  
2  
3  
4  
  
  
  
5  
6  
7

**Permissions**

<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.

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

# WebRequest.Create(System.Uri) Method

```
[ILASM]
.method public hidebysig static class System.Net.WebRequest
Create(class System.Uri requestUri)

[C#]
public static WebRequest Create(Uri requestUri)
```

## Summary

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

## Parameters

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

## Return Value

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

## Description

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

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

## Exceptions

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.

## 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

# WebRequest.CreateDefault(System.Uri)

## Method

```
[ILASM]
.method public hidebysig static class System.Net.WebRequest
CreateDefault(class System.Uri requestUri)

[C#]
public static WebRequest CreateDefault(Uri requestUri)
```

### Summary

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

### Parameters

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

### Return Value

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

### Description

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

### Exceptions

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.

## Example

This example demonstrates the use of the **System.Net.WebRequest.Create** and **System.Net.WebRequest.CreateDefault** methods.

```
[C#]

using System;
using System.Net;

public class ContosoTextRequest: WebRequest,
IWebRequestCreate
{
    public new WebRequest Create(Uri uri)
    {
        return new ContosoTextRequest();
    }
}

public class CreateDefaultExample
{
    public static void Main()
    {
        ContosoTextRequest contoso = new
ContosoTextRequest();
        Uri contosoUri = new
Uri("http://www.contoso.com/text");

        WebRequest.RegisterPrefix("http://www.contoso.com/text",
contoso);

        WebRequest httpContoso =
WebRequest.CreateDefault(contosoUri);
        Console.WriteLine("CreateDefault --> {0}",
httpContoso);

        WebRequest textContoso =
WebRequest.Create(contosoUri);
        Console.WriteLine("Create --> {0}", textContoso);
    }
}
```

The output is

```
CreateDefault --> System.Net.HttpWebRequest
```

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



# WebRequest.EndGetRequestStream(System.IAsyncResult) Method

```
[ILASM]
.method public hidebysig virtual class System.IO.Stream
EndGetRequestStream(class System.IAsyncResult asyncResult)

[C#]
public virtual Stream EndGetRequestStream(IAsyncResult
asyncResult)
```

## Summary

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

## 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> .

## Return Value

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

## Description

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

## Behaviors

As described above.

## Default

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

## 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 [C#]
8 public virtual WebResponse EndGetResponse(IAsyncResult
9 asyncResult)
```

## 10 Summary

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

## 13 Parameters

14  
15

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  
17  
18

## 17 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

# WebRequest.GetRequestStream() Method

```
[ILASM]
.method public hidebysig virtual class System.IO.Stream
GetRequestStream()

[C#]
public virtual Stream GetRequestStream()
```

## Summary

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

## Return Value

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

## Behaviors

As described above.

## Default

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

## How and When to Override

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

## Usage

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

The **System.Net.WebRequest.GetRequestStream** method provides synchronous access to the **System.IO.Stream**. For asynchronous access, use the **System.Net.WebRequest.BeginGetRequestStream** and **System.Net.WebRequest.EndGetRequestStream** methods.

## 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

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

Exception	Condition
<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



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

```
[ILASM]
.method public hidebysig static bool RegisterPrefix(string
prefix, class System.Net.IWebRequestCreate creator)

[C#]
public static bool RegisterPrefix(string prefix,
IWebRequestCreate creator)
```

## Summary

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

## Parameters

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.

## Return Value

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

## Description

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

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

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

## Exceptions

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

## Example

The following example demonstrates how to register a new scheme.

[C#]

```
using System;
using System.Net;

public class ftpWebRequest: WebRequest {
    //implement ftp-specific protocol methods and properties
}

public class ftpCreator: IWebRequestCreate
{
    public WebRequest Create(Uri uri)
    {
        return new ftpWebRequest();
    }
}

public class RegisterPrefixExample
{
    public static void Main()
    {
        ftpCreator creator = new ftpCreator();
        WebRequest.RegisterPrefix("ftp://", creator);
        WebRequest wr = WebRequest.Create("ftp://testFile");
        Console.WriteLine(wr);
    }
}
```

The output is

ftpWebRequest

# 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 [C#]
9 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

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

4

5

6

# WebRequest.ContentLength Property

```
[ILASM]
.property int64 ContentLength { public hideby sig virtual
specialname int64 get_ContentLength() public hideby sig
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.

