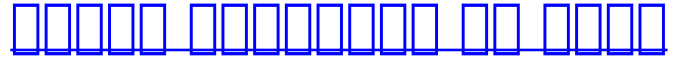


Managed Module Garbage Collection



Managed Module Garbage Collection

Managed Module Garbage Collection

Garbage Collection

### Leak memory

Leak Memory

new delete[]

### swap

swap

swap

swap

### memory page

memory page

C/C++

memory page

memory page

Committed

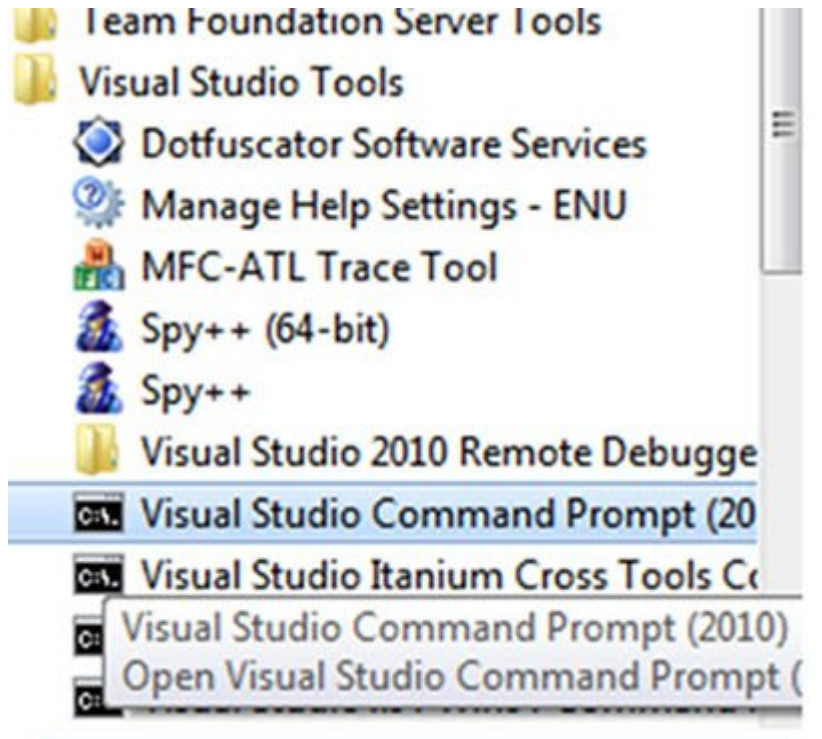












4

Visual Studio 2010 .Net 실행 파일을 실행할 때 corflags 값을 변경할 수 있습니다. corflags 값을 변경하는 방법은 Visual Studio 2010의 Visual Studio Command Prompt (2010)에서 corflags 명령을 사용하여 실행 파일을 실행할 때 CLR 실행 시점에 corflags 값을 변경할 수 있습니다. (Visual Studio 2010의 Visual Studio Command Prompt (2010)에서 corflags 명령을 사용하여 실행 파일을 실행할 때 CLR 실행 시점에 corflags 값을 변경할 수 있습니다.)

corflags c:\hld.exe

```

c:\Program Files (x86)\Microsoft Visual Studio 10.0\VC>corflags c:\hld.exe
Microsoft (R) .NET Framework CorFlags Conversion Tool. Version 4.0.30319.1
Copyright (c) Microsoft Corporation. All rights reserved.

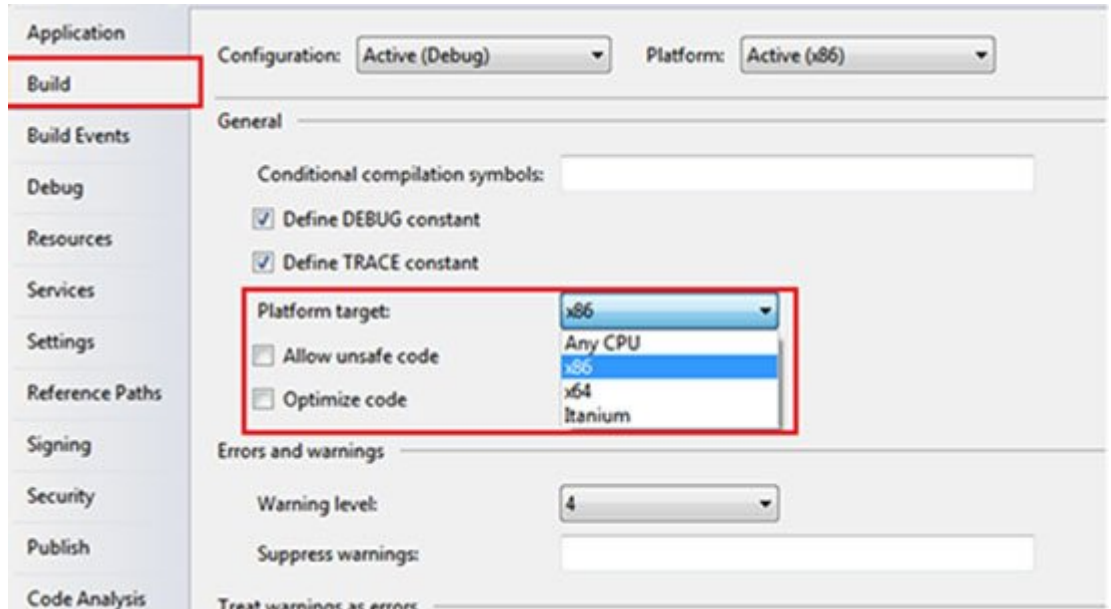
Version      : v4.0.30319
CLR Header   : 2.5
PE           : PE32
CorFlags     : 3
ILONLY      : 1
32BIT       : 1
Signed       : 0

c:\Program Files (x86)\Microsoft Visual Studio 10.0\VC>

