

# System.Net.IPEndPoint Class

```
[ILAsm]
.class public serializable IPEndPoint extends System.Net.EndPoint

[C#]
public class IPEndPoint: EndPoint
```

## 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 a network endpoint as an Internet Protocol (IP) address and a port number.

## Inherits From: System.Net.EndPoint

**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

The `System.Net.IPEndPoint` class contains the IP address of a host system and the number of a port to access on the host. The `System.Net.IPEndPoint` class represents a connection point used by the `System.Net.Sockets.Socket` class.

# 1 IPEndPoint(System.Int64, System.Int32)

## 2 Constructor

```
3 [ILAsm]  
4 public rtspecialname specialname instance void .ctor(int64 address, int32  
5 port)  
  
6 [C#]  
7 public IPEndPoint(long address, int port)
```

## 8 Summary

9 Constructs and initializes a new instance of the `System.Net.IPEndPoint` class with the  
10 specified address and port number.

## 11 Parameters

Parameter	Description
<i>address</i>	A <code>System.Int64</code> containing the IP address of the endpoint.
<i>port</i>	A <code>System.Int32</code> containing the port number to use when accessing <i>address</i> . Specify zero to indicate any available port.

## 13 Exceptions

Exception	Condition
<b>System.ArgumentOutOfRangeException</b>	<i>port</i> is less than <code>System.Net.IPEndPoint.MinPort</code> or greater than <code>System.Net.IPEndPoint.MaxPort</code> .  A negative number was specified for <i>address</i> .

# 1 IPEndPoint(System.Net.IPAddress, 2 System.Int32) Constructor

```
3 [ILAsm]  
4 public rtspecialname specialname instance void .ctor(class  
5 System.Net.IPAddress address, int32 port)  
  
6 [C#]  
7 public IPEndPoint(IPAddress address, int port)
```

## 8 Summary

9 Constructs and initializes a new instance of the `System.Net.IPEndPoint` class with the  
10 specified address and port number.

## 11 Parameters

Parameter	Description
<i>address</i>	A <code>System.Net.IPAddress</code> instance containing the IP address of the endpoint.
<i>port</i>	The port number to use when accessing <i>address</i> . Specify zero to indicate any available port.

## 12 Exceptions 13

Exception	Condition
<b>System.ArgumentNullException</b>	<i>address</i> is null.
<b>System.ArgumentOutOfRangeException</b>	<i>port</i> is less than <code>System.Net.IPEndPoint.MinPort</code> or greater than <code>System.Net.IPEndPoint.MaxPort</code> .

# 1 IPEndPoint.MaxPort Field

```
2 [ILAsm]  
3 .field public static literal int32 MaxPort = 65535  
  
4 [C#]  
5 public const int MaxPort = 65535
```

## 6 Summary

7 Specifies the maximum value that can be assigned to the `System.Net.IPEndPoint.Port`  
8 property.

## 9 Description

10 This field is read-only. The value of this field is 65535.

# 1 IPEndPoint.MinPort Field

```
2 [ILAsm]  
3 .field public static literal int32 MinPort = 0  
  
4 [C#]  
5 public const int MinPort = 0
```

## 6 Summary

7 Specifies the minimum value that can be assigned to the `System.Net.IPEndPoint.Port`  
8 property.

## 9 Description

10 This field is read-only. The value of this field is zero.

# IPEndPoint.Create(System.Net.SocketAddress) Method

```
[ILAsm]  
.method public hidebysig virtual class System.Net.EndPoint Create(class  
System.Net.SocketAddress socketAddress)  
  
[C#]  
public override EndPoint Create(SocketAddress socketAddress)
```

## Summary

Returns a new *System.Net.IPEndPoint* instance containing the address information from the specified *System.Net.SocketAddress* instance.

## Parameters

Parameter	Description
<i>socketAddress</i>	A <i>System.Net.SocketAddress</i> instance that provides the address information for the new <i>System.Net.IPEndPoint</i> instance.

## Return Value

A new *System.Net.IPEndPoint* instance containing the address information from the specified *System.Net.SocketAddress* instance.

## Description

[*Note:* This method overrides *System.Net.EndPoint.Create*.]

## Exceptions

Exception	Condition
<b>System.ArgumentException</b>	The <i>AddressFamily</i> of the specified <i>System.Net.SocketAddress</i> is not equal to the <i>AddressFamily</i> of the current instance.

# 1 IPEndPoint.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 type and value.

## 9 Parameters

Parameter	Description
<i>comparand</i>	The Object to compare to the current instance.

## 11 Return Value

12 `true` if *comparand* represents the same endpoint as the current instance. If *comparand*  
13 is a null reference or is not an instance of `System.Net.IPEndPoint`, returns `false`.

## 14 Description

15 Two `System.Net.IPEndPoint` instances are equal if their  
16 `System.Net.IPEndPoint.Address` and `System.Net.IPEndPoint.Port` properties  
17 contain the same values.

18  
19 [Note: This method overrides `System.Object.Equals`.]  
20  
21  
22

# 1 IPEndPoint.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



# 1 IPEndPoint.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` containing the IP address, in dotted-quad notation, followed by a  
10 colon and the port number for the specified endpoint, for example, 127.0.0.1:80.

## 11 Description

12 [*Note:* This method overrides `System.Object.ToString.`]  
13  
14  
15

# 1 IPEndPoint.Address Property

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

## 9 Summary

10 Gets or sets the IP address of the endpoint.

## 11 Property Value

12 A System.Net.IPAddress instance containing the IP address of the end point.

# 1 IPEndPoint.AddressFamily Property

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

## 8 Summary

9 Gets the Internet Protocol (IP) address family.

## 10 Property Value

11 Returns System.Net.Sockets.AddressFamily.InterNetwork.

## 12 Description

13 This property is read-only.

# IPEndPoint.Port Property

```
[ILAsm]
.property int32 Port { public hidebysig specialname instance int32
get_Port() public hidebysig specialname instance void set_Port(int32
value) }

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

## Summary

Gets or sets the port number of the endpoint.

## Property Value

A System.Int32 value that is between System.Net.IPEndPoint.MinPort and System.Net.IPEndPoint.MaxPort inclusive.

## Exceptions

Exception	Condition
System.ArgumentOutOfRangeException	The value specified for a set operation was less than System.Net.IPEndPoint.MinPort or greater than System.Net.IPEndPoint.MaxPort.