

1 System.Net.HttpWebRequest Class

2
3

```
4 [ILASM]  
5 .class public serializable HttpWebRequest extends  
6 System.Net.WebRequest  
7  
8 [C#]  
9 public class HttpWebRequest: WebRequest
```

9 Assembly Info:

- 10 • Name: System
- 11 • Public Key: [00 00 00 00 00 00 00 00 04 00 00 00 00 00 00]
- 12 • Version: 1.0.x.x
- 13 • Attributes:
 - 14 ○ CLSCompliantAttribute(true)

15 Summary

16

17 Provides an HTTP-specific implementation of the
18 **System.Net.WebRequest** class.

19 Inherits From: System.Net.WebRequest

20

21 **Library:** Networking

22

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

25

26 Description

27 This class implements properties and methods defined in
28 **System.Net.WebRequest** and provides additional properties and
29 methods that enable the user to interact directly with servers using
30 the Hypertext Transfer Protocol (HTTP).

31

32 [*Note:* Instances of this class are automatically created by the
33 **System.Net.WebRequest** class. For example, an instance of
34 **System.Net.HttpWebRequest** is created when the
35 **System.Net.WebRequest.Create** method is called and a Uniform
36 Resource Identifier (URI) beginning with `http://` is specified. It is
37 expected that an instance of this class will be constructed for every
38 request made to the server. For example, after a call to
39 **System.Net.HttpWebRequest.Abort** cancels an asynchronous
40 operation, a call to
41 **System.Net.HttpWebRequest.GetRequestStream** causes a
42 **System.Net.WebException** to be thrown.

43

1 Requests can be sent synchronously or asynchronously. The
 2 **System.Net.HttpWebRequest.GetResponse** method sends a
 3 request to a server synchronously and returns a
 4 **System.Net.HttpWebResponse** instance containing the response.
 5 An asynchronous request for a resource is sent using the
 6 **System.Net.HttpWebRequest.BeginGetResponse** and
 7 **System.Net.HttpWebRequest.EndGetResponse** methods.

8
 9 Request data is sent using a request stream. The
 10 **System.Net.HttpWebRequest.GetRequestStream**,
 11 **System.Net.HttpWebRequest.BeginGetRequestStream**, and
 12 **System.Net.HttpWebRequest.EndGetRequestStream** methods
 13 return a **System.IO.Stream** instance used to send data.

14
 15 When errors occur while accessing an Internet resource, the
 16 **System.Net.HttpWebRequest** class throws a
 17 **System.Net.WebException**, and the
 18 **System.Net.WebException.Status** property that indicates the
 19 source of the error. When **System.Net.WebException.Status** is
 20 **System.Net.WebExceptionStatus.ProtocolError**, the
 21 **System.Net.WebException.Response** property contains the
 22 **System.Net.HttpWebResponse** received from the Internet resource.

23
 24 Certain HTTP headers are protected; the user cannot set them directly
 25 in the header collection obtained from the
 26 **System.Net.HttpWebRequest.Headers** property. Instead, these
 27 headers are set using the associated properties of a
 28 **System.Net.HttpWebRequest** instance, or are set by the system.
 29 The following table describes how each protected header is set.

Header	Set by
Accept	System.Net.HttpWebRequest.Accept
Connection	System.Net.HttpWebRequest.Connection System.Net.HttpWebRequest.KeepAlive
Content-Length	System.Net.HttpWebRequest.ContentLength
Content-Type	System.Net.HttpWebRequest.ContentType
Expect	System.Net.HttpWebRequest.Expect
Date	Set to current date by the system.
Host	Set to current host by the system.
if-Modified-since	System.Net.HttpWebRequest.IfModifiedSince
Range	System.Net.HttpWebRequest.AddRange
Referer	System.Net.HttpWebRequest.Referer
Transfer-Encoding	System.Net.HttpWebRequest.TransferEncoding System.Net.HttpWebRequest.SendChunked

User-Agent	System.Net.HttpWebRequest.UserAgent
------------	--------------------------------------------

1
2
3

]

1 `HttpRequest.Abort()` Method

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

6 **Summary**

7 Cancels an asynchronous operation.

8 **Description**

9 **System.Net.HttpHttpRequest.Abort** cancels any pending
10 asynchronous operation. After this method is called, calling
11 **System.Net.HttpHttpRequest.GetResponse**,
12 **System.Net.HttpHttpRequest.BeginGetResponse**,
13 **System.Net.HttpHttpRequest.EndGetResponse**,
14 **System.Net.HttpHttpRequest.GetRequestStream**,
15 **System.Net.HttpHttpRequest.BeginGetRequestStream**, or
16 **System.Net.HttpHttpRequest.EndGetRequestStream** will throw a
17 **System.Net.WebException** with
18 **System.Net.WebException.Status** set to
19 **System.Net.WebExceptionStatus.RequestCanceled**.

20
21 *[Note: If no pending request exists, calling this method does not cause*
22 *an exception to be thrown.*

23
24 This method overrides **System.Net.WebRequest.Abort**.]

25

1 `HttpRequest.AddRange(System.Int32` 2 `, System.Int32) Method`

```
3 [ILASM]  
4 .method public hidebysig instance void AddRange(int32 from,  
5 int32 to)  
  
6 [C#]  
7 public void AddRange(int from, int to)
```

8 Summary

9 Adds a HTTP Range header to the current instance for a specified
10 range.

11 Parameters

12
13

Parameter	Description
<i>from</i>	A System.Int32 indicating the starting byte position of the entity-body data to be returned.
<i>to</i>	A System.Int32 indicating the last byte.

14
15

Description

16 This method is equivalent to
17 **System.Net.HttpWebRequest.AddRange("bytes", *from*, *to*).**

18
19
20
21
22
23
24

[*Note:* The HTTP Range header specifies either a single range of bytes or a set of byte ranges in an entity-body to be returned. If the server accessed by the current instance supports the use of this header, this allows for the partial retrieval of the entity due to, for example, the entity being particularly large or there having been a failed transfer of data.

