

**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.

---

## Chapter 1: Prelude

The .NET Framework is a masterpiece of software engineering. It's so good that you can start writing code with little or no knowledge of the fundamental workings that underpin it. Difficult things like memory management are largely taken care of, allowing you to focus on the code itself and what you want it to do. "You don't have to worry about memory allocation in .NET" is a common misconception. As a result, .NET languages like C# and VB.NET are easier to learn, and many developers have successfully transferred their skills to the .NET technology stack.

The downside of this arrangement is that you develop a fantastic knowledge of language syntax and useful framework classes, but little or no understanding of the impact of what you write on performance and memory management. These "knowledge gaps" usually only catch you out at the end of a project when load testing is carried out, or the project goes live and there's a problem.

There are many thousands of applications out there written by developers with this kind of background. If you're reading this book, then you may be one of them, and so, to fill in those gaps in your knowledge, we're going to go on a journey below the surface of memory management within the .NET runtime.

You will learn how an application actually works "under the hood." When you understand the fundamentals of memory management, you will write better, faster, more efficient code, and be much better prepared when problems appear in one of your applications.

So, to make sure this goes smoothly, we're going to take it slow and build your knowledge from the ground up. Rather than give you all of the complexity at once, I am going to work towards it, starting with a simplified model and adding complexity over the next few chapters. With this approach, at the end of the book you'll be confident about all of that memory management stuff you've heard of but didn't quite understand fully.

The truth is, by the end of the book we will have covered memory management right down to the operating system level, but we will be taking it one step at a time. As the saying goes, "a journey of a thousand miles begins with a single step."