

.NET Memory Profiler

Key Facts

dotMemory helps you optimize memory usage in a variety of .NET applications and answer a whole bunch of questions related to poor memory management:

- What object causes a memory leak?
- Why is the object still in memory? What's holding it?
- What takes so much memory?
- How does garbage collection affect the performance of the app (e.g. high memory traffic)?
- Are there any memory allocation/distribution patterns violated?

Key Benefits

- **Easy and Comprehensive User Interface**
Memory profiling was always considered an operation for pros only. dotMemory unique user interface dramatically lowers the entry barrier and makes memory profiling pretty straightforward. All you do is constantly move from large to small—from set of objects to particular instances (real cause of memory issues).
- **Automatic Inspections**
To ease your life, dotMemory automatically checks the snapshot on most common types of memory issues. These inspections could be a great starting point in analyzing a snapshot if you don't know where to begin.
- **Multiple Views on Data**
Examine objects in the heap from multiple sides-views. Want to know how objects relate to each other? What objects do they reference and through what fields? Want to know what calls created these objects? No problem, dotMemory has a view for everything.
- **Comparing Memory Snapshots**
Comparing two snapshots is the main way to find objects that cause a memory leak. Use the comparison view to find out how many objects were created between snapshots and how many objects were collected.
- **Analyzing Memory Traffic**
Excessive allocations and garbage collections may imply significant memory management overhead. Use the traffic view to understand what objects are created/collected most intensively in your app and what functions cause this memory traffic.
- **Detect Memory Problems with Unit Tests**
Take advantage of dotMemory Unit, a free unit testing framework, to write tests that check your code for all kinds of memory issues. You can extend NUnit, MSTest or another .NET unit testing framework with the functionality of a memory profiler.
- **Remote Profiling**
Profile apps not only on your local computer but on any computer in your network or on the Internet. Remote profiling is especially helpful when you need to profile a web app on a production server.
- **Profiling API**
Taking the right moment for getting a snapshot is very important for memory analysis. Use the dotMemory API calls to take snapshots at the exact places of your code.
- **Support for Various .NET Apps**
Profile apps based on .NET Framework, .NET Core and Silverlight.