25
26
27

For more information on the HTTP Range header, see Section 14.35 of RFC 2616.]

28 Exceptions

29
30

Exception	Condition
System.ArgumentOutOfRangeException	<i>from</i> < 0. -or- <i>to</i> < 0.

1
2
3

	-or- <i>from > to.</i>
System.InvalidOperationException	The range header could not be added.

1 `HttpRequest.AddRange(System.Int32` 2 `)` Method

```
3 [ILASM]  
4 .method public hidebysig instance void AddRange(int32  
5 range)  
  
6 [C#]  
7 public void AddRange(int range)
```

8 Summary

9 Adds a HTTP Range header to the current instance for a specific range
10 from the beginning or end of the requested data.

11 Parameters

12
13

Parameter	Description
<i>range</i>	A System.Int32 that specifies the starting or ending point of the range. If this value is positive, the range is from the beginning of the data to <i>range</i> . If this value is negative, the range is from <i>range</i> to the end of the data.

14
15

15 Description

16 This method is equivalent to
17 **System.Net.HttpWebRequest.AddRange("bytes", range)**.

18
19 [Note: The HTTP Range header specifies either a single range of bytes
20 or a set of byte ranges in an entity-body to be returned. If the server
21 accessed by the current instance supports the use of this header, this
22 allows for the partial retrieval of the entity due to, for example, the
23 entity being particularly large or there having been a failed transfer of
24 data.

25
26 For more information on the HTTP Range header, see Section 14.35 of
27 RFC 2616.]

28 Exceptions

29
30

Exception	Condition
System.InvalidOperationException	The range header could not be added.

31
32
33

1 HttpWebRequest.AddRange(System.String, System.Int32, System.Int32) Method

```
3 [ILASM]
4 .method public hidebysig instance void AddRange(string
5 rangeSpecifier, int32 from, int32 to)
6
7 [C#]
8 public void AddRange(string rangeSpecifier, int from, int
to)
```

9 Summary

10 Adds a HTTP Range header to the current instance for a specified
11 range.

12 Parameters

13
14

Parameter	Description
<i>rangeSpecifier</i>	A System.String that contains the description of the range.
<i>from</i>	A System.Int32 designating the position at which to start sending data.
<i>to</i>	A System.Int32 designating the position at which to stop sending data.

15
16

Description

17 [Note: The HTTP Range header specifies either a single range of bytes
18 or a set of byte ranges in an entity-body to be returned. If the server
19 accessed by the current instance supports the use of this header, this
20 allows for the partial retrieval of the entity due to, for example, the
21 entity being particularly large or there having been a failed transfer of
22 data.

23
24
25

For more information on the HTTP Range header, see Section 14.35 of IETF RFC 2616.]

26 Exceptions

27
28

Exception	Condition
System.ArgumentNullException	<i>rangeSpecifier</i> is null .
System.ArgumentOutOfRangeException	<i>from</i> < 0.
	-or-

	<p><i>to</i> < 0.</p> <p>-or-</p> <p><i>from</i> > <i>to</i>.</p>
System.ArgumentException	<i>rangeSpecifier</i> is invalid.
System.InvalidOperationException	The range header could not be added.

- 1
- 2
- 3

1 HttpWebRequest.AddRange(System.String, System.Int32) Method

```
3 [ILASM]  
4 .method public hidebysig instance void AddRange(string  
5 rangeSpecifier, int32 range)  
  
6 [C#]  
7 public void AddRange(string rangeSpecifier, int range)
```

8 Summary

9 Adds a HTTP Range header to the current request for a specific range
10 from the beginning or end of the requested data.

11 Parameters

12
13

Parameter	Description
<i>rangeSpecifier</i>	A System.String that contains the description of the range.
<i>range</i>	A System.Int32 that designates the starting or ending point of the range. If this value is positive, the range is from the beginning of the data to <i>range</i> . If this value is negative, the range is from <i>range</i> to the end of the data.

14
15

Description

16 [Note: The HTTP Range header specifies either a single range of bytes
17 or a set of byte ranges in an entity-body to be returned. If the server
18 accessed by the current instance supports the use of this header, this
19 allows for the partial retrieval of the entity due to, for example, the
20 entity being particularly large or there having been a failed transfer of
21 data.

22
23
24

For more information on the HTTP Range header, see Section 14.35 of RFC 2616.]

25 Exceptions

26
27

Exception	Condition
System.ArgumentNullException	<i>rangeSpecifier</i> is null .
System.ArgumentException	<i>rangeSpecifier</i> is invalid.
System.InvalidOperationException	The range header could not be added.

28
29
30

1 **HttpWebRequest.BeginGetRequestStream** 2 **(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 [C#]  
9 public override IAsyncResult  
10 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 instance.

14 **Parameters**

15
16

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

17
18
19

18 **Return Value**

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

22 **Description**

23 This method starts an asynchronous operation. To get the request
24 stream, call the
25 **System.Net.HttpWebRequest.EndGetRequestStream** method and
26 specify the **System.IAsyncResult** object returned by this method.
27 [Note: The **System.Net.HttpWebRequest.EndGetRequestStream**
28 method should be called exactly once for each call to
29 **System.Net.HttpWebRequest.BeginGetRequestStream.**]
30
31 If the *callback* parameter is not **null**, the method(s) referenced by
32 *callback* are invoked when the asynchronous operation completes. The
33 **System.IAsyncResult** object returned by this method is passed as
34 the argument to the method(s) referenced by *callback*.

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22

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.

The value of the **System.Net.HttpWebRequest.ContentLength** property of the current instance is required to be set prior to calling this method.

[*Note:* The method(s) invoked by the callback delegate can call the **System.Net.HttpWebRequest.EndGetRequestStream** method to retrieve the stream.

This method is the asynchronous version of the **System.Net.HttpWebRequest.GetRequestStream** method.

