

How 5 enterprise companies maintain optimal performance in their .NET environments

APPDYNAMICS

# How 5 enterprise companies maintain optimal performance in their .NET environments

## Introduction

Why .NET and Azure need modern, dynamic APM ..... 3

## Chapter 1

How ExactTarget gains visibility into previously undetected performance issues—and fosters a DevOps culture ..... 5

## Chapter 2

How OpenTable found its application performance bottlenecks without having to set thresholds ..... 7

## Chapter 3

How COVIS learned to enjoy troubleshooting application performance ..... 9

## Chapter 4

How Reserve Bank of New Zealand identified .NET performance issues fast—and got its developers off the support desk ..... 11

## Chapter 5

How Macmillan reduced its MTTR on a complex Azure installation—and scale for the future ..... 13

## Conclusion

Why Microsoft enterprise solutions need modern-day APM ..... 14

# Introduction: Why .NET and Azure need modern, dynamic APM

When developers think about modern web-based applications, they often ignore those created within Microsoft's .NET framework. .NET may not be as flexible a framework as PHP or as nimble as Rails, and so it's often unfairly associated with antiquated infrastructures.

However, .NET, along with Microsoft's ASP.NET, are among the most widely used programming languages today. A 2010 Forrester Research survey cited .NET as the top development platform used by its respondents. In 2014, WhiteHat Security found that along with Java, .NET and ASP are the most widely used programming languages at 28.1% and 16% for the latter two respectively. According to WhiteHat, financial services companies use ASP over other languages by a 3-to-1 margin.

Furthermore, Capgemini observed in a 2012 report on cloud adoption in the enterprise that 73% of its respondents stated Microsoft's burgeoning Azure cloud computing platform was a key part of their cloud computing strategy, with strong backing across such verticals as:

- Financial services
- Retail
- Energy and public utilities
- Public agencies
- Telecommunications
- Media

And RightScale in its "2014 State of the Cloud" Report shows IaaS offerings from Azure leads other platforms in enterprise use, as well as interest.

.NET's popularity endures at large enterprises, where legacy infrastructure and applications run large parts of the business. Enterprises already have large investments in Microsoft, and by association with .NET professionals and systems. Most of today's major IT solutions include APIs with built-in .NET integrations.

Microsoft, which at heart is an enterprise company, has made it easy for its customers to implement .NET-based apps onto Azure by providing simple integration between its IDE (Integrated Development Environment) Visual Studio, which is used to develop on .NET, and Azure, making it painless to deploy .NET apps onto Azure. Ease of deployment is a key factor in the success of enterprise technologies, and Microsoft has created a great growth opportunity for itself by ensuring this integration through a familiar developer interface.

When you combine Microsoft's dominance in the enterprise space with its forward-looking app frameworks and cloud platform, you can see why offerings like .NET and Azure need modern, dynamic Application Performance Management (APM) solutions just as much as other frameworks, if not more so. These large enterprises need to run their applications seamlessly regardless of scale or complexity.

This AppDynamics ebook surveys several of our customers that leverage our platform's power and flexibility to run their .NET and Azure environments. We hope it gives you a stronger understanding why AppDynamics is the ideal APM solution to manage your .NET, Azure, and hybrid IT environments.

Chapter 1

How ExactTarget gains visibility into previously undetected performance issues—and fosters a DevOps culture

# Chapter 1: How ExactTarget gains visibility into previously undetected performance issues—and fosters a DevOps culture

The Salesforce Marketing Cloud, more commonly known as **ExactTarget**, is Salesforce.com’s industry-leading 1-to-1 digital marketing platform. Founded in 2000, ExactTarget provides businesses a “single view” to its customers via a wide range of personalized channels, including email marketing, social media like Twitter and Facebook, mobile messaging, targeted advertising, and content tracking.

Toyota, Microsoft, and other well-known brands rely on ExactTarget to deliver real-time communications and data-driven marketing campaigns across multiple channels to its customers. Mid-level brands, such as Deloitte and CNBC, leverage ExactTarget to execute predefined digital marketing campaigns. Even small businesses use ExactTarget’s welcome combination of intuitive drag-and-drop UI and global technical support to automate communications at every stage of the customer lifecycle.

ExactTarget’s environment is complex and rapidly changing, which makes delivering a consistently positive experience a challenge. Kevin Simininski, director of infrastructure operations at ExactTarget, needed to find an APM solution that could monitor all of ExactTarget’s apps in an almost exclusively .NET environment deploying thousands of VM guests, all of which service its entire application suite.

“We have a lot of moving pieces—and we have to keep everything moving at the same pace so that we can serve our clients as needed,” says Simininski.

