

Realtime  
publishers

The Essentials Series: Improving Application  
Performance Troubleshooting

# Becoming Proactive in Application Management and Monitoring

*sponsored by*



by Don Jones

---

Becoming Proactive in Application Management and Monitoring..... 1

*Proactive Performance “Troubleshooting?”* ..... 1

    Monitoring the Entire Application, Not Just Components ..... 1

    Techniques for Proactive Application Management and Monitoring..... 3

    Faster Application Problem Resolution; More Business Value..... 4

## Copyright Statement

© 2009 Realtime Publishers. All rights reserved. This site contains materials that have been created, developed, or commissioned by, and published with the permission of, Realtime Publishers (the “Materials”) and this site and any such Materials are protected by international copyright and trademark laws.

THE MATERIALS ARE PROVIDED “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT. The Materials are subject to change without notice and do not represent a commitment on the part of Realtime Publishers or its web site sponsors. In no event shall Realtime Publishers or its web site sponsors be held liable for technical or editorial errors or omissions contained in the Materials, including without limitation, for any direct, indirect, incidental, special, exemplary or consequential damages whatsoever resulting from the use of any information contained in the Materials.

The Materials (including but not limited to the text, images, audio, and/or video) may not be copied, reproduced, republished, uploaded, posted, transmitted, or distributed in any way, in whole or in part, except that one copy may be downloaded for your personal, non-commercial use on a single computer. In connection with such use, you may not modify or obscure any copyright or other proprietary notice.

The Materials may contain trademarks, services marks and logos that are the property of third parties. You are not permitted to use these trademarks, services marks or logos without prior written consent of such third parties.

Realtime Publishers and the Realtime Publishers logo are registered in the US Patent & Trademark Office. All other product or service names are the property of their respective owners.

If you have any questions about these terms, or if you would like information about licensing materials from Realtime Publishers, please contact us via e-mail at [info@realtimepublishers.com](mailto:info@realtimepublishers.com).

# Becoming Proactive in Application Management and Monitoring

---

If you can find an application performance troubleshooting tool that automatically discovers and maps your application's various elements, monitors their performance in real time, and presents simple, intuitive views that help highlight performance problems for faster resolution—well, then you're ready to move beyond mere reactive troubleshooting. You're ready to start adding real value to your business, and ready to start *preventing* slow applications.

## Proactive Performance “Troubleshooting?”

Let's face it: Most companies regard IT as overhead. Technology is expensive, the people that support it are expensive, and it seems to need *so much* support sometimes. No matter how much you spend on IT, something always breaks—which is why so many companies have estimates of how much money an hour of downtime costs them.