This method overrides **System.Net.WebRequest.BeginGetRequestStream.**]

Exceptions

Exception	Condition
System.InvalidOperationException	The stream is being used by a previous call to System.Net.HttpWebRequest.BeginGetRequestStream . -or- No writeable stream is available.
System.Net.ProtocolViolationException	The System.Net.HttpWebRequest.ContentLength property of the current instance is not set. -or- The System.Net.HttpWebRequest.Method property of the current instance is "GET" or "HEAD".
System.Net.WebException	System.Net.HttpWebRequest.Abort was previously called. -or- An error occurred while processing the request.

23
24
25

1 HttpWebRequest.BeginGetResponse(System.AsyncCallback, System.Object) Method

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

10 Summary

11 Begins sending the current HTTP request asynchronously.

12 Parameters

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

16 Return Value

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

20 Description

21 This method starts an asynchronous operation. To get the response,
22 call the **System.Net.HttpWebRequest.EndGetResponse** method
23 and specify the **System.IAsyncResult** object returned by this
24 method. [*Note:* The
25 **System.Net.HttpWebRequest.EndGetResponse** method should be
26 called exactly once for each call to
27 **System.Net.HttpWebRequest.BeginGetResponse.**]

28
29 If the *callback* parameter is not **null**, the method referenced by
30 *callback* is invoked when the asynchronous operation completes. The
31 **System.IAsyncResult** object returned by this method is passed as
32 the argument to the method referenced by *callback*.

33
34 The *state* parameter can be any object that the caller wishes to have
35 available for the duration of the asynchronous operation. This object is

1 available via the **System.IAsyncResult.AsyncState** property of the
2 object returned by this method.

3
4 [Note: The method(s) invoked by the callback delegate can call the
5 **System.Net.HttpWebRequest.EndGetResponse** method to retrieve
6 the response.

7
8 This method is the asynchronous version of the
9 **System.Net.HttpWebRequest.GetResponse** method.

10
11 This method overrides
12 **System.Net.WebRequest.BeginGetResponse.**]

13 Exceptions

14
15

Exception	Condition
System.InvalidOperationException	System.Net.HttpWebRequest.BeginGetResp or System.Net.HttpWebRequest.GetRespon was previously called on this instance.
System.Net.ProtocolViolationException	The System.Net.HttpWebRequest.ContentLe property of the current instance has not been s
System.Net.WebException	System.Net.HttpWebRequest.Abort was previously called.

16
17
18

1 HttpWebRequest.EndGetRequestStream(S 2 ystem.IAsyncResult) Method

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

9 Summary

10 Completes an asynchronous request for a stream that was started by
11 the **System.Net.HttpWebRequest.BeginGetRequestStream**
12 method.

13 Parameters

14
15

Parameter	Description
<i>asyncResult</i>	The System.IAsyncResult object that holds the state information for the asynchronous operation.

16
17
18

17 Return Value

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

20 Description

21 [Note: The caller is responsible for calling the
22 **System.IO.Stream.Close** method to close the stream.

23
24
25

This method overrides
System.Net.WebRequest.EndGetRequestStream.]

26 Exceptions

27
28

Exception	Condition
System.ArgumentNullException	<i>asyncResult</i> is null .
System.ArgumentException	<i>asyncResult</i> was not returned by the current instance from a call to System.Net.WebRequest.BeginGetRequestStream .
System.InvalidOperationException	This method was called previously using <i>asvncResu</i>

	<p>-or-</p> <p>No stream is available.</p>
<p>System.Net.WebException</p>	<p>System.Net.HttpWebRequest.Abort was previously called.</p> <p>-or-</p> <p>An error occurred while processing the request.</p>

1
2
3

1 `HttpRequest.EndGetResponse(System` 2 `.IAsyncResult)` Method

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

10 Summary

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

13 Parameters

14
15

Parameter	Description
<i>asyncResult</i>	The System.IAsyncResult object that hold the state information for the asynchronous operation.

16
17

17 Return Value

18

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

21 Description

22 [Note: This method completes an asynchronous request for an
23 Internet resource that was started by calling
24 **System.Net.HttpWebRequest.BeginGetResponse**.

25

26 This method overrides
27 **System.Net.WebRequest.EndGetResponse**.]

28 Exceptions

29
30

Exception	Condition
System.ArgumentNullException	<i>asyncResult</i> is null .
System.ArgumentException	<i>asyncResult</i> was not returned by the current instance from a call to System.Net.WebRequest.BeginGetResponse .

<p>System.InvalidOperationException</p>	<p>This method was called previously using <i>asyncResult</i>.</p> <p>-or-</p> <p>The System.Net.HttpWebRequest.ContentLength property of the current instance is greater than 0 but the data has not been written to the request stream.</p>
<p>System.Net.WebException</p>	<p>System.Net.HttpWebRequest.Abort was previously called.</p> <p>-or-</p> <p>An error occurred while processing the request.</p>

- 1
- 2
- 3

1 `HttpRequest.GetHashCode()` Method

```
2 [ILASM]  
3 .method public hidebysig virtual int32 GetHashCode()  
4 [C#]  
5 public override int GetHashCode()
```

6 **Summary**

7 Generates a hash code for the current instance.

8 **Return Value**

9

10 A **System.Int32** containing the hash code for the current instance.

11 **Description**

12 The algorithm used to generate the hash code is unspecified.

13

14 [*Note:* This method overrides **System.Object.GetHashCode.**]

15

1 `HttpRequest.GetResponseStream()`

2 Method

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

8 Summary

9 Returns a **System.IO.Stream** for writing data to the Internet
10 resource requested by the current instance.

11 Return Value

12

13 A **System.IO.Stream** for writing data to an Internet resource
14 requested by the current instance.

15 Description

16 The value of the **System.Net.HttpWebRequest.ContentLength**
17 property is required to be set before writing data to the stream.

18

19 [*Note:* This method returns a stream to use to send data for the
20 **System.Net.HttpWebRequest**. Once the **System.IO.Stream**
21 instance has been returned, data can be sent with the
22 **System.Net.HttpWebRequest** by using the
23 **System.IO.Stream.Write** method.