Simininski’s team had to deal with wide and varied challenges, including a complex application that melded legacy components with a host of new technology features. “We were faced with the decision of trying to find a vendor that could cover all those facets of operations,” Simininski says.

Simininski looked at two APM providers, AppDynamics and a larger enterprise solution. Simininski says ExactTarget chose AppDynamics because its incredible agility. “We literally had AppDynamics working in about six hours on the first day of the proof of concept. In comparison, the other vendor still wasn’t working even after two weeks!” Simininski says.

Not only did AppDynamics provide deep code visibility into ExactTarget’s apps almost immediately, Simininski’s team was able to identify previously undetected performance issues. “The day we went live we were given access and visibility into a lot of things our operations team didn’t realize were happening behind-the-scenes inside our application,” Simininski says. “AppDynamics immediately exposed some low-hanging fruit that we could pick off and get tremendous gains in how our application runs.”

In addition, the use of AppDynamics helped ExactTarget foster closer, DevOps-style relationships between its operations and development teams. “We were immediately able to integrate and work very closely with our development team to solve very complex problems that had been around for years inside our application,” Simininski says.

This new DevOps collaboration between development and operations at ExactTarget has provided untold benefits for the entire ExactTarget team. “When I started, we had two very separate organizations,” says Simininski. “Today we have teams that collaborate together to try to solve difficult problems. So we now have a very robust two-way dialogue with our development organization as well as our infrastructure organization.” Simininski notes data generated by AppDynamics was the primary driver of this dialogue.

For his part, Simininski would recommend AppDynamics to any organization with a mission-critical operations center, including QA test teams working with developers to test code before it rolls into production. “They’re clearly an audience who can benefit greatly from AppDynamics APM,” he concludes.

## Chapter 2

How OpenTable found its application performance bottlenecks without having to set thresholds

## Chapter 2: How OpenTable found its application performance bottlenecks without having to set thresholds

**OpenTable** has been the world's leading provider of online restaurant reservations since its inception back in 1998. The company seats more than 14 million diners per month via online bookings across approximately 31,000 restaurants throughout the United States, Canada, Germany, Japan, Mexico, and the UK. In addition to the company's website and mobile applications, OpenTable powers online reservations for nearly 600 partners, integrating its services with such well known services as Apple Pay and Google Maps.

OpenTable runs primarily a .NET environment, as it has since in some form since 2000. The company requires reliable, continuous delivery of its applications because its primary competitor, the humble telephone, can't be beat for its simplicity. OpenTable one-ups the telephone by offering almost instantaneous reservations (without hold times or attitude from those taking the calls), along with APIs that offer the ability to add that event to a calendar, email other guests, and obtain directions to the restaurant.

None of those features matter if OpenTable's apps aren't fast or convenient to use, says Alan Novitskiy, performance lead at OpenTable. "People who come online for reservations expect a faster, more convenient experience. If we don't deliver it for them, they'll go back to using their phone, possibly for a long time, and we'd really be losing out on that business," he explains.

Because speed and reliability is crucial to its success, OpenTable puts every one of its applications through several levels of testing. These include unit tests and automated end-to-end tests in multiple browsers, as well as continuous performance measurements. Until 2013 however, OpenTable lacked full visibility into its apps and was having difficulty getting to the root cause of a given performance problem, even with its rigorous pre-production testing. "We performed thorough code analysis and rigorous benchmarking in pre-production environments, but when we released code into the wild all kinds of new variables were introduced, and we didn't have full visibility into what those different variables were," Novitskiy remembers.

After testing several other APM solutions, Novitskiy says his team tried AppDynamics hoping it would provide the hoped-for visibility OpenTable needed to maintain and enhance application performance. OpenTable first put AppDynamics through its benchmarking environment to verify that its overhead stayed within promised numbers. Once AppDynamics was found to have low overhead, Novitskiy's team installed AppDynamics on a single node in its production environment to see what type of data it would produce.

Novitskiy was pleasantly surprised. AppDynamics detected issues automatically without needing specific guidance from Novitskiy and his team. "It's really able to tell you about problems before you even know they exist," Novitskiy says. "With other solutions, you have to point them specifically at areas that you're interested in monitoring, but AppDynamics is really good about detecting key areas of your application and collecting statistics about them."

And unlike competing solutions, AppDynamics took performance averages and flagged potential outliers without requiring predefined and one-dimensional static thresholds, such as the number of milliseconds a given transaction is supposed to take. Without any special configuration, AppDynamics could grab an average of numbers, compute the standard deviation, and most importantly, let Novitskiy's team know if when an app was underperforming.