```

5

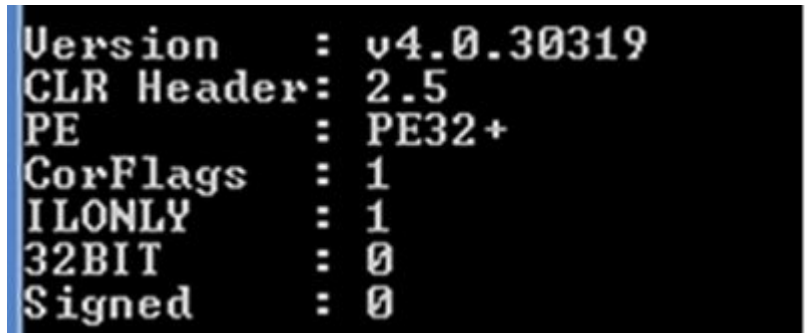
Visual Studio 2010의 Visual Studio Command Prompt (2010)에서 corflags 명령을 사용하여 실행 파일을 실행할 때 CLR 실행 시점에 corflags 값을 변경할 수 있습니다. Visual Studio 2010의 Visual Studio Command Prompt (2010)에서 corflags 명령을 사용하여 실행 파일을 실행할 때 CLR 실행 시점에 corflags 값을 변경할 수 있습니다. Visual Studio 2010의 Visual Studio Command Prompt (2010)에서 corflags 명령을 사용하여 실행 파일을 실행할 때 CLR 실행 시점에 corflags 값을 변경할 수 있습니다. Visual Studio 2010의 Visual Studio Command Prompt (2010)에서 corflags 명령을 사용하여 실행 파일을 실행할 때 CLR 실행 시점에 corflags 값을 변경할 수 있습니다.



6

platform target x64 platform target 64 corflags 64 7 64 32 Is64BitOperatingSystem true 64

Console.WriteLine(Environment.Is64BitOperatingSystem);