24

25 Call the **System.IO.Stream.Close** method to close the stream and
26 release the connection for reuse. Failure to close the stream may
27 cause the application to run out of connections.

28

29 This method overrides
30 **System.Net.WebRequest.GetResponseStream.**]

31 Exceptions

32

33

Exception	Condition
System.Net.ProtocolViolationException	The System.Net.HttpWebRequest.Method property of the current instance is "GET" or "HE" -or- The System.Net.HttpWebRequest.ContentLe

	property of the current instance is not set.
System.InvalidOperationException	The System.Net.HttpWebRequest.GetResponseStream method was called more than once. -or- No writeable stream is available.
System.Net.WebException	System.Net.HttpWebRequest.Abort was previously called. -or- The timeout period for the request expired. -or- An error occurred while processing the request

1
2
3

1 `HttpRequest.GetResponse()` Method

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

7 Summary

8 Returns a response to an Internet request.

9 Return Value

10

11 A **System.Net.WebResponse** containing the response from the
12 Internet resource requested by the current instance.

13 Description

14 [Note: This method returns a **System.Net.WebResponse** instance
15 containing the response from the Internet resource requested by the
16 current instance. The actual instance returned is an instance of
17 **System.Net.HttpWebResponse**, and can be typecast to that class to
18 access HTTP-specific properties.

19

20 This method overrides **System.Net.WebRequest.GetResponse.**]

21 Exceptions

22

23

Exception	Condition
System.Net.ProtocolViolationException	The System.Net.HttpWebRequest.ContentLength property of the current instance is not set.
System.Net.WebException	System.Net.HttpWebRequest.Abort was previously called.
	-or-
	The timeout period for the request expired.
	-or-
	An error occurred while processing the request

24

25

26

1 HttpWebRequest.Accept Property

```
2 [ILASM]
3 .property string Accept { public hidebysig specialname
4 instance string get_Accept() public hidebysig specialname
5 instance void set_Accept(string value) }
6
7 [C#]
8 public string Accept { get; set; }
```

8 Summary

9 Gets or sets a **System.String** containing the value of the HTTP Accept
10 header.

11 Property Value

12

13 A **System.String** containing the value of the HTTP Accept header. The
14 default value of this property is **null**.

15 Description

16 [*Note:* For additional information see section 14.1 of IETF RFC 2068 -
17 HTTP/1.1.]

18

1 HttpWebRequest.Address Property

```
2 [ILASM]
3 .property class System.Uri Address { public hideby sig
4 specialname instance class System.Uri get_Address() }
5 [C#]
6 public Uri Address { get; }
```

7 Summary

8 Gets the URI that responds to the current request.

9 Property Value

10

11 A **System.Uri** identifying the Internet resource that responds to the
12 current request. The default is the URI used by the
13 **System.Net.WebRequest.Create** method to initialize the current
14 instance.

15 Description

16 This property is read-only.

17

18 The value of this property is set to the URI that is the source of the
19 response after all redirections are complete.

20

21 [*Note:* The URI of the original request is kept in the
22 **System.Net.HttpWebRequest.RequestUri** property.]

23

1 **HttpRequest.AllowAutoRedirect**

2 **Property**

```
3 [ILASM]
4 .property bool AllowAutoRedirect { public hidebysig
5 specialname instance bool get_AllowAutoRedirect() public
6 hidebysig specialname instance void
7 set_AllowAutoRedirect(bool value) }
8
9 [C#]
10 public bool AllowAutoRedirect { get; set; }
```

10 **Summary**

11 Gets or sets a **System.Boolean** value that indicates whether the
12 current request will follow redirection responses.

13 **Property Value**

14
15 **true** if the current request will automatically follow redirection
16 responses from the Internet resource; otherwise **false**. The default
17 value is **true**.

18 **Description**

19 [*Note:* Set **System.Net.HttpWebRequest.AllowAutoRedirect** to
20 **true** to allow the current request to automatically follow HTTP
21 redirection headers to the new location of a resource.
22

23 The maximum number of redirections to follow is set by the
24 **System.Net.HttpWebRequest.MaximumAutomaticRedirections**
25 property.]

26

1 HttpWebRequest.AllowWriteStreamBuffering 2 Property

```
3 [ILASM]  
4 .property bool AllowWriteStreamBuffering { public hidebysig  
5 specialname instance bool get_AllowWriteStreamBuffering()  
6 public hidebysig specialname instance void  
7 set_AllowWriteStreamBuffering(bool value) }  
  
8 [C#]  
9 public bool AllowWriteStreamBuffering { get; set; }
```

10 Summary

11 Gets or sets a **System.Boolean** value that indicates whether to buffer
12 the data sent to the Internet resource requested by the current
13 instance.

14 Property Value

16 **true** to enable buffering of the data sent to the Internet resource
17 requested by the current instance; **false** to disable buffering. The
18 default is **true**.

19 Description

20 [Note: When
21 **System.Net.HttpWebRequest.AllowWriteStreamBuffering** is
22 **true**, the data is buffered in memory so it is ready to be resent in the
23 event of redirections or authentication requests.

24
25 Depending on available memory, setting
26 **System.Net.HttpWebRequest.AllowWriteStreamBuffering** as
27 **true** might impact system performance when uploading large amounts
28 of data.]

29

1 HttpWebRequest.Connection Property

```
2 [ILASM]
3 .property string Connection { public hidebysig specialname
4 instance string get_Connection() public hidebysig
5 specialname instance void set_Connection(string value) }
6
7 [C#]
8 public string Connection { get; set; }
```

8 Summary

9 Gets or sets the value of the Connection HTTP header.

10 Property Value

11

12 A **System.String** containing the value of the Connection HTTP header.
13 The default value is **null**.

14 Description

15 The current request sends the
16 **System.Net.HttpWebRequest.Connection** property to the Internet
17 resource as the Connection HTTP header.

18
19 *[Note: If **System.Net.HttpWebRequest.KeepAlive** is **true**, the
20 value "Keep-alive" is appended to the end of the Connection header.*

21
22 For additional information see section 14.10 of IETF RFC 2068 -
23 HTTP/1.1.]

