

System.Net.SocketAddress Class

```
[ILASM]
.class public SocketAddress extends System.Object

[C#]
public class SocketAddress
```

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)

Type Attributes:

- DefaultMemberAttribute("Item") [Note: This attribute requires the RuntimeInfrastructure library.]

Summary

Provides a socket address stored in a **System.Byte** array.

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

At a minimum, a socket address consists of a member of the **System.Net.Sockets.AddressFamily** enumeration stored in the first two bytes of the array.

1 SocketAddress(System.Net.Sockets.AddressFamily) Constructor

```
3 [ILASM]
4 public rtspecialname specialname instance void
5 .ctor(valuetype System.Net.Sockets.AddressFamily family)
6
7 [C#]
8 public SocketAddress(AddressFamily family)
```

8 Summary

9 Constructs and initializes a new instance of the
10 **System.Net.SocketAddress** class.

11 Parameters

Parameter	Description
<i>family</i>	One of the values of the System.Net.Sockets.AddressFamily enumeration.

14 15 Description

16 This method is equivalent to
17 **System.Net.SocketAddress.SocketAddress**(*family*, 32).

18

SocketAddress(System.Net.Sockets.AddressFamily, System.Int32) Constructor

```
[ILASM]
public rtspecialname specialname instance void
.ctor(valuetype System.Net.Sockets.AddressFamily family,
int32 size)

[C#]
public SocketAddress(AddressFamily family, int size)
```

Summary

Constructs and initializes a new instance of the **System.Net.SocketAddress** class.

Parameters

Parameter	Description
<i>family</i>	One of the values of the System.Net.Sockets.AddressFamily enumeration.
<i>size</i>	A System.Int32 containing the number of bytes to allocate for the System.Byte array storing the socket address.

Description

The minimum value for *size* is 2 bytes.

Exceptions

Exception	Condition
System.ArgumentOutOfRangeException	<i>size</i> is less than 2.

1 SocketAddress.Equals(System.Object)

2 Method

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

8 Summary

9 Determines whether the current instance and the specified
10 **System.Object** represent the same socket address.

11 Parameters

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

14 Return Value

17 A **System.Boolean** where **true** indicates *comparand* is an instance of
18 the **System.Net.SocketAddress** class and contains the same data as
19 the current instance; otherwise **false**.

20 Description

21 [Note: This method overrides **System.Object.Equals**.]

1 SocketAddress.GetHashCode() Method

```
2 [ILASM]  
3 .method public hidebysig virtual int32 GetHashCode()  
4  
5 [C#]  
6 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 SocketAddress.ToString() Method

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

6 Summary

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

9 Return Value

10

11 A **System.String** representation of the current instance.

12 Description

13 The returned string contains the string representation of the address
14 family, the length of the array holding the socket address, and the
15 contents of the array from the third to the maximum element, output
16 in the following format:

17

18 **System.Net.SocketAddress.Family: System.Net.SocketAddress.S**
19 **ize:** {array[2], array[3],..., element[Size- 1]}

20

21 [*Note:* This method overrides **System.Object.ToString.**]

22 Example

23

24 The following example writes a socket address to the console.

25

26 [C#]

```
27 using System;
28 using System.Net;
29 using System.Net.Sockets;
30
31 public class SocketAddressToString{
32     public static void Main() {
33         Console.WriteLine("This is a minimal SocketAddress.");
34         SocketAddress socketAddress = new
35             SocketAddress(AddressFamily.InterNetwork);
36         Console.WriteLine("{0}", socketAddress.ToString());
37     }
38 }
39
```

```
1      The output is
2
3      This is a minimal SocketAddress.
4
5
6      InterNetwork:32:{0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
7      0,0,0,0,0,0,0,0,0}
8
```

9

1 SocketAddress.Family Property

```
2 [ILASM]
3 .property valuetype System.Net.Sockets.AddressFamily Family
4 { public hidebyref specialname instance valuetype
5   System.Net.Sockets.AddressFamily get_Family() }
6
6 [C#]
7 public AddressFamily Family { get; }
```

8 Summary

9 Gets the address family which specifies the addressing scheme used to
10 resolve an address.

11 Property Value

13 One of the values defined in the
14 **System.Net.Sockets.AddressFamily** enumeration.

15 Description

16 This property is read-only.

17
18 This property is set by the **System.Net.SocketAddress** constructors
19 and is stored in the first two bytes of the socket address array.

20

SocketAddress.Item Property

```
[ILASM]
.property unsigned int8 Item[int32 offset] { public
hidebysig specialname instance unsigned int8 get_Item(int32
offset) public hidebysig specialname instance void
set_Item(int32 offset, unsigned int8 value) }

[C#]
public byte this[int offset] { get; set; }
```

Summary

Gets or sets the element at the specified index of the **System.Byte** array storing the socket address.

Parameters

Parameter	Description
<i>offset</i>	A System.Int32 containing the zero-based index of the element to get or set.

Property Value

A **System.Byte** containing the element at the specified index.

Exceptions

Exception	Condition
System.IndexOutOfRangeException	<i>offset</i> is < 0. -or- <i>offset</i> >= System.Net.SocketAddress.Size .

1 SocketAddress.Size Property

```
2 [ILASM]  
3 .property int32 Size { public hidebysig specialname  
4 instance int32 get_Size() }  
  
5 [C#]  
6 public int Size { get; }
```

7 Summary

8 Gets the length of the socket address.

9 Property Value

10

11 A **System.Int32** containing the length of the **System.Byte** array
12 storing the socket address.

13 Description

14 This property is read-only.

15

16 This property is set by the **System.Net.SocketAddress** constructors.

17