

System.Reflection.PropertyInfo Class

```
[ILASM]
.class public abstract serializable PropertyInfo extends
System.Reflection.MemberInfo

[C#]
public abstract class PropertyInfo: MemberInfo
```

Assembly Info:

- *Name:* mscorlib
- *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)

Summary

Provides access to property metadata.

Inherits From: System.Reflection.MemberInfo

Library: Reflection

Thread Safety: This type is safe for multithreaded operations.

Description

A property is a named aspect of an object's state whose value is typically accessible through **get** and **set** accessors. [*Note:* Properties can be read-only, in which case the **set** accessor is not available.]

Several methods in this class assume that the **Get** and **Set** accessors of a property have certain formats. The signatures of the accessors are required to match the following conventions:

- The return type of the **Get** accessor and the last argument of the **Set** accessor are required to be identical to the type of the property reflected by the current instance.
- The **Get** and **Set** accessors are required to have the same number, type, and order of indices.

If this format is not followed, the behavior of the **System.Reflection.PropertyInfo.GetValue** and **System.Reflection.PropertyInfo.SetValue** methods is undefined.

1 PropertyInfo() Constructor

```
2 [ILASM]  
3 family rtspecialname specialname instance void .ctor()  
4 [C#]  
5 protected PropertyInfo()
```

6 Summary

7 Constructs a new instance of the **System.Reflection.PropertyInfo**
8 class.

9

PropertyInfo.GetAccessors(System.Boolean) Method

```
[ILASM]
.method public hidebysig virtual abstract class
System.Reflection.MethodInfo[] GetAccessors(bool nonPublic)

[C#]
public abstract MethodInfo[] GetAccessors(bool nonPublic)
```

Summary

Returns an array whose elements reflect the public and, if specified, non-public **Get**, **Set**, and other accessors of the property reflected by the current instance.

Parameters

Parameter	Description
<i>nonPublic</i>	A System.Boolean value that indicates whether non-public accessors will be included in the return value. Specify true to include the non-public accessors; otherwise, specify false .

Return Value

An array of **System.Reflection.MethodInfo** objects whose elements reflect the **Get**, **Set**, and other accessors of the property reflected by the current instance. If *nonPublic* is **true**, this array contains public and non-public accessors. If *nonPublic* is **false**, this array contains only public accessors. If no accessors with the specified visibility are found, returns an array with zero elements.

Behaviors

As described above.

Permissions

Permission	Description
System.Security.Permissions.ReflectionPermission	Requires permission to reflect non-public members of a type in loaded assemblies. See System.Security.Permissions.ReflectionPermissionFlag.TypeInformation .

1 PropertyInfo.GetAccessors() Method

```
2 [ILASM]  
3 .method public hidebysig instance class  
4 System.Reflection.MethodInfo[] GetAccessors()  
  
5 [C#]  
6 public MethodInfo[] GetAccessors()
```

7 Summary

8 Returns an array whose elements reflect the public **Get**, **Set**, and
9 other accessors of the property reflected by the current instance.

10 Return Value

11

12 An array of **System.Reflection.MethodInfo** objects that reflect the
13 public **Get**, **Set**, and other accessors of the property reflected by the
14 current instance, if found; otherwise, returns an array with zero
15 elements.

16 Description

17 This method is equivalent to
18 **System.Reflection.PropertyInfo.GetAccessors(false)**.

19

1 PropertyInfo.GetGetMethod(System.Boolean) Method

```
3 [ILASM]  
4 .method public hidebysig virtual abstract class  
5 System.Reflection.MethodInfo GetGetMethod(bool nonPublic)  
  
6 [C#]  
7 public abstract MethodInfo GetGetMethod(bool nonPublic)
```

8 Summary

9 Returns the public or, if specified, non-public **Get** accessor for the
10 property reflected by the current instance.

11 Parameters

12
13

Parameter	Description
<i>nonPublic</i>	A System.Boolean value that indicates whether non-public Get accessors will be included in the return value. Specify true to include non-public accessors; otherwise, specify false .

