Docs / .NET / .NET API browser / System.Security / C#

SuppressUnmanagedCodeSecurity Attribute Class

Reference

Definition

Namespace: System.Security

Assembly: System.Runtime.dll

Allows managed code to call into unmanaged code without a stack walk. This class cannot be inherited.

In this article

C# Copy

```
[System.AttributeUsage(System.AttributeTargets.Class |
System.AttributeTargets.Delegate | System.AttributeTargets.Interface |
System.AttributeTargets.Method, AllowMultiple=true, Inherited=false)]
public sealed class SuppressUnmanagedCodeSecurityAttribute : Attribute
```

Inheritance Object → Attribute → SuppressUnmanagedCodeSecurityAttribute

Attributes AttributeUsageAttribute

Remarks

ⓘ **Important**

Partially trusted code is no longer supported. This attribute has no effect in .NET Core.

⊗ **Caution**

Use this attribute with extreme care. Incorrect use can create security weaknesses.

This attribute can be applied to methods that want to call into native code without incurring the performance loss of a run-time security check when doing so. The stack walk performed when calling unmanaged code is omitted at run time, resulting in substantial performance savings. Using this attribute in a class applies it to all contained methods.

Generally, whenever managed code calls into unmanaged code (by PInvoke or COM interop into native code), there is a demand for the `UnmanagedCode` permission to ensure all callers have the necessary permission to allow this. By applying this explicit attribute, developers can suppress the demand at run time. The developer must take responsibility for assuring that the transition into unmanaged code is sufficiently protected by other means. The demand for the `UnmanagedCode` permission will still occur at link time. For example, if function A calls function B and function B is marked with SuppressUnmanagedCodeSecurityAttribute, function A will be checked for unmanaged code permission during just-in-time compilation, but not subsequently during run time.

This attribute is only effective when applied to PInvoke methods (or classes that contain PInvoke methods) or the definition of an interface through which interop calls will be made. It will be ignored in all other contexts.

This attribute is useful for implementing a class that provides access to system resources through unmanaged code. Code that does not have permission to access unmanaged code can call a class with this attribute to access unmanaged code. This is only safe if the writer of the class with this attribute has programmed the class to be secure. If not, this attribute is dangerous and can allow the code that uses it to be misused.

This is not a declarative security attribute, but a regular attribute (it derives from Attribute, not SecurityAttribute).

Constructors

SuppressUnmanagedCodeSecurityAttribute()	Initializes a new instance of the SuppressUnmanagedCodeSecurityAttribute class.

Properties

TypeId	When implemented in a derived class, gets a unique identifier for this Attribute. (Inherited from Attribute)

Methods

Equals(Object)	Returns a value that indicates whether this instance is equal to a specified object. (Inherited from Attribute)
GetHashCode()	Returns the hash code for this instance. (Inherited from Attribute)
GetType()	Gets the Type of the current instance. (Inherited from Object)
IsDefaultAttribute()	When overridden in a derived class, indicates whether the value of this instance is the default value for the derived class. (Inherited from Attribute)
Match(Object)	When overridden in a derived class, returns a value that indicates whether this instance equals a specified object. (Inherited from Attribute)
MemberwiseClone()	Creates a shallow copy of the current Object. (Inherited from Object)
ToString()	Returns a string that represents the current object. (Inherited from Object)

Applies to

Product	**Versions**
.NET	Core 2.0, Core 2.1, Core 2.2, Core 3.0, Core 3.1, 5, 6, 7 Preview 3
.NET Framework	1.1, 2.0, 3.0, 3.5, 4.0, 4.5, 4.5.1, 4.5.2, 4.6, 4.6.1, 4.6.2, 4.7, 4.7.1, 4.7.2, 4.8
.NET Standard	2.0, 2.1

Product	Versions
Xamarin.iOS	10.8
Xamarin.Mac	3.0

See also

- Extending Metadata Using Attributes

Recommended content

GCHandle Struct (System.Runtime.InteropServices)

Provides a way to access a managed object from unmanaged memory.

Default Marshalling for Strings - .NET Framework

Review the default marshalling behavior for strings in interfaces, platform invoke, structures, & fixed-length string buffers in .NET.

Marshal.GetDelegateForFunctionPointer Method (System.Runtime.InteropServices)

Converts an unmanaged function pointer to a delegate.

Customizing structure marshalling - .NET

Learn how to customize how .NET marshals structures to a native representation.

UnmanagedType Enum (System.Runtime.InteropServices)

Identifies how to marshal parameters or fields to unmanaged code.

Marshalling Different Types of Arrays - .NET Framework

Marshal different array types, like integers by value or reference, 2-dimensional integers by value, strings by value, and structures with integers or strings.

Default Marshalling for Arrays - .NET Framework

Understand default marshalling for arrays. Review managed arrays, unmanaged arrays, passing array parameters to .NET code, and passing arrays to COM.

Marshal.StructureToPtr Method (System.Runtime.InteropServices)

Marshals data from a managed object to an unmanaged block of memory.

Show more