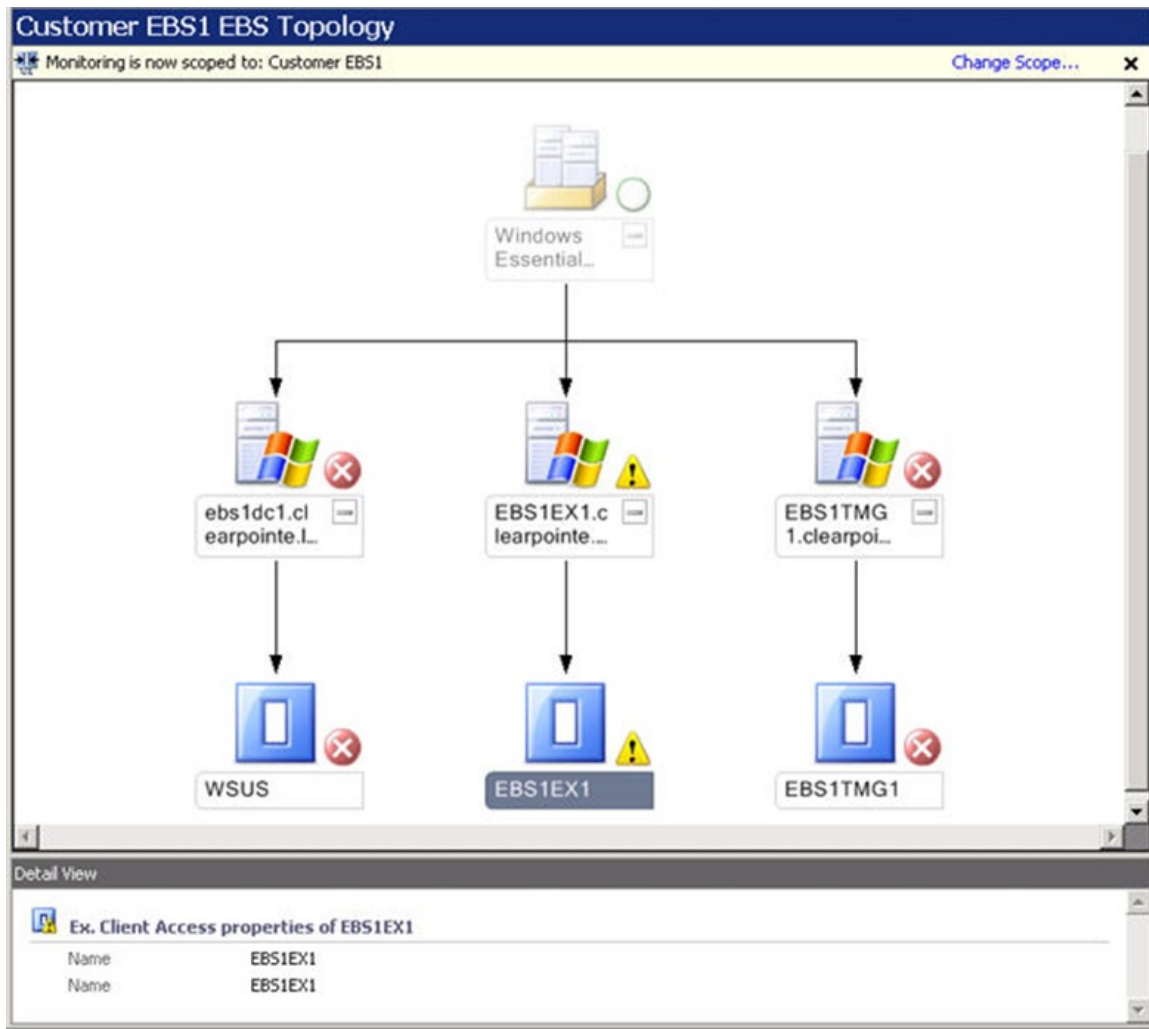
But when IT can start *anticipating* problems, solving them *before* they become noticeable, and *preventing* downtime—well, then IT has just become *valuable* to the business. And with the right toolset, you can do exactly that for slow applications.

The idea is simple: If you can have your toolset notify you when an application *starts* to exhibit out-of-bounds performance, you can immediately jump on the problem. You can identify the specific system elements that are causing the slowdown, fix or mitigate the problem, and keep the application's performance from ever slowing to the point where it significantly impacts your production components.

## Monitoring the Entire Application, Not Just Components

You may think that you have, or have seen, enterprise monitoring applications that do exactly what I'm talking about: They monitor the health of network services such as email, and they know how to also monitor dependent elements that support the service. All true, but they are typically neither automatic nor capable of a sufficient level of complexity. Let me explain.

The idea of monitoring multiple elements of a system is definitely nothing new; enterprise monitoring tools have been doing so for years. Figure 1, for example, is the type of display many enterprise monitoring consoles might offer, showing various network services—such as an email server—and their dependencies. When a given element exhibits out-of-bounds performance, the system can flag it—and flag anything that depends upon that element. It helps trace the root cause of the problem, and helps direct efforts to the right location to solve the problem more quickly.



**Figure 1: Enterprise monitoring tool.**

But there are a couple of things that these enterprise monitoring systems often lack: First is *automation*. If you're going to be monitoring an application, rather than just a network server or a network service such as email, you don't want to have to dig through the various application elements and dependencies. That should be done automatically by *observing* the application to ensure that you're getting *all* the application's elements.

Second is *complexity*. An email service may depend on a server, which may depend on a network, which may incorporate a router. Fine—that's within the realm of an enterprise monitoring tool, which can watch each of those elements. But when you dive into an individual *application*, things get a *lot* more complex. You've got database calls, multiple sub-components, and other minutiae. These elements span multiple disciplines—network, software development, database, infrastructure, and more. A special kind of tool is called for, one that is specifically designed to deal with complex applications.

But that special kind of tool *can* provide some of the same advantages that an enterprise monitoring system would provide: It can understand performance *thresholds* so that it knows what levels of performance are considered “good,” “borderline,” and “bad.” When real-time performance monitoring shows elements’ performance moving beyond the border of “good,” it can send out email or pager alerts. In fact, such an application monitoring tool might even be able to raise alerts *into* an enterprise monitoring tool, helping keep the entire IT organization informed of a problem.