14
15
16

Return Value

17 If *nonPublic* is **true**, returns a **System.Reflection.MethodInfo**
18 instance that reflects the **Get** accessor for the property reflected by
19 the current instance if that accessor exists. If *nonPublic* is **false** and
20 the **Get** accessor is non-public, or *nonPublic* is **true** but no **Get**
21 accessor exists for the property reflected by the current instance,
22 returns **null**.

23 Behaviors

24 As described above.

25 Exceptions

26
27

Exception	Condition
System.MethodAccessException	<i>nonPublic</i> is true , the Get accessor for the property reflected by the current instance is non-public, and the caller does not have System.Security.Permissions.ReflectionPermission to reflect on non-public methods.

1
2
3
4

Permissions

Permission	Description
System.Security.Permissions.ReflectionPermission	Requires permission to reflect non-public members of a type in loaded assemblies. See System.Security.Permissions.ReflectionPermissionFlag.TypeInformation .

5
6
7

1 PropertyInfo.GetGetMethod() Method

```
2 [ILASM]  
3 .method public hidebysig instance class  
4 System.Reflection.MethodInfo GetGetMethod()  
  
5 [C#]  
6 public MethodInfo GetGetMethod()
```

7 Summary

8 Returns the public **Get** accessor for the property reflected by the
9 current instance.

10 Return Value

11

12 A **System.Reflection.MethodInfo** instance that reflects the public
13 **Get** accessor for the property reflected by the current instance.
14 Returns **null** if no public **Get** accessor exists.

15

1 PropertyInfo.GetIndexParameters() 2 Method

```
3 [ILASM]  
4 .method public hidebysig virtual abstract class  
5 System.Reflection.ParameterInfo[] GetIndexParameters()  
  
6 [C#]  
7 public abstract ParameterInfo[] GetIndexParameters()
```

8 Summary

9 Returns an array of the indexers of the property reflected by the
10 current instance.

11 Return Value

12

13 An array of **System.Reflection.ParameterInfo** objects that reflect
14 the indexers of the property reflected by the current instance. If no
15 indexers exist for the property reflected by the current instance,
16 returns an array with zero elements.

17 Behaviors

18 As described above.

19 Exceptions

20

21

Exception	Condition
System.MethodAccessException	The property reflected by the current instance is visible but its Get and Set accessors are not, and the caller does not have System.Security.Permissions.ReflectionPermission

22

23 Permissions

24

25

Permission	Description
System.Security.Permissions.ReflectionPermission	Requires permission to reflect non-public members of a type in loaded assemblies. See System.Security.Permissions.ReflectionPermissionFlag.TypeInformation .

26

27

28

1 PropertyInfo.GetSetMethod(System.Boolean) Method

```
3 [ILASM]  
4 .method public hidebysig virtual abstract class  
5 System.Reflection.MethodInfo GetSetMethod(bool nonPublic)  
  
6 [C#]  
7 public abstract MethodInfo GetSetMethod(bool nonPublic)
```

8 Summary

9 Returns the public or, if specified, non-public **Set** accessor for the
10 property reflected by the current instance.

11 Parameters

12
13

Parameter	Description
<i>nonPublic</i>	A System.Boolean value that indicates whether non-public Set accessors are included in the return value. Specify true if non-public accessors are to be included; otherwise, specify false .

14
15
16

Return Value

17 If *nonPublic* is **true**, returns a **System.Reflection.MethodInfo**
18 instance that reflects the **Set** accessor for the property reflected by
19 the current instance if that accessor exists. If *nonPublic* is **false** and
20 the **Set** accessor is non-public, or *nonPublic* is **true** but no **Set**
21 accessor exists for the property reflected by the current instance,
22 returns **null**.

23 Behaviors

24 As described above.

25 Exceptions

26
27

Exception	Condition
System.MethodAccessException	<i>nonPublic</i> is true , the Set accessor for the property reflected by the current instance is non-public, and the caller does not have System.Security.Permissions.ReflectionPermission to reflect on non-public methods.

1
2
3
4

Permissions

Permission	Description
System.Security.Permissions.ReflectionPermission	Requires permission to reflect non-public members of a type in loaded assemblies. See System.Security.Permissions.ReflectionPermissionFlag.TypeInformation .

