

# System.WeakReference Class

```
[ILAsm]
.class public serializable beforefieldinit WeakReference extends
System.Object implements System.Runtime.Serialization.ISerializable

[C#]
public class WeakReference: System.Runtime.Serialization.ISerializable
```

## Assembly Info:

- *Name:* mscorlib
- *Public Key:* [00 00 00 00 00 00 00 00 04 00 00 00 00 00 00 00]
- *Version:* 4.0.0.0
- *Attributes:*
  - CLSCompliantAttribute(true)

## Implements:

- **System.Runtime.Serialization.ISerializable**

## Summary

Represents a weak reference, which references an object while still allowing that object to be reclaimed by garbage collection.

## Inherits From: System.Object

**Library:** BCL

## Permissions

Permission	Description
<b>System.Security.Permissions.SecurityPermission</b>	for the ability to call unmanaged code. Demand value: System.Security.Permissions.SecurityAction.InheritanceDemand; Permission value: System.Security.Permissions.SecurityPermissionFlag.UnmanagedCode

## Description

A weak reference allows the garbage collector to collect an object while still allowing an application to access the object. If you need the object, you can still obtain a strong reference to it and prevent it from being collected.

# WeakReference(System.Object) Constructor

```
[ILAsm]
.method public hidebysig specialname rtspecialname instance void
.ctor(object target) cil managed

[C#]
public WeakReference (object target)
```

## Summary

Initializes a new instance of the `System.WeakReference` class, referencing the specified object.

## Parameters

Parameter	Description
<i>target</i>	The object to track or null.

## Description

This constructor creates a short weak reference to *target*.

# WeakReference(System.Object, System.Boolean) Constructor

```
[ILAsm]  
.method public hidebysig specialname rtspecialname instance void  
.ctor(object target, bool trackResurrection) cil managed  
  
[C#]  
public WeakReference (object target, bool trackResurrection)
```

## Summary

Initializes a new instance of the `System.WeakReference` class, referencing the specified object and using the specified resurrection tracking.

## Parameters

Parameter	Description
<i>target</i>	An object to track.
<i>trackResurrection</i>	Indicates when to stop tracking the object. If <code>true</code> , the object is tracked after finalization; if <code>false</code> , the object is only tracked until finalization.

## Description

If *trackResurrection* is `false`, a short weak reference is created. If *trackResurrection* is `true`, a long weak reference is created.

# WeakReference.Finalize() Method

```
[ILAsm]  
.method family hidebysig virtual instance void Finalize() cil managed  
  
[C#]  
~WeakReference ()
```

## Summary

Discards the reference to the target represented by the current `System.WeakReference` object.

## Description

To be added.

# WeakReference.IsAlive Property

```
[ILAsm]  
.property instance bool IsAlive  
  
[C#]  
public virtual bool IsAlive { get; }
```

## Summary

Gets an indication whether the object referenced by the current `System.WeakReference` object has been garbage collected.

## Property Value

`true` if the object referenced by the current `System.WeakReference` object has not been garbage collected and is still accessible; otherwise, `false`.

## Description

Because an object could potentially be reclaimed for garbage collection immediately after the `System.WeakReference.IsAlive` property returns `true`, using this property is not recommended unless you are testing only for a `false` return value.

# WeakReference.Target Property

```
[ILAsm]  
.property instance object Target  
  
[C#]  
public virtual object Target { set; get; }
```

## Summary

Gets or sets the object (the target) referenced by the current `System.WeakReference` object.

## Property Value

null if the object referenced by the current `System.WeakReference` object has been garbage collected; otherwise, a reference to the object referenced by the current `System.WeakReference` object.

## Description

After setting this property to the target object, make sure that there are no other strong references to the object; otherwise, it will not be collected. The object should also be set to null.

## Exceptions

Exception	Condition
<b>System.InvalidOperationException</b>	The reference to the target object is invalid. This exception can be thrown while setting this property if the value is a null reference or if the object has been finalized during the set operation.

# WeakReference.TrackResurrection Property

```
[ILAsm]  
.property instance bool TrackResurrection  
  
[C#]  
public virtual bool TrackResurrection { get; }
```

## Summary

Gets an indication whether the object referenced by the current `System.WeakReference` object is tracked after it is finalized.

## Property Value

`true` if the object the current `System.WeakReference` object refers to is tracked after finalization; or `false` if the object is only tracked until finalization.

## Description

If `true`, the weak reference is a long weak reference and `true` was specified for the *trackResurrection* parameter in the `System.WeakReference` constructor.