

# 1 System.Security.Permissions.SecurityAction Enum

2  
3  
4

```
5 [ILASM]  
6 .class public sealed serializable SecurityAction extends  
7 System.Enum  
  
8 [C#]  
9 public enum SecurityAction
```

## 10 Assembly Info:

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

## 16 Summary

17

18 Specifies security actions that can be performed using declarative  
19 security.

## 20 Inherits From: System.Enum

21

22 Library: BCL

23

## 24 Description

25 [Note: For information about using declarative security and security  
26 actions, see Partition II of the CLI Specification.]

27

28 [Note: Declarative security is specified using types derived from  
29 **System.Security.Permissions.SecurityAttribute**. The following  
30 table describes the attribute targets supported by each of the security  
31 actions.

Security action	Attribute Targets
Assert	Class, Method
Demand	Class, Method
Deny	Class, Method
InheritanceDemand	Class, Method
LinkDemand	Class, Method
PermitOnly	Class, Method

RequestMinimum	Assembly
RequestOptional	Assembly
RequestRefuse	Assembly

1  
2  
3  
4

For additional information on attribute targets, see  
**System.Attribute.**]

# 1 SecurityAction.Assert Field

```
2 [ILASM]  
3 .field public static literal valuetype  
4 System.Security.Permissions.SecurityAction Assert = 3  
  
5 [C#]  
6 Assert = 3
```

## 7 Summary

8 Specifies that callers of the code performing the assert need not have  
9 the permissions specified by the current security attribute, and that a  
10 check for any such permission can stop after the code that asserted it.  
11 [*Note:* An assert can change the default behavior of a security check  
12 (such as that caused by a Demand, LinkDemand, etc.).]

13 This action can be applied to classes and methods.

14 [*Note:* This action should only be used by code that can assure that its  
15 callers cannot manipulate it to abuse the asserted permission.]  
16  
17

18

# 1 SecurityAction.Demand Field

```
2 [ILASM]  
3 .field public static literal valuetype  
4 System.Security.Permissions.SecurityAction Demand = 2  
  
5 [C#]  
6 Demand = 2
```

## 7 Summary

8 Specified that all callers are required to have the permissions specified  
9 by the current security attribute.

10  
11 This action can be applied to classes and methods.

12

# 1 SecurityAction.Deny Field

```
2 [ILASM]  
3 .field public static literal valuetype  
4 System.Security.Permissions.SecurityAction Deny = 4  
  
5 [C#]  
6 Deny = 4
```

## 7 Summary

8 Specifies that access to the resource or operation described by the  
9 current security attribute be denied to callers, even if they have been  
10 granted permission to access it. [*Note:*  
11 **System.Security.Permissions.SecurityAction.Deny** causes a  
12 security check for the permissions specified by the current security  
13 attribute to fail even when it would otherwise succeed.]

14  
15 This action can be applied to classes and methods.

16

# 1 SecurityAction.InheritanceDemand Field

```
2 [ILASM]  
3 .field public static literal valuetype  
4 System.Security.Permissions.SecurityAction  
5 InheritanceDemand = 7  
  
6 [C#]  
7 InheritanceDemand = 7
```

## 8 Summary

9 Specifies the permissions that a derived class is required to have.  
10 When the target is a class, classes inheriting from the target are  
11 required to have the permissions specified by the current security  
12 attribute. When the target is a method, classes overriding the target  
13 are required to have the permissions specified by the current security  
14 attribute.

15  
16 This action can be applied to classes and methods.

17

# 1 SecurityAction.LinkDemand Field

```
2 [ILASM]  
3 .field public static literal valuetype  
4 System.Security.Permissions.SecurityAction LinkDemand = 6  
  
5 [C#]  
6 LinkDemand = 6
```

## 7 Summary

8 Specifies that the immediate caller be required to have the specified  
9 permissions.

10  
11 This action can be applied to classes and methods.

12

# 1 SecurityAction.PermitOnly Field

```
2 [ILASM]  
3 .field public static literal valuetype  
4 System.Security.Permissions.SecurityAction PermitOnly = 5  
  
5 [C#]  
6 PermitOnly = 5
```

## 7 Summary

8 Specifies that access is limited to only those resources or operations  
9 specified by the current security attribute, even if the code has been  
10 granted permission to access others. A security check for a permission  
11 not described by the current security attribute fails regardless of  
12 whether or not callers have been granted this permission.

13  
14 This action can be applied to classes and methods.

15

# 1 SecurityAction.RequestMinimum Field

```
2 [ILASM]  
3 .field public static literal valuetype  
4 System.Security.Permissions.SecurityAction RequestMinimum =  
5 8  
  
6 [C#]  
7 RequestMinimum = 8
```

## 8 Summary

9 Specifies that the current security attribute describes the minimum  
10 permissions required for an assembly to run.

11  
12 This action can be applied to assemblies.

13

# 1 SecurityAction.RequestOptional Field

```
2 [ILASM]  
3 .field public static literal valuetype  
4 System.Security.Permissions.SecurityAction RequestOptional  
5 = 9  
  
6 [C#]  
7 RequestOptional = 9
```

## 8 Summary

9 Specifies that the current security attribute describes optional  
10 permissions that an assembly can be granted.

11  
12 This action can be applied to assemblies.

13

# 1 SecurityAction.RequestRefuse Field

```
2 [ILASM]  
3 .field public static literal valuetype  
4 System.Security.Permissions.SecurityAction RequestRefuse =  
5 10  
  
6 [C#]  
7 RequestRefuse = 10
```

## 8 Summary

9 Specifies that the current security attribute describes resources or  
10 operations that an assembly cannot access.

11  
12 This action can be applied to assemblies.

13