5
6
7

1 PropertyInfo.GetSetMethod() Method

```
2 [ILASM]  
3 .method public hidebysig instance class  
4 System.Reflection.MethodInfo GetSetMethod()  
  
5 [C#]  
6 public MethodInfo GetSetMethod()
```

7 Summary

8 Returns the public **Set** accessor for the property reflected by the
9 current instance.

10 Return Value

11

12 A **System.Reflection.MethodInfo** instance that reflects the public
13 **Set** accessor for the property reflected by the current instance.
14 Returns **null** if no public **Set** accessor exists.

15 Description

16 This method is equivalent to
17 **System.Reflection.PropertyInfo.GetSetMethod(false)**.

18

1 **PropertyInfo.GetValue(System.Object,**
 2 **System.Reflection.BindingFlags,**
 3 **System.Reflection.Binder,**
 4 **System.Object[],**
 5 **System.Globalization.CultureInfo) Method**

```
6 [ILASM]
7 .method public hidebysig virtual abstract object
8 GetValue(object obj, valuetype
9 System.Reflection.BindingFlags invokeAttr, class
10 System.Reflection.Binder binder, class System.Object[]
11 index, class System.Globalization.CultureInfo culture)

12 [C#]
13 public abstract object GetValue(object obj, BindingFlags
14 invokeAttr, Binder binder, object[] index, CultureInfo
15 culture)
```

16 **Summary**

17 Returns the value of the property that is reflected by the current
 18 instance in the specified object and corresponds to the specified
 19 criteria.

20 **Parameters**

21
 22

Parameter	Description
<i>obj</i>	The object whose property value is returned. Specify null to invoke a static Get accessor on the property reflected by the current instance.
<i>invokeAttr</i>	A System.Reflection.BindingFlags value that controls the binding process. [Note: Specify System.Reflection.BindingFlags.Public or System.Reflection.BindingFlags.NonPublic , and System.Reflection.BindingFlags.Instance or System.Reflection.BindingFlags.Static ; or this method will not invoke any get accessors of the property reflected by the current instance.]
<i>binder</i>	A System.Reflection.Binder that enables the binding, coercion of argument types, invocation of members, and retrieval of System.Reflection.MemberInfo objects via reflection. If <i>binder</i> is null , the default binder is used.
<i>index</i>	An array of objects that is an index or values for indexed properties. This value is required to be null for non-indexed properties.
<i>culture</i>	The only defined value for this parameter is null .

1
2
3
4
5
6
7
8
9

Return Value

A **System.Object** that contains the property value for *obj*.

Behaviors

As described above.

Exceptions

Exception	Condition
System.ArgumentException	<i>index</i> does not contain the exact type of arguments needed. The Get accessor of the property reflected by the current instance was not found.
System.MethodAccessException	The Get accessor of the property reflected by the current instance is non-public and the caller does not have System.Security.Permissions.ReflectionPermission to reflect on non-public methods.
System.Reflection.TargetException	The property reflected by the current instance is not static, and <i>obj</i> is null or is of a type that does not implement the property reflected by the current instance.
System.Reflection.TargetParameterCountException	The current instance reflects an indexer and <i>index.Length</i> does not equal the rank of the indexer.

10
11
12
13

Permissions

Permission	Description
System.Security.Permissions.ReflectionPermission	Requires permission to reflect non-public members of a type in loaded assemblies. See System.Security.Permissions.ReflectionPermissionFlag.MemberAccess .

14
15
16

1 PropertyInfo.GetValue(System.Object, 2 System.Object[]) Method

```
3 [ILASM]  
4 .method public hidebysig virtual object GetValue(object  
5 obj, class System.Object[] index)  
  
6 [C#]  
7 public virtual object GetValue(object obj, object[] index)
```

8 Summary

9 Returns the value of the property reflected by the current instance in
10 the specified object, using the specified index values.