## Techniques for Proactive Application Management and Monitoring

I’ll add one item to my wish list from the previous article: *reporting*. If your application performance monitor is already collecting performance data in real time, there’s no reason it can’t store that data and produce management reports—much as an enterprise monitoring solution might produce reports for overall network performance over time. Figure 2 shows what such reports might look like.

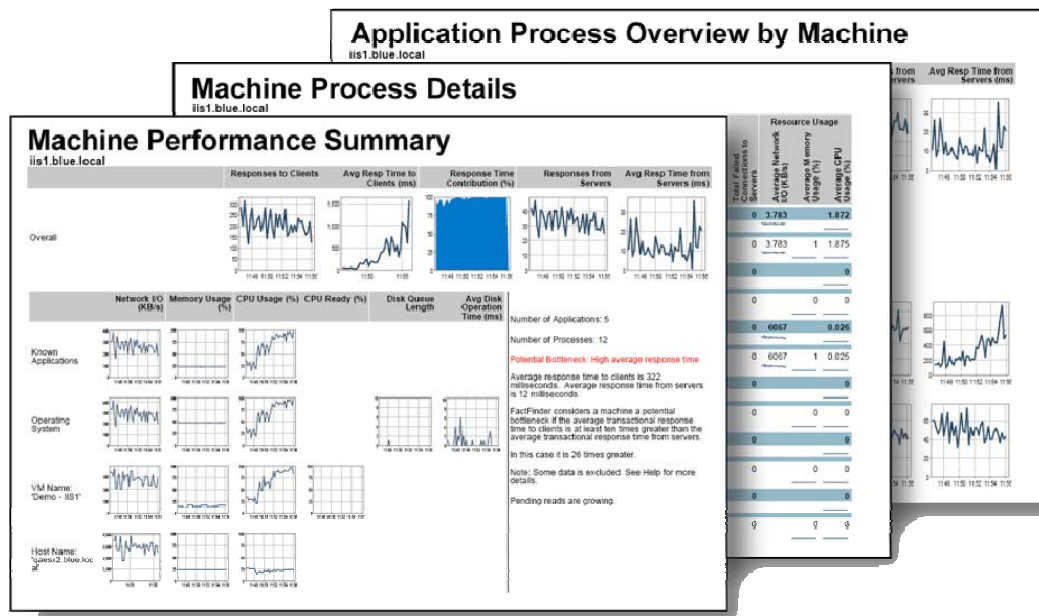


Figure 2: Application performance reports.

Again, these help move you from the realm of reactive troubleshooting and into the better world of proactive application *management*.

Another valuable item for our wish list: *load curves*. These are designed to show your application's overall performance as workload increases. Your developers may have produced something like this during application development, but in reality, those are just estimates and best guesses. Truly usable load curves come from monitoring application performance, down to the system element level, in a real, production environment. A properly-done load curve will show you how much workload your application is capable of handling while still exhibiting acceptable performance, and how much workload it can handle before the application grinds to a complete halt. Those curves can help you project application growth, plan for application expansion, and stay ahead of the curve—pun intended—so that the application is always ready to handle whatever the business needs to throw at it.

### Faster Application Problem Resolution; More Business Value

Most organizations I've worked with are still firmly in what I call the "silo world:" When an application slows down, everyone tackles their own independent elements, trying to guess where the problem might be. That approach is largely driven by the silo-centric tools that IT experts rely upon; change the tools, however, and you change the world. With tools that focus on the *entire application*, rather than on specific elements, the root cause of performance problems jumps right out at you.

That means the first tier of technical support can not only verify a problem but also identify the specific silo needed to fix the problem. That silo can be engaged to solve the problem—letting everyone else in the IT team continue working on their own valuable projects. There's no guessing; the right experts can move on to fixing the problem.

With that kind of capability in place, you not only significantly reduce application downtime but also enable a more proactive style of application management—one that offers real additional value to the business as a whole. Start seeing problems *before* they result in downtime. Start fixing those problems before anyone notices. Start *managing* the application over the long term with performance statistics, reports, and load curves—staying in front of the business, rather than running behind, trying to catch up.

It all comes from having the right tools: Ones that look at the *entire* application, automatically discovering the individual elements of a system and intuitively displaying performance in real time. *That's* efficient troubleshooting.