Epsilon GC, also known as Java's no-op garbage collector, is an experimental garbage collector that provides memory allocation facilities but doesn't do any memory reclamation. This means that once the available Java heap is exhausted, the JVM will shut down.

Epsilon GC is designed for applications that know that they will not exceed a certain amount of memory. This can be useful for applications that are:

* Very short-lived
* Only allocate a small amount of memory
* Designed to run in a constrained environment

To enable Epsilon GC, you need to specify the following runtime switches on the command line:

-XX:+UseEpsilonGC

-XX:+ExplicitGCInvokesConcurrent

For example, to run a Java application with Epsilon GC, you would use the following command:

java -XX:+UseEpsilonGC -XX:+ExplicitGCInvokesConcurrent myapp

Epsilon GC is not a replacement for the other garbage collectors that are available in Java. It is a specialized tool that can be used in certain situations.

Here are some of the pros and cons of using Epsilon GC:

Pros:

* Lowest GC overhead possible
* No pauses for garbage collection
* Suitable for short-lived or memory-constrained applications

Cons:

* Does not reclaim memory
* Can lead to an OutOfMemoryError if the application exceeds the available memory
* Not suitable for long-running or memory-intensive applications

Overall, Epsilon GC is a useful tool for certain types of applications. However, it is important to understand its limitations before using it.