

1 System.Net.SocketPermission Class

2
3

```
4 [ILASM]  
5 .class public serializable SocketPermission extends  
6 System.Security.CodeAccessPermission  
7 [C#]  
8 public class SocketPermission: CodeAccessPermission
```

9 Assembly Info:

- 10 • Name: System
- 11 • Public Key: [00 00 00 00 00 00 00 00 04 00 00 00 00 00 00]
- 12 • Version: 1.0.x.x
- 13 • Attributes:
 - 14 ○ CLSCompliantAttribute(true)

15 Implements:

- 16 • System.Security.IPermission

17 Summary

18

19 Secures socket connections.

20 Inherits From: System.Security.CodeAccessPermission

21

22 Library: Networking

23

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

26

27 Description

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

32

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

- 36 • All non-literals in the grammar below are shown in normal type.
- 37 • 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
19 HostName refers to a host name such as `www.contoso.com`.

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

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

24
25 SocketPermissionXML ::=

26
27 **<IPermission class="**

28
29 **System.Net.SocketPermission,**

30
31 **System,**

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

34
35 **Culture=neutral,**

36
37 **PublicKeyToken=ECMAPubKeyToken"**

38
39 **version="1"**

40
41 **(**

42
43 **Unrestricted="true"**

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

1 SocketPermission(System.Security.Permissions.PermissionState) Constructor

```
3 [ILASM]
4 public rtspecialname specialname instance void
5 .ctor(valuetype System.Security.Permissions.PermissionState
6 state)
7
8 [C#]
9 public SocketPermission(PermissionState state)
```

9 Summary

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

13 Parameters

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

16 Description

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

22

1 SocketPermission(System.Net.NetworkAc 2 cess, System.Net.TransportType, 3 System.String, System.Int32) Constructor

```
4 [ILASM]  
5 public rtspecialname specialname instance void  
6 .ctor(valuetype System.Net.NetworkAccess access, valuetype  
7 System.Net.TransportType transport, string hostName, int32  
8 portNumber)  
9  
10 [C#]  
11 public SocketPermission(NetworkAccess access, TransportType  
transport, string hostName, int portNumber)
```

12 Summary

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

15 Parameters

16
17

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.

18

19 Description

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

22 Exceptions

23
24

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

25
26
27

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

1 SocketPermission.FromXml(System.Security.SecurityElement) Method

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

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

9 Summary

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

12 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.

16 Description

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

19
20 [Note: For the XML schema for this class, see the
21 **System.Net.SocketPermission** class page.

22
23 This method overrides
24 **System.Security.CodeAccessPermission.FromXml.**]

25 Exceptions

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

1 SocketPermission.Intersect(System.Security.IPermission) Method

2

```
3 [ILASM]
4 .method public hidebysig virtual class
5 System.Security.IPermission Intersect(class
6 System.Security.IPermission target)
7
8 [C#]
9 public override IPermission Intersect(IPermission target)
```

9 Summary

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

12 Parameters

13
14

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

15 Return Value

16
17

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

23 Description

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

28
29 This method overrides
30 **System.Security.CodeAccessPermission.Intersect** and is
31 implemented to support the **System.Security.IPermission**
32 interface.]

33 Exceptions

34
35

1
2
3

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

1 SocketPermission.IsSubsetOf(System.Security.IPermission) Method

```
3 [ILASM]  
4 .method public hidebysig virtual bool IsSubsetOf(class  
5 System.Security.IPermission target)  
  
6 [C#]  
7 public override bool IsSubsetOf(IPermission target)
```

8 Summary

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

11 Parameters

12
13

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

14
15
16

15 Return Value

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

22 Description

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

25
26 [Note: This method overrides
27 **System.Security.CodeAccessPermission.IsSubsetOf** and is
28 implemented to support the **System.Security.IPermission**
29 interface.]

30 Exceptions

31
32

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

1 SocketPermission.Union(System.Security. 2 IPermission) Method

```
3 [ILASM]  
4 .method public hidebysig virtual class  
5 System.Security.IPermission Union(class  
6 System.Security.IPermission target)  
  
7 [C#]  
8 public override IPermission Union(IPermission target)
```

9 Summary

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

12 Parameters

13
14

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

15
16
17

16 Return Value

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

22 Description

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

28
29 This method overrides
30 **System.Security.CodeAccessPermission.Union** and is
31 implemented to support the **System.Security.IPermission**
32 interface.]

33 Exceptions

34
35

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

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

1
2