Within the first few months after it was deployed in OpenTable's production environment, AppDynamics provided the deep visibility and insight Novitskiy was seeking and even identified new opportunities to optimize OpenTable applications. As a result, OpenTable has a much clearer picture of its systems and the way everything interacts—and discovered several specific issues it could address quickly.

Adds Novitskiy: "You should let your application monitoring solution tell you about problems you haven't thought about. We often tend to focus on those key business transactions that directly contribute to the bottom line, and sometimes we fail to see the other things in the system that could be affecting performance. If your solution can give you a better, clearer picture of your whole system, that's really powerful knowledge."



Chapter 3  
How COVIS learned to enjoy  
troubleshooting application performance



# Chapter 3: How COVIS learned to enjoy troubleshooting application performance

Founded in 1983 as a full-service IT provider, Dusseldorf, Germany-based **Dr. Glinz COVIS GmbH** specializes in developing enterprise-level CRM solutions for large and medium-sized enterprises, including Deutsche Post DHL and automobile paint and finishing expert Standox. Its end-to-end CRM applications are designed to be used across large organizations, from the service desk to production, through a single intuitive UI. The largest of COVIS' installations handles upwards of 200,000 transactions an hour during peak times, files over 14 million customer questions and complaints per year, and services more than 4,000 users within client organizations.

COVIS needed more from its .NET-based applications than just keeping them up and running. They needed to have instantaneous response times even under the heaviest loads. Furthermore, Berger's team needed to be alerted whenever its applications weren't meeting its SLAs and then ideally have the deep code visibility to troubleshoot and fix any performance problem in under 10 minutes.

"With our distributed architecture, communication between the front-end, caches, database and services has to be robust, scalable and fast, as nearly every request touches more than one of these components," notes Kristof Berger, chief software architect at COVIS.

Increasingly, however, COVIS struggled to keep its MTTR (Mean Time To Repair) under that 10-minute mark. Its developers were limited to trace tools to test and resolve performance problems. None of these point tools could provide a unified overview of an app's .NET-based distributed environment, and so determining the root cause of a problem required engineers to puzzle together what limited information they could provide. Not only was this a poor use of human resources, it was a slow, ineffective way to troubleshoot application performance.

"In development you always have a rather reduced perspective of what the application does. In production there usually is no application- or performance-centric monitoring, only infrastructure monitoring, which in most cases is misleading. A server in good health might still perform miserably due to miserable code," Berger says.

COVIS needed an APM tool that offered complete visibility across its distributed environment while providing real-time baseline knowledge into the way specific components were functioning. And this solution needed to work seamlessly in a live production environment.

COVIS evaluated AppDynamics along with another APM vendor. According to Berger, COVIS quickly dropped the other vendor from contention because it was confusing to use and failed to supply visibility into COVIS' complex infrastructure, let alone offer insight into performance bottlenecks.

In contrast, AppDynamics delivered exactly what Berger sought: a solution that could monitor transactions across all COVIS' tiers, gathering and storing contextual information for later analysis, while also providing an application-centric perspective on how a given application performs both in test and in production across its environment. "The AppDynamics platform delivered just that: no-time installation, transaction detection, the possibility to leverage instrumentation with custom rules, and a performance impact analysis that can easily be assigned a dollar value," Berger says.

According to Berger, the AppDynamics solution was up and running in a matter of minutes. "The controller took 20 minutes, the agents took five minutes and configuring the app agents took 20 seconds. Smaller environments can be done manually in less than half an hour, and for larger ones you can create custom rules pretty simply. Your system can be fully configured within an hour," he says.

Thanks to AppDynamics, COVIS was able to reduce its MTTR to that desired under 10-minute mark. "Finding root causes for errors or bad performance used to be a nightmare, and most of our development team believed this would completely change when using another tool," Berger says. "Well, it did. As soon as the AppDynamics agents started monitoring the application's components, we had a crystal-clear view of what happened, how long it took and which components were bottlenecks in the specific scenarios."

Moreover, AppDynamics helped COVIS quickly resolve persistent performance issues. For example, one of the biggest application problems COVIS faced was its business rule engine, which served about 1,000 calls per minute, but wasn't performing as Berger's team expected. The AppDynamics solution enabled his team to discover within five minutes that the engine was recompiling each rule on every call instead of using the precompiled versions, which was putting a heavy load on the machine. They made the fix in 30 seconds, deployed it a minute later, and within two minutes AppDynamics let them verify they had solved the problem.

