System.GC Class

```
[ILASM]
.class public sealed GC extends System.Object

[C#]
public sealed class GC
```

Assembly Info:

- *Name:* mscorlib
- *Public Key:* [00 00 00 00 00 00 00 00 04 00 00 00 00 00 00 00]
- *Version:* 1.0.x.x
- *Attributes:*
  - CLSCompliantAttribute(true)

Summary

Provides a mechanism for programmatic control of the garbage collector.

Inherits From: System.Object

Library: BCL

Thread Safety: All public static members of this type are safe for multithreaded operations. No instance members are guaranteed to be thread safe.

Description

[*Note:* The *garbage collector* is responsible for tracking and reclaiming objects allocated in managed memory. Periodically, the garbage collector performs a garbage collection to reclaim memory allocated to objects for which there are no valid references. Garbage collections happen automatically when a request for memory cannot be satisfied using available free memory.

A garbage collection consists of the following steps:

1. The garbage collector searches for managed objects that are referenced in managed code.
2. The garbage collector attempts to finalize unreferenced objects.
3. The garbage collector frees unreferenced objects and reclaims their memory.

During a collection, the garbage collector will not free an object if it finds one or more references to the object in managed code. However, the garbage collector does not recognize references to an object from unmanaged code, and may free objects that are being used exclusively in unmanaged code unless explicitly prevented from doing so. The **System.GC.KeepAlive** method provides a mechanism that prevents the garbage collector from collecting objects that are still in use in unmanaged code.

Implementations of the garbage collector should track the following information:

- Memory allocated to objects that are still in use
- Memory allocated to objects that are no longer in use
- Objects that require finalization prior to being freed

Other than managed memory allocations, implementations of the garbage collector should not maintain information about resources held by an object, such as file handles or database connections. When a type uses unmanaged resources that must be released before instances of the type are reclaimed, the type should implement a *finalizer*. In most cases, finalizers are implemented by overriding the **System.Object.Finalize** method; however, types written in C# or C++ implement destructors, which compilers turn into an override of **System.Object.Finalize**.

In most cases, if an object has a finalizer, the garbage collector calls it prior to freeing the object. However, the garbage collector is not required to call finalizers in all situations. Also, the garbage collector is not required to use a specific thread to finalize objects, or guarantee the order in which finalizers are called for objects that reference each other but are otherwise available for garbage collection.

In scenarios where resources must be released at a specific time, classes should implement the **System.IDisposable** interface, which contains a single method (**System.IDisposable.Dispose**) that is used to perform resource management and cleanup tasks. Classes that implement **System.IDisposable.Dispose** must specify, as part of their class contract, if and when class consumers call the method to clean up the object. The garbage collector does not, by default, call the **System.IDisposable.Dispose** method; however, implementations of the **System.IDisposable.Dispose** method can call methods in the **System.GC** class to customize the finalization behavior of the garbage collector.]

GC.KeepAlive(System.Object) Method

```
[ILASM]
.method public hidebysig static void KeepAlive(object obj)

[C#]
public static void KeepAlive(object obj)
```

Summary

Creates a reference to the specified object.

Parameters

Parameter	Description
obj	A **System.Object** that is not to be reclaimed by the garbage collector.

Description

The purpose of the **System.GC.KeepAlive** method is to ensure the existence of a reference to an object that is at risk of being reclaimed by the garbage collector prematurely.

The **System.GC.KeepAlive** method performs no operations and does not produce any side effects.

This method is required to be implemented in such a way as to prevent optimizing tools from omitting the method call from the executable code.

During program execution, after the call to the **System.GC.KeepAlive** method is executed, if there are no additional references to *obj* in managed code or data, *obj* is eligible for garbage collection.

GC.ReRegisterForFinalize(System.Object) Method

[ILASM]
```
.method public hidebysig static void
ReRegisterForFinalize(object obj)
```

[C#]
```
public static void ReRegisterForFinalize(object obj)
```

Summary

Requests that the specified object be added to the list of objects that require finalization.

Parameters

Parameter	Description
obj	The **System.Object** to add to the set of objects that require finalization.

Description

The **System.GC.ReRegisterForFinalize** method adds *obj* to the list of objects that request finalization before the garbage collector frees the object. *obj* is required to be the caller of this method.

Calling the **System.GC.ReRegisterForFinalize** method does not guarantee that the garbage collector will call an object's finalizer. [*Note:* For more information, see **System.Object.Finalize**.]

[*Note:* By default, all objects that implement finalizers are added to the list of objects that require finalization; however, an object may have already been finalized, or may have disabled finalization by calling the **System.GC.SuppressFinalize** method.]

Exceptions

Exception	Condition
System.ArgumentNullException	*obj* is a **null** reference.
System.ExecutionEngineException	The caller of this method is not the same as the object passed to this method.

GC.SuppressFinalize(System.Object) Method

[ILASM]

```
.method public hidebysig static void
SuppressFinalize(object obj)
```

[C#]

```
public static void SuppressFinalize(object obj)
```

Summary

Instructs the garbage collector not to call the **System.Object.Finalize** method on the specified object.

Parameters

Parameter	Description
obj	The **System.Object** whose **System.Object.Finalize** method will not be called.

Description

The method removes *obj* from the set of objects that require finalization. *obj* is required to be the caller of this method.

[*Note:* Objects that implement the **System.IDisposable** interface should call this method from the **System.IDisposable.Dispose** method to prevent the garbage collector from calling **System.Object.Finalize** on an object that does not require it.]

Exceptions

Exception	Condition
System.ArgumentNullException	*obj* is a **null** reference.
System.InvalidOperationException	The caller of this method is not the same as the object passed to this method.

GC.WaitForPendingFinalizers() Method

[ILASM]

```
.method public hidebysig static void
WaitForPendingFinalizers()
```

[C#]

```
public static void WaitForPendingFinalizers()
```

Summary

Suspends the current thread until the set of objects waiting for finalization is empty.

Description

System.GC.WaitForPendingFinalizers blocks an application until all objects that are awaiting finalization have been finalized.

When the garbage collector finds objects that can be reclaimed, it checks each object to determine the object's finalization requirements. If an object implements a finalizer and has not disabled finalization by calling **System.GC.SuppressFinalize**, the object is passed to the thread that handles finalization. The **System.GC.WaitForPendingFinalizers** method blocks until all finalizers have run to completion.