

System.Net.SocketPermission Class

```
[ILASM]
.class public serializable SocketPermission extends
System.Security.CodeAccessPermission

[C#]
public class SocketPermission: CodeAccessPermission
```

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)

Implements:

- **System.Security.IPermission**

Summary

Secures socket connections.

Inherits From: System.Security.CodeAccessPermission

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

System.Net.SocketPermission instances control permission to accept connections or initiate socket connections. A socket permission can secure access based on host name or IP address, a port number, and a transport protocol.

The XML encoding of a **System.Net.SocketPermission** instance is defined below in EBNF format, in particular the following conventions are used:

- All non-literals in the grammar below are shown in normal type.
- All literals are in bold font.

- 1 The following meta-language symbols are used:
- 2 • '*' represents a meta-language symbol suffixing an expression
3 that can appear zero or more times.
 - 4 • '?' represents a meta-language symbol suffixing an expression
5 that can appear zero or one time.
 - 6 • '+' represents a meta-language symbol suffixing an expression
7 that can appear one or more times.
 - 8 • '(',')' is used to group literals, non-literals or a mixture of
9 literals and non-literals.
 - 10 • '|' denotes an exclusive disjunction between two expressions.
 - 11 • ':=' denotes a production rule where a left hand non-literal is
12 replaced by a right hand expression containing literals, non-
13 literals or both.

14 BuildVersion refers to the build version of the shipping CLI. This is a
15 dotted build number such as '2412.0'.

16 ECMAPubKeyToken ::= **b77a5c561934e089**

17
18 HostName refers to a host name such as `www.contoso.com`.

19
20 Portnumber denotes a **System.Int32** value indicating a port.

21
22 TransportProtocol ::= **1 | 2 | 3** /* 1 = UDP, 2 = TCP, 3 = both */

23
24 SocketPermissionXML ::=

25
26 **<IPermission class="**

27
28 **System.Net.SocketPermission,**

29
30 **System,**

31
32 **Version=1.0.BuildVersion,**

33
34 **Culture=neutral,**

35
36 **PublicKeyToken=ECMAPubKeyToken"**

37
38 **version="1"**

39
40 **(**

41
42 **Unrestricted="true"**

43
44 **)**
45

```

1
2   |
3
4   >
5
6   (<ConnectAccess>
7
8   (
9
10
11   <ENDPOINT>HostName#PortNumber#TransportProtocol</ENDPO
12   INT>
13
14
15   )+
16
17
18   </ConnectAccess>
19
20
21   )
22
23
24   |
25
26
27   >
28
29
30   (<AcceptAccess>
31
32
33   (
34
35
36   <ENDPOINT>HostName#PortNumber#TransportProtocol</ENDPO
37   INT>
38
39
40   )+
41
42
43   </AcceptAccess>
44
45
46   </IPermission>
47
48
49   )
50
51
52   |
53
54
55   />

```

1
2
3
4
5

SocketPermission(System.Security.Permissions.PermissionState) Constructor

```
[ILASM]
public rtspecialname specialname instance void
.ctor(valuetype System.Security.Permissions.PermissionState
state)

[C#]
public SocketPermission(PermissionState state)
```

Summary

Constructs and initializes a new instance of the **System.Net.SocketPermission** class with the specified **System.Security.Permissions.PermissionState** value.

Parameters

Parameter	Description
<i>state</i>	A System.Security.Permissions.PermissionState value.

Description

[*Note:* This constructor creates either fully restricted (**System.Security.Permissions.PermissionState.None**) or **System.Security.Permissions.PermissionState.Unrestricted** access to sockets.]

SocketPermission(System.Net.NetworkAccess, System.Net.TransportType, System.String, System.Int32) Constructor

```
[ILASM]
public rtspecialname specialname instance void
.ctor(valuetype System.Net.NetworkAccess access, valuetype
System.Net.TransportType transport, string hostName, int32
portNumber)

[C#]
public SocketPermission(NetworkAccess access, TransportType
transport, string hostName, int portNumber)
```

Summary

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

Parameters

Parameter	Description
<i>access</i>	A System.Net.NetworkAccess value indicating the type of access to secure.
<i>transport</i>	A System.Net.TransportType value indicating the transport type to secure. Specify System.Net.TransportType.All to create a permission that secures all transport types.
<i>hostName</i>	A System.String containing the host name for the transport address.
<i>portNumber</i>	A System.Int32 containing the port number for the transport address. Specify System.Net.SocketPermission.AllPorts create a permission that secures all ports.

Description

No exception is thrown if the specified **System.Net.TransportType** or **System.Net.NetworkAccess** is invalid.

Exceptions

Exception	Condition
System.ArgumentNullException	The <i>hostName</i> parameter is null .

1 SocketPermission.AllPorts Field

```
2 [ILASM]  
3 .field public static literal int32 AllPorts = -1  
  
4 [C#]  
5 public const int AllPorts = -1
```

6 Summary

7 Defines a constant value that represents all ports.

8
9 This field is read-only. The value of this field is -1.

10

1 SocketPermission.Copy() Method

```
2 [ILASM]  
3 .method public hidebysig virtual class  
4 System.Security.IPermission Copy()  
  
5 [C#]  
6 public override IPermission Copy()
```

7 Summary

8 Returns a new **System.Net.SocketPermission** object containing the
9 same values as the current instance.

10 Return Value

11

12 A new **System.Net.SocketPermission** containing the same values as
13 the current instance.

14 Description

15 [Note: The object returned by this method represents the same level
16 of access as the current instance.

