

# System.Net.IPAddress Class

```
[ILAsm]
.class public serializable IPAddress extends System.Object

[C#]
public class IPAddress
```

## Assembly Info:

- *Name:* System
- *Public Key:* [00 00 00 00 00 00 00 00 04 00 00 00 00 00 00 00]
- *Version:* 2.0.x.x
- *Attributes:*
  - CLSCompliantAttribute(true)

## Summary

Represents an Internet Protocol (IP) address.

## Inherits From: System.Object

**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

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

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

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

[*Note:* The address space is fragmented into different types determined by bits 31-28 as 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 |

]

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 |

# IPAddress(System.Int64) Constructor

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

## Summary

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

## Parameters

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

## Exceptions

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

# IPAddress.Any Field

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

## Summary

Indicates that the protocol will select which address to use.

## Description

This field is read-only.

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

# IPAddress.Broadcast Field

```
[ILAsm]  
.field public static initOnly class System.Net.IpAddress Broadcast  
  
[C#]  
public static readonly IPAddress Broadcast
```

## Summary

Provides the IP broadcast address.

## Description

This field is read-only.

This is equivalent to `System.Net.IpAddress.IpAddress (0x00000000FFFFFFFF)` and represents the address 255.255.255.255.

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

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

]

# IPAddress.Loopback Field

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

## Summary

Provides the IP loopback address.

## Description

This field is read-only.

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

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

# IPAddress.None Field

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

## Summary

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

## Description

This field is read-only.

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

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

]

# 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. |

## 13 Return Value

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

## 18 Description

19 [Note: This method overrides `System.Object.Equals`.  
20  
21 ]



# 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  
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  
16 ]

# IPAddress.HostToNetworkOrder(System.Int64) Method

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

## Summary

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

## Parameters

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

## Return Value

A System.Int64 in network-byte-order.

## Description

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

# IPAddress.HostToNetworkOrder(System.Int32) Method

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

## Summary

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

## Parameters

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

## Return Value

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

## Description

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

# IPAddress.HostToNetworkOrder(System.Int16) Method

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

## Summary

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

## Parameters

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

## Return Value

A System.Int16 in network-byte-order.

## Description

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

## IPAddress.IsLoopback(System.Net.IPAddress) Method

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

### Summary

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

### Parameters

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

### Return Value

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

### Description

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

# IPAddress.NetworkToHostOrder(System.Int64) Method

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

## Summary

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

## Parameters

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

## Return Value

A System.Int64 in host-byte-order.

## Description

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

# IPAddress.NetworkToHostOrder(System.Int32) Method

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

## Summary

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

## Parameters

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

## Return Value

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

## Description

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

# IPAddress.NetworkToHostOrder(System.Int16) Method

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

## Summary

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

## Parameters

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

## Return Value

A System.Int16 in host-byte-order.

## Description

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



# IPAddress.Parse(System.String) Method

```
[ILAsm]  
.method public hidebysig static class System.Net.IPAddress Parse(string  
ipString)
```

```
[C#]  
public static IPAddress Parse(string ipString)
```

## Summary

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

## Parameters

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

## Return Value

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

## Description

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

]

## 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. |

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

## 12 Description

13 [*Note:* The `System.Net.IPAddress.ToString` method converts the IP address stored in  
14 the `System.Net.IPAddress.Address` property of the current instance to a  
15 `System.String` containing the address in dotted-quad notation (for example,  
16 "192.168.1.2").

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

18  
19  
20 ]

# IPAddress.Address Property

```
[ILAsm]
.property int64 Address { public hidebysig specialname instance int64
get_Address() public hidebysig specialname instance void set_Address(int64
value) }

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

## Summary

Gets or sets an Internet Protocol (IP) address.

## Property Value

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

## Description

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

Values greater than 0x00000000FFFFFFFF are permitted for IPv6 extensibility.

]

## Exceptions

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

# IPAddress.AddressFamily Property

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

## Summary

Gets the address family.

## Property Value

System.Net.Sockets.AddressFamily.InterNetwork.

## Description

This property is read-only.