24 Exceptions

25

26

Exception	Condition
System.ArgumentException	The value of System.Net.HttpWebRequest.Connection is set to "Keep-alive" or "Close". This value is case insensitive.

27

28

29

1 **HttpRequest.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 override string ConnectionGroupName { get; set; }
```

10 **Summary**

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

13 **Property Value**

14
15 A **System.String** that contains the name of the connection group for
16 the current instance. The default value is **null**.

17 **Description**

18 [*Note:* The **System.Net.HttpWebRequest.ConnectionGroupName**
19 property enables a request to be associated with a connection group.
20 This is useful when an application makes requests to one server for
21 different users, such as a Web site that retrieves customer information
22 from a database server.

23
24 Each connection group creates additional connections for a server. This
25 may result in exceeding **System.Net.ServicePoint.ConnectionLimit**
26 for that server.

27
28 This property overrides
29 **System.Net.WebRequest.ConnectionGroupName.**]

30

1 HttpWebRequest.ContentLength Property

```
2 [ILASM]  
3 .property int64 ContentLength { public hidebysig virtual  
4 specialname int64 get_ContentLength() public hidebysig  
5 virtual specialname void set_ContentLength(int64 value) }  
  
6 [C#]  
7 public override long ContentLength { get; set; }
```

8 Summary

9 Gets or sets the Content-length HTTP header.

10 Property Value

11

12 A **System.Int64** value that specifies the number of bytes of data to
13 send to the Internet resource. The default is -1, which indicates that
14 this value has not been set.

15 Description

16 The **System.Net.HttpWebRequest.ContentLength** property
17 contains the value to send as the Content-length HTTP header of the
18 request.

19

20 Any value other than -1 in the
21 **System.Net.HttpWebRequest.ContentLength** property indicates
22 that the request will upload data; only methods that upload data are
23 allowed in the **System.Net.HttpWebRequest.Method** property.

24

25 This property is required to be set prior to writing data to the request
26 data stream. Once the

27 **System.Net.HttpWebRequest.ContentLength** property is set to a
28 value, that number of bytes is required to be written to the request
29 data stream. [Note: Get the request data stream by calling
30 **System.Net.HttpWebRequest.GetRequestStream**, or
31 **System.Net.HttpWebRequest.BeginGetRequestStream** and
32 **System.Net.HttpWebRequest.EndGetRequestStream**.]

33

34 [Note: For additional information see section 14.13 of IETF RFC 2068 -
35 HTTP/1.1.

36

37 This property overrides **System.Net.WebRequest.ContentLength**.]

38 Exceptions

39

40

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

1
2
3

System.InvalidOperationException	Data has already been written to the request stream.
System.ArgumentOutOfRangeException	A value less than zero is specified for a set operation.

1 **HttpRequest.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 [C#]  
7 public override string ContentType { get; set; }
```

8 **Summary**

9 Gets or sets the value of the Content-type HTTP header of the current
10 instance.

11 **Property Value**

12

13 The value of the Content-type HTTP header of the current instance.
14 The default value is **null**.

15 **Description**

16 The **System.Net.HttpWebRequest.ContentType** property contains
17 the media type of the current instance. Values assigned to the
18 **System.Net.HttpWebRequest.ContentType** property replace any
19 existing contents when the request sends the Content-type HTTP
20 header.

21
22 [*Note:* To clear the Content-type HTTP header, set the
23 **System.Net.HttpWebRequest.ContentType** property to **null**.
24

25 For additional information see section 14.17 of IETF RFC 2068 -
26 HTTP/1.1.

27
28 This property overrides **System.Net.WebRequest.ContentType.**
29

1 HttpWebRequest.ContinueDelegate 2 Property

```
3 [ILASM]  
4 .property class System.Net.HttpContinueDelegate  
5 ContinueDelegate { public hidebysig specialname instance  
6 class System.Net.HttpContinueDelegate  
7 get_ContinueDelegate() public hidebysig specialname  
8 instance void set_ContinueDelegate(class  
9 System.Net.HttpContinueDelegate value) }  
10 [C#]  
11 public HttpContinueDelegate ContinueDelegate { get; set; }
```

12 Summary

13 Gets or sets the delegate method whose methods are invoked when an
14 HTTP 100-continue response is received by the current instance.

15 Property Value

16

17 A **System.Net.HttpContinueDelegate** that references the methods
18 that are invoked when an HTTP Continue response is received. The
19 default value is **null**.

20 Description

21 [*Note:* This delegate is useful to display the status of responses
22 received by the current instance.]

23

1 HttpWebRequest.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 override ICredentials Credentials { get; set; }
```

10 Summary

11 Gets or sets the credentials used for authenticating the current
12 request.

13 Property Value

14

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

17 Description

18 [Note: The **System.Net.HttpWebRequest.Credentials** property
19 contains authentication information to identify the client making the
20 request. The **System.Net.HttpWebRequest.Credentials** property
21 can be either an instance of **System.Net.NetworkCredential**, in
22 which case the user, password, and domain information contained in
23 the **System.Net.NetworkCredential** instance is used to authenticate
24 the request, or it can be an instance of
25 **System.Net.CredentialCache**, in which case the uniform resource
26 identifier (URI) of the request is used to determine the user, password,
27 and domain information to use to authenticate the request.

28

29 This property overrides **System.Net.WebRequest.Credentials.**]

30

1 HttpWebRequest.Expect Property