17

18 This method overrides
19 **System.Security.CodeAccessPermission.Copy** and is implemented
20 to support the **System.Security.IPermission** interface.]

21

SocketPermission.FromXml(System.Security.SecurityElement) Method

```
[ILASM]
.method public hidebysig virtual void FromXml(class
System.Security.SecurityElement securityElement)

[C#]
public override void FromXml(SecurityElement
securityElement)
```

Summary

Reconstructs the state of a **System.Net.SocketPermission** object using the specified XML encoding.

Parameters

Parameter	Description
<i>securityElement</i>	A System.Security.SecurityElement instance containing the XML encoding used to reconstruct the state of a System.Net.SocketPermission object.

Description

The state of the current instance is changed to the state encoded in *securityElement*.

[Note: For the XML schema for this class, see the **System.Net.SocketPermission** class page.

This method overrides **System.Security.CodeAccessPermission.FromXml.**

Exceptions

Exception	Condition
System.ArgumentNullException	<i>securityElement</i> is null .
System.ArgumentException	<i>securityElement</i> is not a System.Net.SocketPermission permission element.

SocketPermission.Intersect(System.Security.IPermission) Method

```
[ILASM]
.method public hidebysig virtual class
System.Security.IPermission Intersect(class
System.Security.IPermission target)

[C#]
public override IPermission Intersect(IPermission target)
```

Summary

Returns a **System.Net.SocketPermission** object that is the intersection of the current instance and the specified object.

Parameters

Parameter	Description
<i>target</i>	A System.Net.SocketPermission instance to intersect with the current instance.

Return Value

A new **System.Net.SocketPermission** instance that represents the intersection of the current instance and *target*. If *target* is **null**, returns **null**. If the intersection is empty, returns **null**. If the current instance is unrestricted, returns a copy of *target*. If *target* is unrestricted, returns a copy of the current instance.

Description

[Note: The intersection of two permissions is a permission that secures the resources and operations secured by both permissions. Specifically, it represents the minimum permission such that any demand that passes both permissions will also pass their intersection.

This method overrides **System.Security.CodeAccessPermission.Intersect** and is implemented to support the **System.Security.IPermission** interface.]

Exceptions

1
2
3

Exception	Condition
System.ArgumentException	<i>target</i> is not of type System.Net.SocketPermission.

SocketPermission.IsSubsetOf(System.Security.IPermission) Method

```
[ILASM]
.method public hidebysig virtual bool IsSubsetOf(class
System.Security.IPermission target)

[C#]
public override bool IsSubsetOf(IPermission target)
```

Summary

Determines whether the current instance is a subset of the specified object.

Parameters

Parameter	Description
<i>target</i>	A System.Net.SocketPermission instance that is to be tested for the subset relationship.

Return Value

true if the current instance is a subset of *target*; otherwise, **false**. If the current instance is unrestricted, and *target* is not, returns **false**. If *target* is unrestricted, returns **true**. If *target* is **null** and the current instance does not secure any resources and is not unrestricted, returns **true**.

Description

The subset relationship is **true** if every resource secured by the current instance is secured by *target*.

[Note: This method overrides **System.Security.CodeAccessPermission.IsSubsetOf** and is implemented to support the **System.Security.IPermission** interface.]

Exceptions

Exception	Condition
System.ArgumentException	<i>target</i> is not null and is not of type

1
2
3

	System.Net.SocketPermission.
--	-------------------------------------

1 SocketPermission.ToXml() Method

```
2 [ILASM]  
3 .method public hidebysig virtual class  
4 System.Security.SecurityElement ToXml()  
  
5 [C#]  
6 public override SecurityElement ToXml()
```

7 Summary

8 Returns the XML encoding of the current instance.

9 Return Value

10

11 A **System.Security.SecurityElement** containing the XML encoding of
12 the state of the current instance.

13 Description

14 [Note: For the XML schema for this class, see the
15 **System.Net.SocketPermission** class page.

16
17 This method overrides
18 **System.Security.CodeAccessPermission.ToXml.**]

19

SocketPermission.Union(System.Security.IPermission) Method

```
[ILASM]
.method public hidebysig virtual class
System.Security.IPermission Union(class
System.Security.IPermission target)

[C#]
public override IPermission Union(IPermission target)
```

Summary

Returns a **System.Net.SocketPermission** that is the union of the current instance and the specified object.

Parameters

Parameter	Description
<i>target</i>	A System.Net.SocketPermission instance to combine with the current instance.

Return Value

A **System.Net.SocketPermission** instance that represents the union of the current instance and *target*. If the current instance or *target* is unrestricted, returns a **System.Net.SocketPermission** instance that is unrestricted.

Description

[Note: The result of a call to **System.Net.SocketPermission.Union** is a permission that represents all of the access to socket connections represented by the current instance as well as the access represented by *target*. Any demand that passes either the current instance or *target* passes their union.

This method overrides **System.Security.CodeAccessPermission.Union** and is implemented to support the **System.Security.IPermission** interface.]

Exceptions

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

1
2

System.ArgumentNullException	<i>target</i> is null .
System.ArgumentException	<i>target</i> is not of type System.Net.SocketPermission .