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 00]
- *Version:* 2.0.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.

The following meta-language symbols are used:

- '*' represents a meta-language symbol suffixing an expression that can appear zero or more times.

- '?' represents a meta-language symbol suffixing an expression that can appear zero or one time.

- '+' represents a meta-language symbol suffixing an expression that can appear one or more times.

- '(',')' is used to group literals, non-literals or a mixture of literals and non-literals.

- '|' denotes an exclusive disjunction between two expressions.

- '::= ' denotes a production rule where a left hand non-literal is replaced by a right hand expression containing literals, non-literals or both.

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

ECMAPubKeyToken::= `b77a5c561934e089`

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

Portnumber denotes a `System.Int32` value indicating a port.

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

SocketPermissionXML::=

```
<IPermission class="

System.Net.SocketPermission,

System,

Version=1.0.BuildVersion,

Culture=neutral,

PublicKeyToken=ECMAPubKeyToken"

version="1"

(

Unrestricted="true"

)

|
>
(<ConnectAccess>
(
<ENDPOINT>HostName#PortNumber#TransportProtocol</ENDPOINT>
)+
</ConnectAccess>
)
|
>
(<AcceptAccess>
(
<ENDPOINT>HostName#PortNumber#TransportProtocol</ENDPOINT>
)+
</AcceptAccess>
</IPermission>
)
|
/>
```

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
state	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
access	A `System.Net.NetworkAccess` value indicating the type of access to secure.
transport	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.
hostName	A `System.String` containing the host name for the transport address.
portNumber	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 *hostName* parameter is `null`.

SocketPermission.AllPorts Field

[ILAsm]

```
.field public static literal int32 AllPorts = -1
```

[C#]

```
public const int AllPorts = -1
```

Summary

Defines a constant value that represents all ports.

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

SocketPermission.Copy() Method

```
[ILAsm]
.method public hidebysig virtual class System.Security.IPermission Copy()

[C#]
public override IPermission Copy()
```

Summary

Returns a new `System.Net.SocketPermission` object containing the same values as the current instance.

Return Value

A new `System.Net.SocketPermission` containing the same values as the current instance.

Description

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

This method overrides `System.Security.CodeAccessPermission.Copy` and is implemented to support the `System.Security.IPermission` interface.

]

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
securityElement	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	*securityElement* is `null`.
System.ArgumentException	*securityElement* 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
target	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

Exception	Condition

System.ArgumentException	*target* 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
target	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	*target* is not `null` and is not of type `System.Net.SocketPermission`.

SocketPermission.ToXml() Method

```
[ILAsm]
.method public hidebysig virtual class System.Security.SecurityElement
ToXml()

[C#]
public override SecurityElement ToXml()
```

Summary

Returns the XML encoding of the current instance.

Return Value

A `System.Security.SecurityElement` containing the XML encoding of the state of the current instance.

Description

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

This method overrides `System.Security.CodeAccessPermission.ToXml`.

]

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
target	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
System.ArgumentException	*target* is not of type `System.Net.SocketPermission`.