

How can I profile the performance of a .NET app already deployed to a customer?

Asked 15 years, 11 months ago Modified 15 years, 11 months ago

Viewed 2k times



3



I have several customers where my WinForms app does not perform as well as at other customers. I use JetBrains's dotTrace here in the office, but that does not help me find bottlenecks on the machines of our customers. How can I profile the performance of a .NET app already deployed to a customer? Are there any profiling tools with a redistributable component that lets me produce profiling results that cannot be used until I bring the results (a snapshot in dotTrace) back to the full profiling tool?

Update: Since [NProf](#) is freely distributable, I'm thinking about deploying it to a customer's workstation. I will report the results here.

.net

winforms

deployment

profiler

dottrace

Share

edited Jan 7, 2009 at 13:05

Improve this question

Follow

asked Jan 6, 2009 at 13:57



flipdoubt

14.4k ● 16 ● 66 ● 98

2 Answers

Sorted by:

Highest score (default)



3

I'm not familiar with dotTrace, but I have used the Visual Studio 2008 instrumentation and profiling tools in the context of Visual Studio and I believe they are distributed with .NET 3.5 SDK.



The VSInstr.exe tool: <http://msdn.microsoft.com/en-us/library/ms182402.aspx> will instrument the deployed binary.



VSPerfCmd: <http://msdn.microsoft.com/en-us/library/ms182403.aspx> can capture the performance data from the instrumented app.

I've never done it directly and solely from the command line, so I can't offer you much help there, but I figured this was another option for you, at least.

Share Improve this answer

answered Jan 6, 2009 at 16:09

Follow



ZeroBugBounce

3,670 ● 3 ● 33 ● 40



So I think what you might be needing is some method of measuring performance and deployment time. As ungraceful as they are I might look at adding

1



Performance Counters to your application. See [System.Diagnostics](#). I have used the extensively to monitor applications in deployment. I have found that profilers while awesome sometimes conceal certain problems due to their overhead. For instance if your application hits a database and the disk is slow causing queries to take longer that might not be apparent in a profiler since the application will be running slower in general.

For the client side if your app does not require administrative access on install I might make perf counters part of a "diagnostics mode" since installing them requires admin access. You would want to put perf counters to measure things like the duration of database and service calls, the size of data structures that can grow and the number of failures that occur.

We already know that your code works since on certain customers setups it works fine. So think about adding instrumentation to your application. It is one of those unglamorous but important elements of writing reliable software like good logging. It's not fun but in my opinion it needs to be done. It is often the only thing that gives you visibility into what your application is doing in deployment. The current application I am working on went from about 20 perf counters in Rev 1 to about 150 in Rev 2. I don't monitor most of them all the time but for alerting me to performance and reliability problems they have been well worth the time.

Share Improve this answer

answered Jan 6, 2009 at 16:11

Follow



Steve Severance

6,646 ● 2 ● 35 ● 44

I see the usefulness for memory usage and database speed, but not execution speed, such as what are the bottlenecks during app startup. Do you have any suggestions on using performance counters for app startup? – [flipdoubt](#) Jan 6, 2009 at 17:35
