

# System.Security.Permissions.SecurityAction Enum

```
[ILAsm]  
.class public sealed serializable SecurityAction extends System.Enum  
  
[C#]  
public enum SecurityAction
```

## Assembly Info:

- *Name:* mscorlib
- *Public Key:* [00 00 00 00 00 00 00 00 04 00 00 00 00 00 00]
- *Version:* 2.0.x.x
- *Attributes:*
  - CLSCompliantAttribute(true)

## Summary

Specifies security actions that can be performed using declarative security.

## Inherits From: System.Enum

**Library:** BCL

## Description

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

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

For additional information on attribute targets, see `System.Attribute`.

]

# SecurityAction.Assert Field

```
[ILAsm]  
.field public static literal valuetype  
System.Security.Permissions.SecurityAction Assert = 3  
  
[C#]  
Assert = 3
```

## Summary

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

This action can be applied to classes and methods.

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

# SecurityAction.Demand Field

```
[ILAsm]  
.field public static literal valuetype  
System.Security.Permissions.SecurityAction Demand = 2  
  
[C#]  
Demand = 2
```

## Summary

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

This action can be applied to classes and methods.

# SecurityAction.Deny Field

```
[ILAsm]  
.field public static literal valuetype  
System.Security.Permissions.SecurityAction Deny = 4  
  
[C#]  
Deny = 4
```

## Summary

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

This action can be applied to classes and methods.

# SecurityAction.InheritanceDemand Field

```
[ILAsm]  
.field public static literal valuetype  
System.Security.Permissions.SecurityAction InheritanceDemand = 7  
  
[C#]  
InheritanceDemand = 7
```

## Summary

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

This action can be applied to classes and methods.

## SecurityAction.LinkDemand Field

```
[ILAsm]  
.field public static literal valuetype  
System.Security.Permissions.SecurityAction LinkDemand = 6  
  
[C#]  
LinkDemand = 6
```

### Summary

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

This action can be applied to classes and methods.

## SecurityAction.PermitOnly Field

```
[ILAsm]  
.field public static literal valuetype  
System.Security.Permissions.SecurityAction PermitOnly = 5
```

```
[C#]  
PermitOnly = 5
```

### Summary

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

This action can be applied to classes and methods.



# SecurityAction.RequestMinimum Field

```
[ILAsm]  
.field public static literal valuetype  
System.Security.Permissions.SecurityAction RequestMinimum = 8  
  
[C#]  
RequestMinimum = 8
```

## Summary

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

This action can be applied to assemblies.

## SecurityAction.RequestOptional Field

```
[ILAsm]  
.field public static literal valuetype  
System.Security.Permissions.SecurityAction RequestOptional = 9  
  
[C#]  
RequestOptional = 9
```

### Summary

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

This action can be applied to assemblies.

## SecurityAction.RequestRefuse Field

```
[ILAsm]  
.field public static literal valuetype  
System.Security.Permissions.SecurityAction RequestRefuse = 10  
  
[C#]  
RequestRefuse = 10
```

### Summary

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

This action can be applied to assemblies.