

1 System.Net.IPAddress Class

```
2 [ILAsm]  
3 .class public serializable IPAddress extends System.Object  
  
4 [C#]  
5 public class IPAddress
```

6 Assembly Info:

- 7 • Name: System
- 8 • Public Key: [00 00 00 00 00 00 00 00 04 00 00 00 00 00 00 00]
- 9 • Version: 2.0.x.x
- 10 • Attributes:
 - 11 ○ CLSCompliantAttribute(true)

12 Summary

13 Represents an Internet Protocol (IP) address.

14 Inherits From: System.Object

15

16 **Library:** Networking

17

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

20

21 Description

22 An instance of the `System.Net.IPAddress` class contains the value of an address on an
23 IP network. This address is stored internally as a `System.Int64` in network-byte-order.

24
25 [*Note:* Different conventions are in use for ordering bytes within multi-byte data types.
26 All IP address values must be sent over the network in network-byte-order. Network-
27 byte-order puts the most significant byte first (also known as big-endian order). On the
28 host, the ordering of bytes is platform-specific and this ordering is referred to as host-
29 byte-order.]

30

31

32
33 The IP address can be represented as four numbers in the range 0-255 separated by
34 periods (for example, "192.168.1.2"), known as dotted-quad notation.

35

36 [*Note:* The address space is fragmented into different types determined by bits 31-28 as
37 shown in the following table.

Bits 31-28	Address type	Address range
------------	--------------	---------------

0xxx	class A	0.0.0.0-127.255.255.255
10xx	class B	128.0.0.0-191.255.255.255
110x	class C	192.0.0.0-223.255.255.255
1110	multicast	224.0.0.0-239.255.255.255
1111	reserved	240.0.0.0-255.255.255.255

1
2
3
4
5

]

Instances of the `System.Net.IPAddress` class are provided for common IP address values as shown in the following table.

Field	IP Address
Any	0.0.0.0
Broadcast	255.255.255.255
Loopback	127.0.0.1
None	255.255.255.255

6
7
8

1 IPAddress(System.Int64) Constructor

```
2 [ILAsm]  
3 public rtspecialname specialname instance void .ctor(int64 newAddress)  
4 [C#]  
5 public IPAddress(long newAddress)
```

6 Summary

7 Constructs and initializes a new instance of the `System.Net.IPAddress` class.

8 Parameters

Parameter	Description
<i>newAddress</i>	A <code>System.Int64</code> containing the IP address in host-byte-order.

9

10 Exceptions

Exception	Condition
System.ArgumentOutOfRangeException	<i>newAddress</i> is less than 0 or greater than 0x00000000FFFFFFFF.

11

12

1 IPAddress.Any Field

```
2 [ILAsm]  
3 .field public static initOnly class System.Net.IPAddress Any  
4 [C#]  
5 public static readonly IPAddress Any
```

6 Summary

7 Indicates that the protocol will select which address to use.

8 Description

9 This field is read-only.

10

11 This is equivalent to `System.Net.IPAddress.IPAddress (0x0000000000000000)` and
12 represents the address 0.0.0.0.

13

1 IPAddress.Broadcast Field

```
2 [ILAsm]  
3 .field public static initOnly class System.Net.IPAddress Broadcast  
4 [C#]  
5 public static readonly IPAddress Broadcast
```

6 Summary

7 Provides the IP broadcast address.

8 Description

9 This field is read-only.

10
11 This is equivalent to `System.Net . IPAddress . IPAddress (0x00000000FFFFFFFF)` and
12 represents the address 255.255.255.255.

13
14 This value is used to simultaneously address every host on the network.

15
16 [*Note:* Multiple fields are defined for this IP address based on prior art. This field is
17 identical to `System.Net . IPAddress . None`.

18
19]

20

1 IPAddress.Loopback Field

```
2 [ILAsm]  
3 .field public static initOnly class System.Net.IPAddress Loopback  
4 [C#]  
5 public static readonly IPAddress Loopback
```

6 Summary

7 Provides the IP loopback address.

8 Description

9 This field is read-only.

10
11 This is equivalent to `System.Net.IPAddress.IPAddress (0x000000000100007F)` and
12 represents the address 127.0.0.1.

13
14 The loopback address is used to specify the address of the local computer.