AppDynamics has provided value not just for Berger, but also the entire IT organization at COVIS. "Our developers can analyze user problems much better now, as they always have the information about what happened at their fingertips. Our service desk is able to proactively monitor the applications, and then give the developers the exact location of a problem," Berger says.

The AppDynamics solution also is being used in the COVIS test environment. "With every build, we ran load tests to verify our solutions, and we had direct feedback from the AppDynamics platform about whether these solutions changed the performance for the better, or not," Berger says. "Running the AppDynamics solution within our test infrastructure did not influence the system's performance too much (only about 2-3% according to our load tests), and it enabled us to completely analyze what happened."

Adds Berger: "I never thought I'd say this, but I've actually started looking forward to problems, since they're so easy to find and fix with AppDynamics."

## Chapter 4

How Reserve Bank of New Zealand identified .NET performance issues fast—and got its developers off the support desk

# Chapter 4: How Reserve Bank of New Zealand identified .NET performance issues fast—and got its developers off the support desk

Established in 1934, **The Reserve Bank of New Zealand (“RBNZ”)** manages New Zealand’s monetary policies by maintaining stable pricing and keeping its financial system in order. RBNZ also prints and distributes New Zealand’s physical currency and oversees its payment systems.

RBNZ runs dozens of .NET-based applications that are critical to the bank’s daily operations and long-term objectives. But slow transactions and stalls related to .NET services were plaguing the bank. Because time-series data was calculated on the fly in application logic and memory, rather than using stored procedures in the database, many requests took too long to complete.

Greg Perrott, software architect at RBNZ, is responsible for the performance of several critical financial applications. His team of 15 developers handles support, maintenance, and enhancement of existing applications, as well as development of new applications. The problems RBNZ was experiencing with poorly performing apps meant that too often the development team was hijacked from its core job of developing new apps and refining current ones that would benefit the bank to support help desk issues.

“Our ability to deliver enhancements was greatly affected by support,” Perrott explains. “There were occasionally periods of a week or more when no development progressed because of ongoing support issues.”

Moreover, Perrott’s team found they had to extend the time-out period for the client in order to reduce the number of failed transactions. “It wasn’t ideal because the end user still had to wait, but at least that way the application would eventually return data,” Perrott says.

Targeting these transactions was an ongoing project for Perrott, but without adequate visibility into the application, it was a daunting task. “Without access to production data we simply couldn’t find which areas we needed to redesign in order to improve the slowest services and scale the application,” Perrott says.

Perrott needed to find an APM solution that could solve the immediate problem of getting to the root cause of performance issues and provide visibility across its entire app infrastructure, and it needed to provide comprehensive support of the .NET framework. He began a 30-day trial of AppDynamics, and the timing couldn’t be better. RBNZ was in the midst of another IT problem—clients disconnecting from RBNZ services—a timely test for the solution. “We had very little information about what was causing the problem, and we weren’t able to replicate it in any other environment,” Perrott says.

During the trial period, Perrott installed the AppDynamics platform in the production environment. Almost immediately he found the issue and its root cause through the deep code visibility AppDynamics provides, along with its ability to perform call graphs of code execution. “That pretty well sold us on enlisting the solution full-time,” he said.

The AppDynamics platform also helped identify issues that were going unreported before. “I started finding things that clearly were impacting end users, but either they hadn’t reported them or had reported them in such a way that we were unable to identify the problem,” Perrott says. “Now we’re able to fix those things and we’re being more effective in our optimization. We’re targeting our .NET services with the most load and the longest response times.”

The result of this additional visibility is that Perrott and his team spend significantly less time troubleshooting than before. Before AppDynamics, Perrott indicated the team would spend about half of their time on support and QA, which shortchanged enhancements and other development projects. Perrott estimated that the AppDynamics platform has provided RBNZ with over \$38,000 in productivity savings in the first year by reducing the time developers spend on support activities. Furthermore, AppDynamics helped Perrott’s team build stronger arguments for getting additional resources when needed.

“With AppDynamics there has definitely been an improvement in our ability to provide support, our ability to upgrade the system, and our ability to be more proactive,” Perrott says. “Now we’re aware of problems and can fix them before they impact the business.”

## Chapter 5

How Macmillan reduced its MTTR on a complex Azure installation—and scale for the future

# Chapter 5: How Macmillan reduced its MTTR on a complex Azure installation—and scale for the future

**Macmillan Publishing Group**, a division of Stuttgart-based Holtzbrinck Publishing Group, is one of the leading publishers of books, journals, and educational course materials in the world. A large portion of Macmillan’s business is made up of publishing top-of-the-line textbooks, which Macmillan provides to thousands of students around the world. In addition to paper textbooks, Macmillan also offers online course materials and activities to supplement students’ learning. Macmillan’s application teams in the UK are responsible for building, maintaining and supporting applications that are fast and scalable, no matter where in the world its users are.

