

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 USC 107) 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.

Excessive Writes and Hitting the Write Barrier

Writing to a memory address by modifying an old object causes a range of memory to be flagged as modified. The GC then treats those objects as roots it needs to analyze in order to determine if any objects need to be collected. Older objects aren't typically written to, so this is efficient in many programs. However, if a complex model is stored, as detailed in the previous section, and is modified, it will affect performance by forcing the GC to start tracking more objects.