15

1 IPAddress.None Field

```
2 [ILAsm]  
3 .field public static initOnly class System.Net.IPAddress None  
4 [C#]  
5 public static readonly IPAddress None
```

6 Summary

7 Provides the IP address that indicates that no network interface should be used.

8 Description

9 This field is read-only.

10 This is equivalent to `System.Net . IPAddress . IPAddress (0x00000000FFFFFFFF)` and
11 represents the address 255.255.255.255.
12

13
14 [*Note:* Multiple fields are defined for this IP address based on prior art. This field is
15 identical to `System.Net . IPAddress . Broadcast`.
16

17]
18

1 IPAddress.Equals(System.Object) Method

```
2 [ILAsm]  
3 .method public hidebysig virtual bool Equals(object comparand)  
4 [C#]  
5 public override bool Equals(object comparand)
```

6 Summary

7 Determines whether the current instance and the specified `System.Object` represent the
8 same IP address.

9 Parameters

Parameter	Description
<i>comparand</i>	A <code>System.Object</code> to compare to the current instance.

10

11 Return Value

12 A `System.Boolean` where `true` indicates *comparand* is an instance of the
13 `System.Net.IPAddress` class and has the same `System.Net.IPAddress.Address`
14 property value as the current instance; otherwise `false`.

15 Description

16 [Note: This method overrides `System.Object.Equals`.
17]
18]

19

1 IPAddress.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 A `System.Int32` containing the hash code for the current instance.

10 Description

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

12

13 [*Note:* This method overrides `System.Object.GetHashCode`.

14

15]

16

1
2 **IPAddress.HostToNetworkOrder(System.Int64) Method**
3

```
4 [ILAsm]  
5 .method public hidebysig static int64 HostToNetworkOrder(int64 host)  
6 [C#]  
7 public static long HostToNetworkOrder(long host)
```

8 **Summary**

9 Converts a `System.Int64` from host-byte-order to network-byte-order.

10 **Parameters**

Parameter	Description
<i>host</i>	A <code>System.Int64</code> in host-byte-order.

11
12 **Return Value**

13 A `System.Int64` in network-byte-order.

14 **Description**

15 This method performs conversions on systems where the host-byte-order differs from
16 network-byte-order. On systems where this is not the case, this method does nothing.

17

1
2 **IPAddress.HostToNetworkOrder(System.Int32) Method**
3

```
4 [ILAsm]  
5 .method public hidebysig static int32 HostToNetworkOrder(int32 host)  
6 [C#]  
7 public static int HostToNetworkOrder(int host)
```

8 **Summary**

9 Converts a `System.Int32` from host-byte-order to network-byte-order.

10 **Parameters**

Parameter	Description
<i>host</i>	A <code>System.Int32</code> in host-byte-order.

11
12 **Return Value**

13 A `System.Int32` in network-byte-order.

14 **Description**

15 This method performs conversions on systems where the host-byte-order differs from
16 network-byte-order. On systems where this is not the case, this method does nothing.

17

1
2 **IPAddress.HostToNetworkOrder(System.Int16)**
3 **Method**

```
4 [ILAsm]  
5 .method public hidebysig static int16 HostToNetworkOrder(int16 host)  
6 [C#]  
7 public static short HostToNetworkOrder(short host)
```

8 **Summary**

9 Converts a `System.Int16` from host-byte-order to network-byte-order.

10 **Parameters**

Parameter	Description
<i>host</i>	A <code>System.Int16</code> in host-byte-order.

11
12 **Return Value**

13 A `System.Int16` in network-byte-order.

14 **Description**

15 This method performs conversions on systems where the host-byte-order differs from
16 network-byte-order. On systems where this is not the case, this method does nothing.

17

1
2 **IPAddress.IsLoopback(System.Net.IPAddress**
3 **) Method**

```
4 [ILAsm]  
5 .method public hidebysig static bool IsLoopback(class System.Net.IPAddress  
6 address)  
7 [C#]  
8 public static bool IsLoopback(IPAddress address)
```

9 **Summary**

10 Returns a `System.Boolean` that indicates whether the specified IP address is a loopback
11 address.

12 **Parameters**

Parameter	Description
<i>address</i>	A <code>System.Net.IPAddress</code> containing the IP address to check.

13
14 **Return Value**

15 `true` if *address* is a loopback address; otherwise `false`.

16 **Description**

17 All IP addresses of the form 127.X.Y.Z, where X, Y, and Z are in the range 0-255, are
18 forwarded to the IP loopback address 127.0.0.1. The `System.Net.IPAddress.Loopback`
19 address is used to specify the address of the local computer.

20

1
2 **IPAddress.NetworkToHostOrder(System.Int64)**
3 **Method**

```
4 [ILAsm]  
5 .method public hidebysig static int64 NetworkToHostOrder(int64 network)  
6 [C#]  
7 public static long NetworkToHostOrder(long network)
```

8 **Summary**

9 Converts a `System.Int64` from network-byte-order to host-byte-order.

10 **Parameters**

Parameter	Description
<i>network</i>	A <code>System.Int64</code> in network-byte-order.