```
2 [ILASM  
3 .property string Expect { public hidebysig specialname  
4 instance string get_Expect() public hidebysig specialname  
5 instance void set_Expect(string value) }  
6  
7 [C#  
8 public string Expect { get; set; }
```

8 Summary

9 Gets or sets the value of the HTTP Expect header.

10 Property Value

11

12 A **System.String** that contains the contents of the HTTP Expect
13 header. The default value is **null**.

14 Description

15 [Note: By default, **System.Net.HttpWebRequest.Expect** is **null**.
16 Other values can be added to the list that
17 **System.Net.HttpWebRequest.Expect** maintains, or all values
18 except "100-continue" can be deleted from the list by setting
19 **System.Net.HttpWebRequest.Expect** to **null**.
20

21 For additional information see section 14.20 of IETF RFC 2068 -
22 HTTP/1.1.]

23 Exceptions

24

25

Exception	Condition
System.ArgumentException	The value specified for a set operation is "100-continue". This value is case insensitive.

26

27

28

1 HttpWebRequest.HaveResponse Property

```
2 [ILASM]  
3 .property bool HaveResponse { public hidebysig specialname  
4 instance bool get_HaveResponse() }  
5 [C#]  
6 public bool HaveResponse { get; }
```

7 Summary

8 Gets a **System.Boolean** value indicating whether a response has
9 been received for the current instance.

10 Property Value

11

12 **true** if a response has been received; otherwise **false**.

13 Description

14 This property is read-only.

15

1 HttpWebRequest.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 override WebHeaderCollection Headers { get; set; }
```

10 Summary

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

13 Property Value

14

15 A **System.Net.WebHeaderCollection** containing the name/value
16 pairs of the headers for the current instance.

17 Description

18 The following table lists the HTTP headers that cannot be set using the
19 collection returned by this property.

Header	Set by
Accept	System.Net.HttpWebRequest.Accept.
Connection	System.Net.HttpWebRequest.Connection.
	System.Net.HttpWebRequest.KeepAlive.
Content-Length	System.Net.HttpWebRequest.ContentLength.
Content-Type	System.Net.HttpWebRequest.ContentType.
Expect	System.Net.HttpWebRequest.Expect.
Date	Set by the system to the current date.
Host	Set by the system to the current host information.
Range	System.Net.HttpWebRequest.AddRange.
Referer	System.Net.HttpWebRequest.Referer.
Transfer-Encoding	System.Net.HttpWebRequest.TransferEncoding.
	System.Net.HttpWebRequest.SendChunked.
User-Agent	System.Net.HttpWebRequest.UserAgent.

1
2

[*Note:* This property overrides **System.Net.WebRequest.Headers.**]

3
4
5

Exceptions

Exception	Condition
System.InvalidOperationException	A set operation was requested but data has already been written to the request data stream.

6
7
8

1 **HttpRequest.IfModifiedSince Property**

```
2 [ILASM]
3 .property valuetype System.DateTime IfModifiedSince {
4 public hidebysig specialname instance valuetype
5 System.DateTime get_IfModifiedSince() public hidebysig
6 specialname instance void set_IfModifiedSince(valuetype
7 System.DateTime value) }
8
9 [C#]
public DateTime IfModifiedSince { get; set; }
```

10 **Summary**

11 Gets or sets the value of the HTTP If-Modified-Since header.

12 **Property Value**

13

14 A **System.DateTime** that contains the contents of the HTTP If-
15 Modified-Since header. The default value is the current date and time
16 of the system.

17 **Description**

18 [Note: For additional information see section 14.25 of IETF RFC 2068 -
19 HTTP/1.1.]

20

1 HttpWebRequest.KeepAlive Property

```
2 [ILASM]
3 .property bool KeepAlive { public hidebysig specialname
4 instance bool get_KeepAlive() public hidebysig specialname
5 instance void set_KeepAlive(bool value) }
6
7 [C#]
8 public bool KeepAlive { get; set; }
```

8 Summary

9 Gets or sets a **System.Boolean** value indicating whether to make a
10 persistent connection to the server hosting the Internet resource
11 requested by the current instance.

12 Property Value

13

14 **true** indicates that the current request will contain an HTTP
15 Connection header with the value "Keep-alive"; otherwise, **false**. The
16 default value is **true**.

17 Description

18 [*Note:* An application uses **System.Net.HttpWebRequest.KeepAlive**
19 to indicate a preference for persistent connections. When this property
20 is **true**, the application makes persistent connections to the servers
21 that support them.]

22

1 HttpWebRequest.MaximumAutomaticRedi 2 rections Property

```
3 [ILASM  
4 .property int32 MaximumAutomaticRedirections { public  
5 hidebysig specialname instance int32  
6 get_MaximumAutomaticRedirections() public hidebysig  
7 specialname instance void  
8 set_MaximumAutomaticRedirections(int32 value) }  
9  
10 [C#]  
public int MaximumAutomaticRedirections { get; set; }
```

11 Summary

12 Gets or sets the maximum number of redirects that the current
13 instance will follow.

14 Property Value

15
16 A **System.Int32** value that indicates the maximum number of
17 redirection responses that the current instance will follow. The default
18 value is implementation-defined.

19 Description

20 [Note: This property sets the maximum number of redirections for the
21 request to follow if the
22 **System.Net.HttpWebRequest.AllowAutoRedirect** property is
23 **true**.]

24 Exceptions

25

Exception	Condition
System.ArgumentException	The value specified for a set operation is less than or equal to zero.

27

28

29

1 **HttpRequest.MediaType Property**

```
2 [ILASM]
3 .property string MediaType { public hidebysig specialname
4 instance string get_MediaType() public hidebysig
5 specialname instance void set_MediaType(string value) }
6
7 [C#]
8 public string MediaType { get; set; }
```

8 **Summary**

9 Gets or sets the media type of the current request.

10 **Property Value**

11

12 A **System.String** that identifies the media type of the current request.
13 The default value is **null**.

14 **Description**

15 [*Note:* The value of this property affects the
16 **System.Net.HttpWebResponse.CharacterSet** property. When this
17 property is set in the current instance, the corresponding media type is
18 chosen from the list of character sets returned in the response HTTP
19 Content-type header.]

