

# System.Net.DnsPermission Class

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

[C#]
public class DnsPermission: 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

Controls access to Domain Name System (DNS) servers on the network.

## 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

The XML encoding of a **System.Net.DnsPermission** instance is defined below in EBNF format. 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.

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

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

13  
14          ECMAPubKeyToken ::= **b77a5c561934e089**

15  
16          DnsPermissionXML ::=

17  
18  
19          <IPermission

20  
21  
22          class="

23  
24  
25          System.Net.DnsPermission,

26  
27  
28          System,

29  
30  
31          Version=1.0.BuildVersion,

32  
33  
34          Culture=neutral,

35  
36  
37          PublicKeyToken=ECMAPubKeyToken"

38  
39  
40          version="1"

41  
42          (  
43  
44

45  
46          Unrestricted="true"/>

47  
48          )  
49

1  
2  
3  
4  
5

| />

# 1 DnsPermission(System.Security.Permissio 2 ns.PermissionState) Constructor

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

## 9 Summary

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

## 13 Parameters

14  
15

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

16  
17

## 17 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 DNS information.]

## 22 Exceptions

23  
24

Exception	Condition
<b>System.ArgumentException</b>	<i>state</i> is not a valid <b>System.Security.Permissions.PermissionState</b> value.

25  
26  
27

# 1 DnsPermission.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.DnsPermission** object containing the  
9 same values as the current instance.

## 10 Return Value

11

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

## 14 Description

15 [*Note:* The object returned by this method represents the same access  
16 to DNS information as the current instance.]

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

21

# 1 DnsPermission.FromXml(System.Security. 2 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.DnsPermission** object using  
11 the specified XML encoding.

## 12 Parameters

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

## 15 Description

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

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

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

## 25 Exceptions

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

28  
29  
30

# 1 DnsPermission.Intersect(System.Security. 2 IPermission) Method

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

## 9 Summary

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

## 12 Parameters

13  
14

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

15  
16  
17

## 16 Return Value

18 A new **System.Net.DnsPermission** instance that represents the  
19 intersection of the current instance and *target*. If the intersection is  
20 empty, returns **null**. If the current instance is unrestricted, returns a  
21 copy of *target*. If *target* is unrestricted, returns a copy of the current  
22 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.ArgumentNullException	<i>target</i> is <b>null</b> .

# 1 DnsPermission.IsSubsetOf(System.Security.Permission) 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 <b>System.Net.DnsPermission</b> instance that is to be tested for the subset relationship.

14  
15  
16

## 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**, returns **false**.

## 20 Description

21 [Note: The current instance is a subset of *target* if the current instance  
22 specifies a set of accesses to resources that is wholly contained by  
23 *target*. For example, a permission that represents read access to a file  
24 is a subset of a permission that represents read and write access to  
25 the file.

26  
27  
28  
29  
30

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

31

# 1 DnsPermission.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 encoding for this class, see the  
15 **System.Net.DnsPermission** class page.

16

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

19

# 1 DnsPermission.Union(System.Security.IPe 2 rmission) 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 new **System.Net.DnsPermission** object that is the union  
11 of the current instance and the specified object.

## 12 Parameters

13  
14

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

15  
16  
17

## 16 Return Value

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

## 22 Description

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

1  
2

<b>System.ArgumentNullException</b>	<i>target</i> is <b>null</b> .
-------------------------------------	--------------------------------