11
12 **Return Value**

13 A `System.Int64` in host-byte-order.

14 **Description**

15 This method performs conversions on systems where the host-byte-order differs from
16 network-byte-order. On systems where this is not the case, this method does nothing.

17

1
2 **IPAddress.NetworkToHostOrder(System.Int32) Method**
3

```
4 [ILAsm]  
5 .method public hidebysig static int32 NetworkToHostOrder(int32 network)  
6 [C#]  
7 public static int NetworkToHostOrder(int network)
```

8 **Summary**

9 Converts a `System.Int32` from network-byte-order to host-byte-order.

10 **Parameters**

Parameter	Description
<i>network</i>	A <code>System.Int32</code> in network-byte-order.

11
12 **Return Value**

13 A `System.Int32` in host-byte-order.

14 **Description**

15 This method performs conversions on systems where the host-byte-order differs from
16 network-byte-order. On systems where this is not the case, this method does nothing.

17

1
2 **IPAddress.NetworkToHostOrder(System.Int16)**
3 **Method**

```
4 [ILAsm]  
5 .method public hidebysig static int16 NetworkToHostOrder(int16 network)  
6 [C#]  
7 public static short NetworkToHostOrder(short network)
```

8 **Summary**

9 Converts a `System.Int16` from network-byte-order to host-byte-order.

10 **Parameters**

Parameter	Description
<i>network</i>	A <code>System.Int16</code> in network-byte-order.

11
12 **Return Value**

13 A `System.Int16` in host-byte-order.

14 **Description**

15 This method performs conversions on systems where the host-byte-order differs from
16 network-byte-order. On systems where this is not the case, this method does nothing.

17

1 IPAddress.Parse(System.String) Method

```
2 [ILAsm]  
3 .method public hidebysig static class System.Net.IPAddress Parse(string  
4 ipString)  
  
5 [C#]  
6 public static IPAddress Parse(string ipString)
```

7 Summary

8 Converts a `System.String` representation of an IP address in dotted-quad notation, to a
9 `System.Net.IPAddress` instance.

10 Parameters

Parameter	Description
<i>ipString</i>	A <code>System.String</code> in dotted-quad notation containing the IP address to convert.

11 Return Value

13 A new `System.Net.IPAddress` instance that represents the address specified in *ipString*.

14 Description

15 [Note: An example of a string in dotted-quad notation is "127.0.0.1".
16]
17]

18 Exceptions

Exception	Condition
<code>System.ArgumentNullException</code>	<i>ipString</i> is null.
<code>System.FormatException</code>	<i>ipString</i> is not a valid IP address.

19
20

1 IPAddress.ToString() Method

```
2 [ILAsm]  
3 .method public hidebysig virtual string ToString()  
4 [C#]  
5 public override string ToString()
```

6 Summary

7 Returns a `System.String` representation of the value of the current instance.

8 Return Value

9 A `System.String` representation of the current instance. The returned string is an IP
10 address expressed in dotted-quad notation (for example, "192.168.1.2").

11 Description

12 [*Note:* The `System.Net.IPAddress.ToString` method converts the IP address stored in
13 the `System.Net.IPAddress.Address` property of the current instance to a
14 `System.String` containing the address in dotted-quad notation (for example,
15 "192.168.1.2").

16 This method overrides `System.Object.ToString`.

17]
18]
19]

20

1 IPAddress.Address Property

```
2 [ILAsm]  
3 .property int64 Address { public hidebysig specialname instance int64  
4 get_Address() public hidebysig specialname instance void set_Address(int64  
5 value) }  
  
6 [C#]  
7 public long Address { get; set; }
```

8 Summary

9 Gets or sets an Internet Protocol (IP) address.

10 Property Value

11 A System.Int64 containing the IP address in host-byte-order.

12 Description

13 [Note: To convert System.Net.IPAddress.Address to dotted-quad notation, use the
14 System.Net.IPAddress.ToString method.

15
16 Values greater than 0x00000000FFFFFFFF are permitted for IPv6 extensibility.

17
18]

19 Exceptions

Exception	Condition
System.ArgumentOutOfRangeException	The value specified in a set operation is less than 0.

20

21

1 IPAddress.AddressFamily Property

```
2 [ILAsm]  
3 .property valuetype System.Net.Sockets.AddressFamily AddressFamily {  
4 public hidebysig specialname instance valuetype  
5 System.Net.Sockets.AddressFamily get_AddressFamily() }  
6 [C#]  
7 public AddressFamily AddressFamily { get; }
```

8 Summary

9 Gets the address family.

10 Property Value

11 System.Net.Sockets.AddressFamily.InterNetwork.

12 Description

13 This property is read-only.

14