# WebRequest.ContentType Property

```
[ILASM]
.property string ContentType { public hidebysig virtual
specialname string get_ContentType() public hidebysig
virtual specialname void set_ContentType(string value) }

[C#]
public virtual string ContentType { get; set; }
```

## Summary

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

## Property Value

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

## Description

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

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

## Behaviors

As described above.

## 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 media type of request.

## Exceptions

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

1  
2  
3

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



# WebRequest.Credentials Property

```
[ILASM]
.property class System.Net.ICredentials Credentials {
public hidebysig virtual specialname class
System.Net.ICredentials get_Credentials() public hidebysig
virtual specialname void set_Credentials(class
System.Net.ICredentials value) }

[C#]
public virtual ICredentials Credentials { get; set; }
```

## Summary

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

## Property Value

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

## Behaviors

As described above.

## 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 store or access the user, password, and domain information of the current instance.

## Exceptions

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

1  
2  
3

# WebRequest.Headers Property

```
[ILASM]
.property class System.Net.WebHeaderCollection Headers {
public hidebysig virtual specialname class
System.Net.WebHeaderCollection get_Headers() public
hidebysig virtual specialname void set_Headers(class
System.Net.WebHeaderCollection value) }

[C#]
public virtual WebHeaderCollection Headers { get; set; }
```

## Summary

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

## Property Value

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

## Description

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

## Behaviors

As described above.

## Default

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

## How and When to Override

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

## Usage

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

## Exceptions

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

1  
2  
3

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

# WebRequest.Method Property

```
[ILASM]
.property string Method { public hidebysig virtual
specialname string get_Method() public hidebysig virtual
specialname void set_Method(string value) }

[C#]
public virtual string Method { get; set; }
```

## Summary

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

## Property Value

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

## Behaviors

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

## Default

This property throws **System.NotSupportedException**.

## How and When to Override

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

## Usage

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

## Exceptions

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

# 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

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

4

5

6

# WebRequest.Proxy Property

```
[ILASM]
.property class System.Net.IWebProxy Proxy { public
hidebysig virtual specialname class System.Net.IWebProxy
get_Proxy() public hidebysig virtual specialname void
set_Proxy(class System.Net.IWebProxy value) }

[C#]
public virtual IWebProxy Proxy { get; set; }
```

## Summary

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

## Property Value

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

## Description

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

## Behaviors

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

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

## Default

This property throws **System.NotSupportedException**.

## How and When to Override

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

## Usage

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



1   **Exceptions**

2

3

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

4

5

6

# WebRequest.RequestUri Property

```
[ILASM]
.property class System.Uri RequestUri { public hidebysig
virtual specialname class System.Uri get_RequestUri() }

[C#]
public virtual Uri RequestUri { get; }
```

## Summary

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

## Property Value

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

## Description

This property is read-only.

## Behaviors

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

## Default

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

## How and When to Override

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

## Usage

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

1   **Exceptions**

2

3

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

4

5

6

# WebRequest.Timeout Property

```
[ILASM]
.property int32 Timeout { public hidebysig virtual
specialname int32 get_Timeout() public hidebysig virtual
specialname void set_Timeout(int32 value) }

[C#]
public virtual int Timeout { get; set; }
```

## Summary

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

## Property Value

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

## Behaviors

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

## Default

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

## How and When to Override

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

## Usage

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

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

1   **Exceptions**

2

3

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

4

5