

Username: Pralay Patoria **Book:** Under the Hood of .NET Memory Management. No part of any chapter or book may be reproduced or transmitted in any form by any means without the prior written permission for reprints and excerpts from the publisher of the book or chapter. Redistribution or other use that violates the fair use privilege under U.S. copyright laws (see 17 USC107) or that otherwise violates these Terms of Service is strictly prohibited. Violators will be prosecuted to the full extent of U.S. Federal and Massachusetts laws.

More on the LOH

Earlier we talked about the SOH in terms of segments and virtual memory commitment. The LOH follows the same rules and will de-commit portions of a segment that aren't being used to store live objects. This takes place during a full GC, and [Figure 3.4](#) illustrates the process whereby Object B has become rootless and, after a full GC, is marked as free.

What actually happens is that, as mentioned in [Chapter 2](#), the occupied memory block is marked in the Free Space table as being available for allocation. The actual pages used in virtual memory are, in fact, reset, which means that they are not paged to and from disk.

The rest of the segment, from the last live object to the end of the heap, isn't committed until needed.

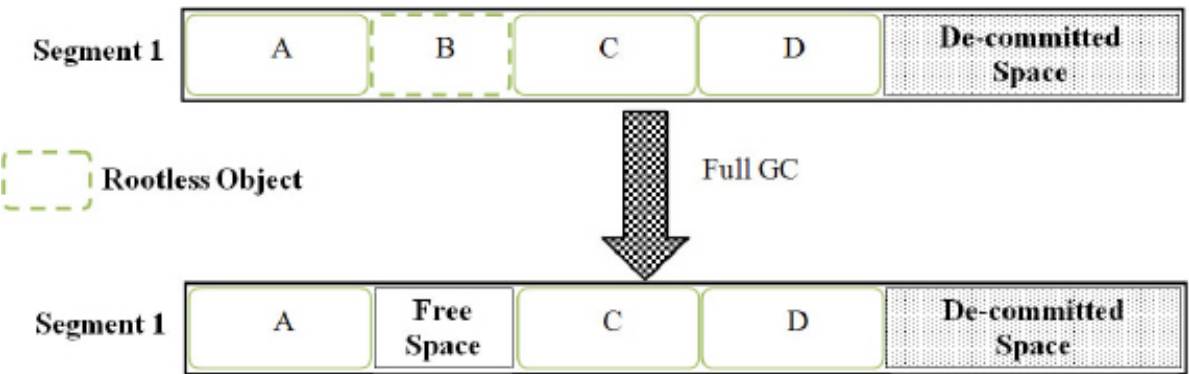


Figure 3.4: LOH memory model.