20

1 HttpWebRequest.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 override string Method { get; set; }
```

8 Summary

9 Gets or sets the HTTP protocol request method used by the current
10 instance.

11 Property Value

13 A **System.String** containing an HTTP method. The default value is
14 "GET".

15 Description

16 If the **System.Net.HttpWebRequest.ContentLength** property is set
17 to any value other than -1, the
18 **System.Net.HttpWebRequest.Method** property is required to be set
19 to a protocol method that sends request data.

21 The **System.Net.HttpWebRequest.Method** property can be set to
22 any of the following HTTP 1.1 protocol methods:

HTTP Method	Description
GET	Retrieves in entity-body form the information identified by the System.Net.HttpWebRequest.RequestUri property of the current instance.
HEAD	Identical to GET except that the message-body is not returned in the response.
POST	Requests that the origin server accept the entity enclosed in the request as a new subordinate of the resource identified the Request-URI in the Request-Line.
PUT	Requests that the enclosed entity be stored under the supplied Request-URI.
DELETE	Requests that the origin server delete the resource identified by the Request-URI.
TRACE	Invokes a remote, application-layer loopback of the request message.
OPTIONS	Requests information about the communication options available on the request/response chain identified by the Request-URI. [Note: This allows

the client to determine the options and/or requirements associated with a resource, or the capabilities of a server, without implying a resource action or initiating a resource retrieval.]

1
2
3
4
5
6
7
8
9
10
11

[*Note:* For detailed information regarding these methods, see sections 9.2 to 9.8 of RFC 2616.

This property overrides **System.Net.WebRequest.Method**.]

Exceptions

Exception	Condition
System.ArgumentException	null, System.String.Empty , or an invalid value was specified for a set operation.

1 HttpWebRequest.Pipelined Property

```
2 [ILASM]
3 .property bool Pipelined { public hidebysig specialname
4 instance bool get_Pipelined() public hidebysig specialname
5 instance void set_Pipelined(bool value) }
6
7 [C#]
8 public bool Pipelined { get; set; }
```

8 Summary

9 Gets or sets a **System.Boolean** value indicating whether to pipeline
10 the current request to the Internet resource.

11 Property Value

12

13 **true** if the current request can be pipelined; otherwise, **false**. The
14 default is **true**.

15 Description

16 An application uses this property to indicate a preference for pipelined
17 connections. If **System.Net.HttpWebRequest.Pipelined** is **true**, an
18 application makes pipelined connections to servers that support them.

19

20 [*Note:* Pipelined connections are made only when the
21 **System.Net.HttpWebRequest.KeepAlive** property is **true**.]

22

1 HttpWebRequest.PreAuthenticate 2 Property

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

9 Summary

10 Gets or sets a Boolean value that indicates whether to send HTTP
11 preauthentication header information with current instance without
12 waiting for an authentication challenge from the requested resource.

13 Property Value

14

15 **true** to send a HTTP WWW-authenticate header with the current
16 instance without waiting for an authentication challenge from the
17 requested resource; otherwise, **false**. The default is **false**.

18 Description

19 When **System.Net.HttpWebRequest.PreAuthenticate** is **true** and
20 credentials are supplied, the HTTP WWW-authenticate header is sent
21 with the current instance without waiting for an authentication
22 challenge from the requested resource; otherwise the request uses
23 standard authentication procedures.

24

25 [*Note:* Set this property to **true** to allow clients to improve server
26 efficiency by avoiding extra round trips caused by authentication
27 challenges.

28

29 This property overrides
30 **System.Net.WebRequest.PreAuthenticate.**]

31

1 HttpWebRequest.ProtocolVersion 2 Property

```
3 [ILASM]  
4 .property class System.Version ProtocolVersion { public  
5 hidebysig specialname instance class System.Version  
6 get_ProtocolVersion() public hidebysig specialname instance  
7 void set_ProtocolVersion(class System.Version value) }  
  
8 [C#]  
9 public Version ProtocolVersion { get; set; }
```

10 Summary

11 Gets or sets the version of the HTTP protocol to use for the current
12 request.

13 Property Value

15 A **System.Version** that represents the HTTP version to use for the
16 request. The default is **System.Net.HttpVersion.Version11**.

17 Description

18 The **System.Net.HttpWebRequest** class supports only versions 1.0
19 and 1.1 of HTTP. Setting
20 **System.Net.HttpWebRequest.ProtocolVersion** to a different
21 version causes a **System.ArgumentException** exception to be
22 thrown.

24 [*Note:* To set the **System.Net.HttpWebRequest.ProtocolVersion**
25 property of the current instance, specify one of the members of the
26 use the **System.Net.HttpVersion** class.]

27 Exceptions

28
29

Exception	Condition
System.ArgumentException	The HTTP version is set to a value other than 1.0 or 1.1.

30
31
32

1 HttpWebRequest.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 override IWebProxy Proxy { get; set; }
```

9 Summary

10 Gets or sets network proxy information for the current instance.

11 Property Value

12
13 The **System.Net.WebProxy** instance to use as a proxy for the
14 current instance. The default value is set by calling
15 **System.Net.GlobalProxySelection.Select**.

16 Description

17 The **System.Net.HttpWebRequest.Proxy** property identifies the
18 **System.Net.WebProxy** instance to use to communicate with the
19 destination server.

20
21 *[Note: To specify that no proxy should be used, set the*
22 **System.Net.HttpWebRequest.Proxy** property to the proxy instance
23 returned by the
24 **System.Net.GlobalProxySelection.GetEmptyWebProxy** method.
25

26 This property overrides **System.Net.WebRequest.Proxy**.]

27 Exceptions

Exception	Condition
System.ArgumentNullException	A set operation was requested and the specified value was null .
System.InvalidOperationException	A set operation was requested but data has already been sent to the request stream.
System.Security.SecurityException	The caller does not have permission for the requested operation.

31 Permissions

1
2

Permission	Description
System.Security.Permissions.WebPermission	Requires unrestricted System.Net.WebPermission . See System.Security.Permissions.PermissionState.Unrestricted .