11 Parameters

12
13

Parameter	Description
<i>obj</i>	The object whose property value will be returned. Specify null to invoke a static Get accessor of a property.
<i>index</i>	An array of objects that is an index of values for indexed properties. This value is required to be null for non-indexed properties.

14
15
16

15 Return Value

17 A **System.Object** that contains the property value for *obj*.

18 Behaviors

19 As described above.

20 Exceptions

21
22

Exception	Condition
System.ArgumentException	<i>index</i> does not contain the exact type of arguments needed. -or- The Get accessor of the property reflected by the current instance is not found.
System.MethodAccessException	The Get accessor of the property reflected by the

	current instance is non-public and the caller does not have System.Security.Permissions.ReflectionPermission to reflect on non-public methods.
System.Reflection.TargetException	The property reflected by the current instance is not static, and <i>obj</i> is null or is of a type that does not implement the property reflected by the current instance.
System.Reflection.TargetParameterCountException	The current instance reflects an indexer and <i>index.Length</i> does not equal the rank of the indexer.

1
2
3
4

Permissions

Permission	Description
System.Security.Permissions.ReflectionPermission	Requires permission to reflect non-public members of a type in loaded assemblies. See System.Security.Permissions.ReflectionPermissionFlag.MemberAccess .

5
6
7

1 **PropertyInfo.SetValue(System.Object,**
 2 **System.Object,**
 3 **System.Reflection.BindingFlags,**
 4 **System.Reflection.Binder,**
 5 **System.Object[],**
 6 **System.Globalization.CultureInfo) Method**

```

7 [ILASM]
8 .method public hidebysig virtual abstract void
9 SetValue(object obj, object value, valuetype
10 System.Reflection.BindingFlags invokeAttr, class
11 System.Reflection.Binder binder, class System.Object[]
12 index, class System.Globalization.CultureInfo culture)

13 [C#]
14 public abstract void SetValue(object obj, object value,
15 BindingFlags invokeAttr, Binder binder, object[] index,
16 CultureInfo culture)
  
```

17 **Summary**

18 Sets the value of the property that is reflected by the current instance
 19 on the specified objects and corresponds to the specified properties.

20 **Parameters**

21
 22

Parameter	Description
<i>obj</i>	The object whose property value is returned. Specify null to invoke a static Set accessor on the property reflected by the current instance.
<i>value</i>	A object that contains the new value for the property.
<i>invokeAttr</i>	A System.Reflection.BindingFlags value that controls the binding process. [Note: Specify System.Reflection.BindingFlags.Public or System.Reflection.BindingFlags.NonPublic , and System.Reflection.BindingFlags.Instance or System.Reflection.BindingFlags.Static ; otherwise, this method will not invoke any Set accessors of the property reflected by the current instance.]
<i>binder</i>	A System.Reflection.Binder that enables the binding, coercion of argument types, invocation of members, and retrieval of System.Reflection.MemberInfo objects via reflection. If <i>binder</i> is null , the default binder is used.
<i>index</i>	An array of objects that is an index or values for indexed properties. This value is required to be null for non-indexed properties.

<i>culture</i>	The only defined value for this parameter is null .
----------------	--

1
2
3
4
5
6
7

Behaviors

As described above.

Exceptions

Exception	Condition
System.ArgumentException	The <i>index</i> array does not contain the exact type of arguments needed. The Set accessor of the property reflected by the current instance is not found.
System.Reflection.TargetException	The property reflected by the current instance is non-static, and <i>obj</i> is null or is of a type that does not implement the property reflected by the current instance.
System.MethodAccessException	The Set accessor of the property reflected by the current instance is non-public and the caller does not have System.Security.Permissions.ReflectionPermission to reflect on non-public methods.
System.Reflection.TargetParameterCountException	The current instance reflects an indexer and <i>index.Length</i> does not equal the rank of the indexer.

8
9
10
11

Permissions

Permission	Description
System.Security.Permissions.ReflectionPermission	Requires permission to reflect non-public members of a type in loaded assemblies. See System.Security.Permissions.ReflectionPermissionFlag.MemberAccess .