To achieve the global scale required of its .NET-based applications, Macmillan decided to host them in Microsoft’s Azure Cloud. The company believed Azure would offer the most seamless integration for .NET applications, given that Microsoft developed both. Macmillan had data centers in all of the major areas the company served, and the company management concluded that hosting identical versions of the online learning applications in each data center would ensure students would have the same user experience using Macmillan materials, regardless of their location.

Because Macmillan’s .NET applications had complex architectures, those responsible for these apps struggled to identify performance bottlenecks using Azure’s monitoring tools. “The monitoring reports in Windows Azure are very high-level,” says James Graham, project manager at Macmillan. “We had difficulty determining whether issues were arising in the application or the network, let alone which service they were coming from.”

Graham’s team needed to find an APM tool that could give them visibility of all their applications running in Azure. They wanted a modern, dynamic solution designed to monitor all aspects of a complex, distributed .NET application within Azure; however, few tools supplied that necessary combination of dynamic, unified APM that could also provide visibility into a Microsoft-centric environment in real time, let alone provide deep insight into .NET function calls.

According to Graham, AppDynamics was far and away the best choice for Macmillan’s needs. Graham’s team found AppDynamics to be very easy to install and deploy on Azure. Within a few hours of getting the license keys from AppDynamics, Macmillan’s developers successfully identified a problem they had been trying to locate for over a month.

“We realized that one of our database queries was pulling back a large amount of data from the database, but only serving a small amount,” says Graham. “This was causing unnecessary overhead, and slowing down the application for the end user.”

Without the AppDynamics solution, Graham doesn’t think his team would have been able to find the root cause of the problem. “AppDynamics APM shows us how the different services interact, and allows us to sort through the data to fix the real problems,” Graham says.

Graham initially used the AppDynamics platform as a troubleshooting tool. But once his team’s average MTTR (Mean Time To Repair) dropped from weeks to hours, Graham started leveraging AppDynamics’ ability to glean insight into areas for optimization within Macmillan’s applications.

“Every month we look at the data we get from AppDynamics APM and figure out what code needs to be refactored, or what optimizations need to be done in the database, to improve performance,” Graham says. “We can also take that data and figure out how our application would perform if we plan on scaling the application, say, from 80,000 to 500,000 students. It helps us plan what resources we’ll need in the future.”

AppDynamics also helped Macmillan gain visibility into third-party service calls and decrease response time from 15 seconds to less than three seconds. “Now we’re able to go back to third-party companies that our application interacts with and show them where the problems are. We’ve worked with some of our partners to optimize their services, which has resulted in high gains for us,” Graham says.

Adds Graham: “AppDynamics gives us a much better visualization of what’s going on under the hood in our applications. It shows you the difference between what you think you’re doing in the code and what’s actually being executed in the application runtime. Ultimately, that’s helped us improve the end-user experience for our students.”

## Conclusion: Why Microsoft enterprise solutions need modern-day APM

Of the five companies discussed in this ebook, only ExactTarget was founded after Y2K. Most enterprises that date from this era and before still use Microsoft products like .NET and Azure in at least part of their IT environment, typically along with older Windows servers and other enterprise standbys like SharePoint and Microsoft Office. These longstanding relationships with Microsoft aren't going to be jettisoned just because other frameworks are in vogue, particularly when Microsoft's offerings provide similar functionality and better integration with legacy infrastructure than other major vendors.

To assume that these organizations don't need the level of application performance management AppDynamics provides is ludicrous. If anything, they need modern-day APM even more given the level of customer trust they have established over so many years. While a newer, smaller concern may be able to weather a service outage (or avoid one given their infrastructures are in most cases less complex), these venerable organizations cannot afford failure despite their greater capacity for it.

By using AppDynamics, companies deploying .NET-based apps and services can run a complex suite of apps with thousands of virtual machines (like ExactTarget), automatically detect performance issues without requiring humans to set parameters (like OpenTable), and find the root cause of a problem—and fix it—within a few minutes (like COVIS). AppDynamics handles the overwhelming chore of maintaining and improving application performance, so that developers and other IT talent can focus on their core duties, rather than put out fire after fire (like Reserve Bank of New Zealand). And AppDynamics works with the same speed and efficiency whether you're running an internal .NET environment, a hybrid IT environment, or some combination thereof hosted by Azure.

APPDYNAMICS

[appdynamics.com](http://appdynamics.com)