3
4
5

1 HttpWebRequest.Referer Property

```
2 [ILASM]
3 .property string Referer { public hidebysig specialname
4 instance string get_Referer() public hidebysig specialname
5 instance void set_Referer(string value) }
6
7 [C#]
8 public string Referer { get; set; }
```

8 Summary

9 Gets or sets the value of the HTTP Referer header.

10 Property Value

11

12 A **System.String** containing the value of the HTTP Referer header.
13 The default value is **null**.

14 Description

15 [*Note:* For additional information see section 14.36 of IETF RFC 2068 -
16 HTTP/1.1.]

17

1 HttpWebRequest.RequestUri Property

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

7 Summary

8 Gets the **System.Uri** of the resource that receives requests sent by
9 the current instance.

10 Property Value

11

12 The **System.Uri** of the resource that receives requests sent by the
13 current instance.

14 Description

15 This property is read-only.

16

17 This property is the **System.Uri** instance passed to the current
18 instance via the **System.Net.WebRequest.Create** method.

19

20 *[Note: Following a redirection header does not change the*
21 **System.Net.HttpWebRequest.RequestUri** property. The URI of the
22 resource that actually responded to the current instance is contained
23 by **System.Net.HttpWebRequest.Address** property of the current
24 instance.

25

26 This property overrides **System.Net.WebRequest.RequestUri.**]

27

1 HttpWebRequest.SendChunked Property

```
2 [ILASM]
3 .property bool SendChunked { public hidebysig specialname
4 instance bool get_SendChunked() public hidebysig
5 specialname instance void set_SendChunked(bool value) }
6
7 [C#]
8 public bool SendChunked { get; set; }
```

8 Summary

9 Gets or sets a value indicating whether to send data in segments.

10 Property Value

11

12 **true** to send data in segments; otherwise, **false**. The default value is
13 **false**.

14 Description

15 When **System.Net.HttpWebRequest.SendChunked** is **true**, the
16 request sends data to the destination in segments. The destination
17 server is required to support receiving chunked data.

18

19 [Note: Set this property to **true** only if the server specified by the
20 **System.Net.HttpWebRequest.Address** property of the current
21 instance accepts chunked data (i.e. is HTTP/1.1 or greater in
22 compliance). If the server does not accept chunked data, buffer all
23 data to be written and send a HTTP Content-Length header with the
24 buffered data.]

25 Exceptions

26

27

Exception	Condition
System.InvalidOperationException	A set operation was requested but data has already been written to the request data stream.

28

29

30

1 **HttpRequest.ServicePoint** Property

```
2 [ILASM]
3 .property class System.Net.ServicePoint ServicePoint {
4 public hidebysig specialname instance class
5 System.Net.ServicePoint get_ServicePoint() }
6
7 [C#]
8 public ServicePoint ServicePoint { get; }
```

8 **Summary**

9 Gets the service point to use for the current instance.

10 **Property Value**

11

12 A **System.Net.ServicePoint** that represents the network connection
13 to the destination. The value of this property is **null** until the
14 **System.Net.HttpWebRequest.GetResponse** method is called.

15 **Description**

16 This property is read-only.

17

1 HttpWebRequest.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 override int Timeout { get; set; }
```

8 Summary

9 Gets or sets the length of time before the request times out.

10 Property Value

11

12 A **System.Int32** indicating the number of milliseconds to wait for a
13 response until the request times out, or
14 **System.Threading.Timeout.Infinite** to indicate that the request
15 does not time out.

16 Description

17 **System.Net.HttpWebRequest.Timeout** is the number of
18 milliseconds that a synchronous request made with the
19 **System.Net.HttpWebRequest.GetResponse** method waits for a
20 response. If a resource does not respond within the time-out period,
21 the request throws a **System.Net.WebException** with the
22 **System.Net.WebException.Status** property set to
23 **System.Net.WebExceptionStatus.Timeout**.

24

25 [*Note:* This property overrides **System.Net.WebRequest.Timeout**.]

26 Exceptions

27

28

Exception	Condition
System.ArgumentOutOfRangeException	A value less than zero and not equal to System.Threading.Timeout.Infinite is specified for a set operation.

29

30

31

1 HttpWebRequest.TransferEncoding

2 Property

```
3 [ILASM]
4 .property string TransferEncoding { public hidebysig
5 specialname instance string get_TransferEncoding() public
6 hidebysig specialname instance void
7 set_TransferEncoding(string value) }
8
9 [C#]
10 public string TransferEncoding { get; set; }
```

10 Summary

11 Gets or sets the value of the HTTP Transfer-encoding header.

12 Property Value

13

14 A **System.String** that contains the value of the HTTP Transfer-
15 encoding header. The default value is **null**.

16 Description

17 This property can be set in the current instance only if the
18 **System.Net.HttpWebRequest.SendChunked** property in the
19 current instance is **true**.

20

21 [Note: Clearing **System.Net.HttpWebRequest.TransferEncoding**
22 by setting it to **null** has no effect on the value of
23 **System.Net.HttpWebRequest.SendChunked**.

24

25 Values assigned to the
26 **System.Net.HttpWebRequest.TransferEncoding** property replace
27 any existing contents.

28

29 For additional information see section 14.41 of IETF RFC 2068 -
30 HTTP/1.1.]

31 Exceptions

32

33

Exception	Condition
System.InvalidOperationException	System.Net.HttpWebRequest.TransferEncoding is set when System.Net.HttpWebRequest.SendChunked is false .
System.ArgumentException	System.Net.HttpWebRequest.TransferEncoding

1
2
3

	is set to the value "Chunked". This value is case insensitive.
--	----------------------------------------------------------------

1 HttpWebRequest.UserAgent Property

```
2 [ILASM]
3 .property string UserAgent { public hidebysig specialname
4 instance string get_UserAgent() public hidebysig
5 specialname instance void set_UserAgent(string value) }
6
7 [C#]
8 public string UserAgent { get; set; }
```

8 Summary

9 Gets or sets the value of the HTTP User-agent header.

10 Property Value

11

12 A **System.String** containing the value of the HTTP User-agent header.
13 The default value is **null**.

14 Description

15 [*Note:* For additional information see section 14.43 of IETF RFC 2068 -
16 HTTP/1.1.]

17