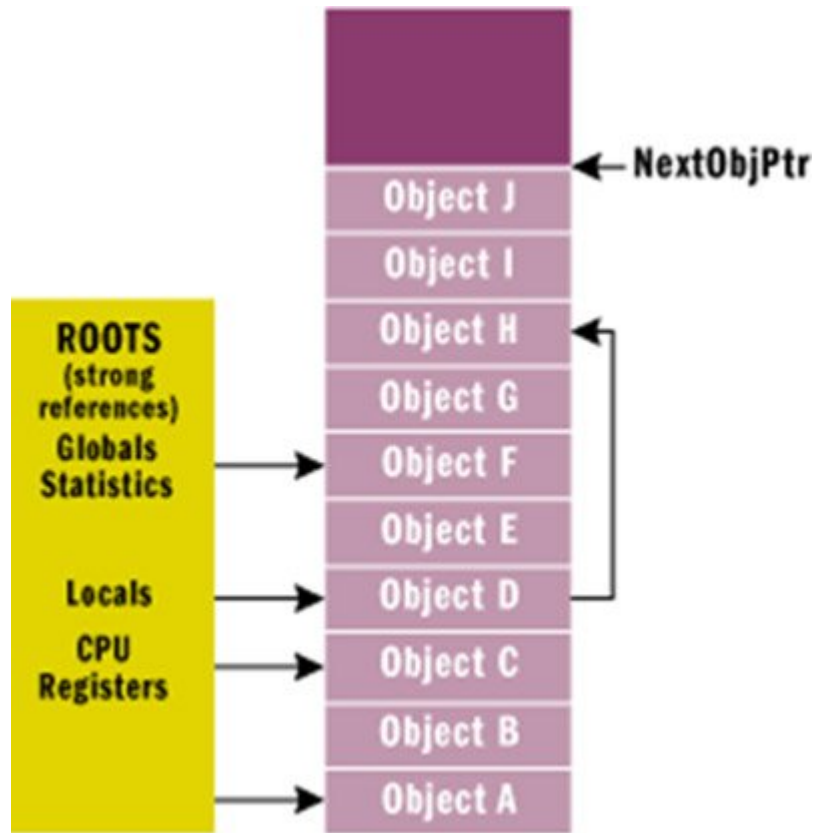


7

CLR garbage collection managed heap managed heap heap manage heap managed heap



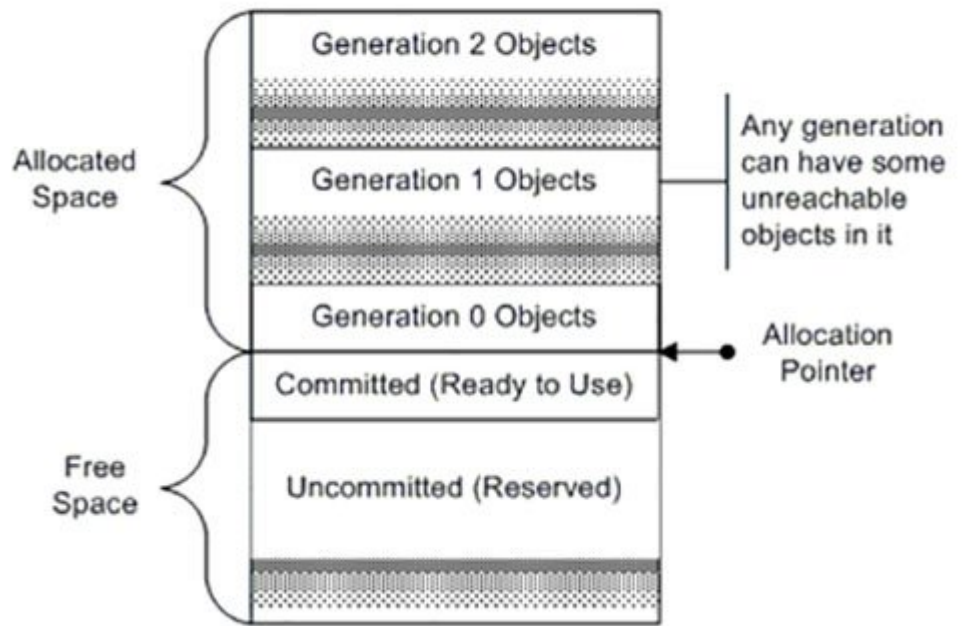
garbage collector managed heap base  
 collector (garbage collector) managed heap base  
 garbage collector managed heap base 8  
 Heap managed heap base 8



8

unmanaged heap managed heap base  
 runtime stack  
 Managed Heap base 9

## Simplified Model



9

CLR garbage collection is a managed memory management system. It automatically tracks the state of objects in memory and reclaims space for objects that are no longer in use. The CLR uses a generational garbage collection algorithm, which divides memory into generations based on the age of objects. The most recent objects are in Generation 0, and as objects age, they move to Generation 1 and then Generation 2. The CLR also uses a mark-and-sweep garbage collection algorithm to identify and remove unreachable objects. The garbage collection process is triggered when a certain threshold is reached, and it can be controlled using various parameters.

- 1 threshold parameter
- 2 garbage collection parameters
- 3 garbage collection parameters

### Garbage Collection

Garbage collection is a process of automatically identifying and removing objects that are no longer in use. The CLR uses a generational garbage collection algorithm, which divides memory into generations based on the age of objects. The most recent objects are in Generation 0, and as objects age, they move to Generation 1 and then Generation 2. The CLR also uses a mark-and-sweep garbage collection algorithm to identify and remove unreachable objects. The garbage collection process is triggered when a certain threshold is reached, and it can be controlled using various parameters.

The CLR uses a managed heap for memory allocation. The heap is divided into two parts: the managed heap and the unmanaged heap. The managed heap is used for objects that are managed by the CLR, and the unmanaged heap is used for objects that are managed by the operating system. The CLR uses a garbage collection algorithm to identify and remove unreachable objects from the managed heap. The garbage collection process is triggered when a certain threshold is reached, and it can be controlled using various parameters.



Out Of Memory Exception  
try/catch  
CLR

Out Of Memory Exception

Generations Weak Reference  
GC

Character User

Character User CUI  
Interface  
Graphic User Interface GUI  
32  
64  
32

:  
: 22:31 - 11/05/1394

:  
: 22:31 - 11/05/1394

:  
: 22:31 - 11/05/1394

:  
: 22:31 - 11/05/1394

:  
: 22:31 - 11/05/1394

:  
: 22:31 - 11/05/1394

:  
: 22:31 - 11/05/1394

:  
: 22:31 - 11/05/1394