12
13
14

1 PropertyInfo.SetValue(System.Object, 2 System.Object, System.Object[]) Method

```
3 [ILASM]  
4 .method public hidebysig virtual void SetValue(object obj,  
5 object value, class System.Object[] index)  
  
6 [C#]  
7 public virtual void SetValue(object obj, object value,  
8 object[] index)
```

9 Summary

10 Sets the value of the property reflected by the current instance on the
11 specified object, using the specified index values.

12 Parameters

13
14

Parameter	Description
<i>obj</i>	The object whose property value is returned. Specify null to invoke a static Set accessor on the property reflected by the current instance.
<i>value</i>	A object that contains the new value for the property.
<i>index</i>	An array of objects that is an index or values for indexed properties. This value is required to be null for non-indexed properties.

15
16

17 Behaviors

18 As described above.

19 Exceptions

20
21

Exception	Condition
System.ArgumentException	<i>index</i> does not contain the exact type of arguments needed. -or- The Set accessor of the property reflected by the current instance was not found.
System.MethodAccessException	The Set accessor of the property reflected by the current instance is non-public and the caller does n

	have System.Security.Permissions.ReflectionPermiss to reflect on non-public methods.
System.Reflection.TargetException	The property reflected by the current instance is no static, and <i>obj</i> is null or is of a type that does not implement the property reflected by the current instance.
System.Reflection.TargetParameterCountException	The current instance reflects an indexer and <i>index.Length</i> does not equal the rank of the indexer.

1
2
3
4

Permissions

Permission	Description
System.Security.Permissions.ReflectionPermission	Requires permission to reflect non-public members of a type in loaded assemblies. See System.Security.Permissions.ReflectionPermissionFlag.MemberAccess .

5
6
7

1 PropertyInfo.Attributes Property

```
2 [ILASM]
3 .property valuetype System.Reflection.PropertyAttributes
4 Attributes { public hidebysig virtual abstract specialname
5 valuetype System.Reflection.PropertyAttributes
6 get_Attributes() }
7
8 [C#]
9 public abstract PropertyAttributes Attributes { get; }
```

9 Summary

10 Gets the attributes of the property reflected by the current instance.

11 Property Value

12

13 A **System.Reflection.PropertyAttributes** value that specifies the
14 attributes of the property reflected by the current instance.

15 Behaviors

16 This property is read-only.

17

18 This property gets a **System.Reflection.PropertyAttributes** value
19 that indicates the attributes set in the metadata of the property
20 reflected by the current instance.

21 Usage

22 Use this property to determine if the property reflected by the current
23 instance has a special name or a default value.

24

1 PropertyInfo.CanRead Property

```
2 [ILASM]  
3 .property bool CanRead { public hidebysig virtual abstract  
4 specialname bool get_CanRead() }  
5 [C#]  
6 public abstract bool CanRead { get; }
```

7 Summary

8 Gets a **System.Boolean** value indicating whether the property
9 reflected by the current instance has a **Get** accessor.

10 Property Value

11

12 **true** if the property reflected by the current instance has a **Get**
13 accessor; otherwise, **false**.

14 Behaviors

15 This property is read-only.

16

1 PropertyInfo.CanWrite Property

```
2 [ILASM]  
3 .property bool CanWrite { public hidebysig virtual abstract  
4 specialname bool get_CanWrite() }  
5 [C#]  
6 public abstract bool CanWrite { get; }
```

7 Summary

8 Gets a **System.Boolean** value indicating whether the property
9 reflected by the current instance has a **Set** accessor.

10 Property Value

11

12 **true** if the property reflected by the current instance has a **Set**
13 accessor; otherwise, **false**.

14 Behaviors

15 This property is read-only.

16

1 PropertyInfo.PropertyType Property

```
2 [ILASM]
3 .property class System.Type PropertyType { public hidebysig
4 virtual abstract specialname class System.Type
5 get_PropertyType() }
6
7 [C#]
8 public abstract Type PropertyType { get; }
```

8 Summary

9 Gets the type of the property reflected by the current instance.

10 Property Value

11

12 A **System.Type** that represents the type of the property reflected by
13 the current instance.

14 Behaviors

15 This property is read-only.

16