dotTrace

.NET Performance Profiler

Key Facts

dotTrace is a .NET performance profiler—that is, a tool that helps you find out exactly what causes a .NET application to run slower than expected. Locating performance bottlenecks in a .NET application is easy with dotTrace, thanks to a rich user interface and robust processing of large-scale snapshots.

Key Features

- **Support for Various .NET Applications**
  dotTrace helps you locate performance bottlenecks in a variety of .NET applications, including those based on .NET Framework, .NET Core, and Silverlight.
- **Multiple Profiling Modes**
  Choose sampling mode for fast profiling, tracing for high accuracy, line-by-line profiling to analyze complex business logic, and timeline mode to see how calls are distributed in time or optimize a multi-threaded application.
- **New Profiling Experience**
  Timeline profiling gives you a whole new profiling experience. Now, you can slice and dice data using filters, the call tree, or diagrams. The result of filters' work is a set of time intervals selected by specific condition. It's as easy as using a query to get data from some database.
- **Deep Visual Studio Integration**
  dotTrace is deeply integrated with Visual Studio, making profiling experience virtually seamless. You are able not only to start profiling of the applications you currently develop, but also view and analyze timeline profiling results right in the IDE.
- **Profile Static Methods**
  dotTrace allows you to instantly profile any static method in your code, right after you've written it. No need to build your solution, run it under the profiler, perform a usage scenario that runs your new code, etc.
- **Profile SQL Queries and HTTP Requests**
  Find out how a particular query affects application performance and why this query is running slow. If you develop a web application, then it will be useful to know how it processes incoming HTTP requests.
- **Remote Profiling**
  Profile remote applications without having to deploy the entire profiler infrastructure. This becomes very useful when you are trying to find and resolve performance issues on a machine that runs in production environment.
- **Profiling Unit Tests**
  If you have ReSharper installed on top of Visual Studio, you can easily start profiling unit tests from the code editor or ReSharper's unit test runner.
- **Forecasting Performance Optimization Results**
  Immediately estimate how removing a bottleneck would affect the overall application performance—without recalculating your snapshot.
- **Profiling API**
  Control profiling workflow from within the profiled application with the dotTrace profiling API.
- **Continuous integration support**
  Make profiling a part of your continuous integration builds. If your CI server is JetBrains TeamCity, you can use a special dotTrace plugin to TeamCity. It can help you prevent application performance regression by profiling integration tests. All you need is set a threshold for the test execution time, and if it's exceeded, the plugin will automatically take performance snapshot and fail the test.

www.jetbrains.com/profiler

THE DRIVE
TO DEVELOP